Content Management System

23-FYP-205

Session 2019-2023

BACHELOR OF SCIENCE

IN

SOFTWARE ENGINEERING

SUBMITTED BY

Mukarram Shahbaz 19-NTU-CS-1117

Wasi Ur Rehman Qamar 19-NTU-CS-1130

SUPERVISED BY

Mr. Muhammad Shahid

Ma’am Humael Hussain

Department of Computer Science

National Textile University, Faisalabad

CERTIFICATION

DECLARATION

We hereby declare that we completely write this document, and it is totally our effort and none of anyone from outside of our group has copied it. This report is purely written in a technical way in accordance with our project.

Group Members:

Mukarram Shahbaz _____ _____

19-NTU-CS-1117 Signature Date

E-Mail: mukarramshahbaz8@gmail.com

Wasi Ur Rehman Qamar _____ _____

19-NTU-CS-1130 Signature Date

E-Mail: waasiqamar@gmail.com

Acknowledgement

More than anything else, For the sake of Allah, the Most Advantageous, the Most Benevolent. Every one of the gestures of recognition also, thanks be to Allah. A ton of adoration for our cherished Heavenly Prophet MUHAMMAD (S.A.W), his direction assists us with getting the correct way. From that point onward, we additionally gladly announce that the achievement boundaries of this venture are the vast supplications and backing of our folks. We likewise need to recognize the endeavors of our supervisor, who helped us a lot in the project, particularly in the documentation. Each time we need to meet him, he is continuously there for us with good thoughts, conversation and ideas that help us a great deal in fruition of our project and documentation. We likewise need to thank the individuals who helped us in various phases of the venture. We are exceptionally appreciative to the Program Director, Bachelor of Software Engineering, and final year project convener for outrageous strength, mindful, persistence, delicacy, liberality, extraordinary mindfulness and for propelling us to do something remarkable and progressive.

Presently venturing ahead, we wish to pay our glow of appreciation and gratefulness to our noteworthy supervisors who delivered their benevolent administrations to help us in various parts of the trouble spot. We are appreciative to our most regarded educators Mr. Muhammad Shahid and Mr. Muhammad Nouman for being dependably some assistance in all respects of programming standards, this aided us in all advancement levels of the undertaking also, particularly for making us ready to push ahead in the field of computer programming furthermore, plan. We gladly proclaim that the main driver of our outcome in the field of programming designing are our instructors by their strong disposition for continuously being a making a difference hand for us. We cannot overlook the help of the IT administrations division. Particularly the IT administrations staff for giving us a spot to sit and work in the IT division and the related help on request. Their aiding demeanor was amazing. Finally, yet all the same not the least, we additionally need to thank our class colleagues particularly for sacrificial assistance, counsel, and ideas in various stages going from elicitation to organization.

Abstract

Over the past seven years, NTU's prior content management system (CMS) has presented several difficulties. Outdated technology, limited functionality, and difficult user experience have hampered the effective administration and publication of digital information. Information upgrading has been challenging for faculty, staff, and students due to a lack of sophisticated content editing tools, efficient workflows, and centralized digital asset management. Due to these restrictions, there have been delays in updating material across the university's web presence and problems with productivity.

Our solution integrates a master panel and a user panel to address the issues of controlling user access to website material and giving staff members a user-friendly platform to update and alter content.

Administrators can centrally manage user access permissions through the master panel. It gives administrators the power to manage user roles, provide rights, and impose access restrictions on website areas. As a result, only those who are authorized to do so can access and alter content according to their assigned roles.

The user panel features an easy-to-use interface and is tailored primarily for staff workers. Employees can view their customized dashboard in the user panel by logging in with their credentials. They can then quickly develop, edit, and publish content in their assigned areas. The user panel streamlines content administration so that staff employees can concentrate on updating and changing material without having to have technical knowledge.

Our university gains a streamlined and effective content management system by deploying this solution. The master panel guarantees adequate user rights and access management, boosting security and data privacy. Staff employees are given access to a user-friendly platform through the user panel, which boosts their productivity and makes it possible for them to keep content current. Given the circumstances, this solution offers a thorough and user-centered approach to managing website material, leading to a more effective and efficient content management procedure for our university.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CMS	Campus Management System
SDLC	Software Development Life Cycle

CHAPTER 1

1. Introduction

An academic institution or college's many administrative and academic responsibilities can be streamlined and automated with the help of a campus management system. It has several modules that can aid with managing jobs including scheduling classes, managing the faculty, allocating resources, and more. A centralized platform for managing all facets of campus operations, from financial and administrative responsibilities to academic and research endeavors, is often offered by campus management systems. They frequently contain tools for managing campus facilities, equipment, and other resources in addition to features like online registration, grade management, and student record monitoring.

Campus management systems can assist universities and colleges in increasing productivity, decreasing administrative workload, and giving students, teachers, and staff a more streamlined and integrated experience. Additionally, they can offer insightful data reporting and analysis to assist organizations in making data-driven decisions and enhancing performance over time. Campus management systems are becoming more affordable and accessible for a larger range of institutions, from tiny colleges to huge universities, as cloud-based solutions become more widely available.

1.1. Dynamically created websites

The process of building websites using server-side computer languages and technologies that dynamically produce web pages based on user requests and other factors is known as dynamic website generation or creation. A dynamic website allows for more flexible and adaptable user experiences because material is generated on the fly. The opposite of this is a static website, which is made up of HTML and other static content files and is served to the user directly from the server. Several technologies, including PHP, Java, Python, Ruby on Rails, and others, can be used to create dynamic websites. These tools make it possible to create dynamic web pages that can be altered in response to user input, database queries, and other elements.

In general, dynamic website creation is a key technique for building contemporary, user-friendly websites that are simple to customize and manage. It is a crucial tool for many companies, institutions, and groups, including colleges and universities, which need to manage complicated content and offer unique user experiences to employees, teachers, and students.

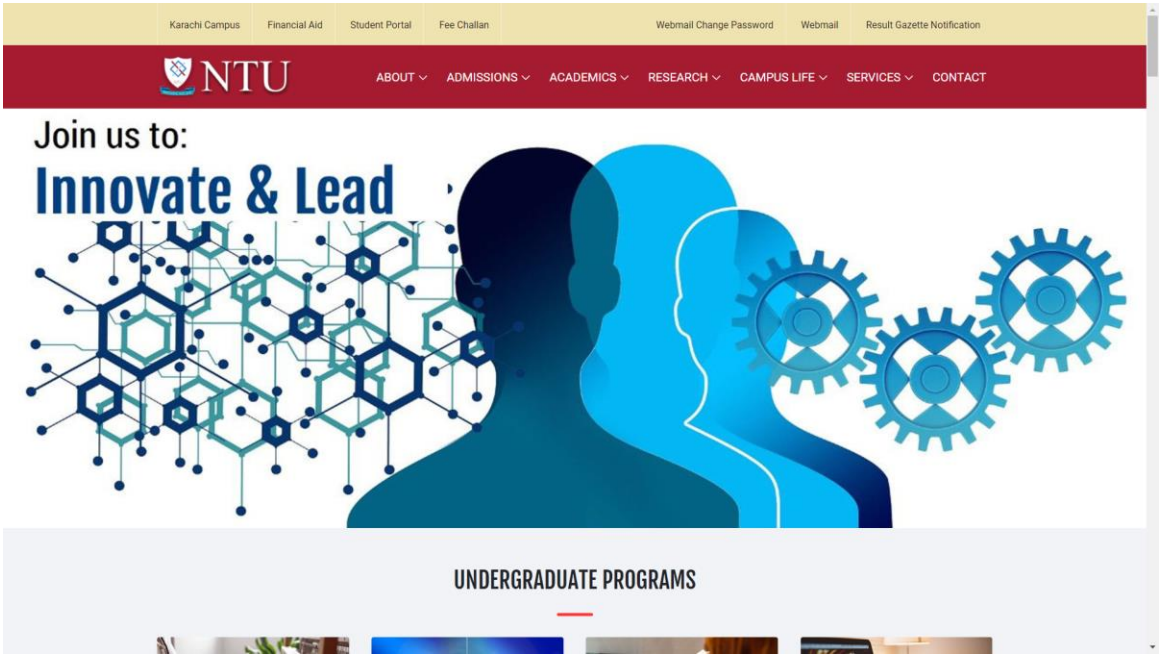

Figure 1.1: Currently working dynamically created website on old technology [1]

1.2. Problem Statement

The current campus content management system at National Textile University has been in use for the past 7 years. It has become increasingly difficult to manage and update the system, leading to challenges for students, faculty, and staff in accessing information and resources.

The lack of these essential functionalities has led to poor user experience for students, faculty, and staff, and has caused inefficiencies in campus operations. This has resulted in increased administrative workload, longer turnaround times for important tasks, and decreased overall satisfaction with the system.

To address these challenges, we are developing a new campus content management system that leverages modern technology and incorporates key functionalities that were missing in the previous system. The new system will provide a dynamic website that is mobile-responsive, customizable, and accessible on different devices. The new campus content management system will help address these issues and provide a more streamlined and user-friendly system for the university community.

1.3. Purpose

The purpose of creating a new campus content management system is to overcome the shortcomings of the existing one, which is based on antiquated technology and lacks key components needed for contemporary content management. The new system will

offer an updated and more effective platform for handling and distributing digital content, enabling better user experiences, more efficient processes, and enhanced cooperation and communication among campus residents.

1.4. Project Goals

To get over the constraints of the old system and create a dynamic website utilizing server-side programming languages, a new campus content management system is being developed. The new system will offer a more adaptable, configurable, and effective platform for storing and providing digital information to the university community, allowing for a better user experience, and streamlined workflows. It will do this by utilizing contemporary technologies and programming languages like Python (Django).

1.5. Objectives

The goals of creating a content management system to enable it on campus, get over previous restrictions, and create a website dynamically using a programming language are:

- To give the campus community a centralized and effective platform for managing digital content.
- To use current technologies and programming languages to build a flexible and adaptable system that can respond to the campus's changing needs.
- To incorporate new features and functionality including social media integration, online forms, and customized landing pages to get beyond the constraints of the outdated system.
- To improve the user experience for teachers, staff, and students by making it simple and quick to access crucial data, resources, and services via an engaging website.
- To create a more effective and productive work environment by streamlining workflows and enhancing collaboration and communication among campus residents.

1.6. Project Scope

The university is creating a campus content management system that will allow for the handling of enormous amounts of data and traffic, a user-friendly interface, and the use of contemporary programming languages and tools to create dynamic, adaptable websites. In addition to collaboration and communication among academics, staff, and students, the system should facilitate content production, storage, sharing, and

distribution across departments and campuses. The system also includes crucial elements like analytics and reporting, interaction with developing technologies, security and access control, and analytics.

1.7. Proposed Solution

Our solution integrates a master panel and a user panel to address the issues of controlling user access to website material and giving staff members a user-friendly platform to update and alter content.

Administrators can centrally manage user access permissions through the master panel. It gives administrators the power to manage user roles, provide rights, and impose access restrictions on website areas. As a result, only those who are authorized to do so can access and alter content according to their assigned roles.

The user panel features an easy-to-use interface and is tailored primarily for staff workers. Employees can view their customized dashboard in the user panel by logging in with their credentials. They can then quickly develop, edit, and publish content in their assigned areas. The user panel streamlines content administration so that staff employees can concentrate on updating and changing material without having to have technical knowledge.

1.8. Project Scheduling

A crucial component of effective project management is having a detailed timetable of the project's evolution. You can make sure that every activity is carried out on schedule, within budget, and in accordance with the project's specifications if you have a detailed timeline. This can help to guarantee that the project is finished as planned and to an acceptable standard. Gantt charts are used to show the project's general schedule as well as the start and end periods and duration of each activity. As a result, a chronology for the project is shown in figures.

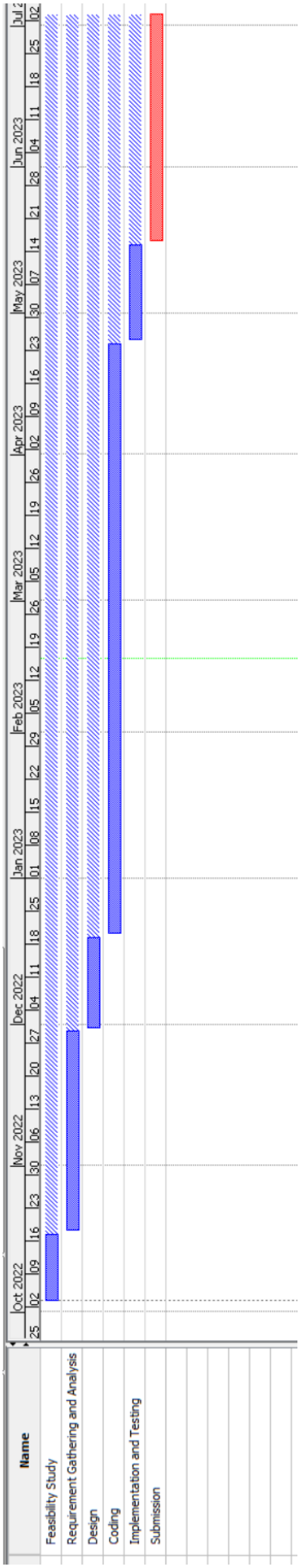

Figure 1.2: Gantt Chart

Feasibility Study
Duration 11 days?
Start 10/3/22 8:00 AM
Finish 10/17/22 5:00 PM

Requirement Gathering and ...
Duration 31 days?
Start 10/18/22 8:00 AM
Finish 11/29/22 5:00 PM

Design
Duration 14 days?
Start 11/30/22 8:00 AM
Finish 12/19/22 5:00 PM

Coding
Duration 90 days?
Start 12/20/22 8:00 AM
Finish 4/24/23 5:00 PM

Implementation and Testing
Duration 15 days?
Start 4/25/23 8:00 AM
Finish 5/15/23 5:00 PM

Submission
Duration 35 days?
Start 5/16/23 8:00 AM
Finish 7/3/23 5:00 PM

Figure 1.3: Gant Chart Deadlines

1.9. Risks and Risk mitigation

There are several common risks that can occur while developing a campus content management system, including:

1.9.1. Technical Risks Time and Cost Overruns

There may be technical limitations or difficulties while integrating modern technology and migrating from an old system.

