(You can click the project title to go to that project)

Title	Domain/Category	Supervisor Email
AI Recruitment & Staffing Web Application	Web Based	shakeel@vu.edu.pk
An online Virtual Study Group Platform	Web Application	qamar@vu.edu.pk
Arts Gallery	Web Application	manahil.hassan@vu.edu.pk
AutoEase: Online Car Showroom System	Web Application	komal.saleem@vu.edu.pk
Buy and Sell Pre-Owned Goods	Web Application	hina.rafique@vu.edu.pk
Career Hub Website	Web Programming	ehsan.haq@vu.edu.pk
CARWASH	Web Application	aun.ali@vu.edu.pk
"Connect & Converse": A Web-based Interactive Discussion Hub for University Students	Web Programming	muhammad.saeed@vu.edu.pk
CPU Scheduling Algorithms Animated Simulator	Web Programming	amna.bibi@vu.edu.pk
Daycare Management System	Database	ammara.rasheed@vu.edu.pk
E Loan Application & Verification System	Web Programming	qaiser.shabbir@vu.edu.pk
Event Management System	Web Application	rehan@vu.edu.pk
Health Tracker: Monitor and Record Health Data	Web Based Application	asmabatool@vu.edu.pk
Online Maze Game for Learning	Web Programming	ahmadlodhi@vu.edu.pk
One Click Desert Providers	Web Application	shafaq.nisar@vu.edu.pk
Online Bank Loan Portal	Web Programming	ghulam.abbas@vu.edu.pk
Online Handmade Crafts Store	Web Application	soahilaamer@vu.edu.pk
Online Tutor Finding Application	Web Programming	akmalkhan@vu.edu.pk
PRMS: Patient Record Management System	Web Application	syed@vu.edu.pk
Consultant Appointment Booking System (CABS)	Web Based Application	asifhussain@vu.edu.pk
Derma Elixir Studio	Web Programming	komal.khawer@vu.edu.pk
Exam Seating Arrangement System	Web Application	nidaanwar@vu.edu.pk
Frozen Food Panda	Web Application	neelam.alam@vu.edu.pk
Identity Management User Interface (IDMUI)	SE/Infrastructure Management	arif.husen@vu.edu.pk
ONLINE BAKERY MANAGEMENT SYSTEM	Web Application	jibrankhan@vu.edu.pk
Online Book Exchange Platform	Web Based Application	fazitahir@vu.edu.pk
Online Cosmetics Store	Web Application	warda.fiaz@vu.edu.pk
Online Portal for Real Estate Management System	Web Application	mir.salam@vu.edu.pk
Online Tourist Guide	Web Application	kamran.qureshi@vu.edu.pk
Prayer Tracker	Web Programming	musaddiq.hussain@vu.edu.pk
Responsive PHP-Based Web Application for Image-to-Hex Conversion with Batch Processing, GIF Handling, and Image Library	Web Application Development / Embedded Systems Interface	khaqan@vu.edu.pk
Used Laptops Buying and Selling Store	Web Application	hassan.ali@vu.edu.pk
VU Alumni-Student ConnectBook	Web Programming	tooba.khan@vu.edu.pk
Web-Based Animated Simulator for Sorting Algorithms	Web Application	umra.naeem@vu.edu.pk
Web-based Gardening Care App	Web Application	haseebakmal@vu.edu.pk

An Intelligent Auto Grader Learning Management System for Academic Courses	Web Application	umarfarooq@vu.edu.pk
Anomaly Detection System in Blockchain	Blockchain/Machine Learning	fouziajumani@vu.edu.pk
Book Genre Classification using Text Analysis in Python	Text Analysis/Classification	rizwana.noor@vu.edu.pk
Developing a Career Counseling Platform Using Artificial Intelligence	AI/ML/NLP & recommendation Systems, Educational Technology	saima.munawar@vu.edu.pk
Emotion Recognition from Text in Android	Mobile Apps	imran.afzal@vu.edu.pk
Human Skin Disease Detection System Using CNN	Software Application	mehboob.ali@vu.edu.pk
MCQ-Based Test System for Exam Preparation Using Python and Django	Data Science / Web Application Development	m.kaleem@vu.edu.pk
PRODIGENIUS – AI POWERED TASK MANAGEMENT APPLICATION	Mobile App + Machine Learning	bilal.saleem@vu.edu.pk
Real-Time Deepfake Detection for Video Streaming Platforms	AI Web Application / Machine Learning	sonia.salman@vu.edu.pk
Implement a system of Heart Disease Detection Using Machine Learning	Machine Learning based project	anam.naveed@vu.edu.pk
Stationery Shop Management System	Desktop Application	asad.ullah@vu.edu.pk
IoT-Based Smart Aquarium Management System	Digital Logic Design & IoT	waqar.ahmad@vu.edu.pk
3D MRI Brain Tumor Segmentation	Artificial Intelligence/Image Processing	hashir.khan@vu.edu.pk
Advanced Face Recognition System with Real-Time Detection	Image Processing	madiha.hussain@vu.edu.pk
Automated Ripeness Detection of Fruits using Deep Learning	Deep Learning / Computer Vision	zaid.ismail@vu.edu.pk
Brain Tumor detection using MRI Scans	DL/AI/Web/Image Processing	m.luqman@vu.edu.pk
EyeGuard: Eye Strain Detection	Image Processing/Health and Wellness	sana.rao@vu.edu.pk
Face Mask Classifier	Image Processing/Deep Learning	umairali@vu.edu.pk
Food Image Classification	Machine Learning/Image Processing	fizzah@vu.edu.pk
Skin Cancer Detection	Image Processing/AI/Web App	taliah.tajammal@vu.edu.pk
Virtual Proctor	Image Processing	noor.rahman@vu.edu.pk
AI-Based Smart Teaching Assistant for Personalized Exam Preparation	Mobile Application / Artificial Intelligence (AI)	waqas.ahmad@vu.edu.pk
Android Based Expense Management System	Android Application	tahir.jan@vu.edu.pk
Campus Connect	Mobile Apps.	mohsin@vu.edu.pk
Campus Navigator	Mobile Apps.	nadiatabassum@vu.edu.pk
Event Management Mobile Application	Mobile Application	manwar@vu.edu.pk
Online family wear sale app.	Mobile Apps.	imran.akhtar@vu.edu.pk
Parent Portal Mobile App	Android Mobile Application	humairanaeem@vu.edu.pk
Personalized Nutrition and Fitness Management System for Android	Android Application	irshad.nasir@vu.edu.pk
Smart Solar Solution Android App	Mobile App	saeednasir@vu.edu.pk
NLP Chatbot Development using Dialogflow	Web Application/NLP	abdullah.qamar@vu.edu.pk
Hate Speech Detection using Machine Learning for Roman Urdu	Data Science / Machine Learning / Natural Language Processing (NLP)	tayyab.waqar@vu.edu.pk
Identifying Fake Product Reviews Using Opinion Mining	NLP/Information Retrieval	tayyaba.sehar@vu.edu.pk
Keyword-Based Search Engine for Text Documents	Information Retrieval	muhammad.ilyas@vu.edu.pk
Real-Time Network Intrusion Detection Using Wireshark and Advanced Ensemble Learning Techniques	Networking/Machine Learning/ Research	laraib.sana@vu.edu.pk

Expense Tracker Application	Web Application	salmanbashir@vu.edu.pk
Resources Monitoring for systems on Network	Desktop/Networking	asimmehmood@vu.edu.pk
Website for VU Sports Society	Web Application	abdur.rafay@vu.edu.pk
On-Demand Food Delivery App	Web Programming	adnanasif@vu.edu.pk
Modern E-Learning Platform	Web Application	amjad.iqbal@vu.edu.pk

AI Recruitment & Staffing Web Application

Project Domain / Category:

Web Based

Abstract / Introduction

If you run an Human Resource (HR) staffing company then you need a web application that enables you effectively manage all aspects of your agency's operations. With features such as Order Management, Employer and Candidate Management, Placements Management, User Records, Candidate Recruitment and lots more, your agency will never be the same again.
We will develop the web application with powerful AI features that leverage OpenAI's GPT models. AI Features will include candidate recommendations, candidate bio filters, employer job vacancy filters, contract generation, candidate bio generator, Job vacancy generator and more.

This AI recruitment web app will have a complete job request management module that makes it easy for employers to place orders for your candidates.

Functional Requirements:

Our Web Application will have deep integration with OpenAI's GPT models. Our app AI features will include following functional requirements:

1. Candidate Recommendations: Employers and Admins can now find the best candidates for a given role using the power of AI. Our Candidate Recommendation feature is a powerful tool that will change the way you recruit forever!

2. Candidate Bio filter: our candidate bio filter will drastically reduce the manpower needs of your agency. You can configure the AI model to filter each candidate's bio to remove any text that violates your terms of service.

3. Employer Job Vacancy filter: Our job vacancy filter will ensure that each time an employer creates a job vacancy, it will comply with your policies. You can even have the Gpt model automatically re-write each job posting to enforce quality standards.

4. Contract Generation: Automatically create contracts with the AI contract generation feature. The system will use the signatories you have configured to draft a professional legal document.

5. Candidate Bio Generator: Easily create professional bios for your candidates with the click of a button. The system will gather all the candidate's data and draft a professional bio.

6. Job vacancy generator: Employers and admins can use AI to easily create professional job listings.

7. Blog post generator: Easily create blog posts with the built-in blog post generator.

8. Email creation: Save time drafting emails with the built-in email template creator

9. Roles Supported: 3 roles supported a) Candidate b) Employer c) Administrator

10. Vacancies posting on Portal: Easily post vacancies on your portal. Receive applications for each vacancy from your candidates. Download resumes for each applicant. This app also provides powerful filtering features for selecting the right candidates for each position.

11. **Employers can optionally shortlist candidates while placing orders.** You get to define candidates that are available for shortlisting on your front end. You can also create orders from your backend and shortlist candidates yourself.

Tools: **AX**

Programming Language: Python, Java.
Databases: SQLite, Couchbase, HBase, PostgreSQL, Maria DB, MySQL and Microsoft SQL Server.
IDE: Jupyter, PyCharm, Visual Studio, Atom, Spyder, Google Colab

Supervisor:

Name: Shakeel Saeed
Email ID: shakeel@vu.edu.pk
Skype ID: shakeelsaeedvurnd

An online Virtual Study Group Platform

Project Domain / Category

Web Application

Abstract/Introduction

The proposed project aims to develop an efficient Virtual Study Group Platform aims to facilitate collaboration among students by providing a web-based solution for organizing and participating in study groups. This platform will enable students to create groups, schedule study sessions, share resources, and engage in discussions, enhancing their learning experience and promoting effective teamwork. The solution will have both an Admin view and a User view, ensuring that both can access the features and functionalities that they require. This web application will be developed using ASP.NET / PHP as the front end and SQL Server / MYSQL as the back-end. The database management system will store all necessary information about the items and their related information.

Functional Requirements:

1. **User Authentication:** The system should provide secure registration and login system along with profile management with study preferences.
2. **User Views**
    - **Group Management:** The system should provide an interface to create and join study groups easily. It should also provide an effective mechanism to manage group members and their rols.
    - **Scheduling:** The developed system should be capable of scheduling study sessions with automated reminders. Integration with Google Calendar is also required.
    - **Resource Sharing:** The system should provide a convenient interface to upload and share study materials and notes. It should also provide facility to organize resources by topics or subjects.
    - **Communication Tools:** The platform should provide discussion boards for asynchronous communication.
    - **Notifications:** The should be capable of sending alerts for scheduled sessions and new resources.
3. **Admin View**
    - **User Management:** The platform should can approve or deny user registrations. It should also monitor user activity and group interactions.
    - **Content Moderation:** The system should review and manage shared resources and discussions.
    - **Reporting:** The system should generate reports on platform usage and engagement metrics.
    - **System Configuration:** The system should also be capable of managing platform settings and integrations.

Non-Functional Requirements:

- **Security:** The solution will implement proper security measures to ensure that customer and bakery owner data is protected.
- **Performance:** The system will be designed to handle a large number of concurrent users and transactions, ensuring that the platform is always available and responsive.
- **Scalability:** The system should be easily scalable to accommodate new features and functionalities as the business grows.
- **Usability:** The solution will be designed to be user-friendly and intuitive, ensuring that customers and bakery owners can easily navigate the system and complete their tasks.

Tools: ASP.NET, C#, SQL Server, PHP, MYSQL

Supervisor:

Name: Muhammad Qamar Usman
Email ID: qamar@vu.edu.pk
Skype ID: qamarvu

Arts Gallery

Project Domain / Category

Web Application

Abstract / Introduction

The Arts Gallery project is a web-based application that allows artists to create profiles, upload their artwork, and exhibit their pieces in a virtual gallery. The system will also allow visitors to browse and purchase artwork, as well as provide features for gallery administrators to manage the website and its content.

Users: There will be four users of this website: Visitors, Customers, Admin, and Sellers (Artists).

1. **Visitors:** People who browse the website without creating an account. They can view artwork listings, read descriptions, and learn about the artists and their work.
2. **Customers:** Individuals who create an account on the website to purchase artwork. They can browse artwork listings, add items to their cart, and complete transactions.
3. **Administrators:** Gallery owners or managers who have access to backend functionalities for managing the website and its content.
4. **Sellers (Artists):** Individuals or organizations who list their artwork for sale on the platform. They can create listings, upload photos, and manage their inventory.

Functional Requirements:

Following are the functional requirements:

1. **Registration:** Customers & Sellers (Artists) will register first, so that they log in to the system.
2. **Login:** Admin, customers, and sellers (artists) will login to the system (after registration) using id and password.
3. **Manage users:** Admin will manage all customer & seller (artist) details.
4. **Manage Artwork:** Sellers (Artists) will add artwork that they wish to sell and provide all the details related to it like: Title, high-quality images, detailed description, price, etc. Also, seller (artist) should be able to edit or delete them. Also, when the artwork is sold, then seller (artist) should be able to change its status to SOLD or delete that item from the system.
5. **Purchase Artwork:** Customer will make purchases by adding items to their cart and process to checkout.
6. **Payment Method:** Customer can pay through online or by cash on delivery.
7. **View Artwork:** Visitors, Admin, Sellers (Artists), Customers should be able to view all the listed artwork.
8. **Approve/Disapprove Artwork:** Admin can approve or disapprove artwork added by Seller (Artist) on the basis of invalid information.

9. **Search Artwork:** Customer, Visitors & admin should be able to search for a particular artwork.
10. **Social Media:** Customer should be able to share artwork links on different social media platforms.
11. **Artist Profile Management:** Sellers (Artists) should be able to create and manage their own artist profiles, including uploading artwork and editing their bio and contact information.
12. **Order Management:** Admin should be able to manage orders, including processing payments and shipping artwork.
13. **Review and Rating:** Customers should be able to leave reviews and ratings for the artwork they've purchased.

Tools:

C#, Visual Studio, Sql Server, Tomcat or any webserver

Supervisor:

Name: Manahil Hassan
Email ID: manahil.hassan@vu.edu.pk
Skype ID: Manahil.hassan2

AutoEase: Online Car Showroom System

Project Domain / Category:
Web Application
Abstract / Introduction:

The web application is designed for an online Car Showroom, providing a platform for customers in Lahore, Islamabad, Peshawar and Karachi to purchase cars either through full payment or in instalments. The application offers a user-friendly interface with various car categories displayed on the homepage, complete with images, prices, and features. Customers can create profiles, submit their and their guarantors' bank details, and select their desired payment method. Upon completing the order, the system generates a confirmation message. Delivery to the customer's doorstep incurs additional charges. Customers can also cancel their bookings within 24 hours of placing the order for security and flexibility.

Functional Requirements:

1. **Admin Panel:**
    - Login and logout functionality for the showroom's admin users.
    - Ability to add, edit, and manage car inventory.
    - Viewing order history for customers.
    - Managing prices and availability of cars.
    - Generating reports related to car sales, customer orders, and payment methods.
    - Managing delivery charges and overseeing order cancellations.
    - Tracking and updating payment statuses, including instalment plans.
2. **Customer Functions:**
    - User registration process, including adding personal and guarantors' bank details.
    - Login and logout functionality for customers.
    - Browsing available cars categorized by car brands and models.
    - Selecting the desired city for delivery.
    - Placing an order with the option for full payment or an instalment plan.
    - Viewing order status and payment details.
    - Confirmation message displayed upon successful order placement.
    - Option to cancel the order within 24 hours of booking.
    - Viewing additional delivery charges for doorstep delivery.
3. **Payment and Delivery Management:**
    - Payment methods to include both full payment and instalment options.
    - Ability to calculate additional delivery charges based on customer location.
    - Processing and updating payment statuses.
    - Order tracking feature to view the current status of car delivery.
    - Automatic reminders for instalment payments.

Tools:

- Programming languages:
  - PHP
  - HTML, CSS, JavaScript
  - SQL (e.g., MySQL)
- Frameworks:
  - PHP Framework (e.g., Laravel, Symfony)
  - Front-end Framework (e.g., React, Angular, or Vue.js)
- Tools and Editors:
  - Code Editor (e.g., Visual Studio Code, Sublime Text, PHPStorm)
  - Database Management (e.g., phpMyAdmin, MySQL Workbench)
  - API Integration Tools (e.g., Postman, Guzzle for PHP)
  - Project Management Tools (e.g., Trello, Asana, Jira)
  - Web Server (e.g., Apache, Nginx)
  - User Interface Design Tools (e.g., Adobe XD, Sketch, Figma)

(Note: Student can use any other tool/editor as per his/her choice, with same programming language)

Supervisor:

Name: Komal Saleem
Email ID: komal.saleem@vu.edu.pk
Skype ID: komalsaleem123

Buy and Sell Pre-Owned Goods

Project Domain / Category
Web Application

Abstract / Introduction
The purpose of this project is to design and develop a user-friendly web application that allows users to buy and sell second-hand or pre-owned goods easily. The platform will cater to individuals looking to dispose of items they no longer need, while others can purchase these items at lower prices. The application will promote sustainability by encouraging the reuse of products and reducing waste.

Functional Requirements:
1. **User Registration and Profiles:**
    - Users (buyers and sellers) can register with basic information and create profiles.
    - Users can login to the web application.
    - Sellers can showcase a profile with items they've listed.
2. **Product Listing:**
    - Sellers can easily upload products with descriptions, images, and pricing.
    - Categories and tags for easy classification of items (electronics, furniture, clothing, etc.). Each student/group will have different categories.
3. **Search and Filters:**
    - Search function by keywords, category, price range, location, and condition of the item.
4. **Product Pages:**
    - Each listed product has a dedicated page with all details (description, price, seller info, images).
    - A button to initiate contact with the seller or directly buy the item (depending on the preferred sale method).
5. **Payment Gateway Integration:**
    - Secure payment options via credit/debit cards, PayPal, or mobile wallets.
    - Cash on delivery for local transactions can also be an option.
6. **Admin Dashboard:**
    - Admin has control over listings, user activity, and reported items.
    - Reporting system for inappropriate content or fraud prevention.

Tools:
- **Frontend:**
    - HTML5, CSS3, JavaScript (React or Angular)
    - Responsive design to ensure compatibility across devices (desktop, tablet, mobile)
- **Backend:**
    - Node.js with Express or Python with Django/Flask
    - RESTful API for communication between frontend and backend
- **Database:**
    - MySQL or any other DBMS for user and transaction data

Note:

1. **Students can select any technology for development of this project based on their prior knowledge or expertise.**
2. **University or supervisor are not liable to provide any paid resources for the development of this project.**
3. **Students with prior knowledge or willing to learn are encouraged to select this project.**
4. **Please do not opt for this project just because it is a web based project.**

Supervisor:
Name: Hina Rafique
Email ID: hina.rafique@vu.edu.pk
Skype ID: live:hina.rafique

Career Hub Website

Project Domain / Category

Web Programming

Abstract / Introduction

The **Career Hub Website** is a platform designed to connect job seekers and employers efficiently. The platform offers a centralized space where job seekers can search for relevant opportunities and employers can post job vacancies. The website is structured to cater to three types of users: **Admin**, **Employer**, and **Job Seeker**. Each user has specific functionalities tailored to their role. The platform aims to simplify the hiring process by offering advanced filtering options, easy job posting mechanisms, and an admin-controlled environment for managing content.

Functional Requirements:

Admin Role:

Admin oversees platform management. The admin has access to user accounts and job listings for monitoring.

Admin Panel:

The admin can:

Manage User Accounts: Add, edit, suspend, or delete accounts.

Moderate Job Listings: Approve, reject, or remove job postings that do not adhere to guidelines.

Monitor Platform Activity: Review application statistics and track employer and job seeker engagement.

Employer Role:

Employers post job vacancies, view applications, and manage the hiring process.

1. **Job Posting:**

   Employers can post job descriptions with detailed information, such as:

   - **Job Title**
   - **Company Name**
   - **Location**
   - **Job Requirements**
   - **Salary Range**
   - **Job Type** (Full-time, Part-time, Contract)
   - **Application Deadline**

Employers can edit or delete job postings.

Job Seekers Role:

Job seekers create profiles, upload resumes, search and apply for jobs.

1. User Profile:

Job Seekers can create a profile, including:

- **Personal Information** (Name, Contact Information, Location)
- **Resume Upload** (PDF, Word document formats etc.)
- **Education** and **Experience Details**

Job seekers can apply for jobs directly through the platform by attaching their resumes.

Application history is available for job seekers to track their applications.

2. Job Search:

Job seekers can filter job listings by:

- **Category/Industry** (e.g., IT, Finance, Marketing)
- **Location** (e.g., city, region)
- **Job Type** (Full-time, Part-time, Contract)

Search results are displayed with key details like job title, company name, and location.

A search bar enables job seekers to enter keywords for quick job discovery.

Tools:

HTML, CSS, JavaScript, jQuery, Bootstrap (Front-end)
MYSQL (phpMyAdmin) Database
PHP (Server-side programming)
XAMPP — Web Application Server

You are advised not to switch the tools. If you do so, you will handle the technical side yourself.

Note:

- These are the basic requirements of the application. Students may add further functionalities to make the application more useful.
- Virtual University of Pakistan (VU) will not provide any kind of hardware for this project; a student must arrange the required hardware by himself/herself.
- VU will not pay for any license of the software, the libraries /toolkits/APIs used in this project.

Supervisor:

Name: Ehsan ul Haq
Email ID: Ehsan.haq@vu.edu.pk
Skype ID: Ehsan_mtn

CarWash

Project Domain / Category
Web application

Abstract / Introduction
This project presents the development of a comprehensive car wash management system designed to streamline operations for administrators while enhancing the customer experience. The system includes features for admins to efficiently manage user accounts, services, bookings, and inventory, as well as track financial transactions and generate insightful reports. Admins can also run marketing campaigns to promote services and engage customers.

For customers, the system offers a user-friendly interface to create accounts, explore available car wash services, and book appointments easily. Multiple secure payment options are provided, along with notifications for upcoming appointments and promotional offers. Customers can also leave feedback to help improve services.

By integrating these functionalities, the car wash management system aims to optimize operational efficiency, improve customer satisfaction, and business growth in a competitive market.

Functional Requirements:

Admin Side Requirements

User Management:
- Create, edit, and delete user accounts (employees/admins).
- Assign roles and permissions.

Service Management:
- Add, edit, or remove car wash services and packages.
- Set pricing for each service.
- Manage service durations and availability.

Booking Management:
- View and manage customer bookings.
- Confirm, reschedule, or cancel appointments.
- Track booking history.

Inventory Management:
- Track supplies (cleaning products, tools, etc.).
- Set reorder levels and receive alerts for low inventory.

Financial Management:
- Generate invoices for customers.
- Track payments and outstanding balances.
- View financial reports and analytics (sales, revenue, etc.).

Customer Management:

- Access and manage customer profiles and history.
- Send notifications and updates (promotions, reminders).

Reporting:

- Generate reports on service usage, employee performance, and financials.
- Analyze trends for better decision-making.

Marketing Tools:

- Manage promotional campaigns and discounts.
- Send newsletters or offers to customers.

Customer Side Requirements

User Registration/Login:

- Create an account with email and password.
- Login/logout functionality.

Service Selection:

- View available car wash services and packages.
- Select desired services and view prices.

Booking System:

- Choose date and time for appointments.
- Confirm and receive booking details via email or SMS.

Payment Options:

- Provide various payment methods (credit/debit card, PayPal, etc.).
- Secure payment processing.

Profile Management:

- Edit personal information (address, contact details).
- View booking history and receipts.

Notifications:

- Receive reminders for upcoming appointments.
- Get updates on promotions or special offers.

Feedback and Reviews:

- Submit feedback on services.
- Rate the experience and provide comments.

Customer Support:

- Access a help section or FAQs.
- Contact support via chat, email, or phone.

Tools:
HTML/CSS, JavaScript, React.js, Node.js
MongoDB, SQL, MySQL
php, java, C#

Supervisor:
Name: Syed Aun Ali Bukhari
Email ID: aun.ali@vu.edu.pk
Skype ID: syed.aun89

"Connect & Converse": A Web-based Interactive Discussion Hub for University Students

Project Domain / Category

Web Programming

Abstract/Introduction

In today's digital age, effective communication and collaboration are essential for students' academic and social growth. An online dedicated discussion blog titled as "Connect and Converse" will serve as an effective platform for university students to share ideas, ask questions, and engage them in meaningful conversations / discussions on various topics. This project aims to develop a web-based interactive discussion hub (blog) for university students using PHP and any suitable framework to ensure a strong and user-friendly experience. The proposed project will develop comprehensive web development skills, enhance user experience design, and foster online community engagement for students working on it. It will also encourage student interaction, support knowledge sharing, and build a friendly online community among university students as its users.

Functional Requirements

User Authentication

- User registration and login features
- Email verification and password recovery options

User Profiles

- Customizable profiles with profile pictures
- Access to view other users' profiles and posts

Discussion Threads

- Ability for users to create new discussion topics
- Commenting on existing threads
- Upvoting and downvoting comments

Categories and Tags

- Organizing discussions by categories (e.g., academics, extracurricular, lifestyle, trending in daily life, major religious and national events etc.)
- Tagging system for streamlined searching and filtering

Search Functionality

- Effective search bar to find users, discussions and posts via keywords
- Advanced filtering options by category and date etc.

Notifications

- Real-time notifications on email for replies and comments

Admin Panel

- Admin roles for moderating discussions
- Interfaces for managing users, posts, and categories by the Admin

Comment Moderation

- Flagging system for inappropriate comments

- Admin capabilities to review and remove flagged content

Rich Text Editor

- Markdown support for formatting posts and comments
- Ability to attach images and links

Responsive Design

- User-friendly interfaces for accessibility across different devices with ease
- Intuitive navigation and layout

Tools / Languages:

- HTML
- Bootstrap or any other CSS Framework
- JavaScript (for enhanced interactivity). Any JavaScript library/ framework such as jQuery, Vue Js, or React Js.
- PHP as the backend language.
- Framework: You can choose any framework such as Laravel.
- MySQL as Database
- XAMPP
- Notepad++ / Sublime Text Editor

Supervisor:

Name: Muhammad Saeed Amjad
Email ID: muhammad.saeed@vu.edu.pk
Skype ID: saeed.lro

CPU Scheduling Algorithms Animated Simulator

Project Domain / Category

Web Programming

Abstract / Introduction

Creating a **Web-Based Simulator of CPU Scheduling Algorithms** project would allow users to interact with various CPU scheduling algorithms directly from their web browsers. The objective is to develop a web-based simulator that allows users to input processes, select a scheduling algorithm, and view the execution order and performance metrics (e.g., waiting time, turnaround time) in real time. The simulator will support popular CPU scheduling algorithms like FCFS, SJF, Priority, and Round Robin.

Functional Requirements

1. **User Interface**
   - **FR1.1**: User must be a registered user.
   - **FR1.2**: The web interface shall allow users to input process details, including:
     - Process ID (PID)
     - Arrival Time
     - Burst Time (execution time)
     - Priority (if applicable)
   - **FR1.3**: The web interface shall allow users to add, modify, and delete processes dynamically.
   - **FR1.4**: The system shall provide input validation to ensure values are positive, and times are correctly entered.
   - **FR1.5:** Users can save a process set and reload it later to continue the simulation.
2. **Algorithm Selection**
   - **FR2.1**: Users shall be able to select from a list of scheduling algorithms:
     - **First-Come, First-Served (FCFS)**
     - **Shortest Job First (SJF)**: Non-preemptive and preemptive (Shortest Remaining Time First - SRTF).
     - **Priority Scheduling**: Preemptive and non-preemptive versions.
     - **Round Robin (RR)** with a configurable time quantum.
     - **Multilevel Queue** and **Multilevel Feedback Queue** (optional advanced feature).
3. **Execution Simulation**
   - **FR3.1**: The system shall simulate the selected algorithm based on the process input and scheduling rules.
   - **FR3.2**: The system shall visually represent the CPU execution sequence using a **Gantt chart**.
   - **FR3.3**: The system shall display the current time in the simulation and show real-time updates as processes are executed.

