HOSTEL MANAGEMENT SYSTEM

A MINI-PROJECT BY:

Mymoon Beevi. B	**230701201**
Omesh Balamurugan	**230701222**

in partial fulfillment of the award of the degree

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project **“HOSTEL MANAGEMENT SYSTEM”** is the

bonafide work of **“OMESH BALAMURUGAN , MYMOON BEEVI.B”** who carried out the project work under my supervision.

Submitted for the practical examination held on ____________________

SIGNATURE SIGNATURE

Ms. ASWANA LAL

Asst. Professor

Computer Science and Engineering,

Rajalakshmi Engineering College

(Autonomous),

Thandalam,Chennai-602105

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Hostel management system is a simple console application using JDBC in Java programming language. Manager can perform basic Hostel management operations by logging in and do addition, updating, deletion, viewing student details and check room availability and mess fees and students left and yet to leave.

Each student in the hostel has a unique roll number. The hostel manager enters the student details such as name, roll number, room number, department.

Each student gets allocated to their rooms which is registered in the database of the hostel manager.

When the student returns back to room, the room gets filled in the hostel.

The record of the student returned can also be viewed.

TABLE OF CONTENTS

INTRODUCTION

1.1 INTRODUCTION

In this program, Manager can perform basic hostel management operations like adding student, updating student details, viewing students in hostel, deleting student details. Each student in the hostel has a unique roll number. The manager adds the student by entering the roll number and other student details. Manager can add only one student at a time. The record of the added student with student details can also be viewed.

1.1 IMPLEMENTATION

The **HOSTEL MANAGEMENT** project discussed here is implemented using the concepts of **JAVA SWINGS** and **MYSQL**.

1.2 SCOPE OF THE PROJECT

At the end of this program we have seen addition, updating, deleting, viewing and other features related to hostel management. This program is a simple and 'easier to manage' type for a manager of hostel. The program is done in Visual Studio Code along with JDBC and jar frame in order to present the application window.

1.3 APPLICATION FEATURES

- Registering and login page.
- Custom profile for each student.
- Dashboard showing possible actions.
- Manage room availability
- Option to submit project on website.

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS:

PROCESSOR	:	Intel i5
MEMORY SIZE	:	4GB(Minimum)
HARD DISK	:	500 GB of free space

2.2 SOFTWARE SPECIFICATIONS:

PROGRAMMING LANGUAGE	:	Java, MySQL
FRONT-END	:	Java
BACK-END	:	MySQL
OPERATING SYSTEM	:	Windows 10

SAMPLE CODE

3.1 LOGIN PAGE DESIGN

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Login extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private JButton loginButton;

    public Login() {
        setTitle("Hostel Management System - Login");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null);

        JLabel userLabel = new JLabel("Username:");
        userLabel.setBounds(10, 10, 80, 25);
        add(userLabel);

        usernameField = new JTextField();
        usernameField.setBounds(100, 10, 160, 25);
        add(usernameField);

        JLabel passwordLabel = new JLabel("Password:");
        passwordLabel.setBounds(10, 40, 80, 25);
        add(passwordLabel);

        passwordField = new JPasswordField();
        passwordField.setBounds(100, 40, 160, 25);
        add(passwordField);

        loginButton = new JButton("Login");
        loginButton.setBounds(100, 70, 80, 25);
        add(loginButton);

        loginButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String username = usernameField.getText();
                String password = new String(passwordField.getPassword());
                if (username.equals("admin") && password.equals("admin")) {
                    dispose();
                    new HostelManagementApp().setVisible(true);
                } else {
                    JOptionPane.showMessageDialog(null, "Invalid Username or Password!");
                }
            }
        });
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            new Login().setVisible(true);
        });
    }
  }
```

3.2 HOSTELITE FACILITY

```
import javax.swing.*;
import java.awt.*;
import java.sql.*;
import javax.swing.table.DefaultTableModel;

public class HosteliteManager extends JFrame {
    private JTextField nameField, rollNumberField, roomNumberField, genderField,
      courseField, statusField;
    private JButton addButton, removeButton, updateButton, searchButton, displayButton;

    private static final String DB_URL = "jdbc:mysql://localhost:3306/hostel_management";
    private static final String USER = "root";
    private static final String PASS = "777Wonders*@"; // Replace with your MySQL
      password

    public HosteliteManager() {
        setTitle("Hostelites Management");
        setSize(500, 450);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);

        // Input fields and labels
        addLabelAndField("Name:", 10, 10, 100, 25, nameField = new JTextField());
        addLabelAndField("Roll Number:", 10, 50, 100, 25, rollNumberField = new
      JTextField());
        addLabelAndField("Room Number:", 10, 90, 100, 25, roomNumberField = new
      JTextField());
        addLabelAndField("Gender:", 10, 130, 100, 25, genderField = new JTextField());
        addLabelAndField("Course:", 10, 170, 100, 25, courseField = new JTextField());
        addLabelAndField("Status:", 10, 210, 100, 25, statusField = new JTextField());