1.9.2. Data Security Risks

A campus management system contains sensitive information about students, faculty, and staff. Thus, ensuring data security is essential.

CHAPTER 2

2. Literature Review

A content management system (CMS) is a piece of software created with the purpose of organizing, managing, and distributing digital content on a college or university campus. Higher education institutions are progressively implementing campus content management systems to enhance the management and delivery of digital information, according to a survey of the literature on the subject. According to the research, campus content management systems can increase productivity and lower expenses related to the production and distribution of digital material. Additionally, the technologies may promote increased faculty, staff, and student collaboration and communication, which will enhance academic results. Easy usage, flexibility, scalability, and customization are among the crucial characteristics that a successful campus content management system must have, according to several studies. Additionally, according to the literature, the system needs to be safe with strong user identification and access controls. User acceptance is a crucial factor in the design and implementation of a campus content management system. To ensure that the system is used successfully and to its full capacity, studies have underlined the significance of user support and training. In general, the research on campus content management systems points to these systems' potential to revolutionize how universities and colleges organize and disseminate digital content, resulting in increased effectiveness and improved academic results. The design, execution, and user acceptance of these systems are only a few of the variables that will determine if they are successful.

2.1. Related Work

There are several related works to a campus content management system that create dynamic websites through code. Here are a few examples:

2.1.1. WordPress Multisite: A single installation of the popular content management system WordPress enables the building of several websites. It is possible to build a network of websites using WordPress Multisite that all use the same database and codebase, making it simple to administer and update all the websites at once. [2]

2.1.2. Django CMS: The Django CMS is a content management system that is part of the Django web framework. It offers a user-friendly interface for content management and permits the coding of dynamic websites. [3]

2.1.3. Contentful: A robust API for managing and producing website content is offered by the content management system Contentful. It makes it possible for programmers to create dynamic websites and makes it simple to collaborate on and localize content. [4]

2.1.4. Squarespace: Squarespace is a website builder that gives users the option to create dynamic websites using code as well as a user-friendly interface. It offers built-in facilities for e-commerce, blogging, and other site elements and enables simple modification. [5]

2.2. Comparison

Content Management System	**Key Features**	**User Interface**	**Coding Flexibility**
WordPress Multisite	**1.** Enables building multiple websites using a single installation of WordPress. **2.** Easy administration and updates for all websites. **3.** Shared database and codebase.	User-friendly	Moderate
Django CMS	**1.** Part of the Django web framework, allowing dynamic website development. **2.** User-friendly content management interface.	User-friendly	High
Contentful	**1.** Robust API for managing and producing website content. **2.** Supports dynamic website creation and easy collaboration on content. **3.** Localization support.	Developer-focused	High

Squarespace	**1.** Website builder with options for dynamic website creation using code and user-friendly interface. **2.** Built-in facilities for e-commerce, blogging, and other site elements. **3.** Easy customization.	User-friendly	Moderate

2.3. Reason to Develop

There are many advantages to creating a campus content management system as a senior project. It gives students the chance to put the knowledge and skills they have acquired throughout their studies to use in a practical, real-world setting. In addition, it enables the investigation and application of novel and developing technologies, which is essential for keeping up with the rapidly changing field of web development. Furthermore, creating one might make a significant contribution to the academic community. Campus content management systems are useful and practical tools for universities and educational organizations.

CHAPTER 3

3. System Requirements

The system requirements for creating a campus content management system are covered in this chapter. The system requirements specify what the software is expected to do and what hardware and software resources are required to accomplish those goals, making them a crucial component of the software development process. To make sure the system satisfies the demands of its users, it is crucial to have a comprehensive grasp of the system requirements before beginning the development process.

Depending on the demands of the university, different systems may be needed for a campus content management system. A server architecture, a database management system, web development programming languages and frameworks, and user authentication and authorization systems are a few typical requirements. Additionally, the system might need to be safe, scalable, and capable of handling huge traffic volumes.

3.1. Functional Requirements

Functional requirements outline what the system should perform and establish the expected behavior of the system. They outline the precise attributes and capabilities the system must possess to satisfy users' and stakeholders' needs. Software development is dependent on functional requirements, which direct the system's design, development, and testing. Usually stated in a functional requirements specification, they serve as a benchmark for evaluating the system's performance.

3.1.1. Master Panel

3.1.1.1. User Management

The master panel shall allow administrators to add users who will have access to the CMS. Administrators shall be able to assign specific roles and permissions to users.

3.1.1.2. Menu and Group Management

The master panel shall allow the creation of menus and submenus for website navigation. Administrators shall be able to create groups to categorize menus. Menus shall be assigned to specific groups.

3.1.1.3. User Group Management

The master panel shall provide functionality to create user groups. User groups shall be assigned to specific groups (menus) for access control.

3.1.1.4. Common Sections on All Websites

The master panel shall ensure that certain sections (Notice Board, Tenders, Jobs, Events, News Updates, Slider) are displayed on all websites. Users shall have the ability to customize the content and visibility of these common sections. The slider section shall provide options to select which sites the content should be visible on and set an expiry date for hiding the content.

3.1.2. User Panel

3.1.2.1. Login

The user panel shall provide a login mechanism for users to access the CMS using their credentials.

3.1.1.1. Website Selection

After logging in, the user panel shall display a screen showing the websites assigned to the user by the master panel.

3.1.1.2. Content Editing

The user panel shall allow users to select a specific website and edit its content.

3.2.Non-Functional Requirements

Functional requirements outline what the system should perform and establish the expected behavior of the system. They outline the precise attributes and capabilities the system must possess to satisfy users' and stakeholders' needs. Software development is dependent on functional requirements, which direct the system's design, development, and testing. Usually stated in a functional requirements specification, they serve as a benchmark for evaluating the system's performance.

3.2.1. User-Friendly Interface

Both the master panel and the user panel shall have intuitive and user-friendly interfaces to facilitate easy navigation and content management.

3.2.2. Security

The CMS shall implement robust security measures to protect user data and prevent unauthorized access.

3.2.3. Performance

The CMS shall be designed and optimized for efficient performance, ensuring fast loading times and responsiveness.

3.1. Use Case Diagram

Our Content Management System's (CMS) use case diagram illustrates the interactions between actors and the system, making it easy to understand the features it offers. The main actor in the diagram is the Admin. With the Master Panel at their disposal, the admin can control several CMS functions, such as user access, website creation, menu organization, and shared areas. These use case diagrams are a useful tool for our development team to comprehend and effectively express the needs of our CMS.

