HOSTEL MANAGEMENT SYSTEM

A MINI-PROJECT REPORT

Submitted by

BARATHRAJ M 241901015

ARUNESH P M 241901007

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

(CYBER SECURITY)

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI- 602105

An Autonomous Institute

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project **“HOSTEL MANAGEMENT SYSTEM”** is the bonafide work of **“BARATHRAJ M (241901015), ARUNESH P M (241901007)”** who carried out the project work under my supervision.

[illegible]

SIGNATURE

Ms.R. Rupmala

ASSISTANT PROFESSOR

Department of Computer Science

and Engineering (Cyber Security)

Rajalakshmi Engineering College

Chennai

This mini project report is submitted for the viva voce examination to be held on __________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We hereby declare that the mini project report **HOSTEL MANAGEMENT SYSTEM**, submitted as part of the curriculum requirements for the Bachelor of Engineering (B.E) degree affiliated to Anna University, is a bonafide work carried out by us under the supervision of Ms. R. Rupmala, Assistant Professor, Department of Computer Science Engineering and Cyber Security, Rajalakshmi Engineering College, Chennai.

This submission represents our ideas in our own words, and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data, idea, fact, or source in our submission. We understand that any violation of the above will be grounds for disciplinary action by the institute and/or the University and may also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for the award of any degree, diploma, or similar title of any other University.

Rajalakshmi Engineering College,

Chennai

October 2025

BARATHRAJ M

ARUNESH P M

ABSTRACT

The Hostel Management System is a standalone desktop application developed using **Java** and the **Swing framework** for the graphical user interface, with **MySQL** serving as the backend database. The system is designed to provide a reliable, efficient, and user-friendly platform for managing essential hostel-related activities. Its primary focus is on streamlining **student management** and **room management**, which are the core operations in most hostel environments.

The application supports all fundamental **CRUD** (Create, Read, Update, Delete) functionalities for both students and rooms. This enables hostel administrators to easily add new student information, allocate rooms, update existing data, remove outdated records, and view complete lists of all registered students and available rooms. The system ensures that data is maintained in a structured, accurate, and organized manner, thereby reducing manual errors and improving operational efficiency.

To enhance clarity and maintainability, the application follows a **layered architecture**, separating the user interface from the business logic and database operations. This modular design not only makes the system easy to understand and manage but also provides flexibility for future enhancements such as fee management, automated room allotment, attendance tracking, or report generation.

Overall, the Hostel Management System delivers a simple yet effective digital solution for hostel administration, offering improved **data handling**, faster access to information, and a more streamlined workflow. It serves as a strong foundation for building more advanced hostel management tools in the future.

KEYWORKS : Java, Swing Framework, MySQL, Desktop Application, CRUD Operations, Student Management, Room Management, Database Management, Layered Architecture, User Interface, Administration, Data Handling, Record Management, Automation.

ACKNOWLEDGEMENT

We like to convey our sincere appreciation to all who have supported and mentored us during the successful completion of this project work.

We express our profound gratitude to **Mr. Benedict J.N.**, Associate Professor (SG) and Head of the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for furnishing us with essential resources, support, and an enabling environment to execute this project.

We express our profound gratitude to **Ms. R. Rupmala**, Assistant Professor in the Department of Computer Science Engineering and Cyber Security at Rajalakshmi Engineering College, for her unwavering support, invaluable insights, and collaboration throughout our endeavor.

We would like to convey our gratitude to our faculty members and colleagues for their valuable feedback and encouragement.

We express our gratitude to our families and friends for their steadfast support, patience, and encouragement, which were instrumental in the effective execution of this seminar.

BARATHRAJ M

ARUNESH P M

TABLE OF CONTENTS

LIST OF ABBREVIATION

Abbreviation	Full Term
GUI	Graphical User Interface
CRUD	Create, Read, Update, Delete
SQL	Structured Query Language
DAO	Data Access Object
JDBC	Java Database Connectivity
IDE	Integrated Development Environment

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Project Overview

The Hostel Management System is a Java-based application designed to simplify the administration of a hostel. The system provides a centralized platform to manage student information and room allocations. The primary goal of this project is to replace manual record-keeping with an automated system, reducing the risk of errors and improving efficiency.

The application features a simple and intuitive graphical user interface (GUI) built with Java Swing. It connects to a MySQL database to store and retrieve data, ensuring data persistence and reliability. The core functionalities include adding new students, adding new rooms, and viewing lists of all students and rooms.

