



Panskura Banamali College (Autonomous)

Routine Management System

A Project Paper

Submitted By

PRIYA JANA

(Reg No-2019PBC01350)

In partial fulfilment for the award of the degree of

M. Sc. Computer Science

Under the Supervision of

MR. SANTOSH NANDI

(PROFESSOR)



PANSKURA BANAMALI COLLEGE
Autonomous College

Panskura Banamali College (Autonomous)

BONAFIDE CERTIFICATE

Certified that this project report “Routine Management System” is the Bonafide work of
PRIYA JANA (Reg No-2019PBC01350)” who carried out the project work under my
supervision.

Signature of Supervisor
Mr. Santosh Nandi
Professor
Panskura Banamali College



PANSKURA BANAMALI COLLEGE
Autonomous College

Panskura Banamali College (Autonomous)

Acknowledgement

I am immensely grateful to my guide, Prof. Santosh Nandi, for their invaluable guidance and support throughout this project. Their insights and encouragement were instrumental in the successful completion of this work. I also extend my heartfelt thanks to the entire staff of my department for their assistance and cooperation.

Special thanks to Prof. Santosh Nandi, Head of the Department of Information Technology, for their continuous support and encouragement. Additionally, I would like to express my sincere appreciation to our Principal, Prof. (Dr.) Nandan Bhattacharyya, for their valuable support.

Finally, I thank all the teaching and non-teaching staff members of my department, my colleagues, and my parents, whose help and encouragement were vital in completing this project successfully.

Signature of Head

Asst. Prof. Santa Singha
Head of the Department
Panskura Banamali College

Signature of Supervisor

Mr. Santosh Nandi
Professor
Panskura Banamali College

Index

S. No.	Contents	Page No.
1	Abstract	1
2	Introduction	2
3	Scope	4
4	Features	5
5	Requirement Analysis	7
6	Design Approach	12
7	Data flow Diagram	14
8	Use case Diagram	16
9	Sequence Diagram	20
10	Class Diagram	21
11	Entity Relationship Diagram	22
12	Screenshots	28
13	Conclusion	41
14	Reference	42

Abstract:

The Routine Management System (RMS) is a comprehensive web-based application designed to streamline the process of creating and managing schedules for batches, classes, and departments within educational institutions. Developed using ASP.NET Core Web API for the backend and React for the frontend, RMS offers a user-friendly interface that facilitates efficient management of teacher and student information, as well as the creation and editing of schedules and reports.

Key features of the RMS include:

- **Teacher and Student Management Systems:** Efficiently handle information related to teachers and students.
- **Batch and Department Management:** Manage and organize various batches and departments.
- **Schedule Creation and Editing:** Simplify the process of schedule creation and modification with export options to Excel and PDF.
- **Real-Time Schedule Viewing:** Provide teachers and students with real-time access to their schedules.
- **Teacher Attendance Tracking and Absence Notification System:** Track teacher attendance and notify relevant parties of absences.
- **Student Attendance Tracking and Grading System:** Monitor student attendance and integrate it with grading systems.
- **AI-Based Routine Generation:** Utilize artificial intelligence to generate optimized schedules.
- **Notification System:** Send notifications to teachers and students regarding schedule changes and important updates.

The RMS aims to reduce the time and effort required to create and manage schedules, thereby enhancing the overall efficiency and productivity of educational institutions. Its robust features and intuitive interface make it an effective tool for administrators, teachers, and students alike. By leveraging modern technologies, RMS ensures a seamless and integrated approach to routine management, addressing the diverse needs of its users.

Introduction:

The Routine Management System (RMS) offers a significant value proposition for educational institutions by streamlining and automating the process of schedule management. Specifically designed for colleges and universities, RMS addresses the complexities involved in creating and maintaining schedules for various departments and semesters.

In a typical college, each department oversees multiple semesters of students, with Heads of Departments (HODs) or administrators responsible for creating routines. This process involves coordinating available teachers, ensuring no scheduling conflicts, and managing daily adjustments due to teacher absences. Traditionally, this task is labour-intensive and prone to errors when done manually.

RMS provides a comprehensive solution to these challenges by offering the following benefits:

- **Efficient Schedule Creation:** HODs and administrators can easily create and edit schedules using RMS. The system checks for teacher availability and prevents overlapping classes, ensuring a conflict-free timetable.
- **Dynamic Adjustments:** The system allows for real-time adjustments. If a teacher is absent, RMS helps find a suitable replacement quickly, minimizing disruptions to the schedule.
- **Automated Attendance and Grading:** RMS tracks student attendance automatically and integrates this data with the grading system, simplifying the grading process and reducing the administrative burden.
- **Centralized Information Management:** Teachers and students can access their schedules in real-time, receive notifications about changes, and view attendance records. This centralized access improves communication and ensures everyone is informed.
- **Resource Distribution:** The system facilitates the distribution and storage of class notes, assignments, and other educational resources, making them easily accessible for future reference.

By implementing RMS, educational institutions can enhance their operational efficiency, reduce administrative workload, and provide a better experience for both

teachers and students. The system's robust features and user-friendly interface make routine management seamless, allowing institutions to focus more on educational quality and less on administrative tasks.

Purpose of RMS:

The primary purpose of the RMS is to provide an efficient, user-friendly, and automated system for managing educational schedules and related administrative tasks. The specific goals include:

- **Automating Schedule Creation:** To reduce the time and effort required to create and manage class schedules.
- **Enhancing Communication:** To provide real-time updates and notifications to teachers and students.
- **Improving Resource Management:** To streamline the distribution and storage of educational materials.
- **Facilitating Attendance Tracking and Grading:** To integrate attendance records with the grading system for easier evaluation of student performance.
- **Optimizing Teacher Utilization:** To ensure teachers' schedules are efficiently managed without conflicts.

Advantages:

The implementation of RMS brings several advantages to educational institutions:

- **Efficiency:** Automates routine tasks, significantly reducing the administrative workload for HODs and administrators.
- **Accuracy:** Minimizes errors in schedule creation and management, ensuring that all classes are scheduled without conflicts.
- **Flexibility:** Allows for quick adjustments to schedules in response to teacher absences or other changes, ensuring minimal disruption.
- **Transparency:** Provides real-time access to schedules and attendance records for teachers and students, enhancing transparency and communication.
- **Resource Management:** Simplifies the distribution and storage of class notes and assignments, making them easily accessible for future use.
- **Enhanced Productivity:** Frees up time for HODs, teachers, and administrators to focus on more critical tasks, thereby improving overall productivity.
- **Improved Student Performance:** By integrating attendance with grading systems, RMS helps in accurately tracking student performance and identifying areas that need attention.

Scope of RMS:

The Routine Management System (RMS) aims to address the existing limitations in educational institutions' scheduling processes. The current manual approach to creating and managing schedules is prone to errors, inconsistencies, and inefficiencies, leading to several issues:

Existing System with Limitations

Manual Schedule Creation

- Schedules are often created manually, leading to a high risk of errors and inconsistencies.
- This manual process is time-consuming and labour-intensive, requiring significant administrative resources.
- Errors in scheduling can result in conflicts, misunderstandings, and disruptions to the educational process.

Lack of Real-Time Updates

- Changes to schedules, such as teacher absences, are not communicated in real-time, causing confusion and delays.
- This lack of transparency and communication can lead to misunderstandings, miscommunications, and frustration among teachers, students, and administrators.

Inefficient Resource Management

- Distribution and storage of class notes and assignments are not centralized, leading to difficulties in accessing and managing these resources.
- This lack of organization and accessibility can result in wasted time and resources, as well as decreased productivity.

Tedious Attendance Tracking

- Attendance is tracked manually, which is time-consuming and prone to inaccuracies.
- This manual process can lead to errors, inconsistencies, and inaccuracies in attendance records, making it challenging to evaluate student performance accurately.

No Integrated Grading System

- Attendance data is not effectively integrated with the grading system, making it challenging to evaluate student performance accurately.
- This lack of integration can result in inaccurate assessments of student performance, leading to unfair grading and potential negative impacts on student motivation and engagement.

Proposed System Features

To address these limitations, the RMS proposes the following features:

Automated Schedule Creation and Management

- Streamlines the process of creating and editing schedules, ensuring no conflicts and reducing manual workload.
- Automates the scheduling process, eliminating the risk of errors and inconsistencies.

Real-Time Schedule Viewing and Notifications

- Provides teachers and students with real-time access to their schedules and sends notifications for any changes or updates.
- Ensures that all stakeholders are informed and up-to-date, reducing confusion and delays.

Teacher and Student Management

- Efficiently manages information related to teachers and students, including attendance tracking.
- Provides a centralized platform for managing teacher and student data, reducing administrative workload.

AI-Based Routine Generation

- Utilizes artificial intelligence to generate optimized schedules that maximize resource utilization.
- Ensures that schedules are created efficiently, considering teacher availability, student needs, and resource constraints.

Centralized Resource Distribution

- Facilitates the distribution and storage of class notes, assignments, and other educational materials in a centralized manner.
- Reduces the administrative burden of managing resources and ensures that all stakeholders have access to the necessary materials.

Attendance Tracking and Integration with Grading System

- Automates attendance tracking and integrates it with the grading system, simplifying student performance evaluation.
- Ensures that attendance data is accurate, consistent, and integrated with the grading system, providing a comprehensive view of student performance.

Dynamic Adjustments

- Allows for quick adjustments to schedules in response to teacher absences, ensuring minimal disruption to classes.
- Ensures that the scheduling process is flexible and adaptable, responding to changing circumstances and minimizing disruptions.

By addressing these limitations and implementing these features, the RMS aims to improve the efficiency, accuracy, and effectiveness of educational institutions' scheduling processes, ultimately enhancing the overall educational experience.

Analysis:

The analysis phase involves understanding the requirements for the Routine Management System (RMS) to ensure it meets the needs of its users. This section will outline the software and hardware requirements necessary for implementing and running the system effectively.

Requirement Analysis:

The requirement analysis focuses on identifying the functional and non-functional requirements for the RMS. It ensures that all stakeholders' needs are addressed and that the system functions as intended.

Functional Requirements

Functional requirements define the specific behaviour or functions of the RMS. These include:

User Authentication and Authorization

- Secure login mechanism for administrators, teachers, and students
- Role-based access control to ensure that users can only access features and data relevant to their role
- Multi-factor authentication for added security
- User profiling and session management to track user activity and ensure secure access

Schedule Management

- Ability to create, edit, and delete schedules for various batches and departments
- Schedule templates and recurring events for easy scheduling
- Conflict detection and resolution to prevent scheduling conflicts
- Real-time schedule viewing and editing for administrators, teachers, and students
- Export schedules to Excel and PDF formats for easy sharing and archiving

Teacher and Student Management

- Features to manage teacher and student information, including:
 - Adding, editing, and deleting records
 - Managing teacher and student profiles, including contact information and biographies
 - Tracking teacher and student attendance and performance metrics
 - Integrating with grading system for accurate performance evaluation

Attendance Tracking

- Automated attendance tracking for teachers and students
- Integration with grading system to ensure accurate performance evaluation
- Real-time attendance tracking and reporting
- Ability to track attendance for individual classes, batches, and departments

Notification System

- Real-time notifications for schedule changes, teacher absences, and other important updates
- Customizable notification settings for administrators, teachers, and students
- Ability to set reminders and alerts for upcoming events and deadlines
- Integration with other system features for seamless communication

Resource Management

- Upload, distribute, and store class notes, assignments, and other educational resources
- Resource categorization and tagging for easy search and retrieval
- Resource sharing and collaboration features for teachers and students
- Integration with scheduling system for easy access to resources during classes

AI-Based Routine Generation

- Automated generation of optimized schedules using artificial intelligence
- Integration with scheduling system to ensure seamless scheduling
- Ability to generate schedules based on teacher availability, student needs, and resource constraints
- Real-time scheduling adjustments to accommodate changes and updates

Export Functionality

- Export schedules and reports to Excel and PDF formats
- Ability to customize export formats and settings
- Integration with other system features for easy data analysis and reporting

Dashboard and Reporting

- Comprehensive dashboards and reports for administrators to monitor schedules, attendance, and performance metrics
- Real-time reporting and analytics for easy data analysis and decision-making
- Customizable reporting settings and filters for tailored insights
- Integration with other system features for seamless data tracking and analysis

Non-Functional Requirements

Non-functional requirements define the system's operational attributes, including:

- **Performance:** The system should handle multiple concurrent users without significant performance degradation.
- **Scalability:** The system should be scalable to accommodate an increasing number of users and data.
- **Reliability:** The system should be reliable and available with minimal downtime.
- **Security:** The system should ensure data security and privacy, protecting sensitive information from unauthorized access.
- **Usability:** The system should have an intuitive and user-friendly interface to facilitate ease of use for all users.
- **Maintainability:** The system should be maintainable, allowing for easy updates and modifications.

Software Requirement Specification

The Software Requirement Specification (SRS) details the software components and configurations required for the RMS.

Backend

Framework

- ASP.NET Core Web API: A robust and scalable framework for building web APIs
- Version: 8.0

Programming Language

- C#: A modern, object-oriented programming language for building robust and maintainable code
- Version: 12

Database

- SQL Server: A reliable and scalable relational database management system
- Version: 19
- Database Schema: A well-designed schema for storing and retrieving data efficiently

API Documentation

- Swagger: A tool for generating and managing API documentation
- Version: Latest stable version

Authentication

- Identity Server: A secure and scalable authentication service for managing user identities
- Version: Latest stable version

Frontend

Framework

- React: A popular and widely used JavaScript library for building user interfaces
- Version: Latest stable version

Programming Languages

- JavaScript: A widely used and versatile programming language for building dynamic web applications
- TypeScript: A statically typed version of JavaScript for building robust and maintainable code
- Version: Latest stable version

State Management

- Redux: A state management library for managing application state efficiently
- Version: Latest stable version

UI Library

- Material-UI or similar UI libraries: A library for building visually appealing and user-friendly interfaces
- Version: Latest stable version

2.2.3 Development Tools

IDE

- Visual Studio Code: A popular and widely used integrated development environment for building and debugging code
- Version: Latest stable version

Version Control

- Git: A widely used and versatile version control system for managing code changes and collaborations
- Version: Latest stable version

Package Manager

- npm: A package manager for managing dependencies and installing packages
- Version: Latest stable version

Build Tools

- Webpack: A popular and widely used build tool for building and optimizing code
- Version: Latest stable version
- Babel: A transpiler for converting modern JavaScript code to older versions for compatibility
- Version: Latest stable version

Design Approach:

The design approach for the Routine Management System (RMS) focuses on outlining the architecture and data flow within the system. This section includes a brief description of Data Flow Diagrams (DFDs), followed by detailed data flow diagrams and entity relationships.

