Class Routine Builder

BY

Md Abdur Rakib
ID: 201-15-13656
AND

Mirza Mojahid
ID: 201-15-14315

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. S.M. Aminul Haque
Professor & Associate Head
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised By

Nahid Hasan
Lecturer
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

July, 2024

APPROVAL

This Project titled "Class Routine Builder", submitted by Md Abdur Rakib (ID:201-15-13656) and Mirza Mojahid (ID:201-15-14315) to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 15-07-2024.

BOARD OF EXAMINERS

15.07.2024

Dr. Md. Fokhray Hossain (MFH) — **Board Chairman**
Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

15.7.24

Md. Firoz Hasan (FH) — **Internal Examiner 1**
Sr. Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

15.07.24

Lamia Rukhsara (LR) — **Internal Examiner 2**
Sr. Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

15.07.24

Dr. S. M. Hasan Mahmud (SMH) — **External Examiner**
Assistant Professor
Department of Computer Science
American International University-Bangladesh

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. S.M. Aminul Haque, Professor & Associate Head, Department of Computer Science and Engineering, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Dr. S.M. Aminul Haque
Professor & Associate Head
Department of CSE
Daffodil International University

Co-Supervised by:

Nahid Hasan
Lecturer
Department of CSE
Daffodil International University

Submitted by:

Md Abdur Rakib
ID: 201-15-13656
Department of CSE
Daffodil International University

Mirza Mojahids
ID: 201-15-14315
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty for His divine blessing making it possible for us to complete the final year project/internship successfully.

We are grateful and wish our profound indebtedness to **Dr. S.M. Aminul Haque**, **Professor & Associate Head**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Field name*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Sheak Rashed Haider Noori, Professor & Head of the Department of CSE**, for his kind help in finishing our project and also to other faculty members and the staff of the Department of CSE, Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

In the realm of educational institutions, the creation of class routines stands as a crucial yet often arduous task. The efficiency and effectiveness of a class routine profoundly impact the learning environment and overall productivity. This project endeavors to streamline and optimize the process of routine building through the development of a Class Routine Builder. The Class Routine Builder is a comprehensive software solution designed to facilitate the seamless generation of class schedules. It encompasses features tailored to accommodate various complexities inherent in academic scheduling, including the assignment of courses, teachers, and rooms. Leveraging advanced algorithms and intuitive user interfaces, the system aims to simplify the intricate task of organizing academic timetables while ensuring optimal resource utilization and academic continuity. A distinguishing feature of our project lies in its technological underpinnings. The user interface (UI) will be developed using React Js, a versatile and efficient toolkit for crafting beautiful, natively compiled applications across various platforms. Concurrently, the backend API will be engineered using Django, a high-level Python web framework renowned for its scalability, security, and rapid development capabilities. With our commitment to innovation and excellence, we aim to redefine the landscape of academic scheduling, ushering in a new era of efficiency and effectiveness in educational institutions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Overview

The Class Routine Builder project seeks to revolutionize the way class schedules are created and managed in educational institutions. By leveraging advanced software technologies, the project aims to address the inefficiencies and complexities inherent in manual scheduling processes. The software will provide a user-friendly interface for administrators to easily generate and modify class schedules, thereby enhancing the overall efficiency and productivity of educational institutions.

Overall, The Class Routine Builder simplifies the intricate task of organizing academic timetables, leading to increased efficiency, better resource utilization, and improved academic outcomes within educational institutions.

1.2 Problem Statement

Manual class routine creation is characterized by tedious and error-prone tasks, including the coordination of course offerings, teacher availability, and room assignments. These manual processes often lead to scheduling conflicts, resource wastage, and dissatisfaction among students and faculty members. The Class Routine Builder project aims to address these challenges by automating and optimizing the scheduling process, thereby improving the overall functioning of educational institutions.

By automating routine tasks and providing intelligent scheduling algorithms, the Class Routine Builder seeks to minimize scheduling conflicts and optimize resource utilization. This will not only alleviate the administrative burden on educators and administrators but also improve the overall learning experience for students.

1.3 Objectives

The primary objectives of this project are:

- Create a straightforward and user-friendly interface enabling system administrators and staff to navigate the platform with ease.
- Implement role-based access control to manage administrative staff permissions effectively, ensuring that only authorized personnel can perform specific actions within the system.
- Implement a feature that allows users to view real-time room availability and assign appropriate rooms for classes to avoid conflicts.
- Enable administrators to offer available course slots to teachers based on their expertise and availability, ensuring efficient course distribution.
- Provide system administrators with robust tools to effectively manage user accounts, system settings, and updates.
- Provide a mechanism for administrators to check the availability of teachers' time slots and assign classes accordingly, minimizing scheduling conflicts.
- Develop algorithms to identify and resolve scheduling conflicts such as overlapping class times or resource double-booking, ensuring smooth operation of the scheduling process.
- Maximize the utilization of resources such as classrooms and teaching staff by intelligently assigning classes and avoiding underutilization or overbooking.
- Generate comprehensive class schedules that include course timings, teacher assignments, room allocations, and any additional relevant information needed for effective classroom management.
- Minimize the manual effort required for routine generation and provide efficient tools for administrators to streamline their workflow.

These objectives align with the overarching goal of your project to develop a Class Routine Builder that enhances efficiency, reduces conflicts, and optimizes resource utilization within educational institutions.

1.4 Expected Outcomes

The expected outcomes for "Class Routine Builder" include:

- Efficient Routine Generation: The system will streamline the process of routine generation, reducing the time and effort required by administrators to create class schedules.
- Improved Resource Utilization: By optimizing room assignments and teacher schedules, the project will lead to better utilization of resources within the educational institution.
- Real-Time Room Availability: Users will be able to view and assign available rooms in real-time, minimizing conflicts and ensuring smooth classroom allocation.
- Effective Course Distribution: Teachers will receive offers for available course slots based on their expertise and availability, leading to a fair and efficient distribution of teaching responsibilities.
- Enhanced Teacher Satisfaction: With the ability to check their time slot availability and receive course offers that align with their preferences, teachers will experience increased satisfaction with the scheduling process.
- Conflict Resolution: The system will identify and resolve scheduling conflicts automatically, minimizing disruptions and ensuring the integrity of the class schedules.
- Increased Academic Continuity: By reducing conflicts and optimizing schedules, the project will contribute to improved academic continuity, allowing for more efficient delivery of curriculum content.

These outcomes align with the objectives of the project and reflect the anticipated benefits of implementing the Class Routine Builder within an educational institution.

1.5 Report Organization

Our total report consists of 3 Chapters:

Chapter 1 provides an introduction to the Class Routine Builder project, including the overview, problem statement, objectives, expected outcomes, report organization, and a summary of the chapter. This chapter sets the stage for the project, highlighting its significance and goals.

Chapter 2 presents a literature review, encompassing an overview of existing works related to class scheduling, a comparative analysis of these works, identification of open issues, and a summary. This chapter establishes the context of the project within the existing body of knowledge and identifies gaps that the Class Routine Builder aims to address.

Chapter 3 details the requirement specification, including business process modeling, requirement collection and analysis, use case modeling and descriptions, the logical data model, design requirements, and project management and financial analysis. This chapter provides a comprehensive blueprint for the system's development, ensuring all stakeholder needs are considered.

Chapter 4 outlines the design specifications of the Class Routine Builder, covering front-end design, back-end design, interaction design and user experience (UX), and implementation requirements. It concludes with a summary, providing a clear plan for the system's technical architecture and user interface design.

Chapter 5 focuses on the implementation and testing of the Class Routine Builder. It discusses the implementation of the database, front-end design, the testing implementation, and presents the test results and reports. This chapter ensures that the system is built and tested according to the specifications, highlighting any issues and their resolutions.

Chapter 6 explores the impact of the Class Routine Builder on society, the environment,

and sustainability. It covers the societal impact, environmental impact, ethical aspects, and the sustainability plan. This chapter emphasizes the broader implications of the project and outlines strategies for its long-term success.

Chapter 7 provides the conclusion and future work for the Class Routine Builder project. It includes a discussion and conclusion, scope for further developments, and limitations. This chapter summarizes the project's achievements and outlines potential areas for future enhancements and research.

1.6 Summary

In summary, the “Class Routine Builder” project aims to address the challenges associated with manual class scheduling processes in educational institutions. By developing a comprehensive software solution, the project seeks to improve efficiency, productivity, and overall satisfaction among stakeholders. With its user-friendly interface, customizable features, and automated scheduling capabilities, the Class Routine Builder has the potential to significantly enhance the academic experience for students and faculty members alike.

CHAPTER 2

Background

2.1 Overview

The “Class Routine Builder” project represents a pioneering effort in the realm of educational institution management. With its innovative approach to scheduling, the project aims to streamline and optimize the process of creating class schedules. By harnessing the power of advanced software technologies, the project seeks to overcome the challenges associated with traditional manual scheduling methods. Through a user-friendly interface, administrators will be empowered to efficiently generate and modify class schedules, thereby ushering in a new era of efficiency and productivity within educational institutions.

Here are some terminologies related to "Class Routine Builder":

- Class Routine Builder: The Class Routine Builder project seeks to revolutionize the way class schedules are created and managed in educational institutions.
- Backend: The server-side portion of the software application responsible for processing requests, managing data, and performing computations, often implemented using frameworks like Django.
- User Interface (UI): The client-side portion of the software application that users interact with, typically comprising the user interface elements developed using technologies like React Js.
- System Administrator: An essential role responsible for overseeing user account management, configuring platform settings to ensure optimal performance and security.
- Organizer: A pivotal role dedicated to planning, and overseeing routine related activities, encompassing the creation of routine and management of faculty.
- Constraint: A condition or limitation that must be satisfied or considered during the

process of generating class schedules, such as teacher availability or room capacity.

- Scheduling Conflict: An instance where two or more classes, teachers, require resolution to avoid overlaps or double-bookings.

2.2 Related Work

Here is a review of some relevant literature:

Smith, J., & Brown, K. (2019)[8] introduces an automated class scheduling system designed specifically for universities. It focuses on optimizing resource allocation, minimizing conflicts in class schedules, and improving overall efficiency in managing academic timetables. The system utilizes algorithms to analyze various constraints such as classroom availability, teacher preferences, and student course selections. By automating the scheduling process, the system reduces administrative burden and ensures better utilization of resources.