1.2 Scope of the Work

The scope of this project is to develop a functional desktop application for hostel management with the following key features:

- **Student Management:**
  - Add new students with details such as name, age, gender, and assigned room.
  - View a comprehensive list of all students currently residing in the hostel.
- **Room Management:**
  - Add new rooms with details like hostel name, capacity, and current occupancy.
  - View a list of all available rooms and their status.
- **Database Integration:**
  - A robust backend using a MySQL database to store and manage all the data.
  - Use of JDBC for seamless communication between the Java application and the database.

1.3 Problem Statement

In many hostels, the management of student records and room allocations is still done manually using paper-based ledgers or simple spreadsheets. This traditional approach is prone to several issues, including:

- **Data Redundancy and Inconsistency:** Manual data entry can lead to duplicate records and inconsistent information.
- **Inefficiency:** Searching for and retrieving specific information can be a time-consuming process.
- **Risk of Data Loss:** Physical records are susceptible to damage or loss.
- **Lack of Real-time Information:** It is difficult to get a real-time overview of room occupancy and student details.

The Hostel Management System aims to address these problems by providing a digital solution that is efficient, reliable, and easy to use.

1.4 Aim and Objectives of the Project

Aim: To develop a desktop application that automates the process of managing student and room information in a hostel.

Objectives:

- To design and implement a user-friendly graphical interface for easy interaction.
- To develop a robust database schema to store student and room data.
- To implement functionalities for adding and viewing students and rooms.
- To ensure the application is reliable and maintains data integrity.

To create a scalable system that can be enhanced with additional features in the future

CHAPTER 2
SYSTEM SPECIFICATIONS

2.1 Hardware Specifications

- **Processor:** Intel Core i3 or equivalent
- **RAM:** 4 GB or more
- **Hard Disk:** 10 GB of free space
- **Operating System:** Windows, macOS, or Linux

2.2 Software Specifications

- **Programming Language:** Java
- **Development Kit:** JDK (Java Development Kit) 8 or higher
- **IDE:** VS Code, IntelliJ IDEA, Eclipse, or any other Java-supported IDE
- **Database:** MySQL
- **Frameworks/Libraries:** Java Swing for GUI, JDBC for database connectivity.

CHAPTER 3
MODULE DESCRIPTION

The Hostel Management System is designed with a modular architecture, which separates the application into distinct logical components. This makes the system easier to develop, test, and maintain.

3.1 Data Access Module

This module is responsible for all interactions with the database. It abstracts the database operations from the rest of the application. The dao package contains the classes for this module.

- **DBConnection.java:** A utility class to establish a connection with the MySQL database. It provides a static method to get a database connection.
- **StudentDAO.java:** This class handles all the database operations related to students, such as adding a new student and retrieving a list of all students.
- **RoomDAO.java:** This class manages the database operations for rooms, including adding a new room and fetching all room details.

3.2 Data Model Module

This module defines the data structures that represent the core entities of the system. The model package contains the classes for this module.

- **Student.java**: A Plain Old Java Object (POJO) class that represents a student. It contains fields for student ID, name, age, gender, and room ID, along with getter and setter methods.
- **Room.java**: A POJO class that represents a room. It includes fields for room ID, hostel name, capacity, and occupancy, with corresponding getter and setter methods.

3.3 User Interface Module

This module provides the graphical user interface for the application. It is built using the Java Swing framework and is located in the ui package.

- **MainFrame.java:** The main window of the application, which provides navigation to other forms.

- **StudentForm.java:** A form for adding a new student to the system.
- **RoomForm.java:** A form for adding a new room.
- **ViewStudents.java**: A window that displays a list of all students in a table format.
- **ViewRooms.java:** A window that displays a list of all rooms in a table format.

3.4 Database Module

The database is the backbone of the system, responsible for persisting all the data. A MySQL database is used for this purpose. The database contains two main tables: student and room.

3.5 ER Diagram

The Entity-Relationship (ER) diagram below illustrates the relationship between the entities in the system.

