A

Project Report On

# Hostel Management System

Submitted By :

Uday patadiya

BCA Semester – 5

Project Guide :

Heena Medam

Submitted To :



Geetanjali College Of Computer Science And Commerce ( B.B.A ).

Rajkot.

Academic Year : 2022-2023

## Acknowledgement

I Am Happy To Submit My Idea Of "Hostel Management System" In Saurashtra University, Rajkot For BCA Degree In Computer Branch.

I Also Grateful To Prof. Brijesh Shah Head of The Department And All The Faculty Members Of The Department Of Computer Science For Their Kind Support Through Out This Journey.

I Take The Privilege To Acknowledge The Elite Authors Of Numerous Books And Papers And Blogs Which We Have Referred During Progress Of This.

I Also Say The Big Thank You To My Parents For Such A Support And Without Them I Can Do Nothing Not In Just Project But Also In Life. Thankful To My Family For Their Support.

The Feeling Of Gratefulness To Any One's Help Directly Arises From The Bottom Of Heart. A Small But An Important And Timely Help Can Prove To Be A Milestone In One's Life.

Very Thankful To Almighty Of All Of "God" To Give Me Such A Best Persons And All The Thing He Provides Before I Need And I Always Feel That Without Him I Are Nothing.

# Index

**Project Profile ......................................................................................................... 1**

**Software Development Life Cycle ........................................................................ 2**

**System Requirement (For Development) ............................................................. 9**

**About the Tool (Tools & Technology used) ......................................................... 10**

**DFD (Data Flow Diagram) .................................................................................... 15**

**ER (Entity Relationship Diagram) ....................................................................... 17**

**Data Dictionary. ..................................................................................................... 18**

**Screen Shots ............................................................................................................ 20**

**Test Case ................................................................................................................. 33**

**Limitation ................................................................................................................35**

**Bibliography............................................................................................................ 36**

# Project Profile

* Project Title : Hoste Management System

* Development Software : Net bens

* Front End : Java jframe , java aws , java swing

* Backend : my sql Database

* Academic Year : 2022-2025

* Developed By : uday patadiya

* Submitted To : Geetanjali College

* Documentation Tool : Microsoft Word

* Operating System : windows
* Language : Java Application

## System Development Life Cycle

**1. Planning**

* **Define Requirements**: Gather requirements from stakeholders (e.g., hostel managers, students).
* **Feasibility Study**: Assess technical, operational, and economic feasibility.
* **Project Scope**: Define what features will be included, such as:
  + Room allocation
  + Fee management
  + Student registration

**2. Analysis**

* **System Requirements Specification**: Document functional and non-functional requirements.
* **Use Case Diagrams**: Create diagrams to visualize interactions.
* **Data Flow Diagrams (DFDs)**: Illustrate how data moves through the system.

**3. Design**

* **Architecture Design**: Choose a layered architecture (presentation, business logic, data access).
* **UI/UX Design**: Design the user interface using Java Swing components (JFrame, JPanel, JButton, etc.).
* **Database Design**: Design the database schema (e.g., tables for students, rooms, bookings) and choose an AWS service (e.g., Amazon RDS for relational databases).
* **Integration Design**: Plan integration with AWS services (e.g., S3 for file storage, Lambda for serverless functions).

**4. Implementation**

* **Set Up Development Environment**:
  + Install Java Development Kit (JDK).
  + Set up an IDE (e.g., IntelliJ IDEA, Eclipse).
  + Integrate AWS SDK for Java.
* **Code Development**:
  + Develop the front-end using Java Swing.
  + Implement back-end logic (e.g., CRUD operations for managing hostel data).
  + Use AWS SDK to interact with AWS services (e.g., uploading documents to S3).
* **Version Control**: Use Git for version control.

**5. Testing**

* **Unit Testing**: Write tests for individual components (e.g., models, controllers).
* **Integration Testing**: Test the integration of Java Swing with AWS components.
* **User Acceptance Testing (UAT)**: Gather feedback from users to ensure the system meets requirements.
* **Performance Testing**: Ensure the application performs well under expected loads.

**6. Deployment**

* **Deploy on AWS**: Use AWS Elastic Beanstalk or EC2 for deploying your Java application.
* **Database Setup**: Configure Amazon RDS and migrate your database schema and initial data.
* **Security Configuration**: Set up IAM roles, security groups, and VPCs to secure the application.

**7. Maintenance**

* **Monitor Performance**: Use AWS CloudWatch for monitoring application performance and health.
* **Bug Fixes and Updates**: Regularly update the application for bug fixes and feature enhancements.
* **User Support**: Provide documentation and support for users.

**8. Documentation**

* **Technical Documentation**: Create documentation for developers covering architecture, setup, and code standards.
* **User Documentation**: Provide user manuals and help guides for end-users.

**Example Technologies**

* **Java Swing**: For creating the desktop application interface.
* **AWS RDS**: For managing the database.
* **AWS S3**: For storing documents and images (e.g., student IDs, payment receipts).
* **AWS Lambda**: For implementing serverless back-end logic (if needed).
* **AWS IAM**: For managing permissions and security.

Steps :

1 - Requirement gathering

1. - Project Planning
2. - Design
3. - Coding & Implementation
4. - Testing

### 1 - Requirement Gathering

Any application development process must include the requirement gathering stage. After choosing the project topic, thorough research is essential to identify all criteria needed to construct that specific project. For my Hostel Management System, I began gathering requirements by examining existing hostel management solutions and related applications to understand the necessary functionalities and layout designs.

The features needed in the application for users (administrators) are as follows:

1. **Login**: This is the main feature that allows administrators to securely access the system using their credentials.
2. **Home**: The home screen provides an overview of the system, displaying key metrics and quick access to various functionalities.
3. **Manage Rooms**: This feature allows administrators to manage room assignments, availability, and maintenance status for each room in the hostel.
4. **New Student**: This functionality enables the addition of new student records, capturing essential information such as personal details and room assignments.
5. **Update & Delete Student**: Administrators can update student information or remove records for students who have left the hostel.
6. **Student Fees**: This feature manages the tracking and processing of student fees, including payment history and outstanding balances.
7. **All Students Living**: This section provides a comprehensive list of all currently residing students, along with their room details.
8. **Leave Students**: This feature tracks students who have left the hostel, enabling administrators to manage records efficiently.

