

FEDERAL UNIVERSITY OYE-EKITI

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

FUOYE SMART CAMPUS MANAGEMENT SYSTEM

A Course Project Report Submitted to the Department of Computer Science, Federal University Oye-Ekiti

In Partial Fulfillment of the Requirements for CSC 320 (System Analysis and Design)

Submitted By
1. Ajagbe Uthman Alade - CSC/2024/2003
2. Ayeni Toluwani David - CSC/2023/1061
3. Ebute Grace otinu - CSC/2023/1077
4. Stephen Rhema Bayode - CSC/2023/1175

Project Group B

Supervisor: DR AYOGU

Department of Computer Science

Federal University Oye-Ekiti

Academic Session: 2025/2026

June, 2026

CERTIFICATION

This is to certify that this project titled **"FUOYE Smart Campus Management System"** was carried out by the underlisted students of the Department of Computer Science, Federal University Oye-Ekiti, in partial fulfillment of the requirements for CSC 320 (System Analysis and Design).

The project has been examined and approved as meeting the requirements of the Department of Computer Science.

Name
Signature
Date

Project Supervisor

_____ _____ _____

DEDICATION

This project is dedicated to Almighty God for His guidance, strength, wisdom, and protection throughout the duration of this study.

We also dedicate this work to our parents, guardians, lecturers, and colleagues whose encouragement, support, and contributions made the successful completion of this project possible.

ACKNOWLEDGEMENT

We express our profound gratitude to Almighty God for granting us the knowledge, strength, and opportunity to successfully complete this project.

Our sincere appreciation goes to our project supervisor for the valuable guidance, constructive criticisms, and continuous support provided throughout the course of this project.

We also appreciate the Head of Department and all lecturers in the Department of Computer Science, Federal University Oye-Ekiti, for their academic contributions and dedication towards our development.

Special thanks go to our parents, guardians, friends, and fellow students for their encouragement, prayers, and support during the course of this work.

Finally, we acknowledge everyone whose contributions, directly or indirectly, aided the successful completion of this project.

ABSTRACT

The increasing population of students and growing demand for campus facilities have created challenges in managing hostel accommodations and classroom reservations within higher institutions. Traditional methods of handling these resources often result in inefficiencies, delays, double bookings, poor record keeping, and difficulties in monitoring space utilization.

This project presents the design and implementation of the **FUOYE Smart Campus Management System**, a web-based platform developed to streamline the management of hostel accommodations and classroom bookings within Federal University Oye-Ekiti.

The system provides students with the ability to view available hostel spaces, reserve classrooms, manage bookings, and receive updates through a centralized platform. Administrators are provided with tools to monitor facilities, manage student bookings, track occupancy levels, and generate useful insights for decision-making.

The system was developed using React, Vite, Tailwind CSS, JavaScript, and browser local storage technology for data persistence. Various system analysis and design methodologies including use case diagrams, activity diagrams, sequence diagrams, flowcharts, and data flow diagrams were utilized during development.

The implemented solution demonstrates how digital technologies can improve resource allocation, reduce administrative workload, enhance user experience, and increase operational efficiency within university environments.

TABLE OF CONTENTS

Preliminary Pages

Title Page

Certification

Dedication

Acknowledgement

Abstract

Table of Contents

List of Figures

List of Tables

CHAPTER ONE

1.0 Introduction

1.1 Statement of the Problem

1.2 Aim and Objectives

CHAPTER TWO

2.0 Overview of the Existing System

2.1 Overview of the Proposed System

2.2 Functional and Non-Functional Requirements

2.3 Methodology

2.3.1 Use Case Diagram

2.3.2 Activity Diagram

2.3.3 Sequence Diagram

2.3.4 Data Flow Diagram

2.3.5 Flowchart

CHAPTER THREE

3.0 Implementation Tools

3.1 Hardware and Software Requirements

3.2 Description of the New System

3.3 Results and Discussion

3.3.1 Login Page

3.3.2 Student Registration Page

3.3.3 Dashboard

3.3.4 Hostel Management Module

3.3.5 Classroom Booking Module

3.3.6 Student Profile Module

3.3.7 Admin Dashboard

3.3.8 Admin Management Module

3.3.9 System Testing

3.3.10 Screenshots of Implemented System

CHAPTER FOUR

4.0 Conclusion

4.1 Recommendations

REFERENCES

APPENDICES

Appendix A: Source Code Snippets

Appendix B: Additional System Screenshots

LIST OF FIGURES

Figure 2.1: Use Case Diagram

Figure 2.2: Activity Diagram

Figure 2.3: Sequence Diagram

Figure 2.4: Data Flow Diagram

Figure 2.5: System Flowchart

Figure 3.1: Login Page Interface

Figure 3.2: Registration Page Interface

Figure 3.3: Student Dashboard

Figure 3.4: Hostel Listing Page

Figure 3.5: Hostel Details Page

Figure 3.6: Classroom Booking Page

Figure 3.7: Student Profile Page

Figure 3.8: Admin Dashboard

Figure 3.9: Booking Management Interface

LIST OF TABLES

Table 2.1: Functional Requirements

Table 2.2: Non-Functional Requirements

Table 3.1: Hardware Requirements

Table 3.2: Software Requirements

Table 3.3: System Testing Results

Table 3.4: User Roles and Permissions

CHAPTER ONE

1.0 INTRODUCTION

The rapid advancement of information and communication technology has transformed the way organizations manage information, automate processes, and deliver services. Educational institutions are among the sectors that have greatly benefited from these technological developments. Universities now rely on digital systems to improve administrative efficiency, enhance service delivery, and provide students with convenient access to important resources.

One of the major challenges faced by many tertiary institutions is the effective management of campus facilities such as student hostels and classrooms. In many cases, hostel allocation and classroom reservation processes are still handled manually or through fragmented systems that do not provide real-time information. These traditional methods often result in delays, inaccurate records, overcrowding, double booking, and difficulties in monitoring facility usage.

Federal University Oye-Ekiti (FUOYE), like many growing institutions, requires a more efficient approach to managing campus spaces. Students often encounter difficulties in obtaining timely information about available hostel spaces, while administrators face challenges in monitoring bookings and managing accommodation records effectively. Similarly, the process of reserving classrooms for academic activities may become cumbersome when there is no centralized platform for managing requests and approvals.

The FUOYE Smart Campus Management System was developed to address these challenges by providing a centralized web-based platform for hostel management and classroom booking. The system enables students to create accounts, view available hostels, make booking requests, reserve classrooms, monitor booking status, and receive notifications. Administrators are provided with management tools for reviewing requests, approving or rejecting bookings, monitoring student activities, and generating useful insights regarding facility utilization.

The implementation of this system contributes to improved transparency, efficiency, accessibility, and effective management of campus facilities. By leveraging modern web technologies, the platform offers a user-friendly and scalable solution that can be further enhanced to support future smart campus initiatives.

1.1 STATEMENT OF THE PROBLEM

The management of hostels and classrooms within many higher institutions is often characterized by manual procedures that are inefficient and prone to errors. Students frequently experience difficulties obtaining accurate information about available accommodation spaces, while administrators face challenges in tracking occupancy records and processing requests effectively.

The absence of a centralized management system may result in several problems, including delays in booking processes, inaccurate allocation records, lack of transparency, poor communication between students and administrators, and inefficient utilization of available facilities. Students may spend considerable time searching for accommodation information, while administrators struggle to maintain updated records and monitor bookings.

Furthermore, classroom reservation processes can become disorganized when requests are handled manually, leading to scheduling conflicts, double bookings, and difficulties in monitoring facility usage.

These challenges highlight the need for an automated system capable of managing hostel accommodation and classroom reservations efficiently while providing accurate information, improving communication, and enhancing the overall user experience.

1.2 AIM AND OBJECTIVES

The aim of this project is to design and develop a web-based Smart Campus Management System that facilitates efficient hostel management and classroom booking within Federal University Oye-Ekiti.

The objectives of the project are to:

1. Develop a centralized platform for managing hostel accommodation and classroom reservations.

2. Provide students with access to real-time information about available hostel spaces.

3. Enable students to create accounts and securely access the system.

4. Allow students to submit hostel booking requests electronically.

5. Provide classroom reservation functionality for academic activities.

6. Enable administrators to approve or reject booking requests.

7. Maintain records of student bookings and facility utilization.

8. Provide notifications and status updates regarding booking activities.

9. Improve transparency and efficiency in the management of campus facilities.

10. Develop a responsive and user-friendly interface accessible across different devices.

CHAPTER TWO

2.0 OVERVIEW OF THE EXISTING SYSTEM

Space allocation and facility management are important aspects of university administration. At the Federal University Oye-Ekiti (FUOYE), students frequently require access to hostel accommodation and academic spaces such as lecture halls, seminar rooms, and classrooms. The existing process of managing these facilities is largely manual and involves physical inquiries, paper documentation, and direct interaction with administrative staff.

Students seeking hostel accommodation often have to visit multiple locations physically before obtaining information regarding room availability. This process consumes time and effort and may still not provide accurate or updated information. Similarly, booking classrooms for tutorials, departmental meetings, seminars, and academic activities usually requires manual approval procedures that may lead to delays, scheduling conflicts, and poor record keeping.

The existing system lacks a centralized platform where students can conveniently access information about available facilities and make reservations electronically. Administrative officers also face challenges in monitoring occupancy levels, tracking booking requests, and generating reports for decision making.

The manual nature of the existing system introduces inefficiencies that affect both students and administrators. As student populations continue to increase, the need for a digital solution that streamlines accommodation and classroom management becomes increasingly important.

Limitations of the Existing System

The existing system suffers from several shortcomings, including:

1. Lack of real-time hostel availability information.
2. Manual booking and approval procedures.
3. Difficulty in tracking reservation history.
4. Increased paperwork and administrative workload.
5. Poor communication between students and administrators.
6. Risk of double booking and scheduling conflicts.
7. Lack of centralized data management.