4. **Performance Metrics**
    - **FR4.1**: The system shall calculate and display the following performance metrics for each process:
        - **Waiting Time (WT)**
        - **Turnaround Time (TAT)**
        - **Response Time (RT)**
    - **FR4.2**: The system shall calculate and display overall metrics, including:
        - **Average Waiting Time (AWT)**
        - **Average Turnaround Time (ATAT)**
        - **CPU Utilization**
        - **Throughput** (number of processes completed per unit time)
5. **Visualization and Results**
    - **FR5.1**: The system shall display a **Gantt chart** for the selected scheduling algorithm, showing the execution order and timeline.
    - **FR5.2**: The system shall provide a table summarizing the process execution details and performance metrics.
    - **FR5.3**: The system shall allow users to compare the performance of different scheduling algorithms by viewing side-by-side graphs (e.g., bar charts comparing waiting times, turnaround times).
    - **FR5.4**: The system shall allow users to export the simulation results (e.g., CSV or image export of the Gantt chart).
6. **Animation Controls:**
    - Users must be able to start, pause, and reset the simulation at any point during the algorithm execution.
    - The simulator should include a step-through mode that allows users to advance the simulation one step at a time to observe the process.

Detailed Functional Requirements

1. **Input Interface**
    - **Add Process Feature**: The user can dynamically add processes with attributes (ID, arrival time, burst time, priority).
    - **Delete/Edit Process Feature**: Users can delete or edit processes before starting the simulation.
2. **Algorithm Selection and Configuration**
    - **Preemptive and Non-Preemptive Options**: For Priority and Shortest Job First algorithms, users can select whether to allow preemption.
    - **Time Quantum Input**: When selecting Round Robin, the system shall prompt the user to input the desired time quantum.
3. **Gantt Chart Generation**
    - **Dynamic Simulation**: The processes should add dynamically in the queue. Simulation should be animated and interesting to watch.
    - **Gantt Chart Visualization**: The system shall generate a Gantt chart showing the CPU schedule with each process displayed in blocks according to the execution time.
    - **Dynamic Chart Updates**: The chart shall update in real-time as the simulation progresses.
4. **Real-Time Metrics Calculation**

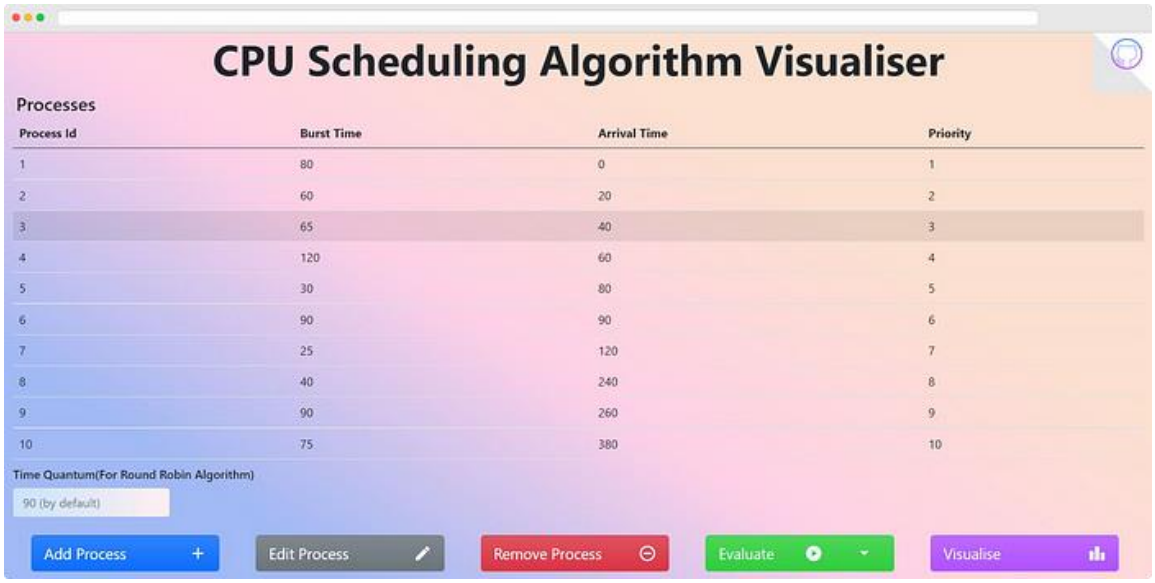
- **Average Time Calculation**: The system shall calculate and display average waiting, turnaround, and response times for the processes dynamically.
- **Performance Comparison**: Users shall be able to compare different scheduling algorithms by running simulations on the same set of processes and viewing the resulting charts side by side.

5. **Simulation Control**
   - **Start/Pause/Stop**: Users can start the simulation, pause it in the middle of execution, and stop or restart at any point.

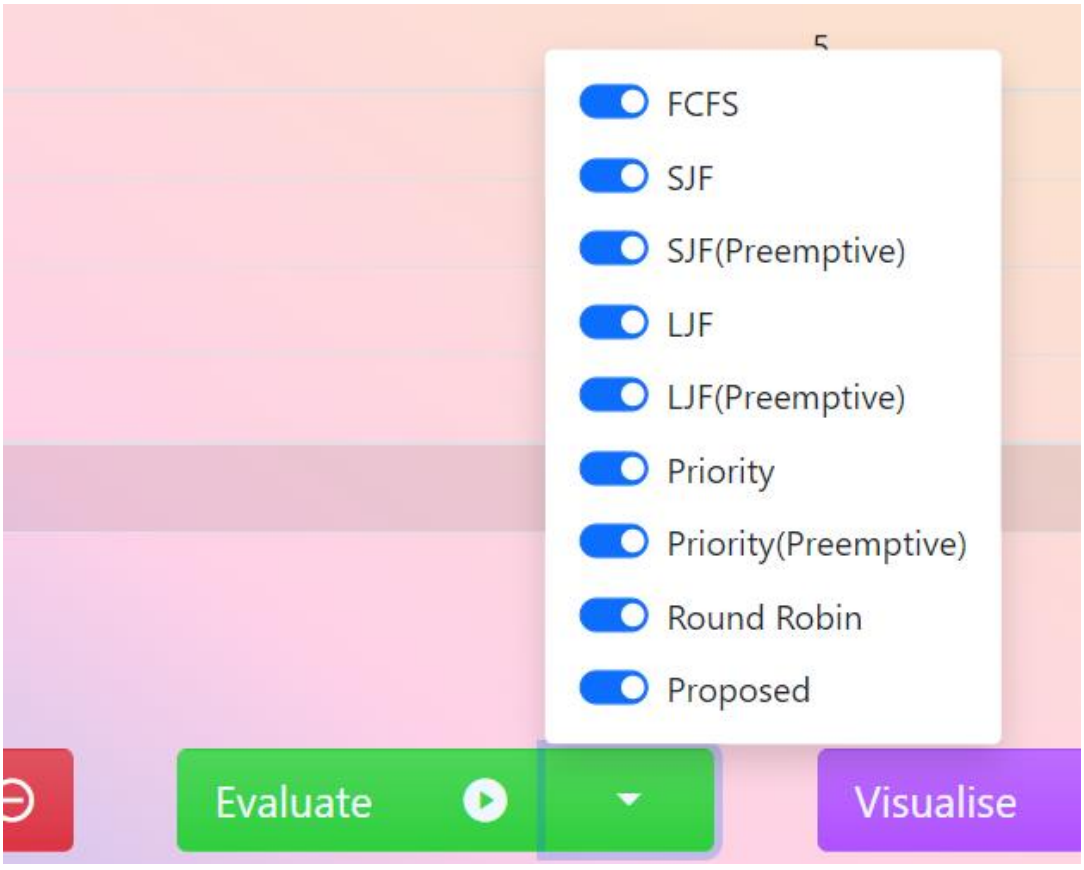
Samples:

The sample images are for reference and it does not mean that you have to develop an application just like in images. Show your own creativity in interface design. This is the basic output but show your creativity in interface and simulation execution.

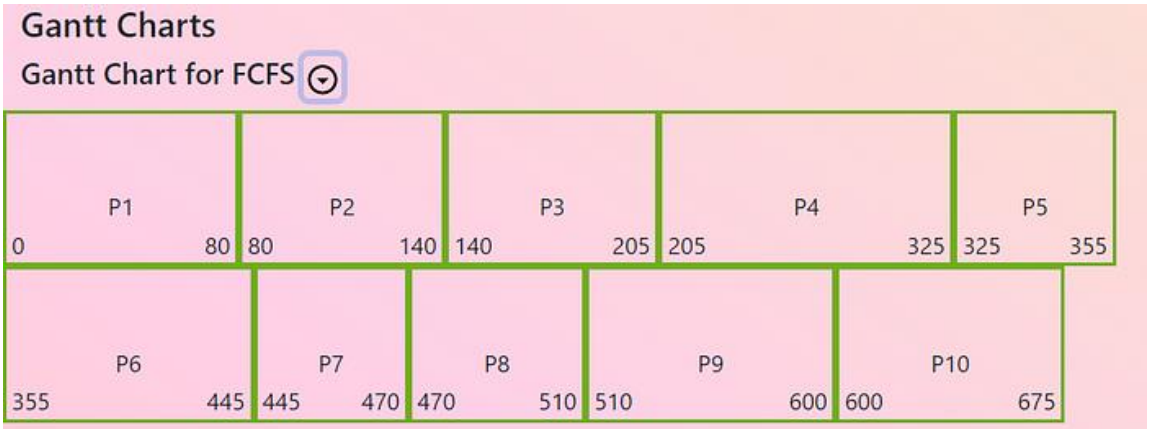
General Interface:



Evaluate:



Result of Evaluate:



Results:

Results

Scheduling Algorithm	Average Turnaround Time	Average Waiting Time
FCFS	251.50	184.00
SJF	191.00	123.50
SJF(Preemptive)	188.50	121.00
LJF	316.00	248.50
LJF(Preemptive)	541.50	474.00
Priority	251.50	184.00
Priority(Preemptive)	251.50	184.00
RoundRobin	264.00	196.50
Proposed	382.00	314.50

Tools

Frontend (Client-Side)

- **HTML/CSS/JavaScript**: For building the user interface.
- **Frameworks/Libraries**:
  - **React.js** or **Vue.js**: For building an interactive and dynamic web interface.
  - **Chart.js** or **D3.js**: For visualizing Gantt charts, graphs, and performance metrics.

Backend (Server-Side)

- **Node.js** or **Python (Flask/Django)**: For server-side logic and handling scheduling algorithms.
- **API**: REST API for handling process inputs, executing the scheduling algorithms, and returning results.

Database (Optional)

- **NoSQL (MongoDB)** or **SQL (PostgreSQL)**: To store simulation data and user configurations.

Supervisor:

- Name: Amna Bibi
- Email ID: amna.bibi@vu.edu.pk
- Skype ID: aamna.bibi26

Daycare Management System

Project Domain / Category

Database

Abstract / Introduction

Early education and care systems play a very significant role in the growth of children, preparing them for school and warranting parents the opportunity to engage in the workforce. It is believed that children need a warm, safe, colorful environment and diversified experience that focuses attention on 'play'. By making these things available, a child will grow and develop at a pace that is just right for them. In this project, you will create a website for **Daycare Management System,** this system will organize the operations of a daycare facility by providing a centralized platform for managing children's records, attendance, meal plans, activities, and staff schedules. The system will also allow parents to register their children, view daily updates, and communicate with the daycare staff efficiently, The goal of this system is to simplify administrative tasks, enhance communication between the daycare and parents, and ensure the safety and well-being of the children.

Functional Requirements:

1. Parents can create accounts, register their children, and manage their profiles.
2. Parents can view attendance, check in/out of their children.
3. Parents can receive notifications when their child is checked in or out.
4. Parents can view daily activity reports including photos or videos of their children.
5. Parents can view the weekly meal schedule and notify staff of any special food requests.
6. Parents and staff can communicate securely through a built-in messaging platform.
7. Parents can view, download, and pay invoices online.
8. Administrators can create, edit, or delete staff accounts and assign roles (staff, carer, manager).
9. Administrator and staff can enroll new children and assign them to specific groups (age-wise).
10. Administrator and staff can view and update child information.
11. Staff members and admin can check children in or out.
12. Staff members can log children's daily activities such as playtime, learning sessions, meals, and naps.
13. Daycare staff can create daily/weekly meals schedule.
14. Administrators can send out message and notifications about events, holidays, or emergency closures.
15. The system will generate monthly invoices based on the child's attendance.
16. Administrators can track all payment transactions and send reminders for pending bills.
17. If user likes their services, they can also share their link directly from the website to their social media apps.

Tools:

- HTML,CSS, JavaScript (Front-end)
- MYSQL (Backend)
- PHP (Server aspect programming)
- XAMP (Web Application Server)

Supervisor:

Name: Ammara Rasheed
Email ID: Ammara.rasheed@vu.edu.pk
Skype ID: ammaratouqeer409

E Loan Application & Verification System

Project Domain / Category
Web Programming

Abstracts/Introduction:
Getting a loan is a very tiring and complicated process in Pakistan. It may take weeks even months for loans to get approved and people have to visit the loan office again and again for document and verification. The proposed system will manage such activities through an online web application.

Functional Requirements:
In this system, involves detailing the system's core functionalities and features to ensure smooth operation and clarity for developers, stakeholders, and users.
Below are the **best functional and non functional requirements** that cover key aspects of an E loan application and verification system.

1. **User Registration & Authentication:**
    - The system shall allow users (borrowers and lenders) to register with personal details, such as name, email, phone number, and password.
    - The system shall provide multi-factor authentication for users to ensure secure access.
    - The system shall allow log in using their registered credentials (email users to /phone and password).
    - The system shall allow users to reset their password through a secure mechanism (email link or OTP).
2. **Loan Application Submission:**
    - The system shall allow registered users to apply for loans by submitting required details, including loan amount, loan type, purpose, and repayment period.
    - The system shall allow users to upload necessary documents, such as proof of income, identification, and credit history, during the loan application process.
    - The system shall allow users to save partially completed loan applications and continue later.
    - The system shall provide users with the ability to view and modify their loan application before final submission.
3. **Loan Eligibility & Verification:**
    - The system shall verify user-provided information, including employment details, income, and credit score.
    - The system shall perform automated eligibility checks based on predefined criteria such as credit score, income level, and debt-to-income ratio.
    - The system shall notify users if their application is ineligible, providing reasons for the denial.
    - The system shall allow users to correct or appeal any discrepancies during the verification process.
4. **Loan Approval Process:**
    - The system shall forward eligible loan applications to an internal team for manual review and approval.

- The system shall notify users via email/SMS about the status of their loan (approved, denied, or pending).
- The system shall allow internal users to approve or deny loan applications based on custom criteria and provide reasons for rejection.
- The system shall generate an approval letter or agreement document once a loan is approved.

5. **Loan Disbursement:**

   - The system shall allow users to provide bank details for loan disbursement.
   - The system shall facilitate automatic transfer of approved loan amounts to users' bank accounts.
   - The system shall provide a receipt or confirmation once the loan amount has been disbursed.

6. **Loan Repayment Management:**

   - The system shall generate a loan repayment schedule based on the loan amount, interest rate, and repayment period.
   - The system shall send reminders to users via email/SMS before the due date of each repayment installment.
   - The system shall allow users to make online payments for loan repayment using various methods (bank transfer, credit/debit cards, e-wallets).
   - The system shall allow users to view their outstanding loan balance and payment history.
   - The system shall notify users about late payments and calculate applicable late fees or penalties.

7. **Notifications & Alerts:**

   - The system shall notify users about any updates or changes to their loan application status via email/SMS.
   - The system shall notify users about upcoming loan repayment dates, approvals, or document submission deadlines.
   - The system shall notify users about any rejected or incomplete applications and suggest possible corrections.

8. **User Dashboard & Loan Tracking:**

   - The system shall provide a user dashboard where borrowers can view the status of their loan applications, repayment schedules, and transaction history.
   - The system shall allow users to track the status of their loan applications in real time (submitted, under review, approved, rejected).
   - The system shall provide summary reports to users showing their loan history, including approved, rejected, and ongoing loans.

9. **Customer Support Integration:**

   - The system shall provide an integrated customer support system (chat or ticketing system) for users to resolve any issues with their applications or loans.
   - The system shall allow users to raise support tickets and track the status of their queries.
   - The system shall provide users with access to FAQs, tutorials, and troubleshooting guides for common loan application issues.

10. **Admin Management:**

- The system shall provide an admin dashboard to manage user accounts, loan applications, and system settings.
- The system shall allow admins to configure loan products, set eligibility criteria, and manage interest rates.
- The system shall allow admins to approve or reject loan applications manually based on the company's loan policies.
- The system shall generate reports on loan disbursements, repayment statuses, and user statistics for analysis by the admin team.

11. **Security & Compliance:**

- The system shall encrypt all sensitive user data, including personal information, financial details, and loan documents.
- The system shall comply with relevant financial regulations, such as GDPR, CCPA, or any local regulatory requirements.
- The system shall log all user activities, including login attempts, application submissions, and document uploads, for audit purposes.
- The system shall automatically log out users after a period of inactivity to prevent unauthorized access.

12. **Reporting & Analytics:**

- The system shall generate detailed reports on loan applications, approval rates, user demographics, and financial statistics.
- The system shall provide real-time data analytics to admins on loan disbursements, repayment defaults, and loan cycle trends.

Tools:

ASP.NET/C#, HTML, CSS, JavaScript, Crystal report, SQL Server,

Supervisor:

Name: Qaiser Shabbir
Email ID: qaiser.shabbir@vu.edu.pk
Skype ID: qaiser_shabir

Event Management System

Project Domain / Category

Web Application

Abstract / Introduction

As event planning becomes increasingly complex, a streamlined approach to managing events can significantly enhance organization and efficiency. This project proposes a web-based Event Management System that allows users to create, manage, and RSVP to events. The application aims to facilitate seamless communication between event organizers and attendees, helping users keep track of upcoming events and manage their schedules effectively. With a focus on simplicity and user experience, this project targets individuals and small organizations looking to simplify their event planning process.

Functional Requirements

In the Event Management System project, there are typically two main types of users/actors:

1. **Event Organizers**
   - These users create and manage events. They can set event details, manage RSVPs, and view attendance counts.
   - Register and log in to the system.
   - Create new events with relevant details (name, date, location, etc.).
   - Edit or delete existing events.
   - View a list of RSVPs and manage attendee counts.
2. **Attendees**
   - These users can browse events, RSVP, and view event details.
   - Register and log in to the system.
   - View a list of upcoming events.
   - Search for events based on date or keywords.
   - RSVP to events and indicate attendance status (attend, maybe, decline).

Tools:

Frontend Development

HTML, CSS, and JavaScript for building the user interface.

Frameworks like Bootstrap for responsive design**.**

Backend Development

PHP: Use PHP for server-side scripting to handle requests and manage application logic.

Laravel: Consider using a PHP framework like Laravel for better structure and ease of development.

Database Management

MySQL: Use MySQL for storing user and event data, leveraging its relational database capabilities.

Supervisor:

Name: Rehan Ahmed
Email ID: rehan@vu.edu.pk
Skype ID: rehan.vu

Health Tracker: Monitor and Record Health Data

Project Domain / Category

Web based Application.

Abstract/Introduction

Health Tracker is a user-friendly desktop application designed to help individuals monitor, manage, and record various health-related data, including diet, exercise, and medical history. The primary goal is to provide users with a centralized platform to track their health progress, improve their wellness routines, and maintain an accessible digital medical history. This project is tailored for users who want to improve their health habits and need a simple, easy-to-navigate system to store and retrieve health data efficiently. We will develop a desktop application that allows users to log and monitor their daily health activities such as meals, workouts, and medical records Implement data visualization tools like charts and graphs to help users analyze trends in their health data over time. Provide a reminder system for users to input data regularly, track exercise goals, and medication schedules. Ensure data privacy and security by incorporating secure login and encryption methods for sensitive medical data. Make the system user-friendly with a simple and intuitive interface suitable for both tech-savvy and non-tech-savvy users.

The primary users of Health Tracker include Admin and Users:

Admin Functionalities:

Admin is responsible for managing the overall functionality and maintenance of the Health Tracker application. They have higher-level access and control over user accounts, data integrity, system configurations, and application settings.

Responsibilities and Privileges:

- **User Management:**
  - Create, edit, or delete user accounts.
  - Assign roles (Admin or User) and manage user privileges.
  - Monitor user activity and ensure compliance with data policies.
- **Data Management:**
  - Backup and restore health data securely.
  - Manage overall data integrity and handle errors or inconsistencies in the database.
  - Access and audit all user health data to ensure proper functioning.
- **System Settings and Configuration:**
  - Configure system settings, such as application preferences and notification settings.
  - Oversee security settings, including password policies, encryption settings, and data access controls.
  - Perform system updates and maintain the application’s performance and security.
- **Reports and Analytics:**
  - Generate reports on user activities, health trends, and overall app usage.

User Functionalities:

The User is the primary consumer of the Health Tracker application, using it to log and monitor their personal health data. They have access to features like tracking diet, exercise, and medical history but have limited control over system-wide settings.

Responsibilities and Privileges:

- **Health Data Entry and Tracking:**
  - Log daily health activities such as meals, exercise routines, medications, and medical history.
  - View and update personal health data at any time.
  - Set personal health goals, such as fitness targets or calorie limits.
- **Data Visualization and Analysis:**
  - Access visual data representation, such as graphs and charts, to track health trends over time.
  - Review historical data to assess progress in terms of diet, exercise, or medical conditions.
- **Reminders and Notifications:**
  - Set and receive personalized reminders for taking medication, logging meals, or completing workouts.
- **Medical Records Management:**
  - Maintain and store medical records, such as prescriptions, lab results, and doctor visits.
  - Download or export personal health data in various formats (e.g., CSV, PDF) for sharing with healthcare providers.
- **User Profile Management:**
  - Manage their own profile information, including password changes, email updates, and health preferences.
  - Customize notification settings for reminders and alerts.
- **Data Privacy:**
  - Maintain control over personal data visibility and accessibility to ensure privacy.
  - Secure access to the application through personal login credentials.

Tools and Technologies:

- **Programming Language**: Python/C#/Java for desktop application development.
- **Database**: SQLite or MySQL for local storage of user health data.
- **User Interface Design**: Tkinter (Python), WPF (C#), or JavaFX for creating a clean and intuitive user interface.
- **Graphical Tools**: Matplotlib (Python) or inbuilt chart libraries for data visualization.

Expected Outcomes

Upon completion, the Health Tracker desktop application will provide users with:

- A centralized and easy-to-use platform for tracking their health and fitness data.
- The ability to analyze and make informed decisions based on personalized health trends.
- A secure repository for managing and accessing medical history, which can be helpful during medical appointments.
- A reliable reminder system to support healthy lifestyle habits and medication compliance.

Supervisor:
Name: Asma Batool
Email ID: asmabatool@vu.edu.pk
Skype ID: asmabatool13

Online Maze Game for Learning

Project Domain / Category

Web Programming

Abstract / Introduction

Using video games in the learning environment has been shown to significantly boost student interest and engagement. Unlike traditional teaching methods that may struggle to capture the attention of today's tech-savvy learners, video games present material in a dynamic and interactive format that resonates with students. The element of challenges and rewards transforms the learning activity into exciting adventures that encourage active participation. This heightened engagement not only makes learning more enjoyable but also fosters a sense of agency and motivation among students, allowing them to explore concepts at their own pace. OMGL (Online Maze Game for Learning) is an interactive online maze game that has some boxes randomly filled in the game window. Each box will have a number which is used as a reward after completing the challenge hidden in the box. This reward is used to unlock doors and finally reach the target destination in the minimum time.

Functional Requirements:

Following are some abstract-level functional requirements of the game. The students who selected this project will provide detailed requirements in SRS documents.

- This online game will run through any browser.
- In the game interface, a maze will have random boxes/gates that will block the path to reach the target place.
- Each box can be crossed with the help of keys that can be earned through programming challenges.
- After completing a programming challenge, some keys will be awarded and added to the total earned keys.
- If the player has enough keys to open a gate, the player will move in front of the gate and use a specific keyboard button to open it.
- Upon opening a gate, the required number of keys will be deducted from the total earned keys.
- When the game starts, a timer will start, which will be used to track how much time is taken by a player to complete the game. This timer will not stop when completing a programming challenge.
- The time will increase or decrease according to the requirements of the programming challenges given below.
- All programming challenges will be stored in a database according to defined categories and correct answers.
- If a player has completed a challenge from any category, it will not be used again unless the player closes the game.

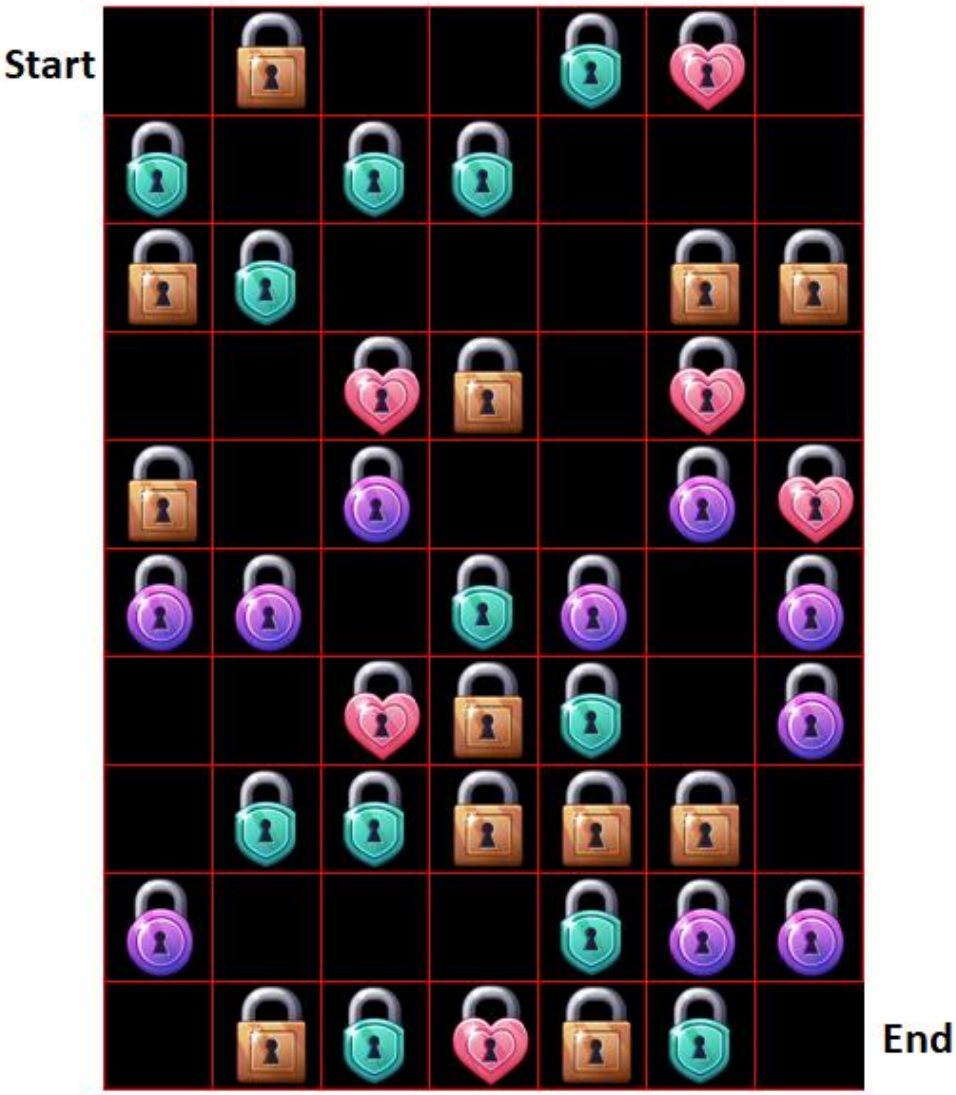
Following are categories of challenges

1. **Quiz Challenges**: Give a multiple choice question with four options. The player will select an option and submit it. Completing this challenge will give 2 keys to the player.
   Sample Question: “What does a loop do in programming?” (Options: a) Repeats code, b) Stops execution, c) Changes data).
2. **Coding Challenge**: Give a challenging task to the player to complete it by giving input through a text box. Completing this challenge will give 3 keys to the player.
   Sample Question: “What is the name of the header file that is required to use “cin” and “cout” statements? Get the answer through the text box.
3. **Interactive Scenario**: Present the player with a scenario where the player has to make decisions based on programming principles. Get the player input through a text box. Completing this challenge will give 4 keys to the player.
   Sample Question: “You need to check if a user input is valid. Do you use an “if” statement or a “loop”?
4. **Debugging Challenge**: Provide the player with a snippet of code containing an intentional error or bug for the identification. A text box and a drop-down list will be used to take input from the player. The text box will be used to take the line number which has an error or bug. The drop-down will be used to take input of syntax, logical, runtime error. Completing this challenge will give 5 keys to the player.
   Sample Question: “Find the error in this function that’s supposed to calculate the average of two numbers”.
5. **Sharp Shooter**: Load random challenge from the above 4 categories which the player needs to complete quickly within the given time. A question will be loaded, a timer of 20 seconds will start, and when the timer reaches zero and the player has not submitted the answer, the question will disappear (close). Completing this challenge will give 10 keys to the player.
   Sample Question: “What will be the output of the given code snippet?”

   Note: Submitting an incorrect answer will add five seconds to global time and finishing every challenging task successfully will decrease the time by five seconds. Only C++ programming language should be used for challenging tasks.