### 2 - Project Planning

After completing the requirement gathering phase, the next step is to create a project plan. This involves determining the various project modules and selecting the optimal technologies for development.

For the Hostel Management System, the project will be divided into several modules based on the identified features. The front-end technology will utilize Java Swing to create a user-friendly desktop interface for the application, allowing administrators to interact seamlessly with the system. Java will serve as the programming language for all operations related to user interactions, ensuring efficient handling of data and processes.

On the back-end, we will utilize AWS services, specifically Amazon RDS for database management to store student and room data securely. This choice provides a robust and scalable solution for data storage. Additionally, AWS IAM will be used to manage security and access controls, ensuring that only authorized personnel can access sensitive information. By leveraging these technologies, the Hostel Management System aims to deliver a comprehensive, secure, and efficient management tool for hostel administration.

### 3 - Design

This Process Can Be Broken Down Into Two Parts: The Preliminary Design And The Final Design. The Project's Basic Layout Is Generated On Paper Or Using Any Design Program During The Preliminary Design Phase, And The Development Team Then Evaluates The Design's Viability.

The Second Phase Is Final Design, In Which The Project's Final Or Nearly Accurate Design Is Constructed After Earlier Concepts Have Been Evaluated For Practicality And Flaws Have Been Identified.

### 4 - Coding & Implementation

The actual application is coded in the chosen programming language following the collection of all requirements, customer approval of the design, and feasibility assessment of the project. This phase is regarded as the longest in the System Development Life Cycle (SDLC) and requires various tools, including IDEs, databases, and libraries.

For the development of the Hostel Management System, I selected IntelliJ IDEA as the IDE for Java, which will facilitate the creation of the desktop application for managing hostel operations. The application will use MySQL as the database management system, providing a reliable way to store and manage all data related to students, rooms, and fees. Adhering to established coding standards is crucial to maintain clarity and consistency throughout the coding phase. The coding standards include:

1. **Variable Naming**: Variable names should reflect their purpose; for example, a variable related to room management should include "Room" in its name.
2. **Array Naming**: Suffixes should be added to arrays for clarity (e.g., studentArray).
3. **Data Class Naming**: Data classes should be named based solely on their functionality (e.g., Student, Room).
4. **Adapter Class Naming**: Classes that serve as adapters should include the suffix "Adapter" (e.g., StudentAdapter).
5. **Camel Case Naming**: All variables and files must follow camel case conventions for consistency.
6. **Function Naming**: Functions used for data retrieval should begin with the prefix "get" (e.g., getStudentDetails).

The application development is divided into two phases:

1. **Database Setup**: The database is structured based on the design specifications and will use MySQL to manage relationships between the data for the desktop application. This includes creating tables for students, rooms, fees, and other necessary entities.
2. **Application Development**: This phase involves translating the design into Java code for the desktop application, incorporating the functionalities required for managing hostel operations effectively.

### 5 - Testing

Once the coding and implementation phases are complete, the application is ready to be tested for usability and to ensure it operates as intended. The testing phase is crucial for identifying any issues or bugs that may affect functionality and user experience.

To facilitate this process, I shared the Hostel Management System with colleagues to verify its features and gather feedback on user-friendliness. This collaborative approach allows for diverse perspectives, helping to identify areas for improvement. If any functionality does not perform as expected, the SDLC process is revisited, addressing the identified issues and refining the application until it meets the required standards. This iterative process ensures that the final product is robust, reliable, and user-centric.

## System Requirement Specifications

To Develop This Project , The Following System Hardware And Network Are Required :

Minimum Hardware Requirement :

For Application :

|  |  |
| --- | --- |
| Operating System | Windows 7 |
| CPU/Processor | Intel i3 |
| Ram | 4 GB |

Software Requirements :

|  |  |
| --- | --- |
| Services Required | laragon, Xampp |

## About The Tools& Technologies

**NetBeans**:  
 NetBeans is an open-source integrated development environment (IDE) that is widely used for Java development. It provides a comprehensive suite of tools for developing Java applications, making it easier to write, debug, and manage code. NetBeans supports various Java technologies, including Java SE, Java EE, and JavaFX. Its user-friendly interface and built-in support for version control systems enhance productivity for developers. Additionally, it offers features such as code templates, refactoring tools, and easy integration with other technologies, making it an ideal choice for developing the Hostel Management System.

**Java**:  
 Java is a class-based, object-oriented programming language designed to have as few implementation dependencies as possible. It enables application developers to write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without needing recompilation. First released in 1995, Java is widely used for developing applications across desktop, web, and mobile devices. Known for its simplicity, robustness, and security features, Java is a popular choice for enterprise-level applications. It allows for the creation of reusable code and modular programs, making it highly efficient for large-scale systems like the Hostel Management System.

**XAMPP**:  
 XAMPP is a free and open-source cross-platform web server solution stack package that contains the Apache HTTP Server, MySQL (now MariaDB), PHP, and Perl. It is designed to provide a reliable local server environment for developers. XAMPP allows for easy installation and configuration, enabling the testing of web applications locally before deployment. For the Hostel Management System, XAMPP serves as the backend server for the PHP-based admin panel, facilitating smooth database interactions and web functionalities.

**MySQL**:  
 MySQL is a widely used relational database management system that allows for the efficient storage and retrieval of data. It is known for its reliability, robustness, and ease of use, making it suitable for various applications, including the Hostel Management System. MySQL will handle all data-related tasks, including managing student records, room allocations, and fee transactions.

**MS Word**:  
 Microsoft Word is a versatile word processing application that allows users to create documents, resumes, contracts, and more. It is commonly used for drafting project documentation, reports, and user manuals for the Hostel Management System.