8. Time-consuming processes that reduce operational efficiency.

These limitations highlight the necessity for an automated campus management system capable of improving service delivery and user experience.

2.1 OVERVIEW OF THE PROPOSED SYSTEM

The proposed system is the FUOYE Smart Campus Management System, a web-based platform developed to automate the management of hostel accommodation and classroom reservations within the university environment.

The system provides a centralized portal where students can register accounts, log in securely, view available hostels, make accommodation requests, reserve classrooms, monitor booking status, and manage personal profiles. Administrators are provided with dedicated management dashboards that enable them to oversee facilities, approve reservations, monitor occupancy rates, and manage student activities.

The proposed system leverages modern web technologies including React, Vite, Tailwind CSS, and Local Storage to create an interactive and responsive user experience suitable for academic demonstration purposes.

The system is designed around role-based access control, ensuring that students and administrators have access only to functionalities relevant to their responsibilities.

Key Features of the Proposed System

The proposed system offers the following features:

Student Features

- Student registration
- Student login and authentication
- Hostel browsing
- Hostel room availability tracking
- Hostel booking requests
- Classroom reservation requests

- Booking history management
- Profile management
- Notification management
- Settings management

Administrative Features

- Admin authentication
- Hostel management
- Classroom management
- Student management
- Booking approval and rejection
- Occupancy monitoring
- Analytics dashboard
- System configuration management

Benefits of the Proposed System

The implementation of the proposed system provides several advantages:

1. Faster access to accommodation information.
2. Reduced administrative workload.
3. Improved booking accuracy.
4. Real-time monitoring of facility utilization.
5. Better record management.

6. Enhanced user experience.
7. Improved transparency in booking operations.
8. Reduced scheduling conflicts.
9. Increased operational efficiency.
10. Better decision-making through analytics and reporting.

2.2 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Functional Requirements

Functional requirements describe the specific operations and services that the FUOYE Smart Campus Management System must perform in order to achieve its objectives.

The system was designed with two major categories of users:

1. Students
2. Administrators

Each category has different responsibilities and privileges within the system.

Student Functional Requirements

The system shall allow students to create accounts through the registration page.

The system shall allow registered students to log into the platform using their matric number and password.

The system shall provide students with access to a personalized dashboard containing summaries of their bookings and activities.

The system shall allow students to browse available hostels and view hostel details.

The system shall display information such as:

- Hostel name

- Hostel type
- Room category
- Occupancy status
- Availability information
- Hostel images

The system shall allow students to submit hostel booking requests.

The system shall allow students to browse available classrooms.

The system shall allow students to reserve classrooms for academic activities.

The system shall allow students to view all previous bookings.

The system shall allow students to monitor booking status such as:

- Pending
- Approved
- Rejected
- Cancelled

The system shall allow students to manage their profile information.

The system shall provide notification alerts regarding booking activities.

The system shall allow students to update application preferences through the settings page.

Administrator Functional Requirements

The system shall provide a secure administrator login interface.

The system shall allow administrators to access an administrative dashboard.

The system shall allow administrators to view all booking requests submitted by students.

The system shall allow administrators to approve hostel reservations.

The system shall allow administrators to reject hostel reservations.

The system shall allow administrators to manage hostel records.

The system shall allow administrators to manage classroom information.

The system shall allow administrators to monitor facility occupancy.

The system shall allow administrators to view student activity records.

The system shall allow administrators to generate statistical reports and analytics.

The system shall allow administrators to configure system settings.

The system shall maintain records of all booking activities.

Authentication Functional Requirements

The system shall support role-based authentication.

The system shall distinguish between student users and administrative users.

The system shall prevent unauthorized access to protected pages.

The system shall automatically redirect users based on their assigned roles.

The system shall maintain user sessions using browser local storage.

The system shall provide password validation during login.

The system shall support account registration for students.

The system shall prevent duplicate registration using existing matric numbers or email addresses.

Booking Management Functional Requirements

The system shall record all booking requests.

The system shall assign booking statuses automatically.

The system shall update booking records when approval actions are taken.

The system shall allow students to cancel pending bookings.

The system shall maintain booking history records.

The system shall notify users when booking statuses change.

Non-Functional Requirements

Non-functional requirements describe the quality attributes of the system rather than specific functions.

Performance Requirements

The system should load pages quickly and efficiently.

The system should respond to user actions within a short period of time.

The system should support multiple booking records without significant performance degradation.

The system should maintain smooth navigation between pages.

Usability Requirements

The system should provide a simple and intuitive user interface.

The system should be easy to learn and use by students with little technical experience.

The system should provide clear navigation menus.

The system should display understandable error messages.

The system should maintain a consistent design throughout all pages.

Reliability Requirements

The system should store user information securely within local storage.

The system should preserve user sessions during browser refresh operations.

The system should maintain accurate booking records.

The system should minimize data loss during normal operation.

Security Requirements

The system should implement role-based access control.

The system should prevent students from accessing administrative pages.

The system should protect administrative functions from unauthorized users.

The system should validate login credentials before granting access.

The system should maintain user authentication status during active sessions.

Maintainability Requirements

The system should be developed using a modular architecture.

The system should separate components, pages, utilities, and layouts into independent modules.

The system should allow future integration with backend services and databases.

The system should support future enhancements without requiring major redesign.

Scalability Requirements

The system should accommodate an increasing number of users.

The system should support future deployment with cloud-based databases.

The system should support additional modules such as:

- Payment Integration
- Hostel Allocation Automation
- Email Notifications
- SMS Notifications
- AI-Based Space Recommendation Systems

Compatibility Requirements

The system should operate correctly on modern web browsers.

The system should support:

- Google Chrome
- Microsoft Edge
- Mozilla Firefox
- Safari

The system should provide responsive layouts for:

- Desktop devices
- Tablets
- Mobile phones

Availability Requirements

The system should remain accessible whenever users require access.

The system should provide uninterrupted access during normal academic operations.

The system should maintain functionality across different screen sizes and operating systems.

Requirement Summary Table

Requirement Category	Description

Authentication	Login, Registration, Role Management
Hostel Management	View Hostels, Availability, Booking
Classroom Management	Reservation and Scheduling
Booking Management	Approval, Rejection, Tracking
Student Management	Profile and Activity Monitoring
Notification System	Booking Updates and Alerts
Analytics	Occupancy and Booking Statistics
Security	Role-Based Access Control
Usability	Responsive and User-Friendly Interface
Scalability	Future Database and Cloud Integration

2.3 METHODOLOGY

Introduction

The methodology adopted for the development of the FUOYE Smart Campus Management System is the Incremental Development Methodology.

This methodology involves developing the system in stages, where each stage adds new functionality while maintaining previously developed features. The approach was chosen because it allows continuous testing, easy modification, and gradual improvement of system components throughout the development process.

The development process was divided into the following phases:

Requirement Gathering

Information was gathered regarding hostel allocation challenges, classroom reservation procedures, and administrative requirements within the university environment.

System Analysis

Existing manual procedures were studied to identify weaknesses and determine the requirements of the proposed solution.

System Design

System architecture, user interfaces, database structure, workflows, and navigation structures were designed.

System Development

The application was developed using React, Vite, JavaScript, Tailwind CSS, and Local Storage technologies.

Testing

The developed system was tested to ensure all functionalities operated correctly and met user requirements.

Deployment

The completed application was prepared for hosting and demonstration purposes.

2.3.1 USE CASE DIAGRAM

The Use Case Diagram illustrates the interactions between users and the system.