Following screenshot shows the start and end location to finish the game.
To remove a lock, a certain number of keys are required.

Lock Type				
Keys Required	**10**	**5**	**4**	**2**



Tools:
Python, PHP, ASP.Net

Supervisor:
Name: Muhammad Ahmad Lodhi
Email ID: ahmadlodhi@vu.edu.pk
Skype ID: ahmad_lodhi

One Click Desert Providers

Project Domain / Category

Web Application

Abstract / Introduction

The aim of this project is to develop an interactive website that facilitates the customers with the one-click functionality to order deserts for any occasion with different categories like Arabian Deserts, Pakistani Deserts, Indian Deserts, Italian Deserts, Turkish Deserts etc. make your own desert basket etc. and customers can buy them at any time. The aim of this application is to reduce the efforts required to the customer to physically visit the Deserts shop to purchase the deserts/ deserts gifts to reduce the manual efforts required for the Store Administration to manage the transactions. The application will also help the administrator in maintaining and updating desert purchases record.

Existing System

In conventional shops/stores, the customers have to physically visit the store and all the orders are taken manually. The stock is maintained on paper that takes more time and it also demands more human resources.

Proposed System

- An online desert shop that allows users to check for various desert categories available at the online shop and purchase online.
- The online shop contains the list of available deserts displayed according to different categories along with their prices.
- The user may browse through these deserts categories as per his/her requirement.
- If the user wants to buy deserts he/she may add it to his/her shopping cart.
- Once user wishes to checkout, he/she must register on the website first. User can then login using same ID and password next time.
- Payment mode may be “Cash on delivery, “Debit card” and “Credit Card”.
- Once the user makes a successful transaction he/she gets a copy of the shopping receipt on his/her E-mail id.
- The system will also reduce the manual operation associated with the maintenance of the records consisting of the product, product order details, and customer details.

Number of Modules:

1. **Login/Registration Module**: In this module, the administrator or the customer will have to authorize his/her access to the modules of this online system.

2. **Deserts Categories Module:** In this module each and every operation related to the Deserts category and Deserts item quantity such as adding new Deserts item category/quantity, editing the existing Deserts item category/quantity, deleting Deserts item category/quantity, getting the lists of Deserts item category/quantity and reports of Deserts item category/quantity will be managed by the Administrator. Deserts items will be arranged and can be viewed in categories. i.e.

Pakistani Deserts/Arabian Deserts etc.

- Pakistani Deserts gifts for Karachi, Pakistani Deserts gifts for Lahore, Pakistani Deserts gifts for Rawalpindi and Pakistani Deserts gifts for all major cities etc.

3. **Personalized Deserts Gifts Basket**
    - Make your own deserts basket from different available deserts items.

    Visit this link for Deserts categories options
    https://jalalsons.com.pk/
4. **Customer Module**: This module helps the customer, to create Account, Sign-in, Search deserts category, Select deserts category/quantity, Buy deserts, Continue Shopping, View/Edit Cart, Checkout, Obtain Bill Information, Confirm order, and Delete Order etc.
5. **Admin Module**: In this module, interfaces for adding deserts items, adding category/quantity, modifying deserts items category/quantity, deleting deserts items category/quantity, Sale Record, Database's data Report, transactions history are included.

Functional Requirements:

1. A Login and registration facility for enabling access to system both for customers and Admin.
2. The customer can view/search the deserts categories without logging into the system or getting him/her registered.
3. To buy deserts, user should follow the registration and login process. Payment will be made through debit/credit card or cash on delivery.
4. Administrator will maintain the system through admin panel which consists of order bills, order Status add/update/delete the product details in the catalogue.

Tools:

Software Requirements:

- Operating System: Window 7 and above
- HTML, AJAX, JavaScript, CSS (Front-end)
- MYSQL (Backend)
- Python(Flask)/ PHP (Server side programming)
- XAMPP-------Web Application Server

Note: If you are going to choose Python, it will be much appreciated.

Supervisor:

Name: Shafaq Nisar
Email ID: shafaq.nisar@vu.edu.pk
Skype ID: shafaqnisar1

Online Bank Loan Portal

Project Domain / Category

Web programming

Abstract / Introduction

The banking sector plays a vital role in the economic development of a country by providing financial services to individuals and businesses. One of the core services offered by banks is the provision of loans. Managing loans manually is often cumbersome and error-prone, leading to delays and customer dissatisfaction. The proposed **Online Bank Loan Portal** aims to simplify and streamline the loan management process through a web-based platform.

The system will allow banks to handle loan applications, track loan disbursements, manage repayments, and ensure regulatory compliance efficiently. Additionally, customers will be able to apply for loans, track their application status, and make repayments through an online portal, making the loan management process transparent and efficient.

The system will cater to different types of users, including administrators (bank staff), loan officers, and customers, providing them with the necessary tools to handle all aspects of loan management, from application to closure.

The users of this application are given below.

1. **Bank Administrators:** Responsible for managing loan products, interest rates, and overseeing the loan approval process.
2. **Loan Officers:** Review loan applications, assess eligibility, and approve or reject applications based on bank policies.
3. **Customers:** Individuals or businesses applying for loans, tracking their application status, and managing loan repayments.

Functional Requirements:

The Online Management System will cater to the following functional requirements:

1. **Bank Administrators**
   - Manage loan types, interest rates, and repayment terms.
   - Oversee loan applications and assign loan officers to process them.
   - Monitor loan disbursements and track repayments.
2. **Loan Officers**
   - Review loan applications and customer profiles.
   - Approve or reject loan applications and provide feedback to customers.
   - Track customer repayments and update loan statuses (active, closed, overdue).
   - Communicate with customers regarding their loan queries.
3. **Customers**
   - Register and manage their profiles on the platform.
   - Apply for different loan products (e.g., personal loans, home loans, business loans).
   - Upload payment receipt after loan installment repayment.

Note:

- These are basic requirements of the application. Students may add further functionalities to make the application more useful.

- Virtual University of Pakistan (VU) will not provide any kind of hardware for this project; student has to arrange the required hardware by himself/herself.
- VU will not pay for any license of the software, the libraries /toolkits/APIs used in this project.

Tools:

Microsoft Visual Studio, SQL Server, Asp.net.

Note: You are advised not to switch the tools. If you do so, you will handle the technical side yourself.

Supervisor:

Name: Ghulam Abbas
Email ID: ghulam.abbas@vu.edu.pk
Skype ID: live:49cf19b1f61ba126

Online Handmade Crafts Store

Project Domain / Category

Web application

Abstract / Introduction

The Handmade Crafts Store is an online marketplace designed to connect artisans with customers who value unique, handmade products. In addition to browsing a variety of handmade goods, customers will have the option to request custom-made designs, enabling artisans to showcase their creativity while fulfilling personalized orders. This platform will facilitate a smooth exchange between buyers and sellers, ensuring a vibrant ecosystem for artisans to thrive and customers to enjoy exclusive products. Furthermore, the purpose of this website is to provide artisans with a digital storefront to display and sell their handmade crafts. The platform will enable customers to discover unique products and connect directly with artisans for custom orders. The website will act as a bridge, empowering local artisans to reach a broader market, expanding their business opportunities, and offering customers a convenient way to access unique, handcrafted items.

Functional Requirements:

1. User (Customer) Features:

- **Registration and Login:**
  - Users can create an account using email and password.
  - Social media login options (e.g., Google, Facebook).
- **Browse Products:**
  - Categories for different types of handmade crafts (e.g., jewelry, clothing, home decor).
  - Search functionality to find products by name, artisan, or category.
  - Filtering options (price range, rating, availability).
- **Product Pages:**
  - View detailed product information, including images, descriptions, price, and available stock.
  - User reviews and ratings for each product.
- **Custom Design Requests:**
  - Option to request a custom-made product by filling out a form with design specifications.
  - Customers can communicate directly with artisans regarding customizations.
- **Shopping Cart and Checkout:**
  - Add products to the cart and modify the quantity.
  - Secure checkout process with multiple payment methods (credit card, PayPal, etc.).
  - Order summary and confirmation.
- **Order Tracking:**
  - Customers can view order status (processing, shipped, delivered).
  - Notifications for order updates.
- **Wishlist:**
  - Option to save favorite products for future purchases.

- **User Profile:**
  - Manage personal details (name, email, address).
  - View past orders and order history.
  - Manage saved payment methods.

2. Artisan (Seller) Features:

- **Registration and Login:**
  - Artisans can create a seller account using email.
- **Product Management:**
  - Upload product images, descriptions, and pricing.
  - Set inventory levels and update availability.
  - Option to accept or decline custom design requests.
- **Order Management:**
  - View and manage incoming orders.
  - Update order status (processing, shipped, etc.).
- **Custom Design Management:**
  - Review customer design requests and communicate with customers.
  - Provide price estimates and delivery timelines for custom orders.
- **Sales Reports:**
  - View sales analytics, including total sales, top-selling products, and customer feedback.
- **Profile Management:**
  - Manage artisan bio, product details, and contact information.

3. Admin Features:

- **Dashboard:**
  - Overview of website activity (total users, total sales, active artisans, etc.).
- **User Management:**
  - Manage customer and artisan accounts (create, update, or delete users).
  - Suspend or ban users for violating terms.
- **Product Management:**
  - Monitor all products listed by artisans.
  - Approve or reject new product listings.
- **Order Management:**
  - Oversee orders and transactions between customers and artisans.
- **Site Management:**
  - Update website content, such as homepage banners, categories, and special promotions.
  - Set up and manage shipping rates, taxes, and payment gateways.
- **Reports and Analytics:**
  - View comprehensive reports on website activity, including sales data, most popular products, and customer behavior.

Tools:

HTML, CSS, Bootstrap, JavaScript, MYSQL, PHP

Supervisor:

Name: Sohail Aamir
Email ID: soahilaamer@vu.edu.pk
Skype ID: sohailaamir22

Online Tutor Finding Application

Project Domain / Category

Web Programming

Abstract / Introduction

In this project, we will build a web based application named as "Online Tutor Finding Application", in which the user/parents will search educational tutors for their children's and select the most appropriate tutor. The admin will register all the tutors to the system and will provide services to the users/parents. The tutors will share all the classes' information and quizzes online to their registered students/users.

Admin panel features

1. The Admin can login/logout.
2. The Admin can add/update/delete different fields/categories like, Computer Science, Mathematics, Medical, Engineering etc.
3. The Admin can register different tutors in different fields/categories. At the time of registration, the admin can provide all the relevant details like, tutor name, qualification, picture, experience, picture, address, mobile number, tutor fee etc.
4. After the registration, the admin can share account details with the tutors through email or through message.
5. The Admin can update/delete a registered tutor.
6. The Admin can view the list of all the students registered against a particular tutor.
7. The Admin can generate fee vouchers for the students.
8. The Admin can view and verify the uploaded fee voucher.
9. The Admin can manage the expenses and profit reports.

User/Parent panel features:

1. The user can register.
2. The user can login/logout.
3. After the login, the user can update his/her profile.
4. The user can view all the tutors' information like his/her name, qualification, experience, picture, resume etc.
5. The user can search different tutors on the basis of different fields like computer science, medical, engineering, etc.
6. The user can select one or more than one tutors for a single or more subjects.
7. After the tutor/s selection, a fee voucher can be generated for the user and the user can download the fee voucher and pay the dues.
8. After paying the dues, the paid voucher can be uploaded by the user.
9. The user can view its status.
10. The user/student can view the class timing, class link and any other details provided by the particular tutor.
11. The user/student can take the Quiz online.
12. The user/student can view his/her progress in the Online Quiz/s.

Tutor panel features:

1. The tutor can login after the registration. The registration process should perform by the Admin, for which the tutor should physically visit the Admin office.
2. After the login, the tutor can upload his/her resume, which should be visible at the user/parent panel.
3. The tutor can view all the information of those students who have selected the particular/logged-in tutor.
4. The tutor can share the class timing, class link etc. with his/her students.
5. The tutor can create class link in any online tools, like Zoom, Google meet etc.
6. The tutor can create Quiz/Quizzes based on MCQs for the students.
7. The tutor can monitor the student progress.
8. The tutor can logout.

Tools:

XAMPP Server, MySQL, PHP language

Supervisor:

Name: Akmal Khan
Email ID: akmalkhan@vu.edu.pk
Skype ID: akmal_vu

PRMS: Patient Record Management System

Project Domain / Category
Web Application

Abstract/Introduction
The purpose of this software is to keep medical records of patients. Medical records may include personal information, medical history, medication, laboratory test result, allergies, radiology images and vital signs.

This software maintains data accuracy and captures the patient information time to time. Proper use of this software will eliminate the need of paper based medical record. Therefore, the risk of loss of paper record will be finished. A single centralized data storage facility will eliminate data redundancy. By using the search facility of the software, the user would be able to extract medical data for the examination of possible trends and long term changes in a patient. Population-based studies of medical records may also be facilitated by the widespread adoption of this software.

Functional Requirements:
Some of these functions include, but are not limited to:

- Maintain patient record
- Manage patient demographics
- Manage patient-specific problem lists
- Manage medication lists
- Manage medical procedural/surgical, family history including the capture of pertinent positive and negative histories, patient-reported or externally available patient clinical history.
- Create, addend, correct, authenticate and close, as needed, transcribed or directly entered clinical documentation and notes.
- Incorporate clinical documentation from external sources.
- Present organizational guidelines for patient care as appropriate to support order entry and clinical documentation.
- Provide administrative tools for organizations to build care plans, guidelines and protocols for use during patient care planning and care.
- Generate and record patient-specific instructions related to pre- and post-procedural and post-discharge requirements.

Tools/Language:
You can use any language which supports the development of web applications.

Supervisor:
Name: Dr. Syed Shah Muhammad
Email ID: syed@vu.edu.pk
Skype ID: mscsvu

Consultant Appointment Booking System (CABS)

Project Domain / Category

Web based Application

Abstract/Introduction

Medical care is the basic necessity of every human being while it is also a fact that new diseases are emerging every day. It rises the need to make the reach to doctors (consultants) as easy as possible. In this regard, Consultant Appointment Booking System (CABS) application enable any hospital or medical institution having consultants of different departments to facilitate its patients in taking consultant's appointment at their doorstep.

This application stores the information of consultants of different departments and their available duty timings. It also stores the information (data) of patient once they are treated and make enable new users to book their appointment from home or somewhere else without physically visiting to the hospital.

Functional Requirements:

A set of functional requirements of the proposed system may include the following.

1. The users must be able to register themselves.
2. The users are able to edit/update their profiles.
3. There must be a login process for users/Admin to access the application.
4. The users must be able to book an appointment for the consultant of their choice.
5. The application enables the users to change/update their appointment (if needed or they want).
6. The application enables the hospital staff (or admin) to add consultants, remove consultants, update their information etc.
7. The application should provide a list of patients treated by a specific consultant.
8. The system should also generate some reports like, list of available consultants' department wise (specialty wise), list of patients treated by a particular doctor, list of patients with respect to their geographical area (city etc.) etc.

Note: Make sure that the above list is a sample of functional requirements, students do not need to be limited with these only. Students are supposed to think (or visit the problem domain) and implement some other functional requirements not mentioned in above list.

Tools:

MySQL, PHP etc.

Supervisor:

Name: Asif Hussain
Email ID: asifhussain@vu.edu.pk
Skype ID: asifnoor1982

Derma Elixir Studio

Project Domain / Category

Web Programming

Abstract / Introduction

The **Derma Elixir Studio** is running a welfare program to help needy people across Pakistan. The derma elixir studio aims to create a website for its newly opened skin care clinic located in Islamabad. The goal of the skin care clinic is to provide free treatment to visitors to determine eligibility for the free services.

You will be developing a website for the skin care clinic. The skin care clinic website will provide visitors with information about the different skin specialists and skin treatments available for recommendations. Patients must physically visit the skin care clinic in Islamabad for a check-up. Before visiting the skin care clinic, users must make an appointment by registering through the online portal on the website.

The patient's complete record will be maintained so that the Skin Specialist (Doctor) can check or revise the patient's history during their next visit to the skin care clinic.

Functional Requirements:

- Only the registered user can access the website portal i.e., Patient or Skin Specialists' Dashboard.
- User will upload a clearance certificate, issued by the skin revive studio, at the time of registration
- The user will log in to the portal and then send a request to schedule an appointment with a Skin Specialist.
- The admin will be able to verify certificate number using the system.
- The admin will set the website homepage for visitors who are not registered users but want to access information about skin specialists and different skin treatments available online or at the skin care clinic.
    - Other details that a visitor can obtain from the website include:
        - About Us, Our Services, Tests, Skin Treatments, Contact Us.

- The Skin Specialist will be able to view the patient's history during their visit to the skin care clinic.
- The report generation option will also be available if the Skin Specialist wants to obtain information about the patient's record.
- The Skin Specialist can input the patient's information, along with related data, like prescribed medication, treatment and details of lab tests etc.
- The feedback interface will be available to users who have completed their appointments for sharing their experiences. The system will provide an interface to give suggestions and reviews.

- Users will be able to view reviews from individuals who have visited the skin care clinic.
- Only verified users can give reviews about staff members (Skin Specialist, helping staff) at the skin care clinic. Review details will be displayed on the website homepage.
- The search option will also be available if a user wants to search for a Skin Specialist, Any skin concern and any skin treatment through the system

Tools:
HTML, CSS, JavaScript, PHP, MySQL, Visual Studio Code/ Sublime Text.

Supervisor:
Name: Komal Khawer
Email ID: komal.khawer@vu.edu.pk
Skype ID: kom.kk

Exam Seating Arrangement System

Project Domain / Category
Web application

1. **Abstract / Introduction**

1.2. Introduction:
This project aims to develop a web application that efficiently manages exam seating arrangements for educational institutions. The system will streamline the process of assigning seats to students, considering various constraints such as student preferences, classroom capacity, and special needs (in the sense that if there are more students the accommodate them in class).

1.3. Objectives:
- To create a user-friendly interface for administrators to input data and generate seating arrangements.
- To ensure fairness and compliance with institutional policies.
- To provide students with easy access to their seating information.

---

2. Scope of the Project

2.1. In Scope:
- User authentication for administrators and students.
- Data input for courses, students, and exam schedules.
- Automated seating arrangement generation based on predefined rules.
- Notification system to inform students of their seating assignments.

2.2. Out of Scope:
- Physical management of seating arrangements (e.g., furniture layout).
- Support for non-exam-related seating arrangements.

---

3. Functional Requirements

3.1. User Authentication:
- Admins can register and log in securely.
- Students can log in to view their seating assignments.

3.2. Data Management:
- Admins can input and manage courses, student lists, and exam schedules.
- The system supports bulk uploads of student data via CSV files.

3.3. Seating Arrangement Generation:
- The system will generate seating arrangements based on constraints such as:
  - Classroom capacity.
  - If there are 100 students in a classroom arrange them in rows (like 3, 4,5 equally) in front of the teacher's table
  - Distance between students' lines must be equal so that one can't see the other's paper.
  - To avoid conflict don't seat any student on the same exam, to be seated nearby or in parallel row.
  - Student preferences (if any).

3.4. Notifications:
- Students receive email/SMS notifications with their seating assignments.
- Admins can send reminders about upcoming exams.

3.5. Reporting:
- Generate reports on seating arrangements for each exam.

- Admins can download seating plans in PDF format.

---

4. Non-Functional Requirements

4.1. Performance:

- The system should generate seating arrangements within 2 minutes for up to 500 students.
- Page load times should not exceed 3 seconds under normal conditions.

4.2. Security:

- Implement SSL encryption for secure data transmission.
- Ensure secure password storage (e.g., hashing).

4.3. Usability:

- The application should have an intuitive interface that requires minimal training.
- User feedback will be collected to improve usability.

4.4. Compatibility:

- The application should function across major web browsers (Chrome, Firefox, Safari, Edge).
- It should be responsive and usable on mobile devices and tablets.

4.5. Scalability:

- The architecture should support future growth, accommodating additional features such as integration with existing student information systems.

5. Tools:

PHP, Java, C#, HTML, CSS, SQL (for the backend), or any other tool of your own choice.

Supervisor:
Name: Dr. Nida Anwar
Email ID: nidaanwar@vu.edu.pk
Skype ID: nida.vu

Frozen Food Panda

Project Domain / Category

Web Application

Abstract / Introduction

The project aims to develop a user-friendly web application specializing in frozen food items and homemade dishes. Utilizing ASP.NET MVC and SQLite Server, the website will provide an efficient shopping experience, allowing customers to easily browse a diverse range of frozen foods and home-cooked meals. The platform will also support features such as user reviews, order tracking, and a secure checkout process. The goal is to create a seamless online environment where customers can discover high-quality food products, place orders, and have them delivered conveniently to their homes. The application will prioritize both user experience and administrative efficiency, ensuring that vendors can easily manage their inventories and orders.

Functional Requirements:

- FR1: User Registration and Login
    i. Users must be able to create an account by providing their email, password, and contact information.
    ii. Passwords should be stored securely and include validation checks (e.g., minimum length, special characters).
    iii. Users must be able to log in using their registered email and password.
    iv. Implement a “Forgot Password” feature that allows users to reset their passwords via email verification.
- FR2: Product Catalog
    i. Display a comprehensive list of available frozen food items and homemade dishes with images, descriptions, and prices.
    ii. Implement search functionality to allow users to find products by name or category (e.g., frozen meals, cooked dishes).
    iii. Allow users to filter products based on criteria such as price range, latest item.
    iv. Each product page should include detailed information and customer reviews.
- FR3 Shopping Cart Management
    i. Users must be able to add items to their shopping cart from the product catalog.
    ii. The shopping cart should display the selected items, quantities, prices, and a total cost.
    iii. Users should have the ability to update quantities or remove items from the cart.
- FR4 Secure Payment Processing
    i. Provide multiple payment options (debit card, cash on delivery, easy paisa, jazz cash).
    ii. Allow users to review their order details before confirming payment.
- FR5 User Account Management
    i. Users should be able to view their order history, including order status and tracking information.
    ii. Provide functionality for users to update their personal information.
    iii. Allow users to manage their payment methods and set default payment options.

- FR6 Admin Panel for Product Management
    i. Admin users should be able to log in to a secure admin panel.

ii. Allow admins to add new products, including uploading images and setting prices and descriptions.
iii. Implement functionality for editing existing products and removing discontinued items.
iv. Provide analytics for sales, inventory levels, and customer engagement metrics.

- FR7 Order Management System
  i. Admins must be able to view all customer orders, including order details and payment status.
  ii. Allow admins to update order status (e.g., processing, shipped, delivered) and notify customers via email.
  iii. Provide functionality for handling refunds or exchanges.
- FR8 Customer Comment and Rating System
  i. Enable customers to leave comments and ratings for products they have purchased.
  ii. Implement a moderation system to approve Comment before they are publicly visible.
- FR9 Responsive Web Design
  i. The website must be mobile-friendly, ensuring optimal viewing and interaction across various devices and screen sizes.
  ii. Implement a consistent and intuitive navigation system that works well on both desktop and mobile interfaces.

Tools:

Development Environment: Visual Studio 2022
Programming Language: C# (for ASP.NET MVC)
Database: SQLite or SQL Server
Front-end Technologies: HTML, CSS, JavaScript, Bootstrap

Supervisor:

Name: Neelam Alam
Email ID: neelam.alam@vu.edu.pk
Skype ID: neelam-cs

Identity Management User Interface (IDMUI)

Project Domain / Category

Software Engineering/Infrastructure Management

Abstract/Summary

In OpenStack, Keystone is the identity service that provides authentication, authorization, and service discovery mechanisms for other OpenStack services. It plays a central role in managing users and their permissions, ensuring access to resources across the OpenStack environment is secure and controlled. Holistically, the key stack provides the following capabilities.
Keystone handles user and service authentication by verifying credentials (username/password, tokens, etc.) and issuing authentication tokens that can be used to access other OpenStack services. After authentication, Keystone manages authorization by defining roles and permissions. It determines what resources users and services are allowed to access based on predefined roles and access control policies. Keystone manages projects (also known as tenants) to isolate resources. Users belong to projects and can have different access levels within those projects. Keystone maintains a service catalog, which lists all available OpenStack services and their endpoints. This allows clients to discover which services are available in the environment. Keystone issues and manages authentication tokens that are used by other OpenStack services to verify identity and authorization without needing to re-authenticate the user on every request. It provides the ability to manage multiple domains, each with its own set of users, groups, and projects, supporting hierarchical multi-tenancy. Keystone handles the creation, management, and association of users, groups, and their roles. In this project, you will develop a web-based User interface for managing the identity service.

Functional Requirements:

you will first prepare a virtual machine using Oracle VirtualBox or VMWARE and install the latest Ubuntu Server Operating System and Keystone package and OpenStack API. You will develop a web application called IDMUI using any framework/language of your choice. The application must be able to perform the following tasks

1. View the status of the Keystone Service running in the VM.
2. Allow start/stop and change the configuration of the keystone service in a user-friendly way.
3. Should provide a top-level menu named: Identity Service and provide the following options and submenu.
    - Service Management
    - Endpoint Management
    - User and Role Management
    - Project (Tenant) Management
    - Token Management
    - Domain Management
    - Group Management
    - Configuration management
4. Create another top-level menu named Database Management
5. Each menu should provide suitable options for configuration/view status/statics which are provided by the keystone service.
6. The IDMUI will communicate with the Keystone server via open stack APIs.

7. You must use the MySQL database as the common database for keystone and application data management requirements.
8. IDMUI should provide state-of-the-art UIs for login, logout, and real-time messaging for the commands that the user executes.
9. the IDMUI should be able to run across multiple platforms including mobile tablets etc.
10. The IDMUI should be able to generate PDF reports of user activity.
11. The web app must use the same keystone authentication for user login and toles assignment
12.

Non-Functional Requirements

1 – IDMUI should use a fluid layout and provide ease of working it
2 – Real-time messages related to events of the IDMUI should be displayed in the message bar at the bottom of the UI.
3 – You must ensure the color scheme conforms to colour-blind guidelines. See the URL: https://accessiblyapp.com/blog/color-blind-accessibility/ for more details

Tools and Reading Material:

- Students must use Ubuntu Server (Latest Version) and any language of choice, preferably Python. For web apps, students may use Native Python Flask, PHP, or Python Django framework.
- Student must use MySQL Database Server
- Oracle Virtual Box: https://www.virtualbox.org/
- Ubuntu Latest: https://ubuntu.com/download/server
- Keystone Reference: https://docs.openstack.org/keystone/latest/
- OpenStack API Reference: https://docs.openstack.org/api-quick-start/

Supervisor:

Name: Arif Husen
Email ID: arif.husen@vu.edu.pk
Skype ID: vu.arifhrashid@outlook.com

ONLINE BAKERY MANAGEMENT SYSTEM

Project Domain / Category

Web Application

Abstract / Introduction

In today's digital age, the demand for online platforms has increased significantly, especially in the food and bakery industries. The Online Bakery Management System aims to provide an efficient, user-friendly, and automated solution for bakery businesses. This platform will allow bakery owners to manage their products, orders, and customers while offering a seamless shopping experience for customers who wish to browse and purchase bakery items online.

Functional Requirements:

1. User Registration: Customers can create an account by providing personal details such as name, email, password, phone number, and address.
2. Login/Logout: Registered users can log in using their email and password. Secure logout functionality should be provided.
3. Password Management: Users should be able to reset their passwords via email if they forget them.
4. Product Listing: Admins can add, update, and delete bakery products, including categories (e.g., cakes, pastries, bread), descriptions, prices, and images.
5. Product Availability: Admins can mark products as "in stock" or "out of stock" depending on inventory.
6. Product Search: Customers can search for bakery items based on product name, category, or price range.
7. Product Details: Customers can view detailed information about each product, including ingredients, sizes, price, and available quantity.
8. Add to Cart: Customers can add selected bakery items to their shopping cart for future purchase.
9. Edit Cart: Customers can update the quantity or remove items from their cart.
10. View Cart: Customers can view all items in their cart, including total price, before proceeding to checkout.
11. Place Orders: Customers can place orders after reviewing items in the cart. They will receive a confirmation message and order summary after successful placement.
12. Order History: Customers can view their past orders, including details such as order date, items purchased, total amount, and delivery status.
13. Order Tracking: Customers can track the status of their current orders (e.g., pending, in progress, delivered).
14. Payment Integration: The system will support various payment methods like credit/debit cards, PayPal, and cash on delivery.
15. Secure Payment: Transactions should be secure and encrypted to protect customer data.

16. Delivery Scheduling: Customers can select a preferred delivery date and time slot.
17. Delivery Tracking: Admins can update the delivery status, and customers can track their order delivery progress.
18. Address Management: Customers can save multiple delivery addresses for future orders.
19. Order Confirmation: Customers will receive email or SMS notifications when an order is placed, confirmed, or delivered.
20. Stock Alerts: Admins will receive alerts when stock levels of any product are low.
21. Promotional Notifications: Customers can receive notifications for discounts, new product launches, and special offers.
22. Order Management: Admins can view all orders placed, update order statuses, and manage order delivery schedules.
23. Product Management: Admins can manage the product catalog, including adding, updating, or deleting products.
24. User Management: Admins can view customer details, and manage or delete user accounts if necessary.
25. Sales Reports: Admins can view sales statistics, track revenue, and generate reports for specific time periods (e.g., daily, weekly, monthly).
26. Product Reviews: Customers can leave reviews and ratings for purchased products to provide feedback.
27. Review Management: Admins can moderate reviews and respond to customer queries or complaints.
28. Data Encryption: Sensitive data such as passwords and payment details must be encrypted.
29. Access Control: Only authorized personnel (admins) should have access to the management backend.
30. Backup System: A regular backup system should be in place to safeguard data in case of system failure.
31. The system should be fully responsive and work smoothly on mobile devices and tablets, providing customers with a seamless shopping experience across devices.
32. Sales Analytics: Generate reports showing top-selling products, sales trends, and customer purchase behavior.
33. Customer Analytics: Provide insights into user demographics, preferred products, and purchase frequency.
34. Inventory Analytics: Track stock levels, low inventory alerts, and restocking requirements.