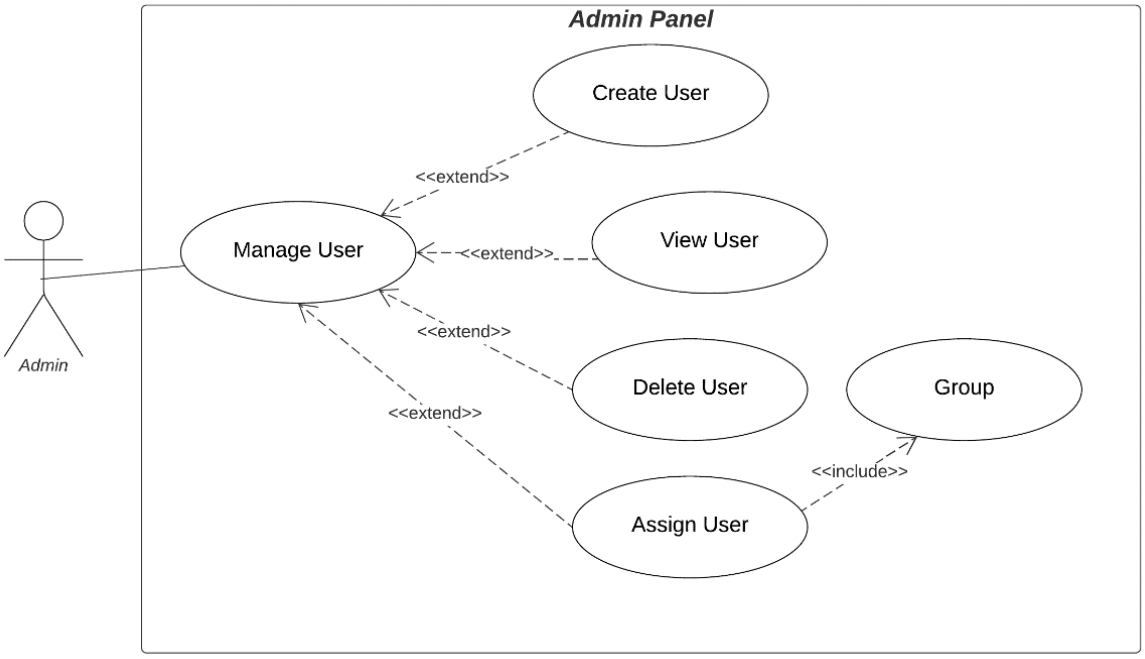

Figure 3.1: Managing User Use Case Diagram

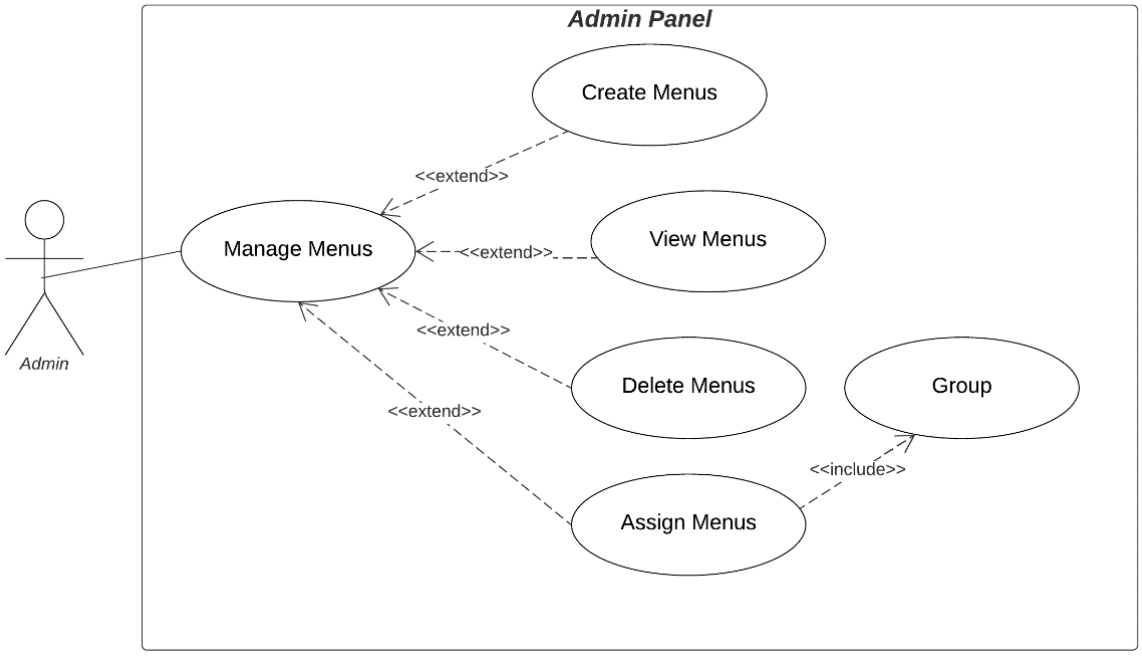

Figure 3.2: Managing Menus Use Case Diagram

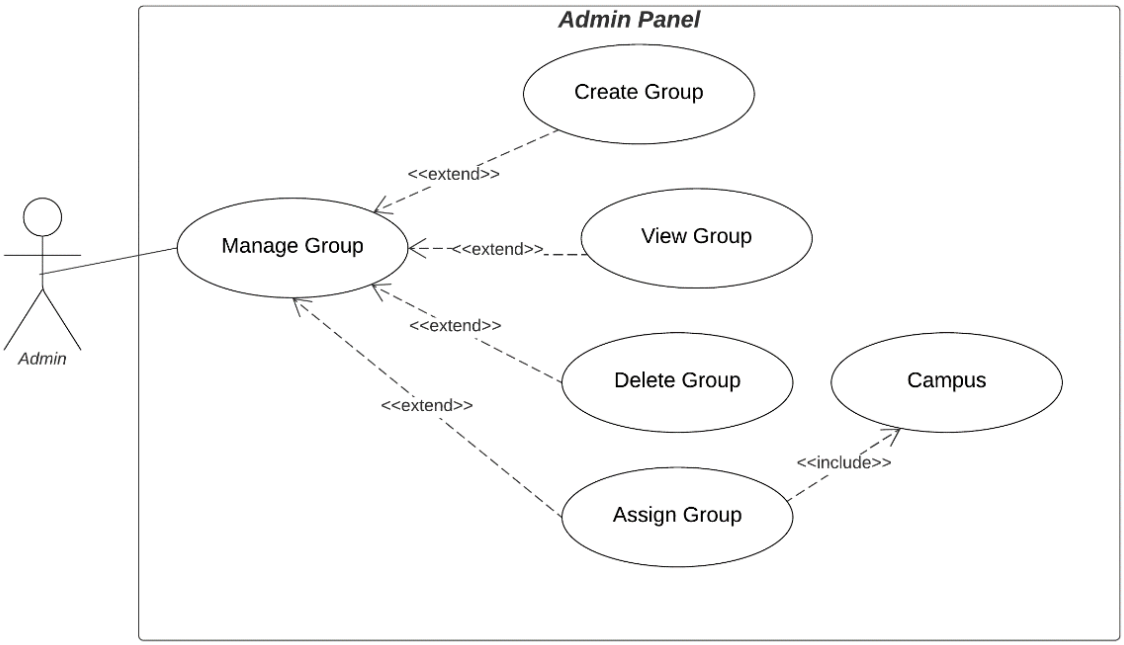

Figure 3.3: Managing Groups Use Case Diagram

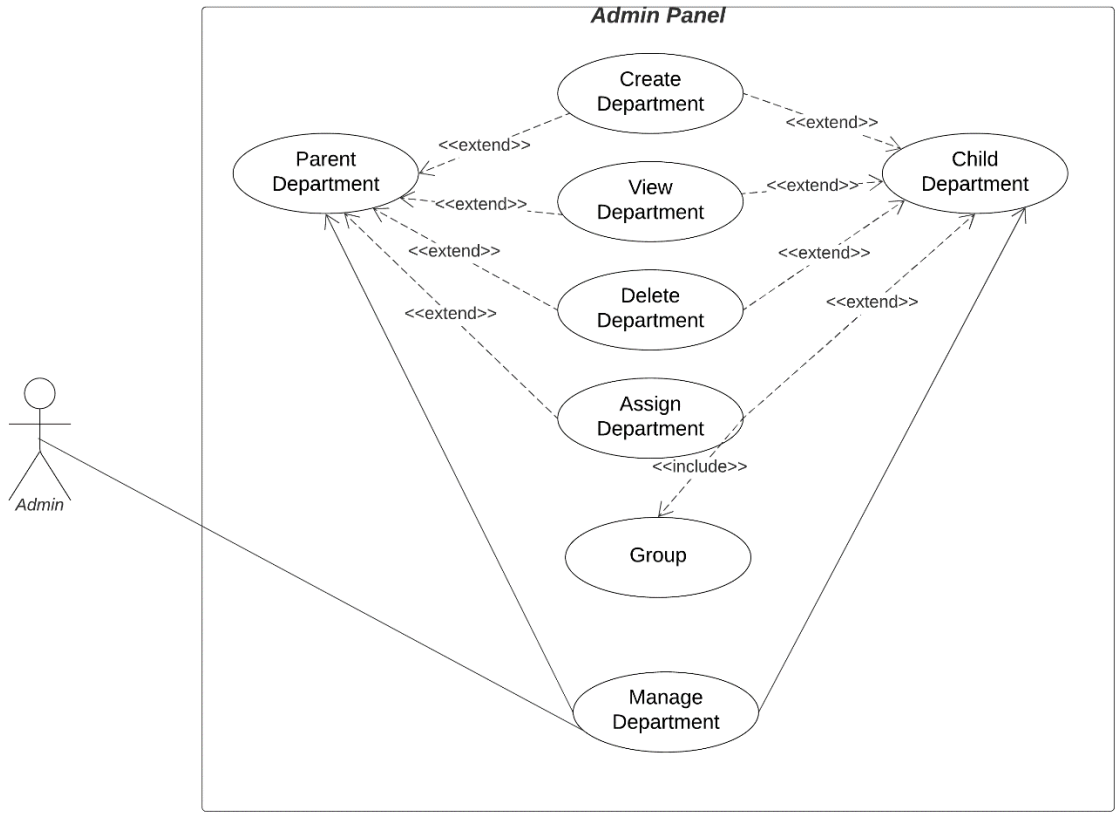

Figure 3.4: Managing Department Use Case Diagram

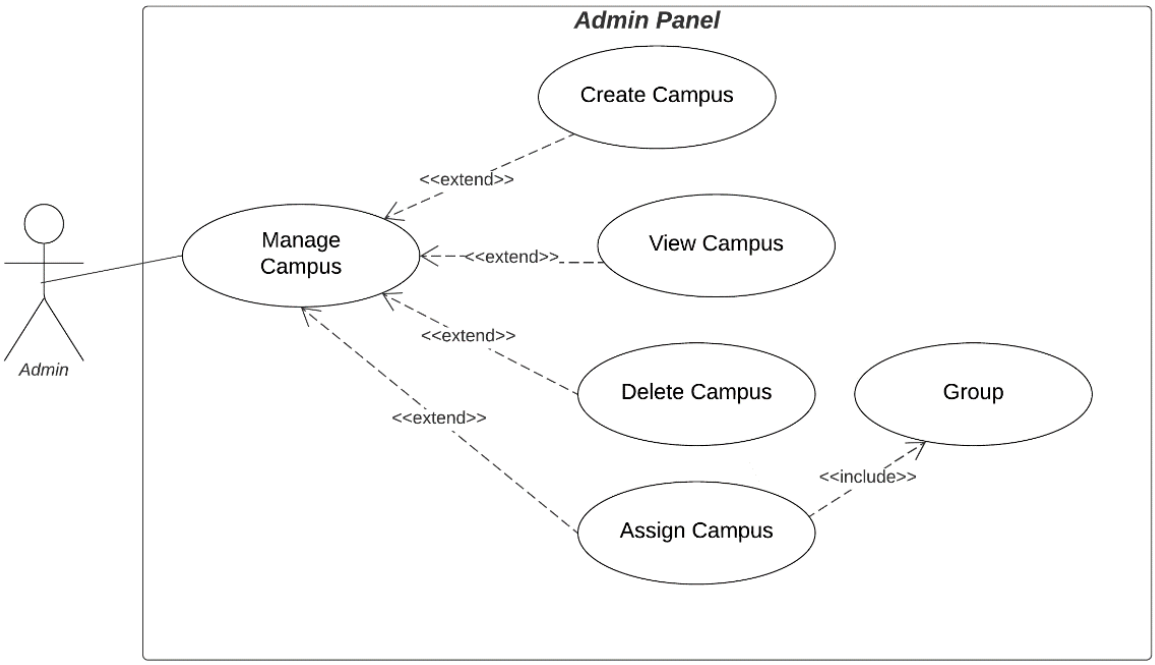

Figure 3.5: Managing Campus Use Case Diagram

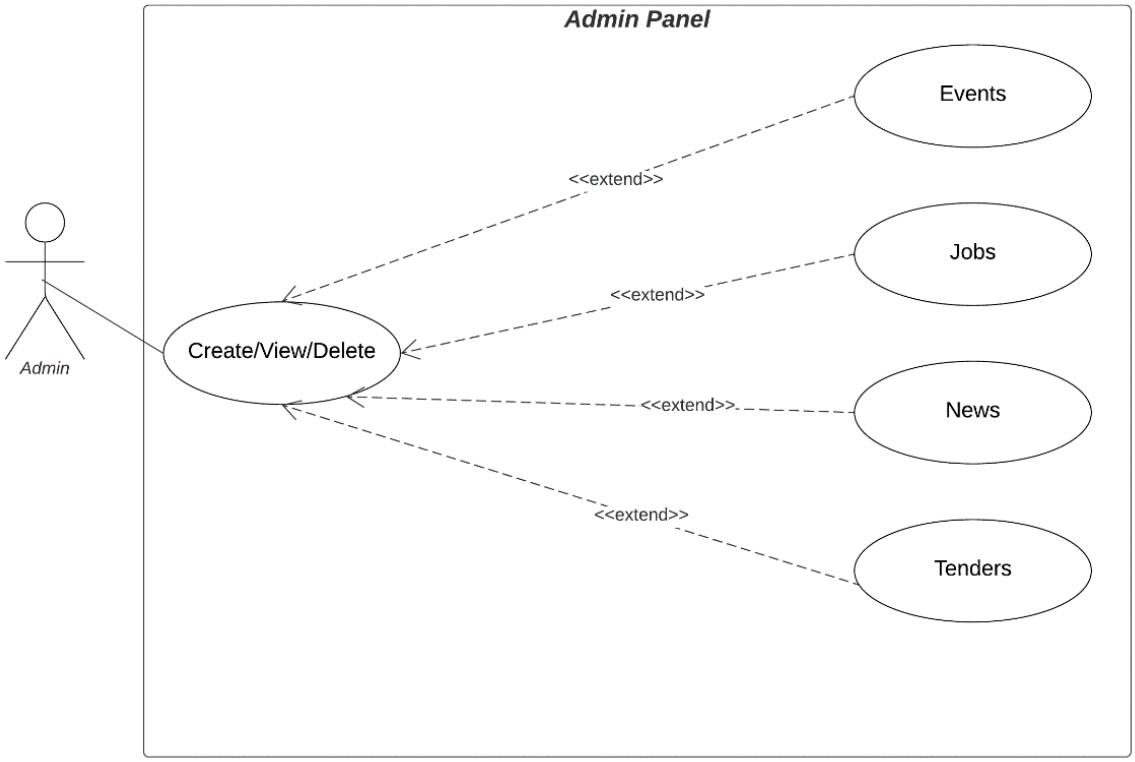

Figure 3.6: Managing Common Section Use Case Diagram

3.2. Use Case Description

These use case descriptions provide a comprehensive overview of the interactions and tasks within our CMS, ensuring that users can efficiently manage website content and administrators can effectively oversee user access and system administration.

3.2.1. Use Case Group

Table 3.1: CRUD operation on Group

ID	1
Name	CRUD Operation on Group
Actor	Admin
Description	Admin will be able to perform CRUD on group
Pre-Condition	Group must not exist already
Post-Condition	CRUD operation successful

3.2.2. Use Case Menu

Table 3.2: CRUD operation on Menu

ID	2
Name	CRUD Operation on Menu
Actor	Admin
Description	Admin will be able to perform CRUD on menu
Pre-Condition	Menu must not exist already
Post-Condition	CRUD operation successful

3.2.3. Use Case Event Category

Table 3.3: CRUD operation on User

ID	3
Name	CRUD Operation on User
Actor	Admin
Description	Admin will be able to perform CRUD on user
Pre-Condition	User must not exist already
Post-Condition	CRUD operation successful

3.2.4. Use Case Event Category Content

Table 3.4: CRUD operation on Parent Department

ID	4
Name	CRUD Operation on Parent Department
Actor	Admin
Description	Admin will be able to perform CRUD on department
Pre-Condition	Parent or Child department must not exist already
Post-Condition	CRUD operation successful

3.2.5. Use Case Master User

Table 3.5: CRUD operation on Child Department

ID	5
Name	CRUD Operation on Child Department
Actor	Admin
Description	Admin will be able to perform CRUD on department
Pre-Condition	Parent department must exist already
Post-Condition	CRUD operation successful

3.2.6. Use Case News Update

Table 3.6: CRUD operation on Campus

ID	6
Name	CRUD Operation on Campus
Actor	Admin
Description	Admin will be able to perform CRUD on campus
Pre-Condition	Campus must not exist already
Post-Condition	CRUD operation successful

3.2.7. Use Case Society

Table 3.7: CRUD operation on Common Sections

ID	7
Name	CRUD operation on common sections
Actor	Admin
Description	Admin will be able to perform CRUD on Events, Jobs, News, Tenders.
Pre-Condition	Events, Jobs, News, Tenders must not exist
Post-Condition	CRUD operation successful

CHAPTER 4

4. Methodology

The general technique or strategy used to plan and carry out a project is referred to as a methodology. It includes a collection of guidelines, procedures, methods, and instruments that direct the project team through every phase of the project lifecycle. The methodology used for a project determines how work will be organized, how decisions will be made, how resources will be allocated, and how risks will be managed, all of which have a substantial bearing on the project's success. The best methodology to use relies on the project's complexity, nature, organizational culture, and project management capabilities. There are many different techniques available, each with its own advantages and disadvantages.

4.1. Project Planning

The process of generating a schedule outlining the tasks, materials, and deadline required to successfully finish a project is known as project planning. It entails establishing the project's goals, figuring out the tasks needed to get there, distributing those duties to team members, calculating the time and money needed to finish each work, and developing a schedule for the project. To guarantee that the project is finished on schedule, within budget, and in accordance with its goals, effective project planning is crucial.

4.2. Methodology for software development

Software development approaches include Waterfall, Agile, Scrum, Spiral, and others. It is crucial to select the technique that is most appropriate for the project objectives because each methodology has advantages and disadvantages of its own. The software development process, encompassing the phases of planning, design, implementation, testing, and deployment, is guided by the methodology that has been chosen.

4.2.1. Waterfall

Software development is done in a linear, sequential manner using the Waterfall methodology. It is divided into several discrete phases, each of which must be finished before moving on to the next. The stages include obtaining requirements, designing, implementing, testing, and maintaining. When needs are clear and a finished product is expected to be delivered as a whole, the waterfall approach is frequently employed. When it comes to flexibility and adaptation to shifting requirements throughout the development process, it does have certain limitations.

4.2.2. Agile

Agile is a software development process that priorities adaptability and cooperation amongst cross-functional teams. It uses an incremental and iterative method to develop with a goal of producing usable software in brief sprints. Customer satisfaction, adapting to change, and ongoing improvement are prioritized throughout the development process by the agile methodology. It is frequently used in software projects where specifications and priorities are fluid or not completely established at the outset.

4.2.3. Spiral

The Agile model's iterative nature and the methodical, regulated elements of the Waterfall model are combined in the software development technique known as the Spiral model. It places a strong emphasis on risk management and priorities seeing and resolving issues early in the development process. The planning, risk analysis, engineering, and assessment phases make up the Spiral model. Until the software is finished, the process spirals back on itself, with each iteration building on the one before.

4.2.4. Scrum

Scrum is an agile software development approach that emphasizes iterative and incremental product delivery. Throughout the development process, it places a strong emphasis on cooperation, adaptability, and constant progress. Scrum teams meet daily to discuss progress and modify their goals between brief sprints, which are typically 1-4 weeks long. The framework comprises distinct artefacts like the product backlog, sprint backlog, and burn-down charts, as well as roles like the scrum master, development team, and product owner.

4.3. Selected methodology

Scrum is the Agile approach that we have chosen for the development of your project. This methodology places a strong emphasis on adaptability and ongoing improvement, making it perfect for your project's requirements.

4.4. Reason for selecting agile methodology

Agile principles Scrum is frequently used for software development projects because it is adaptable to changing needs and flexible. Collaboration and ongoing input between the development team and stakeholders are made possible, which makes the

development process more effective and efficient. Additionally, it promotes an iterative development process, enabling the product to change and advance over time. These elements make it an excellent fit for the creation of a campus content management system, where stakeholder feedback is crucial, and requirements are subject to change.