Actors

Student

Administrator

Student Use Cases

Register Account

Login

View Dashboard

Browse Hostels

Book Hostel

Book Classroom

View Booking Status

Manage Profile

View Notifications

Manage Settings

Logout

Administrator Use Cases

Login

View Dashboard

Manage Hostels

Manage Classrooms

Approve Bookings

Reject Bookings

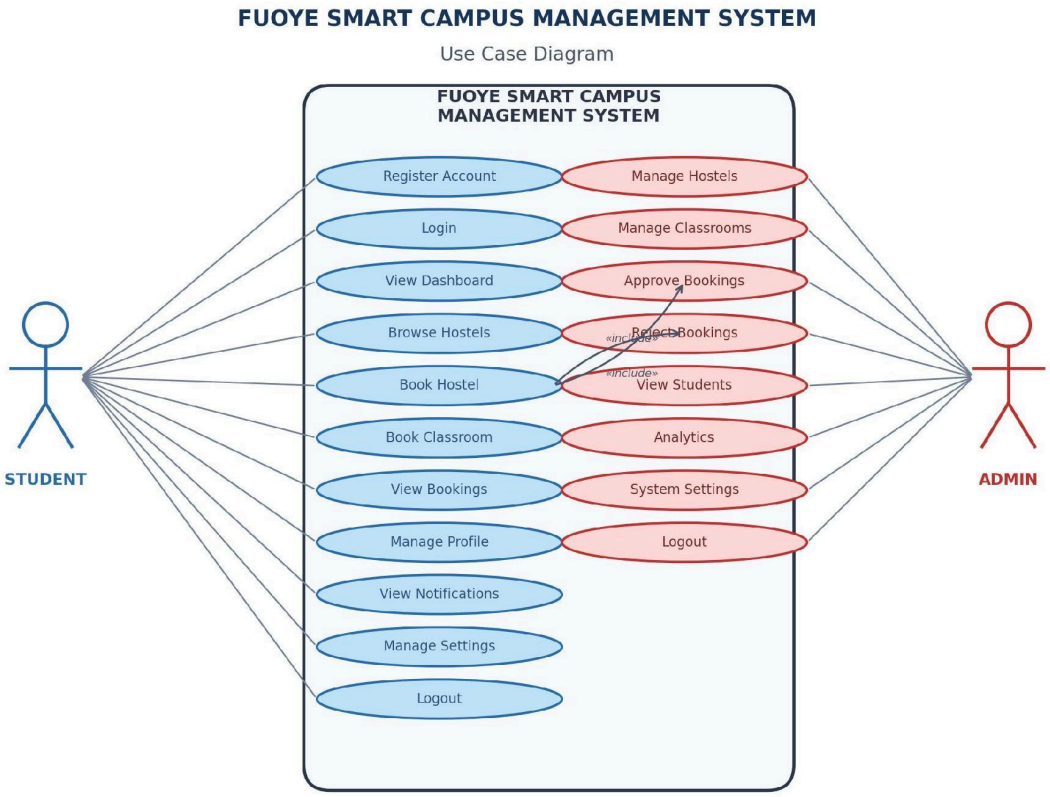
View Students

View Analytics

Manage Settings

Logout

Use Case Diagram



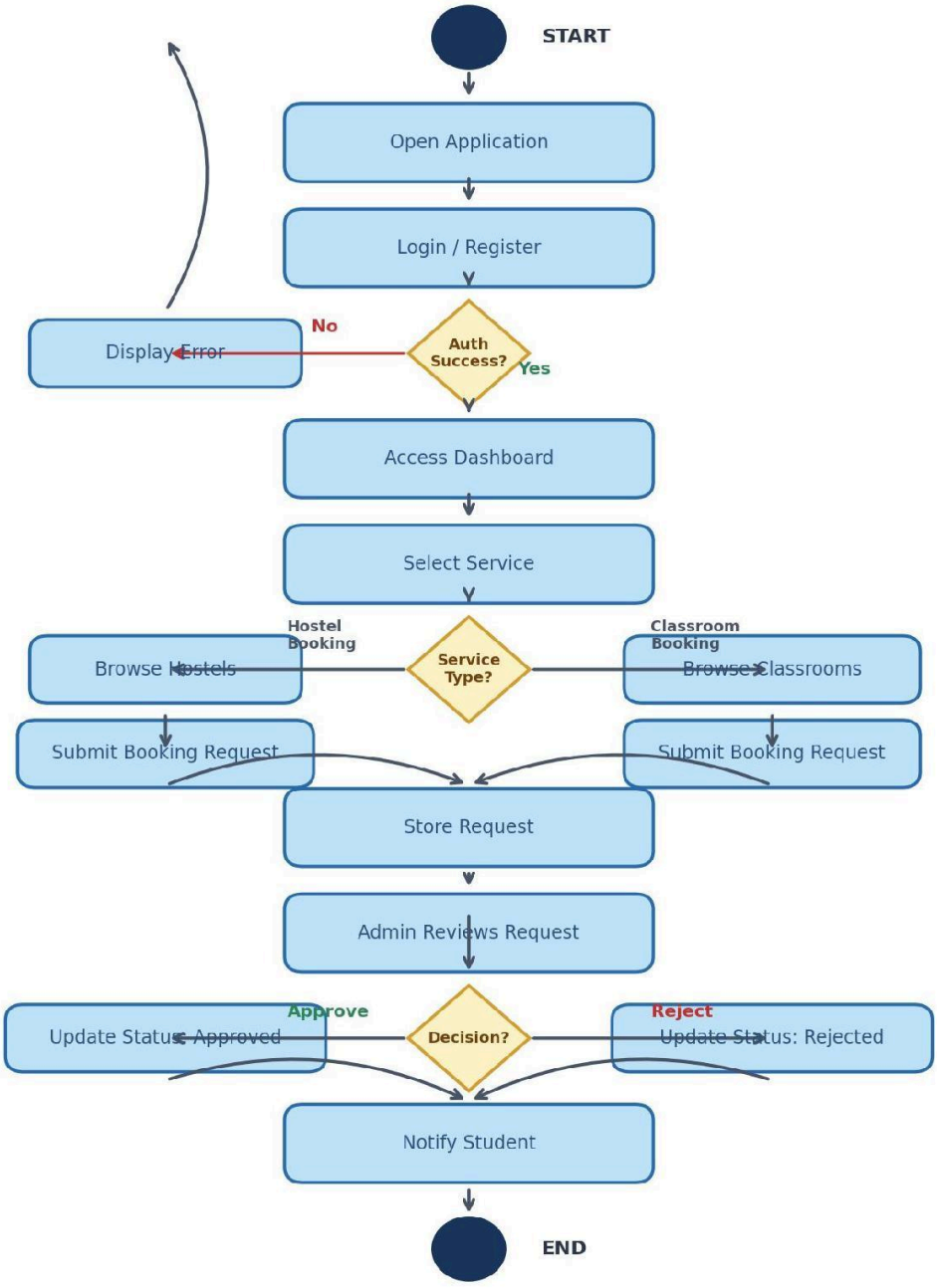
2.3.2 ACTIVITY DIAGRAM

The Activity Diagram illustrates how a student interacts with the system from login to booking completion.

Activity Diagram

ACTIVITY DIAGRAM

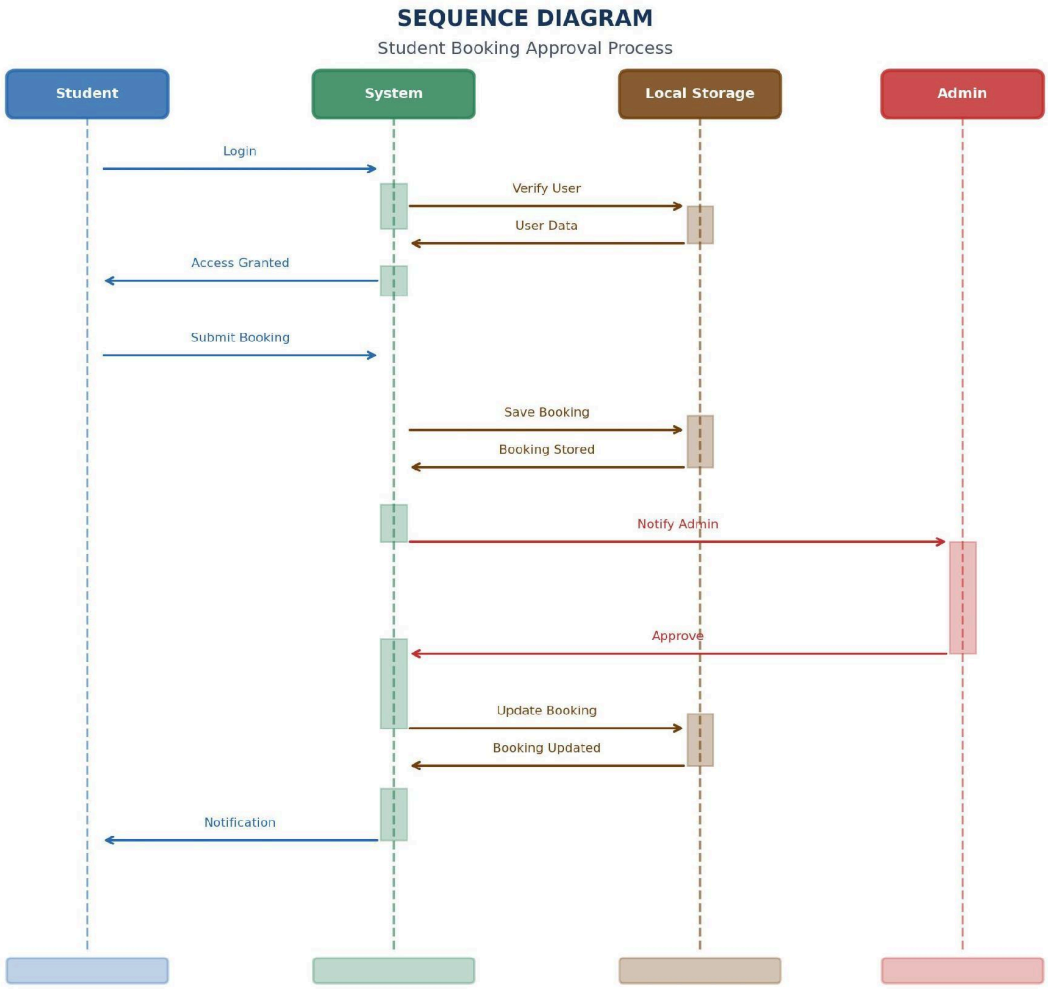
Student Booking Process



2.3.3 SEQUENCE DIAGRAM

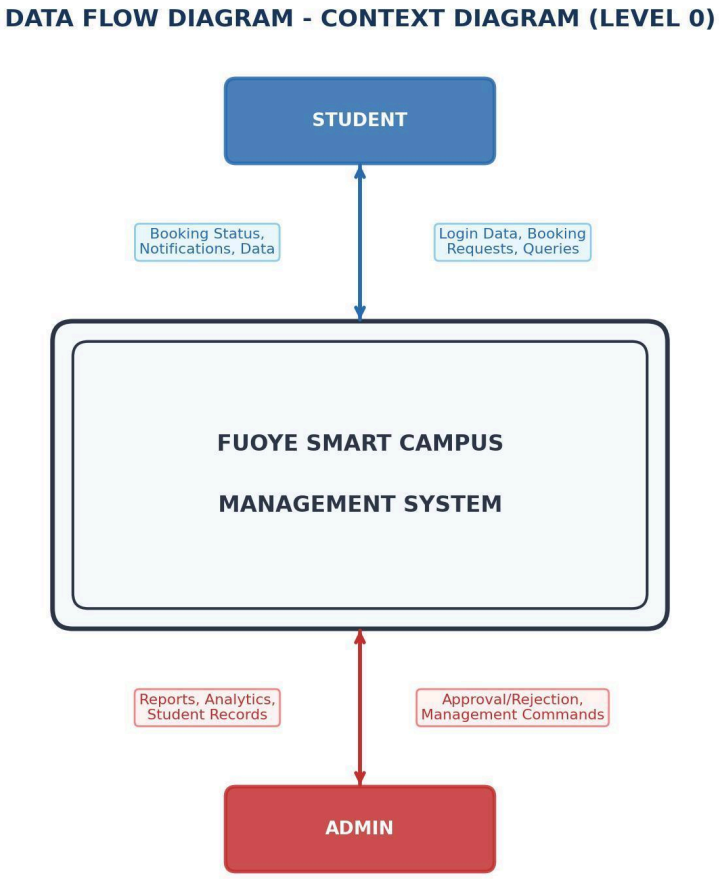
The Sequence Diagram shows the interaction between the Student, System, Local Storage, and Administrator.