Tools:

PHP, MySQL, Xammp, Notepad++.

Supervisor:

Name: Jibran Khan
Email ID: jibrankhan@vu.edu.pk
Skype ID: jibrankhanvu

Online Book Exchange Platform

Project Domain / Category

Web Based Application

Abstract / Introduction

The Online Book Exchange Platform is a web-based application designed to facilitate the exchange of books among users. The platform allows users to list books they own and are willing to exchange or give away, and browse books from other users. It promotes a community-driven system where book lovers can find, request, and exchange books with others based on location or interest.

The platform will have two types of users: Book Owners (who list their books) and Book Seekers (who browse and request books). The system will allow users to communicate and arrange for book exchanges via a messaging feature. This project promotes sustainability by encouraging reusing and sharing resources.

Functional Requirements:

Provide a bulleted list of functional requirements

Functional Requirements:

1. **User Registration & Login:**
    - Users can create an account using email.
    - Secure login functionality with password reset options.
2. **Book Listings:**
    - Users can list books they wish to exchange by entering details like title, author, genre, condition, and location.
    - Option to add photos of the book.
    - Option to add the new book URL like amazon, daraz etc.
3. **Search & Filter Books:**
    - Users can search for books by title, author, genre, or location.
4. **Request Book Exchange:**
    - Users can request a book from the owner by submitting their request from Listing details page.
    - Users can manage incoming and outgoing book requests from Book Requests Listing page.
    - Users (Owner & Seeker) can check their book requests made in the past.
    - Owners can reject OR approve any Seekers request.
5. **Book Exchange Status:**
    - Users can update the status of their book (e.g., Available, Reserved).
    - Once a owner accepts any seekers request, the book will automatically marked as not available
    - Notifications for users when a book is requested or confirmed for exchange.
6. **Messaging System:**
    - Users can communicate with each other through a built-in messaging feature to arrange the exchange details (time, location, etc.).
    - Message history is stored for future reference.(Optional)
7. **Book Reviews & Ratings:**

    - After a successful exchange, users can leave a review or rating for the book and the book owner.
8. **Book Wish List:**
    - Users can create a wish list of books they are looking for and receive notifications if listed by other users.

Tools:

Technologies Required:

- **Frontend:**
    - HTML5, CSS3, JavaScript (React.js or Angular.js)
    - Bootstrap or Foundation framework for responsive UI
- **Backend:**
    - Node.js with Express.js or Python with Django/Flask
    - RESTful API to handle user requests and data exchange
- **Database:**
    - MySQL or PostgreSQL for user and book data storage

Supervisor:

Name: Faizan
Email ID: fazitahir@vu.edu.pk
Skype ID: faizan.vu

Online Cosmetics Store

Project Domain / Category
Web application

Abstract/Introduction
The Internet plays a crucial role in our daily lives, facilitating communication, information sharing, and access to resources. In the modern world of e-commerce, numerous businesses are establishing their presence online. To support this trend, various e-commerce websites are being developed to market and sell products and services.

This project aims to develop a user-friendly website for an online cosmetics store, providing a platform for selling its offerings on the Internet. The website will allow users to purchase a wide range of products and services online. The primary objective of this online cosmetics store is to facilitate seamless interactions and transactions between the business and its customers.

The online cosmetics store project is essentially divided into two main components: the owner interface and the customer interface. The owner interface will support operations such as login, product and service uploads and edits, and order management. The customer interface will enable users to create an account, log in and out, and explore various products and services.

Functional Requirements:
Please remember the following instructions:

1. The owner needs to log in to their portal.
2. The owner has the ability to add or delete various products and services.
3. The owner is responsible for managing products and handling different customer orders.
4. Customers are able to create an account.
5. Customers can log in and log out of their accounts.
6. Customers will be able to view products filtered on the basis of type, price and Brand.
7. Customers have the option to add various products and services to their shopping cart.
8. Customers can also make changes to the contents of their shopping cart.
9. Customers will be able to check their past orders.

Tools:
PHP, MySQL, Notepad++, VS Code.

Supervisor: Warda Fiaz
Email ID: warda.fiaz@vu.edu.pk
Skype ID: live: cid.15e38e2220ca541b

Online Portal for Real Estate Management System

Project Domain:
Web Application

Abstract:
In manual system for Real Estate, the end-users face a lot of problems to search the residential and commercial property for hiring, buying and selling. They manually go and contact to the real estate offices for searching the property. The property dealers also have a very limited reach to their clients in manual system. This system will provide facility to the end-users to search and view the Residential and Commercial property online from any area. The system will also enable the registered property dealers to get the Admin permission and upload the property information to the site and publish it accordingly. The end-users visit the portal and view the required information online about the property. The end-use can also reserve or hire the property for buying and selling.

Functional Requirements:

End-User:

- The end-user visits the real estate web portal through domain name.
- The user searches the listing property
- The user searches the different categories of properties.
- The user can search the location, price range, number of bed rooms and amenities
- The user can search the property agent/property dealer profile
- The user can apply to hire, buy or sell the property
- The user can manage transaction and payment processing

Property Agent:

- The property agent will register to the portal first by providing the requisite information
- After approval, he can upload the property information to the portal after authentication
- Use of google maps to show the property detail.
- Provide relevant contact information
- Provide online payment system

Admin:

- The Admin will grant approval to the registered property agents
- The Admin will provide user name and password to the property agents

Tools:

For Front-End:

Html, CSS, JavaScript

For Backend:

PhP, My Sql

Supervisor Name: MIR SALAM KHAN
Email ID: mir.salam@vu.edu.pk
Skype ID: live:.cid.9a335075328c714a

Online Tourist Guide

Project Domain / Category

Web Application.

Abstract / Introduction

XYZ Technologies offers services based on user experiences, assisting individuals interested in determining the most suitable tourist destination for the season they plan to travel.
The main goal of XYZ Technologies is to automate various tasks in order to provide visitors with more information about different services.

You will develop a web-based application that will help its users provide valuable tourist guidelines based on existing data available as a repository. Major functionalities of the project will include:

Functional Requirements:

- Users will register and provide basic information at the time of registration. Registered users will be able to log in to the online website portal.

- User profile and dashboard will be set automatically when a registered user logs in for the first time.

- Admin will design a user dashboard along with the user's profile information, and all other user-related activities records will be stored and maintained in a database.

- Admin will add information on the web Homepage that will include information about different destinations (with images, descriptions, and activities) that are suitable for the users as per season in Pakistan, i.e., autumn, spring, summer, and winter.

- Users can get more information about different places of his/her interest, like mountains, deserts, or cities.

- Admin will manage the content on the Homepage that will include the most relevant information about visited destinations based on tourists' reviews and comments.

- Admin will be able to manage (Add/Delete/Edit) users' profile-related information, edit/delete destinations, and all relevant activities on the website.

- Admin can be able to share travel tips posted on our platform on different social platforms, like Facebook, Instagram etc.

- The system will allow users to create, save, and share travel interest details, including travel dates, accommodations, and activities.

- The system will provide an interface to contact third parties, like another service provider for the booking of hotels and flights.

- Users will be able to give feedback and review on shared destinations and services.

- Users can see local events, tours, and activities with details like the upcoming event calendar.

- The system will provide an interface to Google Maps API to display destinations and routes.

- The system will provide an interface to view, like, and comment on photos uploaded by users for each destination.

- The system will suggest recommendations to the user according to interest based on past activities and preferences.

- Users' dashboards will be designed for analytical insights on usage and trends.

Tools:

Backend: Python, Django
Frontend: HTML, CSS, JavaScript (optional frameworks like Bootstrap for styling and React or Vue.js for a more dynamic interface)
Database: SQLite/PostgreSQL
IDE: Visual Studio Code/PyCharm/Sublime Text

Supervisor:
Name: Muhammad Kamran Qureshi
Email ID: kamran.qureshi@vu.edu.pk
Skype ID: kamranqureshi99

Prayer Tracker

Project Domain / Category

Web Programming

Abstract / Introduction

Prayer (Nimaz) is one of the fundamental pillars of Islam, and it is obligatory for Muslims to perform five prayers daily. However, many individuals may miss prayers due to busy schedules or other reasons, leading to a backlog of missed (Qaza) prayers. To manage this, the web-based application will help users keep track of their daily prayers and any missed prayers. The application will serve as a digital prayer management tool, where users can view records of their completed and missed prayers, track their progress in performing Qaza prayers, and maintain accountability in their spiritual practice.

The Prayer Tracker will provide users with an intuitive interface to view the record of the five daily prayers (Fajr, Dhuhr, Asr, Maghrib, and Isha) and track any missed prayers.

Functional Requirements:

User Panel

- Users can create an account by providing their basic information, including name, email, or address, etc.
- Users can update and manage their profile, including resetting their progress for daily or missed prayers.
- If a user misses a prayer, they can mark it, and the system will automatically add it to their Qaza (leftover) prayers list.
- A daily report will show users their prayer performance, with color-coded indicators for completed, missed, or Qaza prayers.
- Users can view a total of missed or Qaza prayers, categorized by prayer type.

Admin Panel

- The admin can manage user accounts, including activating or deactivating accounts.
- The admin will have access to a dashboard that provides overall metrics, such as the number of users and missed or Qaza prayers across users.
- The admin can add or update content related to prayer guidance, including textual or video-based tutorials on the importance of Nimaz and managing Qaza prayers.

Tools:

HTML, CSS, JavaScript, jQuery, Bootstrap (Front-end)
MYSQL (phpMyAdmin) Database
PHP (Server-side programming)
XAMPP — Web Application Server

You are advised not to switch the tools. If you do so, you will handle the technical side yourself.

Note:

- These are the basic requirements of the application. Students may add further functionalities to make the application more useful.
- Virtual University of Pakistan (VU) will not provide any kind of hardware for this project; a student has to arrange the required hardware by himself/herself.
- VU will not pay for any license of the software, the libraries /toolkits/APIs used in this project.

Supervisor
Name: Musaddiq Hussain
Email ID: musaddiq.hussain@vu.edu.pk
Skype ID: kashaan.ali13

Responsive PHP-Based Web Application for Image-to-Hex Conversion with Batch Processing, GIF Handling, and Image Library

Project Domain / Category

Web Application Development / Embedded Systems Interface

Abstract / Introduction

This project aims to develop a feature-rich, responsive PHP-based web application that enables users to convert images into hex code for use with various embedded system displays (TFT, OLED, LCD).

The application will allow users to upload images, customize settings (such as image size, color inversion, and compatibility with different display libraries), and process multiple images at once. Additionally, it will include functionality to handle animated GIFs, splitting them into individual frames for hex conversion.

An integrated image library will provide users with pre-loaded images and animations for testing purposes. The application will also offer a REST-based API, allowing external applications to access the hex conversion service programmatically.

This project will provide a comprehensive solution for developers and hobbyists working with embedded systems, offering flexibility, ease of use, and automation.

Functional Requirements

1. **FR1**: Users can upload images (JPEG, PNG, GIF) for conversion to hex code.
2. **FR2**: Provide options to resize images to match different display resolutions.
3. **FR3**: Allow users to invert image colors before conversion to hex.
4. **FR4**: Support various LCD display libraries (Adafruit GFX Library, U8g2 Library, TFT_eSPI Library, LiquidCrystal Library).
5. **FR5**: Convert animated GIFs into individual frames and generate hex code for each frame.
6. **FR6**: Batch processing of multiple images for hex conversion in one go, with downloadable results.
7. **FR7**: Integrated image library with pre-stored images and animated GIFs available for conversion.
8. **FR8**: Export hex code in formats suitable for various embedded libraries (e.g., C arrays for Arduino).
9. **FR9**: REST-based API allowing third-party applications to use the image-to-hex conversion service.
10. **FR10**: Responsive web design to ensure optimal usability on devices of different screen sizes.
11. **FR11**: Live image preview after conversion, showing both static and animated image hex output.

Tools

- **Development Environment**: XAMPP/WAMP for local PHP development and web server setup.
- **Languages**:
  - **Backend**: PHP for core logic and REST API development.
  - **Frontend**: HTML, CSS, and JavaScript for building a responsive, user-friendly interface.
- **Database**: MySQL or SQLite for storing pre-loaded images and user preferences.
- **Browser-based Testing**: Google Chrome, Mozilla Firefox for cross-platform testing.

Supervisor:

Name: Khaqan Khawer
Email ID: khaqan@vu.edu.pk
Skype ID: imkhaqan

Used Laptops Buying and Selling Store

Project Domain / Category

Web Application

Abstract / Introduction

The main aim of this project is to develop an interactive and effective website for facilitating the needs of used Laptops buyers and sellers. It will be like an e-laptop store website where laptops can be bought and can be sold from the comfort of home through the Internet. An online laptop store is a virtual store on the Internet where customers can view the available laptops for sale and can select laptops of their interest as well as add a laptop on the website for selling it. The platform allows users to easily create accounts, manage their profiles, and list laptops with detailed specifications, including images and condition descriptions. The website will feature a user-friendly interface that allows individuals to browse through a wide selection of laptops using advanced search filters to help users find specific models based on criteria like price, brand, model and condition etc. After selecting a laptop of his/her choice, a customer can contact the owner of the laptop whose information will be visible to the buyer on sending a purchase request to the seller.

Functional Requirements:

There will be three categories of Website users:

- Guest (Unregistered User)
- Registered User
- Administrator (Admin)

Guest (unregistered user) user will be able to just view the available Laptops and can search them on the website according to his/her need.
Registered user will have the privileges to place an order for a laptop that is available on the website as well as can add a laptop on the website for selling it. Administrator (Admin) is the super user of the website who can manage everything on the website.

- **User Module:**

a) **User Registration and Sign In:** There will be a proper signup interface for unregistered users to register on the website. A registered user will be able to login to the website by entering the correct credentials in the sign in interface.

b) **View Laptops:** All available Laptops will be available on the website with proper interface. Any registered user or guest can view information of available laptops and can view the complete details of its brand, model, images, demanded price, laptop owner's contact & location and other necessary details. The guest can only view and search the available information and cannot make any upload or any purchase request without any registration on the website.

c) **Search Laptop:** Any user registered or unregistered can search for the available laptops on the website. The search can be done using advanced search filters like price, brand, model and condition etc. If user requirement meets, then system will show results in proper format.

d) **Upload Laptops:** A registered user will need to upload the laptop details if he/she wants to sell a laptop. After login to the website, the user will upload all the necessary details i.e. laptop model, brand, price, images etc. of the laptop on the website.

e) **Update details of uploaded Laptops:** If a registered user has uploaded a laptop on the website, he/she will be able to edit and update the price or any other details of that laptop. The registered user will also be able remove the laptop from the website.

f) **Update profile:** Upon successful login to the website by a registered user, he/she will be able to update any of his/her profile information and can update his/her account password.

g) **User Review and Feedback:** Any registered user will be able to submit his/her review about the purchased laptop and can give feedback about it to its previous owner.

h) **User Complain:** In case of any fraud or serious issue, any registered user will be able to submit complain about the concerned to the admin. After verification of the matter, the admin can either give warning to the concerned or can block the concerned account on the website.

i) **Give Rating:** Customers can give a rating to the concerned seller after the completion of purchase request according to his/her satisfaction.

j) **Buy Laptops:** A user must log in successfully to the website to place an order for buying a laptop. A registered user needs to fill all the order details for buying the laptop of his/her choice. After filling the details, user will be moved to the payment section.

- **Admin Module:**

a) **Login:** Using valid login credentials, admin need to login into the system to access the system.

b) **Manage Laptops:** Admin can upload the laptop's information, view all the added laptops online with their details by the registered users. Admin can block and unblock any laptop for displaying on the website.

c) **Admin Dashboard**: Admin can view the detailed summary of everything such as count of registered users, registered user details, total sold laptops along with buyer and seller details, total available laptops etc.

d) **Manage Users:** All the registered user details will be displayed to the admin. Admin will accept / reject the user registration requests and can block or unblock any user for uploading or buying a laptop on the website.

e) **Report Generation:** The admin will be able to generate a complete report of total sales of laptops purchased on daily, weekly, and monthly basis.

- **Payment Module:**

  Upon successful completion of any order, customers can either pay the amount to the concerned delivery person on the spot or can transfer the amount to the concerned laptop seller's account number available on the website. The responsibility of the delivery will be on the seller and the responsibility of verifying the selected laptop will be on the customer respectively.

  [**Note:** Student can add/enhance requirements as per need and keeping the time span and scope in view.]

  Tools:

  ASP.NET, C#, React JS, Node JS, HTML, CSS, JavaScript, Ajax, jQuery, Bootstrap, MS SQL Server

  Supervisor:

Name: Syed Hassan Ali Shah
Email ID: hassan.ali@vu.edu.pk
Skype ID: syed.hassan.ali.sha@outlook.com

VU Alumni-Student ConnectBook

Project Domain/Category:

Web Programming

Abstract/Introduction:

The ' VU Alumni-Student ConnectBook' web application is designed to strengthen the relationship between alumni, current students, and the university administration. This platform serves as a bridge, allowing alumni to stay engaged with their alma mater or University, contribute to current students' academic and career development, and participate in university events. Additionally, the application will enable university administration to efficiently manage alumni data and leverage it for various purposes such as mentoring programs, events, success stories and job placements.

Functional Requirements:

- **Registration and Profile Management:** Alumni and students must register and create profiles. Alumni profiles include information like graduation year, degree, current occupation, and contact details. Students' profiles include their current course, year of study, and interests.
- **Alumni-Student Mentorship**: Alumni can offer mentorship to students by volunteering through their profiles. Students can request mentorship in specific fields or career guidance. The platform will facilitate communication between the mentor and mentee.
- **Event Management:** The administration can post upcoming university events, including seminars, reunions, alumni dinners and guest lectures. Alumni and students can register for events, and the system will track attendance. Post-event feedback forms and surveys can be generated automatically.
- **Job and Internship Portal:** Administration and Alumni can post job and internship opportunities directly onto the platform. Students can apply for these opportunities through the application, with their profile details auto-filled.
- **Alumni Rewards System:**
    - ❖ Alumni who actively participate in mentorship programs, assist students in securing jobs, or contribute significantly to university events can earn reward points.
    - ❖ These points can accumulate over time and be redeemed for various benefits, such as recognition certificates, access to exclusive university events, or discounted courses.
    - ❖ The platform will feature an "Alumni of the Month" section, where top-performing alumni are highlighted based on their contributions and points earned.
    - ❖ The university administration can offer special rewards to outstanding alumni, such as honorary titles, public recognition at university events, or invitations to speak at seminars.
    - ❖ During alumni dinners or gatherings, prizes can be awarded to top contributors as a token of appreciation, further motivating alumni to stay engaged with the university.

- **Discussion Forums:** Forums where alumni and students can discuss topics related to careers, education, and university life. Moderated by the administration to ensure discussions remain professional and relevant.
- **Data Analytics for Administration:** The administration will have access to dashboards showing alumni engagement metrics, event participation rates etc. Exportable reports for data-driven decision-making.
- **News and Updates:** A news section where the administration can post updates about university achievements, notable alumni, and other relevant news.

Tools and Technologies:

Frontend: HTML5, CSS3, Bootstrap, AngularJS
Backend: PHP
Database: MySQL
Additional Libraries: jQuery for dynamic user interface elements

Note: Kindly read the proposal carefully and please feel free to discuss any project- related questions before selecting it. The supervisor or university is not obligated to provide any paid project development resources

Supervisor:
Name: Tooba Ahmad
Email ID: tooba.khan@vu.edu.pk
Skype ID: ToobaA_CS_Instructor

Web-Based Animated Simulator for Sorting Algorithms

Project Domain / Category

Web Application

Abstract / Introduction

The aim of this project is to develop a web-based simulator that demonstrates the workings of various sorting algorithms step-by-step. Students will create an interactive platform where users can input data, visualize how different algorithms operate, and understand the principles behind these fundamental computer science concepts. The animated simulator will bridge the gap between theoretical knowledge and practical understanding of these algorithms. This project will give a deeper understanding of different sorting algorithms and show their significance in computer science.

Following is brief description of some Sorting Algorithms:

- **Bubble Sort:** Repeatedly swaps adjacent elements. Complexity: $O(n^2)$.
- **Selection Sort:** Selects the smallest element and moves it to the sorted region. Complexity: $O(n^2)$.
- **Insertion Sort:** Inserts elements into a sorted section. Complexity: $O(n^2)$.
- **Merge Sort:** Divides and conquers by merging sorted halves. Complexity: O(n log n).
- **Quick Sort:** Partitions around a pivot for efficient sorting. Complexity: O(n log n) average.

Functional Requirements

The following functional requirements outline the key features and functionalities of the Animated Simulator. These requirements ensure that the simulator meets its educational objectives while providing a user-friendly and engaging experience.

- **Main Dashboard and input Selection:**
  - The simulator must have a dashboard that allows users to navigate easily between different algorithms and settings.
  - Users should be able to input list of numbers, select and apply a sorting algorithm from a list on input data.
  - Input validation must ensure that the data entered is numeric and within specified limits (e.g., maximum array size).
- **Algorithm Selection and Implementation:**
  - Users should have the option to choose from the following algorithms:
    - Sorting Algorithms: Selection Sort, Insertion Sort, Bubble Sort, Quick Sort, and Merge Sort.
  - The simulator should provide a brief description of each algorithm and its time complexity.
  - Implement the logic for sorting algorithms in PHP.

  Note: Students can also add more algorithms as per their requirements.
- **Algorithm Execution and Real-Time Visualization**
  - The simulator must visually depict the state of the array or list during execution, highlighting comparisons, and swaps.
  - Animations should be smooth and clearly illustrate each step of the algorithm.
  - The simulator at the end should display both unsorted and sorted list of numbers. Include a summary of algorithm complexities (best, worst, and average cases).
  - You can use JS libraries for visualization like D3.js , P5.js etc.,
- **Animation Controls and Control Options:**

    - Users must be able to start, pause, and reset the simulation at any point during the algorithm execution.
    - Speed control slider to adjust the speed of the visualization.
        - Receive user input from the frontend.
        - (e.g., tracking most used algorithms, average input size).
- **Database using Xampp Server**
    - Store algorithm selection and input data in MYSQL Database for future analysis.
- **Help and Documentation**
    - Provide a help section explaining how to use the simulator.
    - Include descriptions of each algorithm, including its time complexity and use cases.

Tools:
HTML, CSS, Bootstrap, JavaScript, PHP, Visualization Libraries (D3.js, P5.js etc), Xampp Server, MySQL Database

Supervisor:
Name: Umra Naeem
Email ID: umra.naeem@vu.edu.pk
Skype ID: umra.naeem

Web-based Gardening Care App

Project Domain / Category

Web App

Abstract / Introduction

Gardening Care Application is a web-based application for domestic (home-owners) designed to provide comprehensive gardening guidance and support to users of all levels. The application offers personalized plant care recommendations, interactive features, and a community platform to connect with fellow gardening enthusiasts. By leveraging advanced technology and user-centric design, this application aims to simplify the gardening process and foster a love for nature.

Functional Requirements:

1. **User Registration and Profile Management:**
    - Allow users to create accounts, update personal information, and manage their gardening preferences. The users should include Gardener, Supervisor, Home Owner and System Admin.
2. **Plant Database:**
    - Maintain a comprehensive database of plants, including their types, characteristics, care requirements, age, growth stages etc. The plants are mainly of three types; flower plants, vegetable plants and fruit plants.
    - Enable users to search for plants by name, category, or specific attributes.
3. **Personalized Plant Care Recommendations:**
    - Utilize user-provided data (location, climate, soil type) to offer tailored plant care advice.
    - Suggest appropriate watering schedules, fertilization plans, and pest control measures.
4. **Plant Tracking and Monitoring:**
    - Provide features to track plant growth, record observations, and monitor health indicators.
    - Allow users to set reminders for tasks like watering, repotting, or pruning.
5. **Interactive Tools and Resources:**
    - Offer interactive tools such as plant identification guides, garden planners, and disease diagnosis assistance.
    - Provide access to educational resources, articles, and tutorials on various gardening topics.
6. **Community Features:**
    - Facilitate a community forum / group at Facebook and/or WhatsApp for users to share experiences, ask questions, and connect with other gardeners, home owners etc.
    - Enable users to create and join gardening groups based on interests or locations. There should be location sharing service for social media platforms.
7. **Alerts:**
    - Send timely notifications among all users for important tasks, weather updates, or plant-related alerts.
8. **Integration with External Services:**
    - Consider integrating with weather APIs to provide localized weather forecasts and gardening tips.

- Explore partnerships with gardening supply stores or nurseries for product recommendations and discounts.

Tools & Technologies:

- **Frontend Development:**
  - HTML, CSS, JavaScript
  - React or Angular (for a modern, component-based framework)
  - Material UI or Bootstrap (for responsive design and UI components)
- **Backend Development:**
  - Node.js or Python (for server-side logic and API development)
  - Express.js or Django (for web frameworks)
  - MongoDB or PostgreSQL (for database management)
- **Cloud Platform:**
  - AWS, GCP, or Azure (for hosting the application and scaling resources)
- **Additional Tools:**
  - Git (for version control)
  - NPM or Yarn (for package management)
  - Webpack or Parcel (for bundling and optimization)
  - Testing frameworks (e.g., Jest, Mocha)

Supervisor:

Name: Haseeb Akmal
Email ID: haseebakmal@vu.edu.pk
Skype ID: HaseebAkmal

An Intelligent Auto Grader Learning Management System for Academic Courses

Project Domain / Category

Web Application

Abstract/Introduction

The shift towards digital education has brought both opportunities and challenges, particularly in the domain of students' formative assessment. The manual grading of students' assessments, especially open-ended questions, is time-consuming and subject to human biasness.

Traditionally, grading of subjective questions such as short answers and essays demands significant time and effort from educators, often leading to delays in feedback, inconsistency, and a lack of personalized attention to each student.

To address these challenges, we proposed an Intelligent Auto Grader Learning Management System (LMS) for Academic Courses, integrating advanced Natural Language Processing (NLP) techniques with Machine Learning (ML) and Deep Learning (DL) models. The system leverages ML/DL models to analyze students' performance, identify weak areas, and evaluate subjective questions accurately and effectively.

By providing auto grading and real-time feedback to students, improves student's engagement, leverages learning experiences, and promotes academic success to the students.

Functional Requirements:

An Intelligent Auto Grader Learning Management System for academic courses that evaluates both objective and subjective questions, allows all students to be assessed simultaneously with the same number of test administrations throughout the semester.
The proposed system will have the following main users:
Admin Teacher and Student.

1. Registration module: It will facilitate the registration process for students and teachers. Admin will approve and perform activation of the students and teachers accounts and registration requests.
2. Login Module: After successful registrations, all types of the users will be able to login to the system by using their registered email and password.
3. Your application will assist the teacher with the auto scheduling of quizzes and assignments of different subjects including Python, C++ and Operating Systems.
4. Additionally, application will generate objective and subjective questions for different computer science courses, including Python, C++, and Operating Systems.
5. Teachers can compile a labeled dataset of existing objective and subjective questions along with their correct answers.
6. Learning Management System can auto generate objective questions, and matching questions and an option to specify difficulty level and cognitive levels for each subjective question.
7. The Learning Management System can utilize the Mohler dataset for analytical purposes.
8. Learning Management System can align grading with a pre-defined rubric to ensure consistency.

9. Auto grader can leverage transformer-based Deep Learning models like BERT, GPT, RoBERTa and DistilBERT and XLNet to assess the similarity between a student's answer and a model/desired answer.
10. Learning Management System can assign scores based on semantic similarity by understanding context and meaning of subjective questions rather than simple keyword matching.
11. The System can provide personalized feedback to students, indicating where they went wrong and how they can be improved.
12. The System should allow teachers to export grades and performance reports in common formats (e.g., CSV, Excel, PDF).
13. For evaluation of different Deep Learning and Machine learning models accuracy, the System should use the metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).
14. Firstly, System get each set of student answers and after auto evaluation compare the predicted grades (from the model) to the actual grades (provided by instructors) and apply the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) for the model accuracy.
15. The System should be able to compute the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) for an entire batch of student answers by summarizing the performance of the Machine Leaning/Deep Learning model for a complete set of assignments.
16. The system should allow teachers to define acceptable thresholds for Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). If the evaluation metrics for Machine Learning/Deep learning exceed the thresholds, the system should trigger an alert, indicating that manual review may be necessary for certain assignments.
17. The system should automatically flag questions that have a high percentage of incorrect answers, indicating that they may be particularly challenging or unclear to students.
18. Teachers should be able to use classification tools on different subject areas (e.g., "Python," "C++," "Operating Systems") and should categorize questions after assessments by labeling them as "easy", "medium" or "hard".
19. Teacher be able to create rubrics for different types of subjective questions (e.g., essays, short answers, case studies).
20. Teacher can manually adjust the weight assigned to each rubric criterion (e.g., content could be worth 50%, grammar 20%, creativity 30%).
21. Teacher can generate reports showing how students performed on each question.
22. The system should provide teachers with metrics for each question, including the average score, standard deviation, and difficulty index (the percentage of students who answered correctly).
23. Students should be able to attempt both subjective and objective questions within a single assessment interface, allowing them to seamlessly navigate between question types and submit their responses in one consolidated format.
24. Students can receive instant feedback on both objective and subjective question responses, including correct answers and explanations, to support learning and self-assessment.
25. Students should have the ability to track their progress and performance over time through a personal dashboard that displays their scores, feedback on assessments, and areas for improvement.
26. Enable students to share their accomplishments or ask questions through social media integration, fostering a sense of community and encouraging wider discussions.