Data Flow Diagram (DFD):

A Data Flow Diagram (DFD) is a graphical representation of the flow of data within a system. It illustrates how data is processed by the system in terms of inputs and outputs. DFDs are used to visualize the movement of data between different processes, data stores, and external entities.

DFDs are divided into different levels to show varying degrees of detail. The highest level is the context diagram, which provides an overview of the system. Lower levels show detailed views of specific processes.

Data Flow Relationships:

Users

- One-to-many with UserClaims
- One-to-many with UserLogins
- One-to-many with UserRoles
- One-to-many with UserTokens
- One-to-many with Photos

Roles

- One-to-many with RoleClaims
- One-to-many with UserRoles

Departments

- One-to-many with Batches
- One-to-many with Teachers

Subjects

- One-to-many with BatchSubjects
- One-to-many with TeacherSubjects

Batches

- One-to-many with BatchSubjects
- One-to-many with Students
- One-to-many with Schedules

Teachers

- One-to-many with TeacherSubjects
- One-to-many with Schedules

Routine

- One-to-many with Schedules

Schedules

- One-to-many with TimeSlots

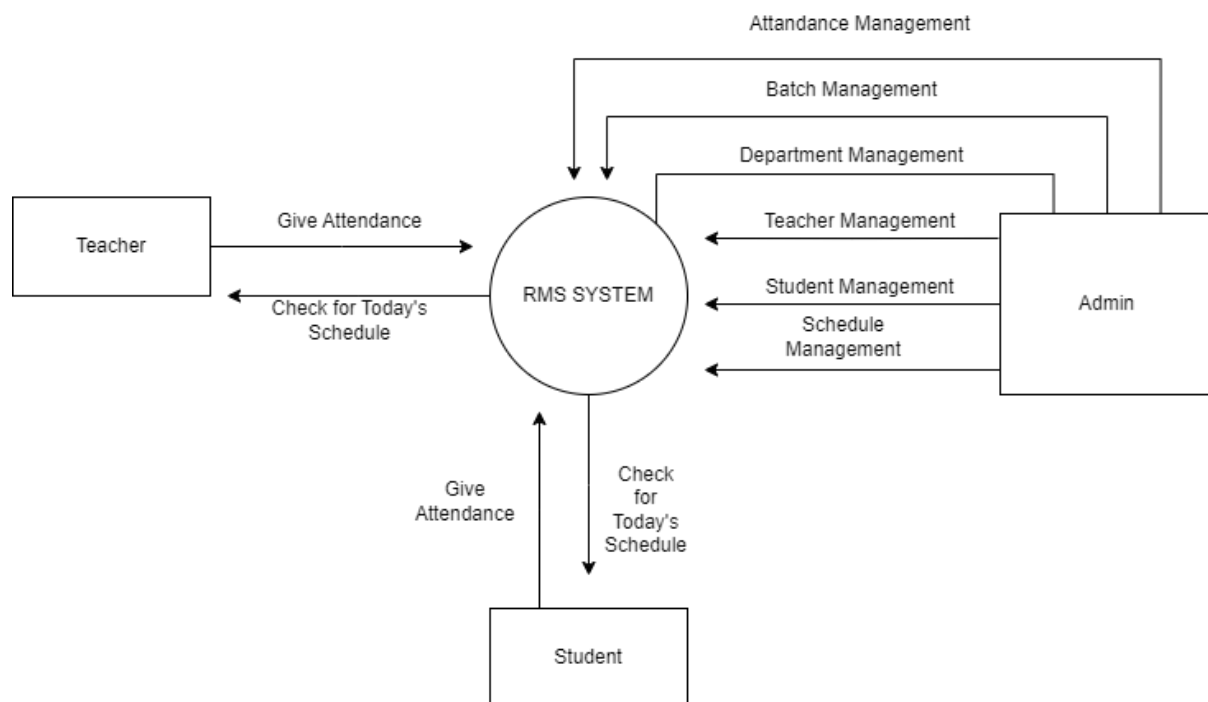
Tables

1. Roles
2. RoleClaims
3. UserClaims
4. UserLogins
5. UserRoles
6. UserTokens
7. Photos
8. Users
9. Departments
10. Subjects
11. Batches
12. Teachers
13. BatchSubjects
14. Students
15. TeacherSubjects
16. Routine
17. Schedules
18. TimeSlots

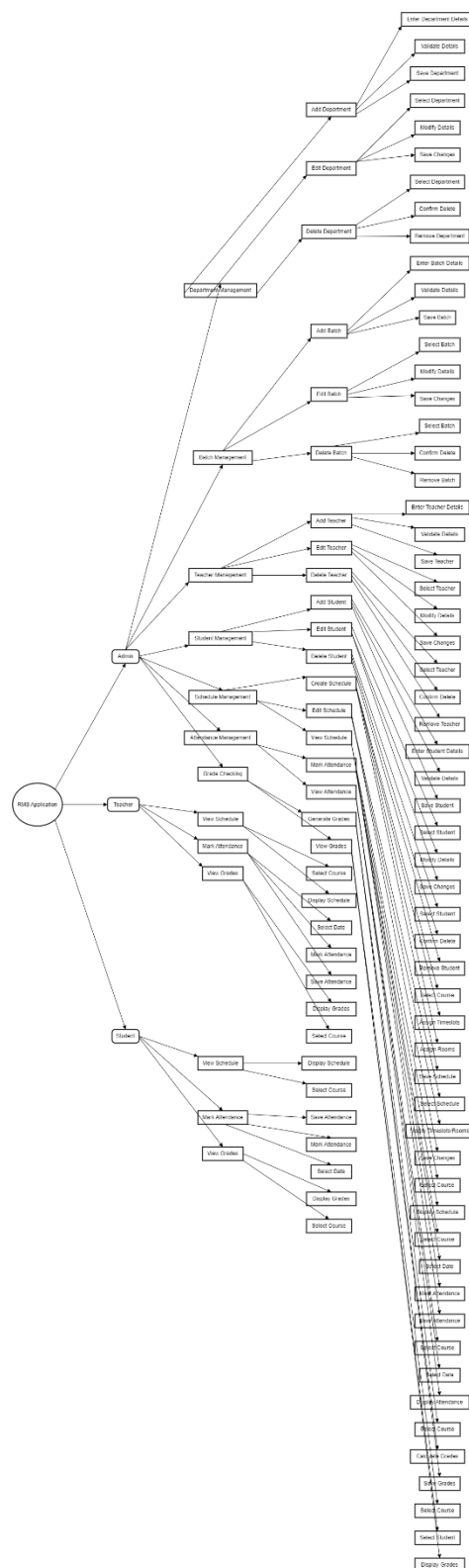
Data Flow Diagram Documentation

Context Level DFD: In the Context Level the whole system is shown as a single process. No data stores are shown. Inputs to the overall system are shown together with data sources (as External entities). Outputs from the overall system are shown together with their destinations (as External entities).

TOP Level DFD:



Detailed Level DFD:



Use Case Documentation:

Use Case Diagrams: A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse. An actor is a person, organization, or external system that plays a role in one or more interactions with your system.

Top-Level Use Case Diagram:

Description

The top-level use case diagram provides an overview of the primary interactions within the Routine Management System (RMS). It includes the main actors (Administrator, Teacher, and Student) and the high-level functionalities they interact with.

Actors

1. Administrator:

- Manages all system functions, including:
 - User Management:
 - Manage Teachers
 - Manage Students
 - Manage Admins
 - Schedule Management:
 - Create Schedule
 - Edit Schedule
 - View Schedule
 - Attendance Management:
 - Mark Teacher Attendance
 - Mark Student Attendance
 - Notification System:
 - Send Notifications
 - Receive Notifications
 - Report Generation:
 - Export to Excel
 - Export to PDF
 - AI-based Schedule Generation:
 - Generate Optimized Schedule

2. Teacher:

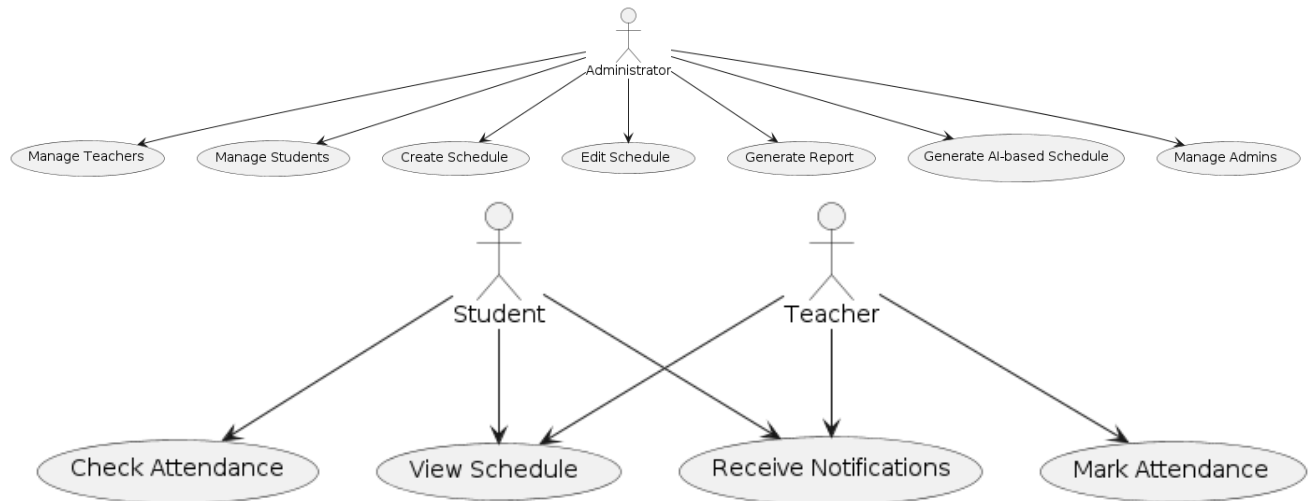
- Views schedules
 - Marks attendance
 - Receives notifications
3. Student:
- Views schedules
 - Checks attendance
 - Receives notifications

Modules

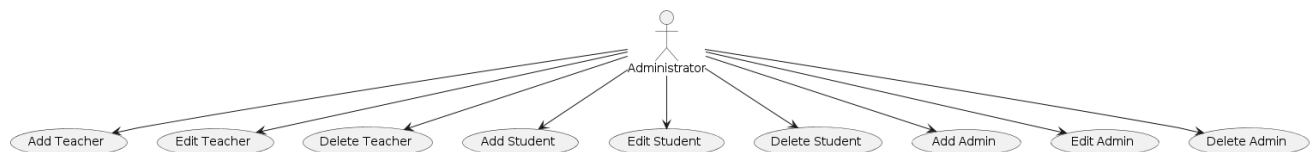
4. User Management:
- Manage Teachers
 - Manage Students
 - Manage Admins
5. Schedule Management:
- Create Schedule
 - Edit Schedule
 - View Schedule
6. Attendance Management:
- Mark Teacher Attendance
 - Mark Student Attendance
7. Notification System:
- Send Notifications
 - Receive Notifications
8. Report Generation:
- Export to Excel
 - Export to PDF
9. AI-based Schedule Generation:
- Generate Optimized Schedule