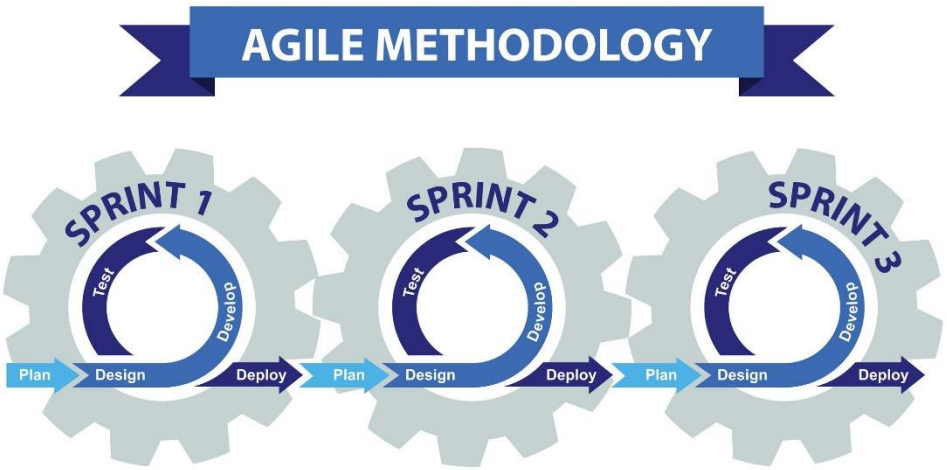

Figure 4.1: Agile Methodology [6]

CHAPTER 5

5. System Architecture

The structural layout of the entire system, including all its modules, components, and relationships, is referred to as system architecture. On the other hand, system diagrams are graphical representations of the system architecture that highlight how the various parts and modules work together to carry out tasks.

5.1. Architecture

Architecture, in general, refers to how a system and its parts are fundamentally organized. To guarantee that a system achieves its intended goal effectively and efficiently, it requires designing and planning the system's general structure, behavior, and functioning. Architecture can refer to a wide variety of systems, including software programmers and computer systems as well as actual physical structures like buildings.

The display layer, the application layer, the data management layer, and the database layer make up the system architecture of a campus content management system.

The user interface is rendered, and user interactions are handled by the presentation layer. Business logic is handled, and user requests are processed by the application layer. The data management layer maintains the integrity of the data and administers it. A database management system can access the structured data that is stored in the database layer.

Diagrams like deployment diagrams, and sequence diagrams can be used to show the system architecture.

5.2. Activity Diagram

The sequential flow of activities and actions within the CMS is depicted by the activity diagram of our system. It illustrates the relationships between various components and the dynamic behavior of the system. The activity diagram offers a visual depiction of the flow of activities and interactions within our CMS, making it easier to comprehend the system's dynamic behavior and sequential order of operations.

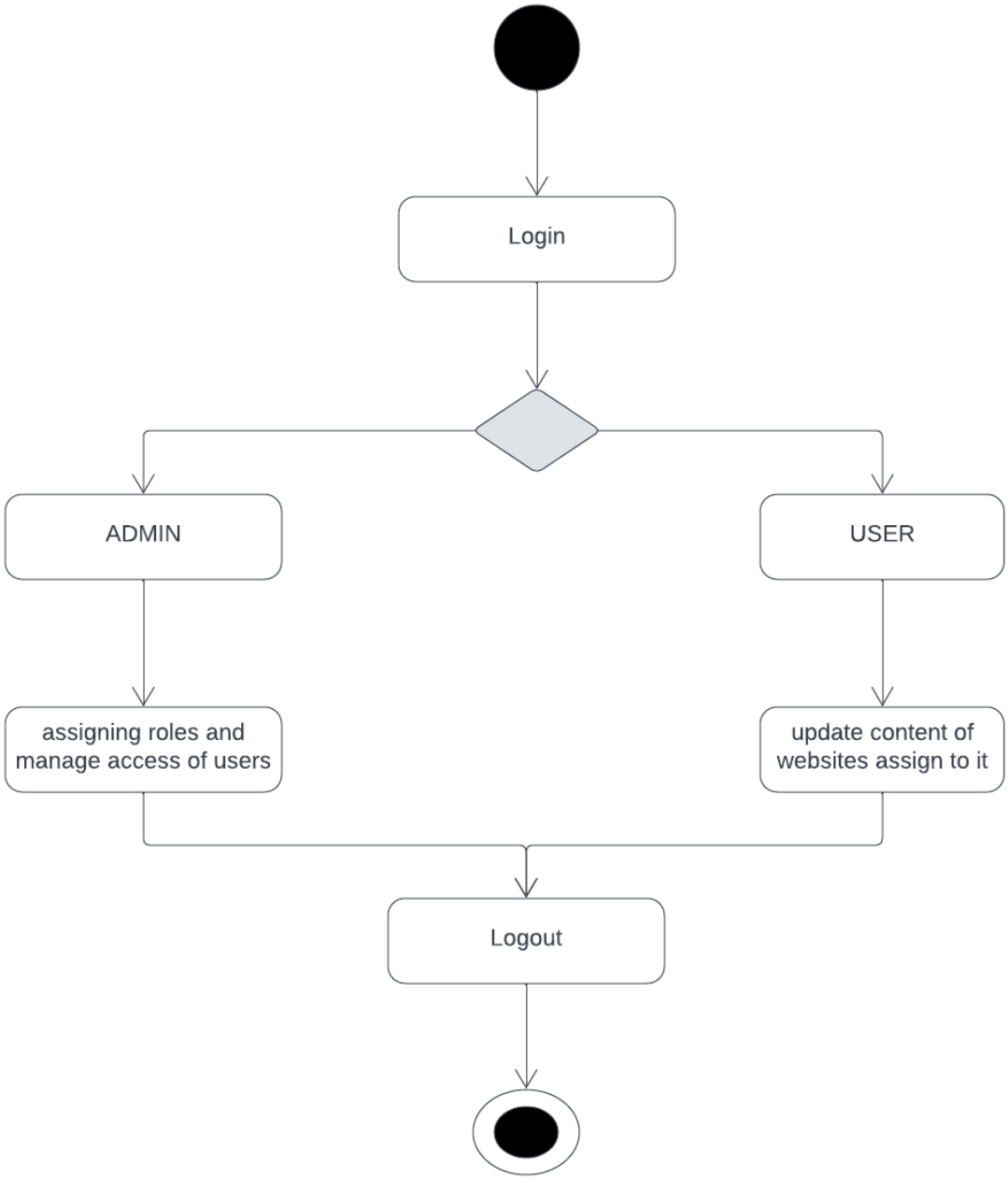

Figure 5.1: Activity Diagram

5.3. Sequence Diagram

The chronological order of interactions between various CMS components and actors is depicted in our system's sequence diagram. It concentrates on message exchanges and their chronological arrangement.

The sequence diagram offers a comprehensive illustration of the orderly progression of interactions within our CMS. It facilitates a clear comprehension of the sequence of

events and the interactions between various components by visualizing the message exchanges between users and the system.

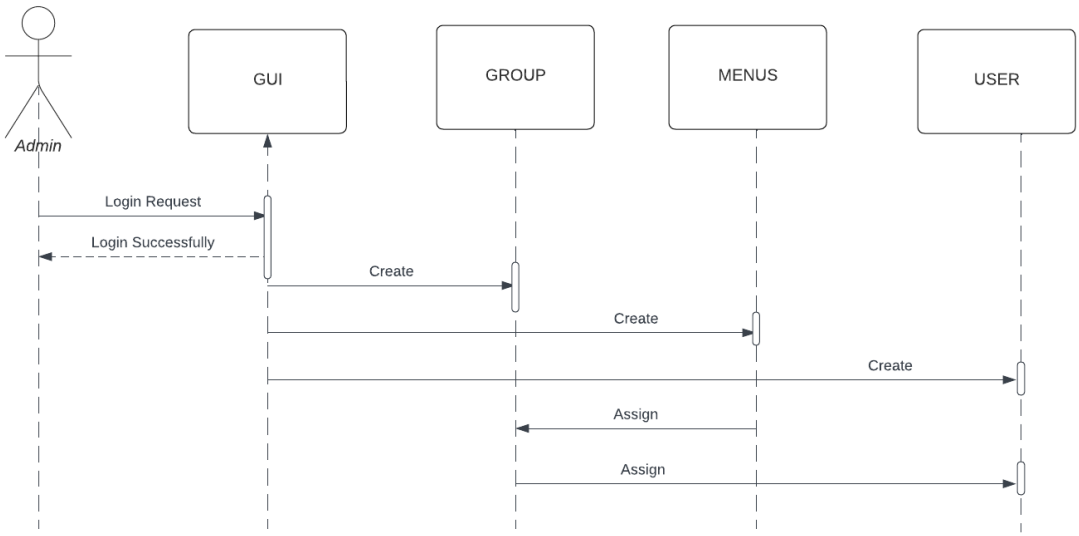

Figure 5.2: Sequence Diagram

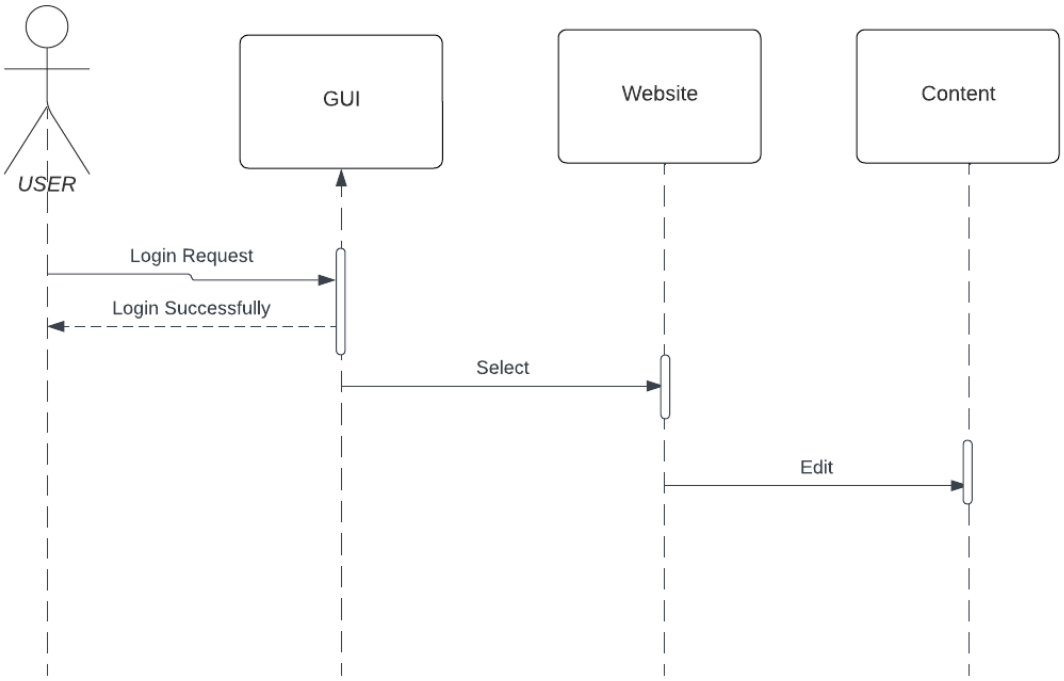

Figure 5.3: User Sequence Diagram

5.4. Deployment Diagram

The physical deployment of hardware and software elements within the environment of our system is shown in the deployment diagram. To demonstrate how the system is implemented and how various pieces interact with one another, it displays how these components are distributed and arranged. The deployment diagram gives a visual depiction of how the various parts of our CMS are set up in the environment of the system. It aids in comprehending the linkages, interactions, and physical infrastructure between various elements, ensuring a seamless and effective system deployment.

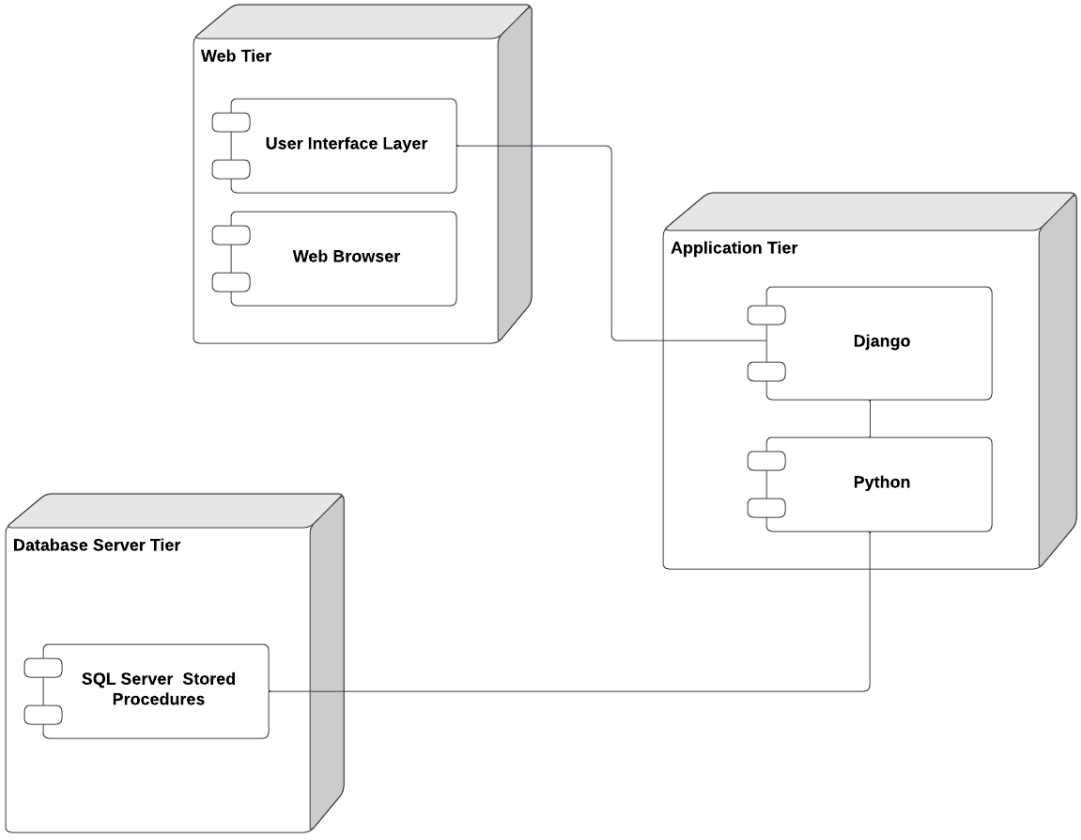

Figure 5.4: Deployment Diagram

CHAPTER 6

6. System Implementation

The process of developing the actual code and constructing the software system to put the software system design into practice is referred to as system implementation. It requires doing a variety of things, including coding, testing, debugging, and deployment. To guarantee that the system is properly implemented and satisfies the demands of the users, this step necessitates careful planning and implementation.