Sequence Diagram



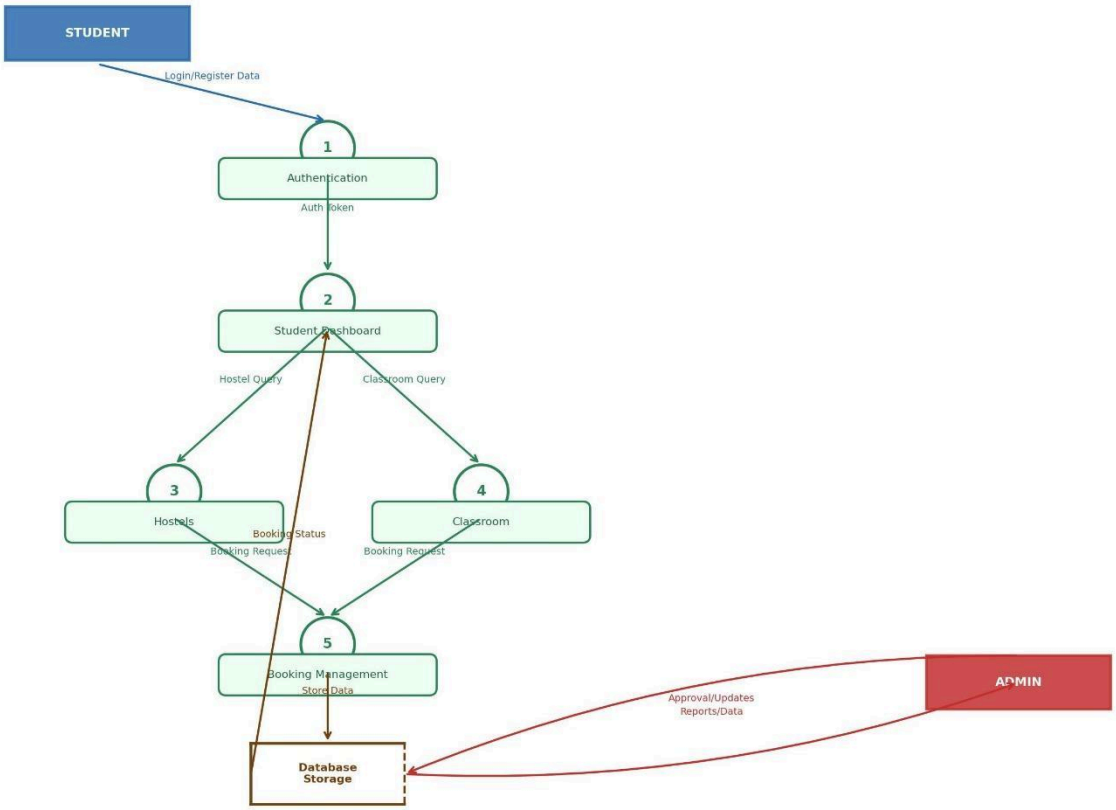
2.3.4 DATA FLOW DIAGRAM (DFD)

Context Diagram (Level 0)



2.3.4 DFD LEVEL 1 DIAGRAM

DATA FLOW DIAGRAM - LEVEL 1

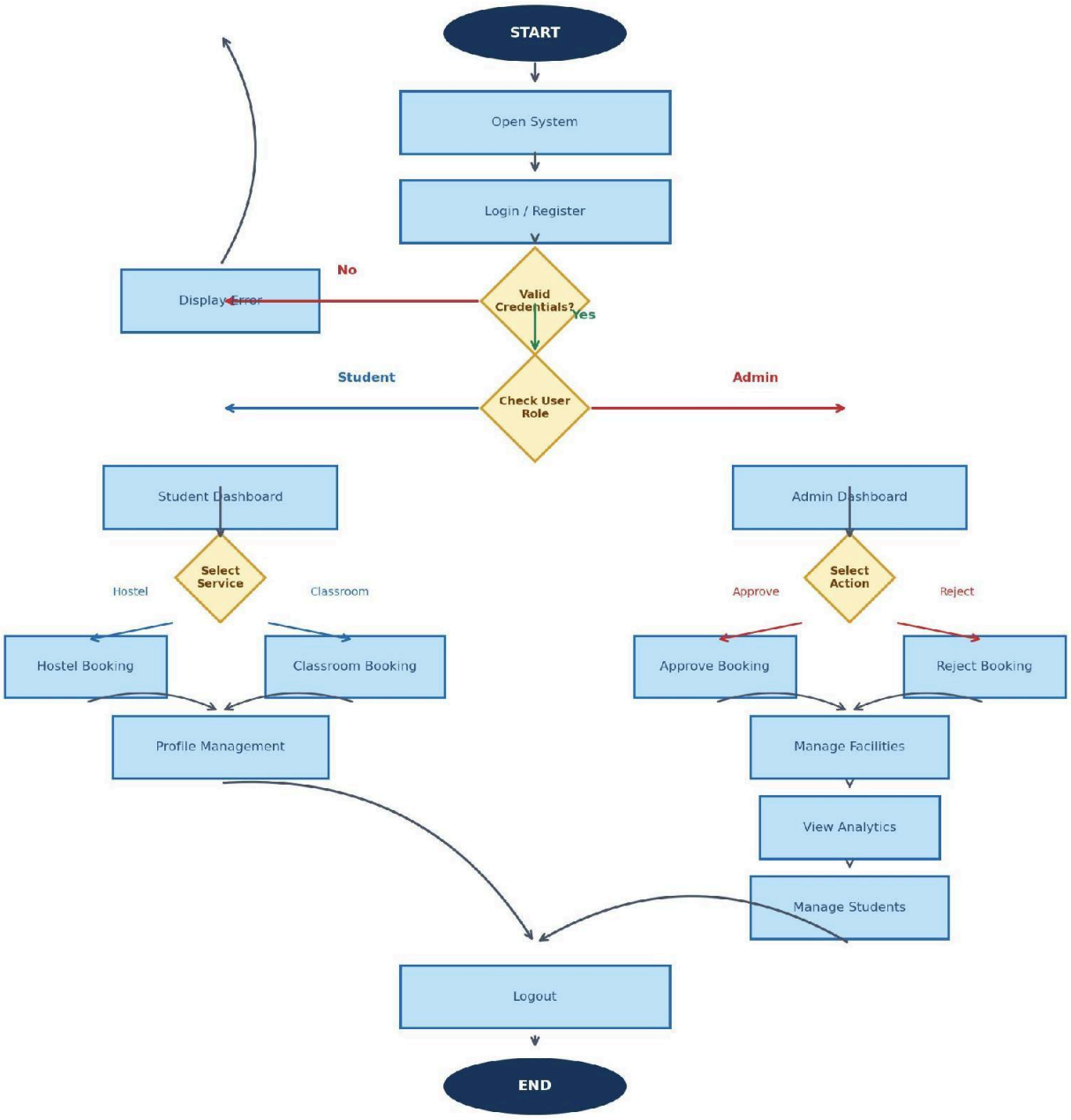


2.3.5 SYSTEM FLOWCHART

The flowchart illustrates the overall operational process of the system.

SYSTEM FLOWCHART

FUOYE Smart Campus Management System



CHAPTER THREE

3.0 IMPLEMENTATION TOOLS

Introduction

This chapter discusses the technologies, tools, software, and hardware components used in the design and implementation of the FUOYE Smart Campus Management System. It also presents the implementation process, system architecture, results obtained, and discussions on the performance of the developed system.

The system was developed as a modern web application using frontend technologies that provide responsiveness, scalability, maintainability, and an improved user experience.

3.1 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements

The development and testing of the system required a computer system with the following minimum specifications:

Component	Specification
Processor	Intel Core i3 or higher
RAM	4GB Minimum
Storage	20GB Available Space
Display	1366 × 768 Resolution
Keyboard	Standard Keyboard
Mouse	Standard Mouse
Internet	Required for Deployment

Development Hardware Used

Component	Specification
Processor	Intel Processor

RAM	64GB
Storage	SSD Storage
Operating System	Windows 11

Software Requirements

The following software tools were used during development:

Software	Purpose
Visual Studio Code	Source Code Development
React.js	Frontend Framework
Vite	Build Tool
Tailwind CSS	Styling Framework
JavaScript	Application Logic
React Router DOM	Navigation and Routing

Git	Version Control
GitHub	Source Code Hosting
Vercel	Application Deployment
Google Chrome	Testing Environment

Development Technologies

React.js

React.js was used for building reusable user interface components. It enabled the creation of modular pages and interactive user experiences.

Vite

Vite served as the development and build tool. It provided fast development startup times and efficient production builds.

Tailwind CSS

Tailwind CSS was used for designing responsive and modern user interfaces throughout the application.

React Router DOM

React Router DOM was used to implement client-side routing and page navigation.

Local Storage

Local Storage was utilized as a lightweight client-side database for storing user information, booking records, notifications, settings, and authentication data.

Git and GitHub

Git and GitHub were used for version control, project tracking, and source code management.

Vercel

Vercel was used for hosting and deployment of the completed application.