Use Case Diagrams

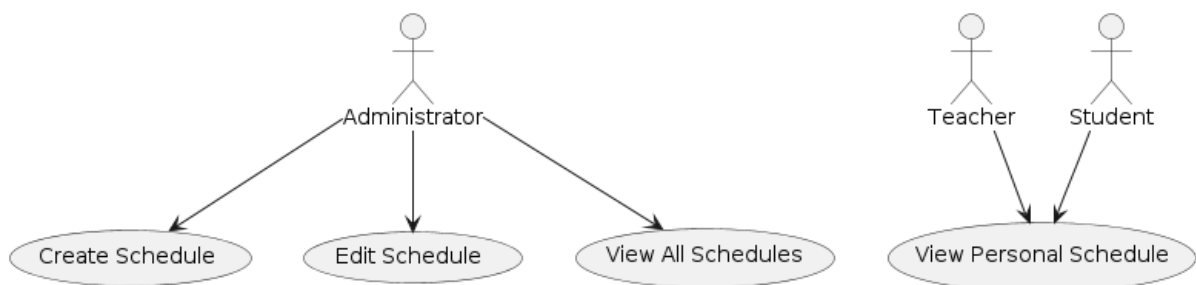
Top-Level Use Case Diagram



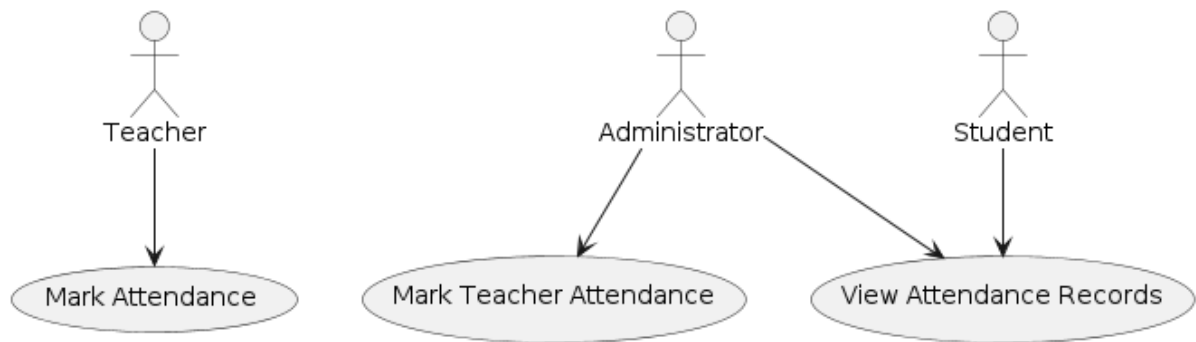
User Management Module



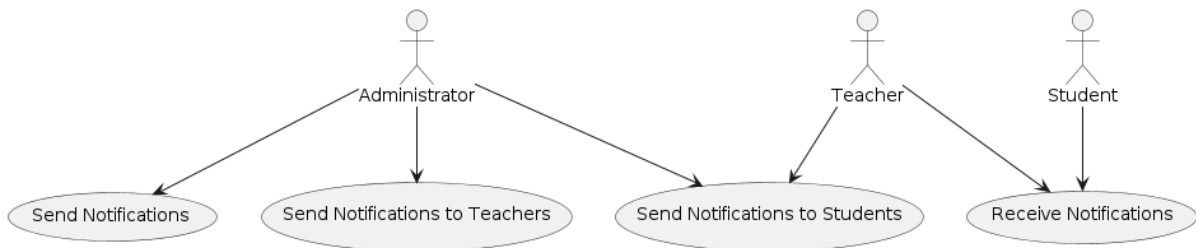
Schedule Management Module



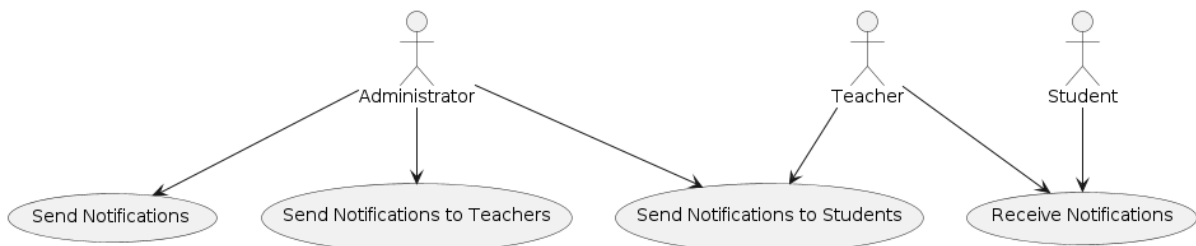
Attendance Management Module



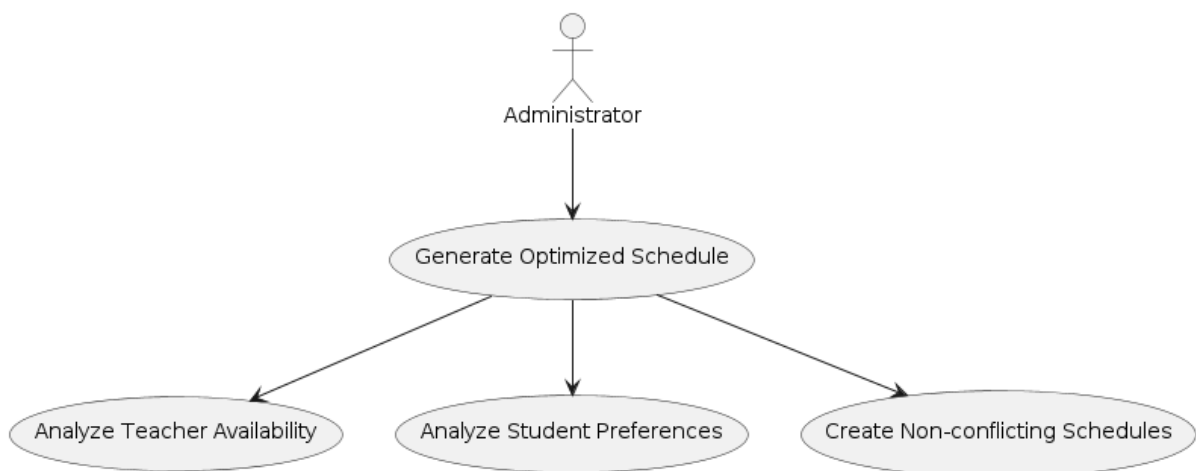
Notification System Module



Report Generation Module



AI-based Schedule Generation Module

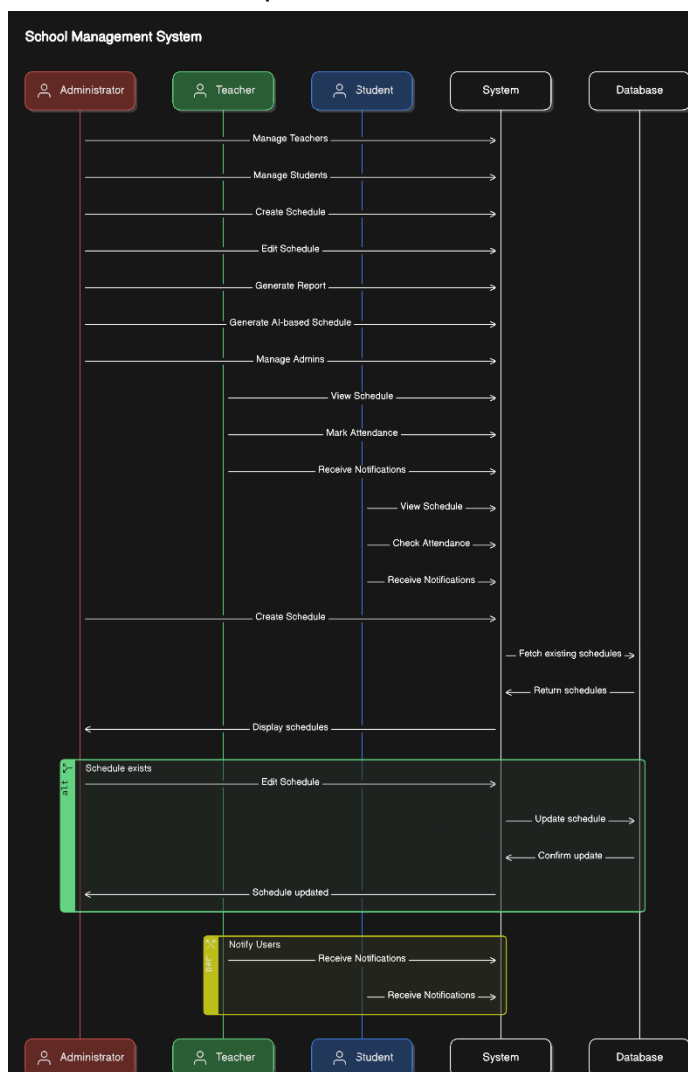


Sequence Diagrams

UML sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifacts for dynamic modelling, which focuses on identifying the behaviour within your system.

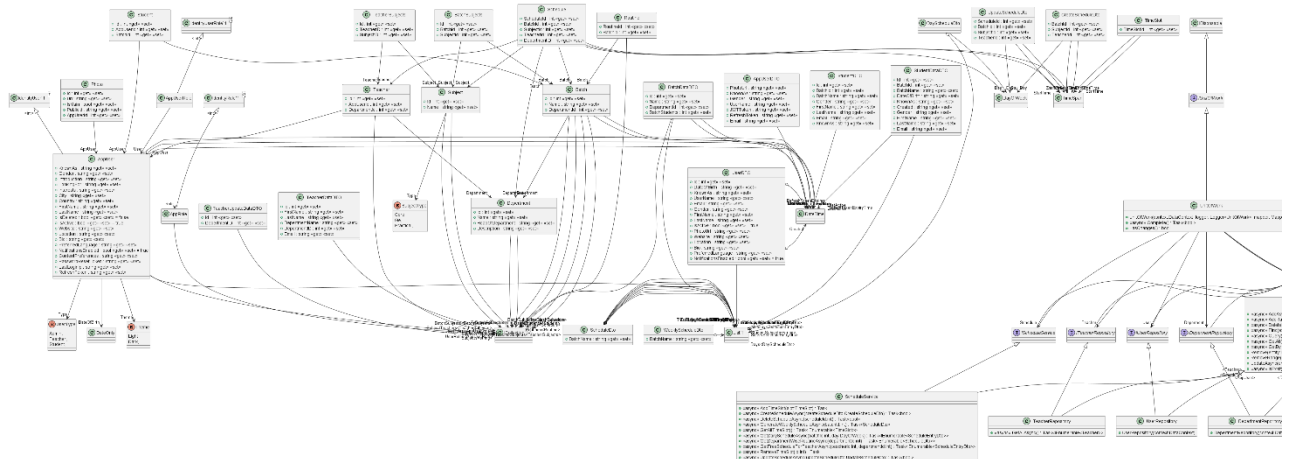
An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them. A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. Graphically, a sequence diagram is a table that shows objects arranged along x-axis and messages, ordered in increasing time, along the y-axis. Contents Sequence diagrams commonly contain the following: Objects Links Messages Like all other diagrams, sequence diagrams may contain notes and constrains.

Administration Sequence



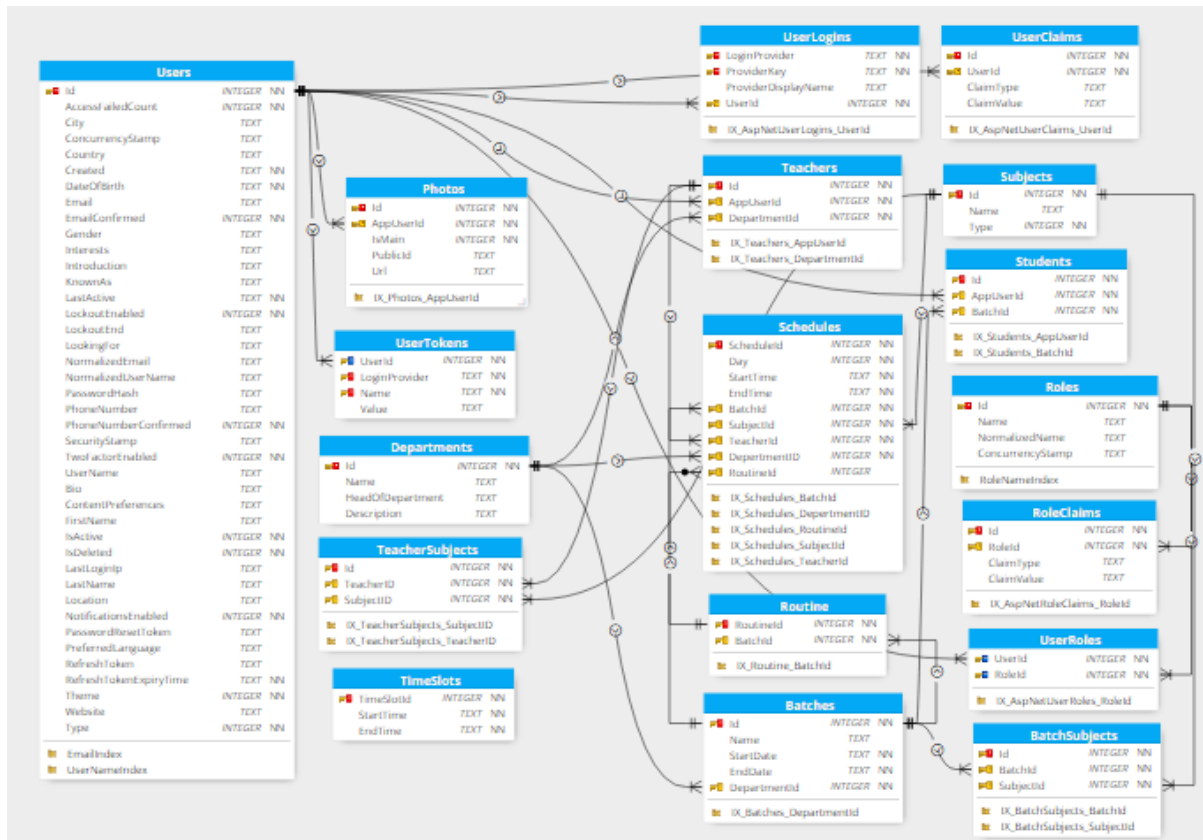
Class Diagrams

A class diagram describes the static structure of the symbols in your new system. It is a graphic presentation of the static view that shows a collection of declaratives (static) model elements, such as classes, types, and their contents and relationships. Classes are arranged in hierarchies sharing common structure and behaviour, and are associated with other classes.



ER Diagram

This depicts relationship between data objects. The attribute of each data objects noted in the entity- relationship diagram can be described using a data object description. Data flow diagram serves two purposes: 1. To provide an indication of how data are transformed as they move through the system. 2. To depict the functions that transformation the data flow.



Problem Statement

Creating and managing schedules for educational institutions is a complex and time-consuming task. Each academic department must ensure that teachers and students are available, and that there are no scheduling conflicts. Additionally, handling teacher absences, managing student attendance, and grading students based on attendance data are cumbersome processes when done manually. The need for an efficient, automated system to handle these tasks is evident.

Advantages

Time Efficiency

- Automates the creation and management of schedules, significantly reducing the time required for these tasks.

Conflict Resolution

- Ensures no scheduling conflicts by checking teacher and student availability.

Real-time Updates

- Provides real-time schedule updates and notifications for teachers and students.

Attendance Management

- Streamlines teacher and student attendance tracking, making the process more efficient and accurate.

Grading Automation

- Utilizes attendance data to automate student grading, reducing administrative workload.

AI-based Optimization

- Uses AI to generate optimized schedules, ensuring efficient use of resources and minimizing idle time.

Comprehensive Reporting

- Generates detailed reports in various formats (Excel, PDF), aiding in administrative decision-making.

User-friendly Interface

- Offers an intuitive interface for administrators, teachers, and students, making the system easy to use.

Disadvantages

Initial Setup Cost

- The initial setup and implementation of the system may require a significant investment in terms of time and resources.

Training Requirements

- Users may need training to effectively use the system, which could temporarily divert resources.

Dependence on Technology

- The system's reliance on technology means that any technical issues (e.g., server downtime) could disrupt operations.

Data Security

- Ensuring the security and privacy of the stored data is crucial, and any breaches could have serious implications.

Maintenance

- Regular maintenance and updates are necessary to keep the system running smoothly, which could incur ongoing costs.