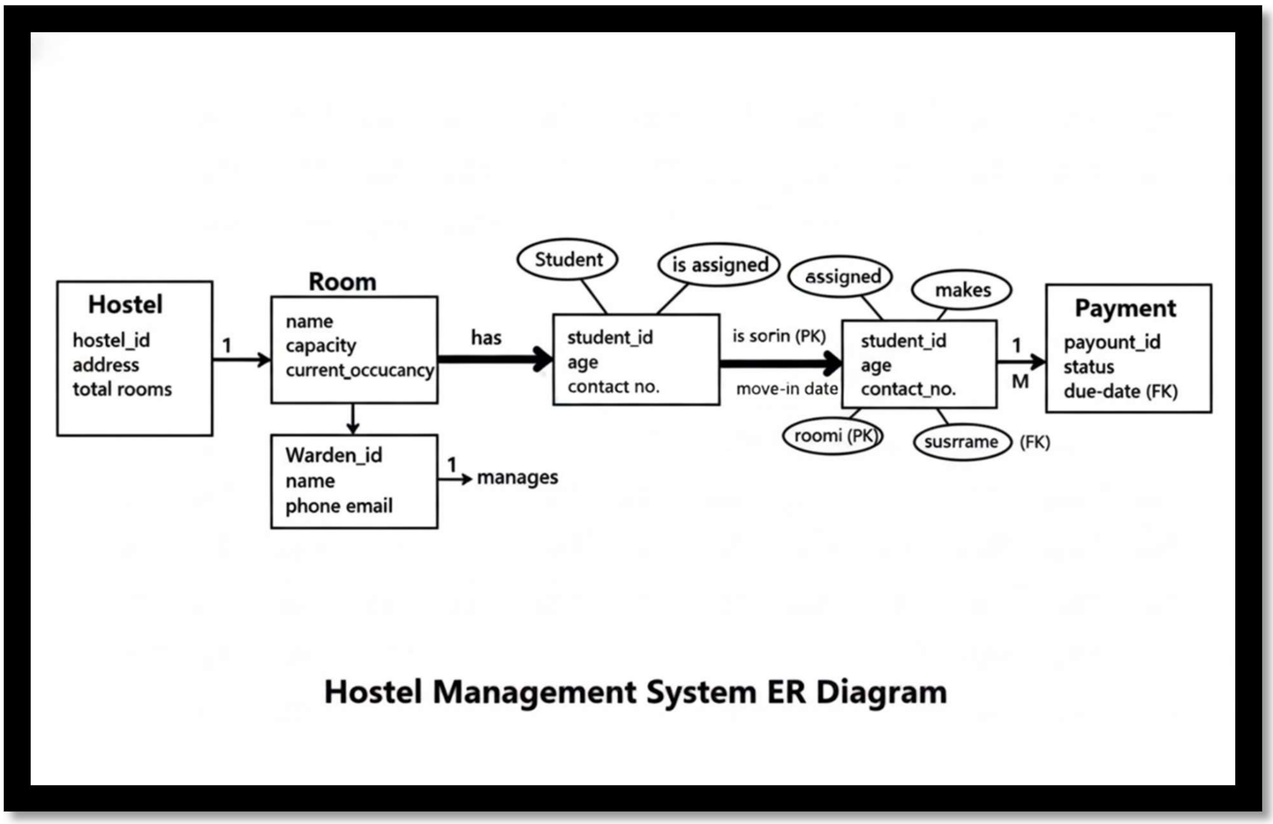


FIG. 3.1 ER Diagram

3.6 Database Schema

The database schema consists of two tables: student and room.

Student Table:

Column	Type	Constraints
student_id	INT	PRIMARY KEY, AUTO_INCREMENT
name	VARCHAR(255)	NOT NULL
age	INT	
gender	VARCHAR(10)	
room_id	INT	FOREIGN KEY (references room.room_id)

Room Table:

Column	Type	Constraints
room_id	INT	PRIMARY KEY, AUTO_INCREMENT
hostel_name	VARCHAR(255)	NOT NULL
capacity	INT	
occupied	INT	

CHAPTER 4
CODING

This chapter provides an overview of the key coding aspects of the Hostel Management System.

4.1 Core Components

The application is built around a few core components that work together to provide the required functionality.

- **MainFrame:** This is the entry point of the user interface. It presents the main menu with options to add and view students and rooms.
- **DAO Classes:** The StudentDAO and RoomDAO classes encapsulate the logic for interacting with the database. They use JDBC to execute SQL queries.
- **Model Classes:** The Student and Room classes are simple data containers that hold the information retrieved from the database or entered by the user.

4.2 Database Integration

Database connectivity is managed by the DBConnection class, which provides a centralized way to get a database connection. The DAO classes use this connection to perform CRUD operations.