## Data Flow Diagram

0 Level Diiagram :

**New Student verification**

Admin

Student

**Allotment login Allotment**

**1 Level Diagram :**

Confirm Alloment

Allotment

Application Process

Verification

Admin

Student

ER Diagram :



## Data Dictionary

**Fees Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no | Field | Data Type | Size | Constraint |
| 1 | MobileNo | Int | 10 | Primary Key |
| 2 | month | Varchar | 50 | - |
| 3 | Amount | Int | 5 | - |

**Room Table:**

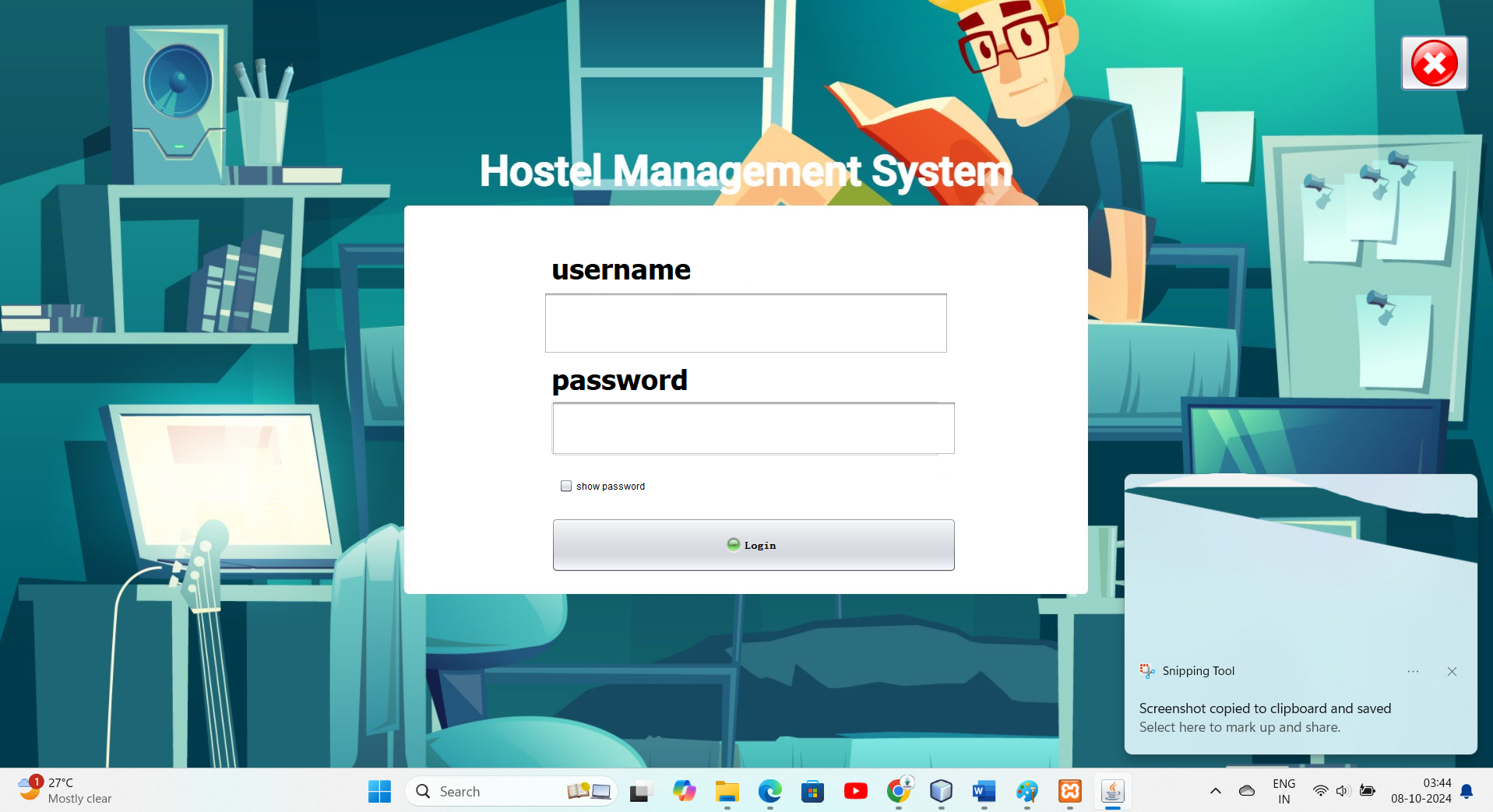
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no | Field | Data Type | Size | Constraint |
| 1 | number | int | 10 | Primary Key |
| 2 | Activate | varchar | 50 | Foreign Key |
| 3 | roomstatus | varchar | 50 | - |

