A Website for DoC, Hamdard University

Final Year Project Report

Submitted by
SHAMEER ABID (2278-2021)
SYEDA NOREEN ZAHRA (2025-2021)
HAMZA SHAIKH (2595-2021)

Supervisor(s)
SIR AFZAAL HUSSAIN

FALL 2024

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science
2024

Faculty of Engineering Sciences and Technology

Hamdard Institute of Engineering and Technology
Hamdard University, Main Campus, Karachi, Pakistan

Certificate of Approval

Faculty of Engineering Sciences and Technology

Hamdard Institute of Engineering and Technology
Hamdard University, Karachi, Pakistan

This project **"A website for DoC, Hamdard University"** is presented by "**Syeda Noreen Zahra, Shahmeer Abid, and Hamza Sheikh"** under the supervision of "Mr. Afzal Hussain" and approved by the project examination committee, and acknowledged by the Hamdard Institute of Engineering and Technology, in the fulfillment of the requirements for the Bachelor degree of **Computer Science**

_____ _____

Mr. Afzaal Hussain
(Project Supervisor)

In-charge FYP-Committee

_____ _____

-
(Project Co-Supervisor)

Chairman
(Department of Computing)

_____

(Dean, FEST)

Authors' Declaration

We declare that this project report was carried out in accordance with the rules and regulations of Hamdard University. The work is original except were indicated by special references in the text and no part of the report has been submitted for any other degree. The report has not been presented to any other University for examination.

Dated:

Authors Signatures:

_____

Shahmeer Abid

_____

Syeda Noreen Zahra

_____

Hamza Sheikh

Plagiarism Undertaking

We, Shahmeer Abid, Syeda Noreen Zahra, and Hamza Sheikh, solemnly declare that the work presented in the Final Year Project Report titled **"A Website for DoC, Hamdard University**" has been carried out solely by ourselves with no significant help from any other person except few of those which are duly acknowledged. We confirm that no portion of our report has been plagiarized and any material used in the report from other sources is properly referenced.

Dated:

Authors Signatures:

_____
Shahmeer Abid

_____
Syeda Noreen Zahra

_____
Hamza Sheikh

Acknowledgments

We sincerely thank Allah Almighty for his direction and support, whose grace and blessings enabled us to finish this project.

We would like to express our sincere gratitude to **Mr. Afzaal Hussain**, our project adviser, for their wise counsel, unwavering support, and priceless contributions during this project. Our success has been greatly attributed to their knowledge and guidance.

For giving us the opportunity and resources to work on this project, we are especially appreciative of the Hamdard Institute of Engineering and Technology. Our success has been greatly aided by the staff and faculty members' expertise and support.

Sincere gratitude is extended to our families, who have consistently provided love, tolerance, and inspiration.

They have been our pillar of support during this journey.

Lastly, we would like to express our gratitude to our friends and coworkers for their support, encouragement, and helpful criticism, all of which have enhanced our project and made this experience genuinely unforgettable.

We are grateful to everyone who helped us along the way.

Document Information

Table 1: Document Information

Customer	Department of Computing, Hamdard University
Project Title	A website for DoC, Hamdard University
Document	Final Year Project Report
Document Version	1.0
Identifier	FYP-001/FL24 Final Report
Status	Final
Author(s)	Syeda Noreen Zahra, Shahmeer Abid, Hamza Shiekh
Approver(s)	Afzaal Hussain
Issue Date	

Definition of Terms, Acronyms, and Abbreviations

This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.

Table 2: Definition of Terms, Acronyms, and Abbreviations

Term	Description
DoC	Department of Computing
LMS	Learning Management System
Dynamic Website	A website that updates its content dynamically based on user interaction or data changes.
Chatbot	An AI-based tool designed to provide automated responses to frequently asked questions.
Timetable	A schedule of classes and activities organized by section and semester.
Database	An organized collection of data stored electronically for easy access, management, and retrieval.
SQL	Structured Query Language, used for managing and querying relational databases.
Prototype Methodology	A software development approach focused on iterative refinement based on user feedback.
Admission Form	A form used to collect details from prospective students applying to the department.
Complaint Box	A feature allowing users to submit feedback or report issues for resolution.
Frontend	The part of a website that users interact with directly, including design and layout.
Backend	The server-side logic of a website that handles data storage, processing, and functionality.

Abstract

This project aims to develop a dynamic and user-friendly website for the Department of Computing (DOC). The primary goal is to create a centralized online platform that enhances communication, resource accessibility, and information dissemination for students, faculty, and prospective students. The website will feature sub-pages dedicated to various disciplines, including Computer Science, Software Engineering, and Artificial Intelligence, providing detailed information relevant to each field. Key functionalities include an integrated admission form, a complaint submission system, section-wise timetables, event announcements, and a searchable library book database. The project employs an evolutionary prototype methodology, allowing for iterative development and refinement based on user feedback. The scope of the project is confined to web development, with a focus on creating a robust, accessible, and efficient platform tailored to the needs of the DOC community.

Keywords:

- Department of Computing (DoC)
- Dynamic Website
- Centralized Platform
- Integrated Admission Form
- Complaint Box
- Section-Wise Timetables
- Searchable Library Database
- User-Friendly Interface
- Evolutionary Prototyping
- Chatbot Integration
- Faculty Profiles
- Academic Resource Accessibility
- Higher Education Technology
- Student Engagement
- Transparent Communication

Table of Contents

List of Figures

Figure No **Description** **Page No.**

List of Tables

CHAPTER 1 INTRODUCTION

1.1 Motivation

The necessity for a contemporary, coordinated, and easily navigable online platform to tackle a number of current issues is what drove the creation of the Department of Computing website. The current procedures for filing complaints, getting departmental information, and keeping track of occurrences are frequently disjointed and ineffective. A centralized website provides the following incentives:

Improved Communication: By offering a single platform with easily accessible updates, events, and significant announcements, the project seeks to enhance communication between students, faculty, and administrative personnel.

Transparency and Inclusivity: The website promotes transparency and guarantees that students can quickly get the information they require for academic assistance and support by offering comprehensive profiles of teachers, batch advisers, and program coordinators.

Community Building: Students, instructors, and staff feel more connected to one another and participate more actively when departmental announcements and updates are centrally located.

Adaptability and Growth: Using an evolutionary prototype process, the website is made to change in response to user feedback. This guarantees that it will continue to be applicable and satisfy the needs of its users in the future.

1.2 Problem Statement

Students, instructors, and administrative staff face a number of difficulties as a result of the Department of Computing's lack of a consolidated and dynamic web platform. Important concerns include:

Ineffective Communication: There are communication gaps as a result of the current channels for sharing information, such as announcements, event updates, or departmental alerts, being dispersed and unavailable to many.

Organization of Events and Schedules: Ineffective planning and communication of departmental events and class schedules causes misunderstandings and lowers involvement in extracurricular and academic activities.

Societies Page: No information is available about societies of department of computing. This website give Information about societies and recent images of recent events.

Lack of Transparency: The lack of comprehensive and freely available departmental and faculty information impedes transparency and deters staff and student participation.

Goals and Objectives

Goals:

Creating a dynamic, centralized, and user-friendly website for the Department of Computing is the main objective of this project. This website seeks to improve communication, expedite procedures, and offer a comprehensive platform for faculty, students, and potential students to access vital information and resources.

Objectives:

Create a Centralized Platform: Develop a website that consolidates departmental information, resources, and updates in one accessible location.

Streamline Admission Processes: Introduce an integrated admission form to simplify the application process for prospective students and reduce administrative workload.

Improve the Accessibility of Resources: Give comprehensive details about the program directors, batch advisors, and faculty members.

Establish a section for announcements, news, and events to notify professors and students about departmental activities. To help users with commonly asked questions, implement a chatbot.

Encourage Academic Pursuits: Give departmental profiles that are current professors and staff to promote transparency and ease of access.

1.3 Project Scope

This project's objective is to create a dynamic and intuitive website for the Department of Computing (DoC). The website will act as a single location for academic staff, administrative personnel, and students to obtain necessary materials, keep up to date, and expedite departmental procedures.

Included:

Sub-Webpages for Particular Fields of Study: Provide thorough explanations of computer science, software engineering, and artificial intelligence in separate sections.

Area for Events and Announcements: To enhance involvement and communication, centralize departmental news, events, and announcements.

Comprehensive Departmental Information: Provide general departmental information as well as dynamic profiles of the faculty, coordinators, and batch advisors.

Societies Page: Give Information about societies and recent images of recent events.

Integration of Chatbots: Implement a chatbot to help users with often asked questions and offer prompt assistance.

Limitations:

Advanced Data Analytics: No sophisticated reporting tools or data analysis features will be included in the project.

Mobile Application: A mobile application is not currently included in the scope of the website; it will only be optimized for web platforms.

Third-Party Integrations: Social media platforms, CMS, and LMS integration are examples of external tools that the system will not support.

Real-Time Updates: Real-time synchronization with internal systems is outside the purview of this project; website updates will be completed manually.

Assumptions:

Evolutionary Prototyping: Using an iterative process, features will be improved over time in response to user input.

CHAPTER 2

RELEVANT BACKGROUND & DEFINITIONS

Background

In fields like computer science, software engineering, and artificial intelligence, the Department of Computing (DoC) is essential to promoting scholarly advancement and creativity. However, the lack of a unified online platform leads to inefficiencies in information sharing, communication, and resource accessibility. The suggested project aims to create a dynamic and user-friendly website in order to address these issues.

With features like a complaint box, section-specific schedules, searchable library databases, and an integrated admission form, the website will work as a comprehensive center for departmental operations. These features are intended to improve user involvement, expedite procedures, and foster transparency for students, instructors, and potential applicants.

Definitions

Dynamic website:
A dynamic website is one that, in contrast to a static website with fixed content, automatically updates or modifies its content in response to user interaction or input.

Centralized platform:
A centralized platform is a single online place that compiles different tools, resources, and data for simpler management and access by several stakeholders.

Faculty profiles:
Faculty profiles provide comprehensive academic and professional details on faculty members, such as their jobs, credentials, and specializations.

Evolutionary prototyping:
Evolutionary prototyping is a software development process in which user feedback directs feature refinement and enhancements as the project progresses through iterative cycles.

User feedback Iteration:
After every development cycle, user feedback is gathered in order to improve and increase system functionality in response to real-world requirements.

Chatbot:
A chatbot is tool that instantly responds to commonly requested queries, enhancing user experience by speeding up response times and enabling self-service.

CHAPTER 3

LITERATURE REVIEW & RELATED WORK

Literature Review

The importance of centralized digital platforms in academic institutions has been extensively studied in recent years. Research highlights the following:

Centralized Information Systems: Centralized websites improve the accessibility and organization of resources for students and faculty. Studies emphasize the role of such platforms in enhancing communication, reducing administrative workload, and improving user experience (Abdullah et al., 2020).
User-centered design (UCD) is a design methodology that prioritizes the needs, preferences, and limitations of end users throughout the design and development process of digital systems. In the context of enhancing accessibility and usability, UCD plays a critical role in ensuring that digital platforms are not only functional but also intuitive and accessible to a diverse range of users, including those with disabilities, varying levels of digital literacy, and different cultural backgrounds. The approach focuses on iterative testing, user feedback, and continuous refinement to create systems that offer seamless, effective, and enjoyable user experiences (Lawrence Emma, 2024)

Evolutionary Prototyping: Iterative development methodologies, such as evolutionary prototyping, allow for gradual improvements based on user feedback, ensuring that the end product meets actual user needs (Brown, 2020).

Digital Resource Accessibility: Effective platforms provide easy access to essential academic resources, such as library databases and timetables, fostering a productive academic environment (Lee & Tan, 2018).

Related Work

Several universities and institutions have implemented similar centralized platforms to enhance academic processes:

University of Lahore (UoL): The UoL website provides dedicated sections for departmental information and announcements, serving as an essential resource hub for students and faculty. However, it lacks features like complaint boxes and dynamic faculty profiles.

Kohat University of Science and Technology (KUST): The KUST website includes an "Institute of Computing" page that provides program details and faculty profiles. However, it lacks integrated admission forms and chatbots, which limits its usability.

Air University, Islamabad: Air University's Faculty of Computing and AI platform includes faculty details and a limited events section. However, the absence of centralized student resources, such as timetables and library databases, reduces its effectiveness.

IBA Karachi: IBA Karachi's undergraduate program page is well-structured but primari

focused on admissions and program details. It does not cater to the daily academic and administrative needs of enrolled students.

University of Sindh, Jamshoro: The website offers a basic interface for departmental information but lacks advanced features like dynamic updates, complaint handling, and faculty interaction tools.

Gap Analysis

While several institutions have implemented departmental websites, significant gaps remain:

Lack of Integrated Features: Existing platforms are often fragmented, focusing on either admissions or program information, with minimal attention to features like complaint resolution, event management, or resource databases.

Limited User Engagement Tools: Features like chatbots, which can provide immediate assistance, and centralized announcement areas are missing in many platforms.

Static Interfaces: Most platforms lack dynamic content updates, resulting in outdated information and reduced utility for users.

CHAPTER 4
PROJECT DISCUSSION

1. Software Engineering Methodology

1.1 **Object-Oriented Design (OOD)**

The project adopts the Object-Oriented Design (OOD) methodology, which emphasizes:

- Modularity
- Reusability
- Scalability
- Maintainability

The project will utilize an Object-Oriented Design (OOD) methodology to ensure scalability, maintainability, and modularity, aligning with industry standards for web development.

2. Project Methodology

2.1 Evolutionary Prototyping

The development follows the Evolutionary Prototyping Model, where the system is built incrementally through continuous user feedback and refinement. The use of an evolutionary prototyping methodology means features will be developed and refined iteratively.

3. Phases of Project

3.1 Requirement Analysis

- Gathering functional and non-functional requirements.
- Identifying system users and use cases.

3.2 System Design

- ER diagrams and database modeling.
- UI/UX mockups.
- Defining system architecture and components.

3.3 Prototyping and Iteration

- Building initial versions of the system.
- Gathering feedback from stakeholders.
- Refining features iteratively.

3.4 Implementation

- Coding frontend using React.js.
- Backend development using Node.js
- Integration of database and REST APIs.

3.5 Testing & Evaluation

- Functional testing of chatbot, login credentials.
- Performance and security testing.
- User feedback sessions.

3.6 Deployment & Maintenance

- Hosting on cloud or university-maintained servers.
- Ongoing content updates by department web administrator.
- Regular backups and updates.

4. Software/Tools that Used in Project

4.1 Frontend Development

- HTML, CSS and Javascript – Main languages for building user interfaces.
- Bootstrap – For styling and responsive design.

4.2 Backend Development

- Node.js – JavaScript-based server-side environment.
- Django (alternative) – Python-based backend framework.

4.3 Database

- PostgreSQL – For relational and structured data (faculty, timetables).
- MongoDB – For unstructured/semi-structured data (chatbot queries, logs).

4.4 Development Environment

- VS Code – Source code editor.
- Git – (implied) for version control and team collaboration.

4.5 Integration and Communication

- REST APIs – For data exchange with chatbot and university systems.
- SMTP (Email Server) – For automated form and complaint notifications.

4.6 Supported Platforms

- Browsers: Chrome, Firefox, Edge, Safari
- Devices: Desktop, Tablet, Mobile

Hardware Requirement: Computers/Laptops and internet for web developer

Chapter 5
IMPLEMENTATION

5.1 Proposed System Architecture/Design

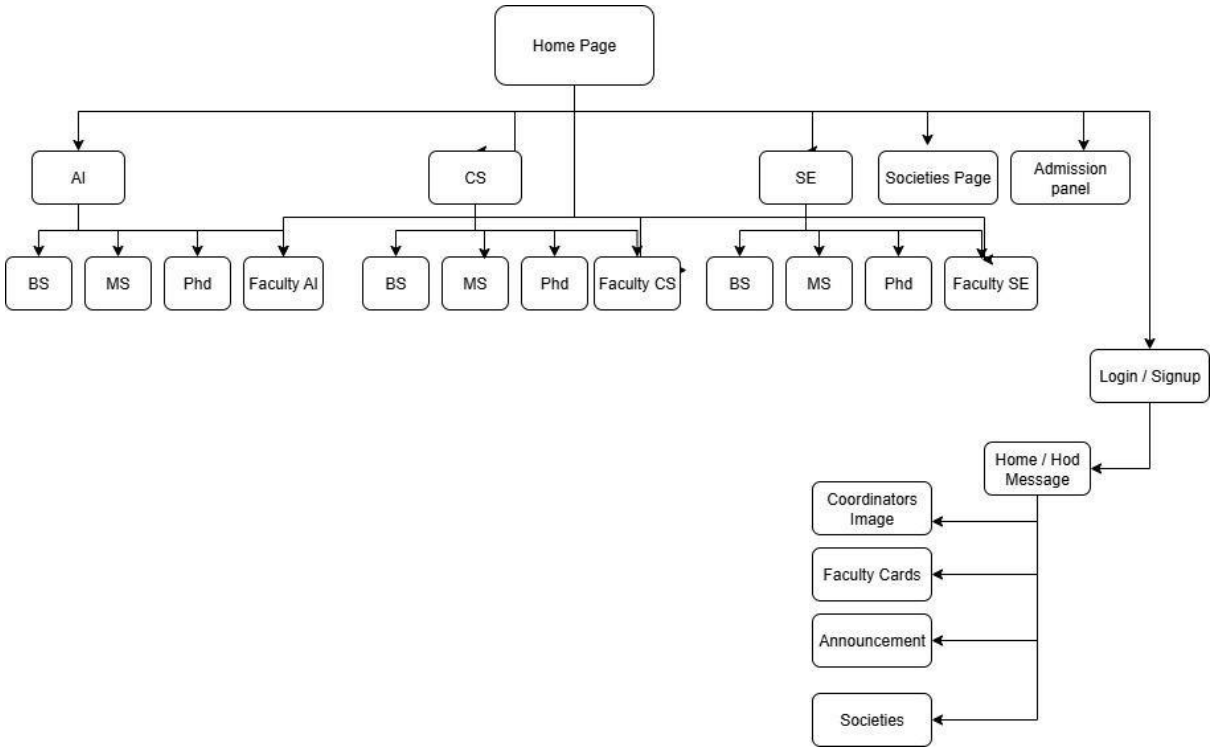


Figure 5.1

Figure 5.1 :The structure shows a hierarchical website design, organizing content based on academic departments and roles. It's designed for easy navigation from a central Home Page and covers:

- Academic programs by department
- Faculty details
- Admission processes
- Societies
- Login-secured areas (with HoD messages, announcements, etc.)