6.1. System Tools and Technology

- Python
- Django
- Microsoft SQL Server

6.2. Python

The simplicity, clarity, and flexibility of Python, a high-level, interpreted programming language, are well known. It is widely utilized in many fields, including artificial intelligence, scientific computing, data analysis, and web development.

6.3. Django

The high-level Python web framework Django uses model-View-Controller (MVC) architecture. It offers a quick and easy solution to create web applications, putting a focus on modularity, reusability, and quick development. The ORM, URL routing, template engine, middleware, and integrated authentication and authorization systems are some of Django's key features. It is frequently employed by developers to create intricate web applications, such as social networks, e-commerce websites, and content management systems.

6.4. Microsoft SQL Server

Microsoft created the relational database management system (RDBMS) Microsoft SQL Server. Data is stored there and retrieved when needed by other software programs. It supports the relational database management language SQL (Structured Query Language), which is used to manage and manipulate databases. Small desktop programs and big enterprise-class databases are only two examples of the many applications that SQL Server can be used for.

6.5. Database Diagram/Design

Figure 6.1: Database Design

6.6. Class Diagram

Figure 6.2: Class Diagram

CHAPTER 7

7. System Testing

After our system has been successfully developed, system testing is extremely important in guaranteeing that all of its components and functionalities have been completely tested and satisfy the required standards, resulting in a dependable and high-quality final product.

7.1. System Testing

System testing is essential for confirming the performance and operation of our CMS (Content Management System). It entails testing the entire system to make sure it complies with the criteria and performs well in the environment for which it is intended. We make use of several testing methodologies, including black box testing, as part of the system testing process.

7.1.1.Black Box Testing

Black box testing is a testing method where the tester is not aware of the CMS's internal structure, design, or implementation specifics. Instead, the tester concentrates on analyzing the system's inputs, outputs, and behavior. The aim is to confirm that the CMS performs as intended, complies with the required standards, and produces the anticipated outcomes.

The tester will create test cases for our CMS's black box testing based on the functional specifications and requirements of the system. Using various inputs, the tester will interact with the CMS and track the relevant outputs. The validation of the CMS's functioning, handling of varied user inputs, and conformance with intended outputs are the main concerns.

We guarantee the testing procedure is impartial by treating the CMS like a black box. We may mimic real-world situations using this method and assess the CMS from the viewpoint of the end user. We can spot potential problems including inaccurate or missing functionality, issues with input validation, handling mistakes, performance snags, and interoperability problems.

We use several approaches, including equivalence partitioning, boundary value analysis, decision table testing, and state transition testing, to effectively execute black box testing. These methods enable us to systematically create test cases that maximize test coverage while covering a variety of usage scenarios.

7.1.1.1. **Equivalence Partitioning:** divides the incoming data into groups or partitions with the expectation that each partition will behave similarly. We may check the functionality of the CMS across several input ranges and make sure that it consistently processes inputs by choosing representative test cases from each partition.

7.1.1.2. **Analysis of Boundary Values:** With analysis of boundary values, we test the limitations and borders of input values. This method seeks to locate any potential problems that might appear at the boundaries of acceptable and unacceptable input ranges. We can find potential mistakes or unexpected behavior in the CMS by choosing test cases at the lower and upper borders, as well as just within and outside those boundaries.

7.1.1.3. **Decision Table Testing:** We can systematically test various combinations of conditions and related actions. We may make sure that the CMS responds appropriately to diverse conditions and business regulations by developing decision tables that map distinct inputs to anticipated outputs. With the use of this technique, we may locate any erroneous or missing decision logic present in the system.

7.1.1.4. **State Transition Testing:** State Transition Testing is a useful technique in systems where the behavior depends on the current state. This technique focuses on testing the transitions between the many states or modes that our CMS may have. We can confirm that the CMS maintains consistency and appropriately manages state changes by considering both valid and invalid state transitions as well as various action sequences.

We may identify a variety of potential problems and evaluate the operation and behavior of our CMS by including these testing methodologies into our black box testing methodology. With the use of these strategies, we can increase test coverage, spot flaws, confirm that requirements are being followed, and improve the CMS's overall quality and dependability.

7.2. Test Cases

Table 7.1: Test Case for Admin Login (Success)

Component Name	Admin Panel
Module Name	Login
Condition being tested	Admin will be able to login
Expected Result	Login successfully
Success Scenarios	Admin will be able to view dashboard
Failure Scenarios	Admin will not be able to view dashboard
Test Result (Pass/ Fail)	Passed

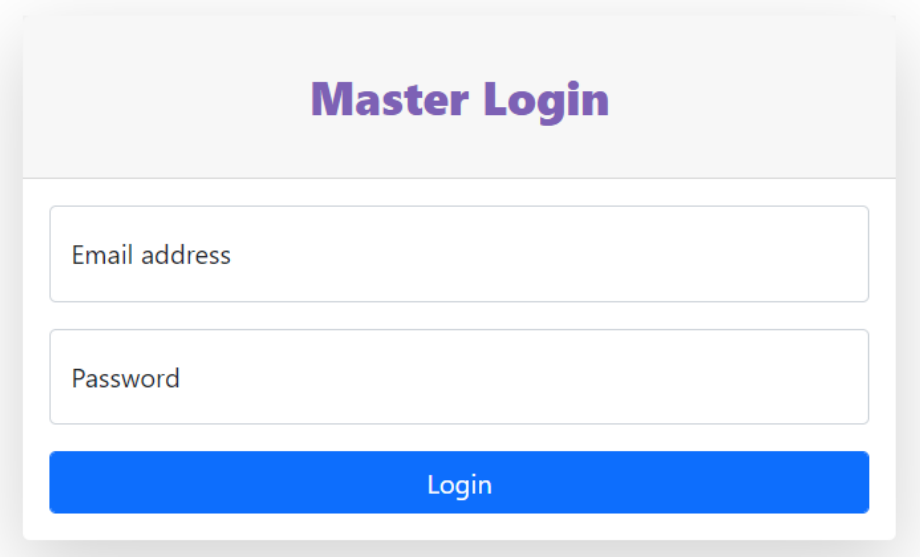

Figure 7.1: Master Panel Login

Table 7.2: Test Case for Creating User

Component Name	Admin Panel
Module Name	Create user
Condition being tested	Admin will be able to create new user
Expected Result	User created successfully
Success Scenarios	User details will be added to database
Failure Scenarios	User will not create
Test Result (Pass/ Fail)	Passed

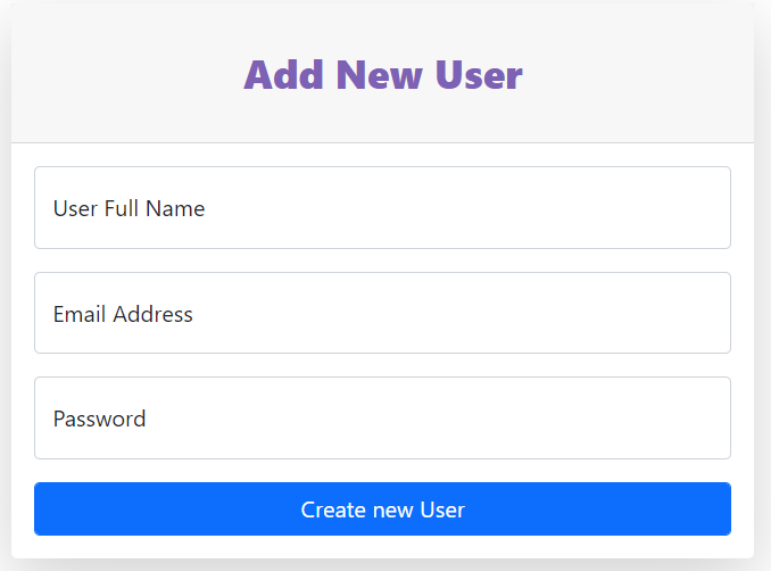

Figure 7.2: Master Panel Creating User

Table 7.3: Test Case for Creating Group

Component Name	Admin Panel
Module Name	Create group
Condition being tested	Admin will be able to create new group
Expected Result	Group created successfully
Success Scenarios	Admin can view created group
Failure Scenarios	Admin cannot view created group
Test Result (Pass/ Fail)	Passed

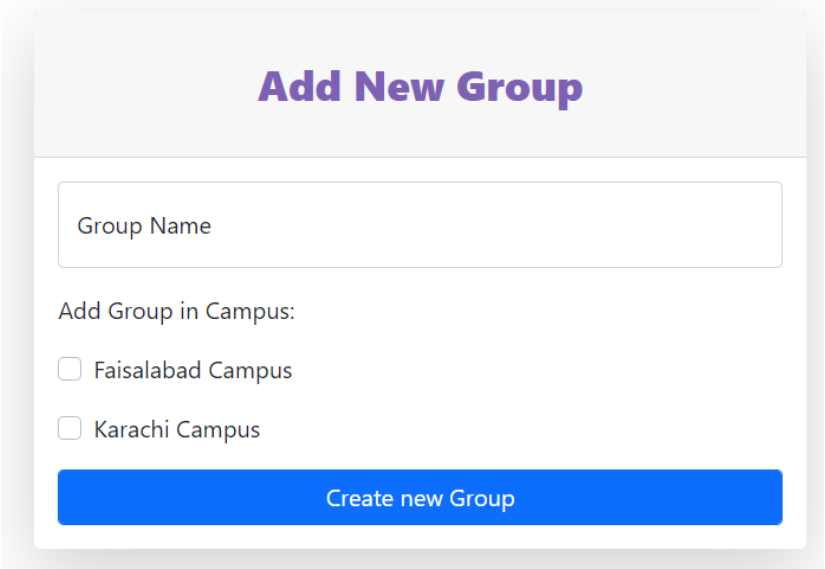

Figure 7.3: Master Panel Creating Group

Table 7.4: Test Case for Creating Menu

Component Name	Admin Panel
Module Name	Create menu
Condition being tested	Admin will be able to create new menu
Expected Result	Menu created successfully
Success Scenarios	Admin can view created menu
Failure Scenarios	Admin cannot view created menu
Test Result (Pass/ Fail)	Passed

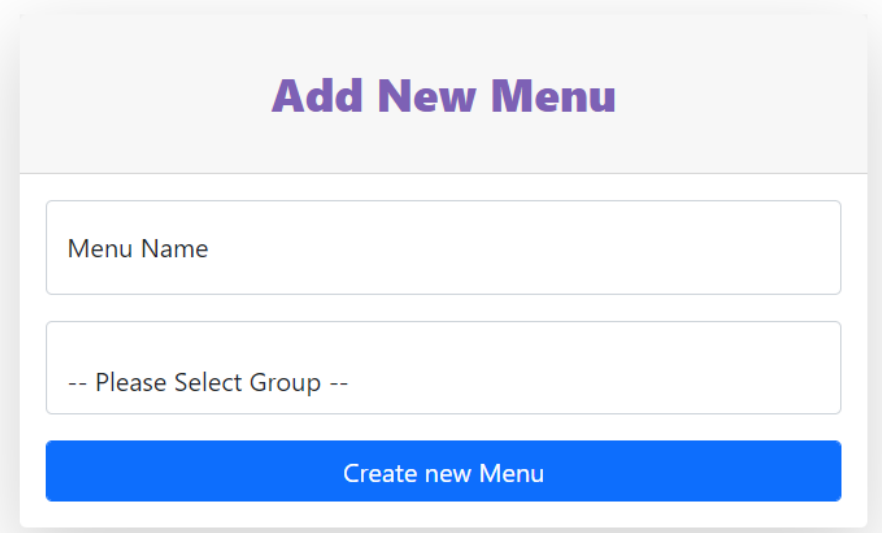

Figure 7.4: Master Panel Creating Menu

Table 7.5: Test Case for Assigning Menu to Group

Component Name	Admin Panel
Module Name	Assign Menu
Condition being tested	Admin will be able to assign menu to group
Expected Result	Menu assigned successfully
Success Scenarios	Admin can view assigned menu
Failure Scenarios	Admin cannot view assigned menu
Test Result (Pass/ Fail)	Passed

