

ITC 3081 – ICT Project
Department of Information and Communication Technology
Faculty of Technology
University of Sri Jayewardenepura



Reserve – It
Classroom Reservation System
Project Report
Group 24

Project Title

Reserve-It: Classroom Reservation Web-Application

Project Personnel

Mr. Malshan Perera,
Lecturer,
Department of Information and Communication Technology,
Faculty of Technology, University of Sri Jayewardenepura.
dinushamalshan064@gmail.com
+94 78 728 0357

Team Members

Name	Index	Specialized Area	E-mail	Mobile number
Dasanayaka L.K.K.S.	ICT/21/953	Multimedia Technology	ict21953@fot.sjp.ac.lk	+94 710673062
Thilakarathna D.S.D.P.	ICT/21/931	Software Technology	ict21931@fot.sjp.ac.lk	+94 765441217
Walimuni W.D.H.D.	ICT/21/938	Software Technology	ict21938@fot.sjp.ac.lk	+94 774406113
Ekanayake L.P.E.T.S.	ICT/21/837	Software Technology	ict21837@fot.sjp.ac.lk	+94 761136338
Anuththara S.A.	ICT/21/807	Network Technology	ict21807@fot.sjp.ac.lk	+94 716668336

Declaration

I certify that this report does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university, and to the best of my knowledge and belief, it does not contain any material previously developed, published, or written by another person except where due reference is made in the text.

Acknowledgement

We would like to express our sincere gratitude to everyone who supported the development of the Classroom Reservation Management System. Special thanks go to our project supervisor for their valuable guidance, as well as to the university administration, lecturers, and staff who provided insights that shaped this project. We also recognize the dedication of our team members; whose collaboration and hard work were essential to creating a system that will enhance classroom scheduling and operational efficiency at our university.

Table of Contents

1. List of Figures	6
2. Introduction and Background	7
3. Business Analysis	8
3.1 Problem Identification.....	8
3.2 Stakeholder Analysis	8
3.3 Requirements Gathering	8
3.4 Gap Analysis	10
3.5 Solution Outline	10
3.6 Expected Impact	11
4. Software Development Process	12
4.1 Requirement Analysis and Planning	12
4.2 System Design.....	12
4.2.1 Use Case Diagram	13
4.2.2 Flow Chart.....	14
4.2.3 ER Diagram.....	15
4.2.4 Class Diagram	16
4.3 Development (Iterative Sprints).....	16
4.4 Testing and Quality Assurance	17
4.5 Deployment	17
4.6 Maintenance and Iteration	17
5. Software Testing Process	18
5.1 Unit Testing.....	18
5.2 Integration Testing.....	18
5.3 System Testing	18
5.4 User Interface (UI) Testing.....	18
5.5 Performance Testing.....	19
5.6 Security Testing.....	19
6. Software deployment process	20
6.1 Code Repository Integration	21
6.2 Environment Configuration	21
6.3 Initial Deployment and Build	21
6.4 Testing in the Production Environment	21

7. Application Interfaces	22
8. Discussion	27
8.1 Challenges.....	27
8.2 Future Work.....	27

1. List of Figures

Figure 1: Manual reservation with Excel.....	8
Figure 2:Manual reservation with Excel.....	9
Figure 3: Use case diagram.....	13
Figure 4: Flow chart.....	14
Figure 5: Entity Relationship diagram	15
Figure 6: Class diagram	16
Figure 7:Datacenter Network.....	20
Figure 8: Home page.....	22
Figure 9: Login page.....	22
Figure 10: User Profile.....	23
Figure 11: Admin dashboard.....	23
Figure 12: Classroom page	24
Figure 13: Add-classroom page.....	24
Figure 14: Add - reservation page.....	25
Figure 15: Lecture dashboard	25
Figure 16: Student schedule.....	26

2. Introduction and Background

The current process of managing classroom reservations at the university is inefficient, relying on manual methods such as Excel sheets. This often results in scheduling conflicts, double bookings, and a lack of real-time updates. As classroom resources are essential for smooth academic operations, the need for a streamlined and optimized solution has become evident. The proposed system will address these issues by providing a centralized platform for scheduling and managing classroom bookings.

This web-based system will allow university administrators to efficiently manage classroom details, create and update reservations, and oversee resource availability. The system will also support user management, enabling administrators to add or update user roles for both administrators and lecturers. Lecturers will benefit from having direct access to their schedules and the ability to request additional classrooms for extra sessions, improving communication and operational transparency.

In addition, the system will feature an open portal where students can view classroom schedules without needing an account. This will allow them to filter and access schedules by department and year, ensuring they stay informed about classroom availability and updates. By offering real-time access and reducing administrative burdens, the system will significantly improve the efficiency of classroom management for the university.

3. Business Analysis

3.1 Problem Identification

The current classroom reservation system relies on manual processes, leading to scheduling conflicts, double bookings, and a heavy administrative workload. This results in inefficiencies and frustrations for both university administrators and lecturers.

3.2 Stakeholder Analysis

Key stakeholders include university administrators (who manage reservations), lecturers (who need access to schedules and booking requests), and students (who require visibility of classroom availability). Understanding their needs ensures the system addresses each group's specific requirements.

3.3 Requirements Gathering

Requirements were gathered through direct feedback sessions with university administrators, who provided detailed insights into current scheduling challenges and system needs.