5.2 Functional Specification

5.2.1 User Roles and Permissions

1. Home Page

- Display a banner with announcements (data fetched from Admin Panel).
- Show HOD’s message (uploaded via Admin Panel).

- Display 3 buttons/links:
  - Computer Science
  - Software Engineering
  - Artificial Intelligence
- Show Department Info (static content).
- Display Coordinator Images (admin-uploaded).

2. Announcements Module
- Admin can:
  - Add and delete announcements.
- Announcements should appear in a banner on the homepage.

3. HOD Message Module
- Admin (HOD) can:
  - Upload or delete HOD's message (title, description, image, name).
- Display HOD message on homepage.

4. Coordinator Section
- Admin can:
  - Upload or delete images of program coordinators.

5. Programs Pages

Each program (CS, SE, AI) has its own page, with the following sub-pages:
- BS
- MS
- PhD
- Faculty

6. Faculty Page
- Admin can:
  - Add or delete faculty member details and photos.
  - Users can view faculty profiles by program (CS, SE, AI).

7. Society Page
- View department societies and their details.
- Admin can: o Upload society images of past events.

8. Authentication System
- Register: New HOD/Admin can register.
- Login: Admin authenticates with id and password.
- On successful login:
  - Redirect to Admin Panel to manage content.

9. Admin Panel
- Admin can:

- Manage Announcements
- Upload HOD Message
- Upload Coordinator Image
- Manage Faculty Profiles
- Upload Society Event Pics

10. Chatbot (Hamdard Bot)

- Answer common FAQs about Hamdard University.

5.3 Non-Functional Specifications

These define how the system performs under constraints and expected quality levels.

1. Usability

- User-friendly and intuitive UI/UX for both frontend users and admin.
- Responsive design for mobile, tablet, and desktop.

2. Reliability

- Website should load without errors and handle incorrect data gracefully.
- Admin actions should not cause crashes or data loss.

3. Security

- Admin panel protected with login credentials.
- Passwords stored securely.

4. Performance

- Fast loading for pages.
- Optimized images for faculty, and events.

5. Maintainability

- Code should be modular and well-documented.
- Future developers can easily update or scale modules.

6. Availability

- Website should have >99% uptime once hosted.
- Pages and chatbot should work even if some data is not present. 7. Scalability
- Can easily add new programs or extend faculty without major structure changes.
- Can later integrate real-time chat, course listings, or student portals. 8. Compatibility
- Supported on all modern browsers (Chrome, Firefox, Edge).
- Works on Android, iOS, and desktop OS platforms.

5.4 Testing

Testing is a critical phase in the development of the Department of Computing Website to ensure that all functionalities work as intended and meet the specified requirements. Various testing strategies were applied throughout the development lifecycle to identify bugs, usability issues, and integration problems.

Both **manual and functional testing** were conducted on the frontend and backend. All major features such as the announcement banner, HOD message, faculty data display, admin panel uploads, login system, and chatbot integration were thoroughly tested. Edge cases, invalid inputs, and unexpected user interactions were also considered to ensure stability and robustness.

The testing process involved:

- **Unit Testing:** Verifying individual functions such as data fetch, form submissions, and dynamic rendering.
- **Integration Testing:** Ensuring that different modules (e.g., admin uploads + frontend display) work together properly.
- **User Interface Testing:** Checking layout consistency, responsiveness, and overall user experience on different devices.
- **Database Testing:** Validating data storage, retrieval, and integrity within SQLite.
- **Security Testing:** Ensuring password protection and safe handling of user inputs on login and admin panel.

5.5 Purpose of Testing

The purpose of testing is to verify that the system performs accurately, reliably, and securely under all expected conditions. It ensures the product meets both functional and non-functional requirements before deployment.

Specifically, the goals of testing in this project were to:

- **Detect and fix errors or bugs** in code functionality or logic.
- **Validate feature correctness**, such as announcement updates reflecting instantly on the homepage.
- **Ensure data integrity** when interacting with the SQLite database.
- **Improve usability and user experience** by identifying and resolving UI/UX issues.
- **Verify responsiveness and cross-browser compatibility** of the interface.
- **Ensure secure access** to sensitive features like the admin panel.

5.6 Test Cases

Test Case 1

Title: HOD Register

Precondition	Show Treatment
Actions	Register HOD
Expected Result	Password mailed to embedded email
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 2

Title: Coordinator Register

Precondition	Show Treatment
Actions	Register Coordinator
Expected Result	Password mailed to HOD
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 3

Title: HOD Login

Precondition	Show Treatment
Actions	Login HOD
Expected Result	Login to HOD admin panel
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 4

Title: Coordinator Login

Precondition	Show Treatment
Actions	Login Coordinator
Expected Result	Login to Coordinator admin panel
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 5

Title: Add HOD Message

Precondition	Show Treatment
Actions	HOD Adds HOD Message
Expected Result	Added to home page
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 6

Title: HOD Announcement

Precondition	Show Treatment
Actions	HOD make announcement
Expected Result	Show on announcement banner
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 7

Title: Upload Coordinator Images

Precondition	Show Treatment
Actions	HOD upload Coordinator images
Expected Result	Showed on home page
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 8

Title: Add faculty cards

Precondition	Show Treatment
Actions	HOD add faculty cards
Expected Result	Card viewed on specific Faculty page
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 9

Title: Upload society event images

Precondition	Show Treatment
Actions	HOD add event images
Expected Result	Uploaded on specific society's images section
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 10

Title: Coordinator announcemet

Precondition	Show Treatment
Actions	Coordinator make announcement
Expected Result	Show on announcement banner
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 11

Title: Coordinator add events

Precondition	Show Treatment
Actions	Coordinator add event images
Expected Result	Uploaded on specific society's images section
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 12

Title: Coordinator add faculty cards

Precondition	Show Treatment
Actions	Coordinator add faculty cards
Expected Result	Uploaded on specific faculty's page
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 13

Title: Coordinator and HOD Logout button

Precondition	Show Treatment
Actions	Coordinators and HOD logout
Expected Result	Redirected to Login page
Tested by	Shahmeer Abid
Result	Verification Successful

‘

Chapter 6

EXPERIMENTAL EVALUATIONS & RESULTS

Evaluation

The Department of Computing website was evaluated based on functionality, usability, responsiveness, performance, and maintainability. The goal was to ensure that all features including the announcement system, HOD message management, faculty and coordinator data display, admin panel operations, chatbot integration, and program-wise pages met user expectations and technical requirements.

Key aspects of evaluation included:

- Successful data flow from admin panel to the frontend.
- Page responsiveness across devices and screen sizes.
- Correct retrieval and display of dynamic content.
- Security validation of login and admin access.
- Overall ease of navigation for both admin and public users.

Testbed

The test environment (testbed) was set up to simulate real-world usage conditions. It included hardware, software, and test data required for validating the system.

Hardware Used:

- Laptop with Intel Core i5 Processor
- 8GB RAM
- Windows 10 Operating System

Software Stack:

- **Frontend:** HTML5, CSS3, Bootstrap, JavaScript (ES6)
- **Backend:** Node.js (Express.js)
- **Database:** SQLite
- **Chatbot:** Chatbase (embedded via script)
- **Version Control:** Git and GitHub
- **Browser Compatibility Testing:** Chrome, Firefox, Microsoft Edge

Test Data:

- Sample announcements, coordinator names and photos, faculty profiles, event images, and dummy login credentials were used to test the platform functionalities.

Results and Discussion

Results

- All core functionalities performed as expected.
- The admin panel successfully allowed the HOD to upload announcements, faculty data, coordinator images, and society content.
- The chatbot accurately responded to common questions based on uploaded data.
- Pages were fully responsive across desktop and mobile devices.
- SQLite database handled read/write operations correctly without data loss.
- Login and access control worked properly, preventing unauthorized entry.

Discussion:

The system met its intended objectives of centralizing departmental information, improving communication, and providing dynamic content management via a secure and easy-to-use admin panel. The integration of Chatbase provided additional value by automating responses to common queries.

However, minor UI enhancements (such as better image preview on uploads) were identified as areas for improvement.

CHAPTER 7

CONCLUSION AND DISCUSSION

7.1 Strength of this Project

The Department of Computing Website offers several key strengths that make it an effective and practical solution for both faculty and students:

- **Dynamic Admin Panel:** Allows the HOD to manage content (announcements, HOD message, faculty data, coordinator images, society info) without needing developer support.
- **User-Friendly Interface:** Clean and responsive design ensures that users can easily access departmental information on all devices.
- **Centralized Information Hub:** Combines all essential data—program details, faculty profiles, events, and announcements—on one platform for easier navigation.
- **Efficient Communication:** Updates and announcements are visible on the homepage in real-time, improving communication between administration and students.
- **Chatbot Integration:** A chatbot developed using Chatbase provides automated assistance to users, reducing common query traffic to the administration.
- **Secure Login System:** Only authorized personnel can access the admin panel, ensuring data protection.

7.2 Limitations and Future Work

Limitations:

- The **chatbot does not support real-time data** and only responds based on the predefined dataset uploaded to Chatbase.
- No **file validation or preview functionality** for images uploaded through the admin panel.
- **Limited role-based access control** only a single admin user is supported (e.g., HOD).
- **No real-time database sync** or live feedback system for students.

Future Work:

- Implement **multi-role admin access**, allowing faculty coordinators and society heads to manage their sections.
- Integrate **file validation and image preview** before uploads.
- Enhance chatbot capabilities to support **live chat** or real-time query routing to departments.
- Add modules for **semester schedules**, **course registration**, or **student feedback forms**.
- Include **analytics dashboards** for tracking website activity or user engagement.

7.3 Reasons for Failure – If Any

There were **no critical failures** encountered during the project development and testing phases. However, a few **challenges** were faced that were resolved during the process:

- Initially, **issues with dynamic data rendering** from the admin panel were observed due to incorrect database connections, but they were fixed after testing SQLite queries.
- A few **bugs in image path rendering and form validation** occurred, especially during upload operations, which were resolved by refining the backend logic.
- **Chatbot embedding limitations** due to Chatbase script configuration delayed integration slightly but were eventually overcome.

REFERENCES

[1]
https://uol.edu.pk/

[2]
https://admissions.kohat.edu.pk/ApplyOnline

[3]
https://www.au.edu.pk/

[4]
https://www.iba.edu.pk/

APPENDICES

List of Appendices

A1a. Project Proposal and Vision Document
A1b. Copy of Proposal Evaluation Comments by Jury
A2. Requirement Specifications
A3. Design Specifications
A4. Other Technical Details
Test cases
UI/UX Details
Coding Standards
Project Policy
A5. Flyer & Poster Design
A6. Copy of Evaluation Comments
Copy of Evaluation Comments by Supervisor for Project – I Mid Semester Evaluation
Copy of Evaluation Comments by Jury for Project – I End Semester Evaluation
Copy of Evaluation Comments by Supervisor for Project – II Mid Semester Evaluation
Copy of Evaluation Comments by Jury for Project – II Mid Semester Evaluation
Copy of Evaluation Comments by Jury for Project – II End Semester Evaluation
A7. Meetings' Minutes

A1A. PROJECT PROPOSAL AND VISION DOCUMENT

Any standard template may be used, as per project need approved by Project Coordinator & Supervisor. Following is a suggestive outline. **Also, the same outline should be used for Project Proposal Presentation.**

1 Introduction

The proposed project is a dynamic website for Department of Computing (DOC) FEST. The main aim of creating this website is to create an online centralized platform that will enhance communication, resource accessibility and will help students, faculty, and future students who are willing to take admission to get information about the department and stay updated. This website will have sub-pages of faculty of Computer Science, Software Engineering and Artificial Intelligence. Each will provide the detailed information relevant to their discipline and domain. As key features of this website we have added an integrated admission form, a complaint box, section-wise timetables, events and announcement area. Additionally, the website will have all the detail about faculty members, batch advisors, program coordinators, departmental detail of computing department and library book listing. This project will create a straightforward, user friendly and easy to manage website.

1.1 Problem Statement

The main aim of creating this website is to create an online centralized platform that will enhance communication, resource accessibility and will help students, faculty, and future students who are willing to take admission to get information about the department and stay updated.

- Events, News Announcement Area: A centralized area for departmental updates, upcoming events and important announcements, keeping the community informed, engaged and will promote participation in events.

- Detailed Departmental, Faculty and other relevant Information: It will facilitate easy access to crucial information for students and staff also. And will promote transparency and accessibility.

- Faculty Information: Provide comprehensive information about the faculty.

- Chat bot: A chat bot to assist users with frequently asked questions and provide quick information.

1.2 Project Motivation

The necessity for a contemporary, coordinated, and easily navigable online platform to tackle a number of current issues is what drove the creation of the Department of Computing website. The current procedures for filing complaints, getting departmental information, and keeping track of occurrences are frequently disjointed and ineffective. A centralized website provides the following incentives:

Improved Communication: By offering a single platform with easily accessible updates, events, and significant announcements, the project seeks to enhance communication between students, faculty, and administrative personnel.

Transparency and Inclusivity: The website promotes transparency and guarantees that students can quickly get the information they require for academic assistance and support by offering comprehensive profiles of teachers, batch advisers, and program coordinators.

Community Building: Students, instructors, and staff feel more connected to one another and participate more actively when departmental announcements and updates are centrally located.

Adaptability and Growth: Using an evolutionary prototype process, the website is made to change in response to user feedback. This guarantees that it will continue to be applicable and satisfy the needs of its users in the future.

1.3 Objectives

To develop a dynamic and user-friendly website for the department of computing.

2 Project Vision

The goal of this project is to create a modern, user-friendly, centralized website for the Department of Computing (DoC) at Hamdard University. The main aim is to improve communication, accessibility, and efficiency for students, faculty, and staff by bringing essential academic and administrative features into one online platform.

This website will act as a digital hub for all departmental activities. It will provide access to faculty profiles, societies information, announcements. It aims to end the reliance on manual processes and scattered information, replacing them with an automated system that is available 24/7 and can be accessed from any device.

With role-based access control, and chatbot integration, the platform will provide timely support, allow users to submit feedback, and keep them updated on academic news. The goal is not only to improve the daily experiences of students and faculty but also to create a digital model that other departments in the university can replicate.

3 Project Scope

- Home page
- Sub Webpages for specific faculties
- Detailed Departmental Information
- Faculty Information
- Events, News Announcement area

- Faculty login
- Chat bot

3.1 In Scope

Sub Webpages for specific disciplines: Web-pages for faculty of computer science, Software Engineering and Artificial intelligence.
• Section wise Timetable: Uploading semester wise timetable for easy student access.
• Events, News Announcement area: A centralized area for departmental updates, events and announcements.
. • Detailed Departmental Information: Dynamic profiles of faculty members, batch advisors, coordinators and other relevant faculty personnel
• Faculty Information: Comprehensive profiles of faculty members. • FYP Forms: Forms related to Final Year Projects.
• Chatbot: A chatbot to assist users with frequently asked questions and provide quick information.

3.2 Out of Scope

• Advance data Analysis: The project will not include some complex data analysis or reporting tools.
• Mobile Application: The focus will be on web platform and we will not develop mobile app.
• Third-party Integration: Integration with external System for example, CMS, LMS or social media are not in the current scope.
• Real-Time Update: The website will be updated regularly, real-time updates will not be implemented at this stage.

4 Proposed Methodology

The Evolutionary prototype methodology is used for developing the Department of Computing website, by using this methodology we will approach each functionality repeatedly, will make changes according to user feedback and improving overtime.

4.1 Team Role & responsibilities

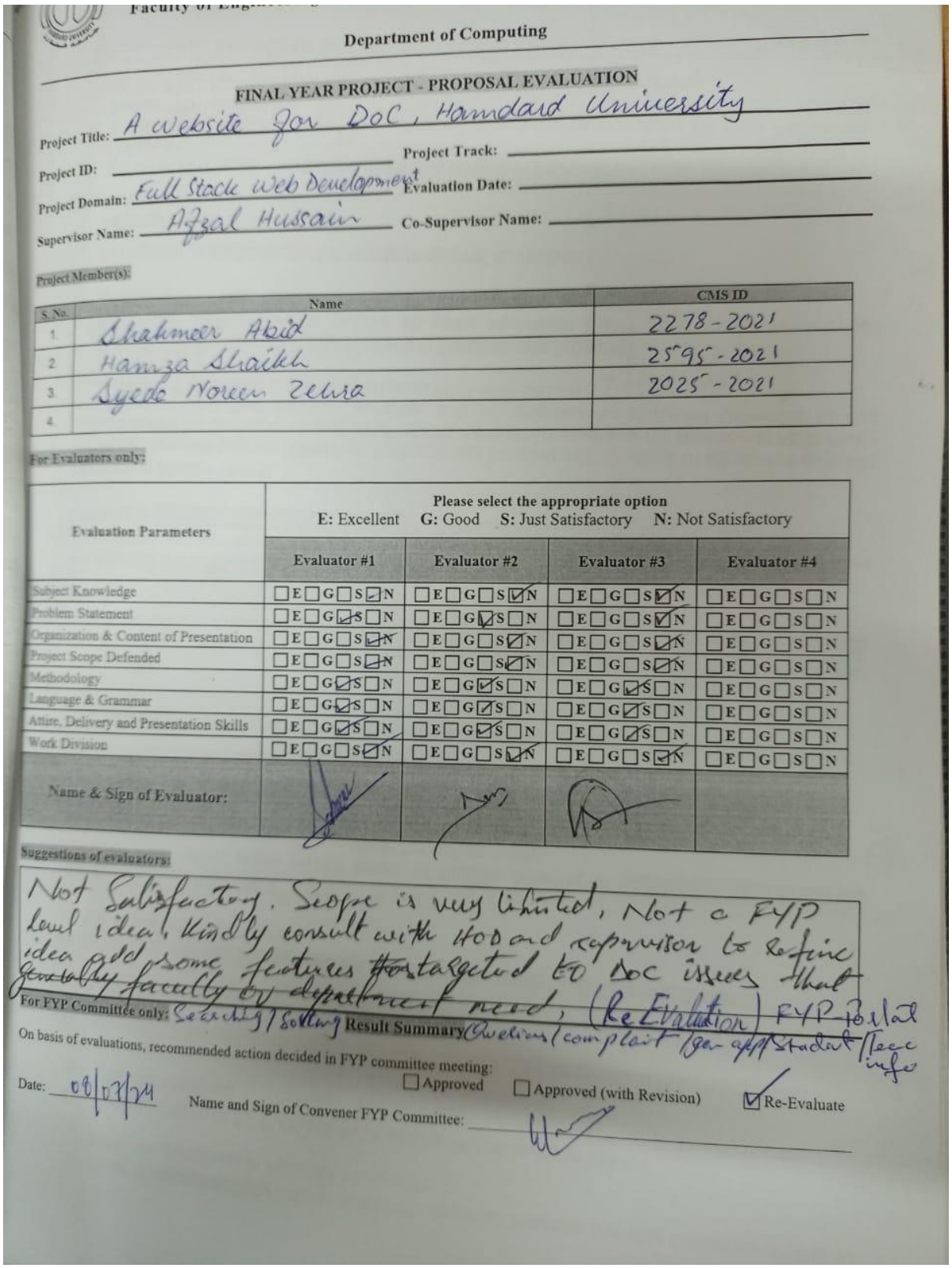
Responsibilty/Task	Shahmeer Abid	Noreen Zehra	Hamza Shaikh	Mr Afzal Hussain
Requirement Gathering	R/A	R/A	R/A	C/I
Analyses	C	R/A	I	C/I
Resource Planning	I	C	R/A	C/I
Frontend Development	R/A	I	C	C/I
Database Management	C	R/A	I	C/I
Backend Development	I	C	R/A	C/I
Testing	R/A	I	C	C/I
Documentation	A	R	C	C/I
Finalize	R/A	R/A	R/A	C/I