27. Ensure the platform is optimized for mobile use, allowing students to access learning materials and assessments on a variety of devices, including smartphones and tablets.
28. Provide a help center, FAQs, or chat support to assist students with any issues they encounter or questions they may have about the platform.

Tools:
JSP, PHP, Python, JavaScript/HTML/CSS, MySQL, PyTorch, Keras, Padas, TensorFlow, BERT, RoBERTa, DistilBERT, XLNet, Transform.

Supervisor:
Name: Muhammad Umar Farooq
Email ID: umarfarooq@vu.edu.pk
Skype ID: live:umarvc

Anomaly Detection System in Blockchain

Project Domain / Category

Blockchain/Machine Learning

Abstract / Introduction

With the growing adoption of Blockchain technologies, ensuring the integrity and security of decentralized systems like Bitcoin has become critical. This project will focus on providing an efficient method to identify malicious activity within the Blockchain, improving the security and reliability of decentralized networks.

This project proposes developing an **anomaly detection system** to detect 51% attack in real-time by analyzing Blockchain transaction data using machine learning techniques. The proposed solution will involve injecting artificial anomalies into a Bitcoin dataset and then building machine learning models such as **SVM, Random Forest**, **AdaBoost**, and **XGBoost** to detect those anomalies.

Functional Requirements:

The functional requirements define the features that the system must implement to fulfill its purpose.

1. **Data Loading and Processing**:
    - The system will load the Bitcoin Blockchain dataset containing transaction data and relevant block information. (Dataset link will be provided)
    - It will preprocess the dataset by cleaning, transforming, and handling missing values or inconsistencies.
2. **Anomaly Injection**:
    - Artificial anomalies mimicking a 51% attack will be injected into the dataset. These anomalies will affect key features such as:
        - **Confirmations**: Set anomalous blocks to have zero or unusually low confirmations.
        - **Height**: Duplicate or irregular block heights.
        - **Number of Transactions**: Set anomalously high or low transaction volumes in certain blocks.
        - **Difficulty**: Simulate reduced difficulty in attack scenarios.
        - **Timestamps**: Create irregular or overlapping timestamps for the blocks.
3. **Anomaly Labeling**:
    - The system will label the dataset with a binary indicator (Anomaly column), where 1 represents an anomalous block, and 0 indicates normal blocks.
4. **Machine Learning Model Building**:
    - The system will implement four machine learning algorithms for anomaly detection:
        - **Support Vector Machine (SVM)**

        - **Random Forest Classifier**
        - **AdaBoost Classifier**
        - **XGBoost Classifier**
    - These models will be trained to identify patterns associated with a 51% attack based on the labeled dataset.

5. **Model Evaluation**:
    - Each model will be evaluated using performance metrics such as:
        - **Accuracy**: Proportion of correctly classified instances.
        - **Precision**: Proportion of correctly predicted positive cases (anomalies).
        - **Recall**: Ability to detect true positives (actual anomalies).
        - **F1-score**: Harmonic mean of precision and recall.
    - Confusion matrices will be generated for each model to visualize the distribution of true positives, true negatives, false positives, and false negatives.

6. **Comparison of Models**:
    - The system will compare the performance of the models and identify the best algorithm based on the above metrics.
    - The results will be visualized using bar plots showing the performance metrics (accuracy, precision, recall, F1-score) for each model.

Tools:

Python (programming language)
Scikit-learn (Library)
Kaggle or Jupyter Notebook (open-source web application) or Google Colab
Matplotlib (library)
Numpy (library for the python)

Supervisor:

Name: Fouzia Jumani
Email ID: fouziajumani@vu.edu.pk
Skype ID: fouziajumani

Book Genre Classification using Text Analysis in Python

Project Domain / Category

Text Analysis/Classification

Abstract / Introduction

With the growing number of e-books available on digital platforms, categorizing books by genre is essential for better searchability and organization. This project, “Book Genre Classification using Text Analysis in Python”, aims to classify books into genres based on their textual descriptions. The system will predict genres such as fiction, non-fiction, mystery, and science fiction by analyzing book descriptions using basic text processing techniques. This will be achieved through simple text classification models and natural language processing techniques implemented in Python.

Functional Requirements:

1. **Data Collection:**
    - Gather a dataset of book descriptions and their corresponding genres from online sources like Goodreads or Kaggle.
    - The dataset should include a variety of genres (e.g., fiction, non-fiction, mystery, science fiction).
2. **Data Pre-processing:**
    - Clean and prepare the text data to make it suitable for classification. This involves:
        - Converting text to lowercase.
        - Removing special characters (e.g., punctuation).
        - Tokenizing text into individual words.
        - Removing common stop words (e.g., "the", "is").
        - Applying lemmatization to reduce words to their root form.
3. **Text Classification Model:**
    - Use Bag of Words (BoW) or TF-IDF (Term Frequency - Inverse Document Frequency) to convert the text into numerical data.
    - Implement a basic machine learning classifier, such as Naive Bayes or Decision Trees, to categorize books based on their description.
4. **Genre Prediction:**
    - After training the model on the dataset, the system will predict the genre of new books based on their description.
    - Display the predicted genre and confidence level (e.g., "Mystery, 90% confidence").
5. **Book Recommendation:**
    - Based on the predicted genre, recommend similar books from the same genre.

Note:

- **More Functional requirements can be added to each deliverable.**

- **A detailed document for each deliverable, required tools and libraries to be used will be provided later after the selection of the project.**
- **Prior knowledge of related concepts is required. Please thoroughly study the proposal and then opt for the project.**

Tools:

- Python: (Programming language for building the system.)
- Natural Language Toolkit (NLTK) or spaCy: (For text preprocessing tasks like tokenization and lemmatization.)
- Scikit-learn: (For implementing machine learning algorithms.)
- Pandas: (For handling the dataset.)
- Matplotlib or Seaborn: (For visualizing classification accuracy and results.)

Supervisor:

Name: Rizwana Noor
Email ID: rizwana.noor@vu.edu.pk
Skype ID: rizwana.noor77

Brain Tumor detection using MRI Scans

Domain/Category

Deep learning/ AI/ Web/ Image Processing

Introduction

Brain tumors are life-threatening conditions that require early detection for effective treatment. Magnetic Resonance Imaging (MRI) scans are commonly used by radiologists to detect brain tumors, but the manual interpretation of these scans can be time-consuming and prone to human error. Automating brain tumor detection through machine learning and deep learning can significantly enhance diagnostic accuracy and efficiency. This project aims to develop a deep learning-based image classification system that detects brain tumors from MRI images. Using a dataset of brain MRI scans, the system will classify the images as either tumor or non-tumor, providing radiologists with a reliable tool for diagnosis.

Functional Requirements:

1. **Data Collection:** Gather a diverse dataset of images which labeled Brain tumor detection using MRI scans
2. **Data Preprocessing:** Prepare and clean the dataset by resizing images, normalizing pixel values, and organizing the data for training and testing.
3. **Datasets:**
   Various Datasets are available on Kaggle and github for Brain Tumor MRI dateset
   https://www.kaggle.com
   https://github.com
4. **Model Selection:** Choose a deep learning model for Brain Tumor Detection suing MRI Scans. Popular choice is the Convolution Neural Networks (CNNs).
5. **Model Training:** Train the selected model using the preprocessed dataset. Fine-tuning on a pre-trained model can significantly improve accuracy. **The** CNN will be designed and trained to detect tumors in MRI images. The network will learn in the images to distinguish between tumor and non-tumor cases.
6. **Model Evaluation:** Evaluate the model's performance using metrics such as accuracy and precision. Fine-tune the model to achieve the desired level of accuracy.
7. **Webapp development:** develop a webapp using Flask.
8. **User Interface:** Create a user-friendly interface that include that provide option for image upload and display the result as Tumor or Non-tumor.
9. **Deployment:** Deploy the model to a webapp.
10. **Testing and Validation:** Conduct thorough testing on various datasets and in real-world scenarios to ensure the system's accuracy and reliability.

Tools: Python, Tensorflow, Keras, Flask, Jupyter Notebook, google colab, opencv, os, scikit-learn, matplotlib among others

Supervisor:

Name: Muhammad Luqman
Email ID: m.luqman@vu.edu.pk
Skype ID: mluqman.vu

Developing a Career Counseling Platform Using Artificial Intelligence

Project Domain / Category

Artificial Intelligence: machine learning, natural language processing, and recommendation systems, **Educational Technology**

Abstract/Introduction

In today's rapidly evolving job market, students and professionals often face challenges in selecting the most suitable career paths. A career counseling platform powered by Artificial Intelligence (AI) can provide personalized recommendations by analyzing individual preferences, skills, interests, and job market trends. This project aims to develop an AI-based platform that offers career guidance by assessing users' profiles, matching them with potential career options, and suggesting relevant skills and educational resources. The platform will use machine learning algorithms to continuously improve its recommendations based on user feedback and job market changes. Additionally, the platform will integrate natural language processing to allow users to interact through text-based queries, offering real-time counseling support. This AI-powered career counseling platform aims to provide accurate, personalized career recommendations, equipping users with the knowledge and skills required to succeed in a competitive job market.

Functional Requirements:

1. **User Profile Creation:**
    - Users (students/professionals) can create profiles with personal information, skills, educational background, interests, and career goals.
2. **Career Path Recommendation:**
    - The AI system suggests career paths based on user profiles, preferences, job market trends, and skill gaps.
3. **Skills Gap Analysis:**
    - The platform assesses user profiles to identify skill gaps and recommends specific courses, certifications, or training programs to bridge these gaps.
4. **Job Market Analysis:**
    - The platform monitors the job market trends using web scraping and public APIs to stay updated on emerging job opportunities and evolving career fields.
5. **Interactive Career Guidance:**
    - Users can ask questions related to career options via a chatbot powered by natural language processing (NLP), which offers real-time guidance.
6. **Career Planning Tools:**
    - Users can access tools for career planning, such as goal setting, milestone tracking, and skill development roadmaps.
7. **User Feedback Loop:**
    - The system will gather feedback on the relevance and accuracy of its recommendations to improve its learning algorithm.
8. **Personalized Notifications and Updates:**
    - The platform will send notifications regarding relevant job opportunities, skill development courses, or industry trends based on user profiles.
9. **Multi-User Role Support:**
    - Different user roles like students, professionals, career counselors, and administrators will be supported, with appropriate access rights for each.

Note: **Attendance at Skype sessions is mandatory for discussing the project with the supervisor; failure to attend may result in the project not being accepted.**

Tools/language:

1. **Programming Languages:**
   - Python (for machine learning algorithms, NLP, backend development)
   - JavaScript (for front-end development)
2. **Machine Learning Libraries:**
   - TensorFlow/PyTorch (for building recommendation models)
   - Scikit-learn (for machine learning algorithms)
3. **Natural Language Processing:**
   - SpaCy or NLTK (for text processing and chatbot development)
   - GPT-based models for generating responses in real-time queries
4. **Database:**
   - MongoDB (for user profiles, career data)
   - MySQL or PostgreSQL (for structured data storage)
5. **Frameworks:**
   - Django/Flask (for backend development)
   - React/Angular (for frontend development)
6. **APIs and Web Scraping:**
   - Beautiful Soup, Scrapy (for web scraping job market data)
   - Job market APIs (e.g., LinkedIn, Indeed API for job opportunities)
7. **Cloud Services:**
   - AWS or Google Cloud (for deployment and scalable infrastructure)
8. **Version Control:**
   - GitHub/GitLab (for code collaboration)
9. **DevOps:**
   - Docker (for containerization)
   - Jenkins (for continuous integration and deployment)

Prerequisite:

To help students grasp the project problem concepts, they will be required to complete a brief course on key AI/machine learning topics, along with developing the Software Requirements Specification (SRS) and initial design documentation. Additionally, relevant course materials and resources will be shared during Skype sessions for further guidance.

Supervisor:

Name: Dr. Saima Munawar
Email ID: saima.munawar@vu.edu.pk
Skype ID: saima.vu1

Emotion Recognition from Text in Android

Project Domain / Category

Mobile Apps

Abstract / Introduction

In today's fast-paced world, people frequently use text messaging on social media, messaging apps, and other online platforms to convey their emotions. It can be difficult to recognize these feelings accurately, particularly in environments like social media monitoring, mental health support, and customer service etc. The goal of this project is to develop an Android application that uses Natural Language Processing (NLP) techniques to analyze text input from user and predicts the corresponding emotions. The application will provide a simple and easy-to-use interface for text input and user-friendly formatted results presentation.

Functional Requirements:

I. **Users:** The app must support two types of users: Admin and Analyst.
   1. Admin: Responsible for managing user accounts, handling technical issues, monitoring app performance, and maintaining the emotion detection model etc.
   2. Analyst: General users who input text to detect and analyse emotions. They can view results, access text history, and share or export emotion insights.

II. **Authentication:** App should provide registration and login pages for admin and designers; store credentials at Firebase Authentication.

III. **Text Input:** Analysts should be able to input text manually or through voice-to-text for emotion analysis.

IV. **Emotion Detection:** The system should detect emotions from the input text using a machine learning model, displaying the relevant emotion labels and confidence scores.

V. **Emotion History:** Analysts should be able to view their previously analysed texts and the associated detected emotions in a history log.

VI. **Real-time Analysis:** Emotion detection results should be displayed in real-time after text input, allowing users to immediately see the detected emotions.

VII. **Customization:** Analysts should have the option to customize which emotion categories they want the system to detect from their text.

VIII. **Sharing Results:** Analysts should be able to share their detected emotions via social media, email, or other messaging apps in various formats, such as text or images.

IX. **Help and Support:** The app should provide help resources, FAQs, and tutorials to guide analysts in understanding the features and resolving issues.

Workflow:

The basic work flow of this project is as follows;

1. **Project Setup:**
   - Set up IDEs (PyCharm for model development/training and Android Studio for app development).
2. **Dataset Preparation:**
   - Load and preprocess GoEmotions dataset.
   - Split the dataset into training, validation, and test sets.
3. **Model Development**
   - Use a pre-trained transformer model (e.g., BERT) for emotion classification.
   - Evaluate the model and export it as TensorFlow Lite for mobile deployment.

4. **App Development:**
    - Use Java/Kotlin to create app's UI and integrate the model for emotion recognition.
    - Implement features for real-time emotion detection, text history, and result sharing etc.

Tools:

1. **IDE:** Android Studio & PyCharm
2. **Programming Language:** Java/Kotlin & Python
3. **Databases:** Firebase Real-Time/Cloud Fire-Store & SQLite/Room
4. **Dataset:** GoEmotions (https://github.com/google-research/google-research/tree/master/goemotions)
5. **Pre-trained Model:** BERT (https://github.com/google-research/bert)
6. **Model Deployment:** LiteRT i.e., TensorFlow Lite (https://ai.google.dev/edge/litert)

Note: VU will not pay for any license of software/library/toolkit/API used in this project.

Supervisor:

Name: Muhammad Imran Afzal
Email ID: imran.afzal@vu.edu.pk
Skype ID: imranafzal126

Human Skin Disease Detection System Using CNN

Project Domain / Category

Software Application

Abstract / Introduction

Skin diseases are widespread, affecting millions of people globally. Many individuals suffer from various types of skin conditions that often pose hidden risks, leading not only to self-esteem issues and psychological distress but also increasing the risk of skin cancer. According to the World Health Organization (WHO), approximately 30% to 70% of the population is affected by skin diseases, with many lacking awareness about the different types.

To address this issue, you are required to develop a Skin Disease Detection System using Convolutional Neural Networks (CNN). The goal of this project is to provide users with an easy way to identify potential skin diseases, allowing them to take proactive measures for treatment. It also assists doctors by offering preliminary insights into the type of skin condition, which helps streamline and improve the efficiency of diagnosis.

The system will have two types of users.

1. End User
2. Admin

1. **End User**

The End users must register and log into the system. Once logged in, they can upload an image of their skin condition, and the system will automatically classify the disease shown in the image. Additionally, users can view a list of doctors based on the identified skin condition and provide can provide feedback.

2. Admin

The admin has the ability to log in with their credentials, can manage users(add/update/delete), and can manage the doctors by adding, updating, or removing entries. They can also access the feedback submitted by users.

Functional Requirements:

The project will comprise of the following functional requirements: -

1. **Data pre-processing**
   Image data pre-processing is a crucial step that can significantly influence the performance of your model. Images of skin lesions are collected and annotated with disease classifications. To enhance the dataset, data augmentation techniques create variations of existing images. Next, images are standardized in size to ensure uniform input for the model and pixel values are normalized to a consistent range. The dataset is then divided into training, validation, and test sets. To address class imbalance, techniques may be employed to manage underrepresented diseases.
2. **Model training and testing**
   After pre-processing, the dataset is ready for training. You're required to distribute 80% data for model training, so the model can adapt to the maximum of cases, while 20% data for testing.
3. **Model tuning**

Model tuning involves optimizing a deep learning model to improve its performance in accurately classifying skin conditions from images.

4. **Build skin disease detection system**
   You are required to develop a simple application in streamlit, which will be used to predict skin disease(s).
5. **Model deployment in real time environments**
   Finally, the application needs to be deployed in streamlit.
6. **Doctor management from admin panel**
   The admin has the ability to manage the doctor information by adding, updating, or removing entries.
7. **User management system from admin panel.**
   The admin has the ability to manage the user's information by adding, updating, or removing user from the system.
8. **Feedback management system.**
   The admin can view the feedback submitted by the user from admin panel.

Tools:

Python, Keras, Pycaret, Colab, VS Code, Github and streamlit.

Features:

1. Exploring the dataset (Skin diseases image dataset).
2. Plotting Heatmap to see dependency of Dependent value on Independent features.
3. Predict skin disease classes.
4. Deploy model using streamlit.

Supervisor:

Name: Mehboob Ali
Email ID: mehboob.ali@vu.edu.pk
Skype ID: mehboobalivu@outlook.com

MCQ-Based Test System for Exam Preparation Using Python and Django

Project Domain / Category:
Data Science / Web Application Development

Introduction:

Students preparing for exams often struggle with finding effective ways to test their knowledge and gauge their preparedness. Traditional study methods, such as reading notes and textbooks, may not provide an interactive or personalized learning experience. Similarly, teachers require a platform to easily create tests tailored to their students' learning needs. The need arises for a web-based system that offers an adaptive, multiple-choice questions (MCQ)-based test platform where teachers can create and manage tests, and students can attempt them to prepare for exams. Where the report of all tests will be maintained in the system for analysis of individual and overall performance of the students.

In this project, an MCQ-based exam preparation system will be developed using Python and Django. Teachers will be able to create and customize tests by selecting from a pool of questions or generating new ones. Students can take tests, review their answers, and track their performance over time. Additionally, the system will feature automatic MCQ generation from input text, making it easier for teachers to design tests from any reading material. This platform will provide a real-time learning environment that simulates actual exam conditions and generates personalized tests for students based on their performance.

Functional Requirements:

The system will be divided into several modules to achieve efficient functionality and performance:

a) Define the Problem and Select Dataset(s):

Identify key subjects and topics to be covered (e.g., mathematics, science, computer science).
Select or create datasets containing multiple-choice questions, answers, and explanations.
b) Data Analysis and Preprocessing:

Analyze the dataset of MCQs to ensure a wide variety of topics and difficulty levels.
Preprocess data by categorizing questions based on topics, difficulty, and question type.
c) Feature Extraction:

Extract features such as question difficulty, correct answer rate, and student performance trends.
d) User Model Development:

Create separate user roles for teachers and students.
Track student progress, question attempts, and scores.
e) Test Creation by Teachers:

Teachers will create tests by selecting questions from the question bank or generating new ones using an automatic MCQ generator (based on input text).
Teachers can set the number of questions, difficulty level, and time limits for each test.

f) Personalized Tests for Students:

Students can take tests assigned by teachers, and the system will provide personalized recommendations based on past performance.
Tests can be generated dynamically by teachers, and students will attempt them with real-time feedback.
g) Test Evaluation and Feedback:

Automatically evaluate students' answers and provide instant feedback.
Teachers can review student performance and provide further feedback or suggestions.
h) Build System:

Develop a web-based interface using Django that supports both teacher and student functionalities.
Ensure the system is responsive and user-friendly.
i) Test System:

Perform thorough testing to ensure accurate question generation, answer evaluation, and system reliability.
j) Automatic MCQ Generation from Text:

Teachers will input a paragraph or chapter from any text.
The system will process the text using Natural Language Processing (NLP) techniques, identify key concepts, and generate relevant MCQs.
Teachers will have the option to review and edit the generated MCQs.

Actors in the System:
Teacher(Admin):

Prepares tests by selecting or creating MCQs.
Manages question bank by adding, editing, or deleting questions.
Generates MCQs from input text (e.g., paragraphs, chapters).
Reviews student performance and provides feedback.
Student(End User):

Attempts tests created by teachers.
Receives real-time feedback after submitting answers.
Reviews correct answers and explanations.
Tracks performance history and progress.

Expected System Modules:

1. **User Registration and Profile Management (Teacher and Student):**

Teachers and students will register and create profiles.
Teachers will have access to test creation and management features, while students with limited access can take tests and view performance.

2. **Question Bank Management (Teacher):**

Teachers can create, edit, or delete questions in a central question bank. Questions will be categorized by subjects, difficulty levels, and topics.

3. **Test Creation and Customization (Teacher):**

Teachers can create personalized tests by selecting questions from the question bank or using the automatic MCQ generation feature. Tests can be customized by the number of questions, difficulty, and time limits.

4. **Test Execution and Real-Time Evaluation (Student):**

Students will attempt tests created by teachers, and the system will automatically evaluate their answers. Real-time feedback will be provided, including correct answers.

5. **Performance Tracking and Analytics (Student and Teacher):**

Students will track their test scores, performance over time, and areas for improvement.

Teachers can review student performance, analyze progress, and provide targeted feedback.

6. **Automatic MCQ Generation from Text (Teacher):**

Teachers can input text (e.g., a paragraph or chapter), and the system will automatically generate MCQs using NLP algorithms. Teachers will have the option to review and modify these questions before finalizing the test.

Dataset:

The dataset will consist of multiple-choice questions and answers categorized by subject and difficulty level. Questions may be sourced from open educational resources or generated based on academic syllabi.

Example datasets:

https://www.kaggle.com/datasets/thedevastator/medmcqa-medical-mcq-dataset

For automatic MCQ generation, the system will rely on textual data input by teachers or publicly available academic resources.

Tools and Technologies:

Python (Django): Backend web framework for handling user interactions and test management.

Jupyter Notebook: For initial data analysis and model development.

PostgreSQL / MySQL: Database to store MCQs, user data, and test history.

NLTK / Spacy: For Natural Language Processing to generate MCQs from text input.

Bootstrap / ReactJS: For building the frontend user interface.

Matplotlib / Plotly: To generate graphs and visualizations for performance tracking.

Supervisor:

Name: Muhammad Kaleemullah

Email ID: m.kaleem@vu.edu.pk

Skype ID: kaleembhatti561

Prodigenius – AI Powered Task Management Application

Project Domain / Category

Mobile App + Machine Learning

Abstract / Introduction

Prodigenius is a streamlined, AI-powered (Artificial intelligence based) task management app built with Flutter, designed to help users prioritize tasks, manage schedules, and track productivity. The app uses free AI libraries and pre-built models for basic functionality, making it ideal to develop without complex custom AI models or cloud costs.

Functional Requirements:

The functional requirements of this project are given below:

1. **Task Input and Simple Categorization:**
   Manual Input: Users can manually create and categorize tasks (e.g., work, study, personal) and assign due dates.
   Basic Prioritization Input: Users can set task priorities manually or let the app suggest priorities using AI-based logic e.g. according to their execution time or urgency.
2. **Basic AI-Powered Task Prioritization:**
   Library Used: Firebase ML Kit (On-Device)
   AI Work: The app uses Firebase ML Kit's basic logic to prioritize tasks based on due dates and urgency. It automatically adjusts task priority to suggest which tasks need attention first, especially as deadlines approach.
   How It Works: The AI evaluates tasks using a few rules, such as how close the deadline is and how important the task is, and prioritizes tasks accordingly.
3. **Simple Task Duration Estimation:**
   Library Used: TensorFlow Lite (Pre-trained Model)
   AI Work: TensorFlow Lite's pre-built model predicts how long tasks will take based on task type. The model estimates time by considering task categories (e.g., work tasks generally take longer than personal tasks).
   How It Works: Users can view estimated task durations, which the app calculates based on pre-trained AI models that recognize task types.
4. **Basic AI-Driven Task Scheduling:**
   Library Used: Firebase ML Kit (On-Device)
   AI Work: Firebase ML Kit helps schedule tasks based on available time and task priorities. It recommends when to complete tasks by analyzing task importance and user availability.
   How It Works: The AI suggests task schedules, such as doing quick tasks first or focusing on high-priority tasks earlier in the day.
5. **Simple Reminders & Notifications:**
   Library Used: Flutter Local Notifications
   AI Work: Smart notifications remind users about high-priority tasks or those approaching deadlines. Simple logic, not advanced AI, is used here to trigger reminders.
   How It Works: Notifications alert users when tasks are nearing deadlines or when it's time to focus on important tasks.
6. **Basic Productivity Tracking:**

Library Used: Firebase ML Kit (On-Device)
AI Work: The app tracks users' task completion rates and generates simple insights, such as the number of tasks completed daily or weekly. It offers basic AI-generated insights like: "You are most productive on Wednesdays."
How It Works: Firebase ML Kit identifies user patterns and provides simple productivity reports based on task completion trends.

7. **Progress Visualization & Task Dashboard:**
   Library Used: Flutter Charts (No AI needed)
   No AI Work Here: Use basic charts and visualizations to show tasks completed, pending, and overall productivity in a clean UI dashboard.

Tech Stack:

1. Flutter: Cross-platform development framework for Android and iOS.
2. Firebase (On-Device ML Kit): Provides on-device machine learning for AI-based prioritization, task scheduling, and productivity insights.
3. TensorFlow Lite: For running pre-trained AI models on the device (e.g., task duration estimation).
4. tflite_flutter Plugin: Helps integrate TensorFlow Lite models with the Flutter app.
5. Flutter Local Notifications: Provides local notifications and reminders for task deadlines and priority alerts.

Tools:

1. **Flutter SDK** – For app development.
   https://flutter.dev/docs/get-started/install
1. **Android Studio / Visual StudioCode** – IDEs for coding.
   Android Studio: https://developer.android.com/studio
   Visual Studio Code: https://code.visualstudio.com/
2. **Firebase Tools** – For authentication, on-device ML Kit, and notifications.
   https://firebase.google.com/products/ml-kit
3. **TensorFlow Lite Model Maker (Optional)** – For customizing AI models.
   https://www.tensorflow.org/lite/guide/model_maker
4. **tflite_flutter Plugin** – For TensorFlow Lite integration in Flutter.
   https://pub.dev/packages/tflite_flutter
5. **Flutter Local Notifications Plugin** – For sending task reminders and notifications.
   https://pub.dev/packages/flutter_local_notifications
6. **Dart DevTools** – For debugging and performance analysis.
   https://dart.dev/tools/dart-devtools
7. **Git** – For version control and collaboration.
   https://git-scm.com/
8. **Postman (Optional)** – For testing APIs.
   https://www.postman.com/downloads/

Supervisor:

Name: Muhammad Bilal
Email ID: bilal.saleem@vu.edu.pk
Skype ID: bilalsaleem101

Real-Time Deepfake Detection for Video Streaming Platforms

Project Domain / Category

AI Web Application / Machine Learning

Abstract / Introduction

With the rise of deepfake technology, it has become increasingly easy to manipulate videos, posing risks to media authenticity, personal privacy, and public trust. In live streaming and video conferencing platforms, deepfake videos can be used for impersonation and misinformation, creating significant challenges for ensuring authenticity.

The objective of this project is to develop a real-time deepfake detection system that can analyze video streams and identify manipulated content. The application will detect deepfakes by analyzing each video frame using advanced machine learning models. Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks will be employed to detect inconsistencies across frames, ensuring the detection is fast and reliable.

The system will be designed for use on streaming platforms, helping institutions, organizations, and individuals identify manipulated content during live sessions. This project will allow students to explore the latest deep learning techniques in multimedia security while focusing on a practical and impactful solution.

Functional Requirements:

1. **User Authentication:**
   - Users will register and log in to access the detection system.
   - Admin-level authentication for platform owners.