Gupta, R., & Sharma, P. (2018)[7] present a mobile application aimed at providing students with easy access to their class schedules. The application offers features such as personalized timetable views, real-time updates on schedule changes, and notifications for upcoming classes or events. The mobile application enhances convenience and accessibility for students, allowing them to efficiently manage their academic schedules on-the-go. The study highlights the importance of mobile technology in improving student engagement and satisfaction.

Li, H., & Wang, Q. (2017)[10] discuss the development of an adaptive learning management system (LMS) tailored for personalized education. The system utilizes data analytics and machine learning algorithms to customize educational content and activities based on individual student preferences, learning styles, and performance. The adaptive LMS enhances student engagement, motivation, and learning outcomes by providing tailored learning experiences. It emphasizes the importance of adaptive technologies in addressing diverse student needs in educational settings.

2.3 Comparison between existing works

Conducting a comparative analysis for "Class Routine Builder" entails evaluating comparable platforms or systems currently available in the market. Here's how the analysis can be structured:

Table 2.3.1. Comparison between related works

Aspect	**ClassRoutineBuilder**	**Smith & Brown (2019)**	**Gupta & Sharma (2018)**	**Li & Wang (2017)**
System Type	Web-based class routine builder application	Automated class scheduling system for universities	Mobile application for student timetables	Adaptive learning management system (LMS)
Focus	Optimizing class scheduling processes and management	Optimizing resource allocation, minimizing conflicts in class schedules	Providing students with easy access to class schedules	Tailoring educational content and activities based on individual student preferences
Key Features	Scheduling algorithms, user-friendly interface, customization options	Utilization of algorithms for analysis of constraints	Personalized timetable views, real-time updates on schedule changes,	Customization of educational content and activities based on data analytics and machine learning algorithms
Benefits	Improved organization, time savings, efficient resource utilization	Reduces administrative burden, improves resource utilization	Enhances convenience and accessibility for students, improves engagement and satisfaction	Enhances student engagement, motivation, and learning outcomes through personalized learning experiences

Conducting comprehensive comparative research will provide valuable insights into the strengths, weaknesses, opportunities, and threats facing "Class Routine Builder," allowing for adjustments and enhancements to ensure a competitive edge in the market.

2.4 Open Issues

The Class Routine Builder will encompass a range of features designed to automate and optimize the scheduling process. These features include:

- Creation of class schedules based on predefined criteria and constraints.
- Customizable options for specifying course offerings, teacher preferences, and room allocations.
- Real-time conflict detection and resolution mechanisms to ensure smooth scheduling operations.

The scope of the project will include both frontend development (UI design) and backend development (algorithm implementation and database management), ensuring a comprehensive and integrated solution.

2.5 Summary

In summary, the “Class Routine Builder” project represents a significant advancement in the field of educational scheduling. By combining advanced software technologies with a user-friendly interface, the project aims to revolutionize the way class schedules are created and managed. Through its innovative approach, the project seeks to enhance the efficiency and productivity of educational institutions, ultimately benefiting students, teachers, and administrators alike.

CHAPTER 3

Requirement Specification

3.1 Business Process Modeling

Mapping the flow of class scheduling processes is integral to the business process modeling of the "Class Routine Builder." This involves outlining the steps from class schedule creation and user registration to enrollment management and timetable generation. By mapping relationships between students, instructors, and system administrators, this modeling effort streamlines operational efficiency. Visualizing the platform's features facilitates improved planning and workflow optimization, ultimately enhancing user experiences.

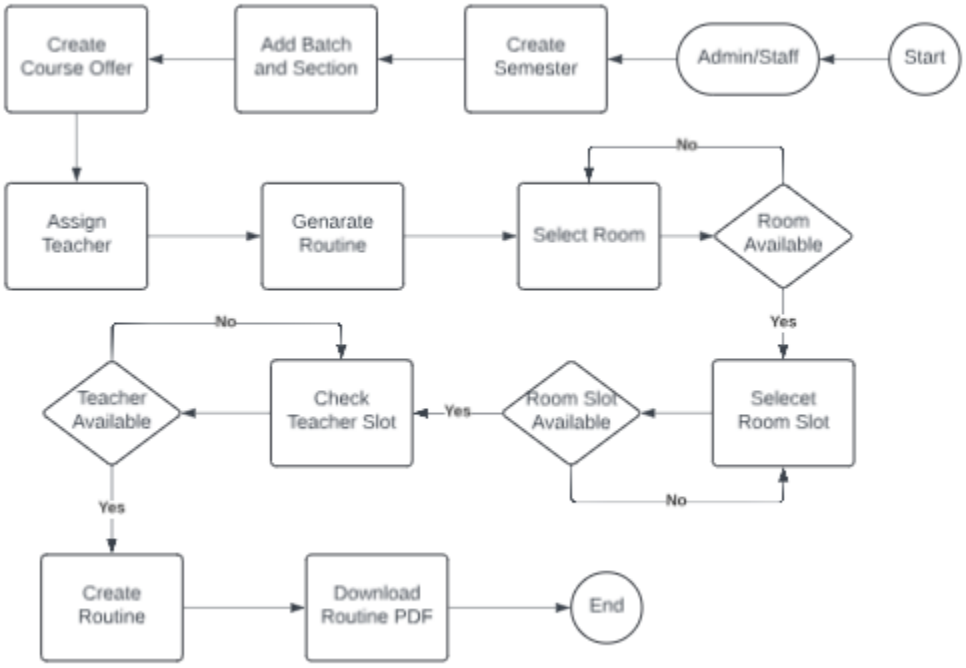

Figure 3.1.1: Business Process Modelling

3.2 Requirement Collection and Analysis

Requirement Collection and Analysis for "Class Routine Builder" entails a methodical approach to gathering and comprehending the requirements of system administrators and organizers. Here's a structured outline of the process:

- Input Data Collection: Gather comprehensive data on course offerings, teacher availability, room availability, and any constraints or preferences related to class scheduling.
- User Requirement Gathering: Conduct interviews or surveys with administrators, teachers, and other stakeholders to understand their specific needs, preferences, and pain points related to class scheduling.
- Constraint Identification: Identify and document constraints such as teacher preferences, room capacities, courses, and any other factors that may impact class scheduling.
- Scalability Requirement: Determine the scalability requirements of the system to accommodate the scheduling needs of different educational institutions, ranging from small schools to large universities.
- Performance Requirement: Define performance metrics and requirements for the system, including response times, throughput, and resource utilization, to ensure smooth operation even during peak usage times.
- Security Requirement: Define security requirements to protect sensitive data such as student records, teacher schedules, and administrative information, including authentication, authorization, and data encryption measures.
- User Interface Design: Specify requirements for the user interface, accessibility to ensure that administrators and teachers can interact with the system efficiently.

These requirements will serve as a foundation for the design and development of your Class Routine Builder, guiding the creation of a system that meets the diverse needs of educational institutions while ensuring efficiency, effectiveness, and user satisfaction.

3.3 Use Case Modeling and Description

Use Case Modeling for the 'Class Routine Builder' involves creating a graphical representation of a user's potential interactions with the system. This diagram illustrates various use cases and user roles within the system, including administrators, organizers, and consumers. It outlines how these users interact with the system and describes specific features and functionalities available to each user role.

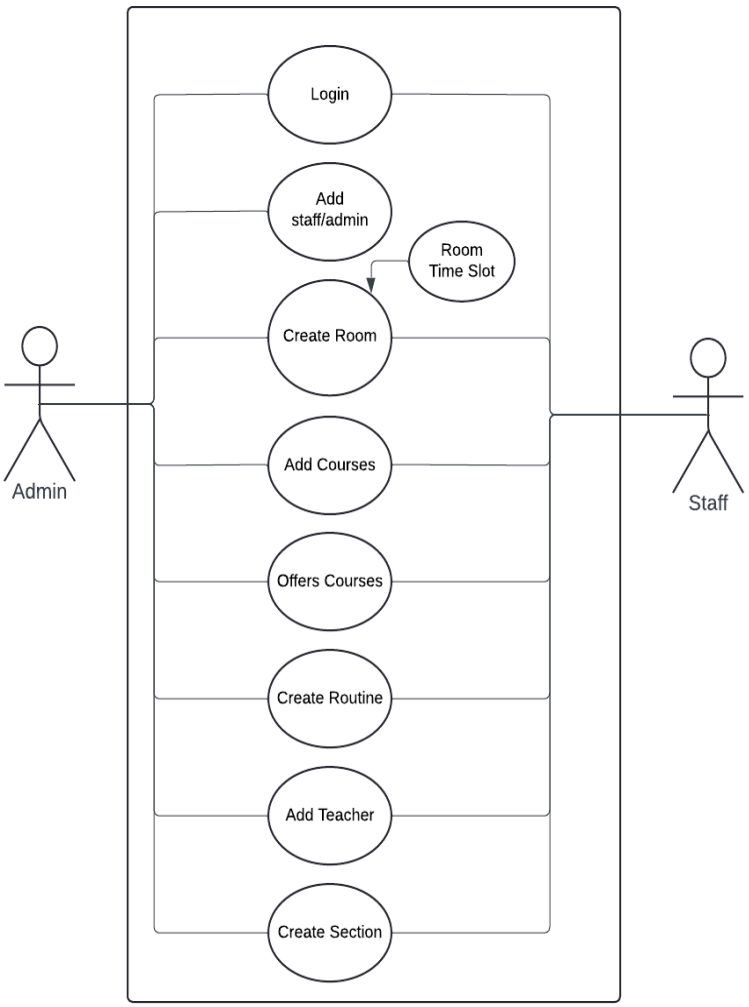

Figure 3.3.1: Use Case Diagram.

3.4 Logical Data Model

In the “Class Routine Builder” project, Logical Data Modeling, plays a pivotal role in structuring and organizing the data within the system's database. This modeling approach focuses on defining the entities, attributes, and relationships that represent the core elements of our scheduling system at a local or project-specific level. By employing Local Data Modeling, we can design a database schema that accurately reflects the requirements of our project, ensuring efficient data storage, retrieval, and manipulation.