5 Project Planning

5.1 Gantt Chart

Tasks / Responsibilities	Assigned To	Start Date	End Date	July	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	June
Proposal Defense	Shahmeer,Noreen,Hamza	July 2024	July 24												
Requirement Analysis and Gathering	Shahmeer,Noreen,Hamza	Aug 2024	Aug 24												
Project Planning	Shahmeer,Noreen,Hamza	Sep 2024	Sep 24												
Task Allocation	Shahmeer,Noreen,Hamza	Oct 2024	Oct 24												
Development	Shahmeer	Nov 2024	Nov 24												
Development	Hamza	Nov 2024	Nov 24												
Development	Noreen	Dec 2024	Dec 24												
Initial Testing	Shahmeer,Noreen,Hamza	Dec 2024	Dec 24												
FYP Report-1	Shahmeer,Noreen,Hamza	Jan 2024	Jan 24												
Development	Shahmeer	Feb 2024	Feb 24												
Development	Hamza	Mar 2025	Mar 25												
Development	Noreen	Mar 2025	Mar 25												
Performance Testing	Shahmeer	Apr 2025	Apr 25												
Documentation	Shahmeer,Noreen,Hamza	July 2024	May 25												
Final Project Report and Presentation	Shahmeer,Noreen,Hamza	June 2025	June 25												

A1B. COPY OF PROPOSAL EVALUATION COMMENTS BY JURY



Faculty of Engineering

Department of Computing

FINAL YEAR PROJECT - PROPOSAL EVALUATION

Project Title: A website for DoC, Hamdard University

Project ID: _____ Project Track: _____

Project Domain: Full Stack Web Development Evaluation Date: _____

Supervisor Name: Afzal Hussain Co-Supervisor Name: _____

Project Member(s):

S. No.	Name	CMS ID
1.	Shahmeer Abid	2278-2021
2.	Hamza Shaikh	2595-2021
3.	Syeda Noreen Zehra	2025-2021
4.		

For Evaluators only:

Evaluation Parameters	Please select the appropriate option E: Excellent G: Good S: Just Satisfactory N: Not Satisfactory			
	Evaluator #1	Evaluator #2	Evaluator #3	Evaluator #4
Subject Knowledge	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☐N
Problem Statement	☐E ☐G ☑S ☐N	☐E ☐G ☑S ☐N	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☐N
Organization & Content of Presentation	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☐N
Project Scope Defended	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☐N
Methodology	☐E ☐G ☑S ☐N	☐E ☐G ☑S ☐N	☐E ☐G ☑S ☐N	☐E ☐G ☐S ☐N
Language & Grammar	☐E ☐G ☑S ☐N	☐E ☐G ☑S ☐N	☐E ☐G ☑S ☐N	☐E ☐G ☐S ☐N
Attire, Delivery and Presentation Skills	☐E ☐G ☑S ☐N	☐E ☐G ☑S ☐N	☐E ☐G ☑S ☐N	☐E ☐G ☐S ☐N
Work Division	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☑N	☐E ☐G ☐S ☐N
Name & Sign of Evaluator:				

Suggestions of evaluators:

Not Satisfactory. Scope is very limited, Not a FYP level idea, Kindly consult with HOD and supervisor to refine idea add some features ~~for~~ targeted to DoC issues that generally faculty or department need, (Re Evaluation) FYP Portal

For FYP Committee only: Searching / Solving Result Summary (Queries / complaint / gen q/a / Student / Teac info

On basis of evaluations, recommended action decided in FYP committee meeting:

Date: 08/07/24 ☐ Approved ☐ Approved (with Revision) ☑ Re-Evaluate

Name and Sign of Convener FYP Committee: _____

A2. REQUIREMENT SPECIFICATIONS

1. Introduction

1.1 **Purpose of Document**

The purpose of this Software Requirements Specification (SRS) document is to describe everything needed to build a website for the Department of Computing at Hamdard University. This website will make it easy for students, faculty, and administrators to access information, and stay updated. Here's why this document is important:

1. **Clarify What Needs to Be Built**
   It explains all the features and tools the website will include, like subpages for programs (Computer Science, Artificial Engineering, and Software Engineer etc.), faculty details, timetables, announcements, and automated forms.

2. **Centralize All Requirements**
   It gathers all the technical and functional details in one place, ensuring everyone working on the project understands the goals and how the website should function.

3. Improve Communication The website will allow:
   - Students to access information easily.
   - Faculty to make announcements and manage tasks efficiently.
   - A chatbot to answer user questions.

4. **Guide the Technical Team**
   It specifies the tools, software, and hardware required to build the website, ensuring the development team knows what technology to use.

5. **Define Roles and Responsibilities**
   It documents who can do what on the website (e.g., only the HOD can post announcements, event images, coordinator images, etc).

1.2 **Intended Audience**

The purpose of this project is to:

1. Prospective students interested in the Department of Computing and its sub-disciplines.

   Visitors seeking information computing department.

1.3 Abbreviations

DoC	Department of Computing
FYP	Final Year Project
Dynamic Website	A website that updates its content dynamically based on user interaction or data changes.
Chatbot	chatbot that operates on a predefined set of rules, scripts, or decision trees to interact with users
Timetable	A schedule of classes and activities organized by section and semester.
Database	An organized collection of data stored electronically for easy access, management, and retrieval.
SQL lite	Structured Query Language, used for managing and querying relational databases.
Prototype Methodology	A software development approach focused on iterative refinement based on user feedback.
Frontend	The part of a website that users interact with directly, including design and layout.
Backend	The server-side logic of a website that handles data storage, processing, and functionality.

1.4 Project Background

The proposed project involves developing a dynamic and user-friendly website for the Department of Computing (DoC) at Hamdard University. The primary purpose is to create an online centralized platform that enhances communication, resource accessibility, and transparency for students, faculty, and prospective students.

Key Objectives:

1. Provide comprehensive departmental information, including faculty profiles, batch advisors, program coordinators, and library resources.
2. Introduce integrated features like:
    - A news and announcements area to keep the department updated.
    - A chatbot for addressing frequently asked questions.
    - Faculty pages for specific program faculties
    - Societies page for societies objective and latest event update

1.5 Problem Statement

Currently, the Department of Computing lacks a dynamic and user-friendly web platform to centralize and streamline the dissemination of departmental information. Important updates such as announcements, faculty profiles, program-specific details (BS, MS, PhD), and society activities are not easily accessible online, leading to reduced engagement and communication gaps among students, faculty, and administrative staff. Additionally, there is no structured system for administrators (like the HOD) to manage content independently. The absence of a chatbot to assist with common queries further adds to student confusion. This project aims to address these challenges by introducing a centralized website with an integrated admin panel, chatbot, and dynamically managed content to improve efficiency, transparency, and communication within the department.

1.6 **Project Scope**

1. **Department Overview Pages**: Comprehensive pages dedicated to Computer Science, and Software Engineering, Artificial Intelligence each detailing faculty members, coordinators, and the department head.
2. **Societies Page:** Give Information about societies and recent images of recent events.
3. **Announcement System**: A portal where only the HOD and Coordinator can log in to make announcements to students and faculty**.**
4. **Chatbot Integration**: A chatbot that provide responses to frequently asked questions and queries, leveraging the university's existing information.

1.7 **Not In Scope**

1. Integration with external educational systems beyond Hamdard University.
2. Non-academic departments (e.g., administration, non-computing departments).
3. Mobile app version of the website (only the web-based version will be developed).
4. Advanced AI features beyond the chatbot for providing automatic suggestions or handling dynamic data interpretation.

1.8 **Project Objectives**

1. **Centralize Information**: Develop a website that serves as a single point of access for faculty details, events, coordinators announcements.
2. **Enhance Student Experience:** Integrate a chatbot to ensure students can easily find the information they need, like timetables, faculty c, and department-specific details.
3. **Ensure Accessibility:** Provide students, faculty, and staff easy access to department-specific pages and chatbot.

1.9 Stakeholders & Affected Groups

The proposed **Department of Computing (DoC) website** involves multiple stakeholders and user groups that will benefit from or interact with the platform. These include the following:

1. **Primary Stakeholders:** 1. **public (Current):**

- Access to detailed faculty and department information.
- Ease in finding and announcements.

2. **Program Coordinators:**

- Streamlined management of student records and departmental updates.
- Tools to upload, sort, and manage data like uploading faculty info event's images or announcements.

3. **Head of Department (HOD):**

- Tools for handling escalated announcements.
- Access to add or delete things from admin panel.

2. **Affected Groups:**

1. **Visitors and Alumni:**

o Prospective students and alumni will gain easy access to departmental information and resources, fostering engagement and trust in the department.

1.10 Operating Environment

The **Department of Computing (DoC) website** will operate in a secure and reliable environment to ensure smooth functionality for all users, including students, faculty, and administrative staff. Below is a detailed description of the operating environment:

1. **Web Hosting Environment:**

Platform:

The website will be hosted on university-maintained server to ensure scalability and reliability.

Operating System:

Windows for local development environments.

2. **User Devices:**

- **Supported Devices**:
  The website will be responsive and accessible via:
- Desktops and laptops (Windows, macOS, Linux).
- Mobile devices (iOS and Android).
- Tablets (all major platforms).

Browsers:

- Google Chrome
- Microsoft Edge

3. Network Requirements:

- **Local Access**:
  The website will be accessible via the university's LAN for administrative purposes.
- **Internet Access**:
  Public access for students, faculty, and prospective students from external locations using secure HTTPS protocols.
- **Bandwidth**:
  Moderate bandwidth usage to accommodate multiple users accessing resources like timetables, forms, and chatbot services simultaneously.

3. **Software:**

- **Frontend**:

  HTML, CSS, JavaScript, and frameworks like React.js or Bootstrap for a responsive and interactive UI.
- **Backend**:

  Node.js or Python for server-side logic.
- **Database**:

  SQL lite/PostgreSQL for managing records like timetables, library resources, and user submissions.

4. **Security Measures:**

- **Authentication**:

Role-based access for HOD, and Coordinator to ensure secure data management.

6. Maintenance and Updates:

The website will be managed by a designated web administrator from the department, who will handle updates for:

- HOD’s Message

- Announcements.
- Faculty information.
- Societies events
- Coordinators Images

1.11 System Constraints

1. Technical Constraints

- **Chatbot Dependency**:

  The chatbot relies on data from website. Inaccurate or outdated data on the website could affect chatbot responses.

2. Resource Constraints

- **Hardware Requirements**:

  The system requires modern computers or servers with sufficient processing power for hosting and data management. Limited access to these resources could delay deployment.

- **Human Resources**:

  Dependence on web developers, database administrators, and testers for development and maintenance may cause delays if skilled personnel are unavailable.

3. Security Constraints

- **Role-Based Access**:

  The system enforces role-based access for different stakeholders (e.g. HOD, coordinators).

  Mismanagement of access roles could compromise data integrity.

4. Development Constraints

- **Methodology**:

  The use of an evolutionary prototyping methodology means features will be developed and refined iteratively.

  This may delay the final delivery of certain functionalities.

5. Performance Constraints

- **Concurrent Users**:
  - The system is designed for moderate traffic and may require scaling for high user loads.
- **Internet Dependency**:

Internet access is required for users to access the website and for functionalities like email and chatbot integration.

1.12 Assumptions & Dependencies

1. Assumptions

- **Data Availability**: provided and maintained by the department.
- **User Feedback**:

  Regular feedback from students, faculty, and administrators will be available to improve the website using the evolutionary prototyping approach.
- **Moderate Traffic**:

  The system assumes a moderate number of concurrent users and is not designed for heavy traffic without scaling.
- **Device Compatibility**:

  Users will access the website through supported devices and modern browsers.
- **Administrative Roles**:

  Role-based access will be appropriately assigned and maintained to ensure system security and functionality.

2. Dependencies

- **Hosting Environment**

  Availability of a secure and reliable hosting environment (e.g., a university-maintained server hosting).
- **Development Tools and Frameworks**:

  The project relies on tools like VS Code, languages HTML, CSS and Javascript, framework Bootstrap or React.js (frontend), Node.js or Python (backend), and SQL lite databases for development and deployment.
- **Skilled Personnel**:

  Access to skilled developers, database administrators, and testers for smooth development and maintenance.
- **Internet Connectivity**:

  Reliable internet access for public users to interact with the website and its features (e.g., chatbot, announcements, form submissions).
- **Timely Inputs**:

  The project depends on timely inputs from stakeholders, such as faculty information, library data, and feedback from users.

2. External Interface Requirements

2.1 Hardware Interfaces

The Department of Computing (DoC) website will interact with various hardware systems to ensure smooth operation, secure access, and efficient functionality. Below are the details of the hardware interfaces:

1. **Web Hosting Server**

- **Purpose**:
  Hosts the website and backend systems for accessibility by students, faculty, and administrators.
- **Specifications**:
  - Processor: Quad-core or higher
  - RAM: Minimum 8 GB (16 GB recommended for scalability).
  - Storage: SSD with at least 500 GB for fast read/write operations.
  - Network: 1 Gbps Ethernet for high-speed connectivity.

• **Behavior**:
Ensures reliable uptime, supports multiple concurrent users, and processes requests efficiently.

2. **User Devices**

- **Purpose**:
  Provides access to the website for students, faculty, and staff.
- **Supported Devices**:
  - **Desktops and Laptops**: Windows.
  - o **Mobile Devices**: Android platforms.
  - **Tablets**: All major tablet devices.

• **Behavior**:
Responsive website design ensures compatibility and consistent user experience across devices.

3. **Administrative Systems • Purpose**:
Used by Coordinator, and HODs for tasks like image uploads, announcements, and faculty cards.

• **Specifications**:
  - Computers or laptops with updated operating systems (Windows 10/11,).

- o Minimum Hardware: Dual-core processor, 4 GB RAM, and 500 GB HDD/SSD.
- Peripherals: Internet access, keyboard, and mouse for input.

• **Behavior**:
Facilitates secure login for authorized personnel to access backend functionalities.

4. **Networking Hardware**

- **Purpose**:
  Supports communication between the website, database..
- **Components**:
  - **Routers and Switches**: Ensure stable connections within the LAN and to the internet.
  - **Firewall Hardware**: Protects the server from unauthorized access and cyber threats.

• **Behavior**:
Provides uninterrupted and secure connectivity for hosting and data transmission.

5. **Backup and Storage Systems**

- **Purpose**:
  Ensures data reliability and recovery in case of system failure.
- **Specifications**:
  - Network-attached storage (NAS) or cloud-based backup solutions.
  - Capacity: At least 1 TB storage for maintaining backups of website data, user records, and forms.

• **Behavior**:
Periodic backups prevent data loss and allow quick recovery during system failures.

2.2 Software Interfaces

The Department of Computing (DoC) website will interface with several software systems and applications to deliver its functionalities. Below are the details for each application:

1. **Information About DoC:**

Purpose:
To provide detailed information about the Department of Computing, including its academic programs (BS, MS, PhD), faculty details, departmental objectives, research areas, and student societies. This information serves students, faculty, and prospective applicants.

External Owner:
Department of Computing (DoC), Hamdard University

Interface Details:

- Presented on the website through dedicated web pages and sections (e.g., About Us, Programs, Faculty, Societies).
- Information is displayed using a user-friendly frontend interface.
- Data is fetched dynamically from the admin panel where applicable (e.g., faculty info, coordinators).
- Accessible through navigation menus for easy exploration by users.

2. **Chatbot**

- **Purpose**:
  Provides conversational assistance to users.
- **External Owner**:
  Department of Computing
- **Interface Details**:
  - **Integration**: chatbot platform.
  - **Behavior**: The chatbot pulls data from the DoC website to answer user queries.

3. **Administrative**

- **Purpose**:
  Allows authorized users to manage announcements, events, and news updates.
- **External Owner**:
  IT Team.
- **Interface Details**:
  - **Integration**: Internal admin panel linked to the website database.
  - **Behavior**: Restricted to authenticated users (e.g., HOD and coordinator) for making updates.

2.3 **Communications Interfaces**

The Department of Computing (DoC) website relies on various communication interfaces to connect with systems, devices, and users. These interfaces ensure secure and efficient data exchange and accessibility.

1. **Local Area Network (LAN)**

- **Purpose**:
  Facilitates secure internal communication for administrative tasks like timetable updates, announcements, and database management.
- **Details**:
  - **Protocol**:
    - HTTP/HTTPS for accessing the website on the intranet.
    - SQL for connecting to the local database.
  - **Behavior**:
    - Administrative staff and faculty can use the LAN to access backend systems for uploads and updates.
  - **Security**:
    - Firewalls and VLAN segmentation to isolate sensitive data and restrict unauthorized access.

2. **Internet Communication**

- **Purpose**:
  Provides public access to the website and supports external communications like email notifications and chatbot integration.

3. **Communication with Database Systems**

- **Purpose**:
  Enables storage and retrieval of information, such as faculty profiles, timetables, and library data.
- **Details**:
  - **Protocols**:
    - SQL for database queries.
  - **Behavior**:
    - Real-time retrieval of data for timetables, records, and library resources.
    - Admins use secure access to update information.
  - **Behavior**:

- Regular backups are sent to a remote or cloud-based storage system for disaster recovery.