        // Buttons
        addButton = new JButton("Add Student");
        addButton.setBounds(10, 250, 150, 25);
        add(addButton);

        removeButton = new JButton("Remove Student");
        removeButton.setBounds(180, 250, 150, 25);
        add(removeButton);

        updateButton = new JButton("Update Student");
        updateButton.setBounds(10, 290, 150, 25);
        add(updateButton);

        searchButton = new JButton("Search Student");
        searchButton.setBounds(180, 290, 150, 25);
        add(searchButton);

        displayButton = new JButton("Display All Students");
        displayButton.setBounds(10, 330, 320, 25);
        add(displayButton);

        // Button actions
        addButton.addActionListener(e -> addStudent());
        removeButton.addActionListener(e -> removeStudent());
        updateButton.addActionListener(e -> updateStudent());
        searchButton.addActionListener(e -> searchStudent());
        displayButton.addActionListener(e -> displayAllStudents());
    }

    private void addLabelAndField(String label, int x, int y, int width, int height, JTextField
      field) {
        JLabel jLabel = new JLabel(label);
        jLabel.setBounds(x, y, width, height);
        add(jLabel);

        field.setBounds(x + 120, y, 150, 25);
        add(field);
    }

    private void addStudent() {
        String name = nameField.getText();
        String rollNumber = rollNumberField.getText();
        String roomNumber = roomNumberField.getText();
        String gender = genderField.getText();
        String course = courseField.getText();
        String status = statusField.getText();

        try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
             PreparedStatement pstmt = conn.prepareStatement("INSERT INTO hostelites
      (name, roll_number, room_number, gender, course, status) VALUES (?, ?, ?, ?, ?, ?)")) {
            pstmt.setString(1, name);
            pstmt.setString(2, rollNumber);
            pstmt.setString(3, roomNumber);
            pstmt.setString(4, gender);
            pstmt.setString(5, course);
            pstmt.setString(6, status);
            pstmt.executeUpdate();
            JOptionPane.showMessageDialog(this, "Student added successfully!");
            clearFields();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void removeStudent() {
        String rollNumber = rollNumberField.getText().trim();

        if (rollNumber.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Please enter a roll number!");
            return;
        }

        try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
             PreparedStatement pstmt = conn.prepareStatement("DELETE FROM hostelites
      WHERE roll_number = ?")) {
            pstmt.setString(1, rollNumber);
            int rowsAffected = pstmt.executeUpdate();

            if (rowsAffected > 0) {
                JOptionPane.showMessageDialog(this, "Student removed successfully!");
            } else {
                JOptionPane.showMessageDialog(this, "Student not found!");
            }

            clearFields();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void updateStudent() {
```

```
    String rollNumber = rollNumberField.getText();
    String name = nameField.getText();
    String roomNumber = roomNumberField.getText();
    String gender = genderField.getText();
    String course = courseField.getText();
    String status = statusField.getText();

    try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
         PreparedStatement pstmt = conn.prepareStatement("UPDATE hostelites SET name
  = ?, room_number = ?, gender = ?, course = ?, status = ? WHERE roll_number = ?")) {
        pstmt.setString(1, name);
        pstmt.setString(2, roomNumber);
        pstmt.setString(3, gender);
        pstmt.setString(4, course);
        pstmt.setString(5, status);
        pstmt.setString(6, rollNumber);

        int rowsAffected = pstmt.executeUpdate();
        if (rowsAffected > 0) {
            JOptionPane.showMessageDialog(this, "Student updated successfully!");
        } else {
            JOptionPane.showMessageDialog(this, "Student not found!");
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

private void searchStudent() {
    String rollNumber = rollNumberField.getText();

    try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
         PreparedStatement pstmt = conn.prepareStatement("SELECT * FROM hostelites
  WHERE roll_number = ?")) {
        pstmt.setString(1, rollNumber);
        ResultSet rs = pstmt.executeQuery();

        if (rs.next()) {
            String name = rs.getString("name");
            String roomNumber = rs.getString("room_number");
            String gender = rs.getString("gender");
            String course = rs.getString("course");
            String status = rs.getString("status");

            String studentDetails = "Name: " + name + "\n" +
                                    "Roll Number: " + rollNumber + "\n" +
                                    "Room Number: " + roomNumber + "\n" +
                                    "Gender: " + gender + "\n" +
                                    "Course: " + course + "\n" +
                                    "Status: " + status;
            JOptionPane.showMessageDialog(this, studentDetails, "Student Details",
  JOptionPane.INFORMATION_MESSAGE);

            nameField.setText(name);
            roomNumberField.setText(roomNumber);
            genderField.setText(gender);
            courseField.setText(course);
            statusField.setText(status);
        } else {
            JOptionPane.showMessageDialog(this, "Student not found!");
            clearFields();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

private void displayAllStudents() {
    JFrame tableFrame = new JFrame("All Students");
    tableFrame.setSize(600, 400);