**Student Table :**

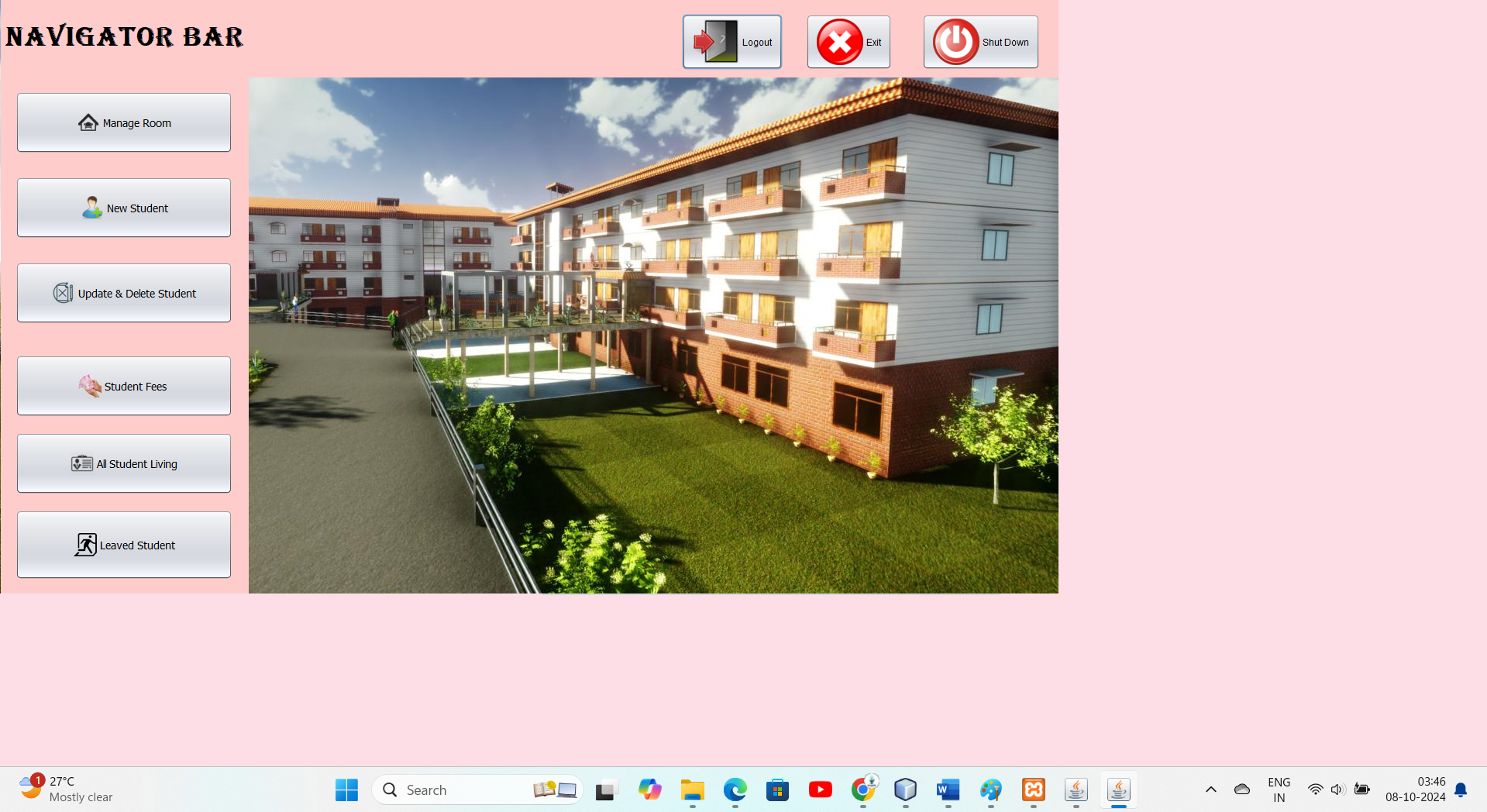
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr no | Field | Data Type | Size | Constraint |
| 1 | mobileNo | int | 10 | Primary Key |
| 2 | Name | varchar | 50 | - |
| 3 | father | varchar | 50 | - |
| 4 | mother | varchar | 50 | - |
| 5 | Email | varchar | 50 | - |
| 6 | Address | varchar | 150 | - |
| 7 | Collage | varchar | 100 | - |
| 8 | Aadhaar | Int | 12 | Primary Key |
| 9 | roomNo | Int | 5 | - |
| 10 | status | varchar | 10 | - |