5. **User Communication Devices**

- **Purpose**:
  Enables interaction between users (students, faculty, and staff) and the website.
  - **Behavior**:
    - Responsive design ensures compatibility across devices (desktops, laptops, tablets, and mobile phones).

6. **Networking Hardware Communication**

- **Purpose**:
  Maintains stable connections between systems and devices.

2.4 System Functions

Based on the project proposal for the Department of Computing website, here is an example of how system functions and attributes can be organized for the website:

1. Home Page

- Display a banner with announcements (data fetched from Admin Panel).
- Show HOD's message (uploaded via Admin Panel).
- Display 3 buttons/links:
  - Computer Science
  - Software Engineering
  - Artificial Intelligence
- Show Department Info (static content).
- Display Coordinator Images (admin-uploaded).

2. Announcements Module

- Admin can:
- Add and delete announcements.
- Announcements should appear in a banner on the homepage.

3. HOD Message Module

- Admin (HOD) can:

o Upload or delete HOD's message (title, description, image,

name).

- Display HOD message on homepage.

4. Coordinator Section

- Admin can:

o Upload or delete images of program

coordinators.

5. Programs Pages

Each program (CS, SE, AI) has its own page, with the following sub-pages:

- BS
- MS
- PhD
- Faculty

6. Faculty Page

- Admin can:
- o Add or delete faculty member details and photos.
- Users can view faculty profiles by program (CS, SE, AI).

7. Society Page

- View department societies and their details.
- Admin can:

o Upload society images of past events.

8. Authentication System

- Register: New HOD/Admin can register.
- Login: Admin authenticates with id and password.
- On successful login:

o Redirect to Admin Panel to manage content.

9. Admin Panel

- Admin can:

o Manage Announcements o

Upload HOD Message o

Upload Coordinator Images o

Manage Faculty Profiles o

Upload Society Event Pics

10. Chatbot (Hamdard Bot)

- Answer common FAQs about Hamdard University.

System Attributes / Nonfunctional Requirements

These define how the system performs under constraints and expected quality levels.

1. Usability

- User-friendly and intuitive UI/UX for both frontend users and admin.
- Responsive design for mobile, tablet, and desktop.

2. Reliability

- Website should load without errors and handle incorrect data gracefully.
- Admin actions should not cause crashes or data loss.

3. Security

- Admin panel protected with login credentials.
- Passwords stored securely.

4. Performance

- Fast loading for pages.
- Optimized images for faculty, and events.

5. Maintainability

- Code should be modular and well-documented.
- Future developers can easily update or scale modules.

6. Availability

- Website should have >99% uptime once hosted.
- Pages and chatbot should work even if some data is not present.

7. Scalability

- Can easily add new programs or extend faculty without major structure changes.
- Can later integrate real-time chat, course listings, or student portals.

8. Compatibility

- Supported on all modern browsers (Chrome, Edge).
- Works on Android, iOS, and desktop OS platforms.

2.5 Use Cases

Describe the following items:

- ✦ Actors & use cases
- ✦ Use case diagrams
- ✦ High level, essential use cases **Actors & Use Cases Actors:**

1. **Student**: View faculty info, announcements, societies event images coordinators.
2. **Chatbot**: Provides automated answers to FAQs.

Use Case Diagram

Actors interact with use cases like:

- **Student**: View info about DOC and, Chat with Chatbot.
- **Chatbot**: Provide FAQ Assistance.

High-Level Use Cases

- **Admin:**: can add message, make announcement, upload images and chat with chatbot..
- **Chatbot**: Provide FAQ assistance.

2.5.1 List of Actors

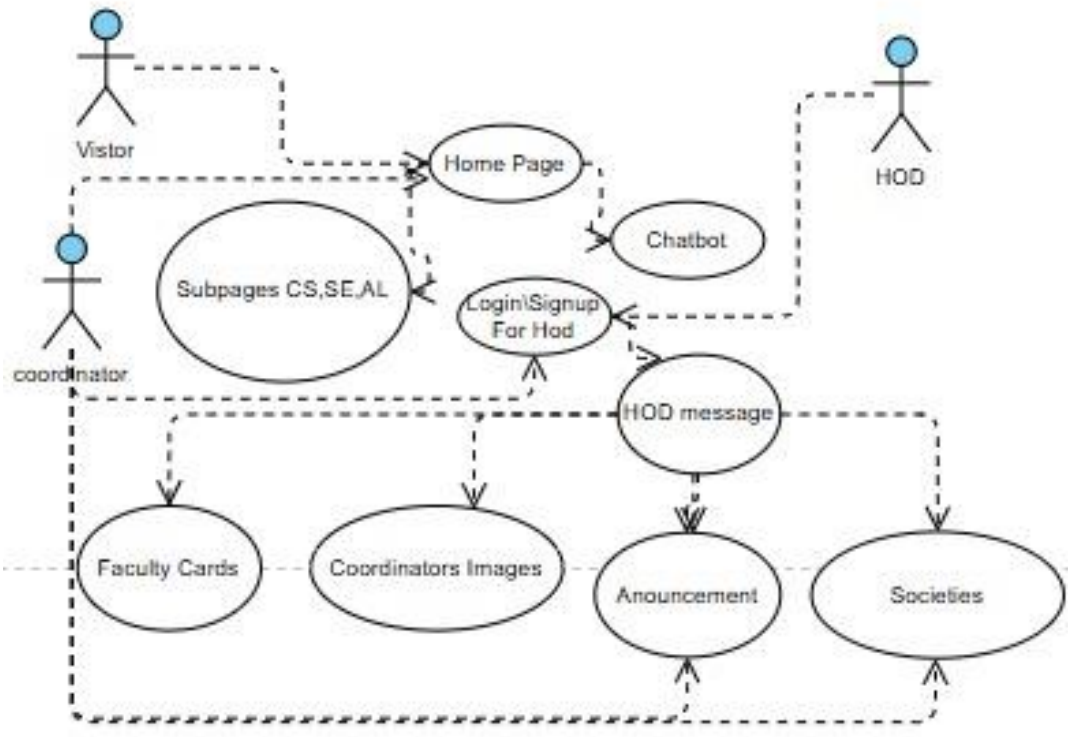
4.2.1 List of Actors and Use Cases

Actor	Use Cases
Student	Societies- Chat with Chatbot - Departmental Information
Faculty	- View Personal and Departmental Information - Societies- Chat with Chatbot
Administrator	- Update Faculty and Departmental Information. Make announcement, add images, add message of HOD
Prospective Student	- View departmental Information
Chatbot	- Provide FAQ Answers - Assist Users with Website Navigation

2.5.2 **List of Use Cases**

Use Case	Description
Chat with Chatbot	Provide automated responses to common questions.
Update Faculty Information	Admin updates faculty and departmental details.
Make Announcement	Admin can make announcements.
Add HOD's message	HOD can add his message from admin panel
Display News and Announcements	Show updates and announcements for the department.

2.5.3 Use Case Diagram



2.5.4 Description of Use Cases

Use Case: Access societies images

Section	Details
Name	Access Societies
Actors	Coordinator, HOD
Purpose	Allow users to view or update societies details.

Typical Course of Events:

Actor Action	System Response
1. User selects "Societies" from the home page.	System displays societies details.

3. Non - Functional Requirements

3.1 . Usability

- User-friendly and intuitive UI/UX for both frontend users and admin.
- Responsive design for mobile, tablet, and desktop.

5.2. Reliability

- Website should load without errors and handle incorrect data gracefully.
- Admin actions should not cause crashes or data loss.

5.3 Security

- Admin panel protected with login credentials.
- Passwords stored securely.

5.4 Performance

- Fast loading for pages.
- Optimized images for faculty, and events.

5.5. Maintainability

- Code should be modular and well-documented.
- Future developers can easily update or scale modules.

5.6. Availability

- Website should have >99% uptime once hosted.
- Pages and chatbot should work even if some data is not present. 5.7. Scalability
- Can easily add new programs or extend faculty without major structure changes.
- Can later integrate real-time chat, course listings, or student portals. 5.8. Compatibility
- Supported on all modern browsers (Chrome, Firefox, Edge).
- Works on Android, iOS, and desktop OS platforms.

A3. DESIGN SPECIFICATIONS

1 Introduction

1.1 Purpose of Document

The purpose of this Software Designs Specification (SDS) document is to describe everything needed to build a website for the Department of Computing at Hamdard University. This website will make it easy for students, faculty, and administrators to access information, perform tasks, and stay updated.

Here's why this document is important:

1. Clarify What Needs to Be Built It explains all the features and tools the website will include, like subpages for programs (Computer Science, AI, etc.), faculty details, timetables, announcements, and automated forms.

2. Centralize All Requirements It gathers all the technical and functional details in one place, ensuring everyone working on the project understands the goals and how the website should function.

3. Improve Communication the website will allow:

- Students to access information easily.
- Faculty and administrators to make announcements and manage tasks efficiently.
- A chatbot to answer user questions, connected to the main university website.

4. Save Time with Automation It defines how various forms (admission, applications, fee concessions, complaints) will be submitted and automatically emailed to the relevant person. If there’s no response in 24 hours, the issue will escalate to the DEAN.

5. Guide the Technical Team It specifies the tools, software, and hardware required to build the website, ensuring the development team knows what technology to use.

6. Define Roles and Responsibilities It documents who can do what on the website (e.g., only the DEAN and HOD can post announcements, while coordinators can update timetables).

Intended Audience

The purpose of this project is to:

1. Current students and faculty of computing department at Hamdard University.

2. Prospective students interested in the Department of Computing and its sub-disciplines.

3. Visitors seeking information about FEST and its computing department.

Document Convention

Font: Calibri (Body)

Font Size: 12

Diagrams: UML diagrams are used to represent the system design visually.

Project Overview

The Department of Computing website is designed to provide a centralized platform for managing academic information and processes. Key functionalities include:

- Sub-web pages for Software Engineering, Computer Science, Artificial Intelligence, and Computer Systems Engineering.
- Announcement areas managed by authorized personnel.
- A chatbot to assist users with queries related to Hamdard University.

The development employs modern web technologies such as HTML, CSS, JavaScript, React.js, Node.js, or Python, with a robust database backend (SQL lite or PostgreSQL).

Scope

Department Overview Pages: Comprehensive pages dedicated to Computer Science, and Software Engineering, Artificial Intelligence each detailing faculty members, batch advisors, coordinators, and the department head.

Announcement System: A portal where only the HOD and Dean can log in to make announcements to students and faculty**.**

Chatbot Integration: A chatbot that connects to the Hamdard University website to provide responses to frequently asked questions and queries, leveraging the university's existing information.

2 Design Considerations

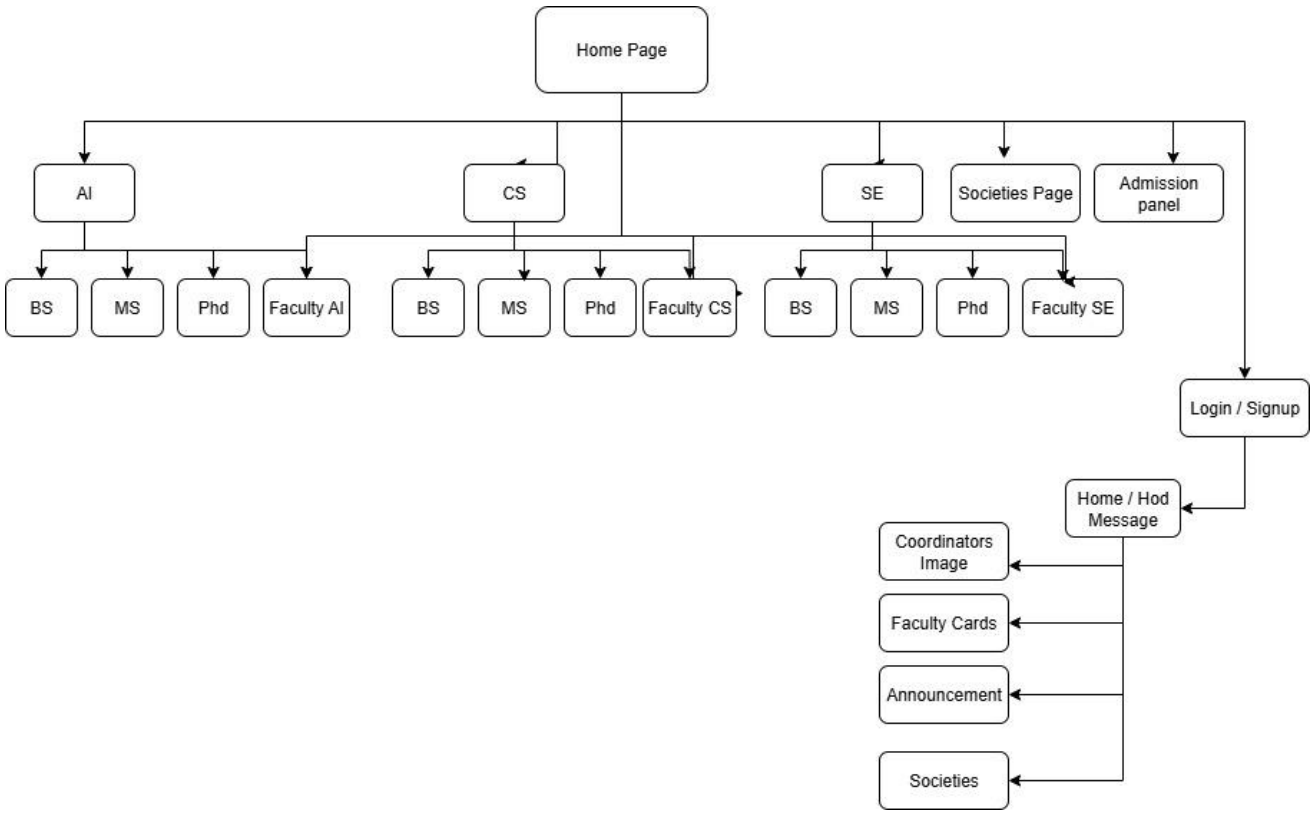
Assumptions and Dependencies

Modular Architecture: Ensuring seamless integration and testing in a scalable modular design. **Technology Compatibility**: Managing potential issues between frameworks like React.js, Node.js/Django, and databases.
UI/UX Challenges: Creating accessible, responsive interfaces across devices per WCAG guidelines.
Database Optimization: SQL lite
Concurrency Management: Handling multiple users with efficient backend programming and load balancing.
Security Measures: Implementing robust access control, encryption, and activity logging without performance loss.
Scalability: Preparing the system for future enhancements like advanced AI and real-time notifications.
Error Recovery: Designing systems for graceful failure handling and maintaining data integrity.
Testing Integration: Ensuring components are testable during iterative development.
Resource Constraints: Addressing limited developer resources with streamlined workflows and tool reliance.

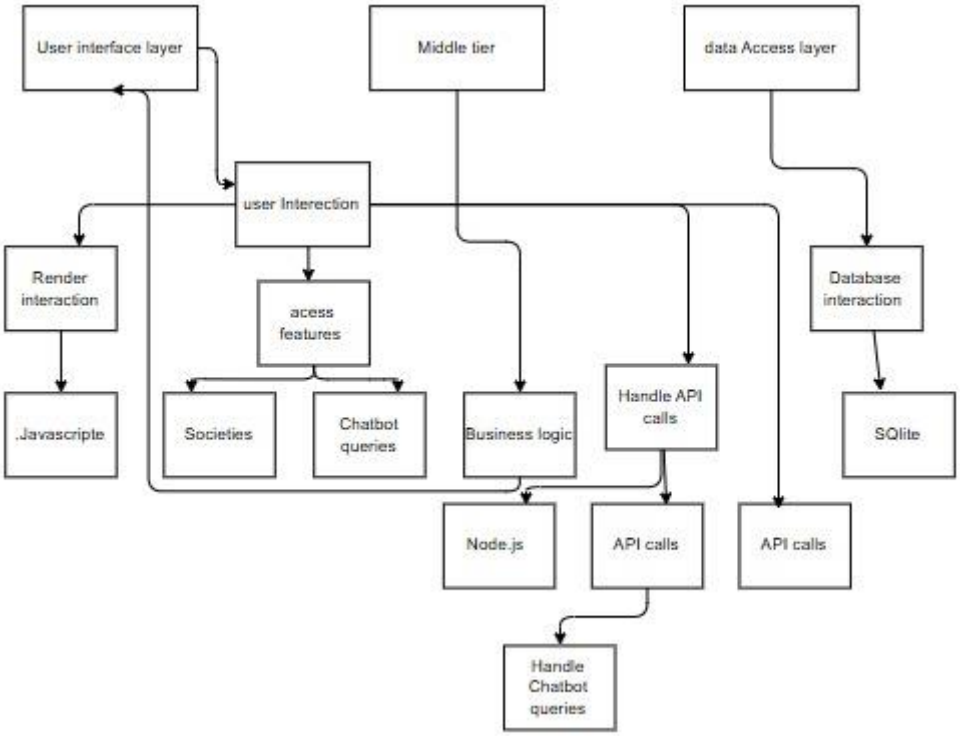
Risks and Volatile Areas

Requirement Changes: New feature requests may arise. Mitigation: Use modular design and agile methods.
Technology Risks: Updates to frameworks may cause compatibility issues. Mitigation: Use wellsupported tools and monitor trends.
Performance Bottlenecks: High user loads or database queries may slow the system. Mitigation:
Conduct load testing and optimize queries.
Security Vulnerabilities: Risks of breaches or attacks. Mitigation: Enforce strict security protocols and conduct audits.
Resource Constraints: Limited skilled personnel could delay progress. Mitigation: Cross-train staff and outsource tasks as needed.
Integration Challenges: Issues with connecting APIs or external systems. Mitigation: Plan robust integrations and test early.
Data Inconsistencies: Outdated or incorrect data may affect usability. Mitigation: Automate validation and verify sources regularly.
Scalability Issues: Future needs might outgrow current design. Mitigation: Use scalable cloud-based and microservices architecture.

System Architecture



Software Architecture



Design Strategy

1. **Future System Extension or Enhancement**

- **Design Strategy**:
  - Modular architecture ensures that components can be added or replaced without significant impact on the overall system.
  - Use of APIs enables the integration of new features such as advanced analytics, mobile applications, or new user interfaces.
  - The system is designed to support additional sub-webpages for new departments or courses with minimal effort.
  - Advanced AI models can be incorporated into the chatbot in future iterations.
- **Reasoning**:
  - Aligning with long-term scalability goals ensures the system remains relevant and adaptable to changing requirements.
- **Trade-offs**:
  - Initial design complexity is increased to accommodate modularity.
  - Slightly higher development time due to the need for generalized and extensible components.