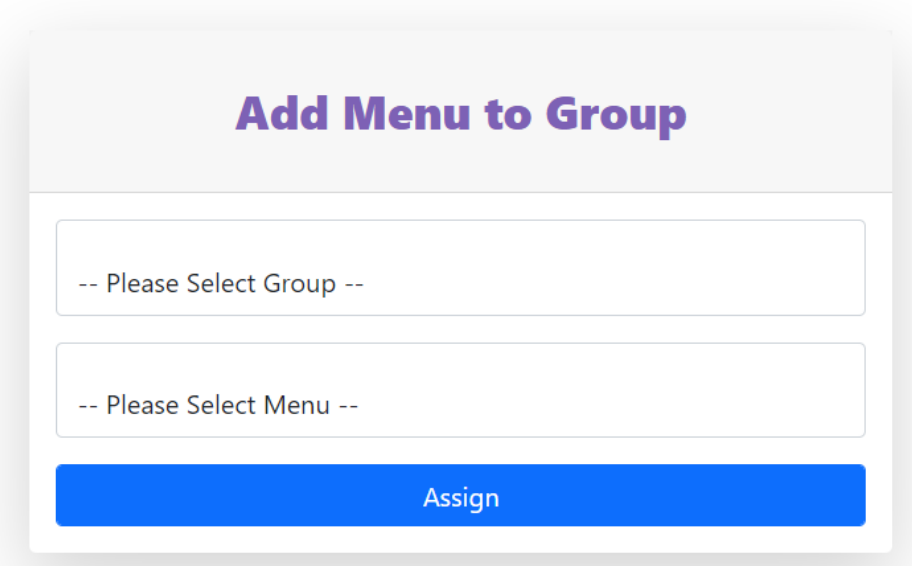

Figure 7.5: Master Panel Assigning Menu to Group

Table 7.6: Test Case for Assigning Group to User

Component Name	Admin Panel
Module Name	Assign Group
Condition being tested	Admin will be able to assign group to user
Expected Result	Menu assigned successfully
Success Scenarios	Admin can view assigned group
Failure Scenarios	Admin cannot view assigned group
Test Result (Pass/ Fail)	Passed

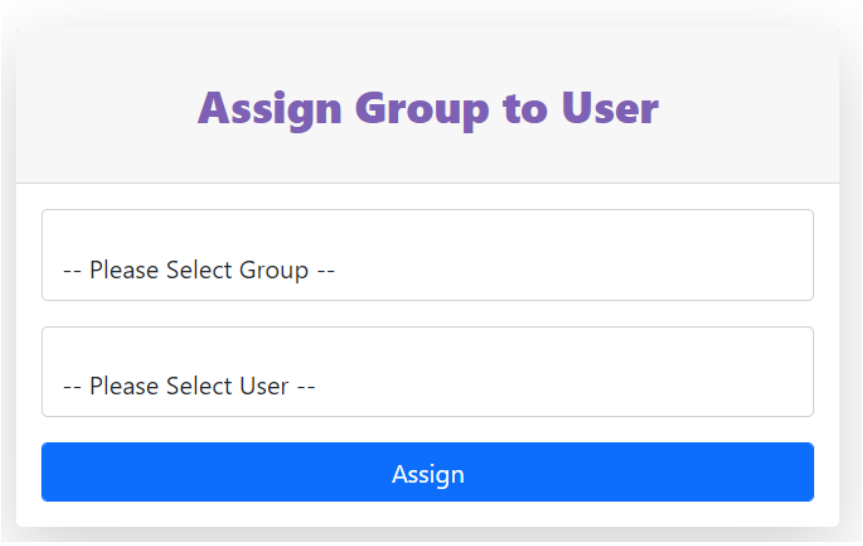

Figure 7.6: Master Panel Assigning Group to User

Table 7.7: Test Case for Admin Login (Failure)

Component Name	Admin Panel
Module Name	Login
Condition being tested	Admin will not be able to login
Expected Result	Login failed
Success Scenarios	User will not be able to view dashboard
Failure Scenarios	User will be able to view dashboard
Test Result (Pass/ Fail)	Passed

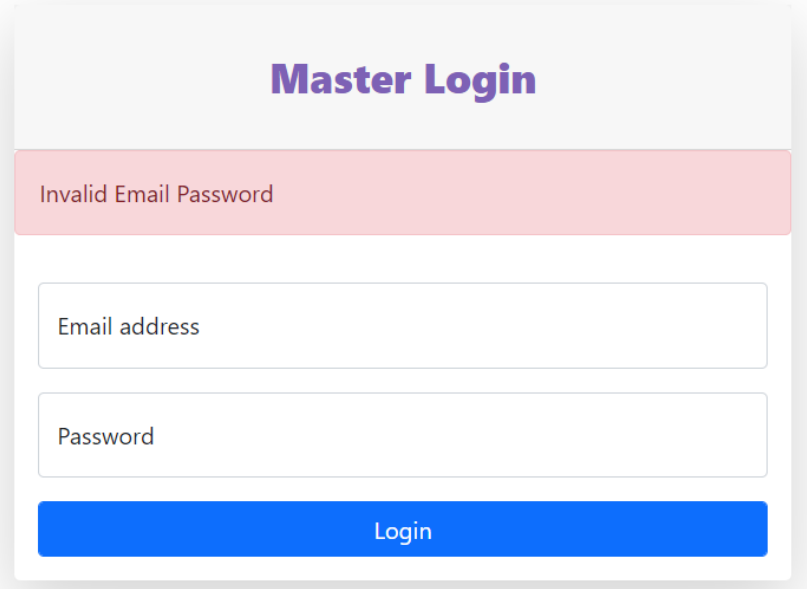

Figure 7.7: Master Panel Invalid Login

Table 7.8: Test Case for User Login (Success)

Component Name	User Panel
Module Name	Login
Condition being tested	Admin will be able to login
Expected Result	Login successfully
Success Scenarios	User will be able to view dashboard
Failure Scenarios	User will not be able to view dashboard
Test Result (Pass/ Fail)	Passed

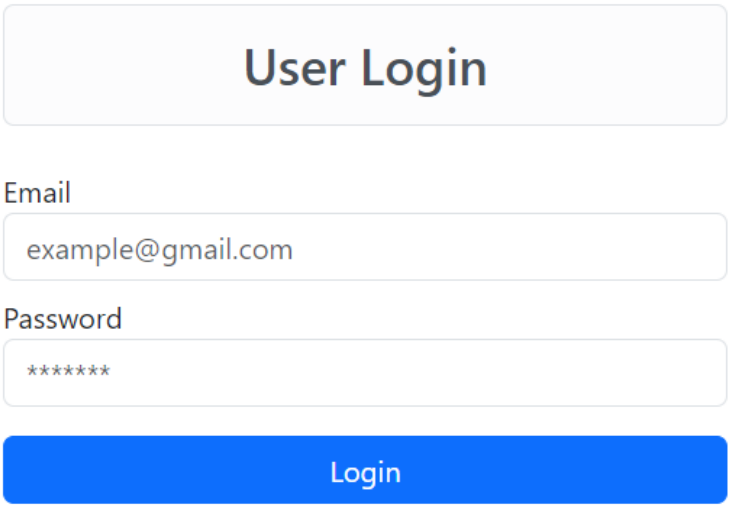

Figure 7.8: User Panel Login

Table 7.9: Test Case for User Login (Failure)

Component Name	User Panel
Module Name	Login
Condition being tested	Admin will be able to login
Expected Result	Login successfully
Success Scenarios	User will be able to view dashboard
Failure Scenarios	User will not be able to view dashboard
Test Result (Pass/ Fail)	Passed

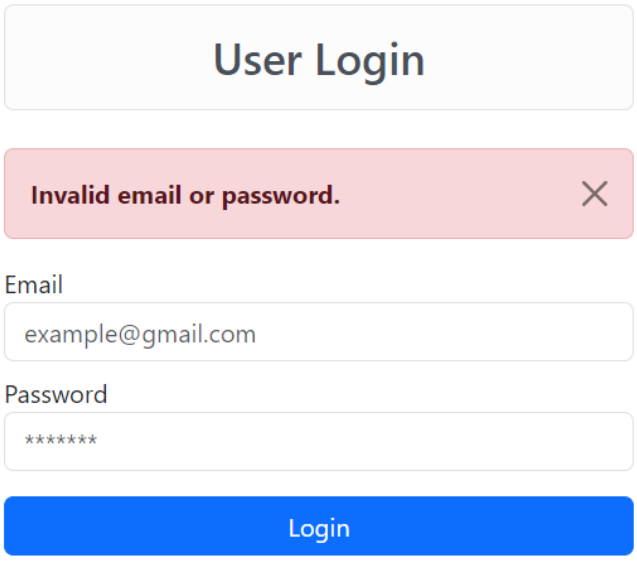

Figure 7.9: User Panel Login Failed

CHAPTER 8

8. Application Prototype

Before the system is fully developed, stakeholders can see and engage with the system thanks to the application prototype we've created. The essential features and user interface are highlighted, allowing for useful input and validation to design the finished product and guarantee that it meets user expectations.

8.1.1. Interfaces

The user-centric design of our CMS's interface provides a simple and visually appealing user experience.