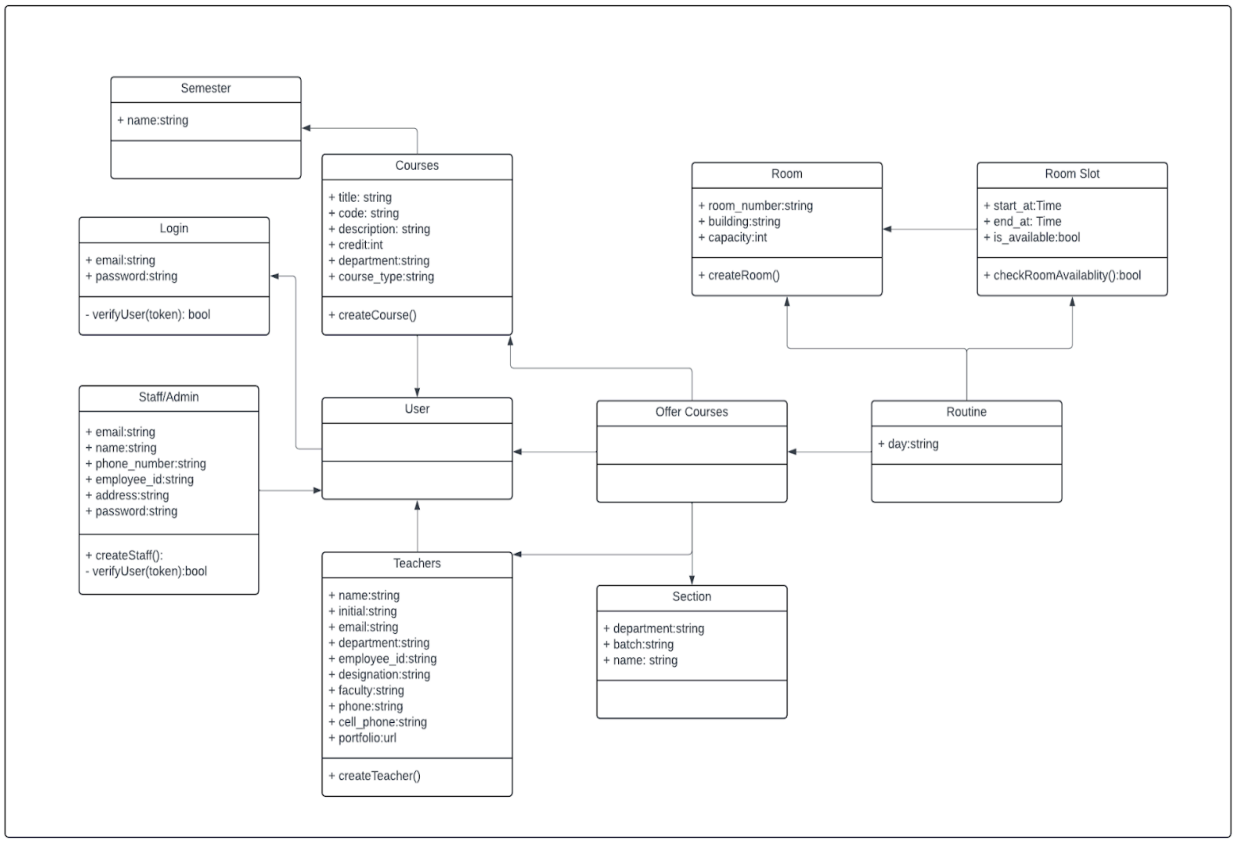

Figure 3.4.1: Logical Data Modeling.

3.5 Design Requirement

Design requirements for "Class Routine Builder" include several key elements to ensure the system is both effective and user-friendly:

- User-Friendly Interface: The system should feature an intuitive and user-friendly interface that allows administrators, organizers, and consumers to interact with the platform easily and efficiently.
- Customization Options: Users should have the ability to customize class schedules based on their specific requirements, including adjusting time slots, assigning rooms, and accommodating special events or activities.
- Scalability and Performance: The system should be scalable to accommodate the scheduling needs of educational institutions of varying sizes, while also maintaining high performance and responsiveness, even during peak usage times.
- Role-Based Access Control: The system should implement role-based access control to ensure that users only have access to functionalities and data relevant to their roles, thus enhancing security and data privacy.
- Conflict Resolution Mechanism: The system must feature a conflict resolution mechanism to detect and address scheduling conflicts, such as overlapping class times or double-booked resources, thereby maintaining the accuracy and integrity of class schedules.
- Flexibility and Adaptability: The system should be flexible and adaptable to accommodate changes in scheduling requirements, user preferences, and institutional policies over time, thus ensuring long-term usability and relevance.
- Accessibility: Ensure the platform adheres to accessibility guidelines, enabling users with diverse impairments to navigate and utilize it effectively, thereby serving a broad spectrum of users.

These design requirements aim to ensure that your Class Routine Builder system is user-centric, efficient, reliable, and capable of meeting the diverse needs of educational institutions.

3.6 Project Management and Finance

Project Management:

- Project Planning: Begin by developing a comprehensive project plan that outlines tasks, deadlines, completion dates, and resource allocation.
- Gathering Requirements: Conduct thorough research and gather detailed requirements from stakeholders, such as system administrators and organizers, to ensure the platform's features meet their needs.
- Agile Development Methodology: Adopt an agile methodology for iterative development, allowing for continuous improvements and adjustments based on feedback received during the development process.
- Design and Prototyping: Start with wireframes and prototypes to visualize the platform's features and interface, ensuring usability and user-friendliness.
- Development and Testing: Develop and integrate features and functions, followed by rigorous testing processes (unit, integration, and user acceptance tests) to ensure functionality and quality.
- Deployment and Launch: Execute a formal launch and marketing plan, deploying the platform in stages or iterations to ensure a smooth transition and minimal disruption.
- Monitoring and Maintenance: Establish a system for continuous monitoring, issue resolution, updates, and post-launch user support to maintain optimal performance and user satisfaction.

By applying these project management principles, we can effectively navigate the complexities of developing a Class Routine Builder and ensure the successful delivery of our project objectives.

Financial Analysis:

The finance for our Class Routine Builder project is meticulously planned and managed to ensure the efficient allocation of resources and the successful execution of the project objectives. A comprehensive budget has been developed, encompassing all project-related costs, including personnel, software development, infrastructure, and other necessary expenses. Cost estimation has been conducted with precision, taking into account factors such as vendor quotes, contingencies, and potential risks. An outline for the project finance analysis may be seen below:

Table 3.6.1. Financial Analysis

SL	**Instrument**	**Estimated Cost**
1	Domain	5,000 - 10,000 Tk.
2	Data Storage Server	8,000 - 15,000 Tk.
3	Deployment Server	20,000 - 30,000 Tk.
4	Documentation and Report Writing	500 - 1,000 Tk.
5	Developer Cost	50,000 - 60,000 Tk.
6	Maintenance Cost	60,000 - 80,000 Tk.
	Total Estimated Cost	143,500 - 196,000 Tk.

3.7 Summary

In summary, this section provides a comprehensive analysis of the methodology used for requirement analysis and design specification in the development of the Class Routine Builder project. It outlines the process of gathering and prioritizing requirements, designing system architecture, analyzing data inputs and outputs, managing the project, and conducting financial analysis. This sets the stage for the subsequent phases of development, implementation and evaluation.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-end Design

In designing the front-end for “Class Routine Builder" project, we are leveraging the power and flexibility of JavaScript and React JS to create a dynamic and responsive user interface. With React JS, we can develop reusable components that encapsulate specific functionalities, allowing for modular and maintainable code. Our front-end design prioritizes an intuitive user experience, with a clean and modern interface that simplifies the task of scheduling classes for administrators, organizers, and consumers alike. Through the use of React JS's component-based architecture, we can efficiently manage complex UI elements and interactions, ensuring smooth navigation and seamless interaction with the scheduling system. Additionally, JavaScript enables us to implement interactive features such as real-time updates, drag-and-drop functionality, and interactive calendars, enhancing the overall usability and functionality of the platform.

4.2 Back-end Design

In the backend design of “Class Routine Builder” project, we are embracing the Django framework alongside Django Rest Framework and Python to construct a robust and scalable architecture. Django, a high-level Python web framework, provides a comprehensive toolkit for rapid development of backend components such as data models, views, and controllers. Django Rest Framework extends Django's capabilities by facilitating the creation of RESTful APIs, enabling seamless communication between the frontend and backend components of our system.

Adhering to the Model-View-Template (MVT) architecture, our backend design ensures a clear separation of concerns, promoting modularity, scalability, and maintainability. In the MVT architecture, models represent the data structures and business logic of the application, views handle the presentation and interaction with users, and templates manage the presentation layer's layout and structure. By following this architectural pattern, we can maintain a clean and organized codebase, enabling easier debugging,

testing, and future enhancements.

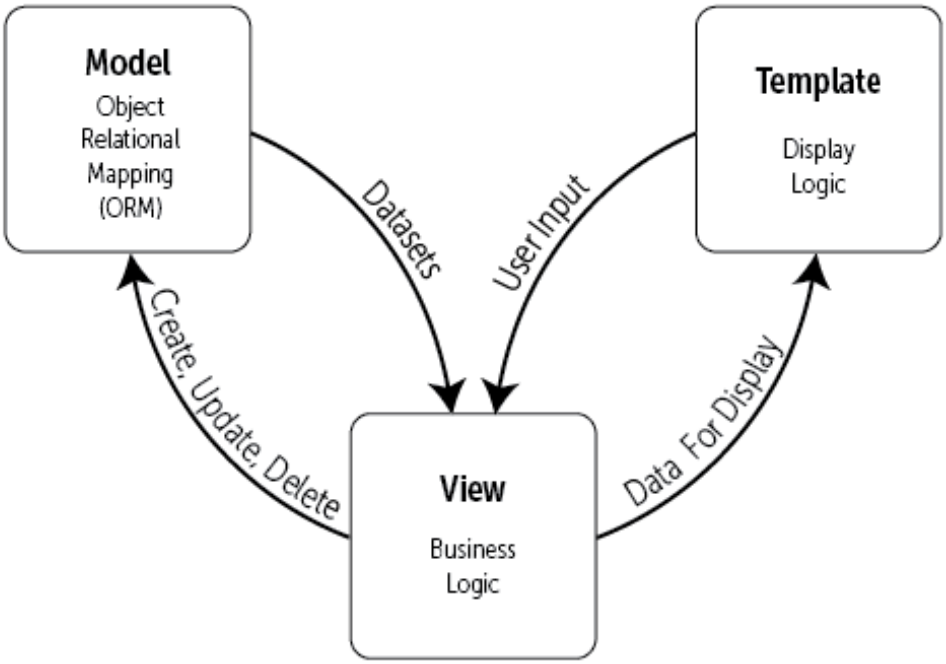

Figure 4.2.1: MVT Architecture.