2. System Reuse

- **Design Strategy**:
  - Use of reusable components such as:
    - React.js components for consistent UI elements.
    - RESTful APIs for handling data across multiple client applications (e.g., web and mobile).
    - Common utility modules for logging, error handling, and data validation.
  - Database schemas designed for flexibility and reusability across different modules.
- **Reasoning**:
  - Reusable components save time and resources during both initial development and future updates.
  - Encourages consistency across the system.
- **Trade-offs**:
  - Slightly higher upfront effort to ensure components are generic and reusable. o Potential underuse of some reusable components in the initial deployment phase.

3. User Interface Paradigms

- **Design Strategy**:
  o The user interface follows a **responsive and user-friendly design**:
    - WCAG (Web Content Accessibility Guidelines) compliance ensures accessibility for all users, including those with disabilities.
    - A mobile-first approach using frameworks like **Bootstrap** or **Material-UI** ensures compatibility across devices.
    - Clear navigation structure reduces the learning curve for users.
    - Contextual help such as tooltips and a chatbot improves user interaction.
- **Reasoning**:
  o A well-designed UI is essential for user satisfaction and adoption of the system.
  - Accessibility ensures the platform can be used by a broader audience.
- **Trade-offs**:
  - Increased development time for ensuring cross-platform compatibility and accessibility compliance.
  - Requires additional testing across a variety of devices and screen sizes.

4. Data Management (Storage, Distribution, Persistence)

- **Design Strategy**:
  - Hybrid database approach:
    - **SQL lite** for structured relational data like timetables and user information.
  - Use of database replication and periodic backups to ensure high availability and data persistence.
  - APIs for efficient data distribution and retrieval.
  - Optimized indexing and query mechanisms to handle large datasets.

- **Reasoning**: o A hybrid database design allows the system to handle diverse data types efficiently.
    - Regular backups and replication ensure data security and availability.
- **Trade-offs**:
    - Increased system complexity due to managing two database systems.
    - Higher resource requirements for maintaining database synchronization and backups.

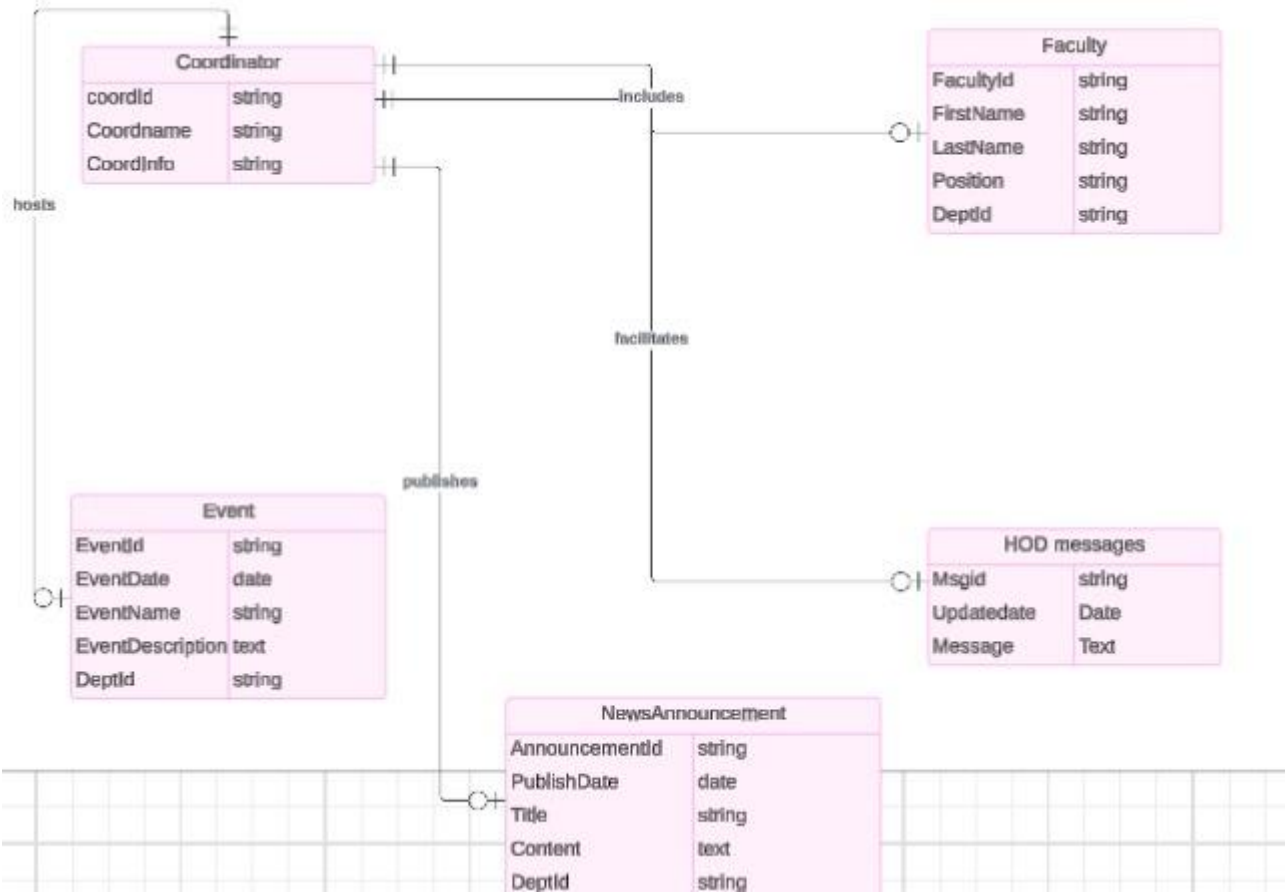
5. Concurrency and Synchronization

- **Design Strategy**:
    - Use of asynchronous programming in **Node.js** or **Django** to handle multiple user requests simultaneously.
    - Synchronization mechanisms such as:
        - **Optimistic concurrency control** to prevent conflicts in simultaneous database updates.
        - Distributed locks or similar mechanisms to ensure data consistency.
    - Load balancing techniques to distribute requests evenly across servers.
- **Reasoning**:
    - High concurrency support ensures smooth operation during peak usage times.
    - Synchronization mechanisms maintain data integrity.
- **Trade-offs**:
    - Higher resource utilization due to concurrency management.
    - Slight increase in system latency when synchronization mechanisms are employed.

Detailed System Design Design Class Diagram

Database Design

ER Diagram



Announcement Table

- **Id:** Unique identifier for each announcement.
- **UserId:** Foreign key referencing the User table.
- **Content:** The announcement text.
- **DatePosted:** Date the announcement was posted.

HOD Message

- **Id:** Unique identifier for each message.
- **UserId:** Foreign key referencing the User table.
- **Content:** The message text.
- **DatePosted:** Date the message was posted.

Coordinator image

- **Id:** Unique identifier for each image.
- **UserId:** Foreign key referencing the User table.
- **Content:** The coordinator image.
- **DatePosted:** Date the image was posted.

Faculty Card

- **Id:** Unique identifier for each card.
- **UserId:** Foreign key referencing the User table.
- **Content:** The coordinator image.
- **DatePosted:** Date the card was posted.

Events / Societies images

- **Id:** Unique identifier for each image.
- **UserId:** Foreign key referencing the User table.
- **Content:** The event image.
- **DatePosted:** Date the imagewas posted.

Login

- **Id:** Unique identifier for each image.
- **UserId:** Foreign key referencing the User table.
- **Content:** The user will be loged in.
- **DatePosted:** Date the user logged in.

Relationships:

- **User** has a many-to-many relationship with **Announcement**, meaning two user can post multiple announcements.
- **User** has a one-to-many relationship with **Messages**, meaning one user can post multiple messages.
- **User** has a one-to-many relationship with **Faculty cards**, meaning one user can post multiple announcements.
- **User** has a one-to-many relationship with **coordinator image**, meaning one user can upload multiple coordinators.
- **User** has a many-to-many relationship with **societies/events image**, meaning two user can upload multiple images.
- **User** has a one-to-one relationship with **HOD login panel**, meaning one user can login to HOD panel.
- **User** has a one-to-one relationship with **coordinator's login**, meaning one user can login to coordinator's panel

Data Dictionary

Entity	Attribute	Description	Type	Constraints
User	user_id (PK)	Unique identifier for users	INT	Primary Key, Auto-increment
	name	Name of the user	VARCHAR(255)	Not Null
	email	Email address	VARCHAR(255)	Not Null, Unique
	password	Password for authentication	VARCHAR(255)	Not Null
	role	Role of the user (e.g., student, faculty, admin)	ENUM	Values: 'student', 'faculty', etc.
Message	message_id (PK)	Unique ID for complaints	INT	Primary Key, Auto-increment
	description	Description of the complaint	TEXT	Not Null
	status	Current status of the complaint	ENUM	Values: 'open', 'closed', etc.
	created_at	Timestamp for when the message was submitted	DATETIME	Not Null
Society	society_id (PK)	Unique ID for society	INT	Primary Key, Auto-increment
	batch	Batch associated with the timetable	VARCHAR(50)	Not Null
	program	Program name	VARCHAR(50)	Not Null
Announcement	announcement_id (PK)	Unique ID for announcements	INT	Primary Key, Auto-increment
	author_id (FK)	User who created the announcement	INT	Foreign Key to User.user_id
	message	Content of the announcement	TEXT	Not Null
	created_at	Timestamp for when the announcement was created	DATETIME	Not Null

Data 1

Data 1						
Name	Give primary name of the data or control item, the data store or an external entity.					
Alias	System User, Account Holder					
Whereused/how-used	Used in authentication processes (input to login). Referenced in complaints, announcements, and chatbot interactions. Acts as a control entity for role-based access.					
Content description	Represents all users (students, faculty, admin) interacting with the system.					
Column Name	**Description of the Column**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
user_id	**Unique identifier for users**	**INT**	**-**	**No**	**Autoincrement**	**PK**
name	**Name of the user**	**VARCHAR**	**255**	**No**	**-**	
email	**Email address**	**VARCHAR**	**255**	**No**	**-**	**UNIQUE**
password	**Password for authentication**	**VARCHAR**	**255**	**No**	**-**	

Data 2

4.2.2.4

Name	Announcement News Update, Notification Stores announcements for the department.					
Alias	News Update, Notification					
Where-used/howused	• Created by admin or HOD for department-wide updates. • Accessible to students and faculty.					
Content description	Stores announcements for the department.					
Column Name	Description of the Column	Type	Length	Null able	Default Value	Key Type
announcement_id	Unique ID for announcements	INT	-	No	Auto-increment	PK
author_id	User who created the announcement	INT	-	No	-	FK

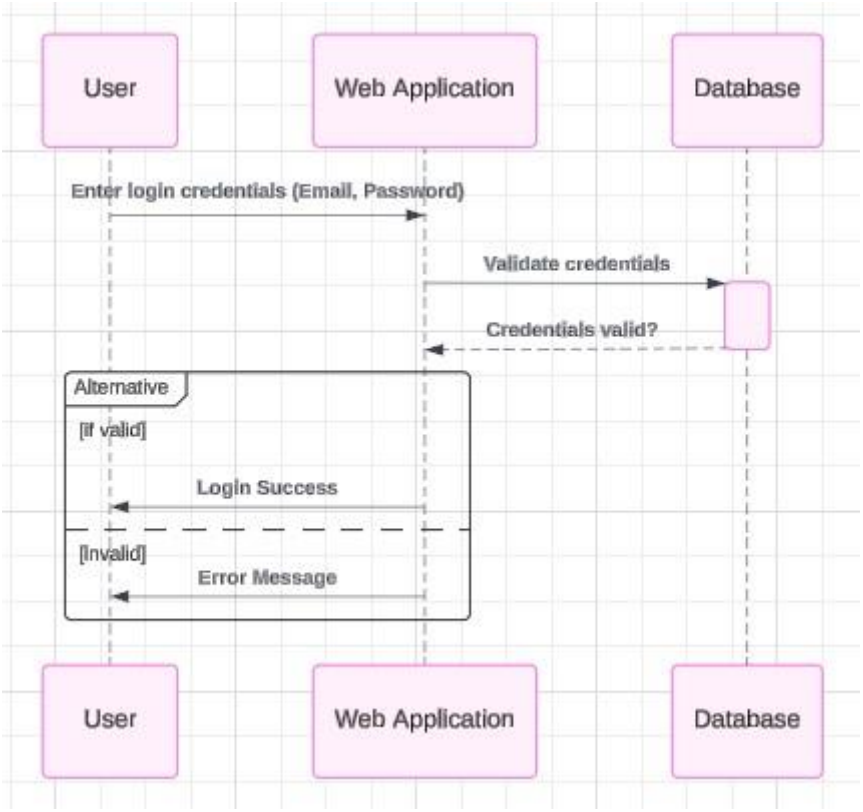
message	Content of the announcement	TEXT	-	No	-	
created_at	Timestamp of announcement creation	DATETIME	255	No	CURRENT_TIMESTAMP	

4.2.2.5
Data 5

Name	Society					
Alias	Society images					
Whereused/howused	• Upload society events images					
Content description	Upload societies event image .					
Column Name	**Description of the Column**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
query_id	Unique ID for society images	INT	-	No	Auto-increment	PK
user_id	User interacting with the images	INT	-	No	-	FK
query	User's query	TEXT		No	-	
timestamp	Timestamp of interaction	DATETIME	-	No	CURRENT_TIMESTAMP	

Application Design

<Sequence Diagram 1> User login process

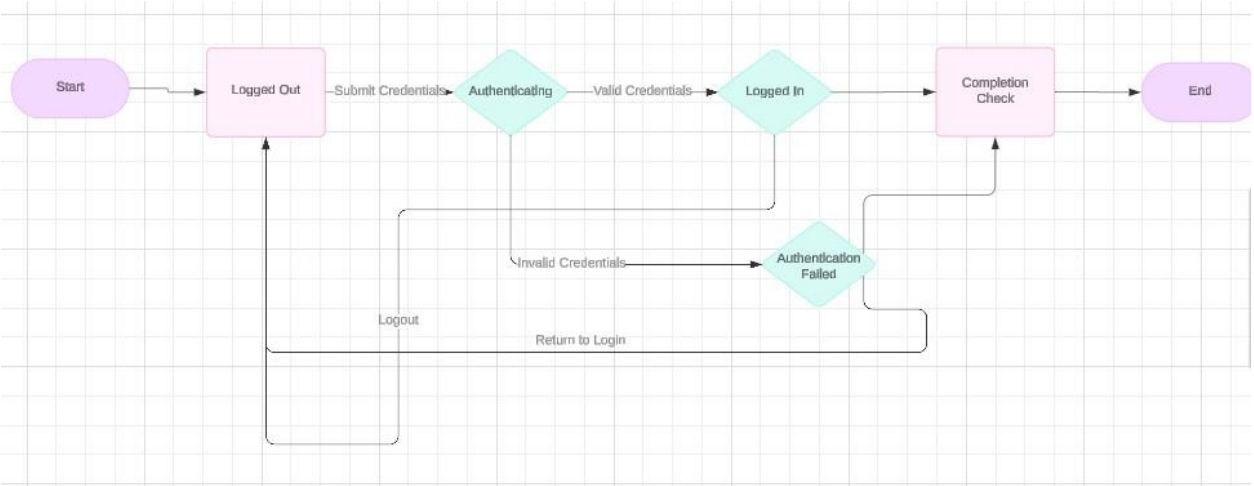


Explanation:

User enters login credentials (Email, Password)
Web Application sends credentials to Database for validation.
Database validates credentials.
Database sends validation result to Web Application.
If credentials are valid, the Web Application sends a "Login Success" message to the User. If credentials are invalid, the Web Application sends an "Error Message" to the User.

State Diagram

<State Diagram 1>User Authentication

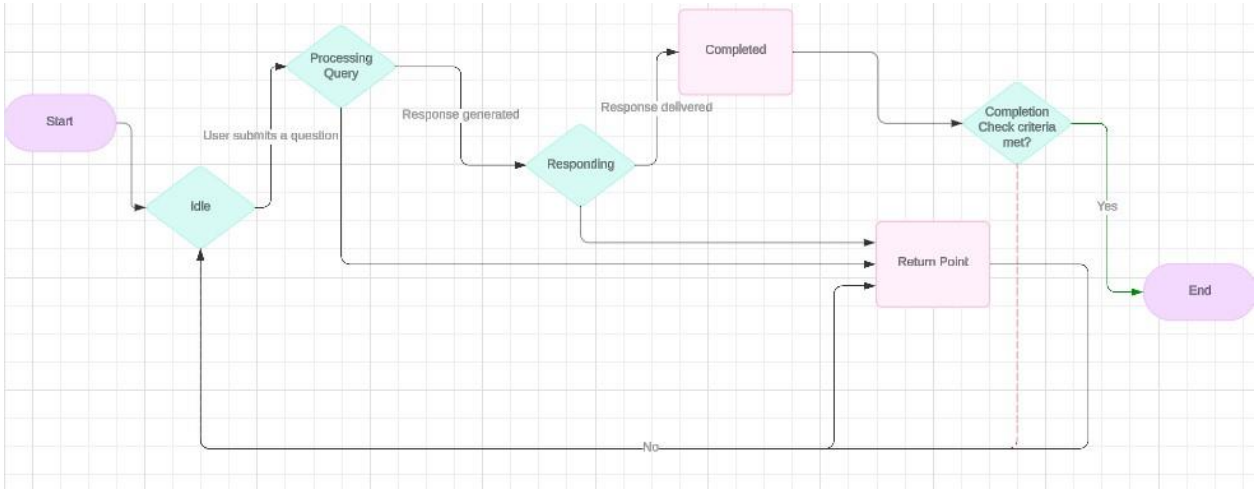


Explanation: Start: The process begins.
Logged Out: The user is not logged in.
Submit Credentials: The user provides their login credentials.
Authenticating: The system validates the provided credentials.

- Valid Credentials: The credentials are correct. The process continues.
- Invalid Credentials: The credentials are incorrect. The user is redirected to the Login page.

Logged In: The user is successfully logged in.
Completion Check: The system verifies the user's login status.
End: The process is complete.
Authentication Failed: If the login fails, the user is presented with an authentication failure message.
Logout: The user can log out of the system, which takes them back to the Logged Out state.
Return to Login: The user is redirected to the Login page after an unsuccessful login attempt.