8.1.1.1. Campus Website

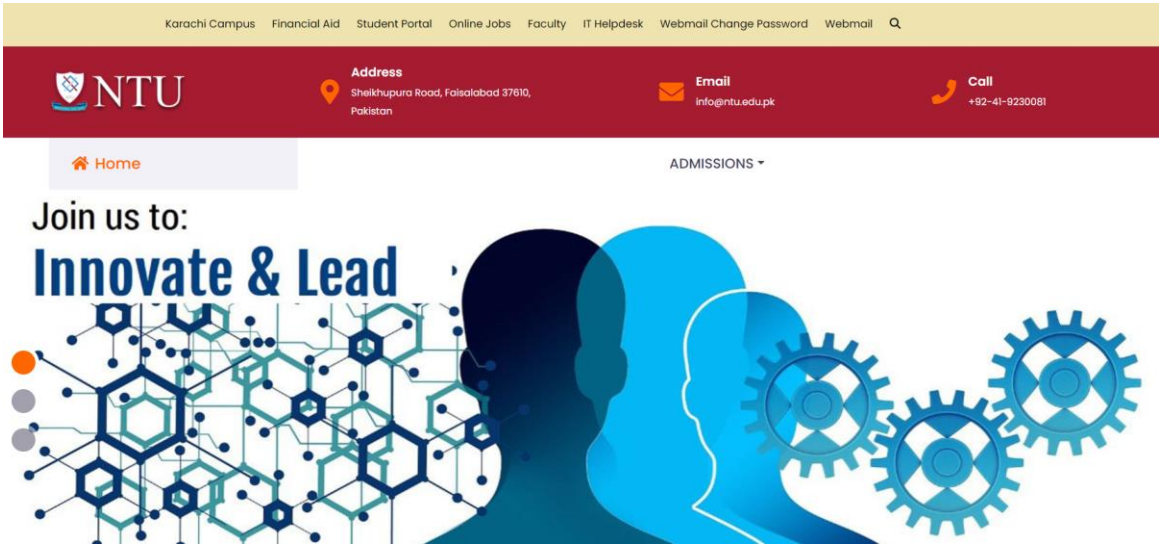

Figure 8.1: Website Interface 1

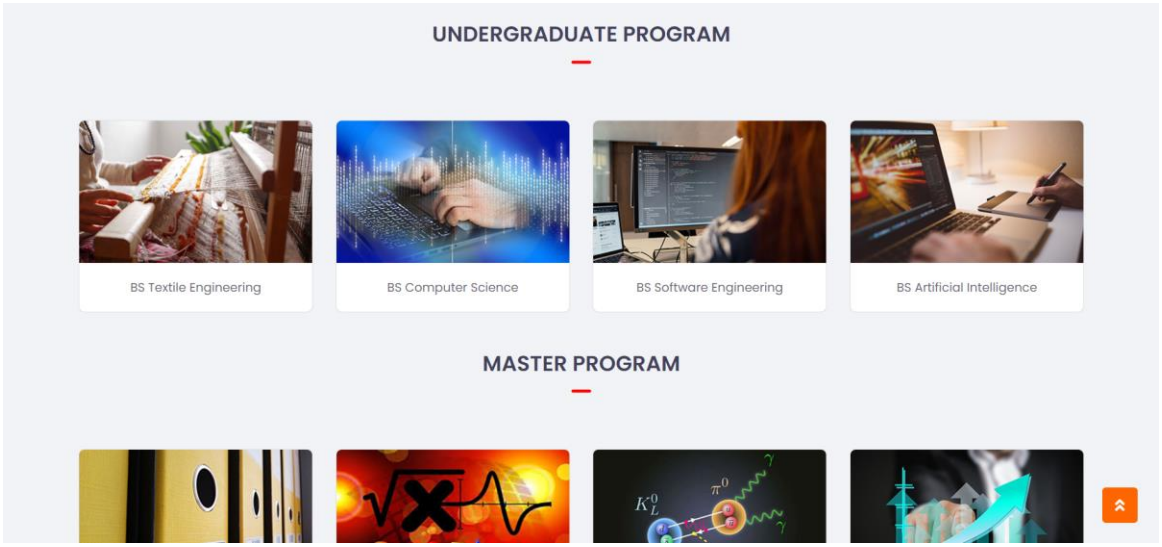

Figure 8.2: Website Interface 2

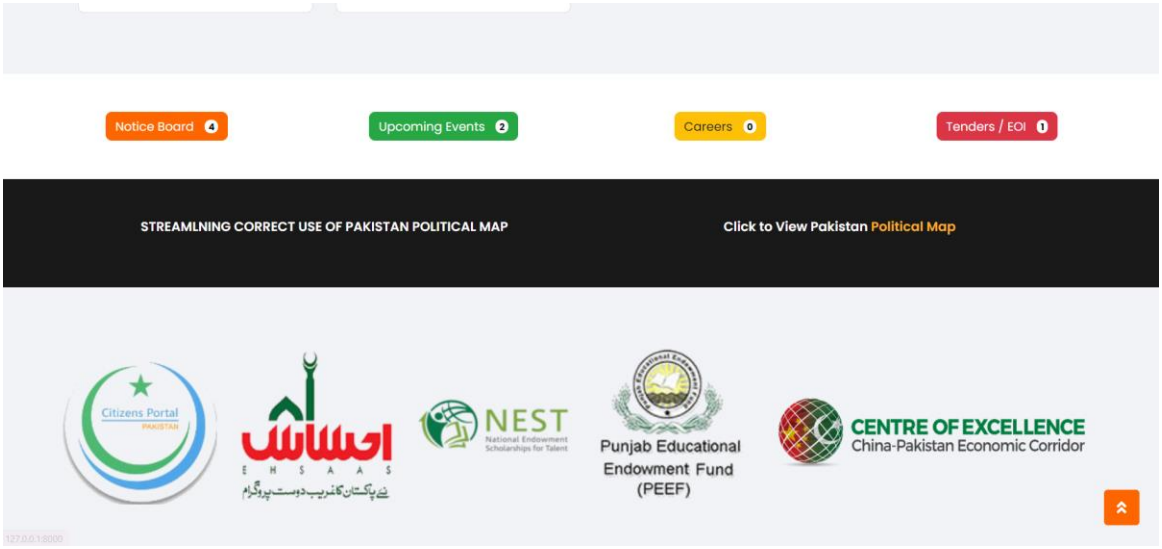

Figure 8.3: Website Interface 3

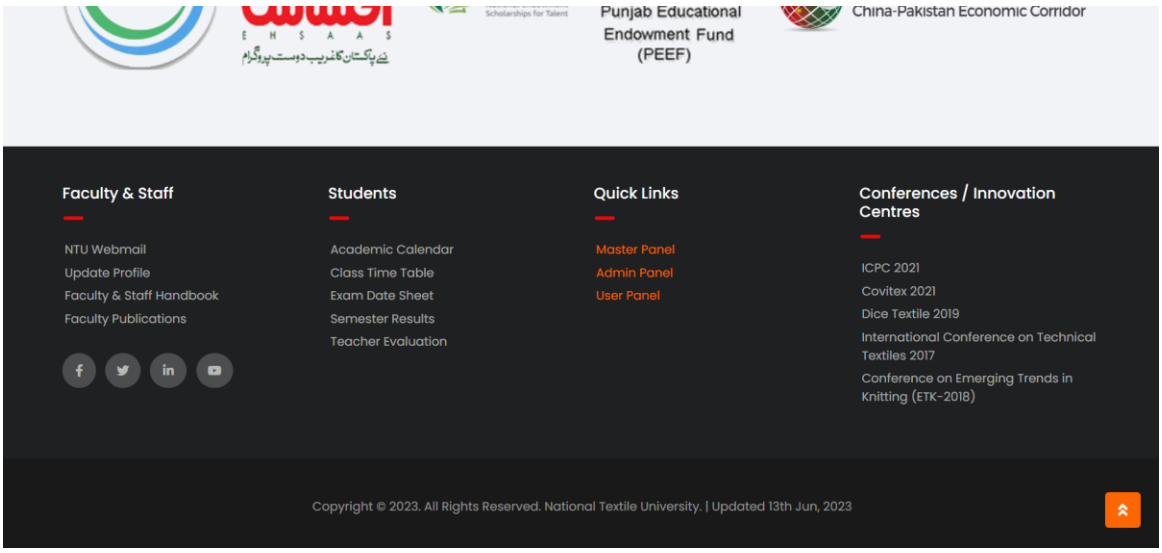

Figure 8.4: Website Interface 4

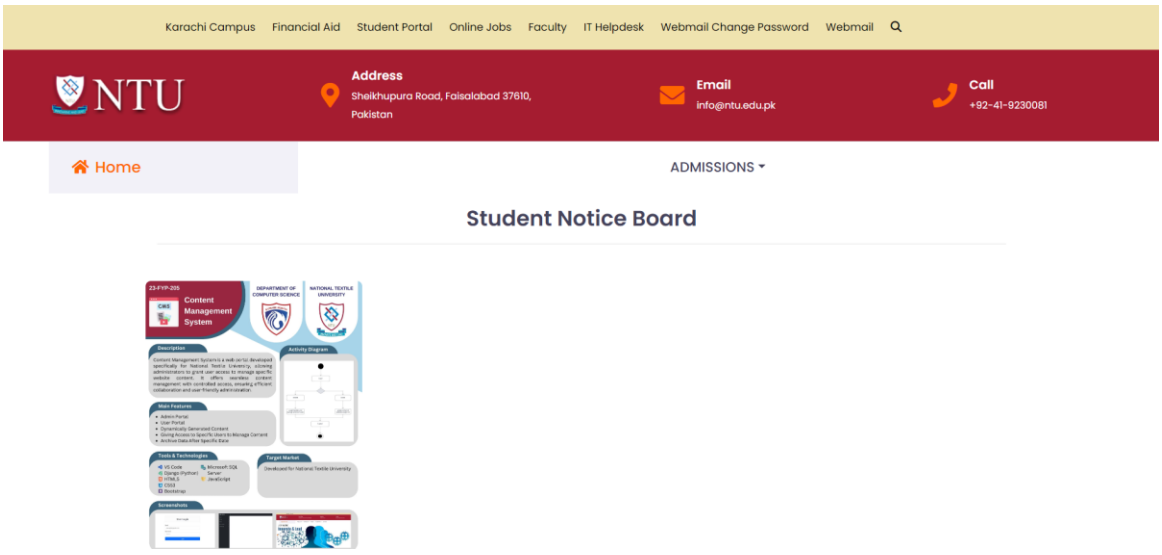

Figure 8.5: Website Interface 5

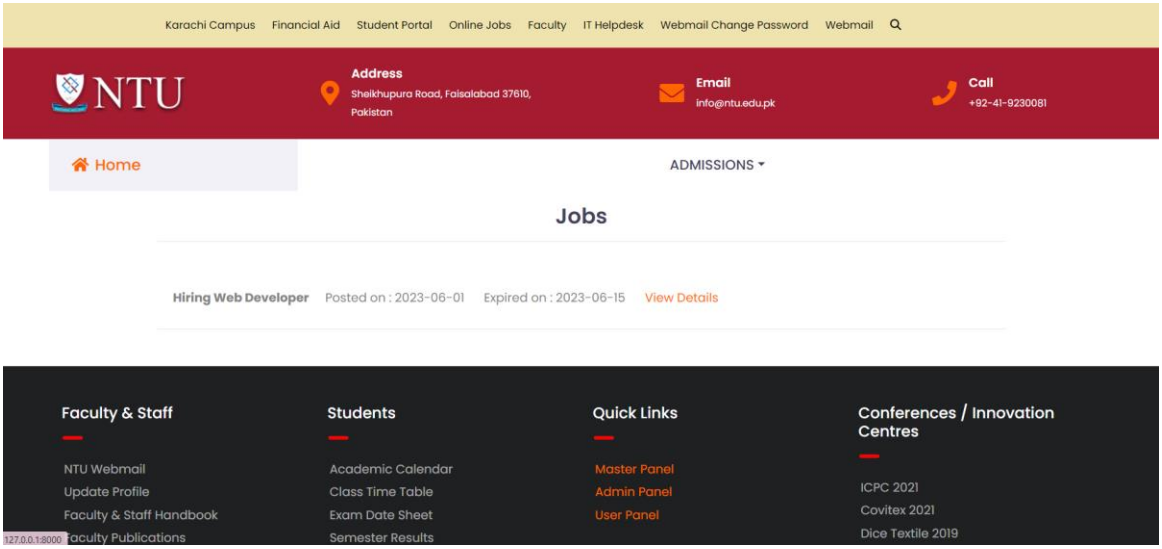

Figure 8.6: Website Interface 6

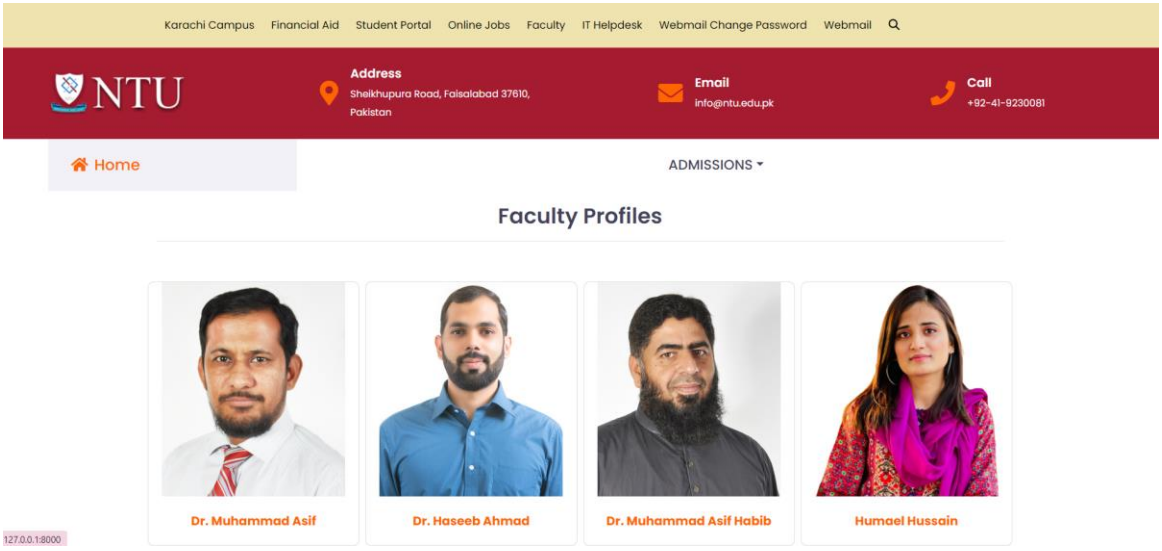

Figure 8.7: Website Interface 7

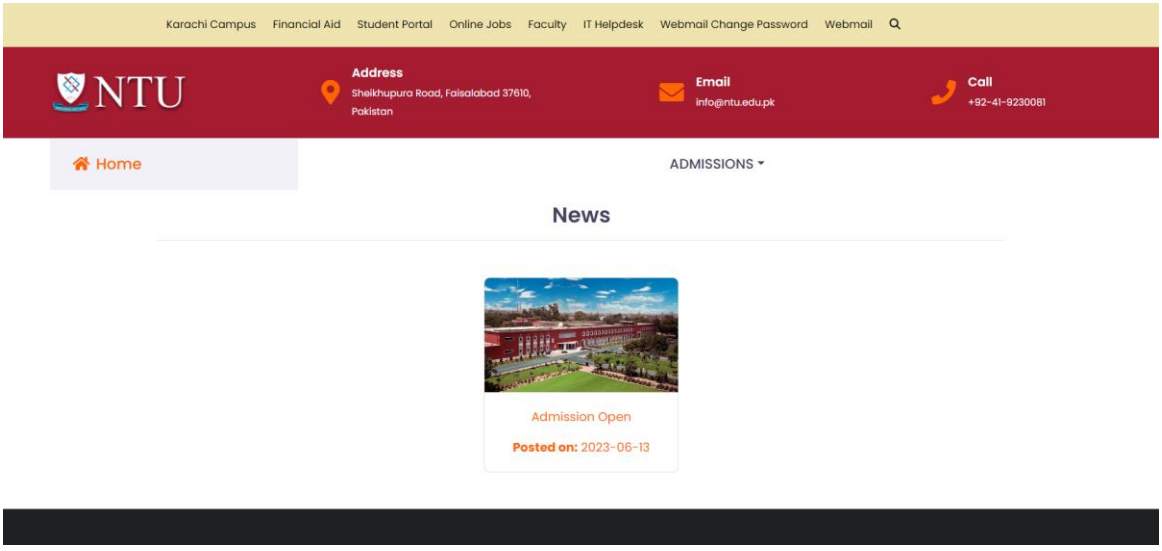

Figure 8.8: Website Interface 8

8.2. Admin Panel

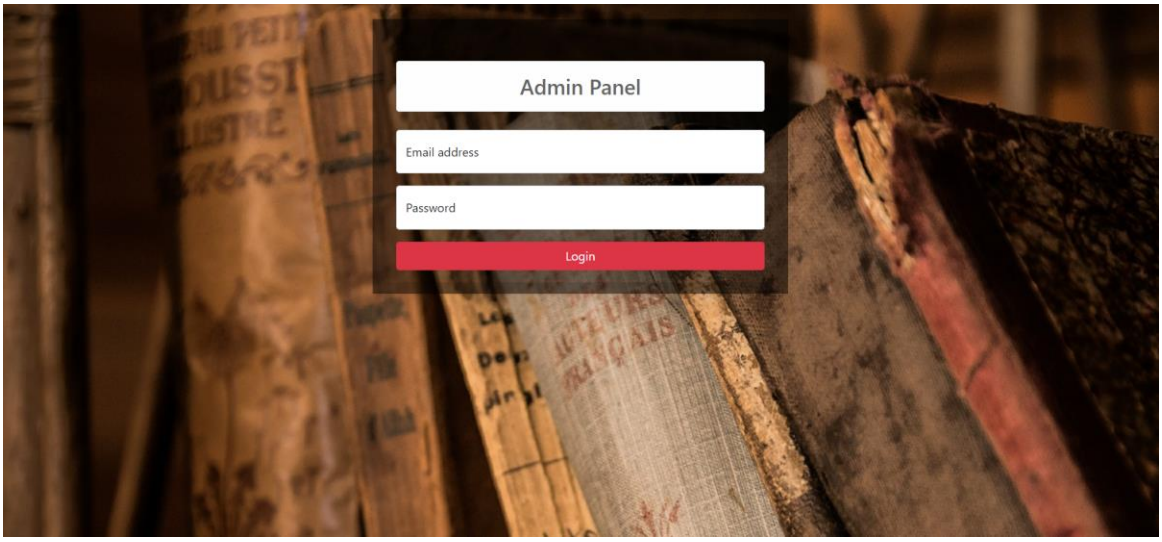

Figure 8.9: Admin Panel Login Interface

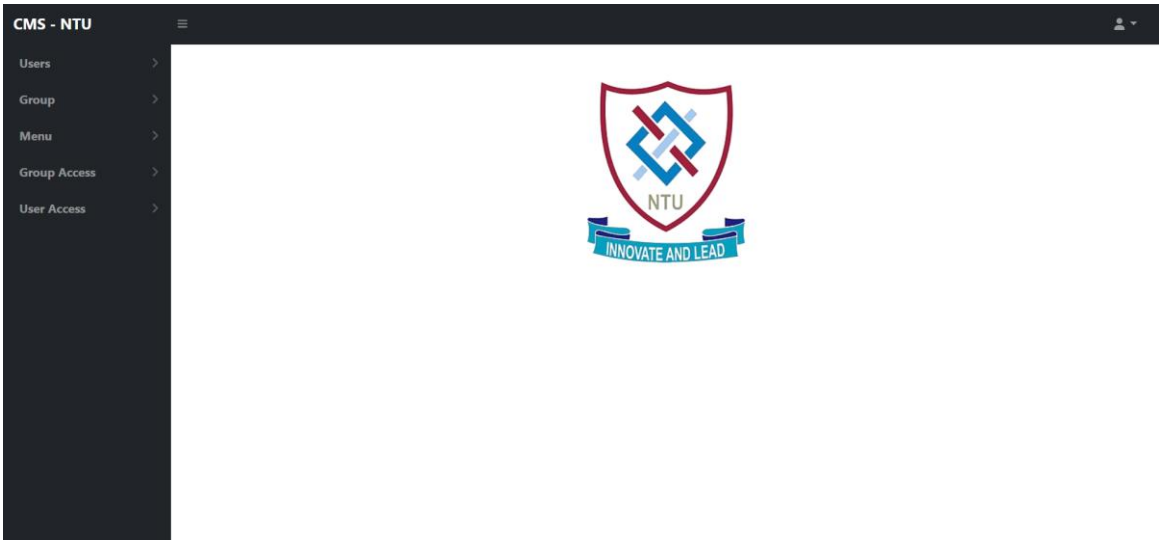

Figure 8.10: Admin Panel Dashboard

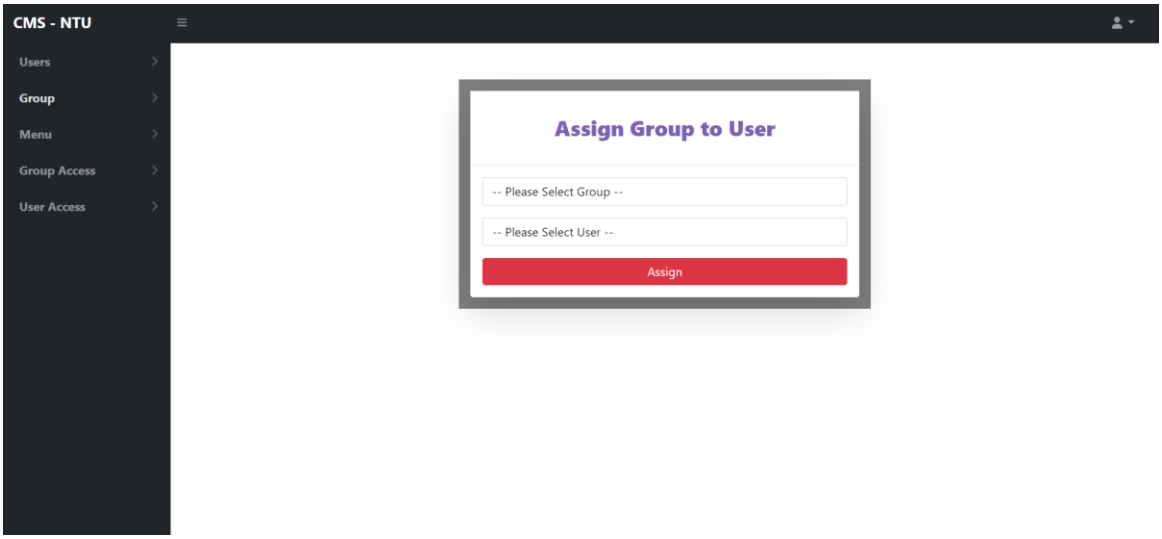

Figure 8.11: Admin Panel Assigning Group to User

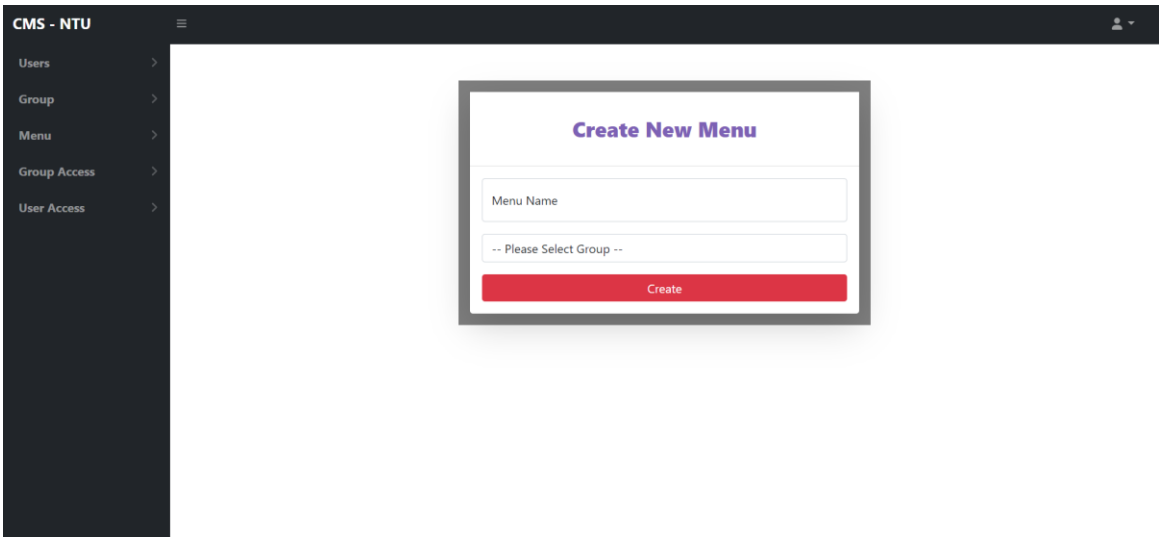