Example: Adding a Student

The addStudent method in StudentDAO.java demonstrates how a new student is added to the database.

```
public void addStudent(Student s) {
   String query = "INSERT INTO student(name, age, gender, room_id)VALUES (?, ?, ?, ?)";
   try (Connection con = DBConnection.getConnection();
        PreparedStatement ps = con.prepareStatement(query)) {
      ps.setString(1, s.getName());
      ps.setInt(2, s.getAge());
      ps.setString(3, s.getGender());
      ps.setInt(4, s.getRoomId());
      ps.executeUpdate();
```

```
        System.out.println("✅ Student added successfully!");
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

This code snippet shows the use of a PreparedStatement to safely insert data into the student table, preventing SQL injection attacks.

CHAPTER 5
SCREENSHOTS

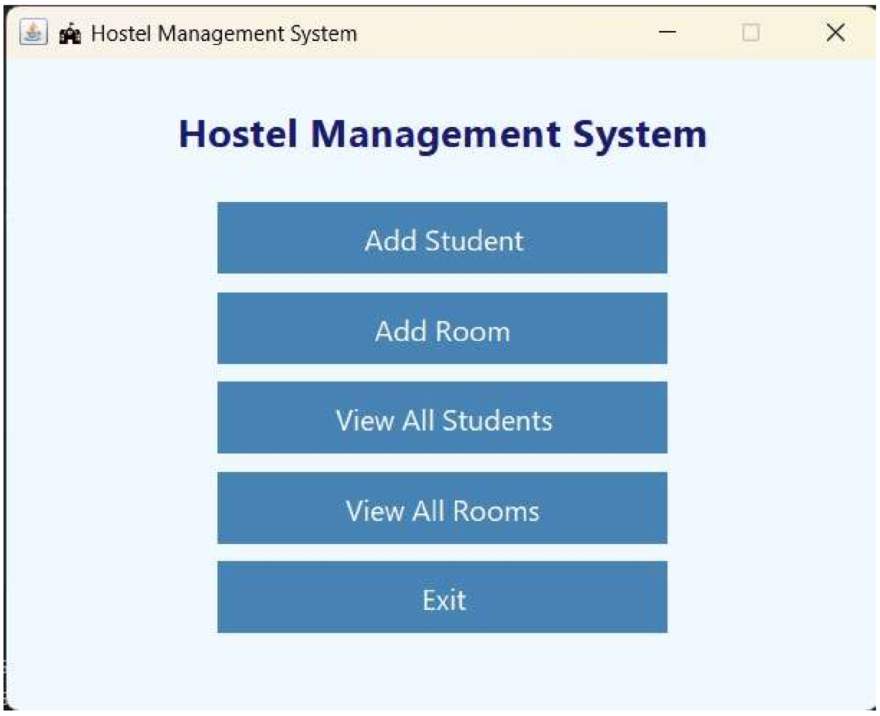


Fig 5.1 Main frame page

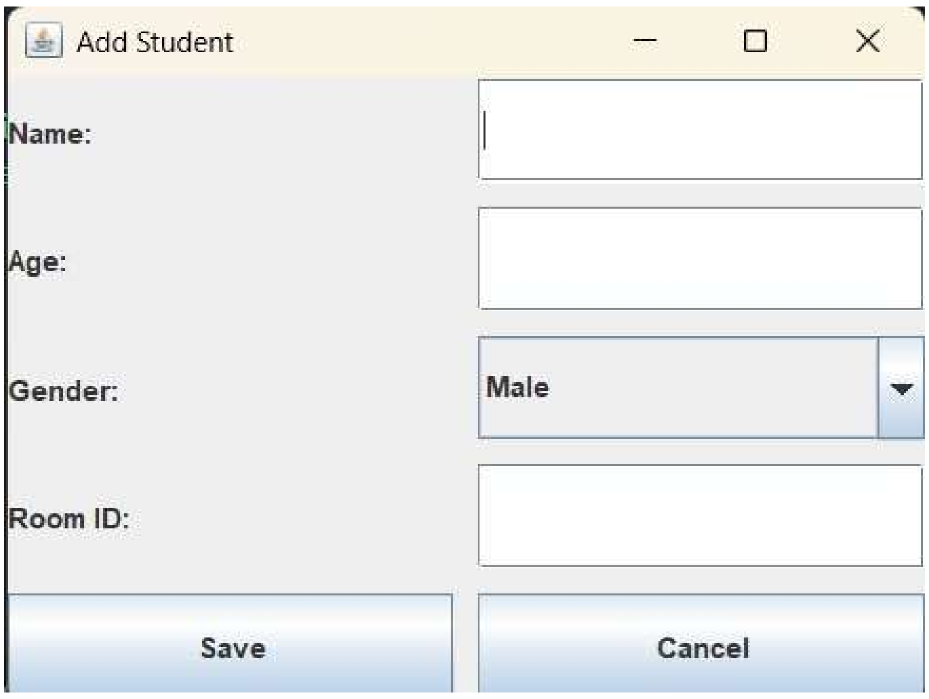
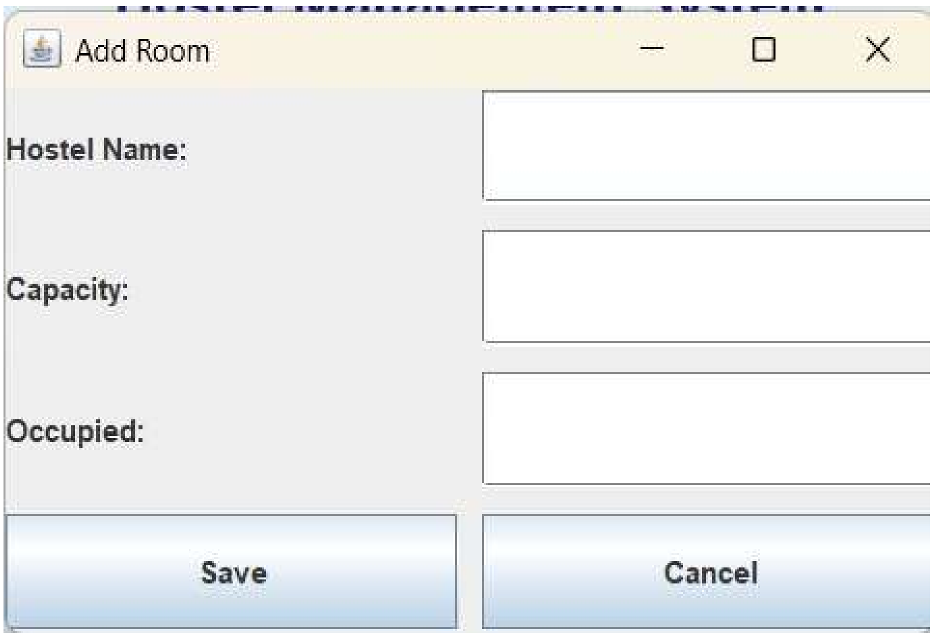
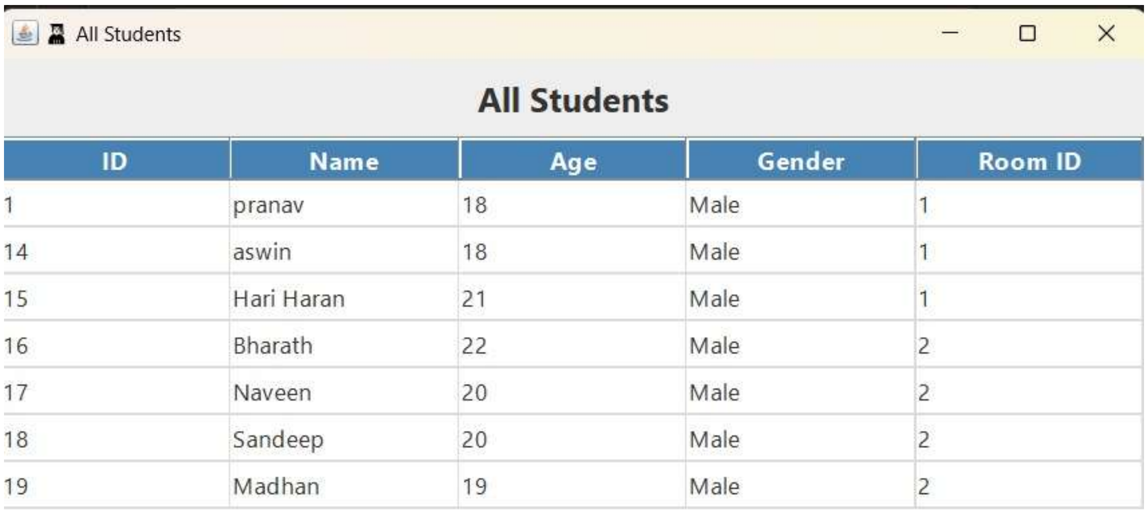


Fig 5.2 Add student details



Hostel Name:	