## Screen Shots

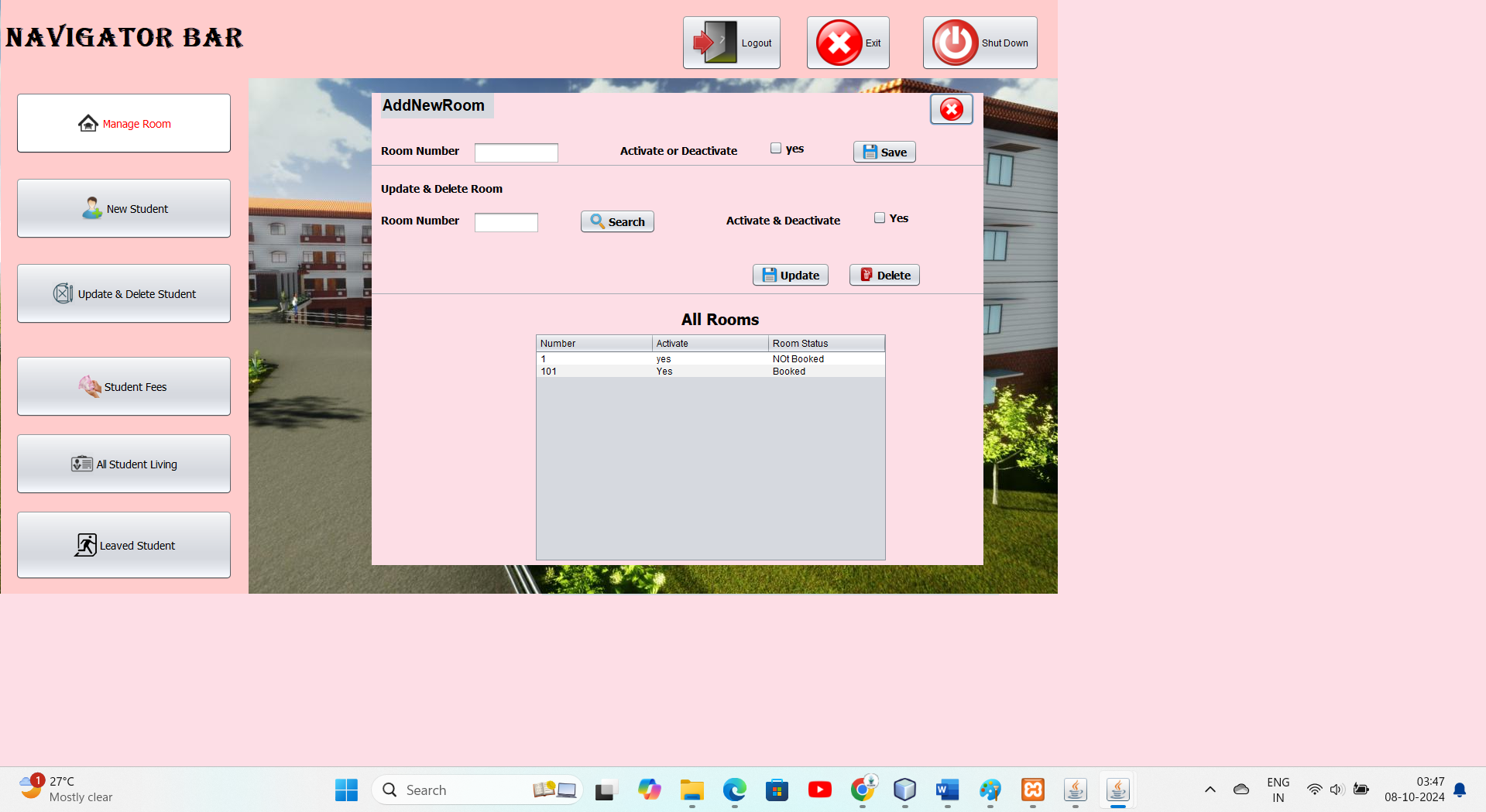
**Login page:**



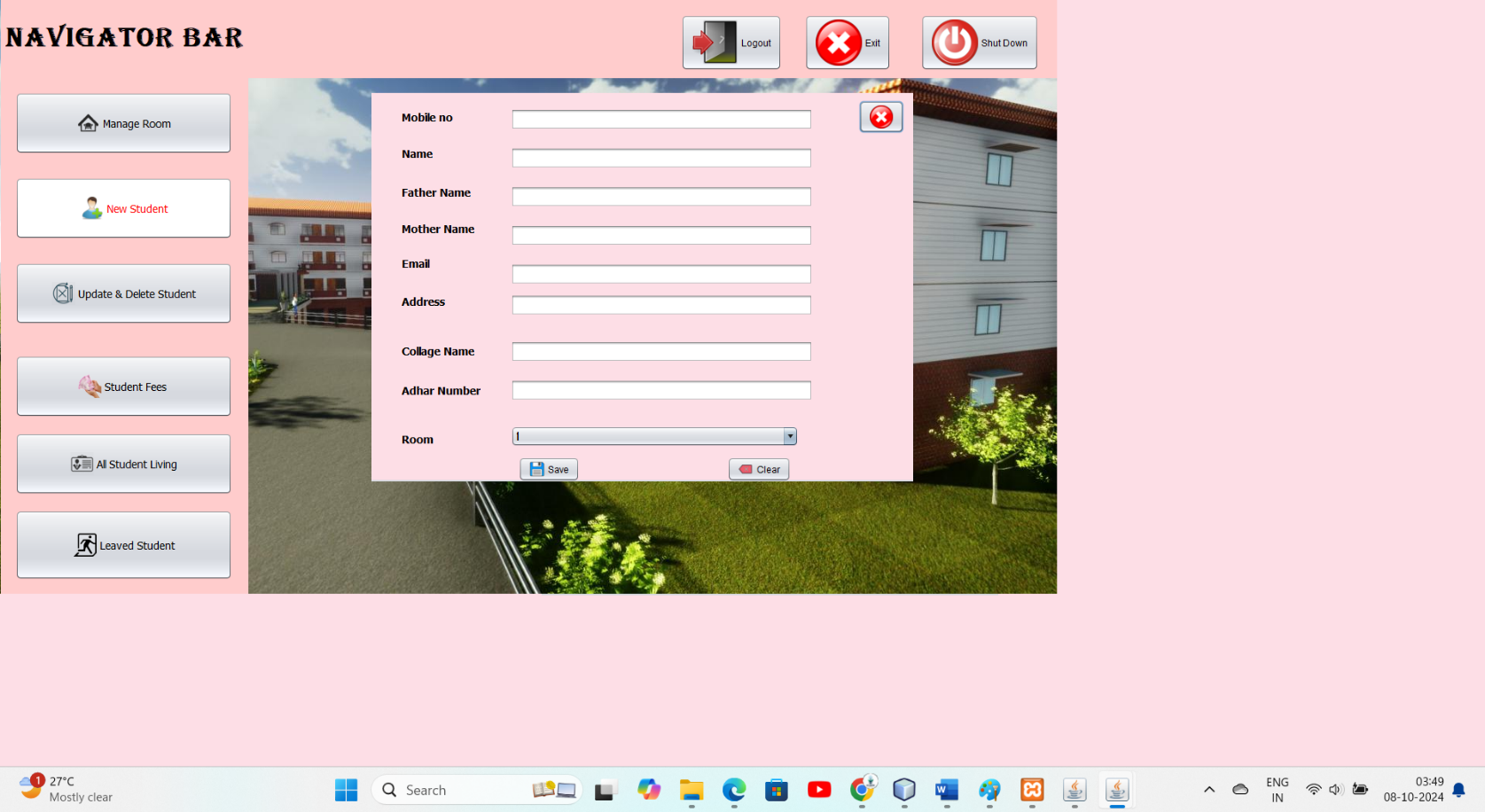
**Home Page :**



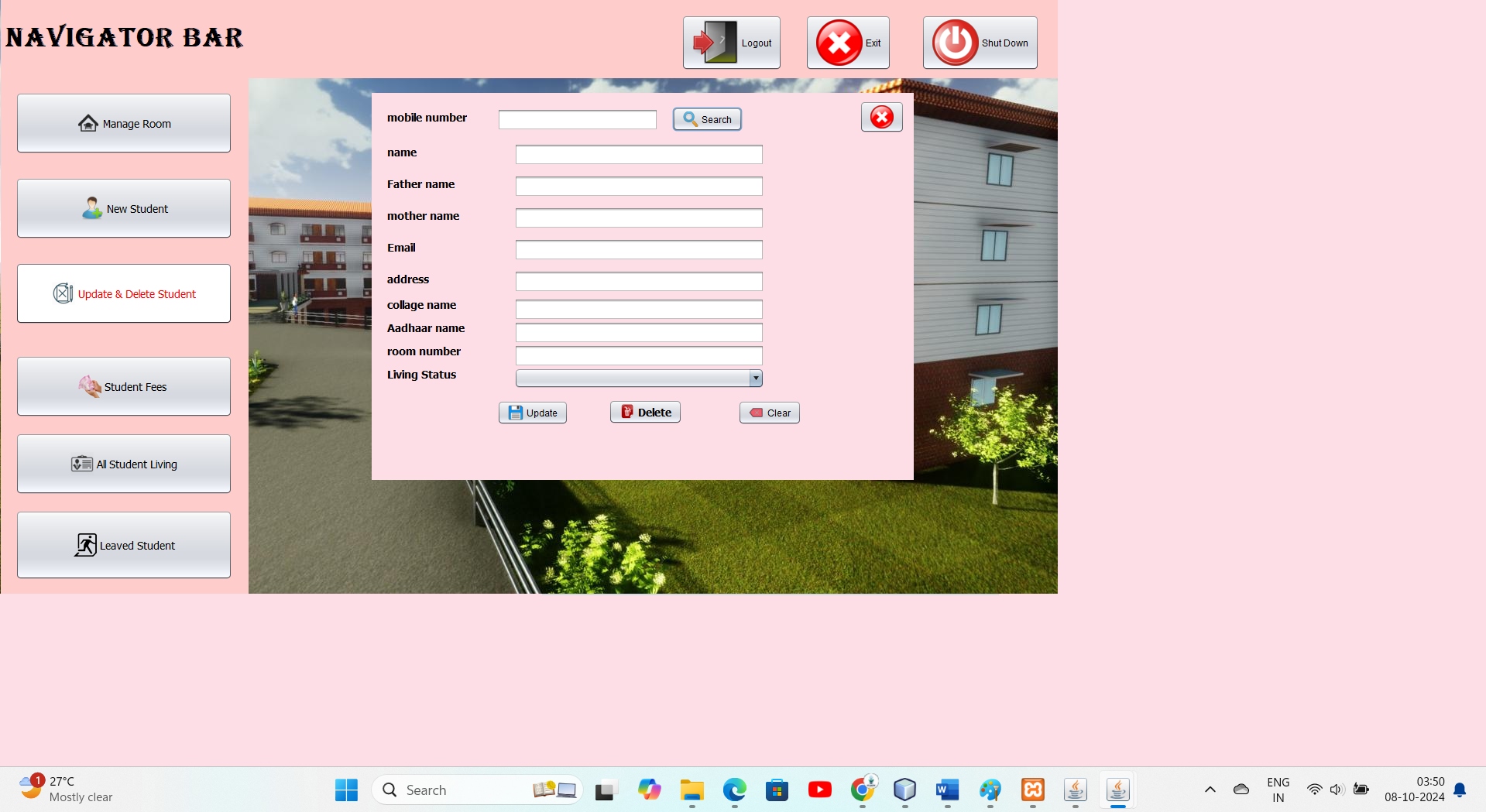
**Manage Room :**

****

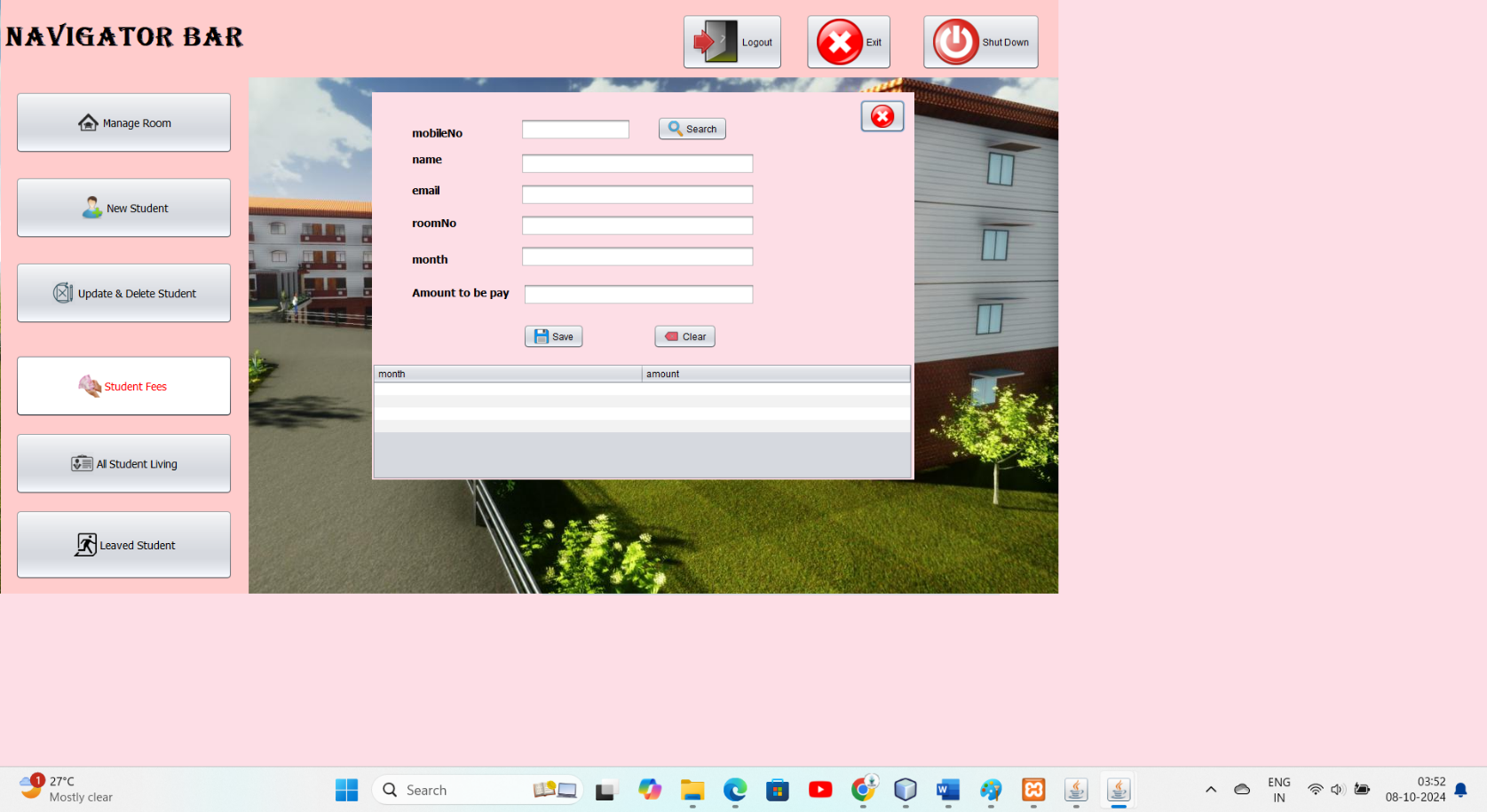
**New Student :**

****

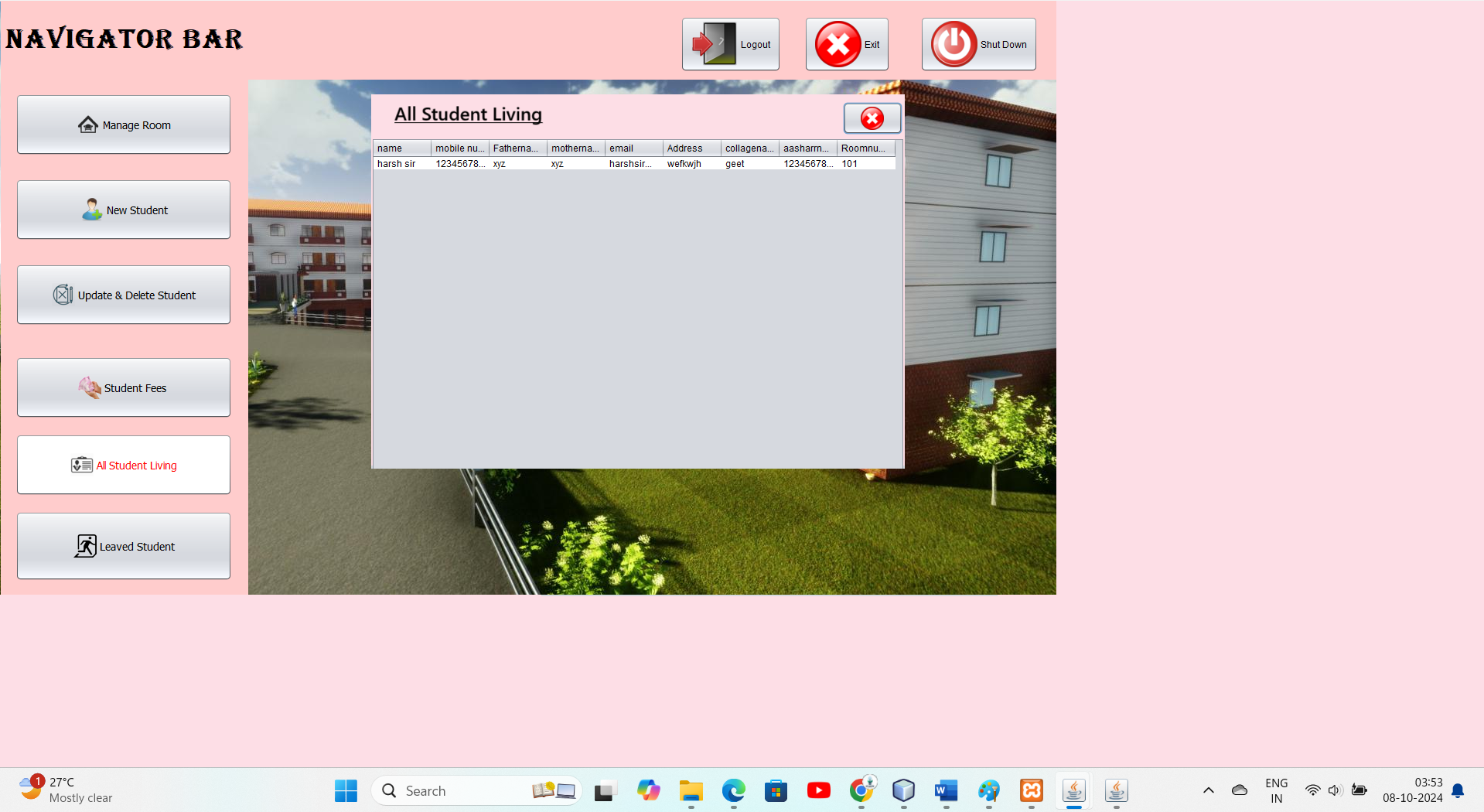
**Update & Delete Student :**

****

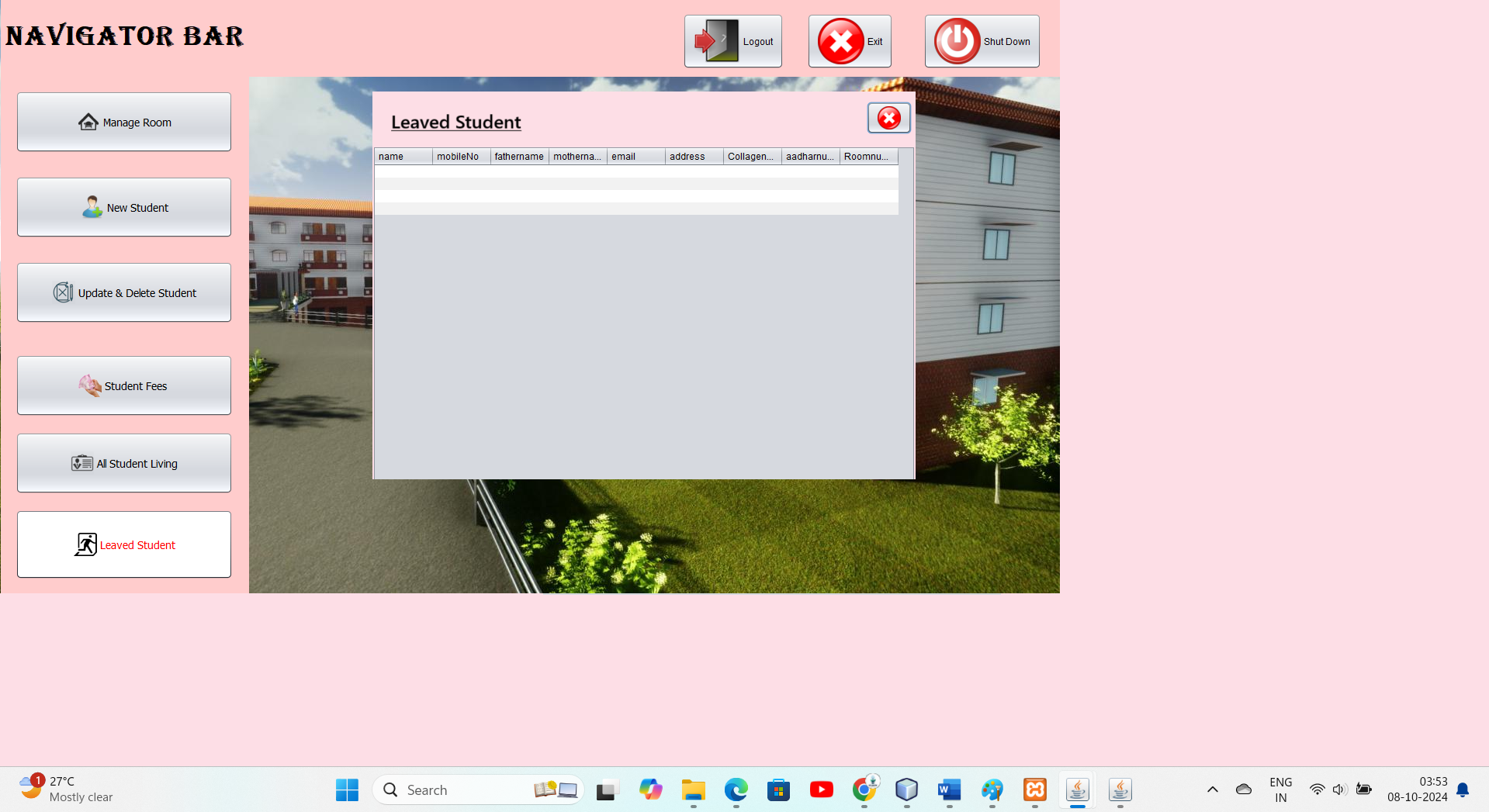
**Student fee :**

****

**All Student Living :**

****

**Leaved Student :**

****

**Future Requirements of the Project**

As the Hostel Management System evolves, several future enhancements and requirements can be anticipated to improve functionality, user experience, and overall efficiency. These include:

1. **Mobile Application Development**: To extend accessibility, a mobile application could be developed for both Android and iOS platforms, allowing students and administrators to manage bookings, payments, and maintenance requests on-the-go.
2. **Enhanced Reporting and Analytics**: Integrating advanced reporting tools will enable administrators to generate detailed insights into occupancy rates, financial performance, and student demographics. This could aid in better decision-making and resource allocation.