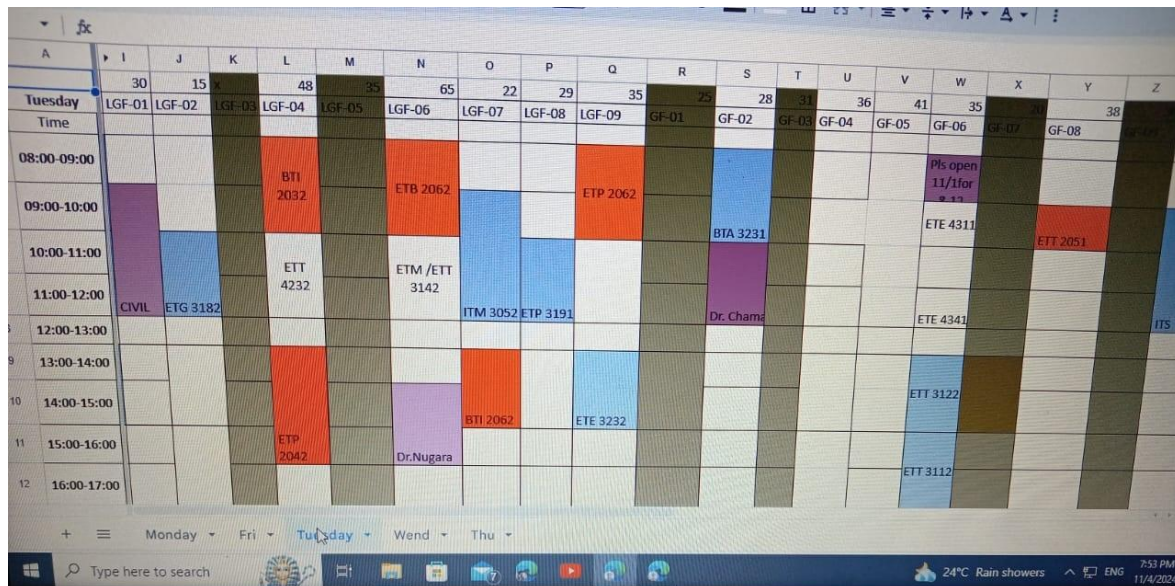


Figure 1: Manual reservation with Excel

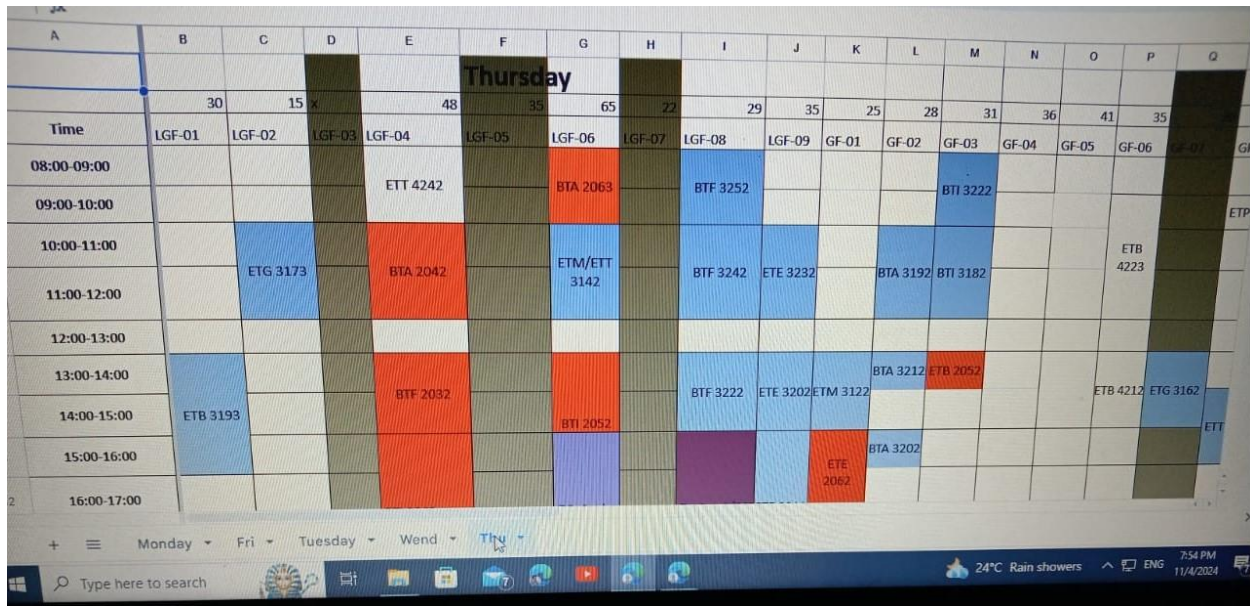


Figure 2: Manual reservation with Excel

Key requirements identified include:

Functional Requirements:

- User Management:
 - Admins can create, update, and delete accounts for other admins and lecturers.
 - Admins assign roles (admin or lecturer) to new users.
 - Users (admins and lecturers) can securely log in and log out.
- Classroom Management:
 - Admins can add, update, and delete classroom details (name, location, seating capacity).
 - Admins can specify resources (e.g., projectors, whiteboards) available in each classroom.
- Reservation Management:
 - Admins can create, update, and delete reservations, specifying subject, course code, year, lecturer, date, time, and recurrence.
 - Lecturers can request additional classrooms for extra sessions, subject to admin approval.

- Schedule View for Lecturers:
 - Lecturers can view their classroom schedules in calendar or list format.
- Public Schedule Portal for Students:
 - Students can access a public schedule without logging in.
 - Students can filter schedules by department, year, and classroom.

Non-functional Requirements:

- Usability
- Reliability
- Performance
- Security
- Scalability
- Maintainability
- Data Privacy

3.4 Gap Analysis

The existing manual system lacks automation and conflict prevention. The proposed solution introduces automated scheduling, real-time updates, and conflict detection, reducing errors and administrative workload.

3.5 Solution Outline

The system will include:

- **Role-Based Access:** Administrators, lecturers, and students with tailored functionalities.
- **Scheduling and Conflict Detection:** Automated booking system with real-time conflict prevention.
- **Public Access:** Students can view schedules without logging in.

3.6Expected Impact

The system is expected to improve scheduling efficiency, reduce conflicts, and optimize classroom resource usage, benefiting the university's operational workflow.

4. Software Development Process

For the Classroom Reservation Management System, we adopted an **Agile development process** to ensure flexibility and continuous feedback integration throughout the project. This approach enabled us to address stakeholder needs effectively and refine the system incrementally. The main phases of our development process were as follows:

4.1 Requirement Analysis and Planning

- We began by gathering requirements through direct discussions with university administrators, identifying key functional and non-functional requirements.
- Requirements were prioritized, and the development roadmap was created, dividing the project into manageable sprints.

4.2 System Design

- During this phase, we outlined the system architecture, designed database schemas, and defined user roles and permissions. We created wireframes and mockups to establish a clear vision of the user interface.
- The design phase included feedback from stakeholders to ensure alignment with user expectations.