3.2 DESCRIPTION OF THE NEW SYSTEM, RESULTS AND DISCUSSION

System Architecture

The FUOYE Smart Campus Management System adopts a role-based architecture consisting of two major users:

Student

Students interact with the system to:

- Create accounts
- Login
- Browse hostels
- Book hostels
- Reserve classrooms
- View booking status
- Manage profiles
- Receive notifications

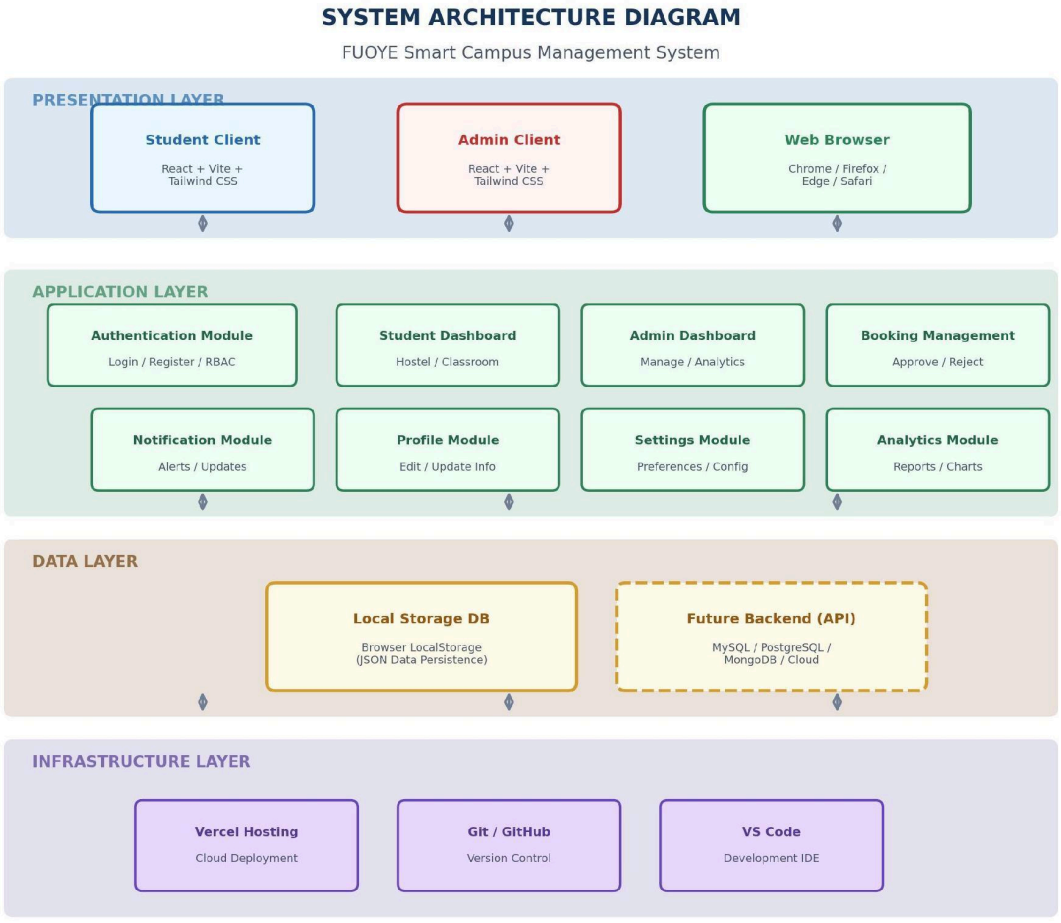
Administrator

Administrators interact with the system to:

- Manage hostels

- Manage classrooms
- View student activities
- Approve bookings
- Reject bookings
- Monitor analytics
- Manage settings

System Architecture Diagram



Authentication Module

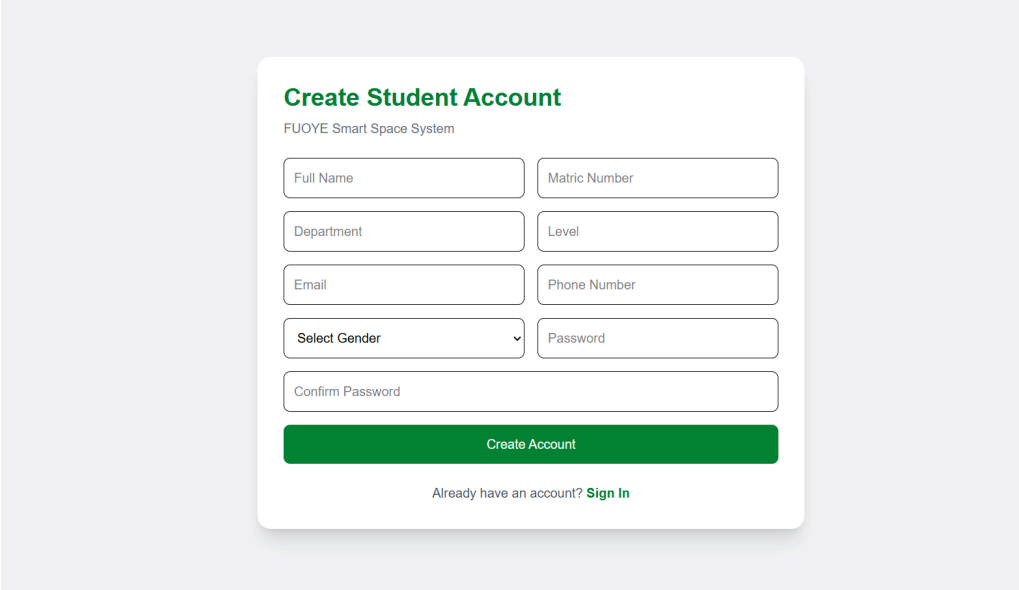
The authentication module controls user access within the system.

The implemented authentication features include:

Student Registration

Students can create accounts by providing:

- Full Name
- Matric Number
- Department
- Level
- Email Address
- Password
- Phone Number
- Gender



The system validates user entries before registration.

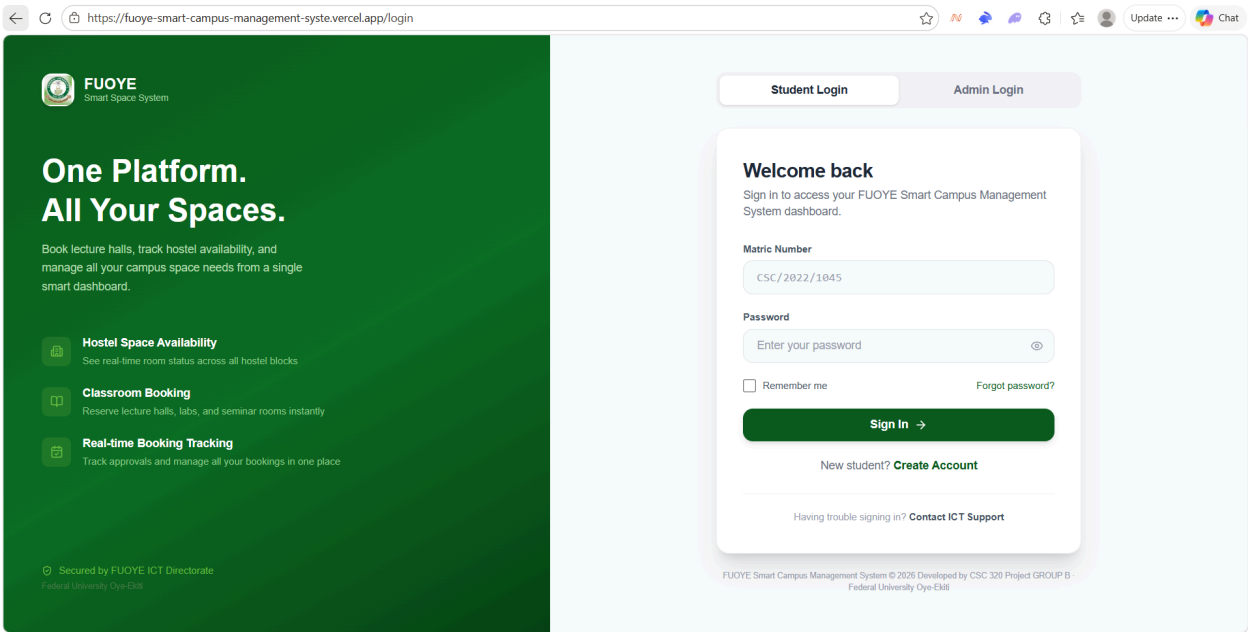
Duplicate matric numbers and email addresses are prevented.

After successful registration, the user is automatically logged into the system.