3. **User Feedback System**: Implementing a feedback mechanism would allow students to share their experiences and suggestions, helping improve services and addressing concerns promptly.
4. **Integration with Payment Gateways**: Adding support for various online payment gateways (e.g., PayPal, Stripe) would facilitate seamless fee transactions, enhancing convenience for students.
5. **Automated Notifications and Alerts**: Developing a notification system to alert students about important updates, such as fee deadlines or maintenance schedules, would improve communication and engagement.
6. **Room Availability and Booking System**: Introducing a real-time room availability and booking feature would allow prospective students to view available accommodations and make reservations directly through the system.
7. **Enhanced Security Features**: Future iterations may require implementing more robust security measures, such as multi-factor authentication, to protect sensitive student and financial data.
8. **Support for Multiple Hostels**: Expanding the system to manage multiple hostel properties under a single platform would enable scalability and serve larger institutions with multiple facilities.
9. **API Integration**: Developing APIs to allow third-party applications to interface with the Hostel Management System could enhance interoperability and expand its capabilities.

## Limitation

While the Hostel Management System aims to enhance operational efficiency and user experience, several limitations may impact its functionality and effectiveness:

1. **Dependency on Internet Connectivity**: The system may require stable internet access for certain features, particularly for remote management and real-time data updates, which could hinder usability in areas with poor connectivity.
2. **User Interface Limitations**: The desktop application may not offer the same level of user experience as modern web or mobile applications. Limited design flexibility could impact user engagement and satisfaction.
3. **Scalability Concerns**: As the number of users and data volume grows, the system may face performance issues unless properly optimized. Scaling the database and application infrastructure may require significant investment.
4. **Limited Mobile Support**: Without a dedicated mobile application, users may face challenges accessing the system on smartphones or tablets, reducing overall accessibility and convenience.
5. **Integration Challenges**: Integrating with third-party services (like payment gateways or external reporting tools) may present challenges in terms of compatibility and additional development effort.