Capacity:	
Occupied:	

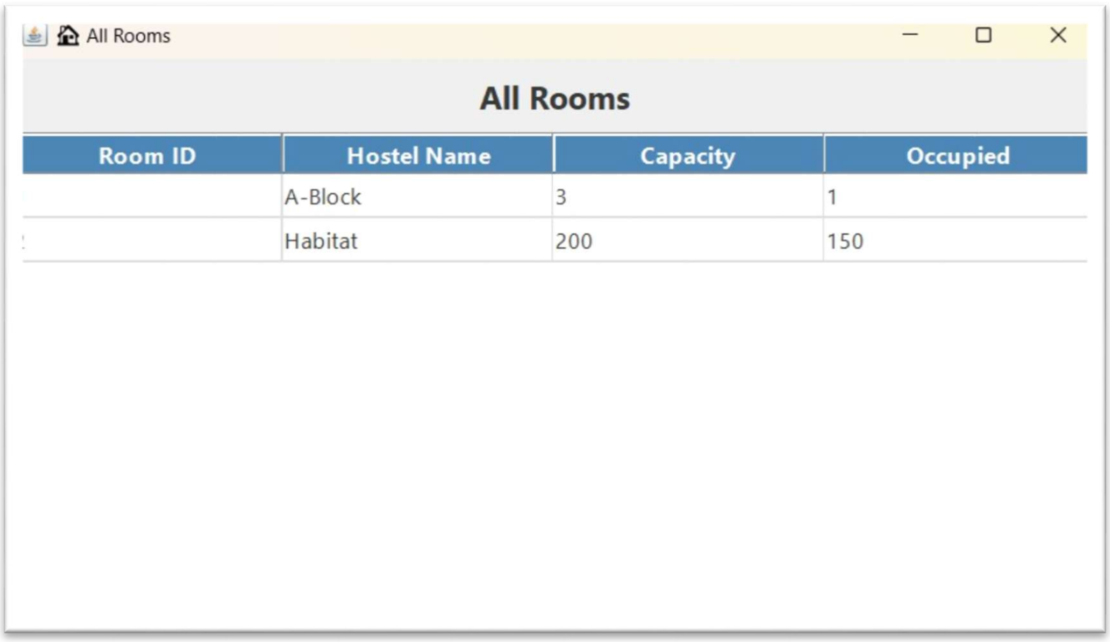
Fig 5.3 Add room details



All Students

ID	Name	Age	Gender	Room ID
1	pranav	18	Male	1
14	aswin	18	Male	1
15	Hari Haran	21	Male	1
16	Bharath	22	Male	2
17	Naveen	20	Male	2
18	Sandeep	20	Male	2
19	Madhan	19	Male	2

Fig 5.4 View all student


All Rooms

All Rooms

Room ID	Hostel Name	Capacity	Occupied
	A-Block	3	1
	Habitat	200	150

Fig 5.5 View all rooms

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

The Hostel Management System has been successfully developed as a desktop application that meets the basic requirements for managing student and room data in a hostel. The system provides a user-friendly interface and a reliable database backend. The modular design of the application makes it easy to maintain and extend. This project serves as a practical example of using Java Swing and JDBC to build a real-world application.

6.2 Future Enhancements

The system can be further enhanced with the following features:

- **User Authentication:** Implement a login system to restrict access to authorized users.
- **Update and Delete Functionality:** Add features to edit and delete student and room records.
- **Search and Filter:** Implement search and filter options to easily find specific records.
- **Reporting:** Generate reports on room occupancy, student demographics, etc.
- **Fee Management:** Add a module to manage student fees and payments.
- **Web-based Interface:** Develop a web-based version of the application for remote access.

REFERENCES

[1] Bloch, J. (2018). Effective Java (3rd ed.). Addison-Wesley Professional.

[2] Deitel, P. J., & Deitel, H. M. (2018). Java How to Program, Early Objects (11th ed.). Pearson Education.

[3] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional

[4] Oracle Corporation. (2023). Java Platform, Standard Edition & Java Development Kit Version 21 API Specification. Retrieved from

[5Oracle Corporation. (2023). MySQL 8.0 Reference Manual. Retrieved from https://dev.mysql.com/doc/refman/8.0/en/

[6] Schildt, H. (2018). Java: The Complete Reference (11th ed.). McGraw-Hill Education.