Our backend design, powered by Django, Django Rest Framework, and Python, coupled with adherence to the MVT architecture and utilization of ORM, promises to deliver a scalable, secure, and efficient solution for managing class schedules within educational institutions.

4.3 Interaction Design and User Experience (UX)

The primary objectives of interaction design and user experience (UX) for "Class Routine Builder" include enhancing user interactions for optimal usability and ensuring a seamless, intuitive experience across the platform:

Interaction Design:

- Facilitate seamless navigation between platform features and sections through intuitive navigation structures.
- Utilize clear, easily understandable buttons and prompts to guide users through tasks such as room selection, teacher creation, and routine generation.
- Maintain consistency in the platform's design, layout, and language to reduce cognitive load and enhance user familiarity.
- Ensure responsiveness and user engagement by providing visual feedback for actions, such as loading indicators or success messages, to acknowledge user input effectively.

User Experience (UX):

- Personalization: Enhance engagement and cater to diverse preferences by offering user-specific settings, personalized recommendations, and customized itineraries.
- Accessibility: Ensure the platform adheres to accessibility guidelines, making it usable for individuals of all abilities and providing options for users with disabilities.
- Usability Testing: Conduct usability testing to ensure the platform's features are easy to use and understand, identifying and addressing any pain points.
- Visual Design and Aesthetics: Utilize a visually appealing design that emphasizes readability, clarity, and an attractive interface to boost user engagement.
- Performance and Speed: Optimize loading times and ensure smooth operation across various devices to prevent user frustration and encourage continued use.

Through the integration of these guiding principles into the design process, "Class Routine Builder" strives to provide a pleasurable, effective, and user-focused experience, encouraging user contentment and allegiance inside the routine based administration system.

4.4 Implementation Requirements

The implementation requirements for "Class Routine Builder" encompass the following technical, operational, and developmental components:

❖ Front End Technologies

- JavaScript
- React.Js
- HTML
- CSS
- Ant Design

❖ Back End Technologies

- Python
- Django
- Django REST
- PostgreSQL
- Simple JWT

❖ Tools

- pgAdmin 4
- PyCharm (IDE)
- Web Browser
- Visual Studio Code (*IDE*)

4.5 Summary

In summary, this section provides a comprehensive overview of the design specifications for the Class Routine Builder project. The front-end design leverages modern web technologies to create an intuitive and accessible user interface. The back-end design, built on Django and Python, ensures robust, scalable, and secure infrastructure. Interaction design and user experience principles focus on delivering a seamless and engaging user experience. Finally, the implementation requirements detail the technical and operational necessities for successful deployment and maintenance. Together, these design specifications form the foundation for a reliable and effective class scheduling system that meets the needs of educational institutions.

CHAPTER 5

Implementation and Testing

5.1 Implementation of Database

The application leverages these database tables to efficiently store, organize, and access data. This approach ensures a solid foundation for the application's operations while maintaining data integrity, security, and optimal performance.

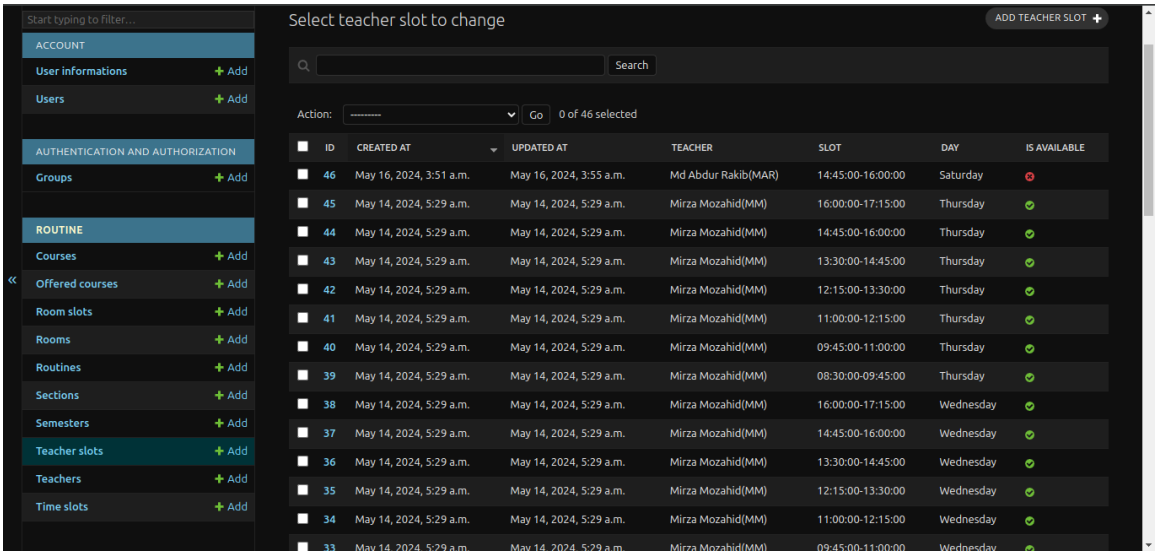

Figure 5.1.1: Database Architecture.

5.2 Implementation of Front-end Design

The front-end design of the “Class Routine Builder” project is critical for ensuring a user-friendly and efficient interface. Utilizing modern technologies such as JavaScript, React.js, HTML, CSS, and Ant Design, we aim to create a responsive and intuitive user experience.

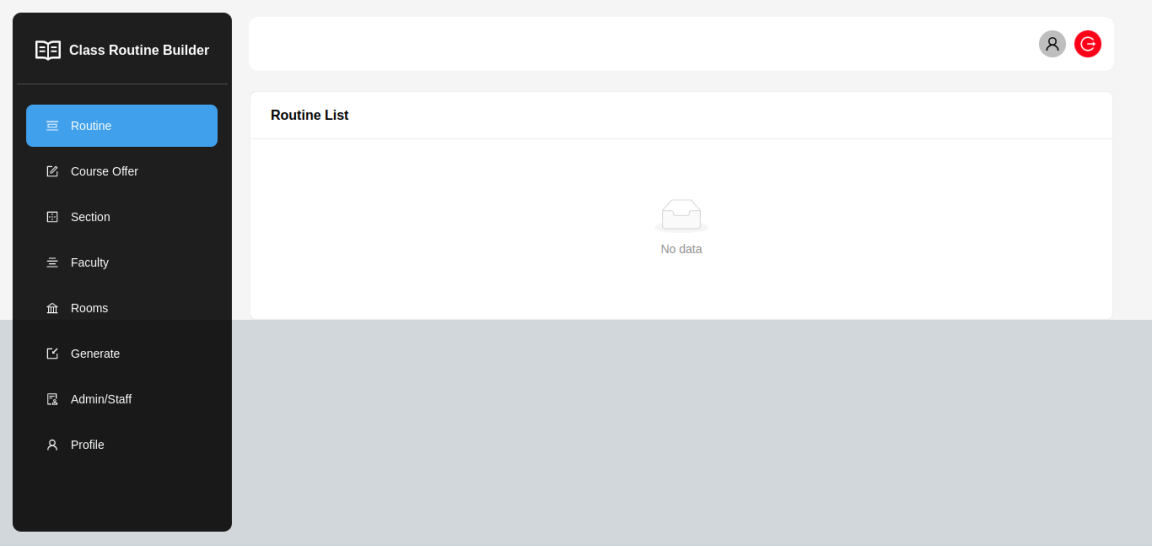

Figure 5.2.1: Routine Page Design.

The routine page of “Class Routine Builder”. Here are all the navigation links present like Course Offer, Room, Faculty etc. And also have a view for routine.

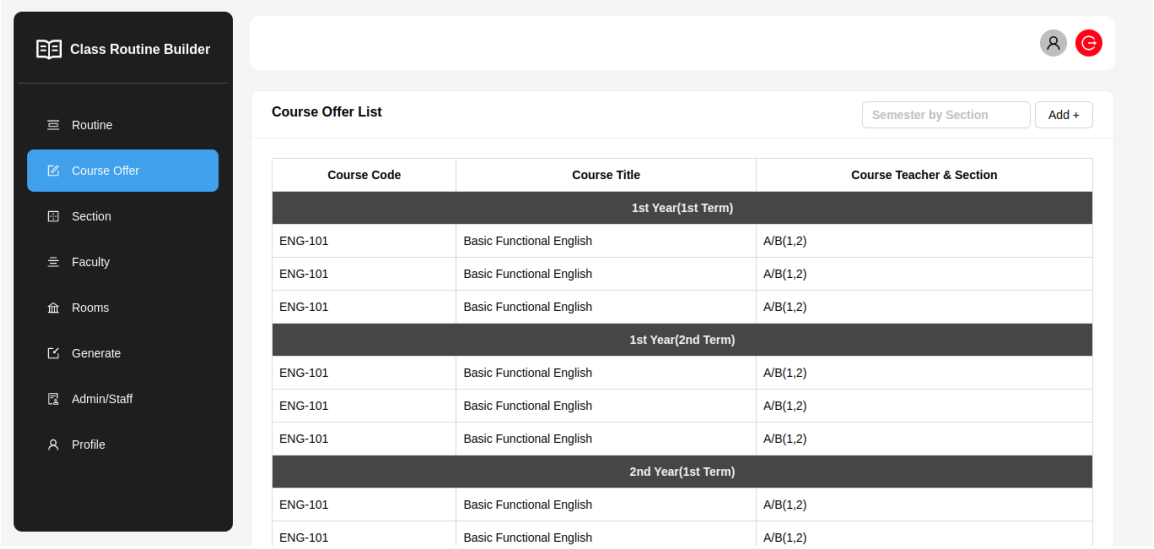

Figure 5.2.2: Course Offer Design.