Customization Limitations

- The system might not perfectly fit the unique needs of every institution without further customization.

Feature Scope

Current Features

1. Schedule Creation and Management
 - Schedule Creation: Administrators can create schedules for individual classes, batches, and departments. This includes setting the class timings, assigning teachers, and allocating classrooms.
 - Schedule Editing: Administrators can easily modify existing schedules, such as changing class timings, reassigning teachers, or updating classroom allocations.
 - Schedule Export: Administrators can export the created schedules to Excel or PDF formats, allowing for easy distribution and record-keeping.
2. User Management
 - Teacher Management: Administrators can add, edit, and delete teacher records, including their personal information, contact details, and subject specializations.
 - Student Management: Administrators can manage student records, including adding new students, updating their information, and removing students from the system.

- Admin Management: The system allows administrators to create, edit, and remove other administrative users, ensuring proper access control and segregation of duties.
3. Real-time Schedule Viewing
- Teacher Schedule View: Teachers can access their class schedules in real-time, allowing them to stay informed about their upcoming classes, assigned subjects, and classroom locations.
 - Student Schedule View: Students can view their class schedules in real-time, ensuring they are always aware of their class timings, subjects, and teachers.
 - Schedule Updates: Any changes made to the schedules by administrators are immediately reflected in the teacher and student views, keeping them up-to-date with the latest information.

Planned Features

4. Attendance Management System
- Teacher Attendance Tracking: Implement a module where teachers can mark their daily attendance, either through a web-based form or a mobile application. If a teacher is absent, the system will automatically notify the administrator.
 - Student Attendance Tracking: Provide teachers with the ability to mark student attendance for each class. The attendance data will be stored in the system and used for grading and performance evaluation purposes.
 - How to Achieve: Develop an attendance module that seamlessly integrates with the existing schedule management system. Use interactive forms and checkboxes to record attendance, and store the data in the database. Implement notification triggers to alert administrators about teacher absences.
5. Grading System
- Automated Grading Based on Attendance: Implement algorithms that can automatically calculate student grades based on their attendance records. The system will apply predefined grading rules to determine the final grades for each student.
 - How to Achieve: Integrate the attendance data with the student management system. Develop algorithms that process the attendance information and apply the institution's grading policies to generate

student grades. Ensure the grading system is flexible and can be customized to accommodate different grading schemes.

6. Class Notes Portal

- Notes Sharing Platform: Provide a centralized platform where teachers can upload and share class notes, presentations, and other educational resources. Students can access these materials through their individual portals.
- How to Achieve: Create a file management system that allows teachers to attach relevant files to specific classes or topics. Implement secure storage and access control mechanisms to ensure that students can only access the materials for their enrolled classes.

7. AI-based Schedule Generation

- Optimized Schedule Creation: Leverage artificial intelligence and machine learning techniques to generate optimized schedules that consider various factors, such as teacher availability, class preferences, and resource utilization.
- How to Achieve:
 - Data Collection: Gather comprehensive data on teacher schedules, class sizes, room capacities, and other relevant constraints.
 - AI Algorithm: Develop or integrate advanced AI algorithms, such as constraint satisfaction problems (CSP) or genetic algorithms, that can process the collected data and generate schedules that maximize resource utilization and minimize conflicts.
 - Testing and Refinement: Continuously test the AI-generated schedules, gather feedback from administrators, and refine the algorithms to ensure the schedules meet the institution's requirements and preferences.

Implementation Plan

The planned features will be implemented in a phased approach to ensure a smooth and controlled rollout of the system:

1. Phase 1: Attendance and Grading System

- Develop and integrate the attendance management module, allowing teachers to mark attendance for both themselves and their students.
- Implement the grading system that automatically calculates student grades based on the attendance data.

- Thoroughly test the attendance and grading features, ensuring they are functioning as expected.
 - Deploy the attendance and grading system to the production environment.
2. Phase 2: Class Notes Portal
- Design and develop the class notes sharing platform, including the file upload, storage, and access control mechanisms.
 - Integrate the notes portal with the teacher and student portals, allowing seamless access to the shared resources.
 - Conduct extensive testing to ensure the security and reliability of the file management system.
 - Deploy the class notes portal and provide training to teachers and students on its usage.
3. Phase 3: AI-based Schedule Generation
- Gather and prepare the necessary data for the AI-based scheduling system, including teacher availability, class sizes, and resource constraints.
 - Research and implement the chosen AI algorithms (e.g., CSP, genetic algorithms) to generate optimized schedules.
 - Develop the AI-based scheduling module and integrate it with the existing schedule management system.
 - Thoroughly test the AI-generated schedules, gather feedback, and refine the algorithms as needed.
 - Deploy the AI-based scheduling feature and provide training to administrators on its usage and configuration.