2. **Real-Time Video Input:**
   - The system will support live video streaming or video conferencing platforms (e.g., Zoom or WebRTC-based platforms).
   - Users can also upload recorded videos for analysis.

3. **Deepfake Detection:**
   - Use CNN-LSTM-based models to process video frames in real-time and detect deepfake content.
   - Incorporate facial landmarks analysis and detect inconsistencies in eye movement, lip-syncing, or lighting patterns.

4. **Visualization of Results:**
   - The system will highlight detected frames with a confidence score indicating the likelihood of the video being a deepfake.
   - Alerts will be provided during live streams when a deepfake is detected.

5. **User Feedback:**

   Users can provide feedback on detection accuracy to improve future models.

6. **Error Handling:**

   Proper messages will be shown in case of processing delays, network issues, or unsupported video formats.

7. **Privacy and Data Handling:**

   - The system will process video data locally or ensure end-to-end encryption to maintain user privacy.
   - No video data will be stored without user consent.

Tools:

Development Environments / IDEs:

- **Backend Development:** Python with Flask or Django.
- **Frontend Development:** HTML, CSS, and JavaScript (or React.js for a dynamic interface).

Libraries and Frameworks:

1. **Deep Learning Models:** TensorFlow or PyTorch for building and training CNN-LSTM-based models.
2. **Video Processing:** OpenCV for extracting and processing video frames.
3. **Facial Detection:** Dlib or MTCNN for detecting facial landmarks.
4. **Model Deployment:** ONNX or TensorFlow Lite for deploying models on the web for real-time performance.
5. **Security Tools:** JWT or OAuth for user authentication.

Hardware Requirements:

- **GPU Support:** A system with GPU acceleration is recommended for real-time video processing or can use cloud based GPUs (e.g, Google Colab, Google Cloud Platform).
- **Web Server:** Use cloud-based servers (e.g., AWS, Azure) to deploy the application.

Supervisor:

Name: Sonia Salman
Email ID: sonia.salman@vu.edu.pk
Skype ID: sonia_salman

Implement a system of Heart Disease Detection Using Machine Learning

Project Domain

Machine Learning based project

Introduction

The scope of the project is to implement a machine learning based system that detect and diagnose the heart disease. Cardiovascular diseases are one of the major causes of mortality throughout the world. Early detection is very important for effective and intime treatment can save many lives. Dataset is first step to obtain in order to train a selected model. The system will use patient date, like medical history, diagnostic test and their results, life style factors in order to predict the likelihood of heart disease. Machine learning algorithms can analyse complex patterns in the data that may not be apparent through traditional diagnostic methods. The main objective of the system is to design, implement and evaluate different machine learning model that can predict the heart disease more accurately in patient. The system will help to identify the most relevant features that hep to predict heart disease. It also compares the performance of different machine learning algorithms. This comparison will be based on result accuracy, precision, recall and F1 score. For interact with system, create a user-friendly interface in order to get input of patient data and receive predictions plus results of models.

Steps include collection of data set, mostly features used for heart disease data set are age, gender, blood pressure, cholesterol level, fasting blood sugar etc. Next steps include data pre-processing, splitting of data and algorithm selection. Several machine learning algorithms can be applied to predict heart disease: Logistic Regression, Decision Trees, Random Forests, Support Vector Machines (SVM), K-Nearest Neighbours (KNN), Neural Networks/Deep Learning, Gradient Boosting Algorithms. The selected algorithm is trained on the training dataset. During training, the algorithm learns the patterns in the data, linking inputs (e.g., cholesterol levels, age) to outputs (heart disease or not). The last step is to evaluate the model on the test set using matrices. You have to create a web based user-friendly interface that allows healthcare professionals to input patient data and receive predictive results.

Functional requirement

Data set collection

1. The system shall allow user to input patient data including demographic, clinical, and lifestyle information.
2. The system shall accept various data inputs such as:
    - Age
    - Gender
    - Blood pressure
    - Cholesterol levels
    - Blood sugar levels
    - Smoking habits
    - Chest pain type
    - Maximum heart rate achieved
    - ST depression values
3. The system shall import the data of patient in batch format like in Excel file.
4. The system shall validate data to ensure complete record has been upload and only valid data will proceed.

Data Preprocessing

5. The system shall handle missing data using imputation techniques

6. The system shall normalize or scale numerical data where necessary to improve model accuracy.
7. The system shall able to remove redundant and irrelevant data/features

Machine Learning Model Module

8. The system shall provide multiple machine learning algorithms for heart disease prediction, including:
   - Logistic Regression
   - Decision Trees
   - Random Forest
   - Support Vector Machines (SVM)
   - Gradient Boosting
   - Neural Networks
9. The system shall allow user to train model using different algorithms
10. The system shall compare the performance of different algorithms
11. The system shall enable cross-validation to ensure model generalizability and avoid overfitting.
12. The system shall store trained models so they can be reused without needing to retrain them for each prediction.

Prediction

13. The system shall predict the likelihood of heart disease based on the trained machine learning model and patient data.
14. The system shall provide predictions as a probability score (e.g., 0-100%) indicating the risk of heart disease.
15. The system shall display prediction results in a user-friendly manner, along with a risk level classification

Tools: For Model Training Python, with libraries like scikit-learn, For Web Development/Interface, PHP JavaScript

Supervisor:
Name: Anam Naveed
Email ID: anam.naveed@vu.edu.pk
Skype ID: live:anam13dec

Stationery Shop Management System

Project Domain / Category

Desktop Application

Abstract/Introduction

Stationery Shop management system is specially designed for the purpose of adding Stationery item's detail. The system elaborates the basic concept for storing and generating Stationery item's detail. These items will be distributed in different categories e.g. Office products, Notepads, Ballpoints, Pencils etc. So we can search it easily under the appropriate label and inform the customer about their price, and other Specification, if needed. In this system, staff can sign up as a system admin, He/she can have full access to the system for maintaining daily records.

It will be a windows-based application which has been developed to make all the operations fast and easy.

The design of Stationery Shop Management system is easy to use for every type of users because a lot of shops have salespersons which are not very qualified. In this application we also add images in front of each Item for identification purpose. Whereas, in our application there is a facility of report generation which gives detail information about Stationery Shop item's sell and purchase and make staff possible to get hard copy of related reports.

Functional Requirements:

1. Registration
    - The administrator can create a new user.
    - New user can login and logout.
2. Stationery Item's
    - Admin add, delete or modify the details of the Item's.
3. Search
    - Admin or System User can search for the required Stationery Shop Item's based on name, id etc.
4. Sell Stationery Item's
    - The sold Stationery Shop Item's need to be deducted from the available stock.
5. Purchase Stationery Item's
    - The purchased Stationery Shop Item's need to be added to the available stock.
6. Stock
    - The stock will be updated after purchasing Stationery Shop Item's.
7. Report generation
    - Depending upon the Stationery Shop needs following reports can be generated
    - There can be daily reports
    - Weekly reports
    - Yearly reports

These reports will be of total available stock, how many Stationery Shop Items are sold out / purchase on daily, monthly and yearly basis.
The system should be able to **generate** a report of profit on daily, monthly and yearly basis.

Hint: Use the **calendar python** module, it is an in-built module in Python that handles operations related to calendars. The output of the module is displayed as a calendar. By default, the first day of the week for the Gregorian calendars is Monday, and the last day is Sunday.

Tools:

Python is mandatory.

We will use **Tkinter (Tkinter** is one of the most popular programming frameworks for Desktop apps and GUIs. It is a combination of the Tk and Python GUI frameworks**),** to render our application's menu and its buttons, as well as Tkinter is a lightweight module and can be used to create cross-platform applications.

Note:

This Application will be fully automatic, Students are required to enter the required data once which will then be shared between all components of the application automatically. The data flow must be achieved by the students through automation rather then repeated data entry at each form. It is the responsibility of your application that the required data will be available for each form when it is once entered.

There is some functionalities like Profit Calculation where we required different prices so it will be provided automatically by the application, cannot be entered by Admin or User. So the profit will be calculated automatically because all fields required data will be provided automatically. For this purpose you need to mention purchase price and sale price in advance which will be used for profit calculation.

Templates (Flask template or any other template and Drag and Drops) are not allowed to use in this application you need to use python libraries.

For this project you need to visit any Stationery Shop and get more functional and non-functional requirements from the Owner and if you have any ambiguity contact at my Skype id mentioned below.

Supervisor:
Name: Asadullah
Email ID: asad.ullah@vu.edu.pk
Skype ID: asad.ullah121

IoT-Based Smart Aquarium Management System

Project Domain / Category

Digital Logic Design & IoT

Abstract / Introduction

The Internet of Things (IoT) is reshaping industries by connecting devices to the internet and enabling automated control and monitoring. In this project, we aim to develop an IoT-based Smart Aquarium Management System that provides real-time monitoring and control of essential aquarium parameters. This system will offer a seamless, automated solution for maintaining the health and well-being of aquatic life by monitoring water temperature, pH levels, light intensity, and oxygen levels, while automating feeding schedules.

This system will provide users with a visual display and indication mechanism at the aquarium, as well as remote monitoring via a web or mobile application. The ultimate goal is to enhance aquarium management by automating the routine tasks, ensuring a stable environment for the aquatic ecosystem, and notifying the user of any deviations from optimal conditions.

Functional and Non-Functional Requirements

Functional Requirements

1. **Real-Time Monitoring:**
    a. The system must monitor essential water parameters such as temperature, pH level, water level, dissolved oxygen, and lighting conditions in real-time.
2. **Automated Control:**
    a. The system must automate feeding and control the lighting based on pre-set schedules or environmental conditions.
    b. Temperature regulation must be automated with heaters or coolers to maintain optimal levels.
3. **Notification Alerts:**
    a. The system should notify the user through a mobile or web application in case of any abnormalities (e.g., high/low temperature, pH imbalance, or water level issues).
    b. Alerts should be visual and audible on the local display.
4. **Display and Indication:**
    a. A local LCD screen must display real-time water parameters and system status.
    b. Visual indicators such as LED lights should signal abnormal conditions (e.g., a red LED for high temperature).
5. **Remote Control and Monitoring:**
    a. Users must be able to access real-time data and control the system remotely via a mobile app or web interface.
    b. The app should allow users to adjust feeding schedules, lighting, and temperature thresholds.
6. **Data Logging:**
    a. The system must log water parameters over time and display historical data through the user interface for analysis of aquarium conditions.

Non-Functional Requirements

1. **Reliability:**
    a. The system must operate continuously, with minimal downtime, to ensure that aquarium conditions are constantly monitored and maintained.

2. **Scalability:**
    a. The system should be scalable to manage aquariums of different sizes, with the ability to add more sensors or functionalities if needed.
3. **Usability:**
    a. The mobile and web interfaces should be user-friendly, providing easy access to aquarium conditions and control functions.
    b. The local display should be intuitive, allowing even novice users to understand system status at a glance.
4. **Security:**
    a. The system must ensure secure access to the mobile or web application, requiring user authentication to prevent unauthorized control of the aquarium.
5. **Energy Efficiency:**
    a. The system should be designed to minimize power consumption, especially for components like sensors, displays, and actuators.

Tools:

1. **Hardware**:
    - **Microcontroller**: An Arduino or ESP8266/ESP32 microcontroller will be used to control the system and connect to sensors.
    - **Sensors**:
        - Temperature sensor (e.g., DS18B20 or DHT11) to monitor water temperature.
        - pH sensor to measure the acidity/alkalinity of the water.
        - Water level sensor to ensure optimal water levels.
        - Dissolved oxygen sensor to monitor oxygen concentration.
        - Light sensor to monitor the intensity of lighting.
    - **Actuators**:
        - Automatic fish feeder to manage feeding schedules.
        - Relays to control heating/cooling and lighting systems.
    - **Display**:
        - LCD display for real-time local status and water parameter display.
        - LEDs for indication of critical conditions.
2. **Software**:
    - **Programming Languages**: C/C++ for microcontroller programming; Python/JavaScript for web/mobile app development.
    - **Mobile App/Web Interface**: Developed using frameworks like React Native (for mobile apps) or React.js/Flask for web apps.
    - **Cloud Services**: Firebase or AWS IoT for real-time data storage and remote access.
    - **Communication Protocols**: MQTT or HTTP for communication between the microcontroller and cloud services.
3. **Development Tools**:
    - **Arduino IDE**: For writing and uploading code to the microcontroller.
    - **Visual Studio Code**: For software development and debugging.
    - **Circuit Simulation Tools**: Tools like Proteus or Tinkercad for simulating the circuit before implementation.
    - **Version Control**: Git for source code management.

This IoT-based Smart Aquarium Management System aims to provide an efficient and intelligent way of managing an aquarium while ensuring that the aquatic life thrives in optimal

conditions. The system will greatly reduce manual intervention, improve user experience, and contribute to a more sustainable approach to aquarium management.

Supervisor:
Name: Waqar Ahmad
Email ID: waqar.ahmad@vu.edu.pk
Skype ID: engr.waqar.ahmad

3D MRI Brain Tumor Segmentation

Project Domain / Category

Artificial Intelligence/Image Processing

Abstract / Introduction

Image processing is a powerful field of study that has numerous real-world applications. It involves the manipulation of digital images to extract useful information or enhance specific features. One of the critical applications of image processing is medical image analysis, where it plays a pivotal role in diagnosis, treatment, and research. In this project, we aim to explore the fascinating realm of image processing and its applications in the healthcare domain, specifically in the context of brain tumor segmentation. This project centers on the realm of medical imaging, specifically the challenging task of brain tumor segmentation using deep learning techniques. Brain tumors are abnormal growths of tissue within the brain that can be cancerous or non-cancerous. Accurate and early diagnosis is essential for timely treatment and improved patient outcomes.

Functional Requirements:

In this project, we will leverage image processing and deep learning techniques to address the following objectives:

1. **Data Collection and Preprocessing:** Gather a diverse dataset of brain MRI scans, ensuring data quality and integrity. Preprocess the images to enhance their suitability for further analysis. Import the image dataset of Brain MRI scans from described link.
2. **Dataset Splitting:** To facilitate model training and evaluation, we will split the dataset into distinct sets for training, validation, and testing. This step ensures that the deep learning model's performance is rigorously assessed and prevents overfitting.
3. **Deep Learning Model Development:** We will design, implement, and fine-tune deep learning models, such Capsule Network, to accurately segment brain tumors from MRI images.
4. **Model Evaluation:** We will establish robust evaluation metrics(dice score and Hausdorff distance) to assess the performance of the deep learning models, ensuring that the segmentation results are precise and reliable.
5. **DataSet:** For this project you need to use BraTs2019 Dataset. https://www.med.upenn.edu/cbica/brats2019/data.html

Important links and Tutorials:

- **Python**
  - https://www.w3schools.com/python/
  - https://www.tutorialspoint.com/python/index.htm
- **Image processing**
  - https://regenerativetoday.com/some-basic-image-preprocessing-operations-for-
  - beginners-in-python/
  - https://www.section.io/engineering-education/image-preprocessing-in-python/
  - https://www.tensorflow.org/tutorials/load_data/images

- **Deep Learning**
    - https://www.simplilearn.com/tutorials/deep-learning-tutorial/guide-to-building-powerful-keras-image-classification-models
    - https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/

Tools:

Language: Python (Only python language)
Framework: Anaconda
IDE: JupyterNotebook, Pycharm, Spyder, Visual Studio Code, etc.
You can also use Google colab environment or google cloud.

Note: VU will not provide you any resources to buy any Software, Framework or any services if used in this project. Student has to manage all the project on its own.

Kindly read the proposal carefully and decide if you have completely understood the project requirements before selecting the project. Please feel free to discuss any project- related questions before selecting it

Supervisor:

Name: Muhammad Hashir Khan
Email ID: hashir.khan@vu.edu.pk
Skype ID: hashir.khan9996

Advanced Face Recognition System with Real-Time Detection

Project Domain / Category

Image Processing

Abstract

This project aims to develop a comprehensive face recognition system using machine learning that not only detects and recognizes faces but also saves captured face images in a database for future reference. Additionally, the system will feature a user interface to manage the database, allowing users to upload new images, view stored data, and manage the system. The core of the system will rely on deep learning techniques for face recognition and database integration to store and retrieve face data efficiently.'

Functional Requirements:

1. Collect and Prepare Data: Collect face images either from public datasets (e.g., LFW, CelebA) or by capturing faces via webcam. Preprocess the images by resizing, aligning, and normalizing them.

2. Develop the Face Recognition System: Implement face detection using OpenCV or Dlib. Use pre-trained models like FaceNet to extract face embeddings from detected faces. Train or fine-tune machine learning models (e.g., KNN, SVM) using the extracted embeddings for recognition tasks.

3. Database Integration: Create a database schema in SQLite or MySQL to store: Face images, Face embeddings, Metadata (e.g., name, ID). Implement functions to store, retrieve, update, and delete records from the database.

4. Build the User Interface: For Desktop (Tkinter): Create an interface where users can view all stored face images along with names and IDs. Implement buttons to upload new face images and save them to the database. Add features to edit or delete face entries. For Web (Flask/Django): Develop a web-based interface to manage the face database, allowing remote access.

5. Real-Time Face Recognition: Capture live image streams using a webcam. Detect and recognize faces from the image feed by comparing the extracted embeddings with the stored database. Display the recognized name or ID on the imagw. Allow users to search for specific individuals in the database by name or ID and display their corresponding face images and embeddings. Enable users to modify or delete stored face entries (e.g., change name, remove a face).

The enhanced face recognition system will not only detect and recognize faces but will also allow for efficient storage and management of face data in a database. The added interface will provide users with an easy-to-use platform to manage face images and metadata, making it suitable for real-world applications like security, attendance tracking, and access control systems.

Tools and Technologies Required

1. **Programming Language:** Python: The primary language for the implementation of machine learning algorithms and the database interface.

2. **Libraries:** OpenCV for real-time video processing, face detection, and webcam integration. Dlib for advanced face detection and feature extraction. TensorFlow / Keras for building deep learning models (CNN, FaceNet, VGGFace). Flask/Django for web-based UI (optional). Tkinter for desktop-based GUI. SQLite / MySQL for database management.

3. **Face Recognition Libraries:** FaceNet, VGGFace, or Dlib for extracting face embeddings. Scikit-learn for classifiers (e.g., KNN, SVM).

4. **Database:** SQLite or MySQL for managing face images, metadata, and embeddings.
5. **Webcam or External Camera:** For capturing real-time video streams and detecting faces.
6. **IDE**: PyCharm, VS Code, or Jupyter Notebook for coding and testing the system.

Face Recognition Dataset Links

1. [Labeled Faces in the Wild (LFW)](http://vis-www.cs.umass.edu/lfw/)
2. [CelebA Dataset](http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html)
3. [YouTube Faces Database](https://www.cs.tau.ac.il/~wolf/ytfaces/)
4. [VGGFace2 Dataset](http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/)
5. [WIDER Face Dataset](http://shuoyang1213.me/WIDERFACE/)

Supervisor:

Name: Madiha Faqir Hussain
Email ID: madiha.hussain@vu.edu.pk
Skype ID: madiha.akhtar74

Automated Ripeness Detection of Fruits using Deep Learning

Project Domain / Category

Deep Learning / Computer Vision

Abstract / Introduction

The goal of this project is to create an automated system that uses deep learning and computer vision to accurately detect the ripeness level of various fruits. Correctly identifying the ripeness of fruits is crucial for both consumers and suppliers to ensure the best quality and reduce food waste.

This project will utilize a dataset of fruits images labeled by ripeness stage (e.g., underripe, ripe, and overripe) to train a Convolutional Neural Network (CNN) for classification.

A ripeness detection system will assist grocery stores, suppliers, and end consumers by providing accurate ripeness classifications, ensuring that the fruits sold are fresh and of the best quality.

Datasets for reference:

https://www.kaggle.com/datasets/sumn2u/riped-and-unriped-tomato-dataset?resource=download

https://universe.roboflow.com/fruit-ripening/banana-ripening-process

https://www.kaggle.com/datasets/cienciacafeto/coffee-fruit-maturity

Functional Requirements:

Admin (Student) will perform all these (Functional Requirements) tasks.

- Data Collection:
  The dataset should include at least four different types of fruits. You may use publicly available datasets or create a custom dataset. Ensure that the chosen dataset is suitable for ripeness detection.
- Image preprocessing:
  The system must preprocess input images by cropping, resizing, and normalizing them for optimal compatibility with the deep learning model.
- Ripeness Detection:
  Implement a CNN model to identify and classify fruits based on their ripeness (underripe, ripe, overripe).

- User interface:
  Provide a user-friendly interface, either desktop or mobile-based, that allows users to upload images and view the ripeness classification results (underripe, ripe, or overripe).
- Performance evaluation:
  The project will assess the system's performance using accuracy, precision, recall, and F1-score metrics to ensure the model's effectiveness.

Tools:

- Python programming language
- TensorFlow or PyTorch for deep learning
- OpenCV for image processing
- Tkinter or PyQt for desktop application / Android Application

Supervisor:

Name: Zaid Ismail
Email ID: zaid.ismail@vu.edu.pk
Skype ID: m.zaid_1994_1

EyeGuard: Eye Strain Detection

Project Domain / Category

Image Processing/Health and Wellness

Abstract / Introduction

"EyeGuard" is a mobile app designed to help users detect eye strain, especially after long periods of screen usage. Prolonged screen time can lead to eye fatigue, which can affect vision and productivity. The app allows users to take a selfie, and it analyzes their face for signs of eye strain, such as redness or droopy eyes. Based on the analysis, the app will suggest whether to take a break or continue working. This app promotes better eye health by reminding users to rest their eyes and prevent eye strain.

Functional Requirements:

1. User Registration and Login
2. Capture a Selfie
3. Analyze the image for Signs of Eye Strain (e.g., Redness, Droopy Eyed)
4. Display Result: "Take a Break" or "Continue Working"
5. Save History of Previous Eye Strain Checks
6. User-Friendly Interface for Simple Navigation

Tools:

1. **Development Environment:** Android Studio (for Android)
2. **Frontend:** XML and Java (Android Studio)
3. **Image Processing Library:** SimpleCV or basis Python scripts for easy image analysis (if using Android Studio)

Supervisor:

Name: Dr. Sana Rao
Email ID: sana.rao@vu.edu.pk
Skype ID: rao.sana10

Face Mask Classifier

Project Domain / Category

Image Processing/Deep Learning

Abstract / Introduction

With the ongoing concerns regarding public health and safety, particularly in the wake of the COVID-19 pandemic, ensuring compliance with mask-wearing protocols has become crucial in public and private spaces. Effective monitoring of these protocols, however, presents a significant challenge, especially in large, crowded environments. An automated solution can bridge this gap by efficiently detecting and classifying individuals based on their adherence to mask-wearing guidelines. The **Face Mask Classifier** capable of identifying individuals who are either wearing a mask correctly, wearing it improperly, or not wearing a mask at all. By leveraging machine learning and computer vision technologies, this classifier will provide a scalable and efficient solution for organizations seeking to enforce mask-wearing policies. The system will promote public health measures with minimal human intervention.

Functional Requirements:

Admin (Student) will perform all the following tasks.

1. **Image Input:** Face Mask Classifier should support the input of images from given dataset. The system should be capable of handling various resolutions and image qualities to accommodate different imaging devices.
2. **Preprocessing:** In the Face Mask Classifier project, preprocessing is essential for transforming raw input data into a suitable format for model training and inference. Preprocessing techniques should include Image Resizing, Normalization, and Handling Class Imbalance etc.
3. **Data Augmentation:** To improve the robustness and generalization of the classifier, data augmentation techniques should be employed. These may include random rotations, flips, shifts, brightness adjustments, and zooming. By artificially increasing the diversity of the training dataset, the model will be better equipped to handle various real-world scenarios and face orientations.
   Apply the Data Augmentation and increase the dataset by 3 folds.
4. **Model Selection and Development:** Investigate different deep learning architectures and select an appropriate architecture (e.g., U-Net, 3D Convolutional Neural Networks (CNNs), V-Net, DeepMedic, Residual Networks (ResNet), DenseNet, Attention Mechanisms, YOLO) for development.
5. **User Interface:** Provide a user-friendly GUI for:
   - Uploading images.
   - Displaying results with visual indications.
   - Accessing historical data and logs.
6. **Mask Classification:** The Face Mask Classifier project categorizes individuals based on their mask-wearing status to promote public health and safety. The three classification categories are:
   - **With Mask:** Identifies individuals wearing masks properly, covering both the nose and mouth, essential for compliance with health protocols.
   - **Without Mask:** Includes individuals not wearing any mask, crucial for addressing non-compliance and encouraging mask usage in public spaces.
   - **Improperly Worn Mask:** Captures individuals wearing masks incorrectly (e.g., below the nose or chin), enabling targeted education on proper mask usage.

7. **Train & Test Data:** Split dataset into 70% training and 30% testing dataset and train the model accordingly.
8. **Logging and Reporting**
    - Log all classification events with timestamps and input sources.
    - Generate reports summarizing mask compliance statistics.
9. **Evaluation and Fine-tuning:** Assess the model's performance using standard evaluation metrics (e.g. Accuracy, F1-score, precision, recall, Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC)) and fine-tune the model for improved accuracy.
10. **Model Updates and Retraining:**
    - Allow users to retrain the model with new data.
    - Deliver with a pre-trained model for immediate use.

Dataset:
https://drive.google.com/drive/folders/1jfwHQN7WXv0mLFXjzZgbgjvDURk1wJ5m?usp=sharing **You must use your VU email id to access/download the dataset.*

Tools:
- Python, jupyter notebook, Colab, PyQt, wxPython, Tkinter, Kivy, PySimpleGUI

Prerequisite:
Artificial Intelligence, Machine Learning, and Image Processing, Computer Vision concepts, "Admin (student) will cover a short course relevant to the mentioned concepts besides SRS and Design initial documentation."

Helping Material
Python
https://www.python.org/
https://www.w3schools.com/python/
https://www.tutorialspoint.com/python/index.htm

Deep Learning:
https://www.tutorialspoint.com/python_deep_learning/index.htm
https://www.tutorialspoint.com/deep-learning-tutorials/index.asp
Deep Learning Crash Course for Beginners (youtube.com)
Deep Learning | What is Deep Learning? | Deep Learning Tutorial For Beginners | 2023 | Simplilearn (youtube.com)

Image Processing:
Python tutorials for image processing and machine learning - YouTube

Supervisor:
Name: Umair Ali
Email ID: umairali@vu.edu.pk
Skype ID: live:umairalihamid_1

Food Image Classification

Project Domain / Category

Machine Learning/Image Processing

Abstract / Introduction

Food image classification is an emerging area in the era of artificial intelligence and computer vision. With the rapid growth of digital platforms in sectors like food delivery, dietary monitoring, and nutrition tracking, automating the process of recognizing food items from images has gained significant relevance. The Food Image Classification Project aims to develop a machine learning model capable of accurately identifying various food items from images and categorizing them into predefined classes. The system will leverage deep learning techniques to analyze visual patterns and textures unique to each type of food. By training the model on a large dataset of food images, the system will learn to recognize common food categories such as "pizza," "burger," "pasta," and "salad," among others. This project addresses the growing demand for AI-powered solutions that can process visual information efficiently and offer practical applications in everyday life. The integration of such a system can significantly improve user experience in food-related applications by providing fast, accurate, and automated recognition of food items.

Objectives

Following are the objectives of the project are as follows:

- Develop a comprehensive dataset of food images with diverse categories.
- Implement preprocessing and augmentation techniques to enhance the quality of input data.
- Design and develop a machine learning model capable of accurately classifying food items from images.
- Evaluate the model's performance using standard metrics.
- Develop a user-friendly application interface for real-time food classification.
- Optimize and fine-tune the model to enhance accuracy and generalization.

Note: Dear Students, Kindly read the methodology carefully and analyze yourself before selecting the project.

Methodology

1. **Learn the Basics of Python and Machine Learning**

   The first step you need to dive in to this project is to get comfortable with python and machine learning concepts. You have to understand how to work with python as most of your code will be written in it. Then you have to learn libraries for data handling and visualization. Familiarize yourself with machine learning concepts like supervised learning, classification, and neural networks. I am suggesting some resources here.

   If you are unable to understand these concepts do not choose this project.

   Suggested Resources:

   Python:

   https://docs.python.org/3/tutorial/

   https://www.coursera.org/projects/image-processing-with-python

   Machine Learning:

   https://www.coursera.org/learn/machine-learning

2. **Setup your development environment:**
   You need a proper environment to code and run the experiment.
   - Download and install the python
   - Install a development environment
   - Install the necessary libraries that helps you with image processing, data handling, and model training.
3. **Data Collection:**
   - Generate a diverse collection of food images featuring a range of different categories.
   - Ensure the dataset includes sufficient samples to represent different foods.
   - Ensure that the dataset if structured properly with folders for each category.
     Note: This is your task to find a relevant and correct dataset. You can use the dataset available online. Your data set should comprise of more than 200 images.
     Hint: Food 101
4. **Load and Visualize the data:**
   Before working on the model, you need to understand the data you're working with. For this you have to load the images and then plot a few images from different categories to see what the data looks like.
5. **Data Pre-processing:**
   - In this step, you have to clean and prepare the data for analysis.
   - The images need to be resized and normalized for efficient processing by your model.
   - Your images should be of size 224x224 pixels and scale the pixel values to a range between 0 and 1.