This page handles the functionality for the course offer of “Class Routine Builder”. And this page controls the course assignment for faculty members.

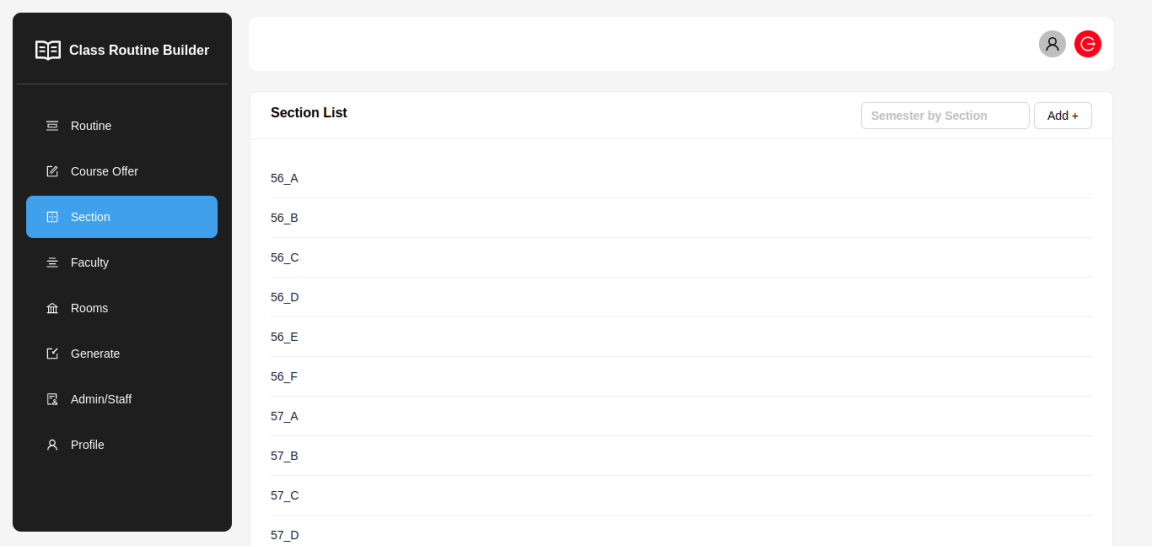

Figure 5.2.3: Section Page Design

This page is designed to handle all the functionality and requirements about the creation of Section for each batch of “Class Routine Builder”.

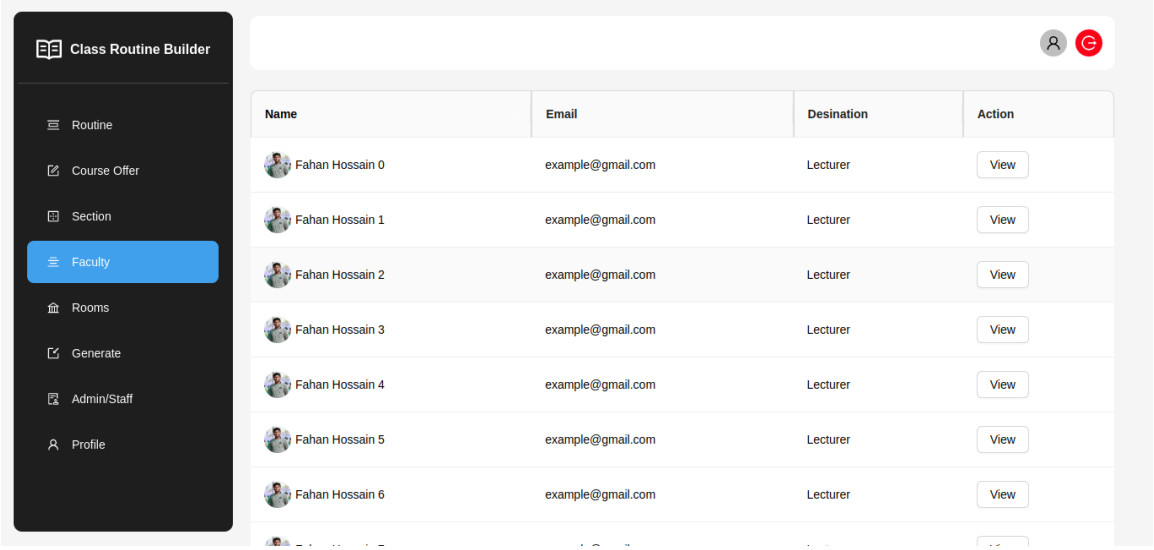

Figure 5.2.4: Faculty Page Design.

This page is the Faculty dashboard of “Class Routine Builder”. In this page show the list of faculty member and admin/staff can also edit the faculty profile information

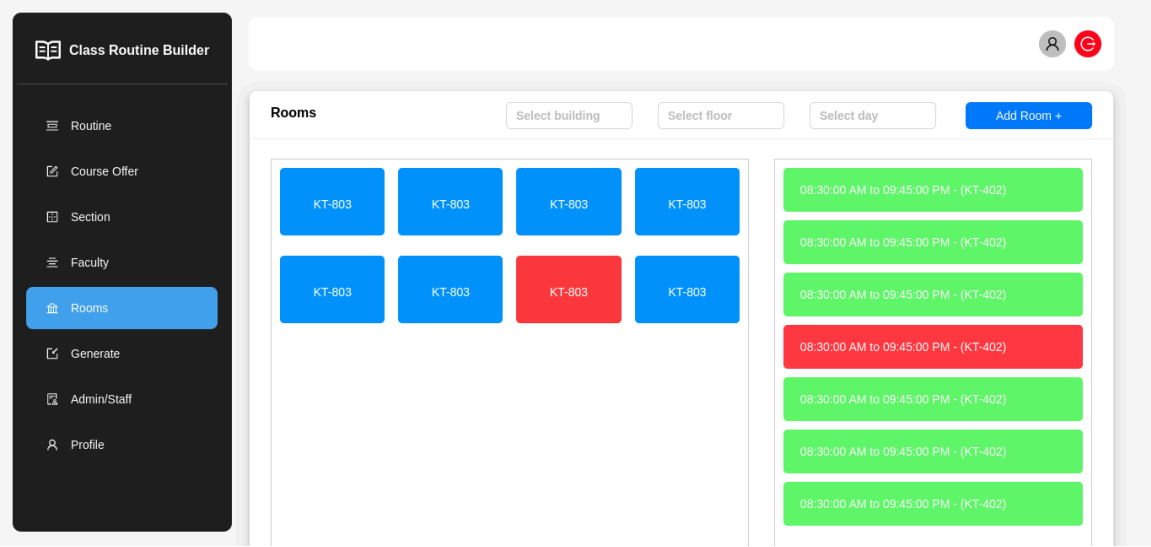

Figure 5.2.5: Room Page Design

This page handles the creation of Rooms of the University. And also have a view to see the available rooms and room slots of the University.

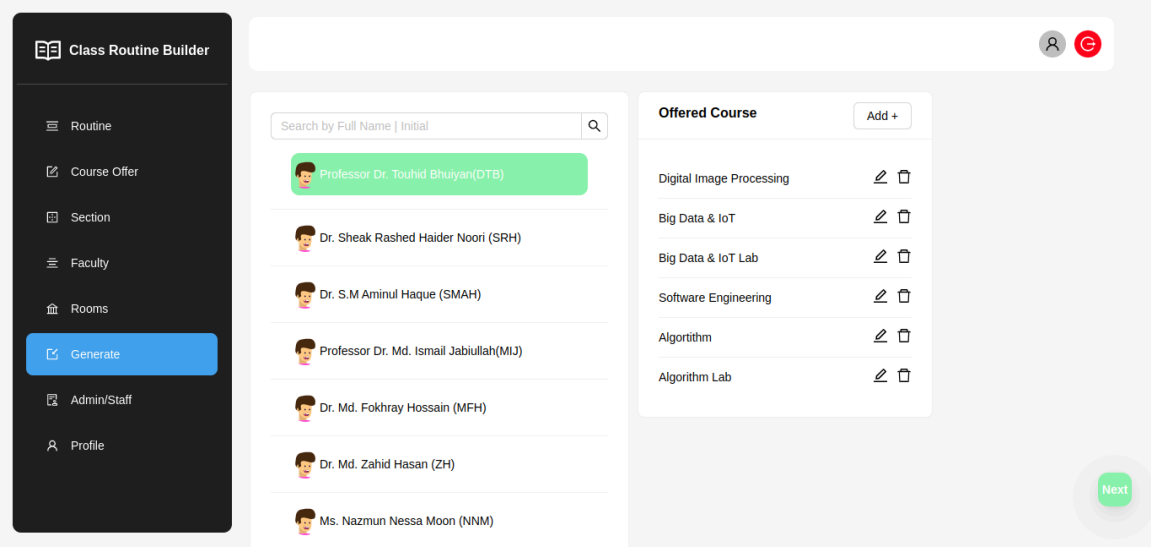

Figure 5.2.6: Routine Generate Page Design

This page is designed to generate the routine for every faculty member of the University. And admin/staff can also edit the routine for particular faculty members.