Screenshots

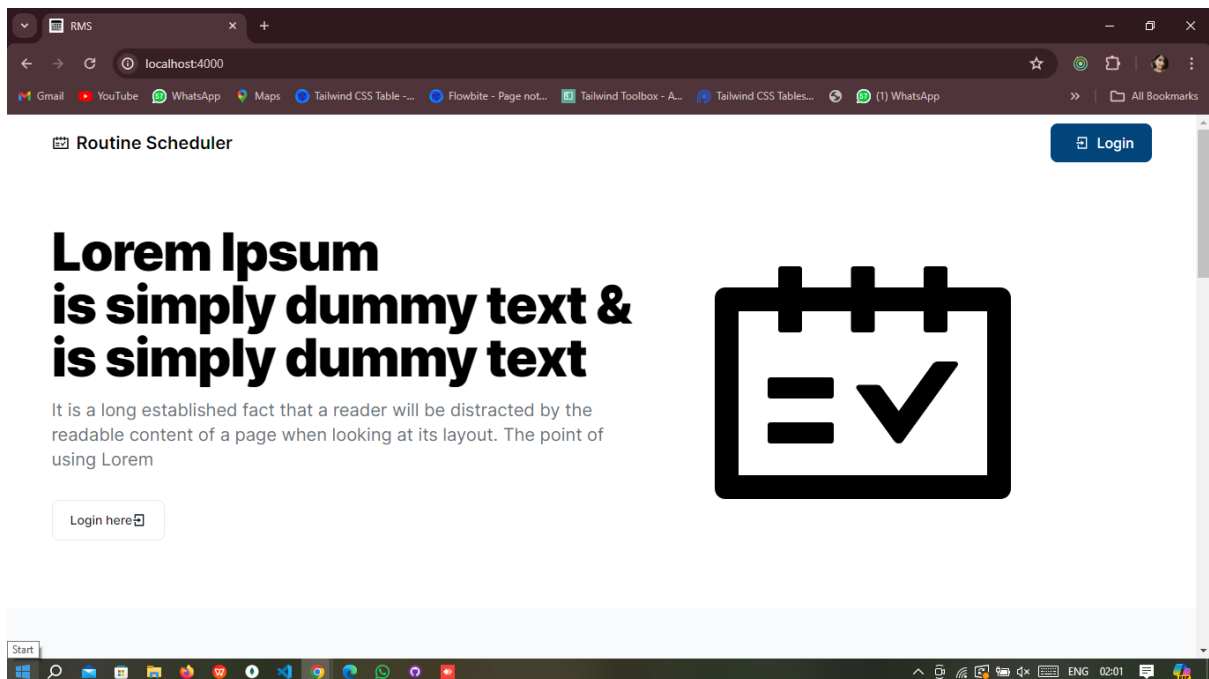


Fig: 1.2 Landing page

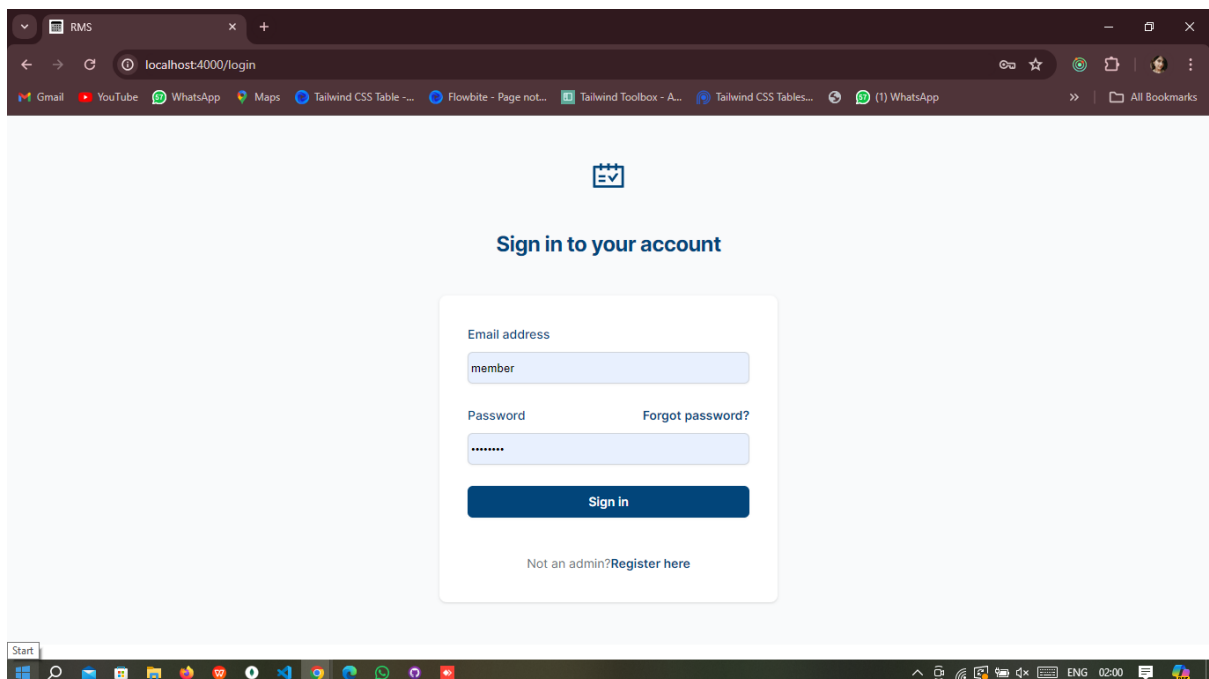


Fig: 1.2 Login

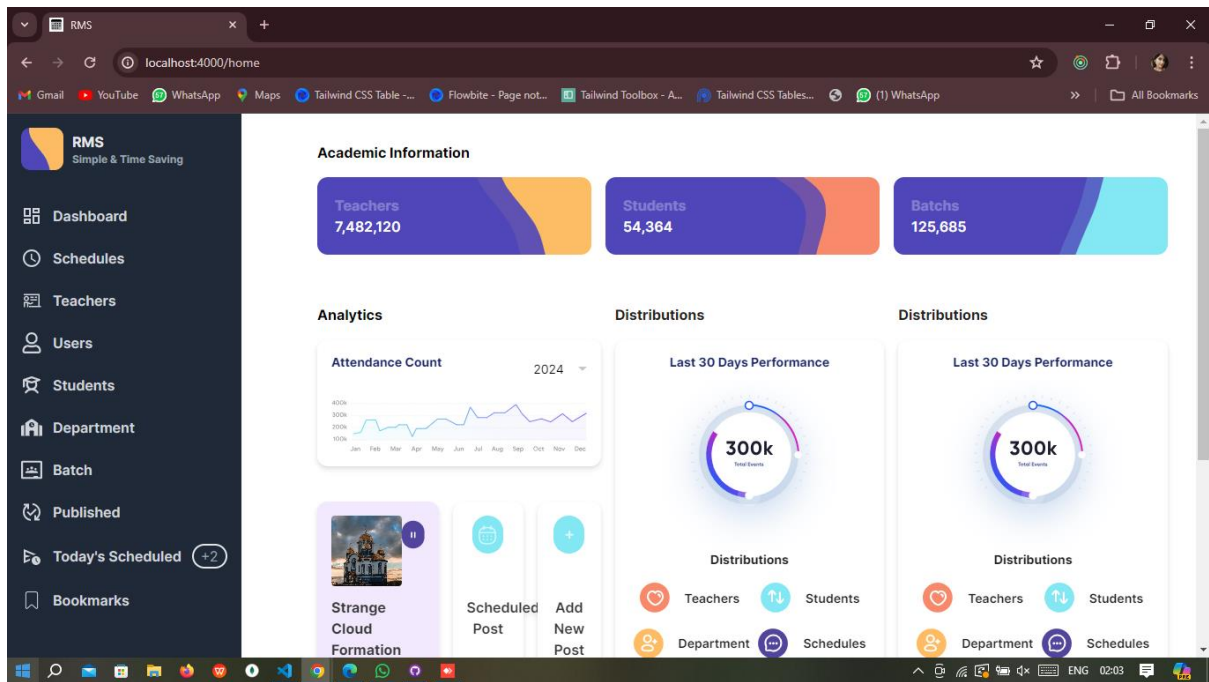


Fig: 1.3 Admin Home Page

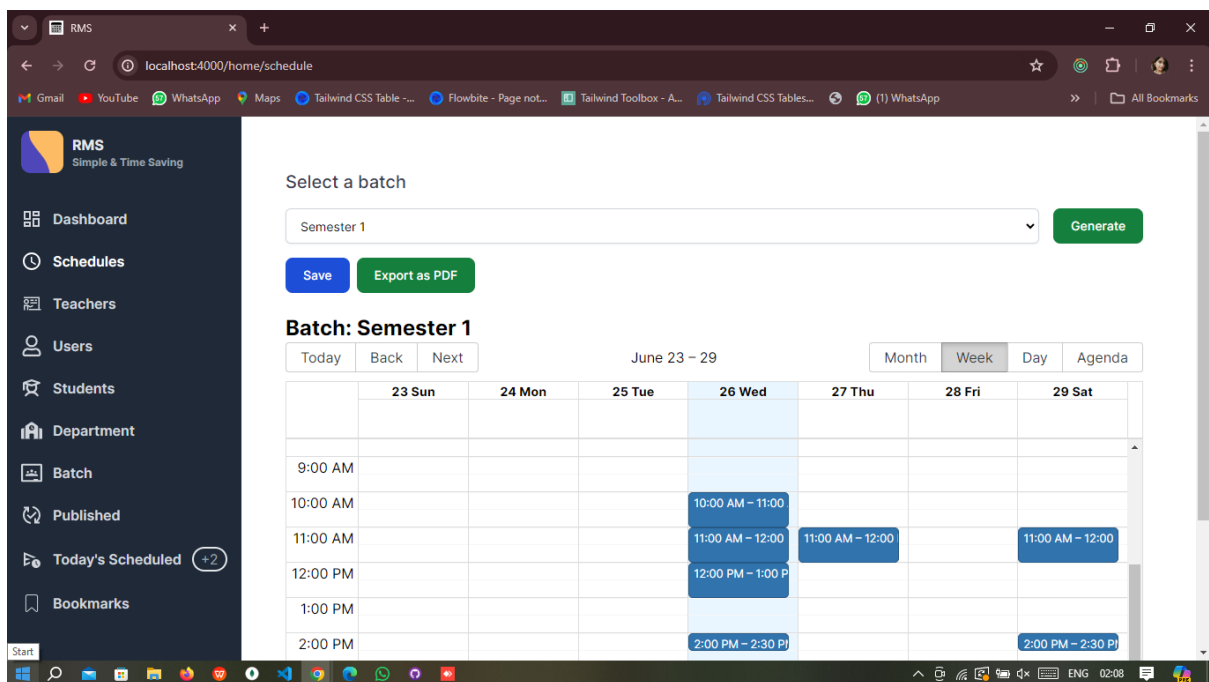


Fig: 1.4 Admin Schedule

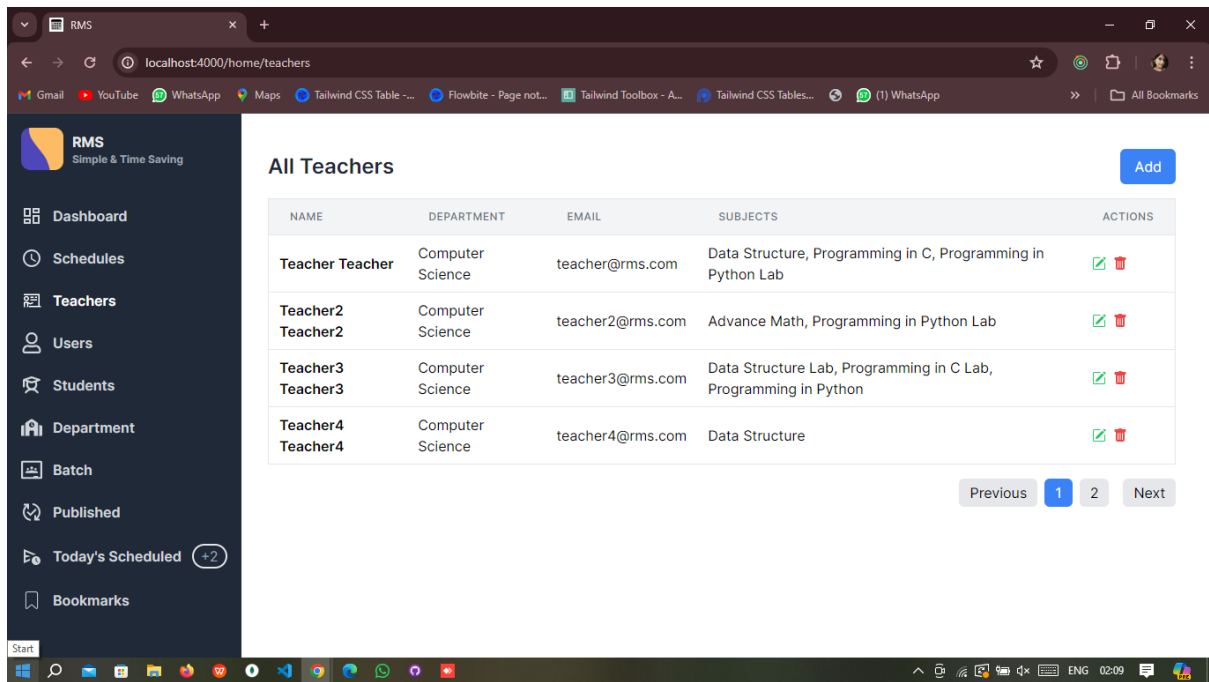


Fig: 1.5Admin Teacher GetAll

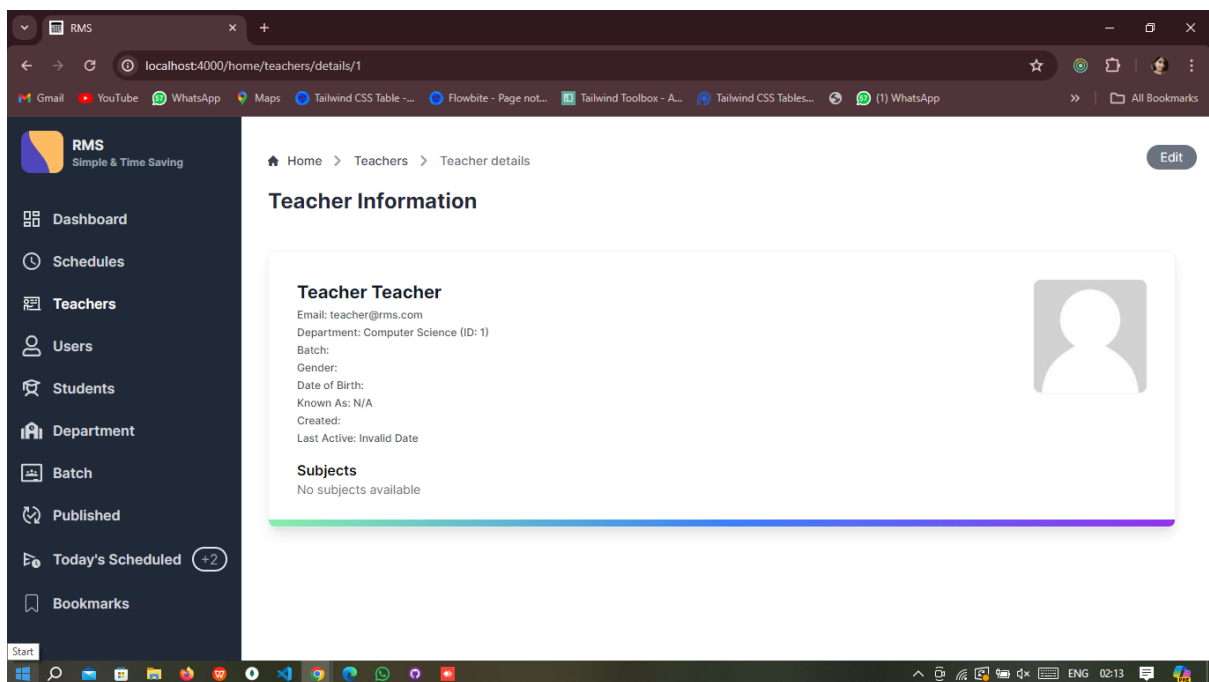


Fig: 1.6Admin Teacher Details

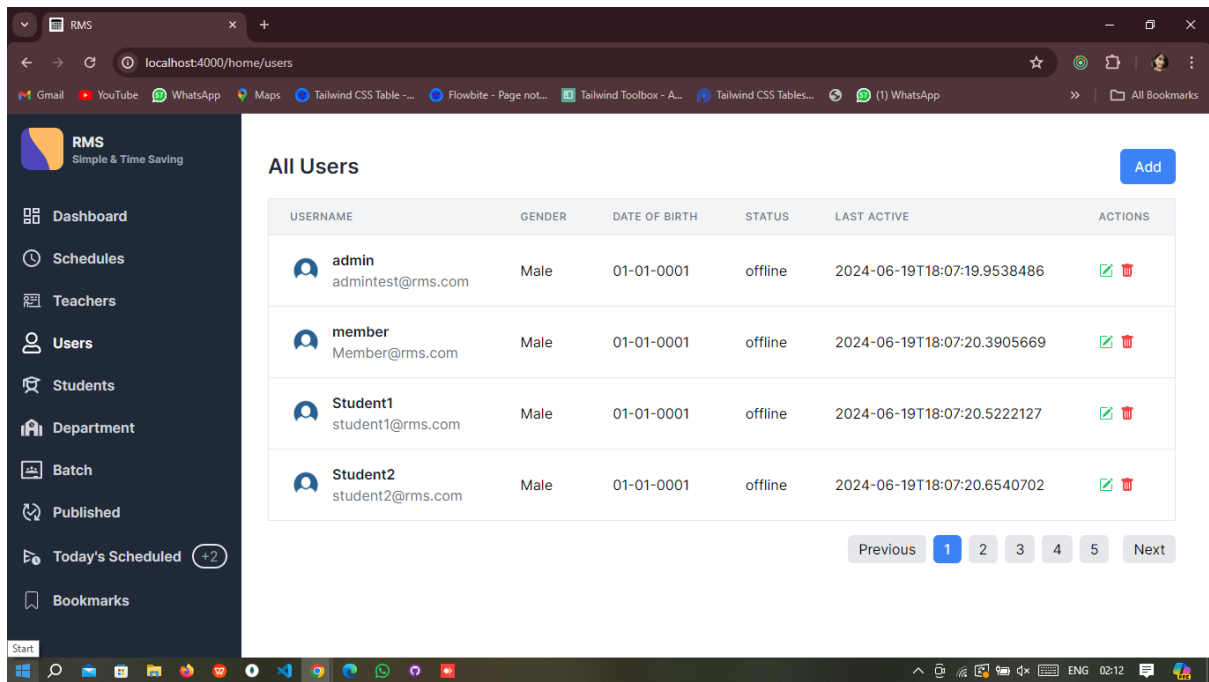


Fig: 1.7 Admin User GetAll

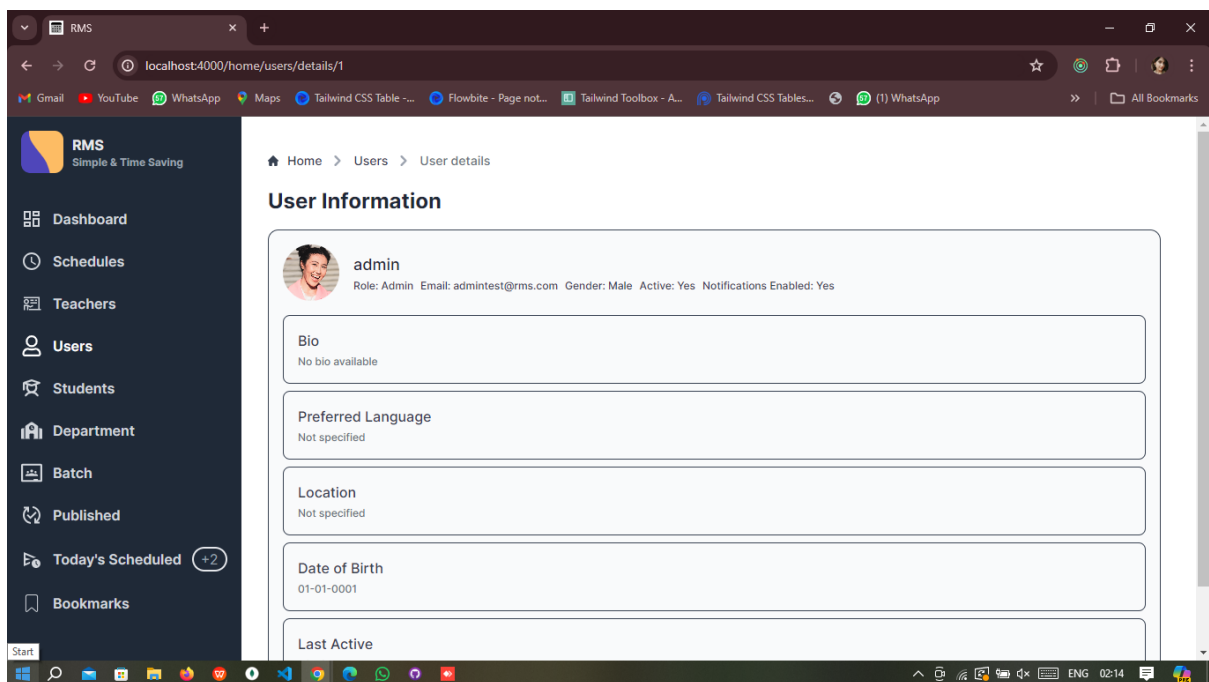


Fig: 1.8 Admin User Details