Figure 8.12: Admin Panel Creating Menu

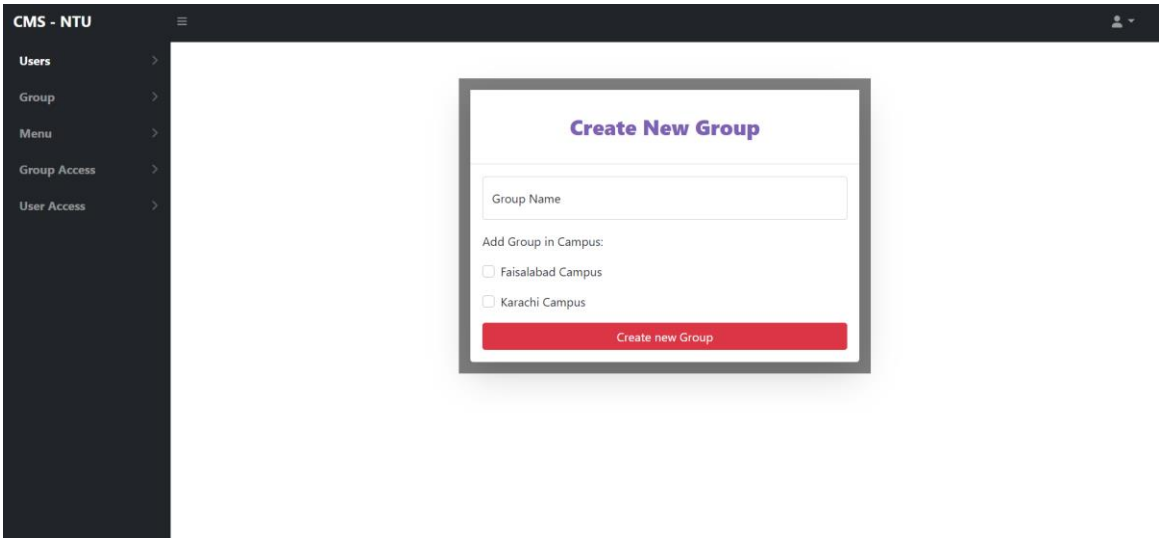

Figure 8.13: Admin Panel Creating Group

8.3. User Panel

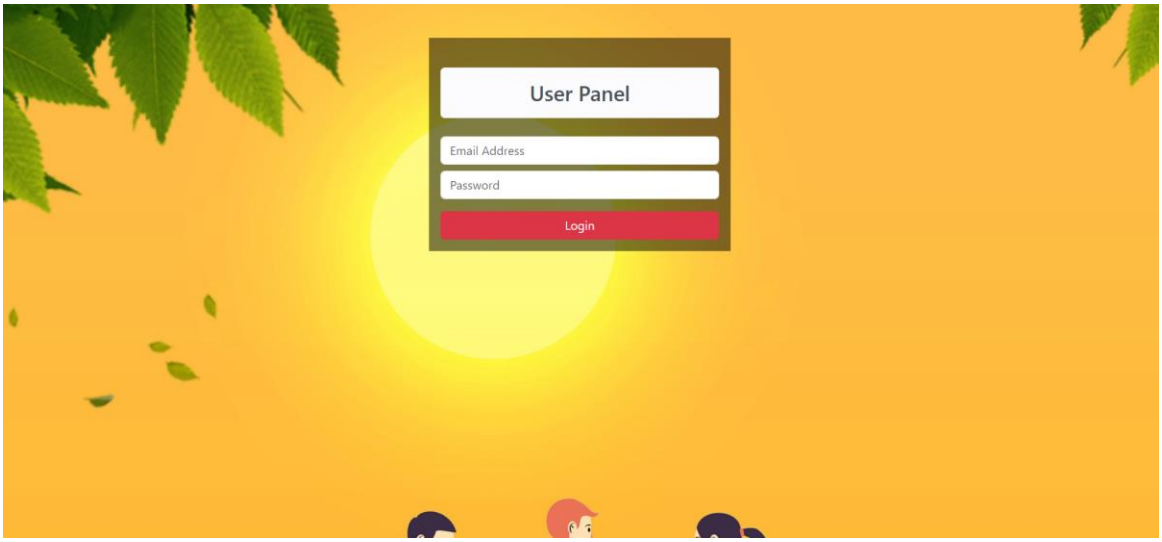

Figure 8.14: User Panel Login Interface

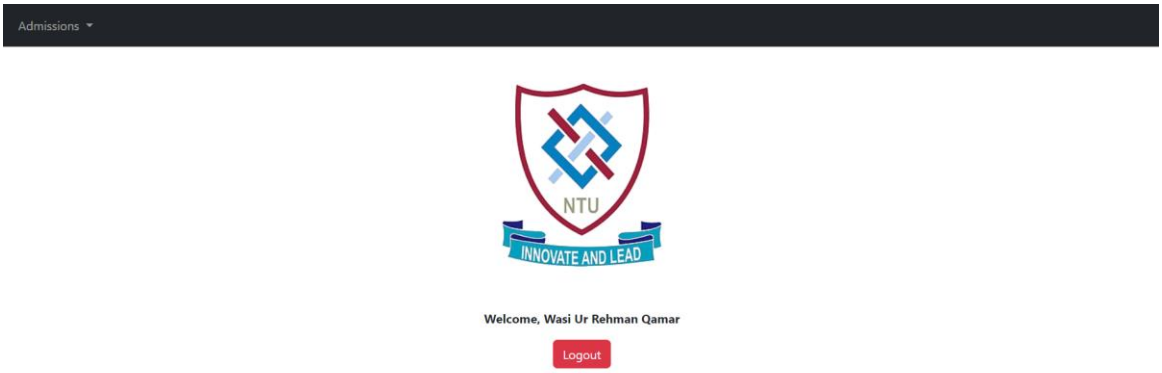

Figure 8.15: User Panel Dashboard

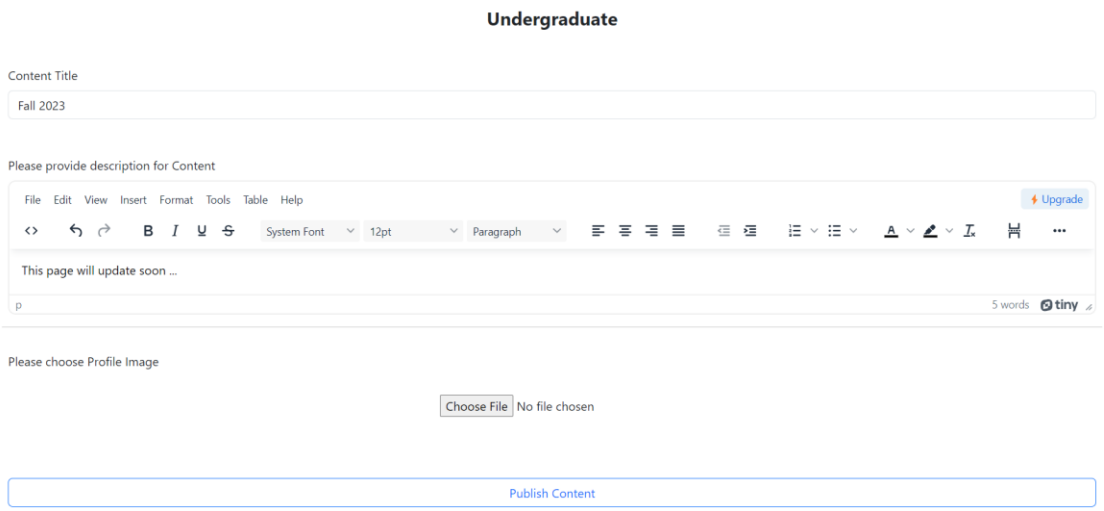

Figure 8.16: User Panel Editing Page

References

[1] National Textile University, Faisalabad
https://ntu.edu.pk

[2] WordPress Multisite: The Everything-You-Need-To-Know Guide to WordPress Network
https://kinsta.com/blog/wordpress-multisite

[3] Django CMS documentation
https://docs.django-cms.org

[4] Contentful - A limitless composable content platform
https://www.contentful.com

[5] Squarespace
https://www.squarespace.com

[6] The Benefits of Agile in Healthcare
https://www.usfhealthonline.com/resources/health-informatics/the-benefits-of-agile-in-healthcare