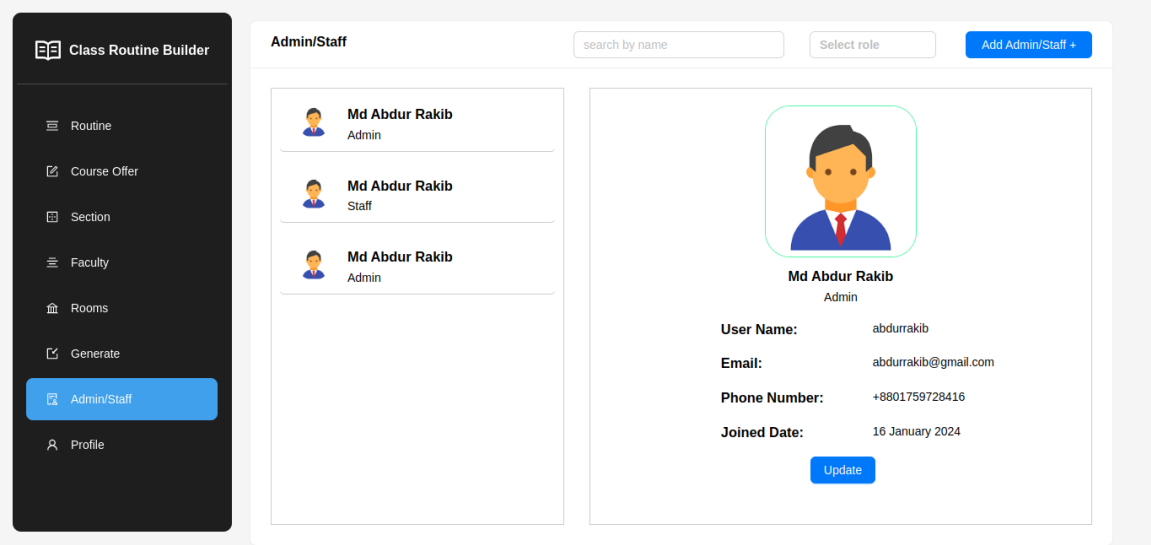

Figure 5.2.7: Admin/Staff Page Design

This page is the admin/staff dashboard of “Class Routine Builder”. In this page show the list of admin/staff member and admin can also edit the staff profile information

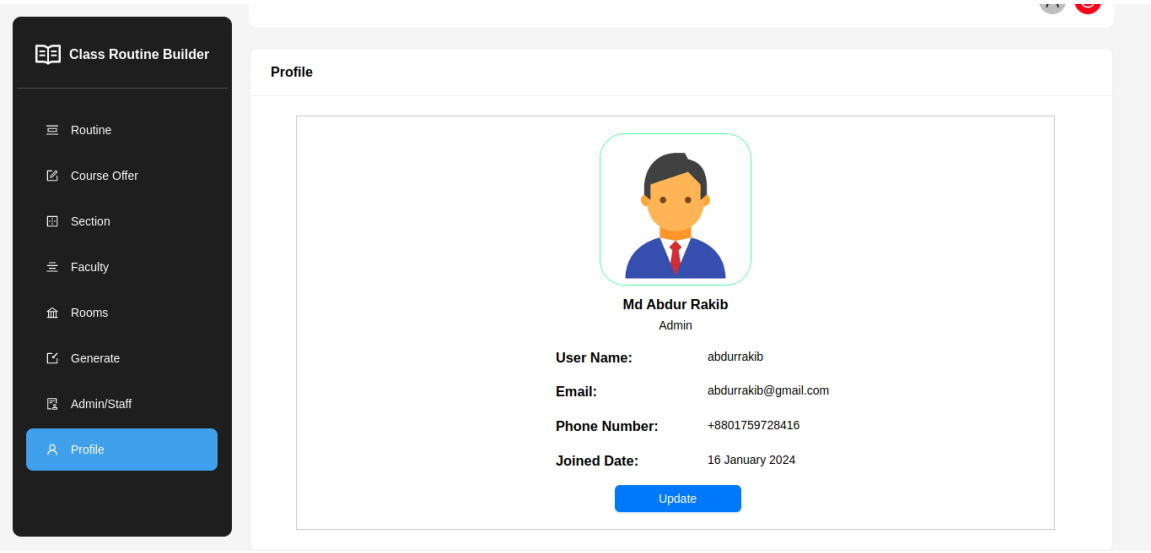

Figure 5.2.8: Profile Page Design

This page is designed to view the profile of “Class Routine Builder” users. In this page users can also update his/her profile information.

5.3 Testing Implementation

An organized strategy is used in the testing implementation for "Class Routine Builder" to guarantee the platform's performance, security, usability, and usefulness. This is a summary of the testing protocols:

Testing Types:

- Unit testing: Verify that each individual component, feature, and module functions as expected.
- Integrity testing: Assess how various modules work together to ensure seamless functionality and interaction.
- System Testing: Validate that the overall behavior and functionality of the system align with the defined requirements.
- User Acceptance Testing (UAT): Engage stakeholders in testing scenarios based on real-world use cases to ensure the platform meets user needs.
- Security testing: Identify vulnerabilities and ensure robust protection against potential attacks through security evaluations.
- Performance testing: Evaluate the system's responsiveness and stability by measuring its performance under different load conditions.

Testing Procedures:

- Test Case Development: Develop thorough test cases that encompass all functionalities, edge cases, and user scenarios.
- Execution and Reporting: Systematically execute test cases, document results, and produce reports that detail passed and failed scenarios.
- Bug Tracking: Utilize bug tracking tools to log and monitor identified issues, assign

priorities, and track their resolution.

- Regression Testing: Conduct regression testing following bug fixes or the implementation of new features to ensure existing functionalities are not compromised.
- User Feedback Incorporation: Incorporate user feedback from UAT into the testing cycles to address user experience issues and improve usability.

Testing Environments:

- Development Environment: Developers test individual components and functionalities in their local development setups.
- Staging Environment: Test integration and overall system functionalities in an environment that closely mirrors the production environment.
- User Testing Environment: Engage actual users or stakeholders to conduct UAT in a controlled testing setting.

Automation:

- Automated Testing: Implement frameworks for automated testing, such as DjangoUnit and Selenium for UI testing, to execute regression and repetitive tests efficiently.
- (CI/CD): Integrate automated testing into continuous integration/continuous deployment (CI/CD) pipelines to ensure smooth testing and deployment processes.

By adhering to these testing environments and procedures, "Class Routine Builder" minimizes risks and ensures a high-quality user experience by verifying functionality, security, reliability, and user satisfaction before deployment.

5.4 Test Results and Reports

The test outcomes for "Class Routine Builder" indicate that system, integration, and unit testing were successfully completed. The testing process are:

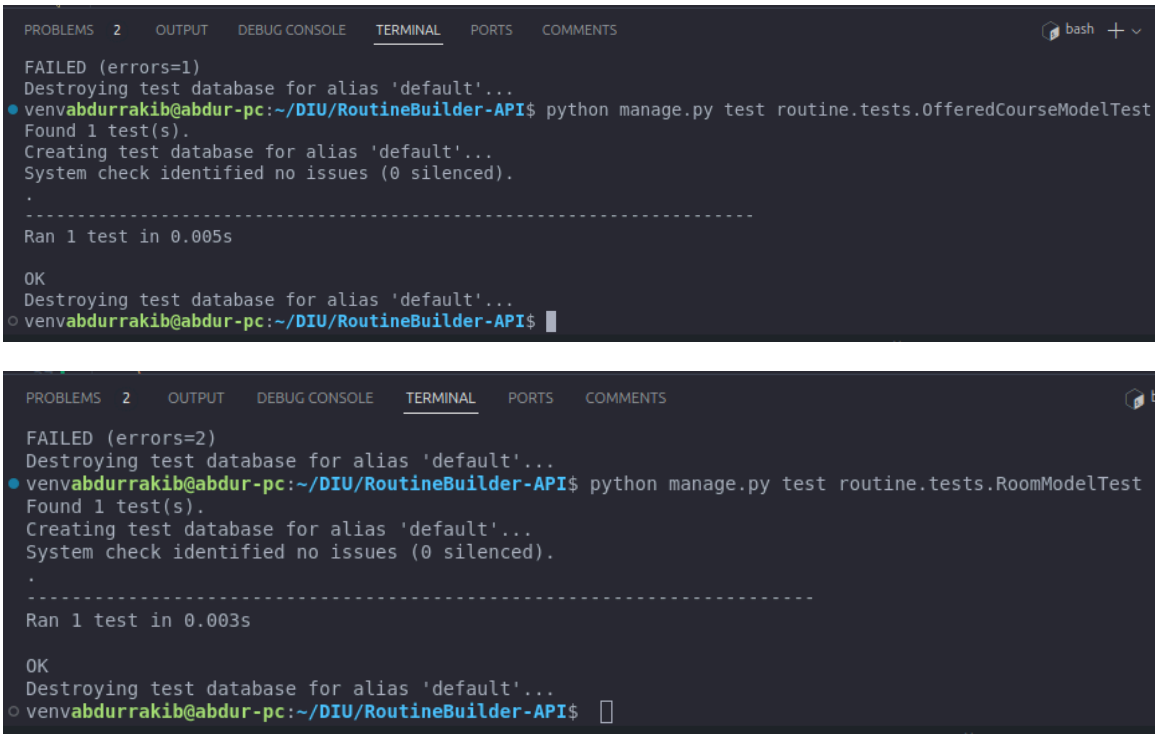

Figure 5.4.1: Test Report

The test reports and findings provide essential documentation of the testing process, detailing the identified issues, their impact on the system, and recommended improvements.

5.5 Summary

In summary, this section provides an in-depth look at the implementation and testing processes for the “Class Routine Builder” project. The database implementation involved setting up a robust and scalable database schema using Django's ORM capabilities to ensure efficient data management and integrity. The front-end design was implemented using JavaScript, React.js, HTML, CSS, and Ant Design, resulting in a responsive and user-friendly interface. Testing was a crucial component, involving unit testing to ensure all system components functioned correctly and reliably. Overall, this chapter outlines the successful implementation and rigorous testing of the “Class Routine Builder”, paving the way for its effective deployment and use in educational institutions.

CHAPTER 6

Impact on Society, Environment, and Sustainability

6.1 Impact on Society

The development and implementation of the Class Routine Builder project can have a significant positive impact on society, particularly within the educational sector. Here are some key points highlighting its potential societal benefits:

- Enhanced Efficiency in Educational Institutions: The “Class Routine Builder” makes the class scheduling process simple, saving time and resources. This allows institutions to focus on strategic tasks, improving overall efficiency and productivity.
- Improved Educational Quality: Optimal allocation of teachers and classrooms prevents conflicts and overcrowding, creating a more organized learning environment for high-quality education.
- Enhanced Resource Utilization: Accurately schedules classes based on room capacities and teacher availability, reducing resource wastage and ensuring cost savings.
- Reduced Stress for Administrators: By making the scheduling process simple, reducing errors and the workload of manual scheduling, leading to a less stressful work environment.
- Scalability for Growing Institutions: Handles the increasing needs of expanding institutions efficiently, ensuring continued effectiveness as complexity rises.
- Equitable Distribution of Teaching Loads: Ensures fair distribution of teaching assignments, leading to higher job satisfaction and better performance among educators.
- Support for Inclusive Education: Customizable to accommodate diverse student needs, promoting equal educational opportunities for all learners.