4.2.1 Use Case Diagram

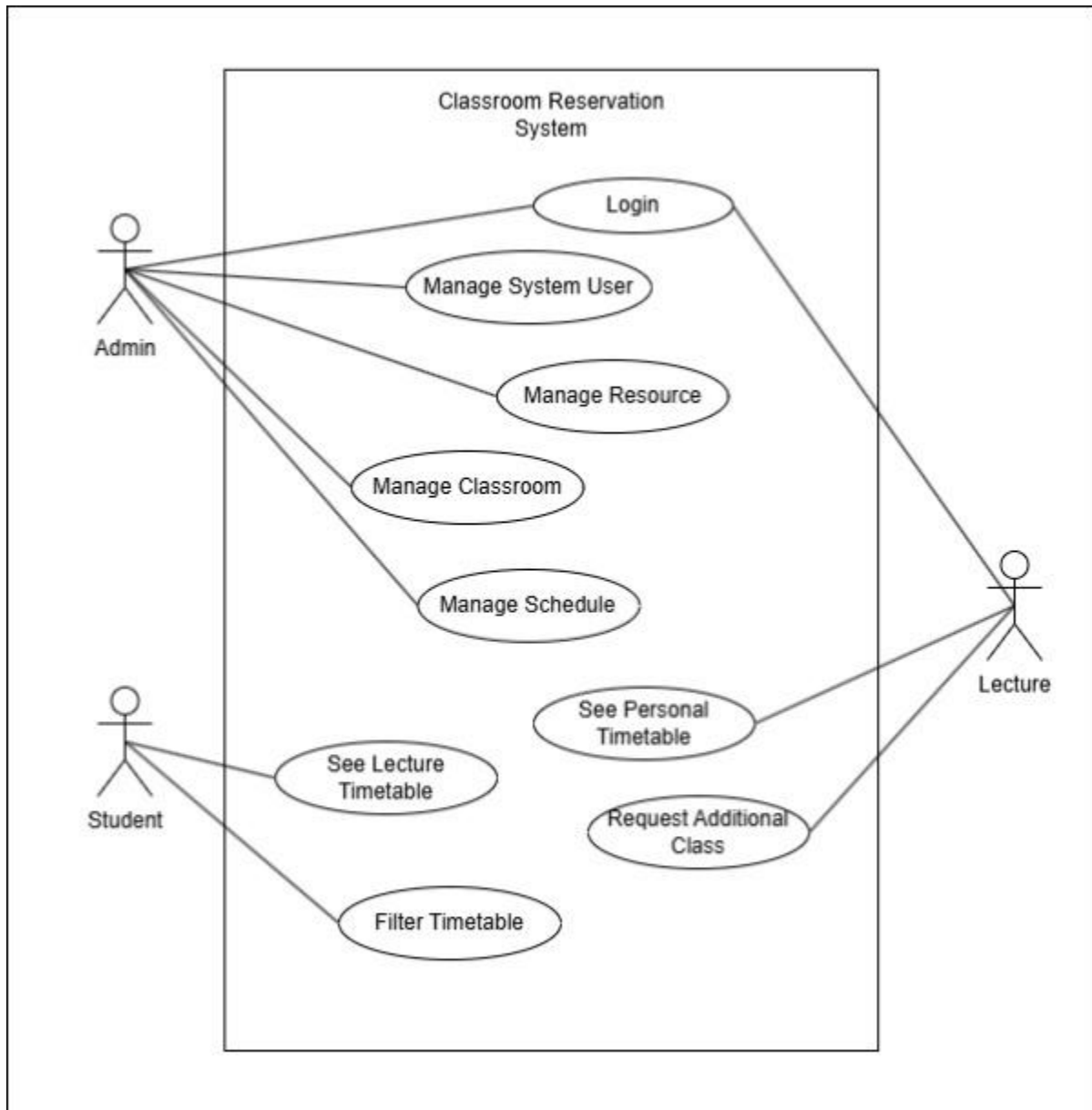


Figure 3: Use case diagram

4.2.2 Flow Chart

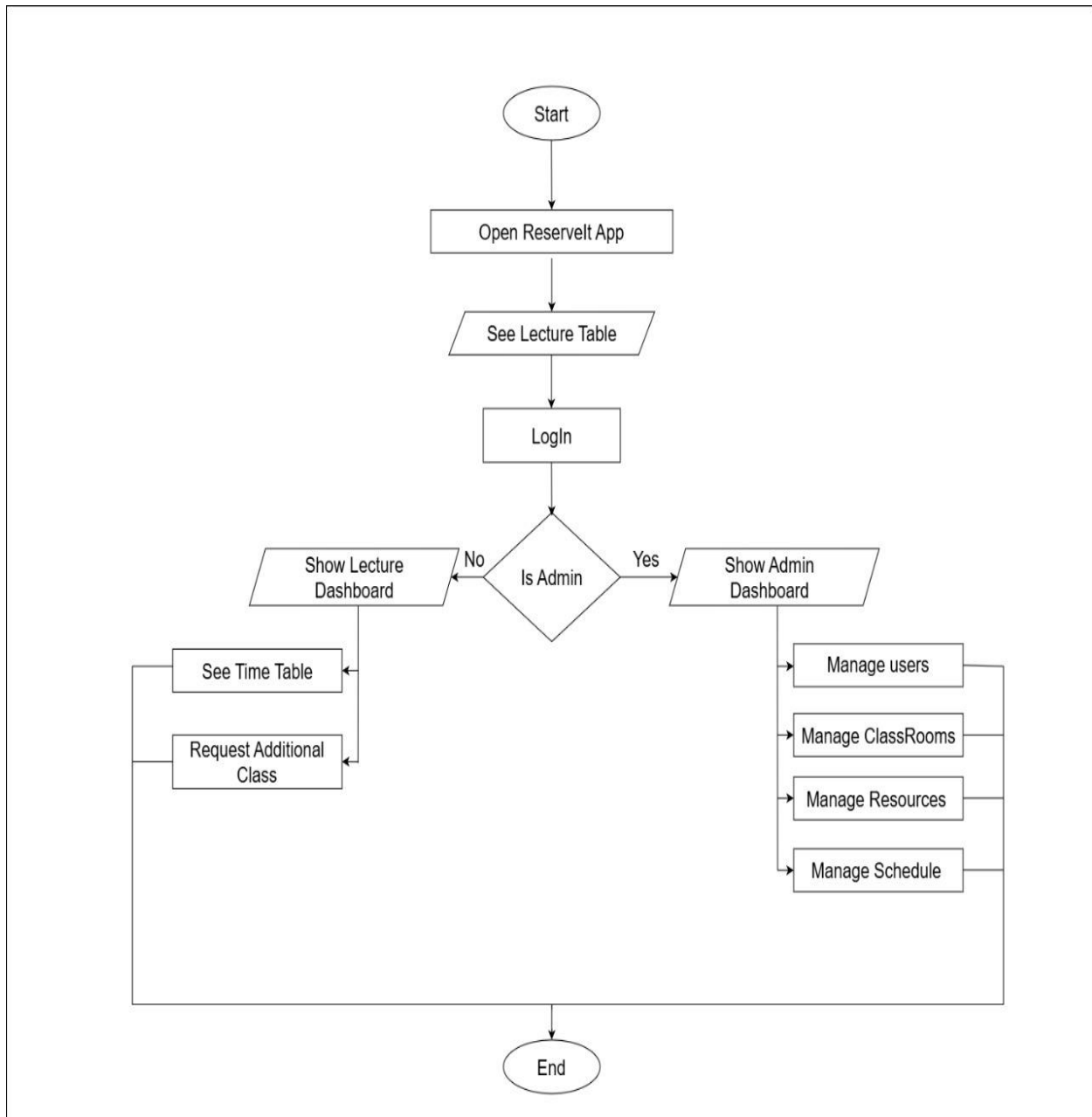


Figure 4: Flow chart

4.2.3 ER Diagram

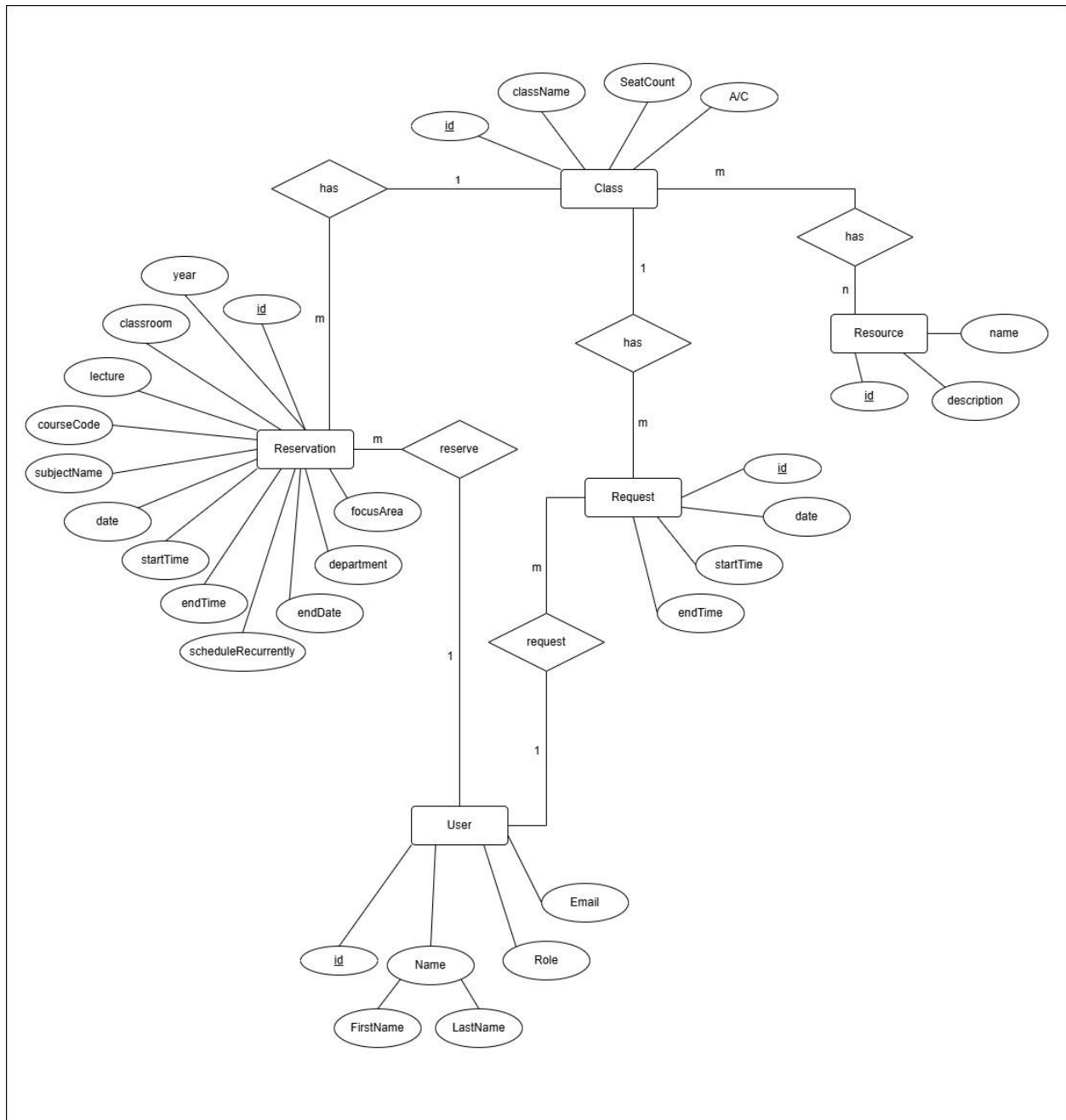


Figure 5: Entity Relationship diagram

4.2.4 Class Diagram

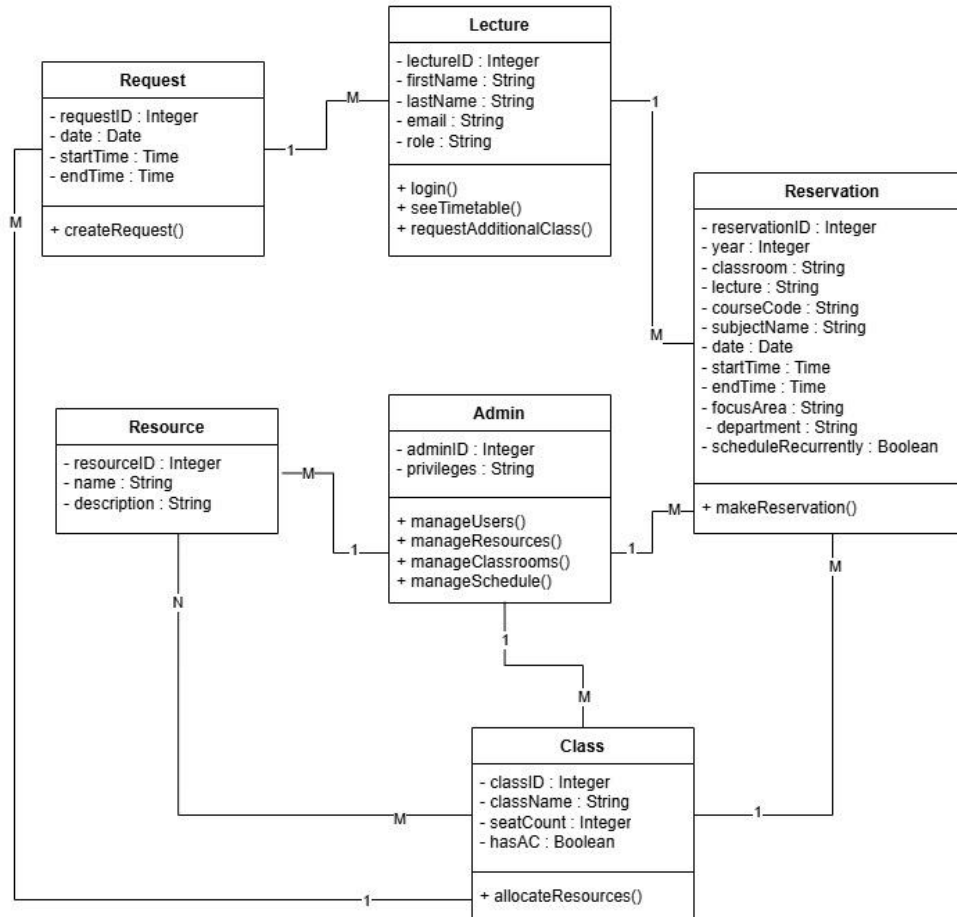


Figure 6: Class diagram

4.3 Development (Iterative Sprints)

- The development was divided into iterative sprints, each focusing on specific modules, such as user management, classroom management, and reservation scheduling.
- Continuous integration and code reviews were conducted to ensure code quality, and each module was tested after completion before moving to the next sprint.

4.4 Testing and Quality Assurance

- Unit and integration testing were conducted for each sprint to ensure the reliability of individual components.
- End-to-end testing was performed after all modules were integrated, focusing on functionality, security, and usability.

4.5 Deployment

- After successful testing, the system was deployed in a controlled environment, allowing administrators and lecturers to test the application's features.
- Feedback from the deployment stage was gathered and addressed to improve the final product.

4.6 Maintenance and Iteration

- Post-deployment, we implemented a maintenance plan to address any issues that arise and continue refining the system based on user feedback and evolving requirements.
- Regular updates and additional features will be added to enhance the application as needed.

5. Software Testing Process

To ensure the quality and reliability of the Classroom Reservation Management System, we followed a comprehensive software testing process. This process included different types of testing at each stage to identify and resolve any issues. The key testing phases were as follows:

5.1 Unit Testing

- Each module, such as User Management, Classroom Management, and Reservation Management, was tested individually to verify that all functions worked as expected.
- Unit tests were automated where possible to quickly detect issues in individual components during each sprint.

5.2 Integration Testing

- After unit testing, modules were integrated, and interactions between them were tested to ensure they functioned correctly as a whole.
- Key integrations, such as user login with reservation management, were tested to ensure data flows accurately across the system.

5.3 System Testing

- Complete end-to-end testing of the entire application was performed to ensure that all requirements were met.
- Functional tests were conducted to verify the core features, including user roles, scheduling, conflict detection, and public schedule views.

5.4 User Interface (UI) Testing

- UI testing was conducted to validate that the application interface is user-friendly and that all elements function as expected.
- This testing ensured the consistency of design, responsiveness on different devices, and intuitive navigation for users.