Student Login

Students log in using:

- Matric Number
- Password

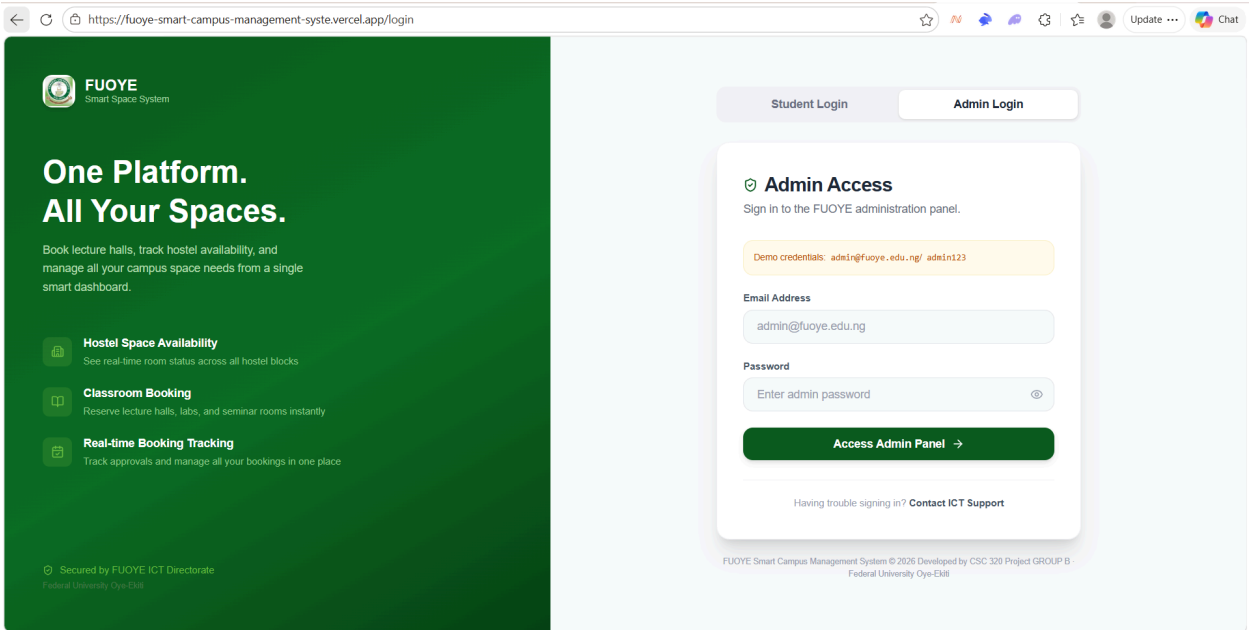


The system verifies credentials against stored records before granting access.

Administrator Login

Administrators access the platform through a dedicated login interface using predefined credentials.

Role validation ensures administrative pages remain inaccessible to students.



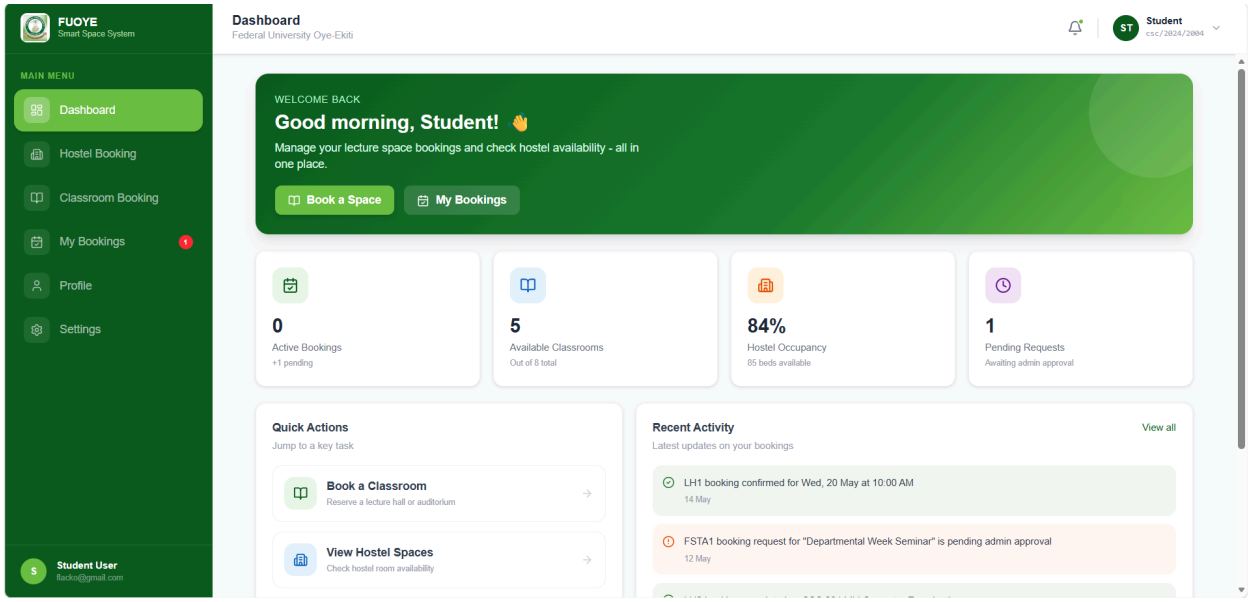
Student Dashboard Module

The student dashboard serves as the primary landing page after login.

The dashboard displays:

- Hostel statistics
- Classroom statistics
- Booking summaries
- Recent activities
- Notifications

The dashboard provides quick access to all major system functions.



Hostel Management Module

The hostel module enables students to:

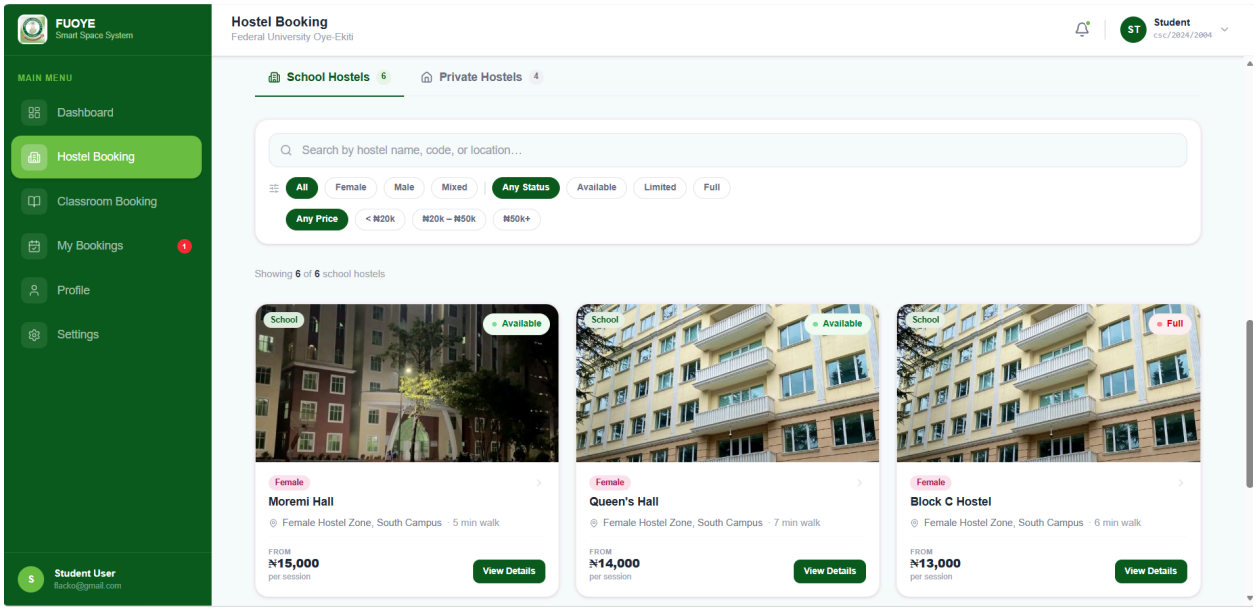
- View hostel listings
- Check room availability
- View hostel details
- Browse hostel images
- Submit hostel booking requests

Hostel information includes:

- Hostel name
- Hostel category
- Room capacity
- Occupancy status

- Availability count

The module supports both university-owned and private hostels.

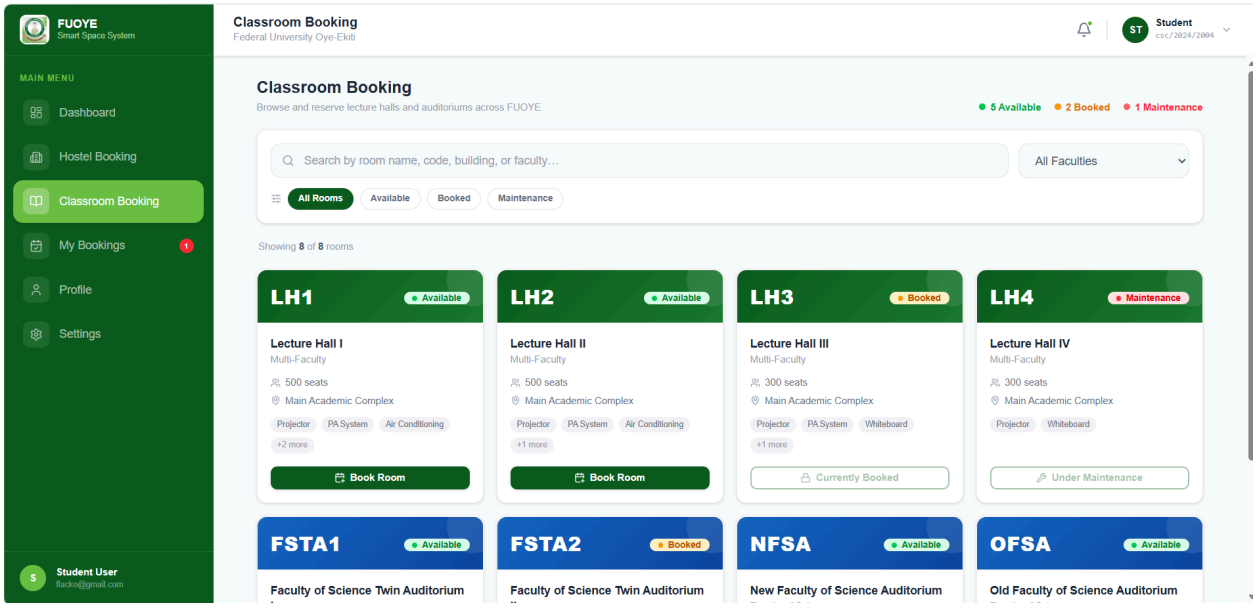


Classroom Reservation Module

The classroom reservation module allows students to:

- Browse available classrooms
- View classroom capacities
- Reserve academic spaces
- Track reservation status

The module minimizes scheduling conflicts by maintaining booking records.