6. **Split the Data into Training and Testing Sets**
   You need to divide the dataset into training and testing sets so you can evaluate the model's performance on unseen data.
   Steps:
   Training Set: 80% of the data, used to train the model.
   Testing Set: 20% of the data, used to evaluate the model.

7. **Build a Model:**
   You have to select the model which is best for image classification tasks.

8. **Model Training:**
   - Train the model using machine learning algorithm on your training dataset.
   - Fine-tune the model to improve accuracy and generalization.

   Note: You have the flexibility to use any algorithm for your system.

9. **Evaluation:**

   Once the model is trained, you need to evaluate its performance on the testing data.
   Plot a confusion matrix to see where the model is performing well and where it's making mistakes.

10. **User Interface (UI) Design**
    You can deploy the trained model as a web application where users can upload food images, and the app returns the predicted food category.
    Create a simple web app where users can upload an image.
    Load the saved model, process the uploaded image, and return the prediction.

Note: Kindly read the following guidelines before choosing the project.

1. Kindly read the proposal carefully and decide if you have completely understood the project requirements before selecting the project.
2. You have to implement the requirements mentioned in project proposal completely. You are not allowed to add irrelevant and un-necessary requirements.
3. You have to implement the project in mentioned tools and technology.
4. Kindly do not request to use php or html for image processing project.
5. Do not ask to share dataset because it is your task to find the appropriate dataset.
6. Student must have knowledge of image processing techniques.
7. Please feel free to discuss any project- related questions before selecting it.

Supervisor:
Name: Fizzah
Email ID: fizzah@vu.edu.pk
Skype ID: fizzahbhatti2020@outlook.com

Skin Cancer Detection

Project Domain / Category

Image Processing/AI/Web App

Abstract / Introduction

Skin cancer is one of the top causes of death globally, and melanoma is one of its most dangerous forms. It accounts for nearly 75% of skin cancer deaths and represents about 4% of all deaths worldwide. Early detection of melanoma can lead to successful treatment, otherwise it has the potential to spread to other parts of the body, making expert diagnosis essential. To diagnose melanoma, doctors first analyse dermoscopic images of the affected skin and then confirm their findings with an expert. However, even highly experienced doctors can struggle to differentiate between melanoma and normal skin parts because of their visual similarities. Unfortunately, the challenge of early detection remains due to the overlapping visual features of melanoma and non-melanoma conditions. The traditional methods of melanoma detection are highly time-consuming and error prone. This situation demands an automated computer aided diagnosis tools and techniques to detect and diagnose cancer at its early stages.

The project includes detecting skin cancer from dermoscopic images and performed binary classification into melanoma or non-melanoma class. You are required to develop a web app in which the user will enter images and check the status.

Functional Requirements:

- **Dataset Collection:** Collect image dataset from available free repositories or any other online source.
- **Pre-Processing**: Use different image processing techniques to create a uniform, normalized image dataset. You may need to perform data augmentation in this step.
- **Model Selection**: Analyze different deep learning-based CNN models and select a suitable one for classification.
- **Dataset Splitting:** Split the dataset into training and testing set for model evaluation.
- **Model Training:** Train the selected model using training dataset.
- **Validation and Hyperparameter Tuning:** Validate the model's performance using the validation set and fine-tune hyperparameters like learning rate, batch size, and network architecture to achieve the best results.
- **Model Evaluation:** Check the performance of the model used using testing dataset and different evaluation metrics.
- **Real-time Detection:** Implement a real-time skin cancer detection pipeline using OpenCV to upload an image from and apply the trained model for skin cancer detection.
- **User-Interface:** Develop a user-friendly interface in which the user can upload dermoscopic images for analysis. The interface should provide visual feedback, such as original images alongside classification results.

Prerequisites:

- Have a good understanding of Python.
- Having knowledge of basic deep learning concepts and models.
- Understanding of basic image processing techniques (preferable but not mandatory).
- Basic idea of working with image related datasets.

Tools:

- **Language:** Only Python
- **IDE:** JupyterNotebook, Pycharm, Spyder, Visual Studio Code, etc.
  Better to use Google colab environment or google cloud.
- OpenCV

Note:

- VU will not provide any kind of paid resources needed for the project.
- A student must find the dataset by himself / herself.
- Use of any other language is strictly prohibited.
- Kindly read the given instructions properly and choose a project only if you have developed a clear understanding of the project.
- A student who wished to select this project must commit to spend 2 hours daily for FYP project. This may include learning through tutorials or getting help from any reading material.
- In case of any query, feel free to contact and discuss with me.

Important links and Tutorials:

- **Python**
  - https://www.w3schools.com/python/
  - https://www.tutorialspoint.com/python/index.htm
  - https://www.programiz.com/python-programming
- **Deep Learning**
  - https://www.simplilearn.com/tutorials/deep-learning-tutorial/guide-to-building-powerful-keras-image-classification-models
  - https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/
- **Image Processing**
  - https://builtin.com/software-engineering-perspectives/image-processing-python
  - https://neptune.ai/blog/image-processing-python
  - https://www.geeksforgeeks.org/image-processing-in-python/
  - https://www.tensorflow.org/tutorials/load_data/images

Supervisor:

Name: Taliah Tajammal
Email ID: taliah.tajammal@vu.edu.pk
Skype ID: live:.cid.1d478ff6231e1aab

Virtual Proctor

Project Domain
Image Processing

Introduction
The project aims to develop a simple yet effective virtual proctoring system to monitor students during online exams by detecting faces and tracking activity. With the increasing reliance on remote learning and online assessments, ensuring exam integrity has become a significant challenge. This project introduces students to the basics of face detection and real-time video monitoring. The system will detect a student's face using the webcam, ensuring the student remains in front of the camera during the exam. If no face or multiple faces are detected, the system will alert the invigilator of potential cheating or irregular activity.

Functional Requirements:

1. Capture real-time video feed from the webcam.
2. Implement face detection using advanced methods (such as deep learning-based approaches like SSD Inception v2 or YOLOv8) to detect if a student's face is visible.
3. Detect multiple faces in the frame; raise an alert if more than one face is detected.
4. Monitor if the student leaves the frame or turns their head away for extended periods, indicating possible cheating.
5. Raise alerts (logs or messages) for unusual behavior such as no face detected, multiple faces, or the student moving out of frame.
6. Record suspicious activity with a logging system for later review.
7. Create a simple graphical interface for invigilators to monitor live video and see real-time alerts.
8. Document system design, face detection methods, experimental setup, and evaluation.

Tools & Technologies:

- **Languages**: Python, MATLAB
- **Python**:
  - **Libraries**: OpenCV (video capture, face detection), NumPy (image handling), Matplotlib (visualization), Dlib (optional for face detection and pose estimation)
  - **IDE**: Spyder
- **MATLAB**:
  - **Toolboxes**: Image Processing Toolbox, Computer Vision Toolbox, UI Components (for GUIs)
  - **IDE**: MATLAB built-in editor

Supervisor:
Name: Noor Rahman
Email ID: noor.rahman@vu.edu.pk
Skype ID: mahsud-cs619

AI-Based Smart Teaching Assistant for Personalized Exam Preparation

Project Domain / Category

Mobile Application / Artificial Intelligence (AI)

Abstract / Introduction

Preparing for exams is often challenging for students of Virtual University of Pakistan, as they need to sift through large amounts of study material and video lectures to organize their learning. This project proposes an AI-based smart teaching assistant that helps VU students prepare for exams by automating the process of creating concise notes from textbooks or handouts and evaluating their knowledge through quizzes. The system will allow users to upload complete study materials, and it will automatically summarize each chapter, generating key concepts and important topics. The platform will also create randomized MCQ-based tests to assess student understanding and preparation level, providing feedback to guide their studies. This personalized learning approach aims to improve student engagement and success rates.

Functional Requirements:

Following will be the functional requirements for the proposed project.

- **FR1:** Allow users (teachers or students) to upload textbooks, handouts, or other study materials in PDF or text format.
- **FR2:** Use AI and NLP algorithms to analyze the content and generate lecture-wise or chapter-wise summaries and key points.
- **FR3:** Automatically create short, concise notes for each lecture, chapter or section.
- **FR4:** Provide the ability to create virtual classes for different courses where students can access study materials and notes.
- **FR5:** Generate random MCQ-based quizzes from the study materials to evaluate student preparation.
- **FR6:** Provide a scoring system with immediate feedback for each quiz to help students identify their weak areas.
- **FR7:** Track student progress and allow teachers to view detailed reports on student performance.
- **FR8:** Implement adaptive learning features that adjust the difficulty of questions based on the student's performance.
- **FR9:** Ensure a user-friendly interface with accessibility features for students and teachers.

Additional Considerations:

- **Real-Time Summarization:** Use a lightweight AI model like TensorFlow Lite for summarizing text to ensure smooth performance on mobile devices.
- **Offline Capabilities:** Ensure that key features (e.g., accessing summaries, taking quizzes) work offline by storing data locally using SQLite.

- **Cloud Integration:** Use Firebase for storing larger data sets, managing user data, and synchronizing quiz performance across multiple devices.
- **Performance Optimization:** Since it's a mobile application, focus on optimizing the AI models for speed and low memory usage.

Tools:

- **Development Environment/IDEs:** Android Studio (Java/Kotlin), Firebase (for backend services)
- **AI and NLP Tools:** TensorFlow Lite (for on-device AI models), NLTK or Hugging Face Transformers for NLP, Tesseract OCR (for extracting text from images if needed)
- **Other Tools Required:** SQLite (for local database storage), Firebase (for user authentication, database, and cloud storage), Retrofit (for network operations), XML (for UI design)

Supervisor:

Name: Waqas Ahmad
Email ID: waqas.ahmad@vu.edu.pk
Skype ID: waqas_vu

Android Based Expense Management System

Project Domain / Category

Android Application

Abstracts/Introduction:

In today's fast-paced world, managing expenses across different domains like personal, corporate, and business expenses has become increasingly complex. People often struggle to keep track of their financial activities, leading to budgeting issues and financial stress. A centralized application that organizes and tracks these expenses by category would greatly simplify the process, allowing users to maintain control of their finances effortlessly.

Functional Requirements:

This project will focus on developing an Android application with the following features:

- ✓ Expense categorization and management.
- ✓ Ability to create custom categories.
- ✓ Interactive dashboard for visualizing expenses.
- ✓ Add and edit expense details.
- ✓ Track expenses over time.
- ✓ Generate monthly, weekly, and yearly reports.
- ✓ Provide insights into spending habits.

1. **User Registration and Authentication:**
    - The application should allow users to register and log in securely.
    - Social media login options (optional).
2. **Expense Category Management:**
    - Users should be able to choose from predefined categories (personal/home, corporate, shop/store, hospital expenses).
    - The option to manually add new categories should be provided if the required category does not exist.
3. **Expense Entry:**
    - The user should be able to add an expense by choosing a category.
    - Each expense entry should contain details like amount, date, description, and payment method (cash, credit card, etc.).
4. **Dashboard and Reports:**
    - The dashboard should provide a quick overview of expenses with filters for specific categories, dates, and time periods.
    - Graphical reports for monthly, weekly, and yearly expenses should be available.
    - Spending patterns should be displayed to give users insights into their financial habits.
5. **Search and Filter:**
    - Users should be able to search and filter expense records by categories, amounts, or date ranges.
6. **Notifications and Reminders:**

- The app should provide users with notifications about upcoming payments or alerts if they are approaching budget limits.

7. **Data Backup and Restore:**
    - Allow users to back up and restore their expense data to cloud storage or locally.
8. **Multi-Currency Support:**
    - The app should support multiple currencies to accommodate users across different regions.
9. **Expense Summary:**
    - Application should provide a summary of expenses in the form of charts or graphs to show expense distribution across categories.

Tools:

Java (for Android app development), relevant Database technology

Supervisor:

Name: Muhammad Tahir Jan
Email ID: tahir.jan@vu.edu.pk
Skype ID: mtahirjan@outlook.com

Campus Connect

Project Domain / Category

Mobile Apps.

Abstract/Introduction:

The purpose of this project is to provide undergraduate (bachelor's) students with the opportunity to develop a mobile application that enhances campus life and student engagement. "Campus Connect" will be a mobile app that allows students to access campus resources, events, and student services efficiently. This app will foster communication, help students stay informed about campus activities, and improve the overall student experience.

Functional Requirements:

1. **Event Calendar**: Displays upcoming campus events, workshops, seminars, and club meetings.
2. **Student Portal Integration**: Access to student accounts, academic schedules, grades, and faculty information.
3. **Campus Map & Navigation**: An interactive map showing key locations like libraries, cafeterias, labs, and lecture halls.
4. **Discussion Forums**: Channels for students to discuss subjects, events, or general campus activities.
5. **Notifications**: Instant notifications for important announcements, deadlines, or event reminders.
6. **Club and Activity Signups**: Join campus clubs and activities via the app.
7. **Emergency Contacts**: Quick access to emergency services, health services, or administrative offices.

Tools:

Android Studio
Database (SQLite or any modern database language)

Supervisor:

Name: Abdul Qahhar Mohsin
Email: mohsin@vu.edu.pk
Skype id: aqmohsin.vu

Campus Navigator

Project Domain / Category

Mobile Apps.

Abstract/Introduction

Campus Navigator is a mobile application aimed at improving campus navigation by offering real-time, location-based guidance to students, faculty, and visitors.

The platform will incorporate GPS for outdoor navigation and an Indoor Positioning System (IPS) for precise indoor directions within campus buildings.

The application is designed to simplify navigation, enhance campus experiences, and assist users in locating rooms, offices, and points of interest (POIs) efficiently.

Functional Requirements

1. **User Authentication and Authorization**
    - Secure Registration/Login: Users should be able to register and securely log into the app to access navigation services.
    - Role-based Access: Different access levels should be established for students, faculty, and administrators, with administrators able to manage maps and locations.

2. **Campus Navigation**
    - Outdoor Navigation (GPS): The app should provide real-time GPS-based navigation to outdoor locations like academic buildings, cafes, and parking lots.
    - Indoor Navigation (IPS): An Indoor Positioning System should guide users within buildings to specific rooms, labs, or offices.

3. **Search Functionality**
    - Location Search: Users should be able to search for locations by name or keyword, such as classrooms, labs, or specific departments.
    - Navigation from Search: The app should display the searched location and offer the option to navigate directly to it.

4. **Course Schedule Integration**
    - Classroom Finder: Students should be able to sync their course schedules and be guided to their classrooms on campus.

5. **Points of Interest (POI)**
    - Nearby Facilities: The app should display nearby POIs like restrooms, libraries, and cafeterias based on the user's current location.

6. **Accessibility Features**
    - Wheelchair-Accessible Routes: The app should offer routes that cater to users with mobility needs, ensuring that paths are accessible for all.

7. **Admin Dashboard**
    - Map Management: Administrators should have the ability to add or update buildings, classrooms, and POIs on the campus map through an easy-to-use interface.

Tools:

Software Requirements

Frontend: Android Studio for native Android development
Backend: ASP.NET Core
Database: Microsoft SQL Server

Supervisor:

Name: Nadia Tabassum
Email ID: nadiatabassum@vu.edu.pk
Skype ID: nadia.vu.cs

Event Management Mobile Application

Project Domain / Category
Mobile Application

Abstract / Introduction

You are required to develop a mobile application for managing various types of events such as corporate meetings, birthdays, seminars, or private parties. Users can use this app to search for venues based on event type, book event spaces, select event services like catering, decorations, and audio-visual equipment.

The app should allow users to customize the event details based on their preferences and requirements. Users can pay a deposit for the booking through an integrated payment system. The event organizers (venue owners or service providers) will receive the booking details and assign their team for execution.

Two days before the event, the user will pay 80% of the total cost, and after the event, the remaining balance will be settled. The app will also send reminders and updates regarding the event's progress.

Functional Requirements:

1. Users should be able to register and log in to the mobile app.
2. The app should display available venues and services based on the type of event (corporate meeting, birthday, etc.).
3. Users should be able to search and filter venues based on location, capacity, and available dates.
4. The app should provide an interface to book venues and select additional services like catering, decorations, and equipment.
5. Users can customize event details, including the number of attendees, food menus (vegetarian, *no*n-vegetarian, drinks), and specific requirements for decorations and audio-visual setups.
6. Upon finalizing the event details, users should submit their booking requests.
7. Event organizers (admin) will receive the booking request and review it. If accepted, confirmation will be sent via email, SMS, or in-app *no*tification.
8. Users will be able to make a token payment through the app to secure the venue and services.
9. If the booking is confirmed, the app should display a breakdown of responsibilities for the event (catering team, decorators, AV equipment setup).
10. Organizers (admin) should assign tasks to their team members for event preparation. The assigned responsibilities should be visible to both the admin and the employees.
11. If a user does *no*t pay the token within 48 hours, the booking should be automatically canceled, and the slot becomes available to other users.
12. Users should be able to cancel their bookings up to a certain time before the event. The token money may be withheld, and the slot will be made available again for others to book.

13. Two days before the event, users must pay 80% of the total event cost. After the event is completed, the user will pay the remaining balance.
14. After the event, the admin should be able to formally close the event in the app.
15. The app should send reminders and updates to the user about payment deadlines and upcoming events.

Tools:

- Android Studio
- Firebase or SQLite for database management

Supervisor:

Name: Muhammad Anwar
Email ID: manwar@vu.edu.pk
Skype ID: lovelyanwar

Online family wear sale app.

Project Domain / Category

Mobile Apps.

Abstract/Introduction

The project aims to deliver an online shopping application to the Android platform. Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project attempts to provide the advantages of online shopping to customers of a real shop. It helps buy the products in the shop anywhere through the internet by using an Android device. Thus, the customer will get online shopping and home delivery service from his favorite shop.

Functional Requirements:

- Admin can add, modify, delete and search Products by first name, Last name and date of registration, category and sub category.
- User can create their log in and profile by first name, Last name, Gender, address, contact no and date of registration, can order, select courier and payment mode.
- Moderators can create their log in and profile by first name, Last name, Gender domain, address, contact no and date of registration, Approve the quality of vender product to sale, track order conformation and delivery,
- Admin can add, modify, delete and search announcements by date and Type.
- Admin can add, modify, delete and search insurance/payment receiving companies and courier for delivery, Manager payments record.
- Admin can add, modify, delete and search products of Moderators by different category .
- Admin can add, modify, delete and search moderators' profile by different product domains.
- Admin can add, modify, delete and search payments and transaction by users.
- Admin can add, modify, delete and search moderators by date, users type and product category.
- Admin/system can add, modify, delete and search employees by id, first name and Last name.

Tools:

Android Studio
Database (SQLite or any modern database language)

Supervisor:

Name: Imran Akhtar
Email: imran.akhtar@vu.edu.pk
Skype id: msisl.net

Parent Portal Mobile App

Project Domain / Category

Android Mobile Application

Abstract / Introduction

In this world of mobile computation, it is necessary to have the school's parent portal accessible to parents via mobile application. With the mobile application, parent will be able to stay informed of their child performance with ease.
Using the mobile application parent will be able to see the student profile, attendance record, time table, syllabus, homework/assignments, results, date sheet, fee vouchers/status, feedback, school transport and announcements/news.
This mobile application will also enable parents to view the notifications sent by the school to parents.
This mobile application would be a valuable tool in bridging the gap between parents and the school, ultimately supporting the student's educational journey.

Functional Requirements:

Parents will be able to

1. Easily check their child's attendance records, ensuring they stay informed about their presence in school. Attendance record includes no. of present, absent, leave and holidays.
2. Access report cards which can help parents track their child's academic progress throughout the term.
3. View the fee vouchers and fee history through the app.
4. Submit the student leave application.
5. View the daily diary including classwork, homework, assignments etc.
6. View the time table and syllabus.
7. View transport facility including the bus driver detail.
8. View the profile of students.
9. Provide feedback and reviews to teachers and administration.
10. Receive feedback and reviews from school.
11. View school announcements, events, and important notices sent directly to their mobile device.

Tools:

This will be an android application. Student can use Java and Android Studio.
Student should not generate Application using any web based/offline tools like "MIT Android App Inventor"

Supervisor:

Name: Humaira Naeem
Email ID: humairanaeem@vu.edu.pk
Skype ID: humera_naeem

Personalized Nutrition and Fitness Management System for Android

Project Domain / Category

Android Application

Abstract / Introduction

The "Personalized Nutrition and Fitness Management System for Android" is supposed to empower users to lead a healthier lifestyle through tailored diet and fitness plans. This mobile application will offer a customized nutritional guidance and fitness routines based on individual user profiles, including factors like age, weight, height, fitness objectives, and dietary preferences. The system uses a combination of data inputs and algorithms to provide recommendations that are specific to the user's needs, making it easier to achieve and maintain personal health goals.

The application integrates essential features such as calorie counting, meal planning, workout tracking, and real-time progress monitoring. With an intuitive and user-friendly interface, it enables users to navigate through various functionalities, promoting consistent engagement and usage. Application Programming Interface (API) integrations allow for accessing nutritional databases and fitness-related resources, enhancing the application's overall capability. To further enhance functionality, integration with Google Fit is recommended. By connecting the app to Google Fit, users can automatically sync fitness data, track daily activity levels, and monitor health metrics in real time, providing a more comprehensive overview of their fitness progress. This integration will improve the accuracy and convenience of the system, offering users a seamless experience across platforms.

Functional Requirements:

1. FR1: The app must allow users to create their profile by including information such as age, weight, height, gender, fitness goals, activity level, and dietary preferences etc.
2. FR2: The system should generate meal plans based on the user's dietary requirements, caloric intake needs, and personal preferences (e.g., vegetarian, low-carb, etc.).
3. FR3: The app must provide specific fitness routines (e.g., cardio, strength, flexibility) according to the user's fitness level and goals (weight loss, muscle gain, endurance).
4. FR4: Users must be able to log their meals, and the app should calculate their daily caloric intake.
5. FR5: The app should track physical activities such as steps, distance, and exercise duration. Integration with Google Fit is recommended for real-time data syncing.
6. FR6: Users should be able to view progress metrics such as weight change, calories burned, and performance improvements through visually in the form of graphs.
7. FR7: The app should send reminders for scheduled meals, workouts, and hydration, along with motivational notifications to keep users engaged.
8. FR8: Users must be able to set fitness and dietary goals, and the app should track their progress towards these goals with regular feedback.
9. FR9: The app should connect to external APIs to retrieve accurate nutritional data.
10. FR10: Users should have the option to share their fitness achievements, progress updates, and milestones on social media platforms like Instagram, Facebook, or Twitter directly from the app.

Tools:

Recommended Programing Languages:

Android Java, Kotlin or any other relevant language.

IDEs:

Android Studio or any other relevant IDE for android applications.

Database:

Firebase, MySQL or any other similar database for storage

Supervisor:

Name: Irshad Nasir
Email ID: irshad.nasir@vu.edu.pk
Skype ID: live:.cid.325c9aa99b0279e4

Smart Solar Solution Android App

Project Domain / Category

Mobile App

Abstract / Introduction

The objective of this project is to develop a mobile application that offers a complete solar system installation service. The app will allow users to browse and order solar products (solar panels, stands, batteries, inverters, etc.), book installation services, request maintenance, and receive expert consultations. The app will cater to customers who are looking to set up a solar system, providing them with product recommendations, cost estimates, and after-sale services

Functional Requirements:

1. User Registration and Profile Management

- **Sign-up/Login:** Users can create an account via email, phone number, or social media.
- **User Profile:** Store user information such as contact details, preferred solar system configurations, and order history.
- **Account Management:** Users can update personal details and view their transaction history.

2. Product Catalog and Filtering

- **Solar Products Listing:** A detailed catalog of solar panels, inverters, batteries, stands (L2, L3, L4), and other accessories.
- **Product Details:** Display detailed descriptions, technical specifications, prices, and installation guides for each product.
- **Filtering Options:** Users can filter products based on capacity, price, brand, and installation type.
- **Product Comparison:** Allow users to compare different products side by side based on their technical specifications and prices.

3. Installation Service Booking

- **Service Types:** Users can choose installation services such as complete solar system setup, battery replacement, or panel cleaning.
- **Installation Packages:** Provide various service packages depending on system size (e.g., small residential, commercial, industrial).
- **Location-Based Booking:** Users can select their location for service availability.
- **Installation Scheduling:** Users can schedule appointments for installation, maintenance, or consultation at their convenience.

4. Cost Estimator

- **Custom Solar System Estimator:** Users can enter their energy requirements and location to receive a detailed cost estimate for solar system setup, including equipment, installation, and maintenance.
- **Dynamic Price Calculation:** Provide users with an automatic cost calculation based on their selected products and services.
- **Financing Options:** Display available financing plans or payment installment options if applicable.

5. Order Management

- **Cart and Checkout:** Users can add products and services to the cart and proceed with the checkout process.
- **Payment Gateway:** bank account transfer

- **Order Tracking:** Users can track the status of their order (pending, shipped, installed, etc.).

6. Solar System Maintenance and Support

- **Maintenance Requests:** Users can submit requests for maintenance services (panel cleaning, inverter checks, etc.).
- **Issue Reporting:** Users can report system issues, and the app will route them to the appropriate service team.

7. Admin Dashboard for Solar Providers

- **Service Management:** Solar service providers can manage incoming service requests, schedule installations, and assign tasks to technicians.
- **Order Fulfillment:** Manage customer orders, including inventory tracking for solar equipment and accessories.
- **Customer Communication:** Admins can send updates, reminders, and promotional offers directly through the app.

8. Customer Reviews and Feedback

- **Product Reviews:** Users can rate and review the products they've purchased and installed.
- **Service Feedback:** After installation or maintenance, users can provide feedback on the quality of service.
- **Rating System:** Display product and service ratings to help other users make informed decisions.

9. Notifications and Alerts

- **Order and Service Updates:** Send notifications to users regarding their orders, service bookings, and maintenance schedules.
- **Product Alerts:** Notify users about new product arrivals, discounts, or offers.
- **Renewal Reminders:** Remind users of upcoming maintenance requirements or product replacements (e.g., battery replacement cycle).

10. Solar Consultation and Knowledge Base

- **Expert Consultation:** Users can request consultations with solar energy experts to get advice on system setup, energy savings, or product selection.
- **Knowledge Base:** Provide a section with articles, tutorials, and videos on solar energy systems, installation guides, and troubleshooting tips.

11. Product and Service Customization

- **Tailored Solutions:** Allow users to customize their solar system configurations based on energy needs, such as selecting specific panel types, battery capacities, and inverter sizes.
- **Bundled Packages:** Offer pre-defined solar system packages (e.g., residential, commercial, off-grid) that include all necessary components and installation services.

Tools:

You can use any android development tool.

Supervisor:

Name: Saeed Nasir
Email ID: saeednasir@vu.edu.pk
Skype ID: saeed-nasir or you can search it through email mansbit@hotmail.com

NLP Chatbot Development using Dialogflow

Project Domain:
Web Application/NLP

Abstract / Introduction:
Chatbots are increasingly becoming integral to industries such as customer service, e-commerce, healthcare, and education. In this project, students will develop a chatbot using Google Dialogflow, an NLP-powered conversational interface. The chatbot should address a specific business need by automating customer interaction, providing relevant responses, and enhancing user experience. Students can select a case study from one of three industries: **Training Company**, **Pharmacy Store**, or **Restaurant**

Functional Requirements
The primary goal of this project is to design and develop an AI-powered chatbot that effectively understands and responds to user inquiries using **Dialogflow**, a prominent tool in natural language processing (NLP). Through this project, students will:

- **Identify a Real-World Case Study**: Choose a specific domain (Training Company, Pharmacy Store, or Restaurant) to explore the unique needs and challenges faced in that industry.
- **Define the Problem**: Analyze the chosen case study to clearly articulate the specific problem or inefficiency that the chatbot aims to address, detailing how its implementation can enhance operational processes and improve user experiences.
- **Build an Interactive Chatbot**: Develop a functional chatbot that not only answers queries but also performs key tasks such as making reservations, processing orders, or addressing frequently asked questions (FAQs). The chatbot should engage users in a conversational manner, ensuring a seamless interaction that meets their needs.

Case Study Options:
Students are encouraged to pick one of the following industries for their case study:

1. **Training Company**: A chatbot that assists with student inquiries, course registrations, schedules, and other administrative tasks.
2. **Pharmacy Store**: A chatbot that helps customers find products, check availability, set reminders for medicine, or request home delivery.
3. **Restaurant**: A chatbot that handles table reservations, order-taking, menu navigation, or customer support.

Tools:
Dialogflow: For chatbot design and natural language processing.
Backend: PHP/Python
Front End: HTML/CSS/JS
Database: MySQL, **IDE:** VSCode

Supervisor: Name: Abdullah Qamar
Email ID: abdullah.qamar@vu.edu.pk
Skype ID: abdulaha462

Hate Speech Detection using Machine Learning for Roman Urdu

Project Domain / Category

Data Science / Machine Learning / Natural Language Processing (NLP)

Abstract / Introduction

With the rise of online social media platforms, the issue of hate speech has become increasingly prevalent. Hate speech can lead to social tension and harm, especially in multilingual countries like Pakistan, where Roman Urdu is commonly used online. This project aims to develop a machine learning model to detect hate speech in Roman Urdu comments. The focus is on gathering a robust dataset of Roman Urdu comments from social media, pre-processing it, extracting relevant features, and training machine learning models to classify hate speech effectively. Additionally, a web interface will be developed post-completion to allow users to test the model's performance with real-time data.