<State Diagram > Chatbot Interaction



Explanation:

Start: The process starts with a user submitting a question.

Idle: The system is in an idle state, waiting for a user query.

Processing Query: Once a question is submitted, the system moves into the "Processing Query" state, where the question is processed.

Responding: The system generates a response to the question and enters the "Responding" state.

Completed: The system delivers the response to the user and marks the query as "Completed".

Completion Check Criteria Met? The system checks whether all completion criteria have been met.

End: If the completion criteria are met, the process ends.

Return Point: If the completion criteria are not met, the process returns to the "Return Point" and continues to cycle through the process until the criteria are met.

A4. OTHER TECHNICAL DETAIL DOCUMENTS

Test Cases Document

Test Case 1

Title: HOD Register

Precondition	Show Treatment
Actions	Register HOD
Expected Result	Password mailed to embedded email
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 2

Title: Coordinator Register

Precondition	Show Treatment
Actions	Register Coordinator
Expected Result	Password mailed to HOD
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 3

Title: HOD Login

Precondition	Show Treatment
Actions	Login HOD
Expected Result	Login to HOD admin panel
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 4

Title: Coordinator Login

Precondition	Show Treatment
Actions	Login Coordinator
Expected Result	Login to Coordinator admin panel
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 5

Title: Add HOD Message

Precondition	Show Treatment
Actions	HOD Adds HOD Message
Expected Result	Added to home page
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 6

Title: HOD Announcement

Precondition	Show Treatment
Actions	HOD make announcement
Expected Result	Show on announcement banner
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 7

Title: Upload Coordinator Images

Precondition	Show Treatment
Actions	HOD upload Coordinator images
Expected Result	Showed on home page
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 8

Title: Add faculty cards

Precondition	Show Treatment
Actions	HOD add faculty cards
Expected Result	Card viewed on specific Faculty page
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 9

Title: Upload society event images

Precondition	Show Treatment
Actions	HOD add event images
Expected Result	Uploaded on specific society's images section
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 10

Title: Coordinator announcemet

Precondition	Show Treatment
Actions	Coordinator make announcement
Expected Result	Show on announcement banner
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 11

Title: Coordinator add events

Precondition	Show Treatment
Actions	Coordinator add event images
Expected Result	Uploaded on specific society's images section
Tested by	Shahmeer Abid
Result	Verification Successful

Test Case 12

Title: Coordinator add faculty cards

Precondition	Show Treatment
Actions	Coordinator add faculty cards
Expected Result	Uploaded on specific faculty's page
Tested by	Shahmeer Abid
Result	Verification Successful

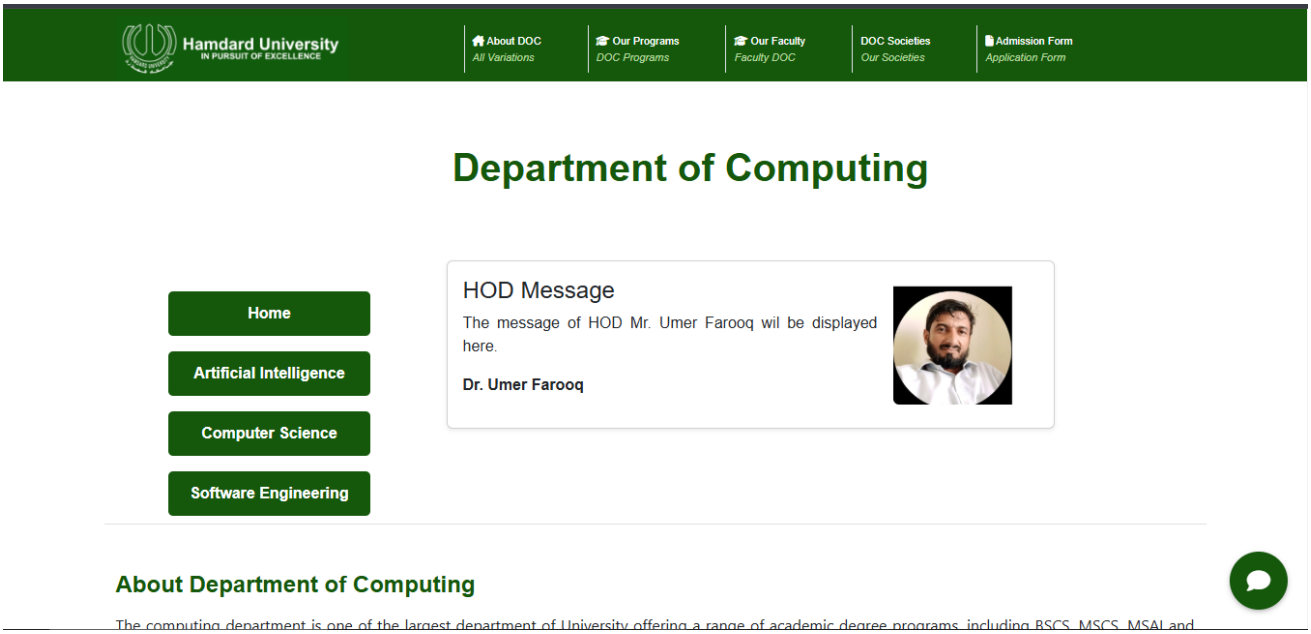
Test Case 13

Title: Coordinator and HOD Logout button

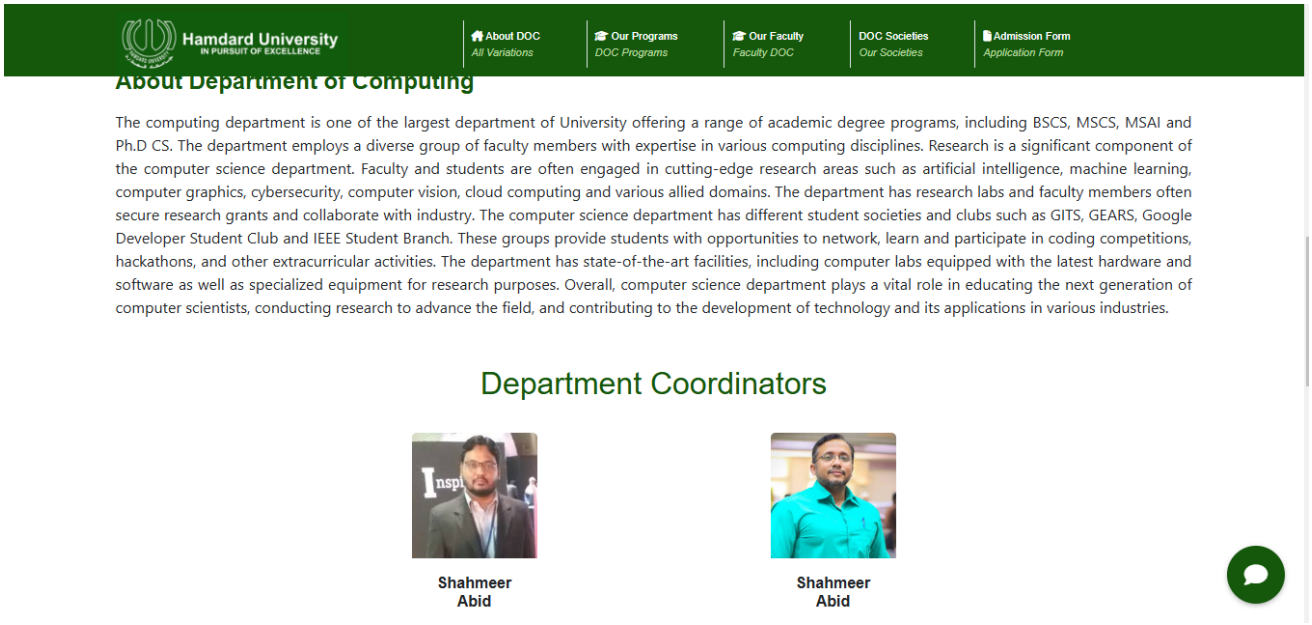
Precondition	Show Treatment
Actions	Coordinators and HOD logout
Expected Result	Redirected to Login page
Tested by	Shahmeer Abid
Result	Verification Successful

UI/UX Detail Document

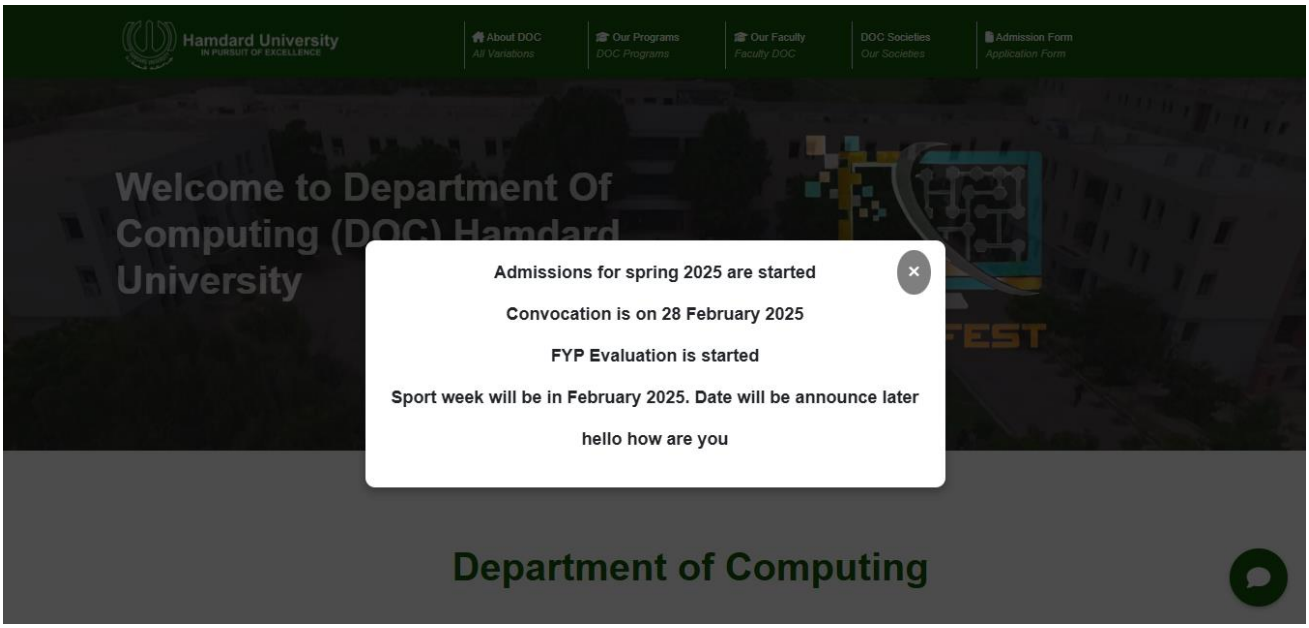
Home Page:



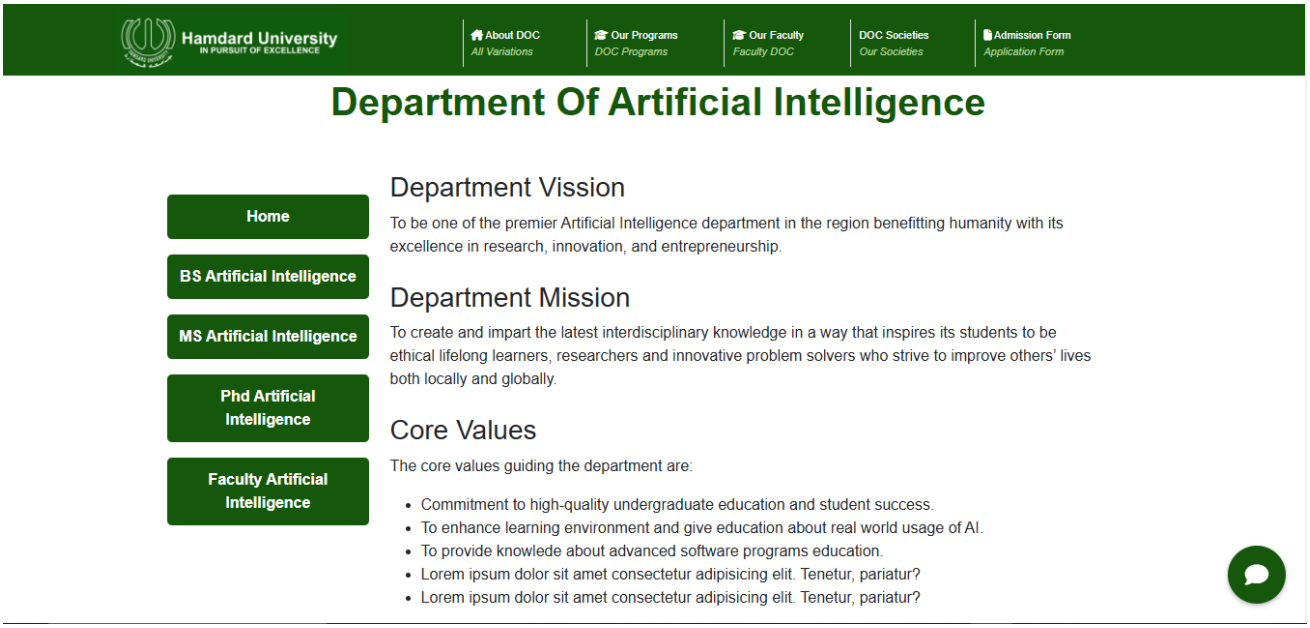
About and Coordinator images:



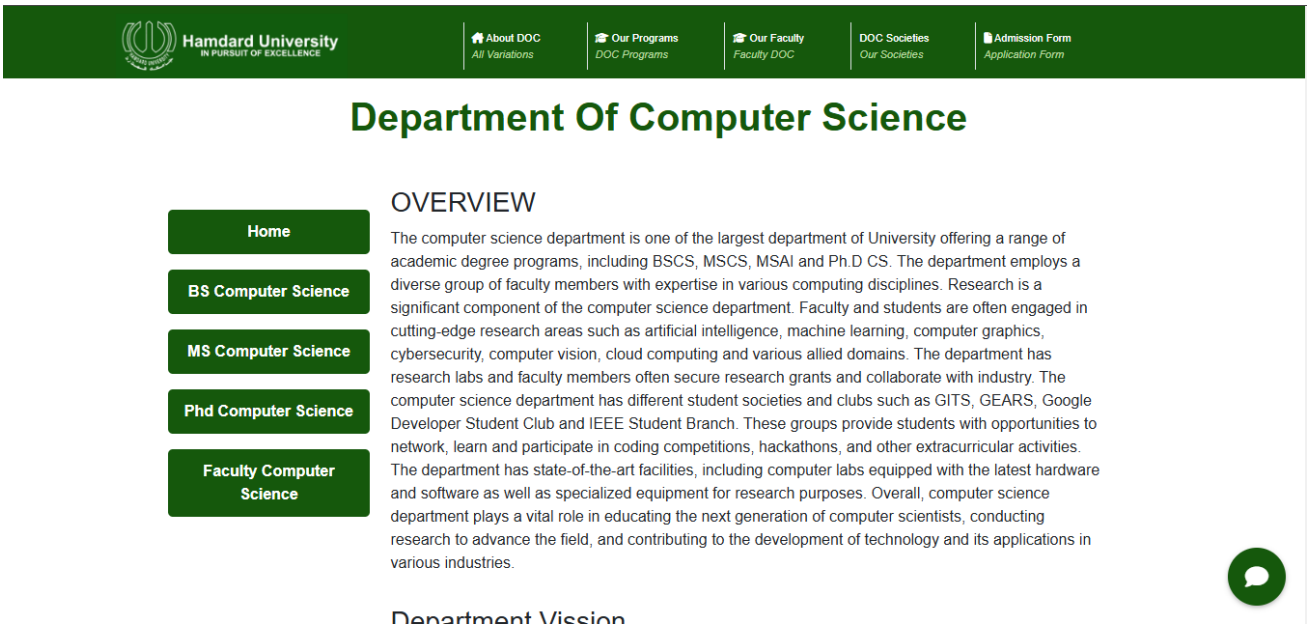
Announcement Banner:



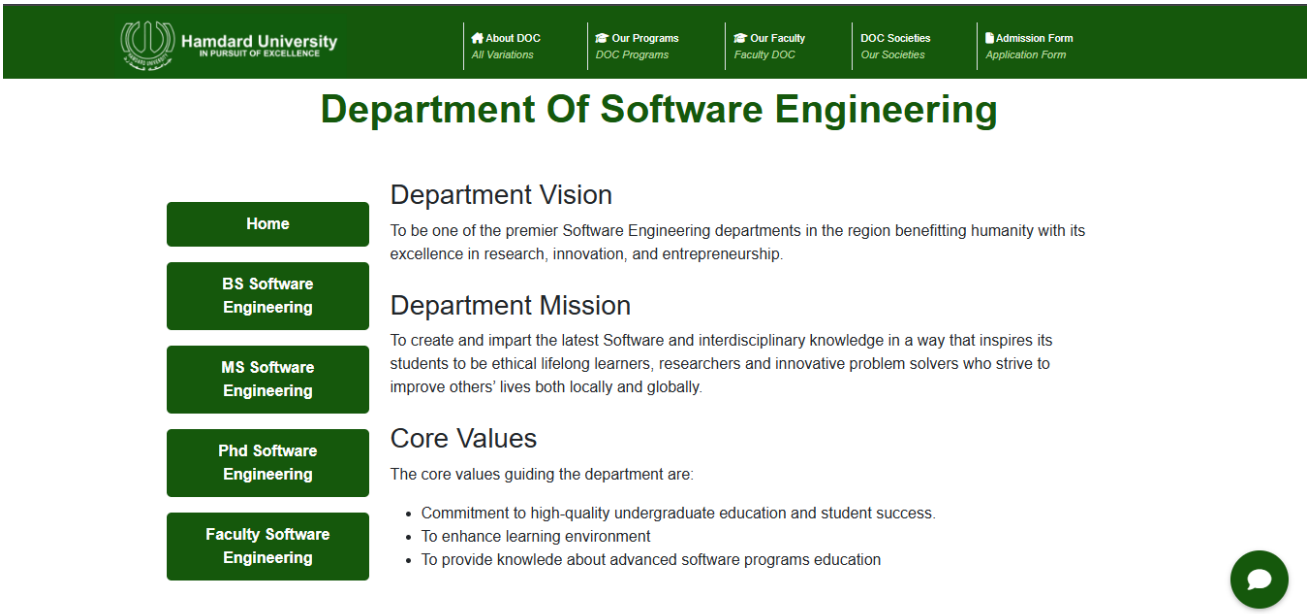
Artificial Intelligence Page:



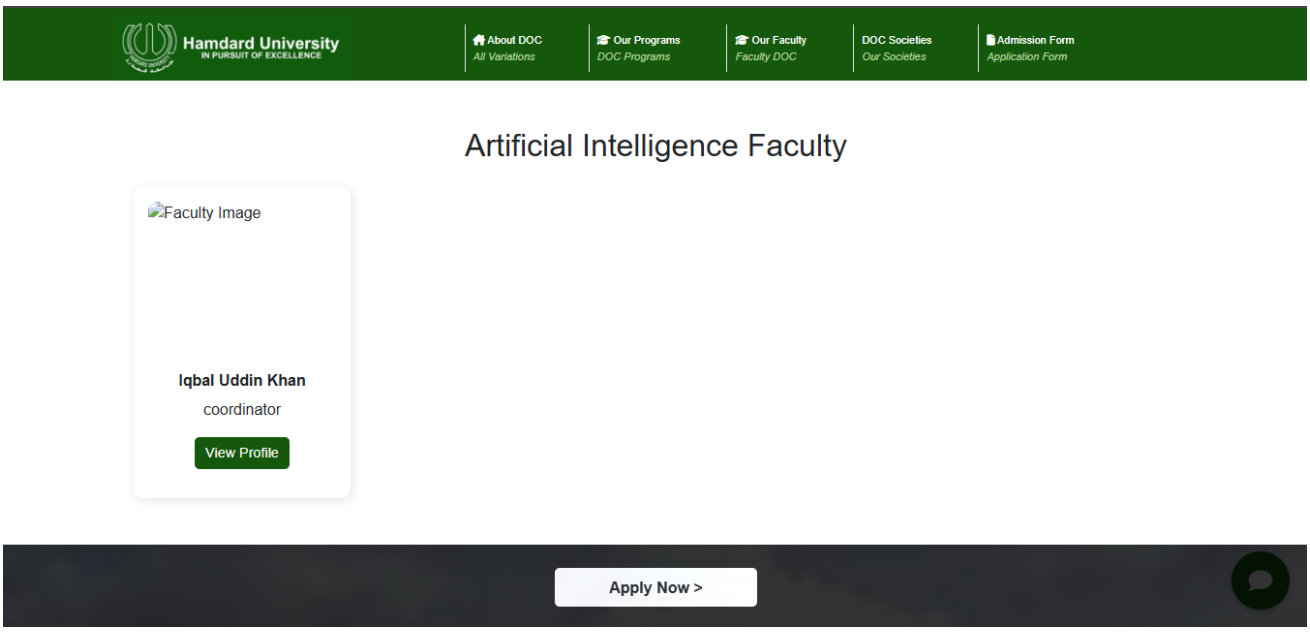
Computer Science Page:



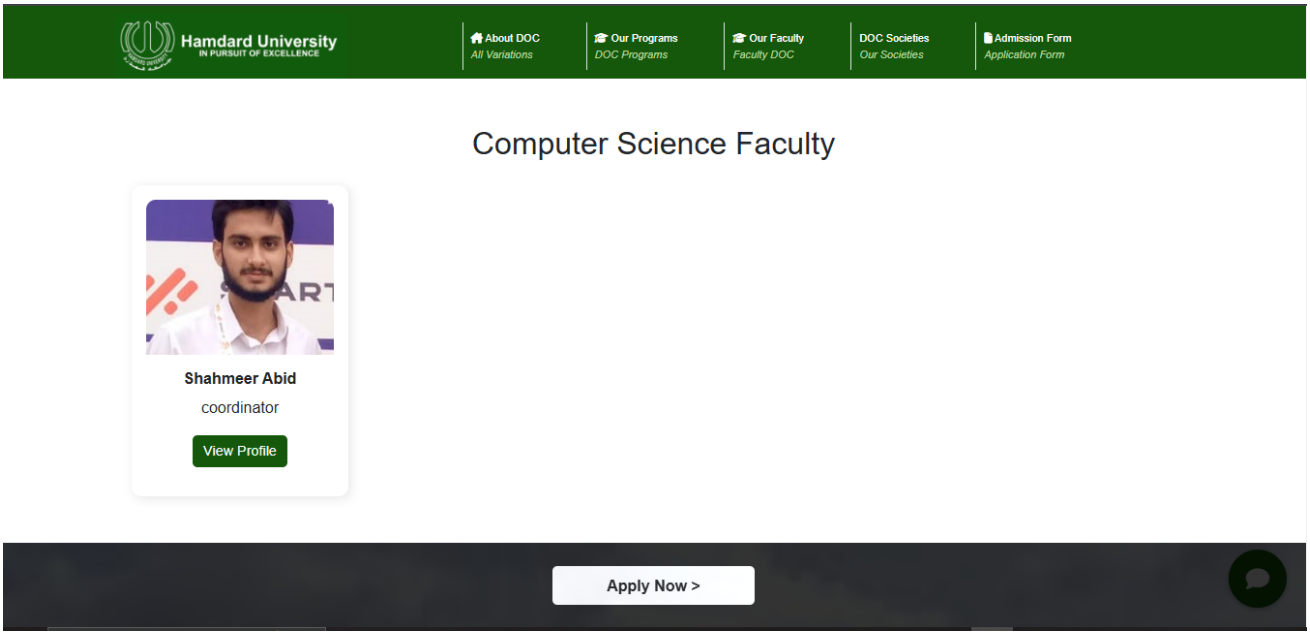
Software Engineering Page:



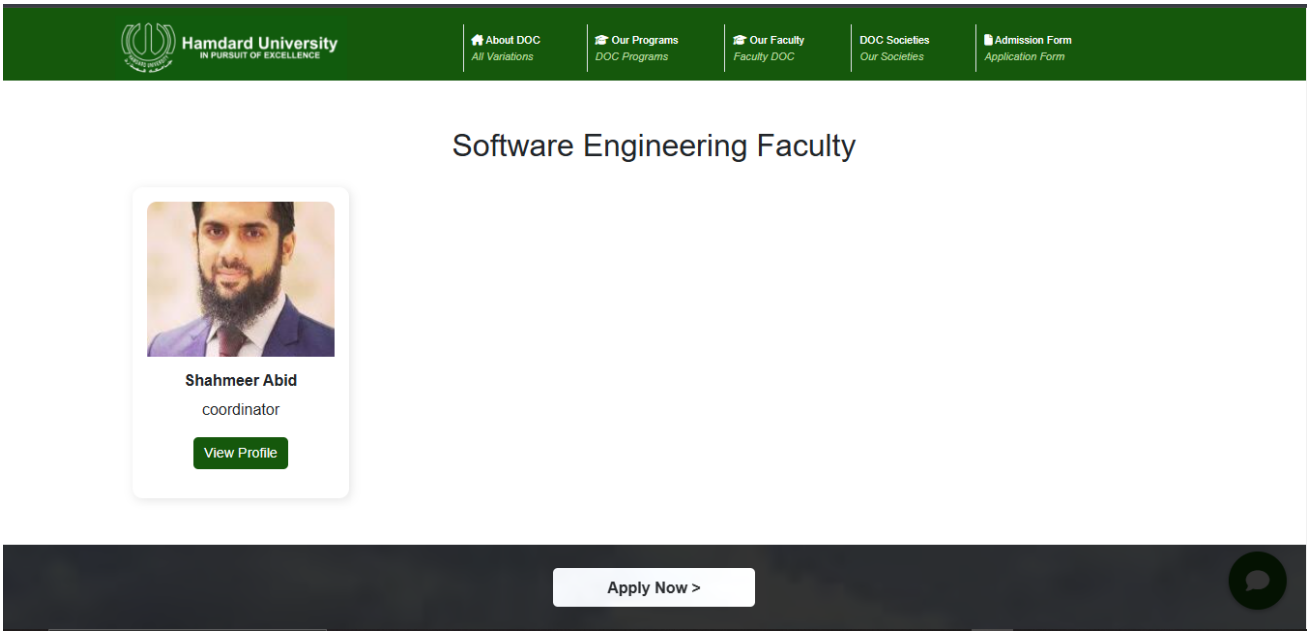
Faculty of AI Page:



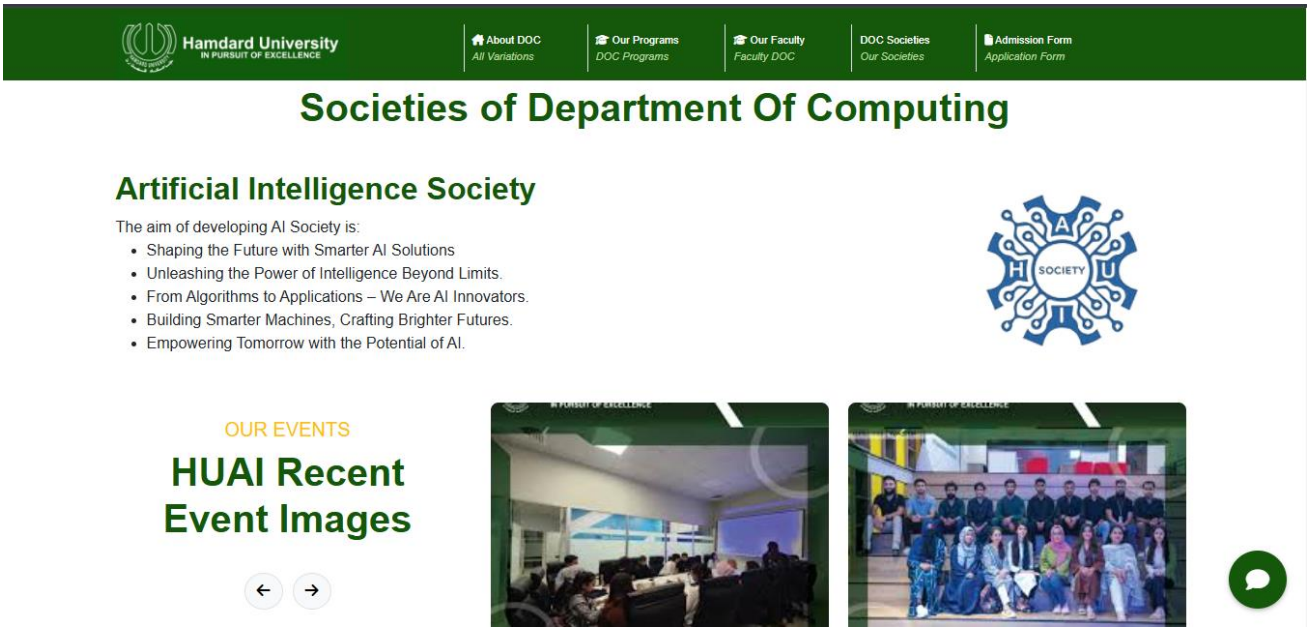
Faculty of CS Page:



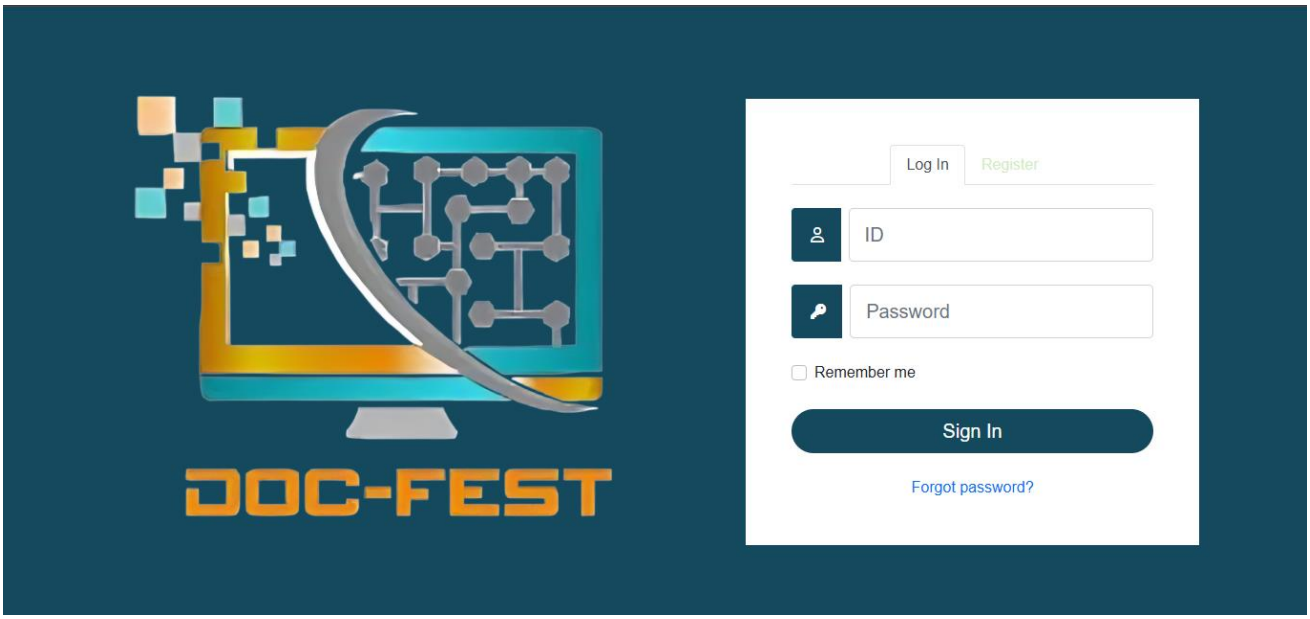
Faculty of SE Page:



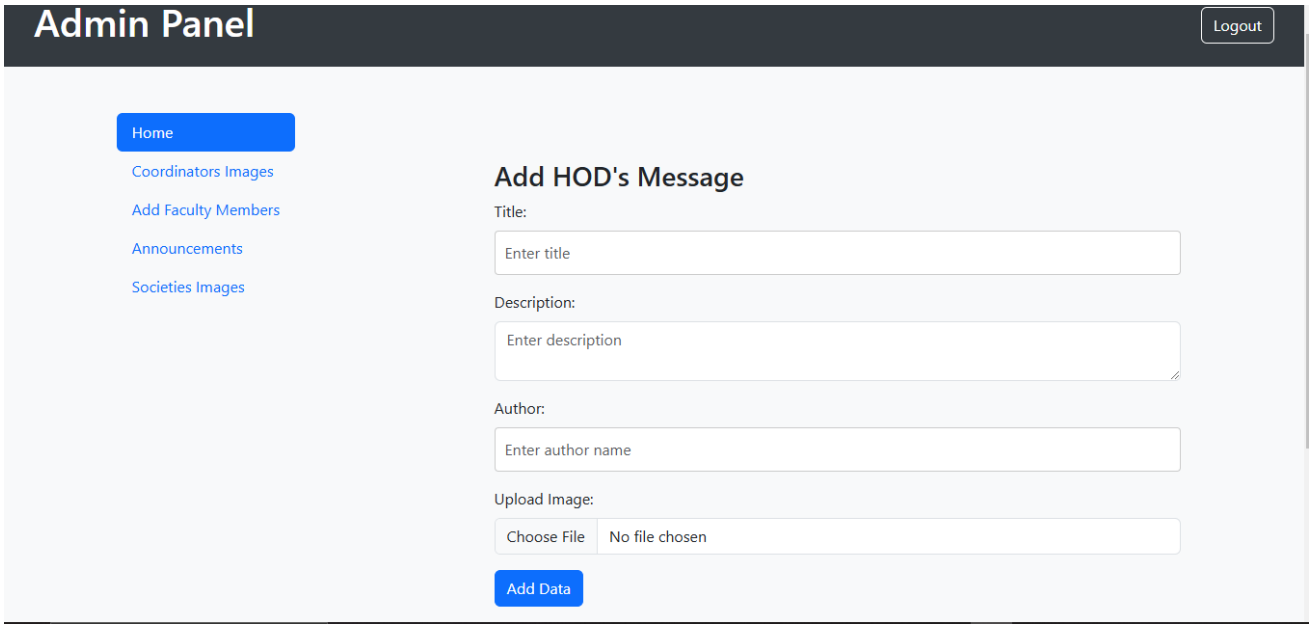
Societies Page:



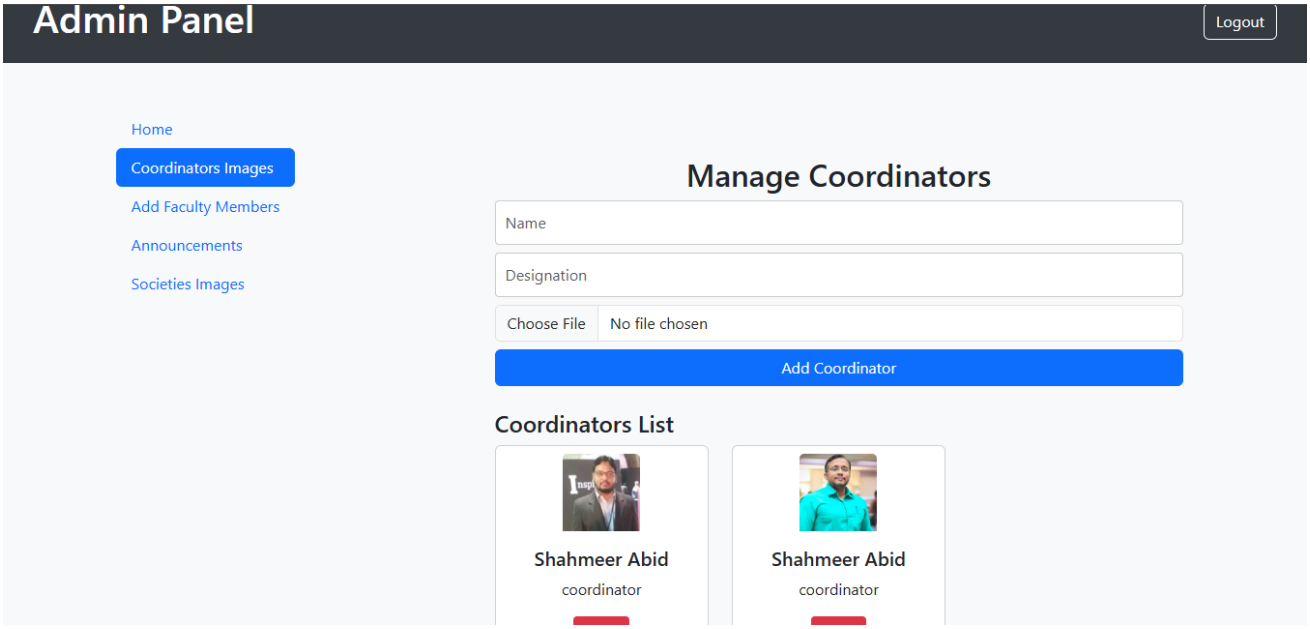
Login and Register Page:



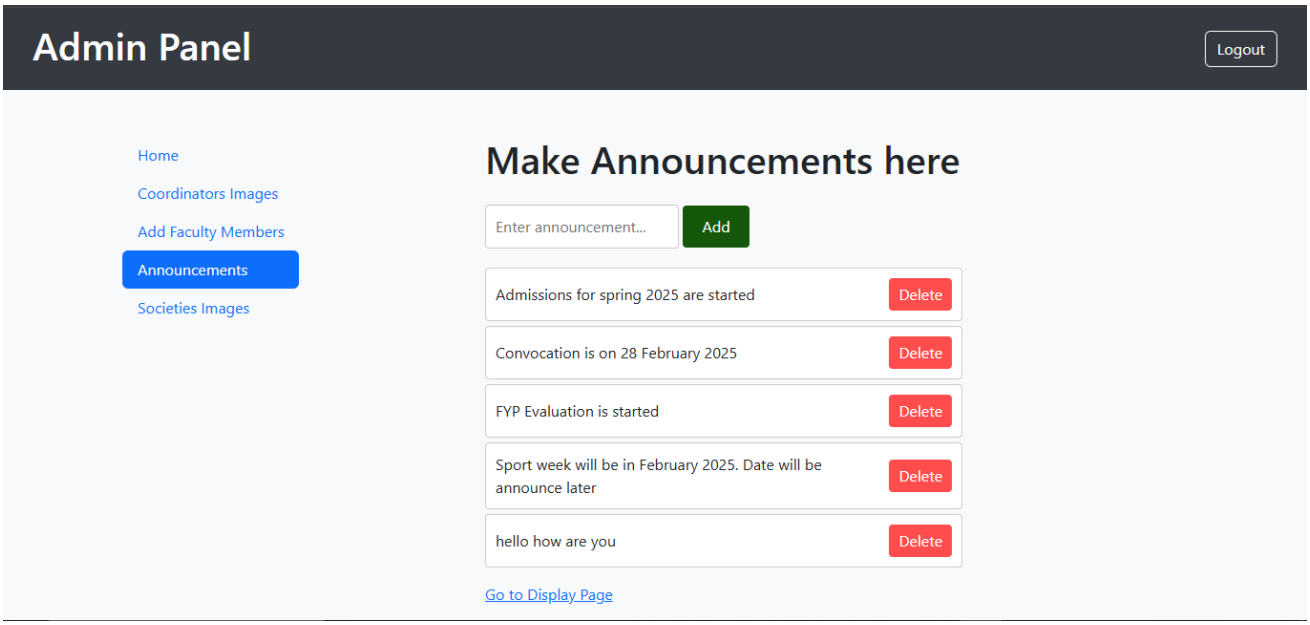
HOD Message panel:



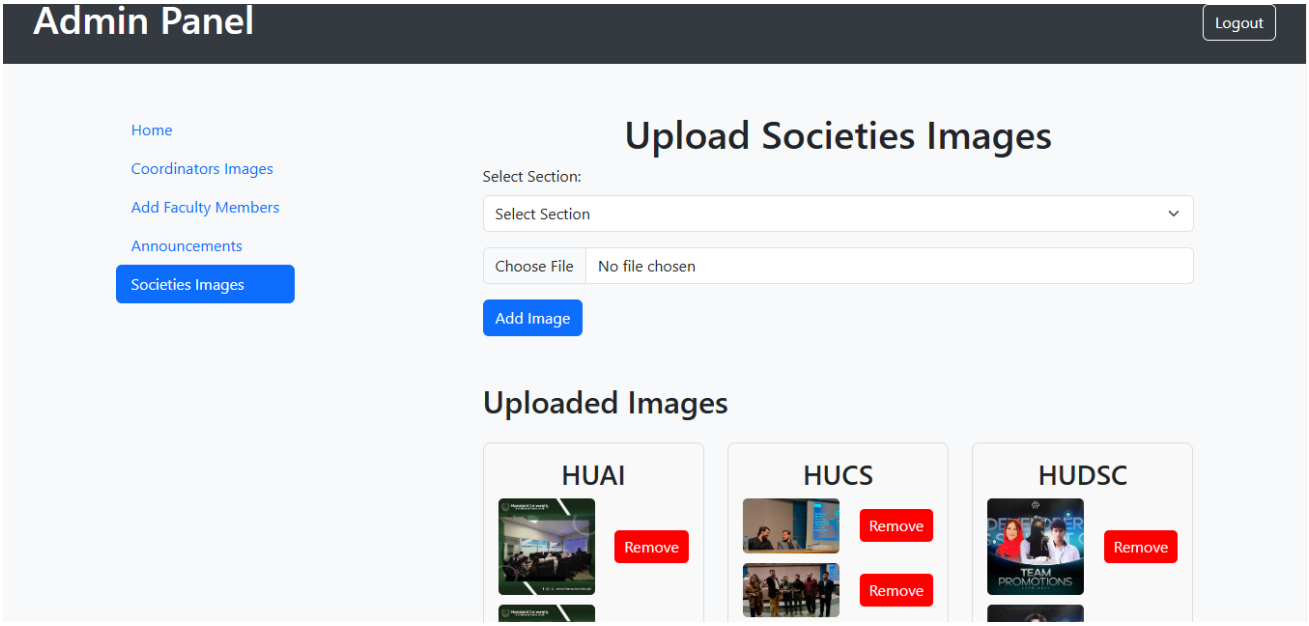
HOD's Coordinator Image Upload panel:



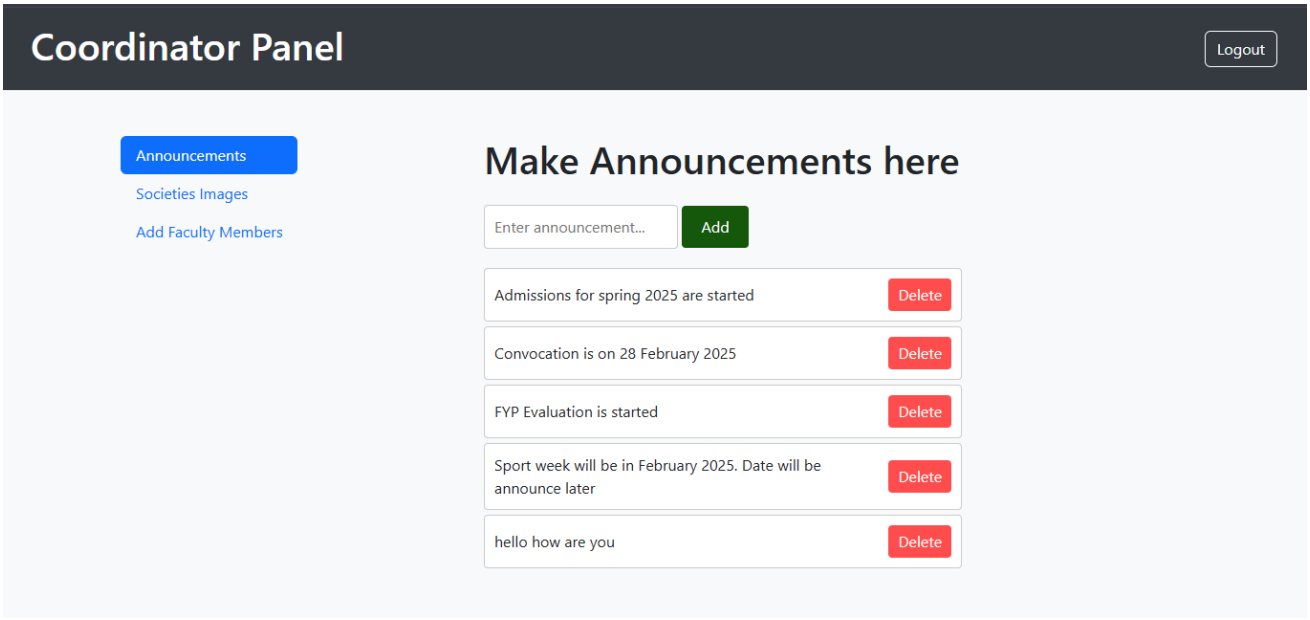
HOD's Announcement Upload panel:



HOD's Societies Image Upload panel:



Coordinator's panel:



Coding Standards Document

To maintain readability, consistency, and long-term maintainability of the **Department of Computing Website**, standard coding practices were carefully followed throughout the development process. These practices ensured that the source code remained clean, well-structured, and easily understandable for any future developers or contributors. Below are the key standards adopted:

- **Meaningful Naming:** Descriptive and meaningful names were used for variables, functions, classes, and files to clearly reflect their purpose and improve code readability
- **Modularity:** The code is organized into reusable functions and components instead of using large inline blocks.
- **Commenting:** Inline comments were added to explain complex logic and workflows. Functions and modules include clear comments or headers describing their purpose, input, and output.
- **Consistency:** A consistent coding style was used throughout the project, including, spacing, bracket placement, and naming conventions like camelCase for JavaScript, kebab-case for CSS classes, etc.
- **Version Control:** GitHub was used for version control to track development progress.
- **Frontend:** Bootstrap library was used when needed, and code was written in vanilla JavaScript to maintain full control and avoid unnecessary dependencies.
- **Backend Standards (Node.js):** The backend logic was developed using Node.js and structured into logical modules, including separate files for routes, controllers, and database interactions.
- **Database Standards (SQLite):** SQLite was used as a lightweight and efficient database for storing structured data such as faculty records, announcements, and coordinator details.

A5. FLYER & POSTER DESIGN

COPY OF EVALUATION COMMENTS BY JURY FOR PROJECT – I END SEMESTER EVALUATION

Dr. Taha Shabbir
Muhammad Salman
Dr. Khalid Charan
Engr. Farooq Iqbal

Enhance the project scope
Overall is ok.
ok
Overall OK. But need more focused study of the project domain, scope and requirements..

A7. MEETINGS' MINUTES & Sign-Off Sheet

MEETINGS' MINUTES

FYP Project Meeting

Minutes of Meeting

Meeting Date: 17/02/2025
Meeting Location: faculty Room
Meeting Time: 12:40

Project Title: A website for DOC Hamdard University
Project Code: FYP-001/FL24

1- List of Participants

Name	Project Role
Shahmeer Abid	Front-end
Syeda Noreen Zahra	Database
Hamza Sheikh	Backend

2- Meeting Agenda

Discussed about the changes according to jury comments.

3- Agenda Points discussed in meeting (in detail)

Discussed about the changes according to jury comments

- We will made changes in our project according to the jury comments.

4- Next Meeting for this project

We will decide as per our feasibility.

Supervisor's Signature: _____

FYP Project Meeting

Minutes of Meeting

Meeting Date: 09/04/2025
Meeting Location: faculty Room
Meeting Time: 12:40

Project Title: A website for DOC Hamdard University
Project Code: FYP-001/FL24

1- List of Participants

Name	Project Role
Shahmeer Abid	Front-end
Syeda Noreen Zahra	Database
Hamza Sheikh	Backend

2- Meeting Agenda

Discussed about the Tables to be made in the database.

3- Agenda Points discussed in meeting (in detail)

Showed the recent work of the project.

4- Next Meeting for this project

We will decide as per our feasibility.

Supervisor's Signature: _____

FYP Project Meeting

Minutes of Meeting

Meeting Date: 17/04/2025
Meeting Location: faculty Room
Meeting Time: 1:40

Project Title: A website for DOC Hamdard University
Project Code: FYP-001/FL24

1- List of Participants

Name	Project Role
Shahmeer Abid	Front-end
Syeda Noreen Zahra	Database
Hamza Sheikh	Backend

2- Meeting Agenda

Showed the demo of the project.

3- Agenda Points discussed in meeting (in detail)

Showed the demo of the project. Took ideas about the database tables and integration from the backend

4- Next Meeting for this project

We will decide as per our feasibility.

Supervisor's Signature: _____

FYP Project Meeting

Minutes of Meeting

Meeting Date: 08/05/2025
Meeting Location: faculty Room
Meeting Time: 1:40

Project Title: A website for DOC Hamdard University
Project Code: FYP-001/FL24

1- List of Participants

Name	Project Role
Shahmeer Abid	Front-end
Syeda Noreen Zahra	Database
Hamza Sheikh	Backend

2- Meeting Agenda

Discuss about Finalization of database.

3- Agenda Points discussed in meeting (in detail)

Completing the design and development phases, ensuring all tables, relationships, constraints, and indexes are properly defined.

4- Next Meeting for this project

We will decide as per our feasibility.

Supervisor's Signature: _____

FYP Project Meeting

Minutes of Meeting

Meeting Date: 22/05/2025
Meeting Location: faculty Room
Meeting Time: 1:40

Project Title: A website for DOC Hamdard University
Project Code: FYP-001/FL24

1- List of Participants

Name	Project Role
Shahmeer Abid	Front-end
Syeda Noreen Zahra	Database
Hamza Sheikh	Backend

2- Meeting Agenda

Discuss about the Finalization of the project and review

3- Agenda Points discussed in meeting (in detail)

Discuss about the Finalization of the project and review.

4- Next Meeting for this project

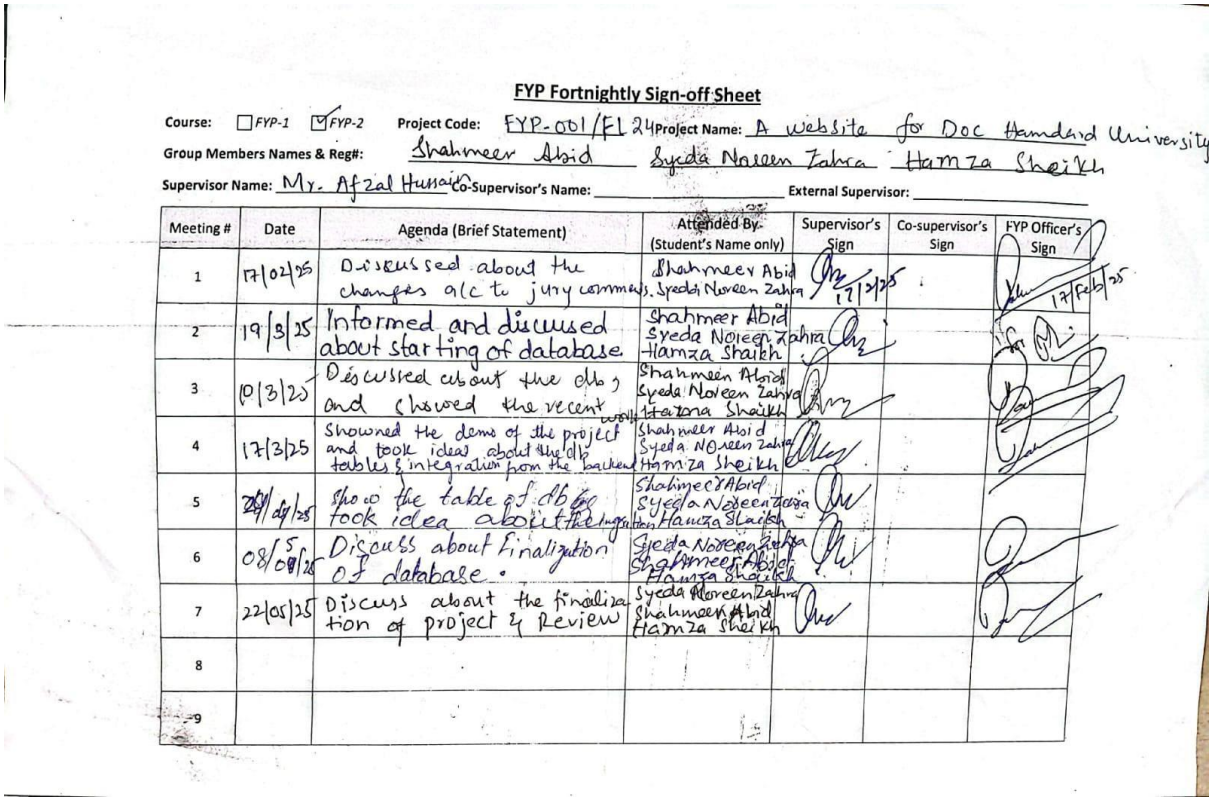
Last meeting of the project.

Supervisor's Signature: _____

A8. DOCUMENT CHANGE RECORD

Date	Version	Author	Change Details
17-01-2025	01	Shahmeer	Removed Student record
02-07-2025	02	Shahmeer	Removed Forms, Library, Complain box, Timetables

A9. PROJECT PROGRESS



FYP Fortnightly Sign-off Sheet

Course: ☐ FYP-1 ☑ FYP-2 Project Code: FYP-001/FL24 Project Name: A website for Doc Hamdard University

Group Members Names & Reg#: Shahmeer Abid Syeda Noreen Zahra Hamza Sheikh

Supervisor Name: Mr. Afzal Husain Co-Supervisor's Name: _____ External Supervisor: _____

Meeting #	Date	Agenda (Brief Statement)	Attended By (Student's Name only)	Supervisor's Sign	Co-supervisor's Sign	FYP Officer's Sign
1	17/02/25	Discussed about the changes a/c to jury comments	Shahmeer Abid Syeda Noreen Zahra	17/2/25		17/Feb/25
2	19/3/25	Informed and discussed about starting of database	Shahmeer Abid Syeda Noreen Zahra Hamza Shaikh			
3	10/3/25	Discussed about the db, and showed the recent work	Shahmeen Abid Syeda Noreen Zahra Hamza Sheikh			
4	17/3/25	Showned the dems of the project and took ideas about the db tables & integration from the backend	Shahmeer Abid Syeda Noreen Zahra Hamza Sheikh			
5	24/03/25	Showed the table of db & took idea about the integration	Shahmeer Abid Syeda Noreen Zahra Hamza Shaikh			
6	08/05/25	Discuss about finalization of database.	Syeda Noreen Zahra Shahmeer Abid Hamza Sheikh			
7	22/05/25	Discuss about the finalization of project & Review	Syeda Noreen Zahra Shahmeer Abid Hamza Sheikh			
8						
9						