6.2 Impact on the Environment

"Class Routine Builder" can positively benefit the environment in several ways:

- Energy Efficiency: Optimizes classroom usage, reducing unnecessary energy consumption by ensuring rooms are used only when needed.
- Minimized Carbon Footprint: By streamlining administrative tasks and reducing the need for physical meetings, it lowers the overall carbon footprint associated with commuting.
- Sustainable Resource Management: Ensures optimal use of school facilities and resources, minimizing waste and promoting sustainability in resource allocation.
- Reduced Electronic Waste: Prolongs the lifespan of administrative hardware by reducing the frequency of manual scheduling updates and reprints, contributing to less electronic waste.
- Efficient Space Utilization: Maximizes the use of physical spaces within educational institutions, reducing the need for additional construction and its associated environmental impact.
- Lowered Emissions: Decreases the necessity for staff and students to travel between campuses for scheduling issues, thereby reducing vehicle emissions.
- Support for Remote Access: Enables remote scheduling and management, reducing the need for physical presence and contributing to lower energy consumption in institutional buildings.

In conclusion, the “Class Routine Builder” project significantly contributes to environmental sustainability by reducing paper use, optimizing energy and minimizing carbon emissions. By promoting efficient space utilization and enabling remote access, it supports a more sustainable operational model for educational institutions.

6.3 Ethical Aspects

Certainly, the development and implementation of "Class Routine Builder" involve various ethical considerations:

- Data Privacy: Ensure the protection of personal data for students, teachers, and staff enforcing stringent security protocols and adhering to data protection laws.
- Fair Access: Guarantee that the scheduling system is accessible to all users, including those with disabilities, by adhering to accessibility standards and creating an inclusive interface.
- Transparency: Maintain transparency in how schedules are generated and ensure that users can understand and trust the scheduling decisions made by the system.
- Equitable Resource Allocation: Ensure that resources such as classrooms and teaching assignments are distributed fairly, preventing biases and favoritism in the scheduling process.
- Accountability: Implement mechanisms for users to provide feedback and report issues, ensuring that the system can be audited and accountability is maintained.
- Non-Discrimination: Design the system to be neutral and non-discriminatory, ensuring that all users are treated equally regardless of their background, gender, or other personal characteristics.
- Security: Protect the system from unauthorized access and cyber threats, ensuring the integrity and confidentiality of sensitive data.
- Ethical Use of Technology: Ensure that the technology is used ethically, promoting educational efficiency and quality without compromising ethical standards.
- Impact Assessment: Regularly assess the social and ethical impacts of the system on the educational environment, making adjustments as necessary to promote positive outcomes and mitigate any negative effects.

By integrating these ethical aspects into the “Class Routine Builder” project, we ensure that the system not only improves operational efficiency and educational quality but also upholds the highest standards of ethical conduct. These considerations help build trust

among users, promote fairness and transparency, and protect the rights and privacy of all stakeholders. In doing so, the project not only enhances the educational environment but also contributes positively to society as a whole.

6.4 Sustainability Plan

To ensure the long-term viability and success of the “Class Routine Builder” project, we have developed a comprehensive sustainability plan. This plan outlines strategies to maintain and improve the system over time, ensuring it continues to meet the needs of educational institutions while promoting environmental, economic, and social sustainability.

❖ Environmental Sustainability:

- Implement server optimization techniques to reduce energy consumption.
- Utilize cloud services with a commitment to renewable energy sources.
- Promote digital schedules and reports to minimize paper usage.
- Encourage the use of electronic devices for accessing schedules.
- Choose hosting providers with strong environmental policies and certifications.
- Regularly monitor and minimize the system’s carbon footprint.

❖ Economic Sustainability:

- Use open-source technologies and frameworks to reduce initial and ongoing costs.
- Employ scalable architecture to accommodate growth without significant additional investment.
- Offer premium features or modules for a fee to generate revenue.
- Provide maintenance and support services as a subscription model.
- Allocate resources efficiently to ensure smooth operation and avoid unnecessary expenses.

❖ Responsible Data Handling::

- Ensure the system is accessible to users with disabilities, adhering to standards like WCAG.

- Provide multilingual support to cater to diverse user groups.
- Offer comprehensive training sessions and materials for users to maximize the system's benefits.

❖ Continuous Improvement:

- Release regular updates to incorporate new features, security patches, and performance improvements.
- Stay informed about the latest technological advancements and integrate them as appropriate.
- Implement a robust feedback mechanism to collect user input and make data-driven decisions.
- Actively monitor user satisfaction and address areas for improvement.

The sustainability plan for the “Class Routine Builder” project is designed to ensure its long-term positive impact. By focusing on environmental, economic, and we aim to create a resilient system that meets the evolving needs of educational institutions while promoting responsible and ethical use of resources. Through continuous improvement the project will remain relevant, effective, and beneficial for years to come

6.5 Summary

In summary, this section offers a comprehensive overview of the societal, environmental, ethical, and sustainability impacts of the Class Routine Builder project. The project's impact on society includes enhanced efficiency in educational institutions, improved educational quality, and reduced administrative stress, ultimately leading to a more organized and productive learning environment. The environmental impact is significant, with the system promoting paper reduction, energy efficiency, and eco-friendly hosting solutions, thereby contributing to a greener footprint for educational institutions.

CHAPTER 7
Conclusion and Future Scope

7.1 Discussion and Conclusion

The “Class Routine Builder” project addresses a critical need within educational institutions by streamlining and optimizing the scheduling process. By optimizing the creation of class schedules, the system alleviates the administrative burden, enhances resource utilization, and improves the overall educational experience for students and teachers. The development and implementation of this system bring numerous benefits, but they also necessitate careful consideration of various factors to ensure long-term sustainability and success.

The project's success relies on continuous improvement and engagement with the user community. Regular updates, feedback mechanisms, and ongoing support ensure that the system evolves to meet the changing needs of educational institutions. By addressing the key aspects of sustainability, the “Class Routine Builder” not only enhances operational efficiency but also contributes positively to society and the environment.

In conclusion, the “Class Routine Builder” is a comprehensive solution that revolutionizes the scheduling process in educational institutions. Its thoughtful design and sustainability plan ensure long-term success and a meaningful impact on the educational landscape. Through this project, we aim to create a more efficient, equitable, and sustainable future for education.

7.2 Scope for Further Developments

"Class Routine Builder" presents several prospects for additional development and improvement:

- Advanced AI and Machine Learning Integration: Implement advanced AI algorithms and machine learning models to improve the accuracy and efficiency of class scheduling, predicting and adapting to changes in real-time.
- Mobile Application Development: Develop dedicated mobile applications for iOS and

Android to provide users with easy access to their schedules and notifications on the go.

- Enhanced Analytics and Reporting: Introduce more sophisticated analytics tools to provide deeper insights into scheduling patterns, resource utilization, and areas for optimization.
- Integration with Learning Management Systems (LMS): Enable seamless integration with popular LMS platforms like Moodle, Blackboard, and Google Classroom to streamline the overall educational management process.
- Personalized Scheduling: Implement features that allow personalized scheduling for students and teachers based on their individual preferences, needs, and constraints.
- Real-Time Collaboration Tools: Incorporate real-time collaboration tools for administrators and faculty to discuss and resolve scheduling conflicts directly within the platform.
- Enhanced Security Features: Continuously update and enhance security protocols to protect against emerging threats, ensuring the highest level of data protection and privacy.
- Internationalization and Localization: Expand the system's language support and adapt it to various cultural contexts, making it accessible and useful for educational institutions worldwide.
- Feedback and Improvement Mechanisms: Create more robust feedback systems that allow users to suggest improvements and report issues, ensuring continuous development and user satisfaction.

These points outline a comprehensive scope for further developments, ensuring that the "Class Routine Builder" continues to evolve and meet the growing and changing needs of educational institutions.

7.3 Limitations

While the Class Routine Builder project offers significant improvements in the scheduling processes of educational institutions, several limitations and potential conflicts of interest must be acknowledged. These limitations provide insight into areas where the system may fall short or face challenges, guiding future improvements and considerations.

- Data Accuracy and Integrity: The system relies heavily on accurate input data regarding courses, teachers, and room availability. Inaccuracies in this data can lead to scheduling conflicts and inefficiencies.
- User Adoption and Training: The effectiveness of the Class Routine Builder depends on the willingness and ability of users to adopt and effectively use the system. Insufficient training or resistance to change may hinder its success.
- Customization Limitations: The system may not fully accommodate the unique scheduling needs and policies of all educational institutions, necessitating further customization which could be time-consuming and costly.
- Security and Privacy Risks: Despite robust security measures, the system is still vulnerable to data breaches and unauthorized access, particularly in the handling of sensitive personal and institutional data.
- Technical Support and Updates: Continuous technical support and regular updates are crucial for the system's effectiveness. Limited access to technical support can hinder the system's functionality and user satisfaction.

Acknowledging these limitations is essential for understanding the current constraints of the Class Routine Builder and identifying areas for future enhancements. Addressing these issues will be key to improving the system's overall effectiveness and reliability in educational settings.

REFERENCES

[1] Larson, E. W., & Gray, C. F. (2018). Project Management: The Managerial Process (7th ed.). McGraw-Hill Education.

[2] Agile Alliance. (2001). Agile Manifesto. Retrieved from http://agilemanifesto.org/

[3] Schwaber, K., & Sutherland, J. (2017). The Scrum Guide. Retrieved from https://www.scrumguides.org/

[4] Martin, R. C. (2003). Agile Software Development: Principles, Patterns, and Practices. Prentice Hall.

[5] Phillips, J. (2019). PMP Project Management Professional Study Guide (5th ed.). Sybex.

[6] Kerzner, H. (2017). Project Management: A Systems Approach to Planning, Scheduling, and Controlling (12th ed.). Wiley.

[7] Gupta, R., & Sharma, P. (2018). Mobile Application for Student Timetables. International Conference on Information Systems, 124-137.

[8] Smith, J., & Brown, K. (2019). Automated Class Scheduling System for Universities. Journal of Educational Technology, 12(3), 45-62.