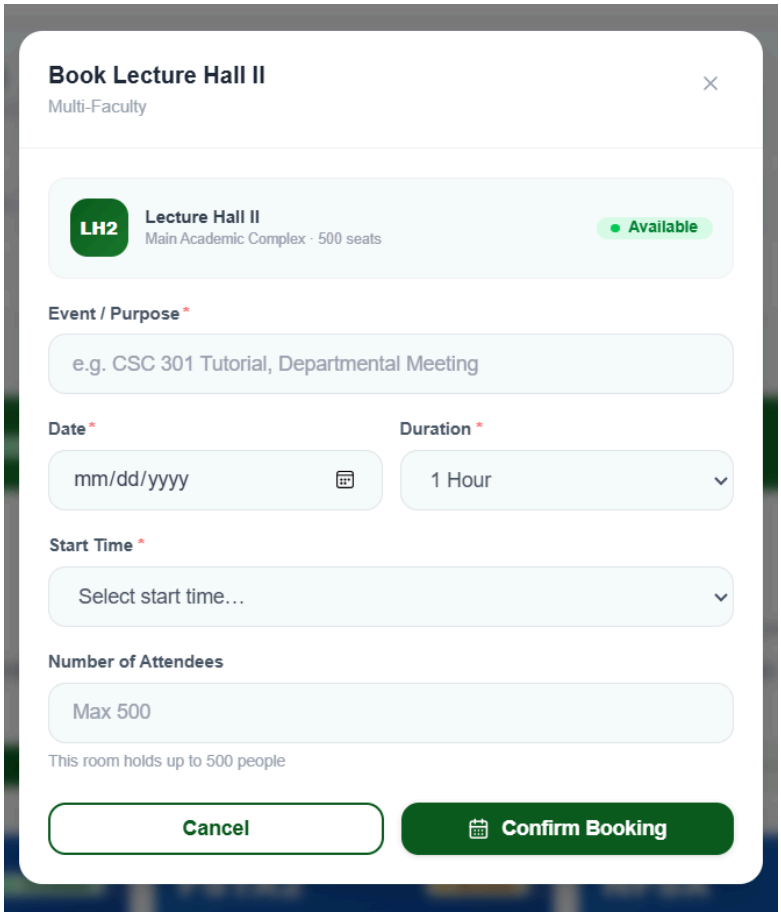


Booking Management Module

The booking module serves as the central component of the system.

The module performs the following functions:

- Creates booking requests
- Stores booking information
- Updates booking status
- Tracks booking history
- Supports cancellation



Booking Workflow

Student submits request.

↓

Booking stored as Pending.

↓

Administrator reviews request.

↓

Approved or Rejected.

↓

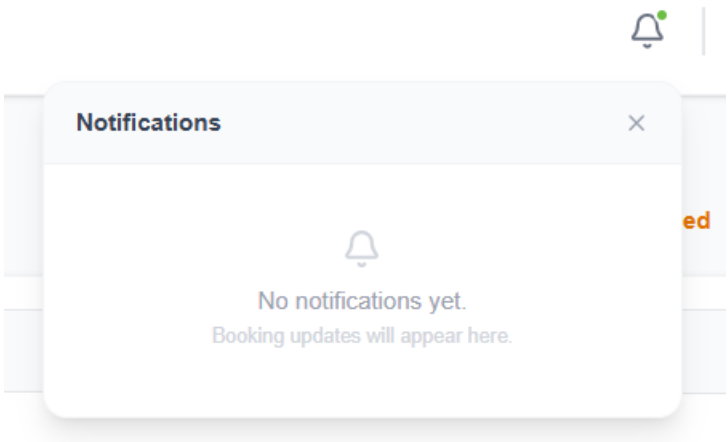
Student receives updated status.

Notification Module

The notification system provides users with updates regarding activities performed within the platform.

Notifications are generated for:

- New bookings
- Booking approvals
- Booking rejections
- System activities



The notification center improves communication between students and administrators.

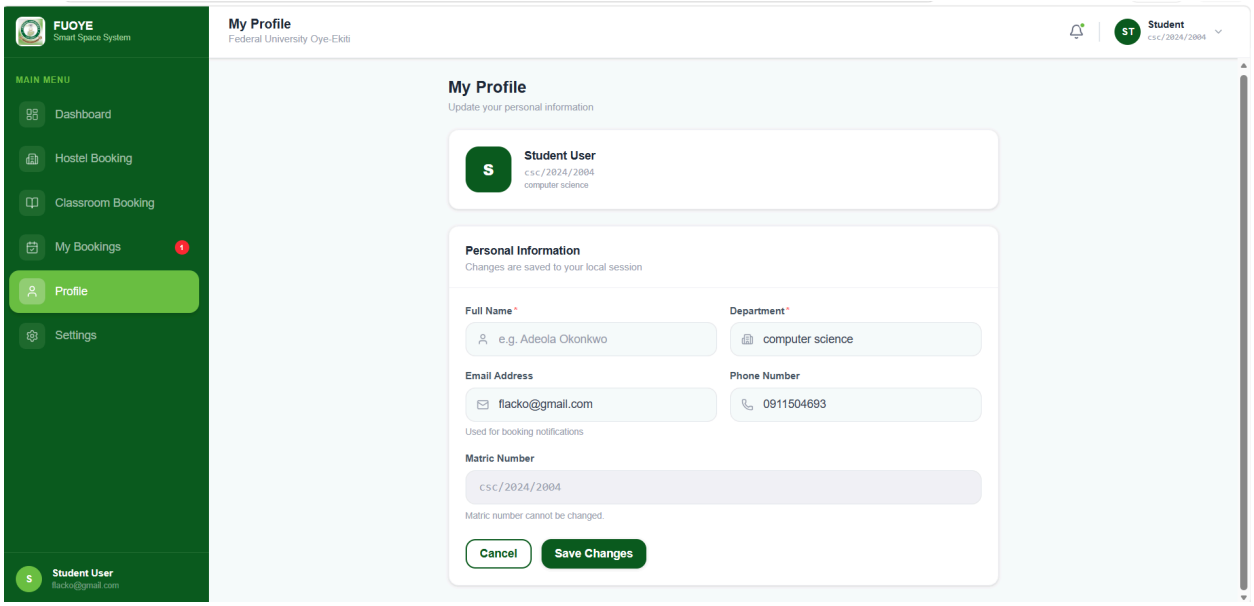
Profile Management Module

Students can manage personal information through the profile page.

Users can:

- View profile information
- Edit profile details
- Update account information

This feature improves personalization and user control.



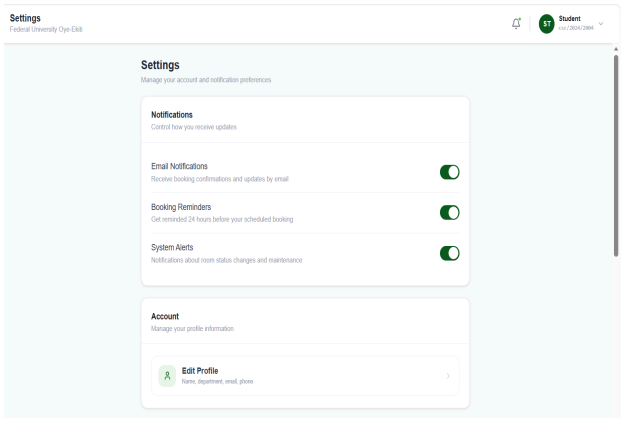
Settings Module

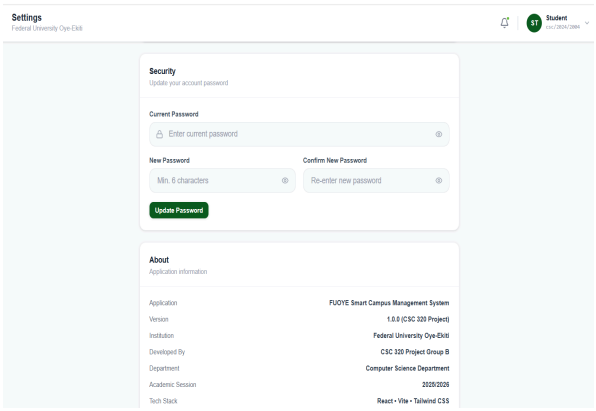
The settings module allows users to configure preferences.

Available settings include:

- Email notifications
- Booking reminders
- System alerts

The settings are stored locally and persist between sessions.





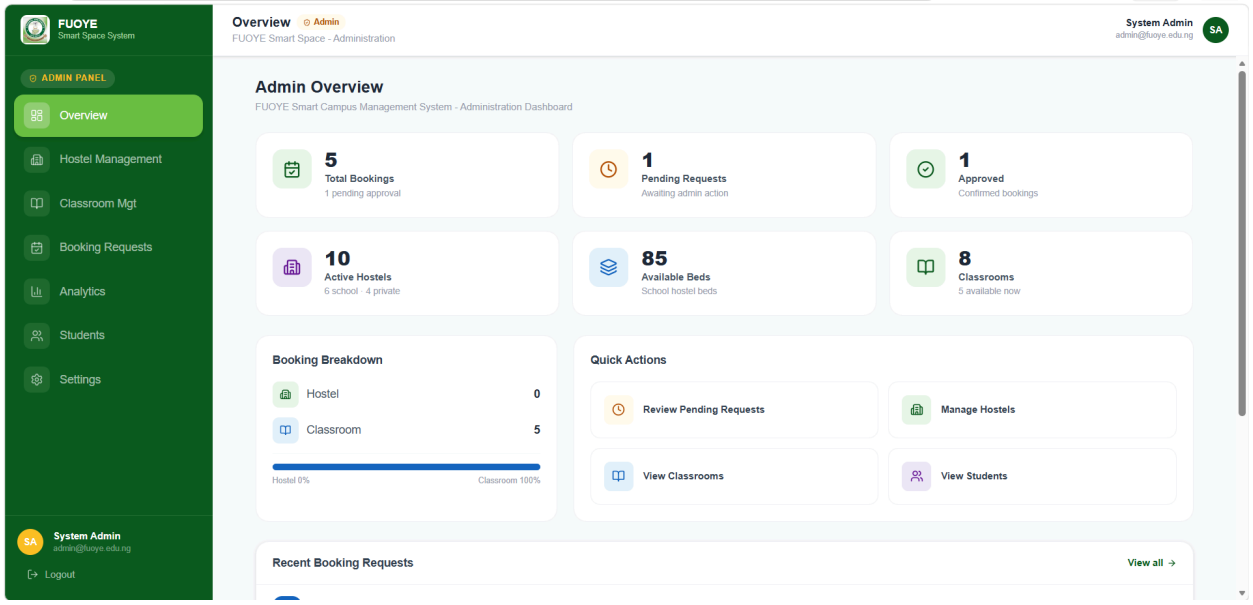
Administrative Dashboard

The administrative dashboard provides system-wide oversight.