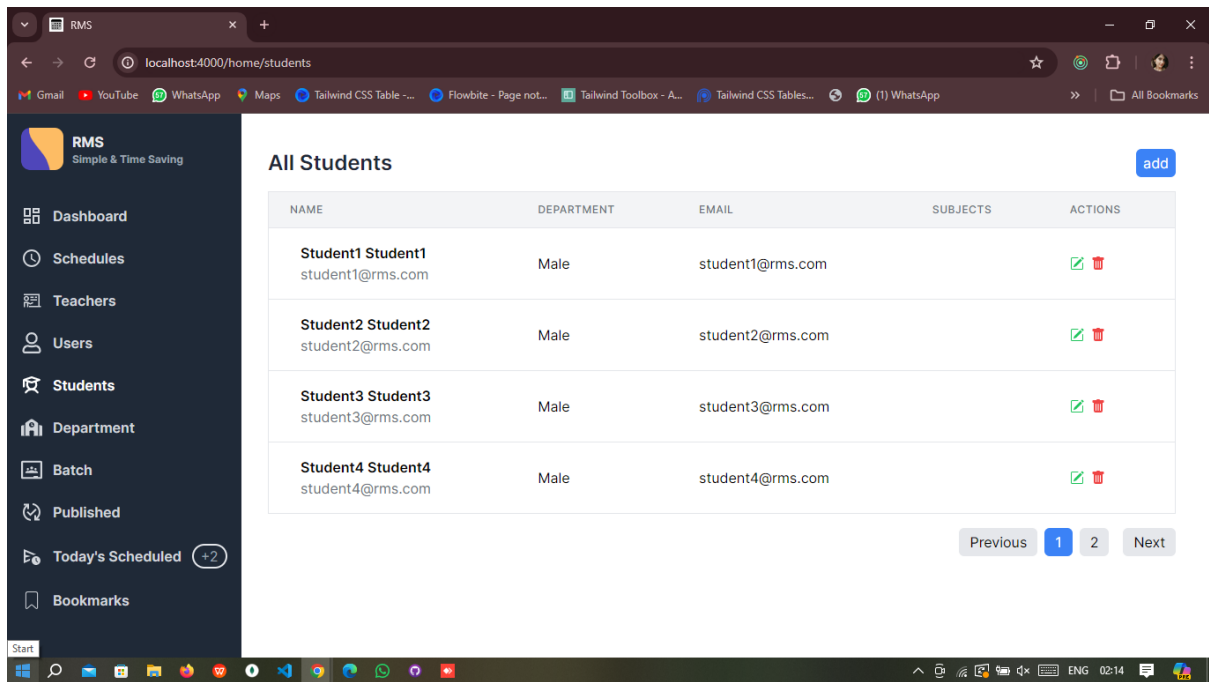


Fig: 1.9 Admin Student GetAll

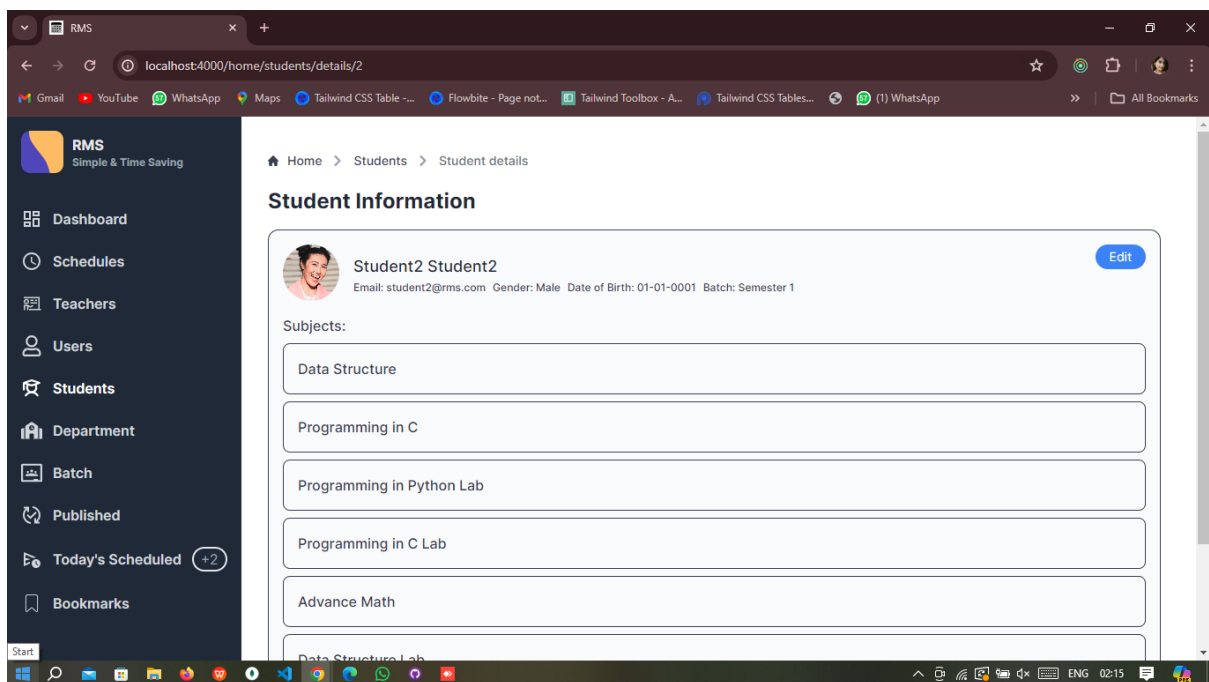


Fig: 1.10 Admin Student Details

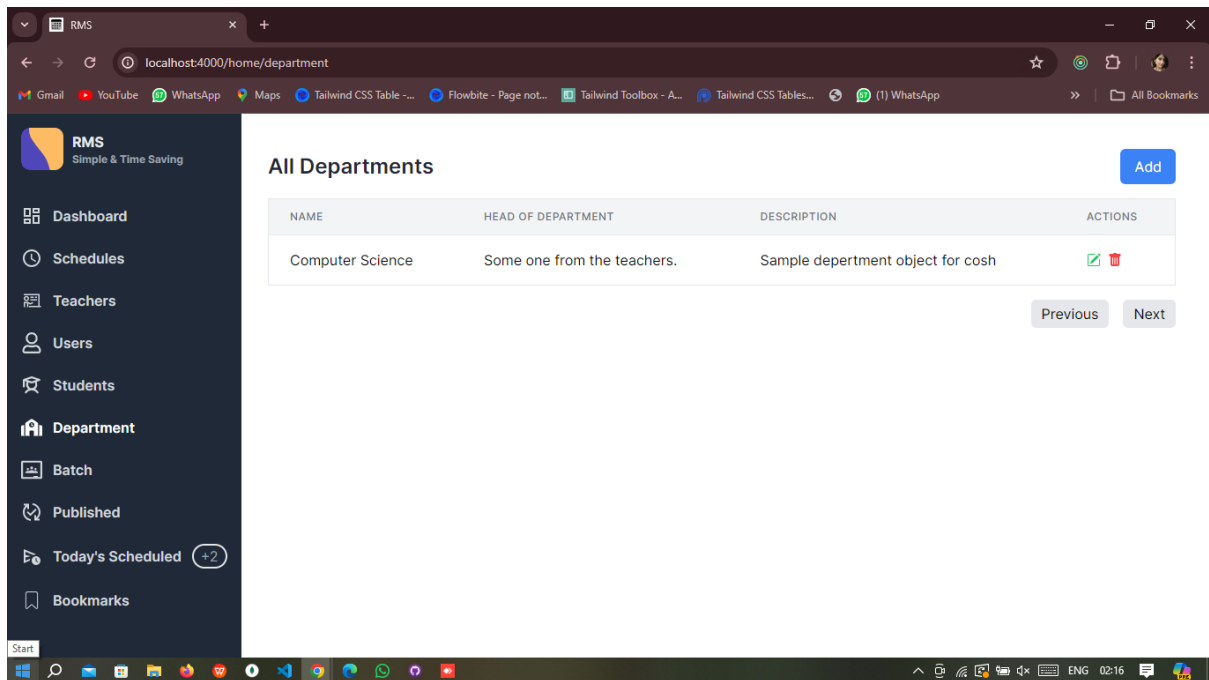


Fig: 1.11 Admin Department Getall

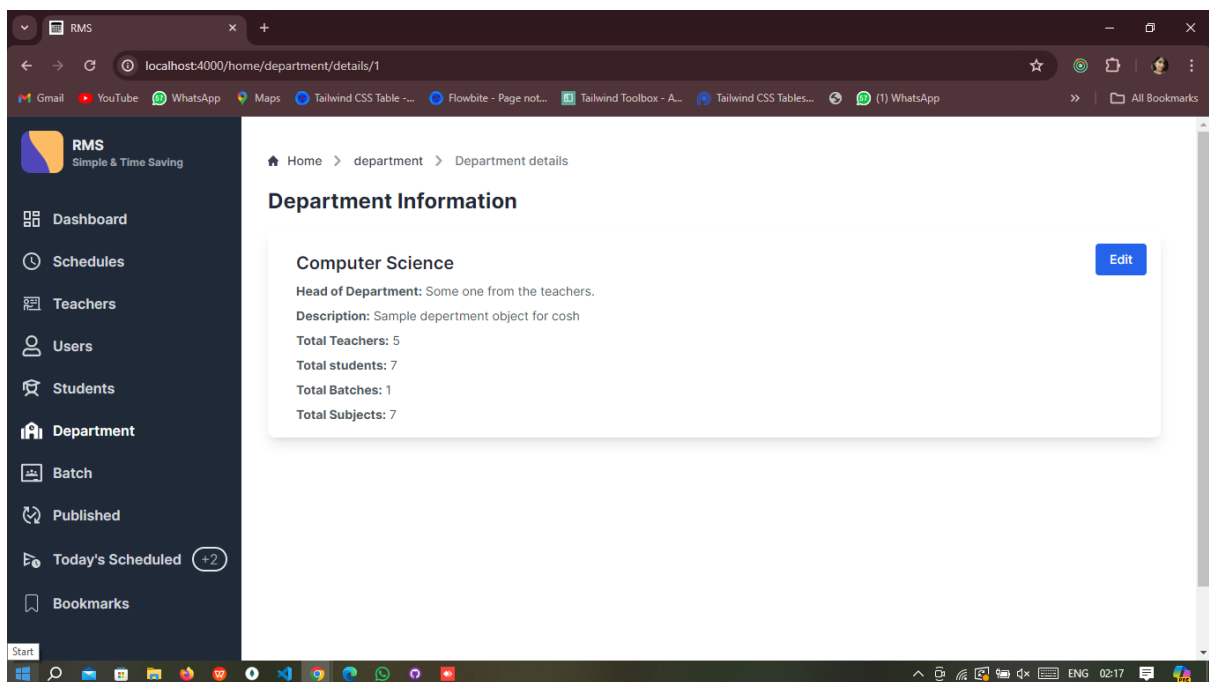


Fig: 1.12 Admin Department Details

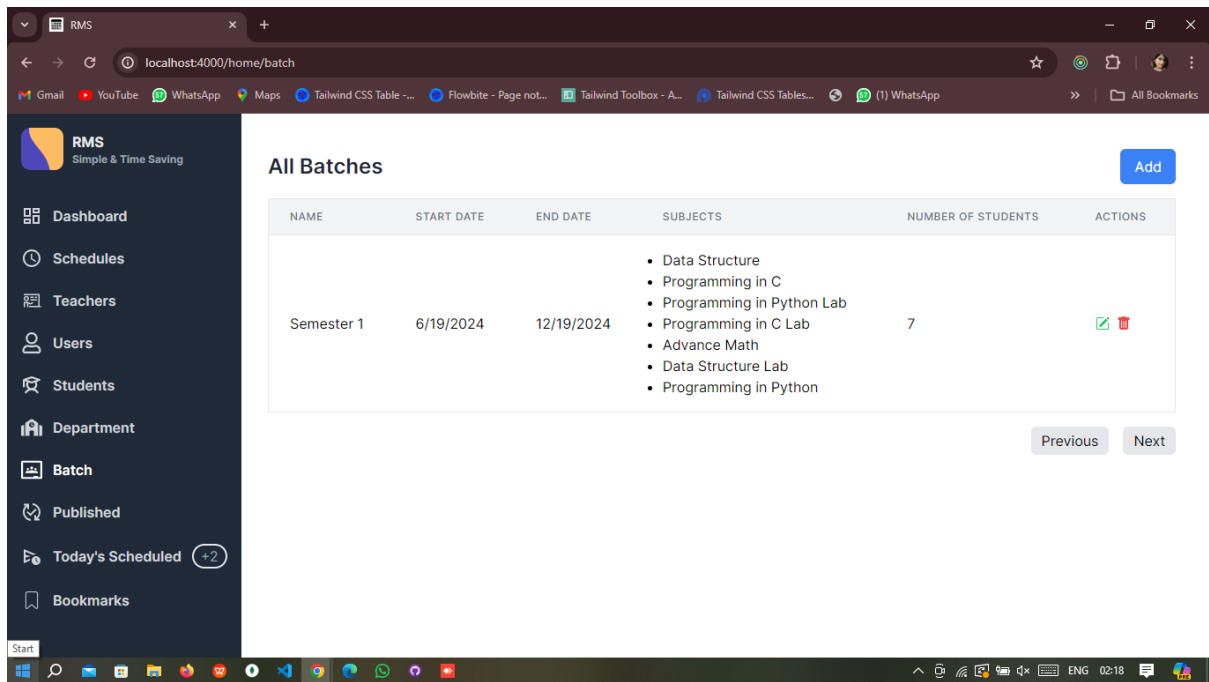


Fig: 1.13 Admin Batch getall

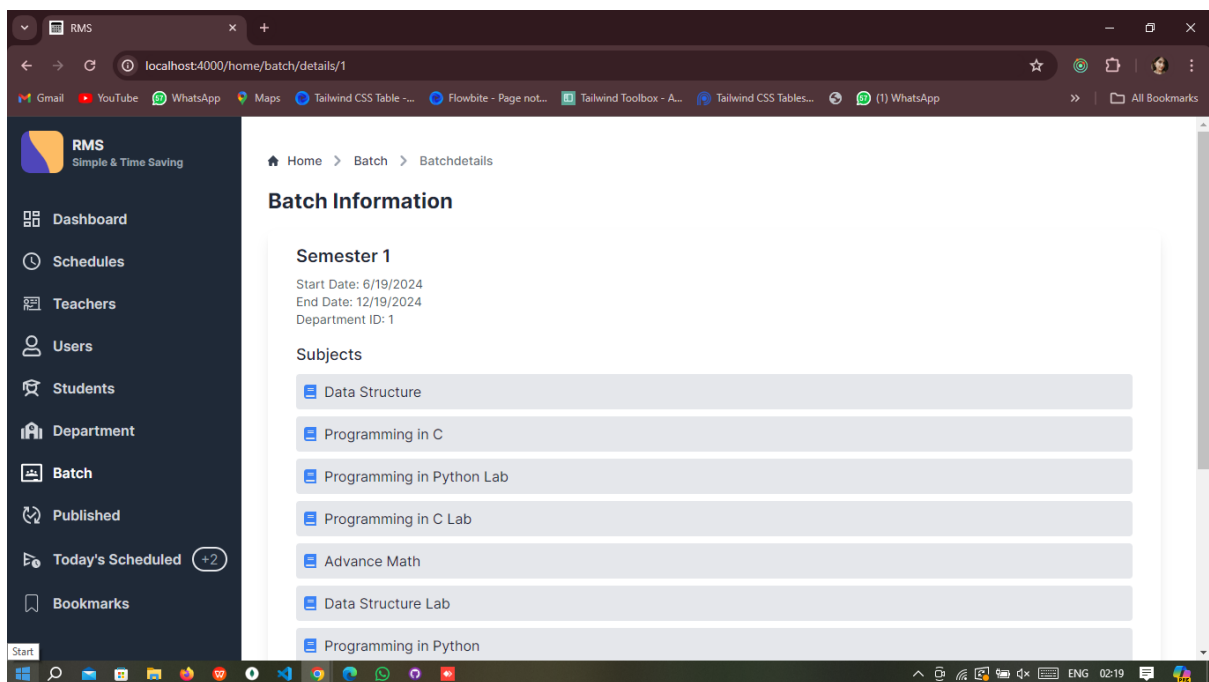


Fig: 1.14 Admin Batch Details

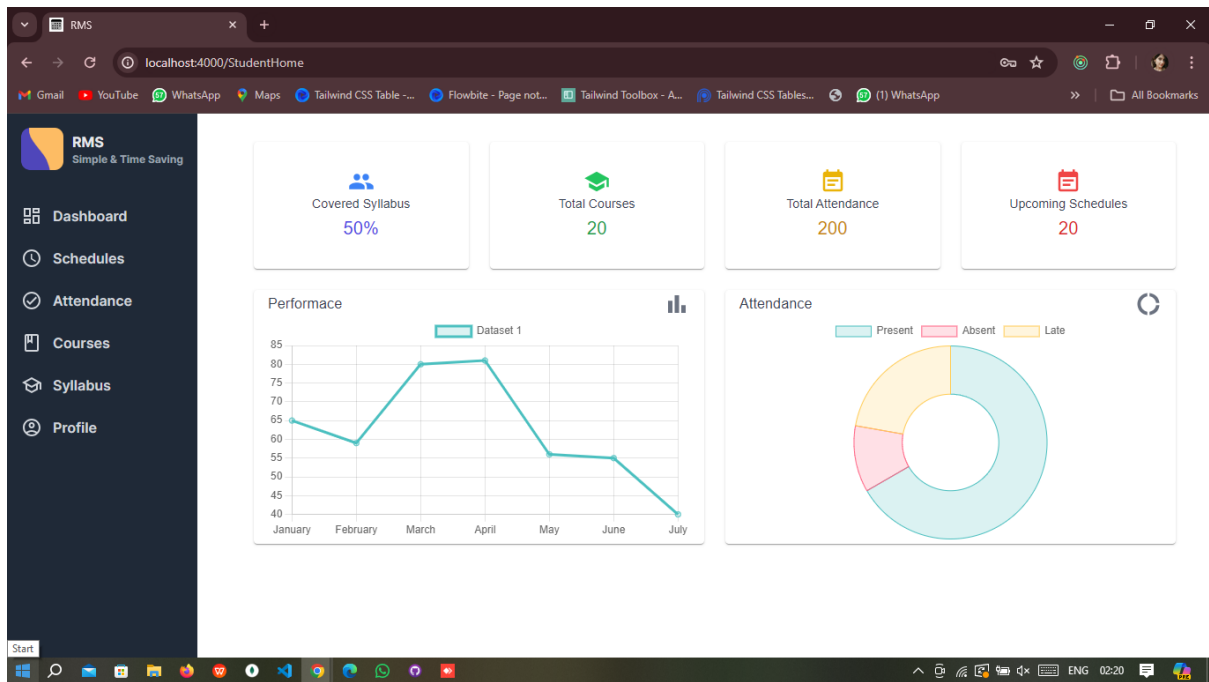


Fig: 2.1 student dahboard

The screenshot shows the 'Student Daily Schedule' page. It includes a search bar and a table with columns for Subject Name, Teacher Name, Start Time, and End Time. The table lists six different practical and core sessions.

SUBJECT NAME	TEACHER NAME	START TIME	END TIME
Programming in Python Lab Practical	Teacher Teacher	10:00	11:00
Programming in C Lab Practical	Teacher3 Teacher3	12:00	13:00
Programming in C Lab Practical	Teacher3 Teacher3	14:00	14:30
Programming in Python Lab Practical	Teacher2 Teacher2	10:00	11:00
Programming in C Core	Teacher Teacher	12:00	13:00
Programming in Python Lab Practical	Teacher2 Teacher2	11:00	12:00