Functional Requirements:

Admin (Student) will perform all these (Functional Requirements) tasks.

1. **Data-Collection**
   - For this project, student will collect data from any social media platform (such as YouTube, Facebook, Twitter, or Instagram) to detect hate speech. The dataset must contain at least 5000 comments, focusing on Roman Urdu. The data set is shared in the link below for the idea.
2. **Data Preparation**
   - Prepare the dataset by labelling it as "Hate Speech (HS)" or "Non-Hate Speech (NHS)." This step involves manually reviewing the data to assign appropriate labels, ensuring the dataset is clean and ready for use in machine learning.
3. **Data Pre-Processing**
   - As most of the data in the real world are incomplete containing noisy and missing values. Therefore, student have to apply pre-processing on data. In pre-processing, student will *normalize* the dataset, handle *stop words, missing values*, and *noise & outliers,* and remove *duplicate values.*
4. **Feature Extraction**
   - After the pre-processing step, student will apply the feature extraction method. Student can use Term Frequency - Inverse Document Frequency (TF-IDF), Uni-Gram (1-Gram), Bi-Grams (2-Grams), Tri-Grams (3-Grams), or N-Grams feature extraction method.
5. **Train & Test Data**
   - Split the dataset into 70% training and 30% testing data for the machine learning models.
6. **Machine learning Techniques**
   - Student must use at least three *classifiers/models* (e.g. Naïve Bayes, Naïve Bayes Multinomial, Poly Kernel, RBF Kernel, Decision Tree, Random Tree or Random Forest Tree etc.) of three different *machine learning techniques/algorithms*.
7. **Confusion Matrix**
   - Generate a confusion matrix to evaluate the performance of each classification model.
8. **Accuracy Evaluation**

- Find the accuracy of all techniques and compare their accuracy.
- This project will also tell us which machine learning technique is better to detect Toxic comments.

9. **Web Interface Integration**
    - After the model development, integrate a web interface to allow users to test the model's performance using real-time comments.

Tools/Techniques:

- **Anaconda:** Python distribution platform for development.
- **Jupiter Notebook:** For implementing machine learning models.
- **Python:** Programming language used for data pre-processing, model training, and feature extraction.
- **Machine Learning Algorithms**: For training and testing hate speech detection.
- **Web Interface:** Basic HTML/CSS, Flask, or Django.

Prerequisite:

- Knowledge of Artificial Intelligence, Machine Learning, and Natural Language Processing concepts is required. Students will cover a short course relevant to these concepts, alongside SRS and Design initial documentation or see the links below.

Helping Material:

Python:

https://www.python.org/
https://www.w3schools.com/python/
https://www.tutorialspoint.com/python/index.htm

Feature Extraction Method*:*

https://towardsdatascience.com/feature-extraction-techniques-d619b56e31be
https://www.analyticsvidhya.com/blog/2021/04/guide-for-feature-extraction-techniques/
https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089
https://www.analyticsvidhya.com/blog/2021/07/feature-extraction-and-embeddings-in-nlp-a-beginners-guide-to-understand-natural-language-processing/
http://uc-r.github.io/creating-text-features

Machine Learning Techniques:

https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0
https://towardsdatascience.com/top-10-algorithms-for-machine-learning-beginners-149374935f3c
https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9
https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623
https://www.youtube.com/watch?v=fG4e4TUrJ3E
https://www.youtube.com/watch?v=7eh4d6sabA0

Dataset:

https://drive.google.com/file/d/1Jq62ErAQiMpWfEz9_DwSkjmyYdmwWWu6/view

Supervisor:

Name: Tayyab Waqar
Email ID: tayyab.waqar@vu.edu.pk
Skype ID: maliktayyab786_1

Identifying Fake Product Reviews Using Opinion Mining

Project Domain / Category:
NLP/Information Retrieval

Abstract / Introduction:
As most of the people require review about a product before spending their money on the product. So people come across various reviews in the website but these reviews are genuine or fake is not identified by the user. In some review websites some good reviews are added by the product company people itself in order to make product famous this people belong to Social Media Optimization team. They give good reviews for many different products manufactured by their own firm. User will not be able to find out whether the review is genuine or fake. To find out fake review in the website this "Identifying Fake Product Reviews Using Opinion Mining" system will be developed. This system will find out fake reviews made by the social media optimization team by identifying the IP address. User will login to the system using his user id and password and will view various products and will give review about the product. To find out the review is fake or genuine, system will find out the IP address of the user if the system observe fake review send by the same IP Address many at times it will inform the admin to remove that review from the system. Once the fake reviews removed, the system will rate and rank the opinions on different products by using opinion mining techniques. This system helps the user to find out correct review of the product.

Functional Requirements:

1. The system should be able to add products to the system.
2. The system should be able to delete the review which is fake.
3. The system should be able to allow user once to access the system, user can view product and can post review about the product.
4. The system should be able to track the IP address of the user.
5. If the system observes fake review coming from same IP address many a times, then system should be able to track this IP address and will inform the admin to remove this review from the system.

❖ **Modules:**

The system comprises of 2 major modules with their sub-modules as follows:

1. **Admin Login:** Admin login to the system using his admin ID and password.
    - Add product:
        - Admin will add product to the system.
    - **Store Review for detection of Fake Reviews:**

The system should store each review along with associated metadata, such as:

- User ID
- Timestamp
- Product ID
- IP Address (for tracking)

- Delete Review:
    - Admin will remove the review which tracked by the system as fake.

- **Review Monitoring (Opinion Mining):**

Once the system detect/remove the fake reviews. The system should analyze each submitted review on the products using opinion mining (text analysis) techniques and rank the product according to their sentiments score. This includes:

- **Sentiment Analysis**: To detect whether the review is positive, negative, or neutral.
  -

2. **User Login:** User will login to the system using his user ID and password.
   - View product:
     - User will view product.
   - Post Review:
     - User can post review about the product.

Tools:

- Python/Visual Studio
- Senti WordNet Dictionary/Vader
- WAMP Server
- My SQL 5.6
- Apache/Nginx (Server-side to capture IP addresses)
- Notepad++

Supervisor:
Name: Tayyaba Sehar
Email ID: tayyaba.sehar@vu.edu.pk
Skype ID: Tayyaba.sehar13@outlook.com

Keyword-Based Search Engine for Text Documents

Project Domain / Category

Information Retrieval

Abstract / Introduction

In today's information-driven world, individuals and organizations generate and store an enormous amount of textual data across various platforms. The ability to efficiently retrieve relevant information from this vast pool is crucial for productivity and decision-making. This project aims to develop a **Keyword-Based Search Engine for Text Documents**, providing a streamlined solution for users to quickly locate specific information within their documents. By offering an intuitive web interface, the search engine empowers users to input keywords and phrases to receive immediate, relevant search results. The system prioritizes the most pertinent documents, ensuring users find what they need swiftly and effectively. This project addresses the increasing demand for efficient information retrieval systems, ultimately enhancing document management processes and boosting overall efficiency.

Functional Requirements

1) **User Interface**:
    a) A clean and intuitive web-based interface that allows users to interact with the search engine seamlessly.
    b) The main page should include:
        i) A prominently displayed search bar for keyword and phrase input.
        ii) Clear instructions or placeholder text to guide users in entering their search terms.
        iii) A section displaying search results in a well-organized manner, allowing easy navigation.
        iv) Responsive design for compatibility across different devices (desktops, tablets, smartphones).
        v) Option to view details of each document directly from the search results.

2) **Document Uploading**:
    a) Users can upload multiple text documents (e.g., .txt, .csv) via a file input interface.
    b) Provide feedback upon successful document upload, indicating the number of documents uploaded.
    c) Users must create an account and login first to upload text documents. The account will require basic information such as username, password, and email address.

3) **Keyword and Phrase Search Functionality with Spell Checker**:
    a) Users can enter keywords and phrases in a search box to retrieve relevant documents.
    b) The system processes the input keywords and phrases, matching them against the content of the uploaded documents.
    c) As users type keywords or phrases, the system automatically checks for spelling errors in real-time, underlining misspelled words with a red wavy line.
    d) When users right-click a misspelled word, the system provides spelling suggestions.
    e) When hovering over a misspelled word, the system displays the correct spellings as a tooltip.

f) Allow users to add custom words to their dictionary if they frequently use specialized terms.
g) Display the most relevant document first, followed by subsequent results ranked by relevance, along with the occurrences of the keywords or phrases within those documents.
h) A user does not need any account or login for using search functionality, viewing and downloading documents.

4) **Search Results Display**:
   a) Present the search results in a user-friendly format, showing:
      i) Document name
      ii) Snippet of text containing the keyword(s) or phrase(s)
      iii) Option to view or download the document.
   b) Highlight the keywords and phrases within the snippets to enhance visibility.

5) **Document Indexing**:
   a) Automatically index the contents of uploaded text documents to optimize search performance.
   b) Store relevant metadata for each document, including:
      i) Document name
      ii) Upload date
      iii) Username (who uploaded the document)
      iv) Keywords and phrases extracted from the content.
   c) Implement efficient indexing algorithms to improve search speed and accuracy, considering both keyword and phrase indexing.

6) **User Notifications**:
   a) Display success messages upon successful upload of documents.
   b) Provide feedback for potential issues, such as no documents uploaded, to guide user actions.

7) **Accessibility Features**:
   a) Ensure the web interface is accessible to all users, including those with disabilities (e.g., using proper HTML semantic elements, keyboard navigation).

8) **Data Persistence**

   a) **Efficient Storage**: Use **SQLite** for metadata and store documents in a dedicated directory.
   b) **Indexing**: Implement **Whoosh** to index document content for fast searches.
   c) **Optimization**: Optimize SQL queries and utilize caching to enhance performance.
   d) **Data Integrity**: Ensure data consistency through transactions and robust error handling.
   e) **Session Management**: Track user interactions and maintain a history of recently accessed documents.
   f) **Backups**: Schedule regular backups for data recovery.
   g) **Scalability**: Design for future growth in document volume and user access.

9) **Admin Management**: There should be an Admin account to manage user accounts, including adding, modifying, or deleting user accounts and monitoring user activities.

Tools:

1. **Python**: The primary programming language for backend development, document processing, and implementing search algorithms.
2. **Flask**: A python web framework for building the web interface and handling user requests.
3. **SQLite**: The database for storing uploaded documents and indexed data.
4. **Whoosh**: A python search library for indexing and searching functionalities.
5. **HTML/CSS**: For structuring and styling the web interface.
6. **JavaScript**: For client-side interactivity and enhancing user experience.

Supervisor:
Name: Muhammad Ilyas
Email ID: muhammad.ilyas@vu.edu.pk
Skype ID: live:.cid.813fc730d8c19cf8

Real-Time Network Intrusion Detection Using Wireshark and Advanced Ensemble Learning Techniques

Project Domain / Category

Networking/Machine Learning/ Research

Abstract / Introduction

With the rapid expansion of internet-connected devices, securing network infrastructures has become a major concern. Real-time network traffic monitoring and analysis are crucial to detect potential cyber threats and intrusions. This project focuses on developing an intrusion detection system (IDS) using real-time network traffic data captured by Wireshark. Students will utilize Wireshark to capture, preprocess, and extract relevant features from the network data. These features will then be used to train advanced ensemble learning models, including TabNet, CatBoost and LightGBM, to identify malicious network activities. The system will also include a web application that allows users to upload network traffic data, analyze it for intrusions, and display results in real-time.

Functional Requirements:

1. Students will use Wireshark to capture real-time network traffic data, exporting the captured data to a CSV file for further analysis. The capture will include features such as protocol types, IP addresses, packet sizes, time intervals, and more.
2. Students will preprocess the captured data by cleaning it, encoding categorical variables (e.g., protocol types), and normalizing numerical features (e.g., packet sizes, time intervals).
3. The processed data will then be structured into a CSV dataset, including labels for normal and malicious traffic activities, to be used for training the machine learning models.
4. Students will explore various machine learning classification techniques and select advanced ensemble models, including TabNet, CatBoost, LightGBM, and stacking ensembles. These models will be trained on the preprocessed network traffic dataset.
5. Model evaluation will be performed using state-of-the-art classification metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
6. A web application will be built using Python Flask or Django, providing an interface where users can upload Wireshark-captured network traffic data for analysis. The application will process the uploaded data, detect potential intrusions using the trained ensemble models, and display the results, along with key network parameters that influenced the classification.

Tools:

- Programming Language: Python
- Traffic Capture Tool: Wireshark for capturing network traffic data in real-time and exporting it to a CSV file for analysis.
  https://www.wireshark.org/download.html
- Machine Learning Libraries: Scikit-learn, XGBoost, LightGBM, pytorch-tabnet, CatBoost
- Web Development Frameworks: Python Flask or Django for building the web application

- Operating System: Any (e.g., Windows, Linux).
- Additional Tools: Jupyter Notebook for data analysis and model training, Matplotlib/Seaborn for data visualization.

Supervisor:

Name: Laraib Sana
Email ID: laraib.sana@vu.edu.pk
Skype ID: Laraib.sana

Expense Tracker Application

Category

Web Application

Introduction:

Develop an application to manage and track personal or business expenses. Users will be able to input their income and expenses, and the system will help them visualize their spending patterns through reports and charts.

Functional Requirements:

- Users can sign up, log in, and manage their accounts securely.
- Users can input their income and categorize expenses (e.g., food, transportation, utilities).
- Users can categorize expenses for easier tracking and analysis.
- Users can view a detailed history of all their transactions, sorted by date, category, or amount.
- Users can set monthly budgets for different categories and track how close they are to exceeding them.
- The system generates charts and reports to visualize users' spending patterns (e.g., pie charts for categories, bar graphs for monthly spending).
- Users can download or view monthly and yearly financial summaries.
- Users can search for transactions using keywords or filters such as date range, amount range, or category.
- The system sends notifications or alerts when users approach their budget limit or when significant changes occur in their financial data.
- Users can export their transaction data to CSV or Excel format and import previous data into the system.

Technologies:

- Python/JavaScript
- SQLite/MySQL (for database)
- HTML/CSS (for frontend)

Name: Muhammad.salman.bashir
Email ID: salmanbashir@vu.edu.pk
Skype ID: Muhammad.salman.bashir

Resources Monitoring for systems on Network

Project Domain / Category
Networking

Abstract/Introduction
Resources Monitoring for system on Network, the purpose of this project is to observe, monitor and evaluate the CPU and RAM usage connected on a network. This is a system that administers the efficiency by assessing various resources against the people and programs that use those resources. This resource monitoring project will enable effective oversight of CPU and RAM usage across a network, ultimately leading to improved performance and resource allocation. By implementing a structured approach to monitoring, evaluation, and reporting, organizations can enhance their operational efficiency and responsiveness to resource-related issues. Implement a system to track CPU and RAM usage in real time across all connected devices. Generate reports to analyze performance metrics and provide insights for optimization.

Functional Requirements:

- Monitoring System for network resources (MSNR), is a desktop application providing the below mentioned tasks:

Module:

User management

- This Monitoring system will have two ends
- One will be the “Administrator” to monitor the system
- The other will be “Simple user”

Communication Management
- This is all about exchange of the information among clients and server of application.

Resource Monitoring
- Observation and monitoring resources on connected net.
- Continuously collect CPU and RAM usage data from each connected system.
- Allow administrators to set and modify thresholds for CPU and RAM usage.

Report manager
- reports generated in the form of graphs
- Generate periodic reports after one hour summarizing resource usage.

Tools:
- VB.net or JAVA, etc.

Supervisor:
Name: Asim Mehmood
Email ID: asimmehmood@vu.edu.pk
Skype ID: sardar-asim

Modern E-Learning Platform

1. Introduction

With the increasing demand for online education, there is a need for modern, scalable e-learning platforms that can deliver high-quality content and foster interaction between students and instructors. This proposal outlines the development of an e-learning platform leveraging **Strapi** as the headless CMS and **Next.js** for the frontend, providing a flexible, scalable, and user-friendly platform for both learners and educators.

2. Objectives

The primary goals of the platform are to:

- Offer a seamless experience for course creation, management, and enrollment.
- Support lesson-based learning with multimedia content.
- Enable student feedback to refine and improve courses.
- Integrate a flexible content management system (Strapi) for easy handling of courses, lessons, and media.
- Ensure a smooth and responsive user experience using Next.js.

3. Key Features

3.1 Course Management

- **Course Catalog:** An organized library of courses with detailed descriptions, objectives, prerequisites, and instructor information. Courses can be filtered by categories, difficulty level, and rating.
- **Lesson Structure:** Each course will be divided into lessons, with each lesson containing videos, readings, and assignments.
- **Dynamic Content Management:** Strapi CMS will be used to manage and update content easily without requiring technical knowledge. Instructors can upload course materials, making content creation flexible.

3.2 Lessons and Learning Materials

- **Interactive Lessons:** Lessons will be composed of video lectures and reading materials, all managed via Strapi.
- **Multimedia Support:** The platform will support multiple media formats, including video, audio, PDFs, and images. Media will be served using a cloud service (e.g., Cloudinary) for efficient loading times.
- **Progress Tracking:** Users can track their progress through each course, with automatic reminders to encourage completion.

3.3 Student Feedback and Interaction

- **Course Reviews:** Students can rate and review courses upon completion, providing valuable feedback to instructors and helping future learners choose the right course.
- **Surveys and Feedback Forms:** Periodic surveys and feedback forms will allow students to share their learning experience and suggest improvements.
- **Discussion Forums:** Integration of community-based features like discussion forums and live chat will facilitate interaction between students and their instructors.

3.4 Instructor Tools

- **Course Creation Dashboard:** Instructors can use Strapi's intuitive interface to create and manage their courses. They can upload lessons, organize course flow, and monitor student progress from a single dashboard.
- **Performance Analytics:** Instructors will have access to analytics on course performance, including student engagement, completion rates, and feedback, all generated via Next.js integration with Strapi.

3.5 Assessments and Certification

- **Certification:** Upon completing a course, students will receive a certificate of completion, which can be shared on their social media profiles or downloaded for personal use.

3.6 Personalization and Adaptive Learning

- **Course Recommendations:** Using data collected via Strapi, the platform will recommend personalized courses based on the learner's interests and previous performance.
- **Adaptive Learning Paths:** The platform will adapt to student performance, offering suggestions for further study and additional resources if needed.

3.7 Mobile-Friendly Design

- **Responsive Design:** The platform will be fully optimized for mobile devices, ensuring that students can access courses and content from smartphones and tablets.
- **Offline Access:** The ability to download lessons and materials for offline access will be supported.

3.8 Accessibility and Inclusivity

- **Accessibility Features:** Features such as subtitles, transcripts, and keyboard navigation will ensure compliance with web accessibility standards (WCAG).

4. Key Roles and Responsibilities

4.1 Administrator

Administrators are responsible for overseeing the entire platform. They manage users, courses, payments, and the overall health of the system.

Permissions:

- Manage Users: Add, remove, or modify user roles (students, instructors, content creators).
- Course Management: View and manage all courses, approve or reject content submitted by instructors.
- System Configuration: Set up payment gateways, configure the platform, manage integrations (e.g., Cloudnary, Strapi, Next.js).
- Reporting and Analytics: Access detailed reports about platform performance, user activity, course completion rates, and financial transactions.
- Content Moderation: Oversee community forums, discussion groups, and feedback sections to maintain a healthy learning environment.

Responsibilities:

- Ensure the platform runs smoothly.
- Approve and verify new content or updates from Content Creators.
- Manage platform monetization options and promotional offers.

4.2 Content Creator / Instructor

Instructors or Content Creators are responsible for creating and managing course content. They engage with students and ensure that the content is up to date and relevant.

Permissions:

- Create and Manage Courses: Instructors can create new courses; add lessons assignments via Strapi.
- Upload Media: Upload videos, images, documents, and other multimedia content required for lessons.
- Manage Student Enrollments: View students enrolled in their courses and monitor their progress.
- View Feedback: Access reviews and feedback provided by students to improve course quality.

Responsibilities:

- Develop engaging, high-quality educational content.
- Monitor student progress and provide timely feedback on assignments and quizzes.
- Engage with students through discussion forums, live Q&A, and personalized feedback.

4.3 Student

Students are the primary users of the platform, focused on learning, progressing through lessons, and completing assignments.

Permissions:

- Access Courses: Enroll in free or paid courses. Track progress, revisit completed lessons, and review content at any time.
- Provide Feedback: Leave reviews and ratings for courses they have completed to help other learners and instructors.
- Track Progress and Certifications: View progress in real-time and download certificates upon course completion.

Responsibilities:

- Engage actively with the course material.
- Provide constructive feedback to help improve the course and the platform.

4.4 Guest User (Optional)

Guest users are visitors to the platform who can browse available courses but cannot enroll in or access full course content.

Permissions:

- Browse Courses: View course catalogs, descriptions, and ratings.
- Limited Access to Lessons: Preview a small portion of the course, such as a free introductory lesson, without signing up.

Responsibilities:

- Explore the platform's offerings and enroll to access full courses.

5. Technology Stack

5.1 Backend: Strapi

- **Content Management System:** Strapi will be the primary CMS used for creating and managing course content, lessons, and feedback.
- **Media Management:** Integration with Cloudinary for media hosting, allowing instructors to upload videos, images, and other materials.
- **APIs:** Strapi will expose RESTful or GraphQL APIs to serve content dynamically to the frontend, enabling a seamless flow of data between the backend and frontend.

5.2 Frontend: Next.js

- **Framework:** Next.js will be used for the frontend, providing server-side rendering (SSR) and static site generation (SSG) for fast loading times and better SEO performance.
- **Dynamic Routing:** Next.js will allow for dynamic course and lesson routing, ensuring a smooth user experience as students navigate through different sections.
- **Interactive UI:** Next.js' React-based architecture will enable the platform to provide a highly interactive user interface, including real-time quizzes, progress tracking, and live discussions.

5.3 Hosting and Infrastructure

- **Hosting:** The platform will be hosted on a scalable cloud platform such as AWS, Vercel, or Railway.app to ensure high availability and performance.
- **CDN Integration:** To serve static assets quickly and efficiently, the platform will utilize a CDN (Content Delivery Network) to cache content closer to users.

5.4 Security and Authentication

- **User Authentication:** Next.js and Strapi will handle user authentication via JWT (JSON Web Tokens) or OAuth2 using the Auth.js (V5) Library to ensure secure login and session management.
- **Role-Based Access Control:** Role-based permissions will be set up in Strapi to control access for different user roles (students, instructors, admins).

6. Monetization Strategies

- **Freemium Model:** The platform will offer a combination of free and paid courses, with premium features such as certificates and instructor feedback for paid users.
- **Subscription Model:** Monthly or yearly subscription plans for users to access premium content or unlock additional features like exclusive courses.

7. Conclusion

Leveraging Strapi and Next.js will provide the platform with a robust, scalable foundation that is easy to manage, flexible, and capable of delivering high-quality educational content. This platform will address the needs of both learners and instructors, making it a valuable tool in modern education.

Supervisor:
Name: Amjad Iqbal Khan
Email ID: amjad.iqbal@vu.edu.pk
Skype ID: amjadiqbalkhanniazi

On-Demand Food Delivery App

Project Domain / Category

Web Programming

Abstract / Introduction

This project focuses on developing a user-friendly food delivery app connecting customers with nearby restaurants. Users can browse menus, place orders, track deliveries, and leave reviews. The app features three core modules: Customer, Restaurant, and Delivery Boy, ensuring smooth interactions between all parties.

Restaurant owners can manage menus and orders, while delivery personnel use a dedicated interface to accept and track deliveries. The goal is to create a reliable, scalable platform that simplifies the food ordering process and provides fast, efficient delivery.

Functional Requirements:

The functional requirements are:

1. **User Registration & Login**
    a. FR1: Users can register with the app using email, phone number, or social media accounts.
2. **Restaurant Search and Filter**
    a. FR2: Users can search for restaurants by location, cuisine type, or price range.
    b. FR3: Filters available for sorting restaurants by ratings, distance, and offers.
3. **Menu Browsing and Selection**
    a. FR4: Users can view restaurant menus, including dishes, prices, and images.
    b. FR5: Users can select food items, specify quantities, and add to cart.
4. **Order Management**
    a. FR6: Users can review their cart before placing the order.
5. **Payment Integration**
    a. FR7: Cash on Delivery
6. **Restaurant Dashboard**
    a. FR8: Restaurants can create, edit, and manage their menu.
    b. FR9: Restaurants can manage orders, view transaction history, and update delivery status.
7. **User Notifications**
    a. FR10: Real-time notifications on order status (order received, preparation, out for delivery, delivered).
    b. FR11: Push notifications for special offers, discounts, or promotions.
8. **Review and Rating System**
    a. FR12: Users can leave reviews and rate restaurants based on their experience.
    b. FR13: Restaurants can respond to customer reviews.
9. **Admin Dashboard**
    a. FR14: Admin can manage users, restaurants, and track overall platform performance.
    b. FR15: Analytics dashboard for viewing user activity, sales reports, and other key metrics.
10. **Loyalty and Discounts**
    a. FR16: Users can receive and apply discount codes, loyalty points, or rewards for repeat purchases.

11. **Delivery Boy Module**
    a. FR17: Delivery personnel can register and log in to the app with their credentials.
    b. FR18: Delivery personnel can receive real-time notifications for new orders and accept/decline deliveries.
    c. FR19: Delivery personnel can view customer details, restaurant location, and delivery address.
    d. FR20: Delivery personnel can update order status (picked up, out for delivery, delivered).
    e. FR21: Delivery history and earnings dashboard for delivery personnel.
    f. FR22: Option for delivery personnel to rate customers and report issues.

Tools:
HTML, CSS, JS, BootStrap, React, Node JS, Next Js, MongoDb,VS Code

Supervisor:
Name: Adnan Asif
Email ID: adnanasif@vu.edu.pk
Skype ID: ch.adnanasif

Website for VU Sports Society

Project Domain / Category:

Web Application

Abstract / Introduction:

The VU Sports Society aims to promote physical activity, sportsmanship, and a healthy lifestyle among students and staff of Virtual University. Currently, the organization of sports events, member registrations, and the management of teams and schedules are often done manually, leading to inefficiencies and a lack of streamlined communication. With the increasing number of students and sports enthusiasts, an automated system is necessary to efficiently manage the growing demand for sports activities.

The objective of this project is to create a web platform where students and faculty members can easily participate in sports activities, register for events, manage teams, and stay informed about upcoming sports programs. The platform will also allow users to track their performance, join teams, view leaderboards, and communicate with coaches and other players. The system will improve the overall management of the sports society by offering features like event scheduling, real-time updates, and user-friendly interfaces for both administrators and participants.

Functional Requirements:

Participants (Students/Staff):

- Registration (Participants need to register to access the platform).
- Login/logout.
- Modify profile (Profiles should include sports preferences, past participation, and achievements).
- View and register for upcoming sports events (e.g., tournaments, friendly matches, fitness sessions).
- Join or create sports teams (e.g., football, cricket, badminton, etc.).
- Track personal and team performance through stats and leaderboards.
- Communicate with team members, coaches, and organizers.

Coaches/Team Leaders:

- Registration (Coach must need an approval from admin after registration)
- Login/logout.
- Modify profile (Include sports expertise, team management, and availability).
- Create and manage teams, assign roles, and coordinate practices or matches.
- Review participant profiles and manage team selections.
- Communicate with team members, monitor performance, and provide feedback.
- Organize events or matches and invite participants.

Administrator:

- Login/logout.
- Modify profiles (Participants, coaches, and team leaders).
- Add/delete/block participants, teams, or coaches.
- Approve team formations and event registrations.
- Schedule sports events and manage event logistics (e.g., venues, equipment, timings).
- Send messages and notifications to participants (event updates, team announcements, etc.).

- Manage sports categories (e.g., football, cricket, basketball) and assign organizers for each.
- View statistics on participation and generate reports on sports activities.

Tools: (You can use any of the following tools)

1. Programming Languages

- **HTML**: To structure web content.
- **CSS**: To style and layout web pages.
- **JavaScript**: For dynamic and interactive elements on the client side.

2. Front-End Development Frameworks

- **React.js**: A JavaScript library for building user interfaces.
- **Angular**: A platform for building mobile and desktop web applications.
- **Vue.js**: A progressive JavaScript framework for building UIs.
- **Bootstrap**: A CSS framework for responsive design.

3. Back-End Development

- **Node.js**: A JavaScript runtime for building scalable server-side applications.
- **Django**: A Python-based web framework.
- **Ruby on Rails**: A server-side framework written in Ruby.
- **Laravel**: A PHP framework for web application development.
- **Flask**: A lightweight Python web framework.

4. Databases

- **MySQL**: An open-source relational database.
- **PostgreSQL**: A powerful open-source relational database system.
- **MongoDB**: A NoSQL database for handling unstructured data.
- **SQLite**: A lightweight database for smaller projects.

3. Web Servers

- **Apache**: An open-source web server.
- **Nginx**: A high-performance HTTP server and reverse proxy.
- **XAMPP**: A local server for PHP and MySQL development.
- **WAMP**: A Windows-based local server for PHP, MySQL.

Supervisor:

Name: Abdur Rafay
Email ID: abdur.rafay@vu.edu.pk
Skype ID: live:abdurafay94