**Bibliography**

**Books**

* + Schildt, Herbert. *Java: The Complete Reference*. 11th ed., McGraw-Hill Education, 2018.
  + Taneja, B. M. *PHP and MySQL Web Development*. 4th ed., Addison-Wesley, 2015.

**Online Resources**

* + Oracle. “Java SE Documentation.” [Oracle Java SE](https://docs.oracle.com/en/java/javase/11/index.html).
  + W3Schools. “MySQL Tutorial.” W3Schools MySQL.
  + MySQL. “MySQL Documentation.” [MySQL](https://dev.mysql.com/doc/).

**Development Tools**

* + Apache Friends. “XAMPP.” XAMPP.
  + Apache NetBeans. “NetBeans IDE.” [NetBeans](https://netbeans.apache.org/).

**Java GUI Development**

* + Eck, Herbert Schildt. *Java: A Beginner’s Guide*. 8th ed., McGraw-Hill Education, 2018.

**Research Papers**

* + Bhatt, S., & Choudhary, A. (2020). “A Study on Hostel Management System.” *International Journal of Computer Applications*, 975, 8887.

**Web Articles**

* + Gajanan, S. “Understanding the Importance of Hostel Management Systems.” *Tech Times*, 2022. Tech Times.
  + Smith, J. “Top Features of a Modern Hostel Management System.” *Hospitality Tech*, 2023. Hospitality Tech.

**Official Documentation**

* + Firebase. “Firebase Documentation.” Firebase.
  + Oracle. “Java Swing Documentation.” [Java Swing](https://docs.oracle.com/javase/tutorial/uiswing/).