Fig: 2.2 student daily schedule

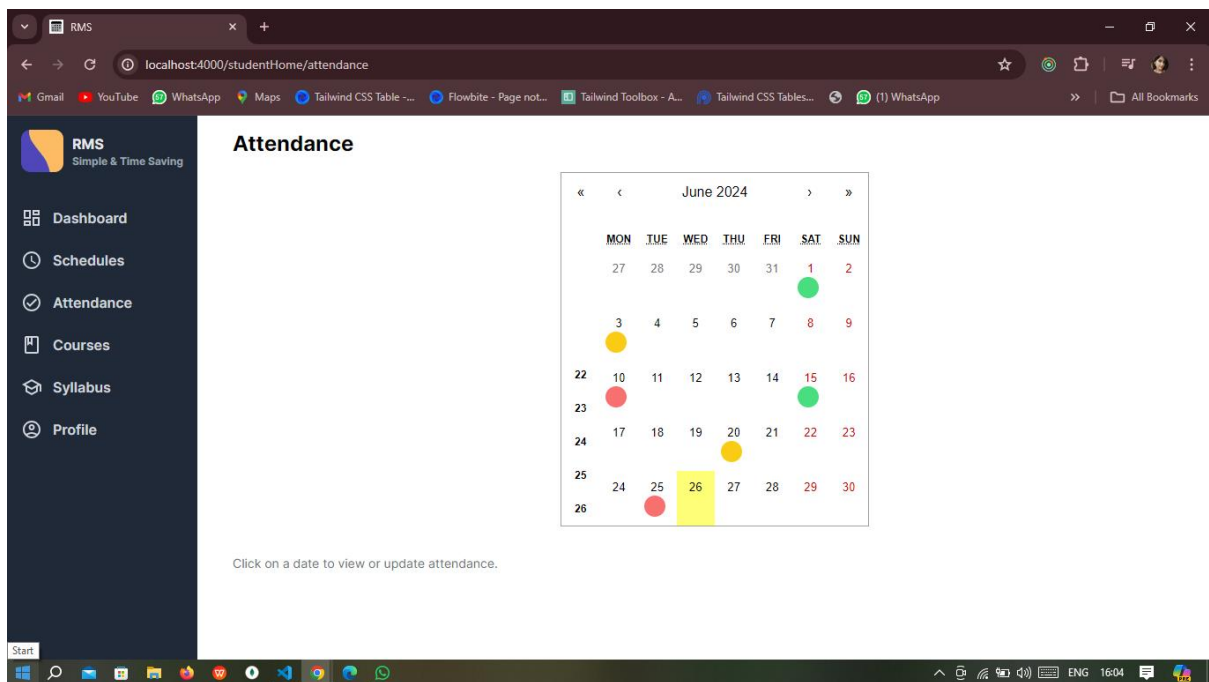


Fig: 2.3 student Attendance

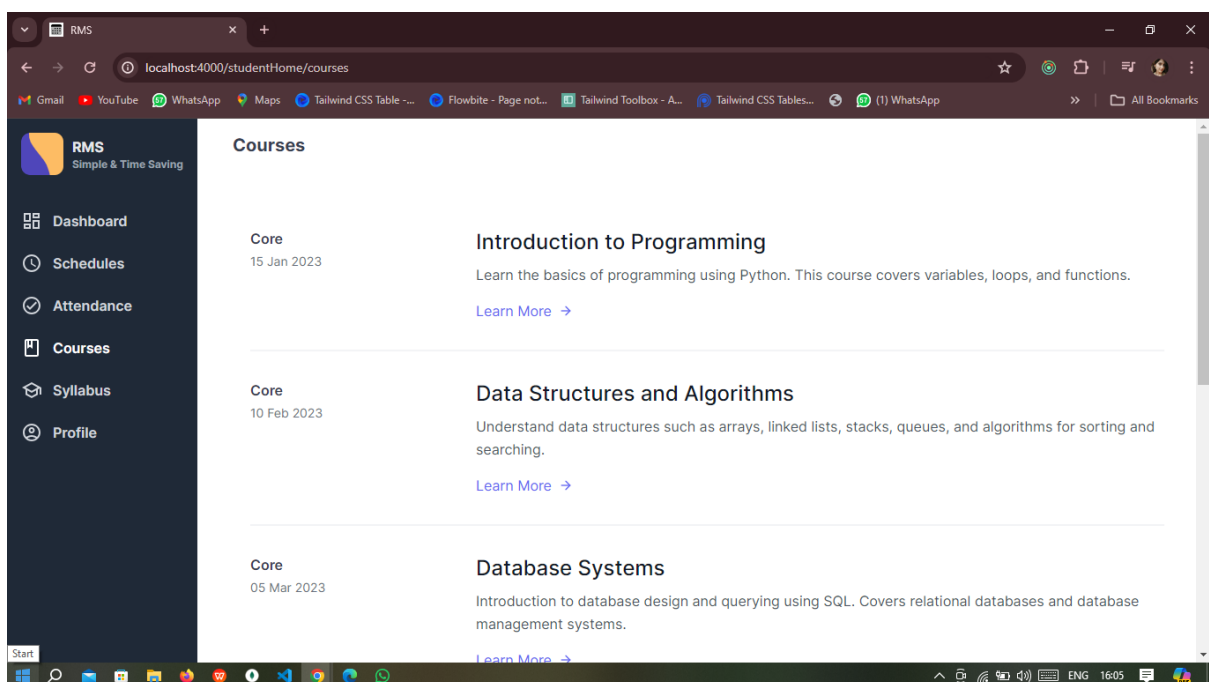


Fig: 2.4 student Courses

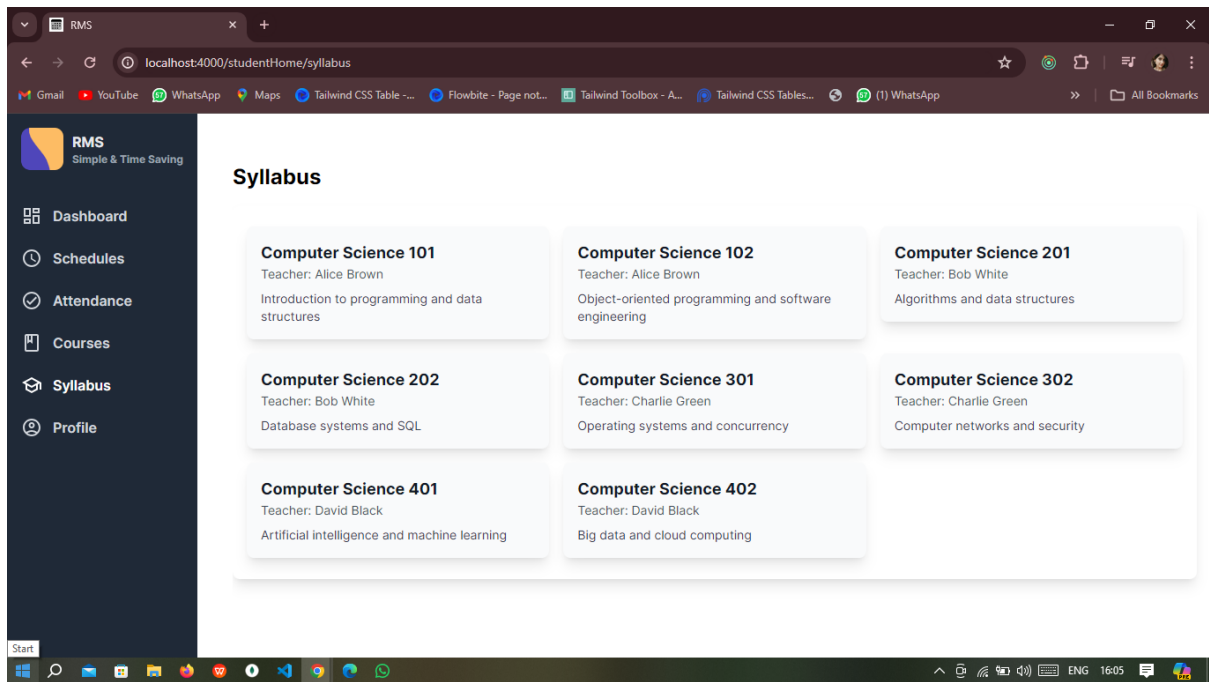


Fig: 2.4 student Syllabus

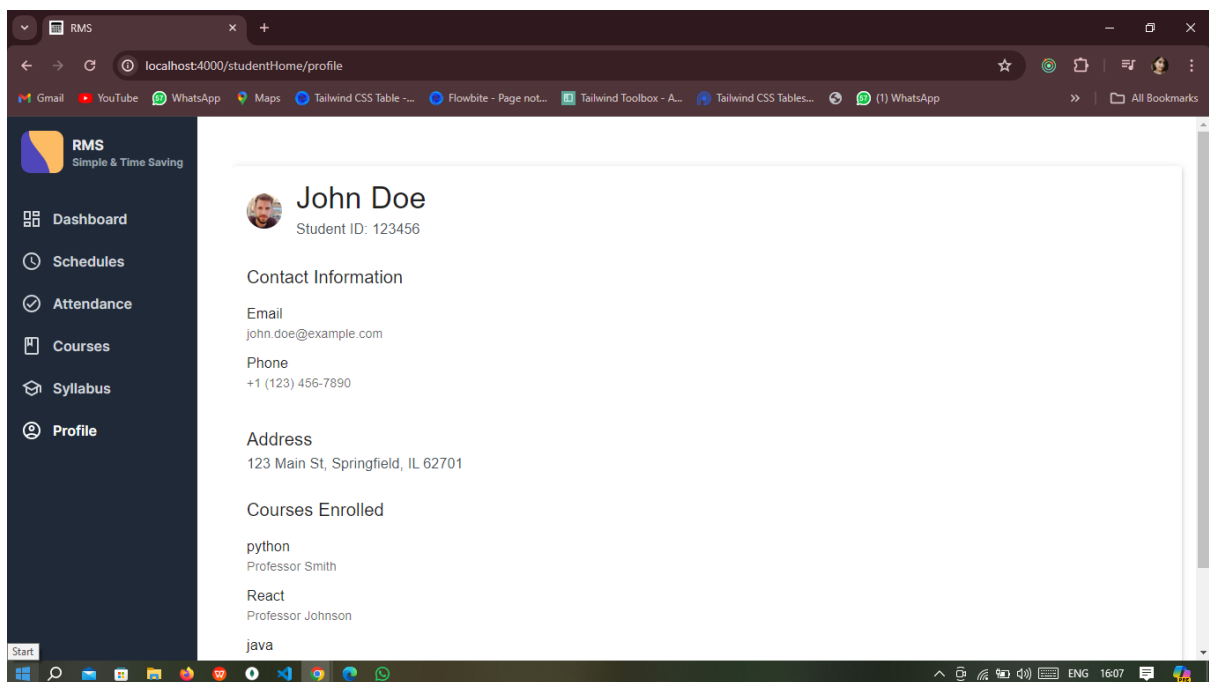


Fig: 2.4 student Profile

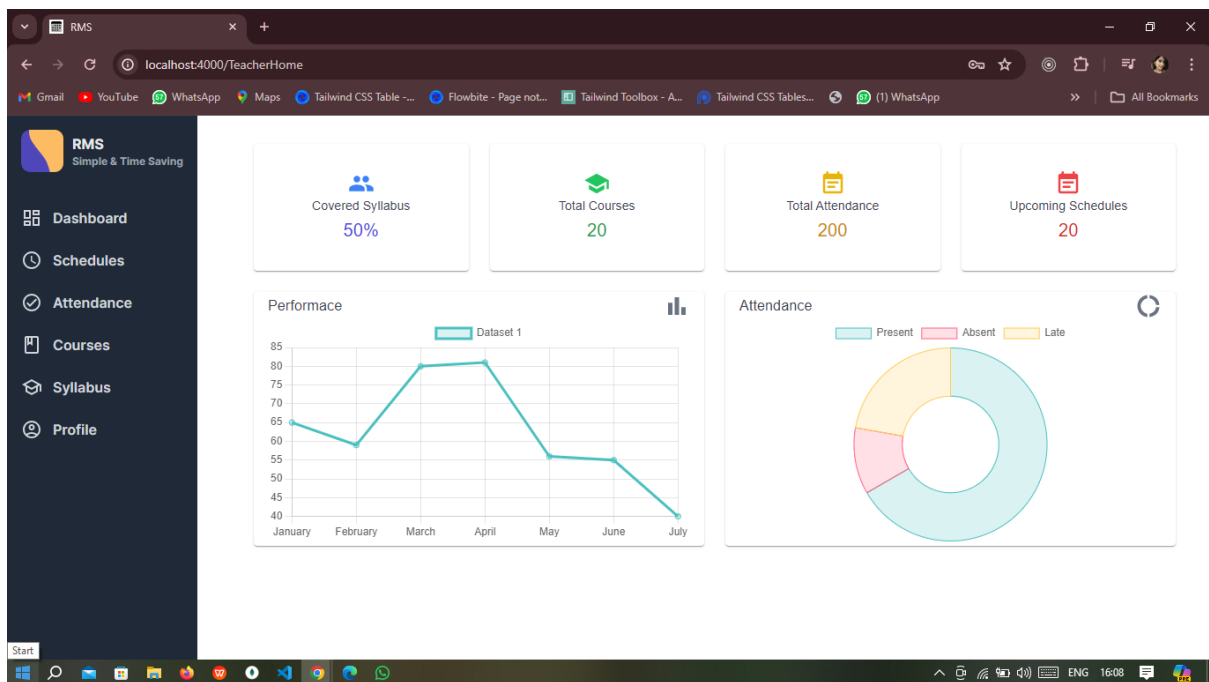


Fig: 2.1 Teacher Dashboard

The Teacher Schedule in the RMS system displays a weekly schedule for a teacher. The schedule is organized into a table with the following columns: Day, Time, and Subject.

Day	Time	Subject
Monday	9:00 AM - 12:00 PM	Computer Science 101
Tuesday	10:00 AM - 1:00 PM	Computer Science 201
Wednesday	8:00 AM - 11:00 AM	Computer Science 301
Thursday	11:00 AM - 2:00 PM	Computer Science 102
Friday	1:00 PM - 4:00 PM	Computer Science 401

Fig: 2.2 Teacher Schedule

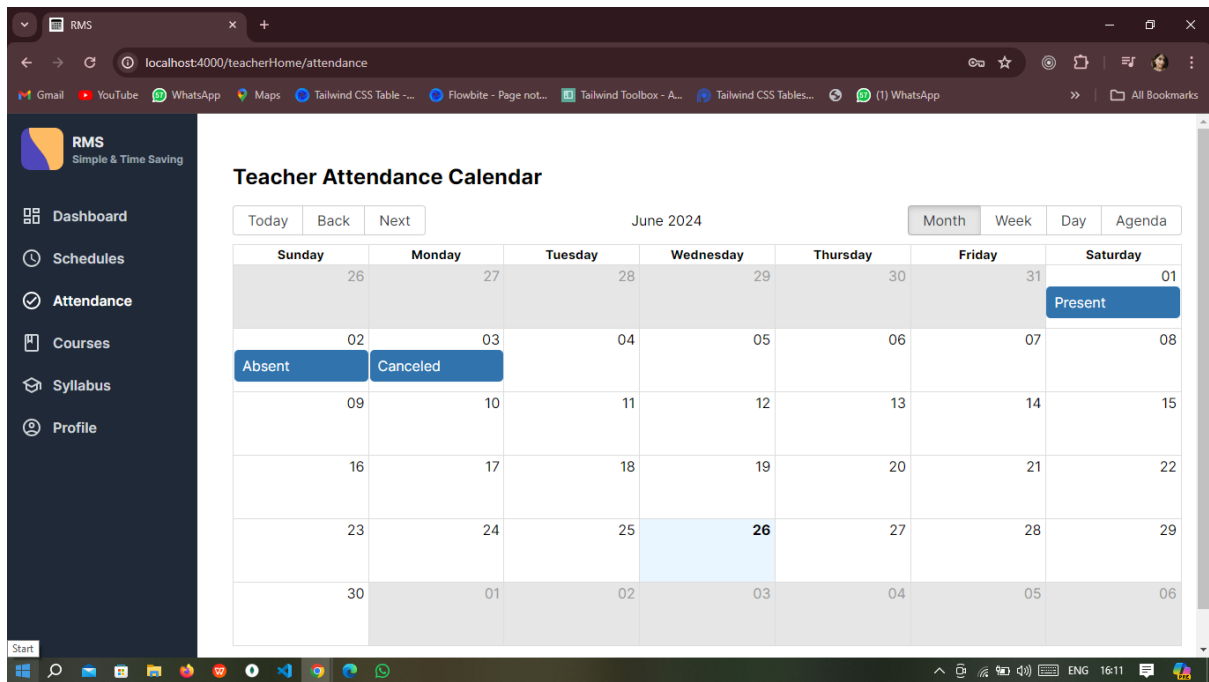


Fig: 2.3 Teacher Attendance

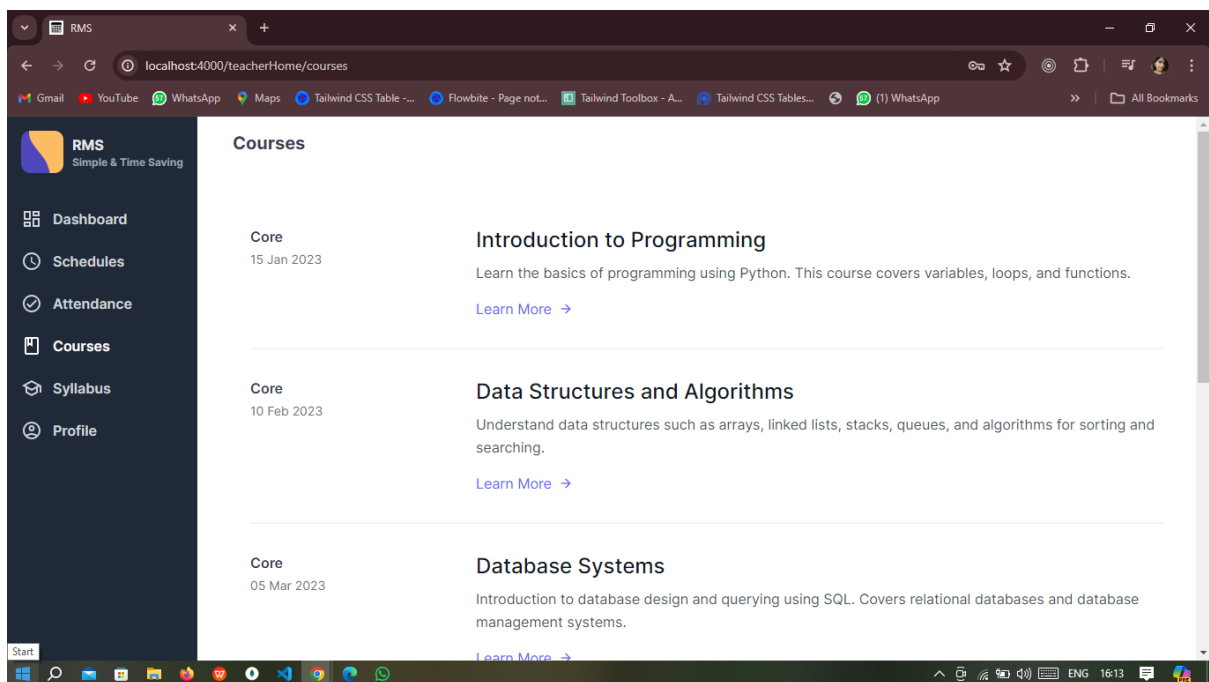


Fig: 2.4 Teacher courses

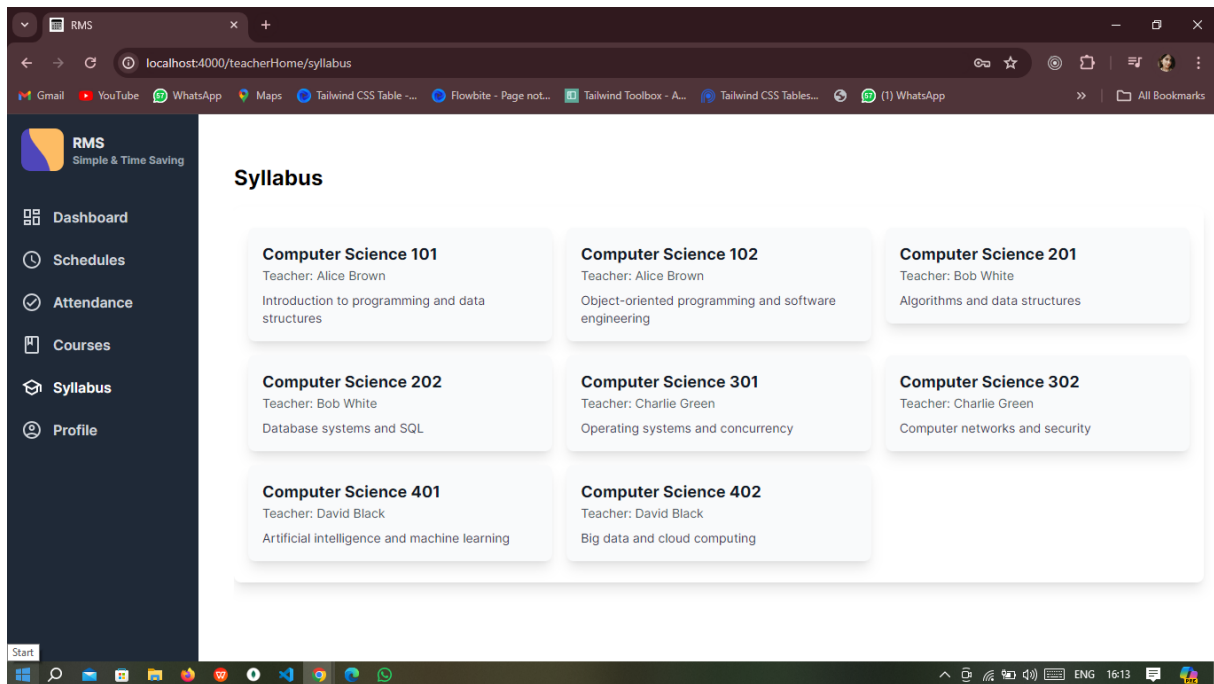


Fig: 2.3 Teacher syllabus

Conclusion

The Routine Management System (RMS) is designed to address the complex challenges associated with managing academic schedules in educational institutions. By automating the scheduling process, managing attendance, and providing real-time updates, RMS significantly reduces the administrative burden on educators and administrators. The system's user-friendly interface ensures that both teachers and students can easily access their schedules and relevant academic information.

With the addition of features like an attendance and grading system, a portal for sharing class notes, and AI-based schedule generation, RMS aims to become an indispensable tool for educational institutions. The AI-based scheduling, in particular, will enhance efficiency by creating optimized schedules that consider teacher availability and student preferences.

The implementation of RMS will lead to a more streamlined, efficient, and transparent process for managing academic schedules, ultimately contributing to the overall improvement in the quality of education. By leveraging modern technologies and methodologies, RMS positions itself as a forward-thinking solution that can adapt to the evolving needs of educational institutions.

References

1. ASP.NET Core Documentation. (2024). Retrieved from <https://docs.microsoft.com/en-us/aspnet/core/>
2. React Documentation. (2024). Retrieved from <https://reactjs.org/docs/getting-started.html>
3. SQL Server Documentation. (2024). Retrieved from <https://docs.microsoft.com/en-us/sql/sql-server/>
4. PlantUML Documentation. (2024). Retrieved from <http://plantuml.com/>
5. "Web-based Academic Management Systems," by Richard Davis, Educational Administration Review, 2022.

These references provide a comprehensive foundation for understanding the technologies, methodologies, and best practices utilized in the development and implementation of the Routine Management System (RMS). They also offer additional insights into related areas.