Administrators can monitor:

- Total students
- Total bookings
- Pending approvals
- Facility occupancy

The dashboard serves as the command center of the application.

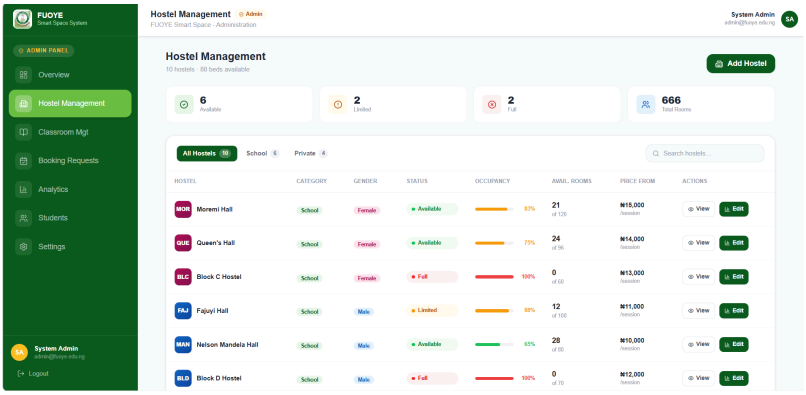


Hostel Administration

Administrators can:

- View hostel records
- Monitor occupancy rates
- Track availability status

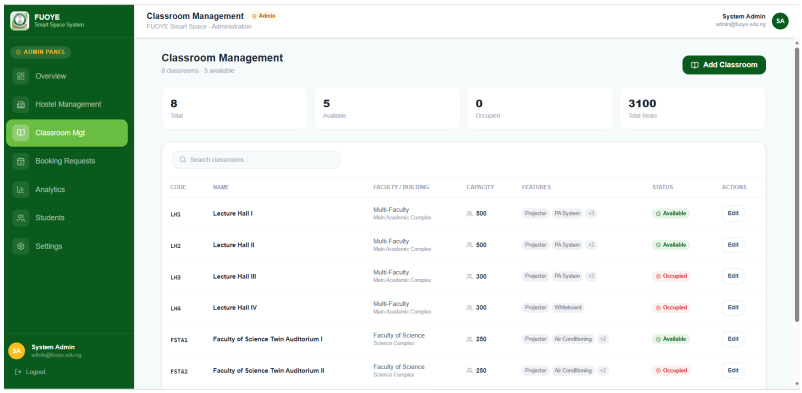
This helps improve accommodation management.



Classroom Administration

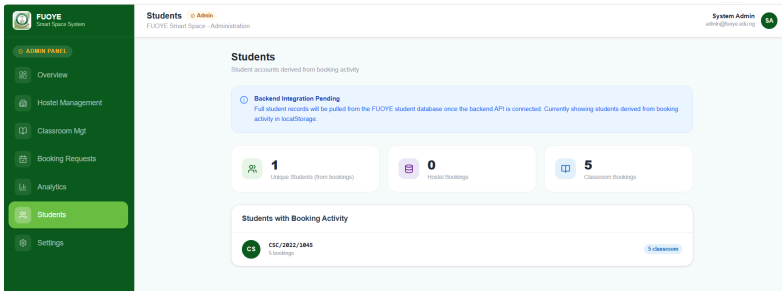
Administrators can manage classroom information and monitor reservations.

This ensures effective utilization of academic facilities.



Student Administration

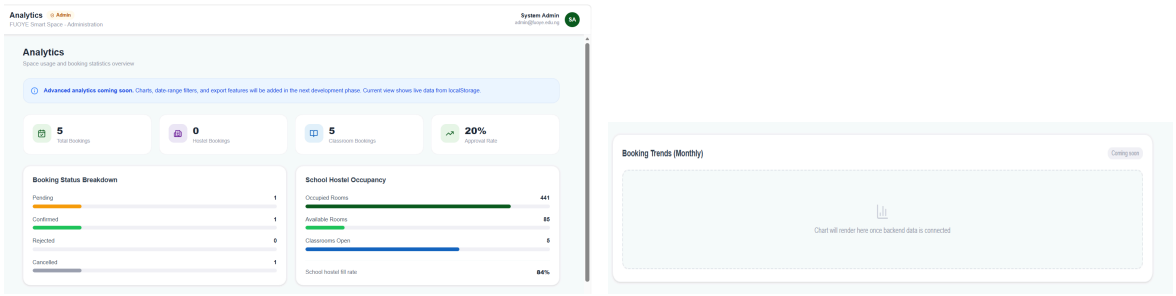
Administrators can view student activity records and monitor engagement within the platform.



Analytics Module

The analytics module provides visual summaries of:

- Booking statistics
- Occupancy rates
- Reservation trends
- Student activity



These insights assist administrative decision-making

Results and Discussion

The developed system successfully addressed the challenges identified in the existing manual process.

Testing demonstrated that students could successfully:

- Register accounts
- Login
- Browse hostels
- Submit bookings
- Reserve classrooms
- Track booking status

Administrative users could successfully:

- Manage facilities
- Review bookings
- Approve requests
- Monitor analytics

The implementation improved efficiency, reduced paperwork, enhanced accessibility, and provided a centralized management solution for campus facilities.

Live System

The developed FUOYE Smart Campus Management System has been deployed online to enable easy demonstration and evaluation without requiring local installation.

Live URL:

https://fuoye-smart-campus-management-syste.vercel.app/login

Source Code Repository:

https://github.com/flackodgoat/fuoye-smart-space-system

CHAPTER FOUR

4.0 CONCLUSION AND RECOMMENDATION

4.1 Conclusion

The FUOYE Smart Campus Management System was developed to address the challenges associated with the manual management of hostel accommodation and classroom reservations within the university environment.

The study examined the limitations of existing methods, which include time-consuming processes, poor record keeping, lack of real-time information, and difficulties encountered by students when attempting to access campus facilities. These challenges often result in inefficiency, delays, and poor utilization of available resources.

To overcome these problems, a web-based system was designed and implemented using modern frontend technologies including React, Vite, JavaScript, and Tailwind CSS. The system provides a centralized platform through which students can register accounts, access hostel information, reserve classrooms, track booking requests, manage personal profiles, and receive notifications. The system also provides administrators with tools for monitoring activities, managing facilities, approving bookings, and viewing analytical information.

The objectives of the study were successfully achieved. The developed system demonstrated the ability to improve accessibility, reduce paperwork, simplify facility management processes, and enhance the overall user experience for both students and administrators.

The implementation and testing of the system confirmed that the proposed solution is efficient, reliable, user-friendly, and suitable for deployment within an academic environment. The project therefore serves as a practical demonstration of how modern web technologies can be applied to solve real-world challenges in educational institutions.

4.2 Recommendation

Based on the findings and implementation of this project, the following recommendations are proposed for future improvements and deployment:

The university management should consider adopting a fully integrated digital campus management platform to replace manual processes associated with hostel allocation and classroom reservations.

Future versions of the system should include a backend database such as MySQL, PostgreSQL, or MongoDB to provide secure and permanent data storage beyond browser local storage.

A real-time notification system should be implemented using email services and push notification technologies to improve communication between students and administrators.

Integration with the university student portal should be considered to enable automatic verification of student records and authentication details.

Online payment functionality can be incorporated to allow students to make accommodation and facility-related payments directly through the platform.

Future versions should include mobile application support for Android and iOS devices to improve accessibility and convenience for users.

Advanced analytics and reporting features should be introduced to assist university management in making data-driven decisions regarding facility utilization and resource planning.

Security enhancements such as encrypted authentication, role-based permissions, audit logs, and cloud database integration should also be implemented to improve system reliability and scalability.

In conclusion, the FUOYE Smart Campus Management System has demonstrated significant potential as a modern solution for improving campus facility management and can serve as a foundation for a larger university-wide smart campus platform.

CHAPTER FIVE

5.0 REFERENCES

React Team. (2025). React Documentation. Retrieved from https://react.dev/

Vite Team. (2025). Vite Documentation. Retrieved from https://vitejs.dev/

Tailwind Labs. (2025). Tailwind CSS Documentation. Retrieved from https://tailwindcss.com/docs

React Router Team. (2025). React Router Documentation. Retrieved from https://reactrouter.com/

Lucide Contributors. (2025). Lucide React Documentation. Retrieved from https://lucide.dev/

Vercel Inc. (2025). Vercel Documentation. Retrieved from https://vercel.com/docs

Mozilla Developer Network. (2025). Web Storage API (localStorage). Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

Federal University Oye-Ekiti. (2025). Official Website. Retrieved from https://www.fuoye.edu.ng/

CHAPTER SIX : APPENDIX

Project Repository and Live Deployment

GitHub Repository:

https://github.com/flackodgoat/fuoye-smart-space-system

Live System:

https://fuoye-smart-campus-management-syste.vercel.app/login