5.5Performance Testing

- Performance tests were conducted to assess the application's response time, load handling, and scalability, ensuring it performs well under normal and peak usage conditions.
- This was essential for handling multiple simultaneous users, especially during peak scheduling periods.

5.6Security Testing

- Security testing was done to identify and mitigate potential vulnerabilities, ensuring user data protection and system security.
- Tests included verifying secure user authentication, access control, and protection against common vulnerabilities.

6. Software deployment process

Our production deployment strategy focuses on leveraging AWS infrastructure to ensure security, scalability, and high availability. The deployment involves setting up both public and private subnets within a Virtual Private Cloud (VPC), with specific roles assigned to each component of the application.

For development purposes, we utilize Vercel to streamline the deployment process, allowing for quick iterations and feedback.

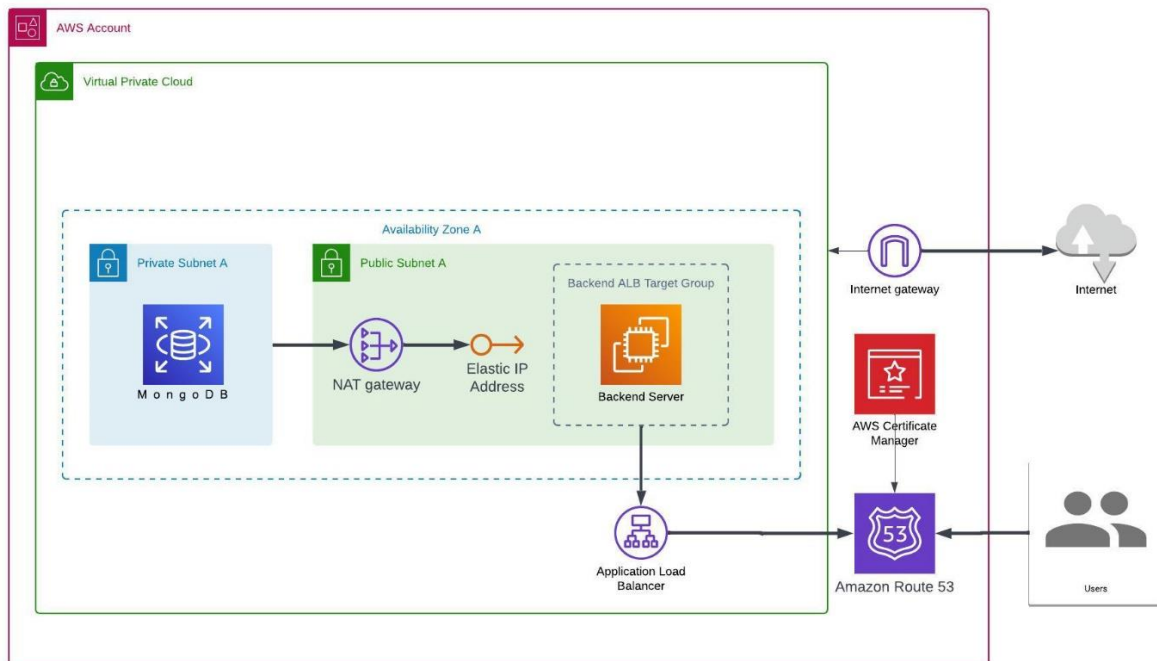


Figure 7:Datacenter Network

Deploying on Vercel allowed us to take advantage of its ease of use, scalability, and continuous deployment features. The deployment process was as follows:

6.1 Code Repository Integration

- Linked the application's Git repository (GitHub/GitLab) to Vercel, enabling seamless continuous deployment.
- Configured Vercel to automatically deploy any changes pushed to the main branch, allowing for easy updates and fast iteration.

6.2 Environment Configuration

- Set up environment variables on Vercel to securely manage sensitive information like database credentials, API keys, and authentication tokens.
- Configured the necessary backend endpoints and connected the application to the database to ensure smooth functionality.

6.3 Initial Deployment and Build

- Triggered the initial deployment through Vercel, which automatically built the application using its integrated build system optimized for modern web technologies.
- Verified that the deployment build was successful, checking for errors in the deployment logs.

6.4 Testing in the Production Environment

- Performed post-deployment tests on Vercel to verify that core functionalities (e.g., login, classroom scheduling, and schedule viewing) worked as expected in the live environment.
- Conducted checks for mobile responsiveness, load performance, and user experience on Vercel's globally distributed CDN.