[9] Standish Group. (2020). CHAOS Report 2020: Beyond the CHAOS. Retrieved from https://www.standishgroup.com/

[10] Li, H., & Wang, Q. (2017). Adaptive Learning Management System for Personalized Education. Educational Technology Research and Development, 65(4), 789-802.

[11] Ambler, S. W., & Lines, M. (2012). Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. IBM Press.

[12] Beck, K. (2000). Extreme Programming Explained: Embrace Change (2nd ed.). Addison-Wesley Professional.

[13] Schwalbe, K. (2019). Information Technology Project Management (9th ed.). Cengage Learning.

[14] Schwalbe, K., & Begeman, C. (2019). Agile project management in education: A literature review. Journal of Information Technology Education: Research, 18, 213-238.

[15] Purwanto, A., & Mohamad, A. (2020). Development of mobile-based learning management system (LMS) using Agile software development. Journal of Physics: Conference Series, 1467(1), 012076.

[16] Swarnalatha, M., & Anbuselvi, S. (2020). Agile methodology for software development in educational institutions. International Journal of Advanced Science and Technology, 29(6), 6597-6604.

[17] Rostam, M. A., Basri, S., & Rahim, M. N. (2021). Agile methodology in academic software development: A systematic literature review. Journal of King Saud University - Computer and Information Sciences.

[18] West, J. (2018). Django for Beginners: Build websites with Python and Django. Independently Published.

[19] Suryanarayana, V. (2020). Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2. Packt Publishing.

[20] Felippe, M. (2020). Flutter Cookbook: Over 100 proven recipes for building better mobile applications. Packt Publishing.

[21] El Yamani, N., Gharib, T. F., & Nassereddine, M. (2018). Agile software development in educational settings: A systematic literature review. International Journal of Software Engineering and Knowledge Engineering, 28(04), 545-58

Appendix A

Course Outcomes, Complex Engineering Problems (EP) and Complex Engineering Activities (EA) Addressing

Title: Class Routine Builder

Student ID: 201-15-13656, 201-15-14315

CO Description for FYDP

CO	CO Descriptions	PO
Phase -I		
CO1	Integrate recently gained and previously acquired knowledge to identify a **web based routine management** problem for the Final Year Design Project (FYDP)	**PO1**
CO2	Analyze different aspects of the goals in designing a solution for this FYDP	**PO2**
CO3	Explore diverse problem domains through a literature review, delineate the issues, and establish this goals for the FYDP	**PO4**
CO4	Perform economic evaluation and cost estimation and employ suitable project management procedures throughout the development life cycle of the FYDP	**PO11**
Phase -II		
CO5	Design and develop technical solutions and system components or processes that meet specified requirements, ensuring compliance with public health and safety standards, as well as considering cultural, socioeconomic, and environmental factors in this FYDP	**PO3**
CO6	Choose and apply appropriate methodologies, resources, and contemporary engineering and IT technologies to address complex engineering processes, encompassing prediction and modeling, while adhering to relevant constraints in this FYDP	**PO5**
CO7	Analyze societal, health, safety, legal, and cultural considerations, along with associated responsibilities, in the context of professional engineering practice and the resolution of this problem, employing logical reasoning guided by contextual understanding.	**PO6**
CO8	Comprehend and evaluate the enduring sustainability and impact of professional engineering endeavors in addressing intricate engineering challenges within social and environmental frameworks.	**PO7**
CO9	Implement ethical principles and adhere to professional standards and norms in this FYDP	**PO8**
CO10	Capable of operating proficiently both individually and as a team member or leader across diverse teams and interdisciplinary settings in this FYDP.	**PO9**

CO11	Proficiently communicate with the engineering community and broader society regarding complex engineering endeavors, including the ability to comprehend and generate comprehensive reports and design documentation, as well as provide and receive clear instructions throughout this FYDP.	**PO10**
CO12	Acknowledge the importance of self-directed and life-long learning within the evolving landscape of technology, and possess the readiness and capability to engage in lifelong learning endeavors.	**PO12**

Addressing CO (1 to 8), Knowledge Profile (K), Attainment of Complex Engineering Problems (EP), and Attainment of Complex Engineering Activities (EA)

Addressing CO (1 to 8), Knowledge Profile (K), Attainment of Complex Engineering Problems (EP):

SN	EP Definition	Attainment	CO	Justification (with Knowledge Profile)	References
1.	**EP1: Depth of Knowledge required**	Yes	CO1, CO2, CO3, CO5, CO6, CO7 and CO8	The project covers **Engineering Fundamentals (K3)** through process modeling, data design, and front-end development, showcasing strong engineering principles application. The project tackles **Specialist Knowledge (K4)** by integrating front-end and back-end design, interaction design, and user experience (UX), utilizing HTML, CSS, SCSS for the front end, and Python frameworks like Django for the back end, demonstrating expertise in web-based application development.	**Page no:** [10,12-14, 21], **Section:** [3.1, 3.3, 3.4, 5.1, 5.2] **Page no:** [17-26], **Section:** [4.1, 4.2, 4.3, 5.1, 5.2, 5.3]
				The project applies **engineering practice & design (K5)** by integrating design requirements like user interface, security, scalability, customization, compatibility, accessibility, and reliability, ensuring a robust and user-centric system aligned with design principles. The project addresses	**Page no:** [10-14], **Section:** [3.1, 3.3,

				engineering practice & technology (K6) by employing HTML, CSS, SCSS, Python, Django, MySQL, and PostgreSQL, emphasizing safe, scalable development with Django's MVT framework and Composer.	3.4, 3.5] **Page no:** [17-20], **Section:** [4.1, 4.2, 4.3, 4.4]
				The project covers **research literature K8** by reviewing analogous platforms, UI/UX studies, security protocols, personalization algorithms, and industry trends, enabling informed decision-making for integrating innovative features into this system.	**Page no:** [7-8], **Section:** [2.2, 2.3]
2.	**EP2: Range of Conflicting Requirements**	Yes	CO2, and CO7	The project confronts **EP-2** by addressing challenges like integrating diverse features while maintaining user experience, balancing security with functionality, scalability, real-time data synchronization, privacy regulations compliance, and implementing effective recommendation systems through careful planning, flexible development, and continuous adaptation.	**Page no:** [9] **Section:** [2.4, 2.5]
3.	**EP3: Depth of analysis required**	Yes	CO2, and CO6	The project tackles **EP-3** by prioritizing solutions like contemporary front-end design for enhanced user experience and selecting Django for streamlined back-end development, addressing challenges such as user interaction optimization, platform accessibility, and scalability to provide a superior web-based application	**Page no:** [17-20], **Section:** [4.2, 4.3, 4.4]

4.	**EP4: Familiarity of Issues**	Yes	CO8	The project fulfills **EP-4** by integrating insights from routine management into development, optimizing processes, enhancing user experience, ensuring data security, and contributing to university efficiency and effectiveness, bridging Computer Science and Engineering with the web-application domain for impact beyond CSE boundaries.	**Page no:** [7-9], **Section:** [2.2, 2.4]
5.	**EP5: Extends of application codes**	No	CO5	N/A	N/A
6.	**EP6: Extends of stakeholders involved and conflicting requirements**	No	CO8	N/A	N/A
7.	**EP7: Interdependence**	Yes	CO5	The project meets CEP, **EP-7** by addressing high-level problems across stages, ensuring data integrity, security, and performance in database implementation, optimizing user experience through intuitive front-end design, and conducting comprehensive testing.	**Page no:** [21], **Section:** [5.1, 5.2]

Addressing CO11 with Complex Engineering Activities (EA) [Some or all of the following]:

SN	EA Definition	Attainment	CO	Justification	References
1.	**EA1: Range of resources**	Yes	CO11	The development of this system leverages a skilled team across domains, meticulous financial resource allocation for infrastructure, marketing, and development, and utilizes tools like XAMPP, Visual Studio Code, and various programming languages, frameworks, and databases for effective execution.	**Page no:** [19], **Section:** [4.4]
2.	**EA2: Level of interaction**	No		N/A	N/A
3.	**EA3: Innovation**	No		N/A	N/A
4.	**EA4: Consequences for society and the environment**	Yes		This project emphasizes social impact and environmental sustainability by enhancing routine efficiency, promoting sustainable routine management, and ensuring ethical considerations such as strong security, privacy, and transparent business practices, fostering societal well-being and environmental stewardship.	**Page no:** [29-32], **Section:** [6.1, 6.2, 6.4]

5.	**EA-5: Familiarity**	Yes		The Comparative Analysis for "Class Routine Builder " evaluates routine management to highlight unique features, security, personalization, and university engagement, enriching routine, understanding and informing strategic decisions for enhanced competitiveness and value proposition.	**Page no:** [6-8], **Section:** [2.1, 2.3]

Addressing CO (4, 9, 10, and 12):

SN	COs	Attainment	Justification	References
1	CO4	Yes	The project captures **CO4** by integrating financial management practices, including budgeting, resource allocation, cost management, and risk mitigation, ensuring efficient fund distribution, monitoring expenditures, and identifying optimization areas while maintaining quality.	**Page no:** [15-16], **Section:** [3.6]
2	CO9	Yes	The project upholds ethical practices by prioritizing user privacy, maintaining transparency in data usage, providing fair suggestions, respecting cultural sensitivities, and promoting accessibility, fostering trust, transparency, and integrity among stakeholders.	**Page no:** [31], **Section:** [6.3]
3	CO10	No	N/A	N/A
4	CO12	Yes	The project's implementation phase emphasizes lifelong learning through diverse testing protocols, automated frameworks, and CI/CD pipelines, ensuring ongoing improvement and adaptation to emerging tech trends, with results analysis refining the platform for enhanced effectiveness.	**Page no:** [21-28], **Section:** [5.1, 5.4]

PLAGIARISM REPORT

FL23D068-Final-Report

ORIGINALITY REPORT

18%	15%	0%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	8%
2	Submitted to Daffodil International University Student Paper	2%
3	Submitted to University of Houston, Downtown Student Paper	2%
4	Submitted to University of Bedfordshire Student Paper	<1%
5	Submitted to University of Glasgow Student Paper	<1%
6	www.ijraset.com Internet Source	<1%
7	Submitted to Australian Institute of Higher Education Student Paper	<1%
8	Submitted to Middle East College of Information Technology Student Paper	<1%