7. Application Interfaces

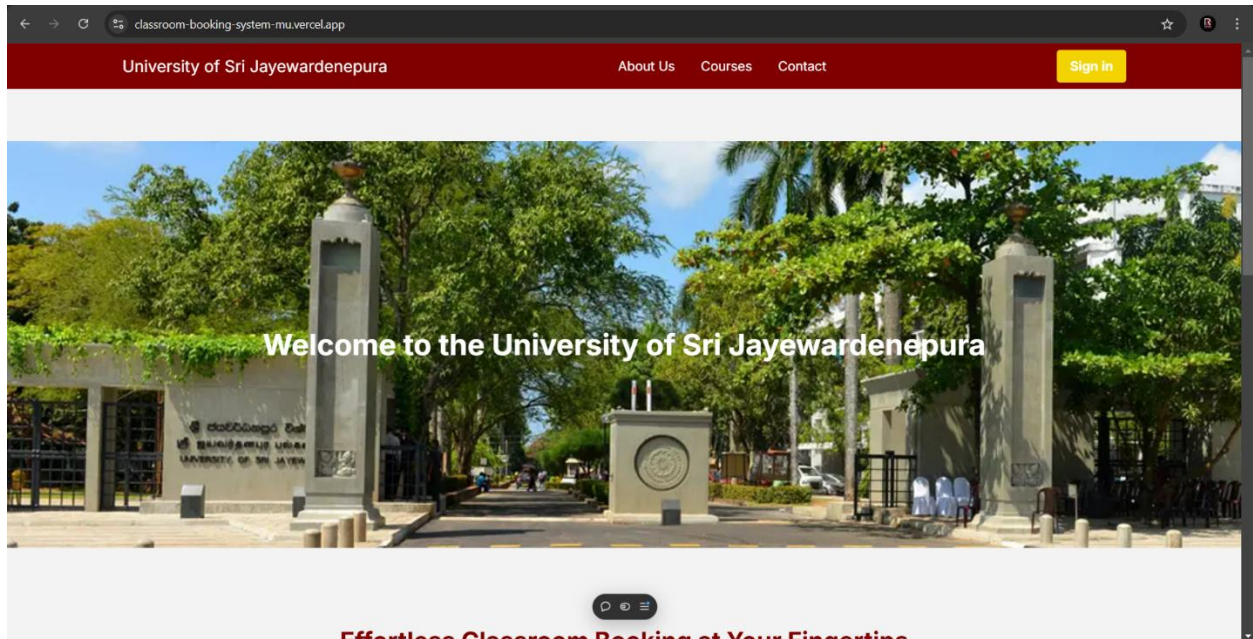


Figure 8: Home page

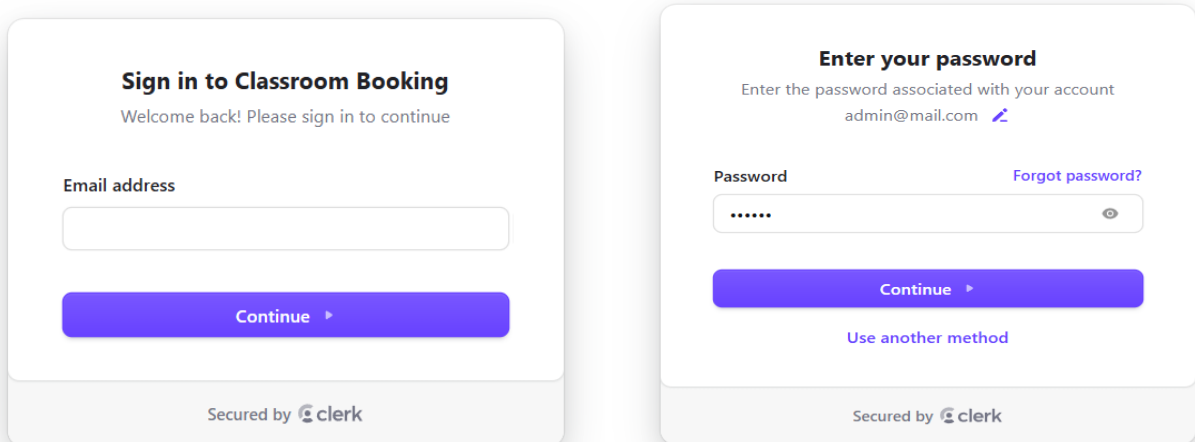


Figure 9: Login page

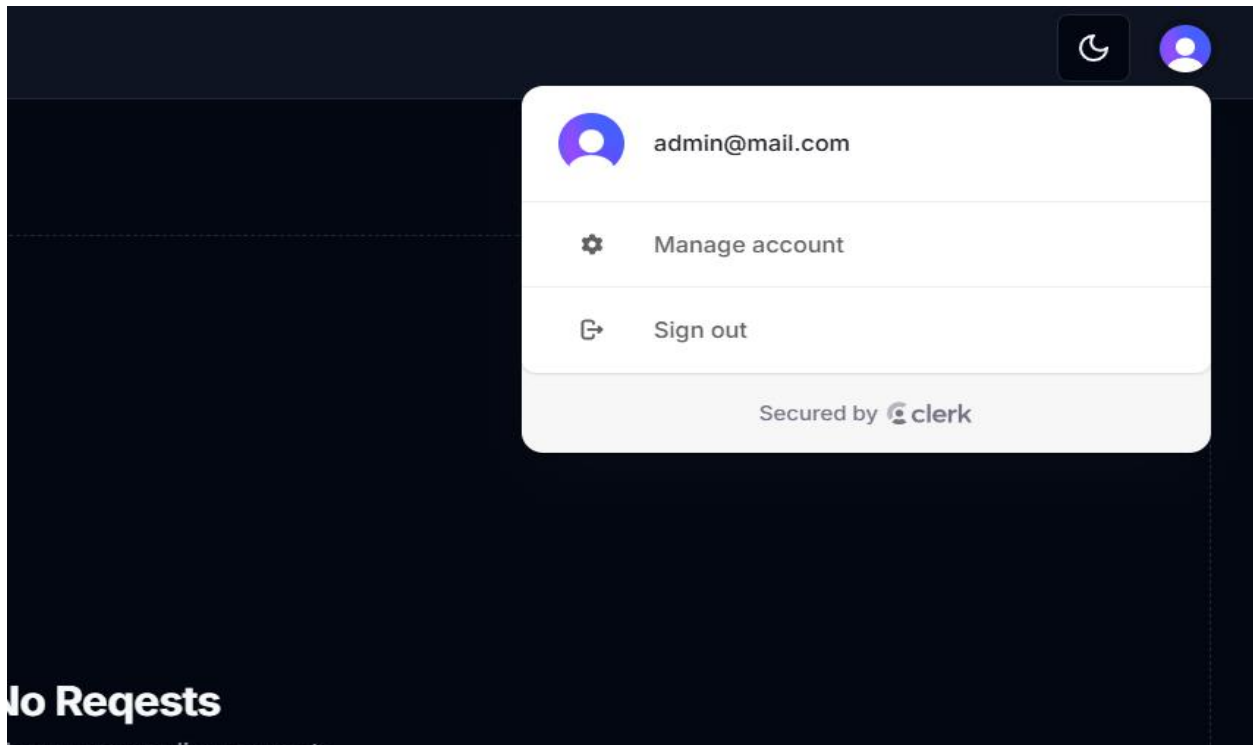


Figure 10: User Profile

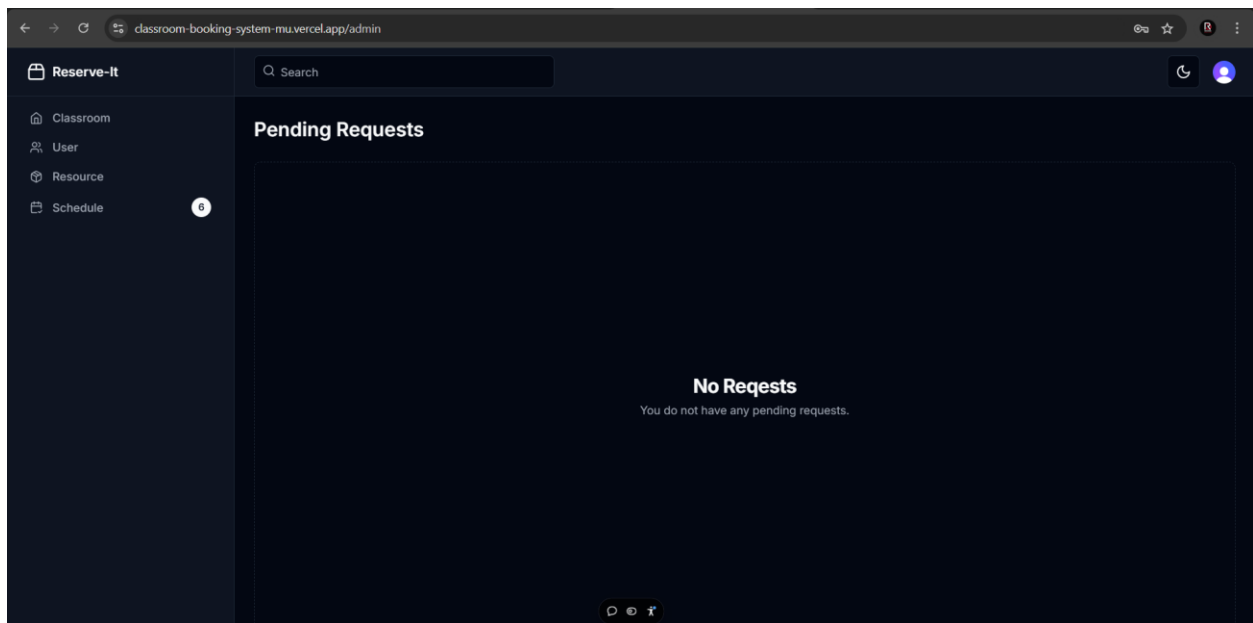


Figure 11: Admin dashboard

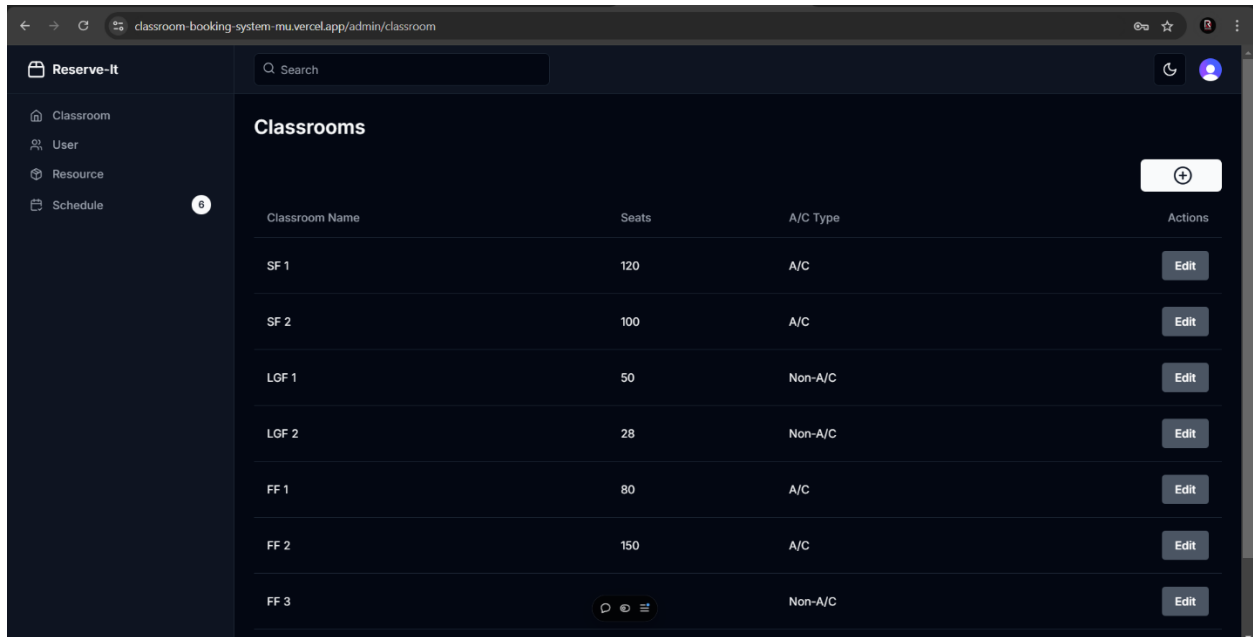


Figure 12: Classroom page

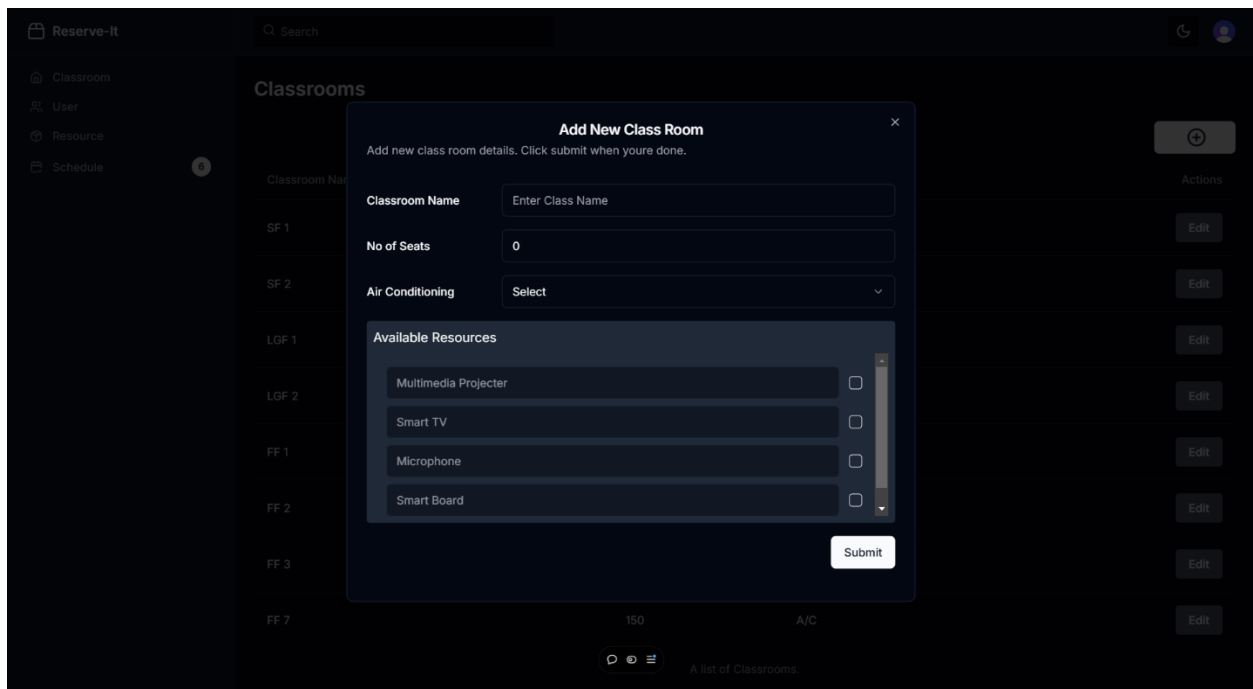


Figure 13: Add-classroom page

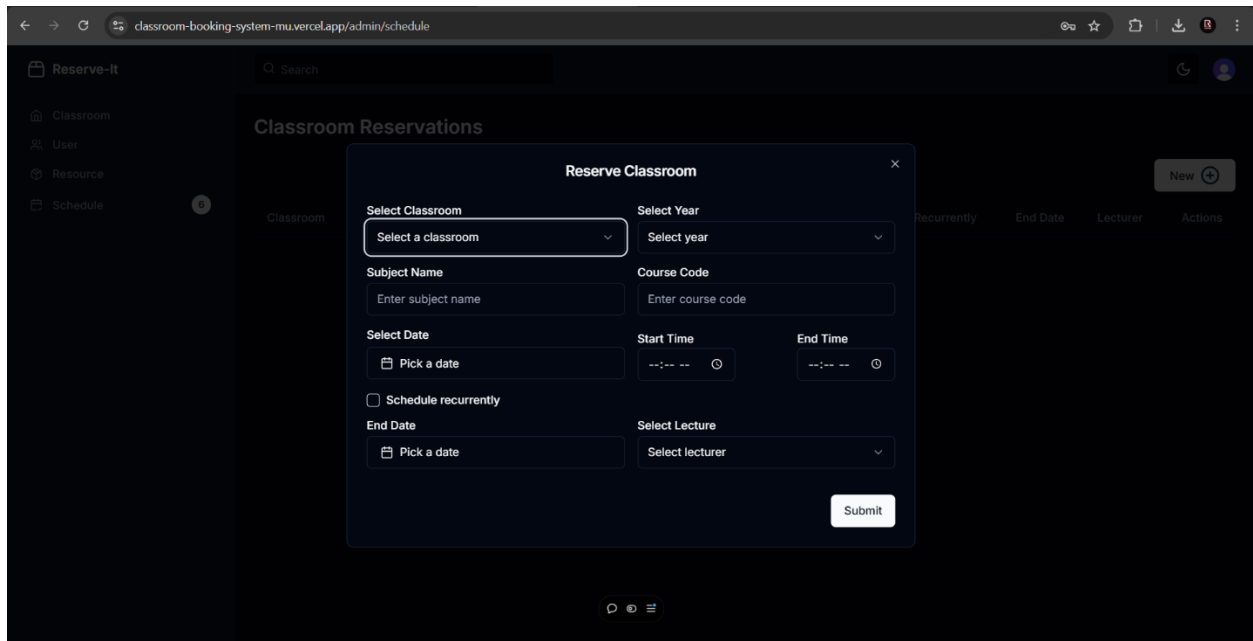


Figure 14: Add - reservation page

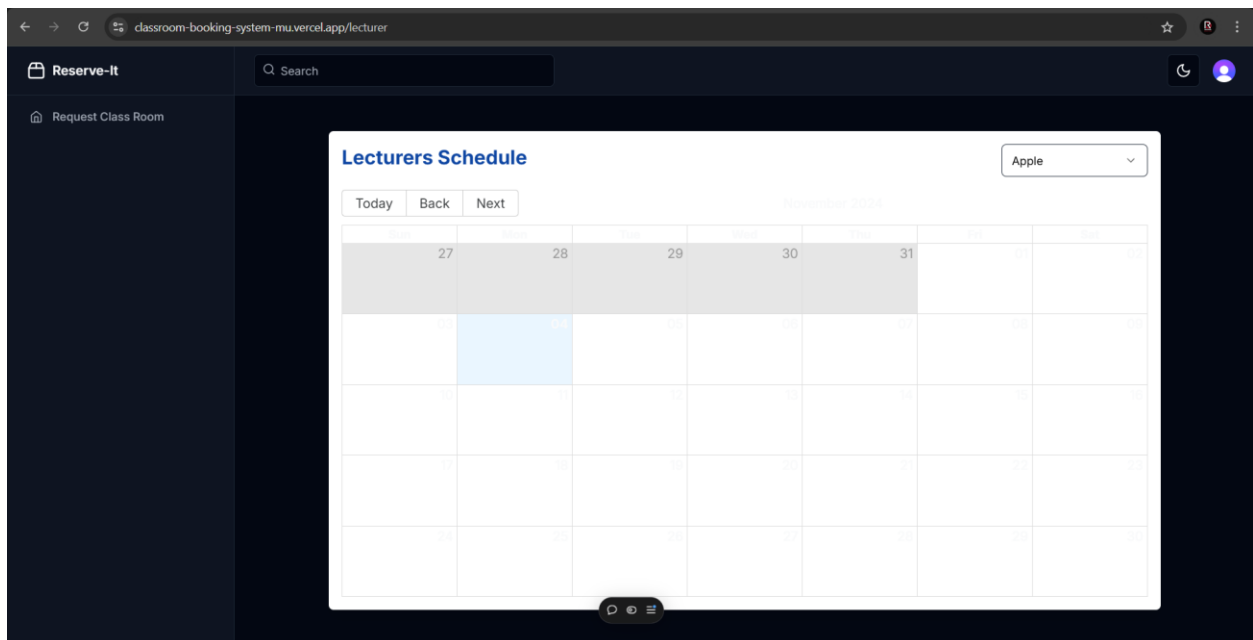


Figure 15: Lecture dashboard

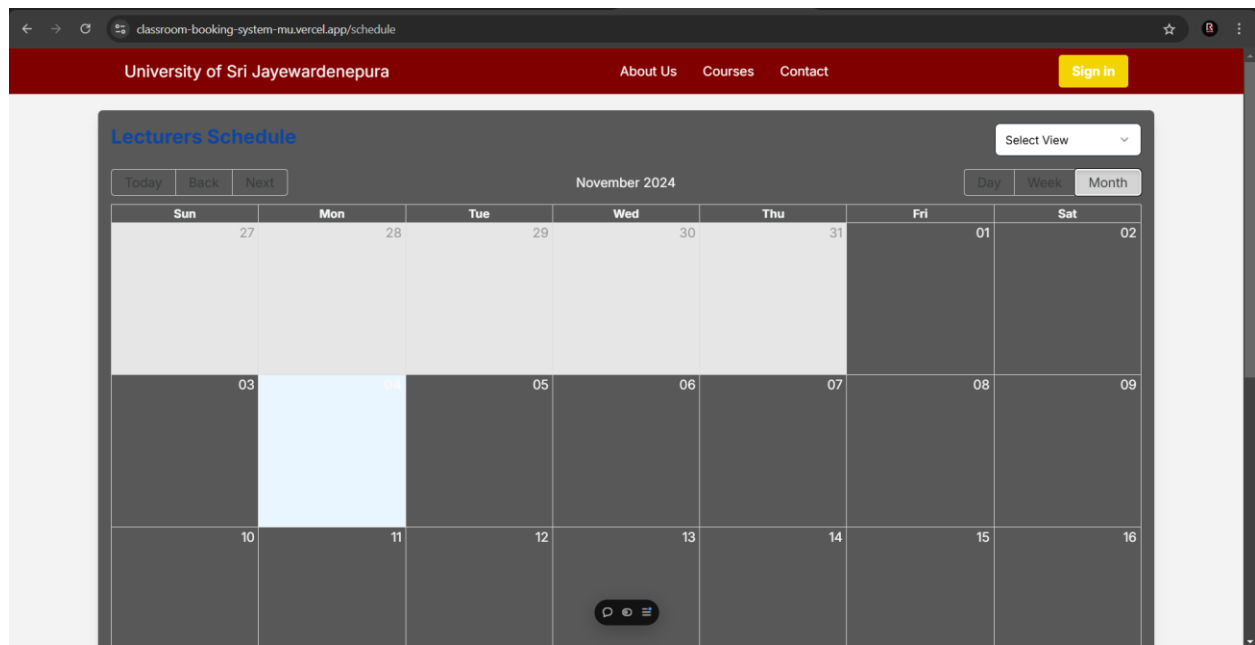


Figure 16: Student schedule

8. Discussion

The Classroom Reservation Management System provides a streamlined, web-based solution for managing university classroom bookings, replacing the manual process. Key features include role-based access for administrators, lecturers, and students, real-time conflict detection to prevent double bookings, and a public schedule portal for students. Deployment on Vercel ensures scalability, security, and reliable performance.

The system improves efficiency by enabling administrators to manage classrooms and users, while lecturers can view schedules and request additional bookings and the public portal allows students to easily view relevant schedules.

8.1 Challenges

- **Data Accuracy:** Ensuring data accuracy during migration from the manual system to the digital platform.
- **User Adoption:** Getting administrators and lecturers accustomed to the new system, given their prior reliance on manual methods.
- **Conflict Resolution:** Managing last-minute conflicts due to unexpected events or classroom changes.

8.2 Future Work

- **Notification System:** Adding email or SMS alerts for booking confirmations, updates, and reminders.
- **Analytics Dashboard:** Providing administrators with insights into classroom usage and booking patterns.
- **Mobile Optimization:** Enhancing the mobile user experience for lecturers and students.
- **Additional Integrations:** Integrating with other campus systems, such as attendance or resource management, for a more comprehensive solution.