    String[] columnNames = {"Name", "Roll Number", "Room Number", "Gender",
  "Course", "Status"};
    DefaultTableModel model = new DefaultTableModel(columnNames, 0);

    try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
         Statement stmt = conn.createStatement();
         ResultSet rs = stmt.executeQuery("SELECT * FROM hostelites")) {

        while (rs.next()) {
            String name = rs.getString("name");
            String rollNumber = rs.getString("roll_number");
            String roomNumber = rs.getString("room_number");
            String gender = rs.getString("gender");
            String course = rs.getString("course");
            String status = rs.getString("status");

            model.addRow(new Object[]{name, rollNumber, roomNumber, gender, course,
  status});
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
```

```
        JTable table = new JTable(model);
        JScrollPane scrollPane = new JScrollPane(table);
        tableFrame.add(scrollPane);
        tableFrame.setVisible(true);
    }

    private void clearFields() {
        nameField.setText("");
        rollNumberField.setText("");
        roomNumberField.setText("");
        genderField.setText("");
        courseField.setText("");
        statusField.setText("");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new HosteliteManager().setVisible(true));
    }
     }
```

3.3 HOSTEL MANAGER APP

```
import javax.swing.*;

public class HostelManagementApp extends JFrame {

    public HostelManagementApp() {
        setTitle("Hostel Management System - Main Menu");
        setSize(400, 350);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null);

        JButton hostelitesButton = new JButton("Hostelites Management");
        hostelitesButton.setBounds(100, 30, 200, 30);
        add(hostelitesButton);

        JButton messBillButton = new JButton("Receive Mess Bill Payment");
        messBillButton.setBounds(100, 70, 200, 30);
        add(messBillButton);

        JButton checkStatusButton = new JButton("Check Student/Employee Status");
        checkStatusButton.setBounds(100, 110, 200, 30);
        add(checkStatusButton);

        JButton roomAvailabilityButton = new JButton("Check Room Availability");
        roomAvailabilityButton.setBounds(100, 150, 200, 30);
        add(roomAvailabilityButton);
        JButton logoutButton = new JButton("Back to Login");
        logoutButton.setBounds(100, 190, 200, 30);
        add(logoutButton);

        JButton exitButton = new JButton("Exit");
        exitButton.setBounds(100, 230, 200, 30);
        add(exitButton);

        // Using lambda expressions to handle button clicks
        hostelitesButton.addActionListener(e -> new HosteliteManager().setVisible(true));
        messBillButton.addActionListener(e -> new PaymentManager().setVisible(true));
        checkStatusButton.addActionListener(e -> new StatusManager().setVisible(true));
        roomAvailabilityButton.addActionListener(e -> new
       RoomAvailabilityManager().setVisible(true));
        logoutButton.addActionListener(e -> {
            dispose();
            new Login().setVisible(true);
        });
        exitButton.addActionListener(e -> System.exit(0));
    }
      }
```

3.4 PAYMENT MANAGER

```
import javax.swing.*;
import java.sql.*;

public class PaymentManager extends JFrame {
    private JTextField rollNumberField;

    private static final String DB_URL = "jdbc:mysql://localhost:3306/hostel_management";
    private static final String USER = "root";
    private static final String PASS = "777Wonders*@"; // Replace with your MySQL
      password

    public PaymentManager() {
        setTitle("Mess Bill Payment");
        setSize(300, 150);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);

        JLabel rollLabel = new JLabel("Roll Number:");
        rollLabel.setBounds(10, 10, 100, 25);
        add(rollLabel);

        rollNumberField = new JTextField();
        rollNumberField.setBounds(120, 10, 150, 25);
        add(rollNumberField);

        JButton payButton = new JButton("Pay Mess Bill");
        payButton.setBounds(10, 50, 150, 25);
        add(payButton);

        payButton.addActionListener(e -> payMessBill());
    }

    private void payMessBill() {
        String rollNumber = rollNumberField.getText();

        try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS)) {
            // Check if roll_number exists in mess_bills
            PreparedStatement checkStmt = conn.prepareStatement("SELECT * FROM
      mess_bills WHERE roll_number = ?");
            checkStmt.setString(1, rollNumber);
            ResultSet rs = checkStmt.executeQuery();

            if (rs.next()) {
                // Update the payment_status to 'Paid' in mess_bills table
                PreparedStatement pstmt = conn.prepareStatement("UPDATE mess_bills SET
      payment_status = 'Paid' WHERE roll_number = ?");
                pstmt.setString(1, rollNumber);
                int rowsAffected = pstmt.executeUpdate();

                if (rowsAffected > 0) {
                    JOptionPane.showMessageDialog(this, "Mess bill paid successfully!");
                } else {
                    JOptionPane.showMessageDialog(this, "Error processing payment.");
                }
            } else {
                JOptionPane.showMessageDialog(this, "Student not found in mess_bills table!");
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
      }
```

3.5 ROOM AVAILABILITY CHECKING

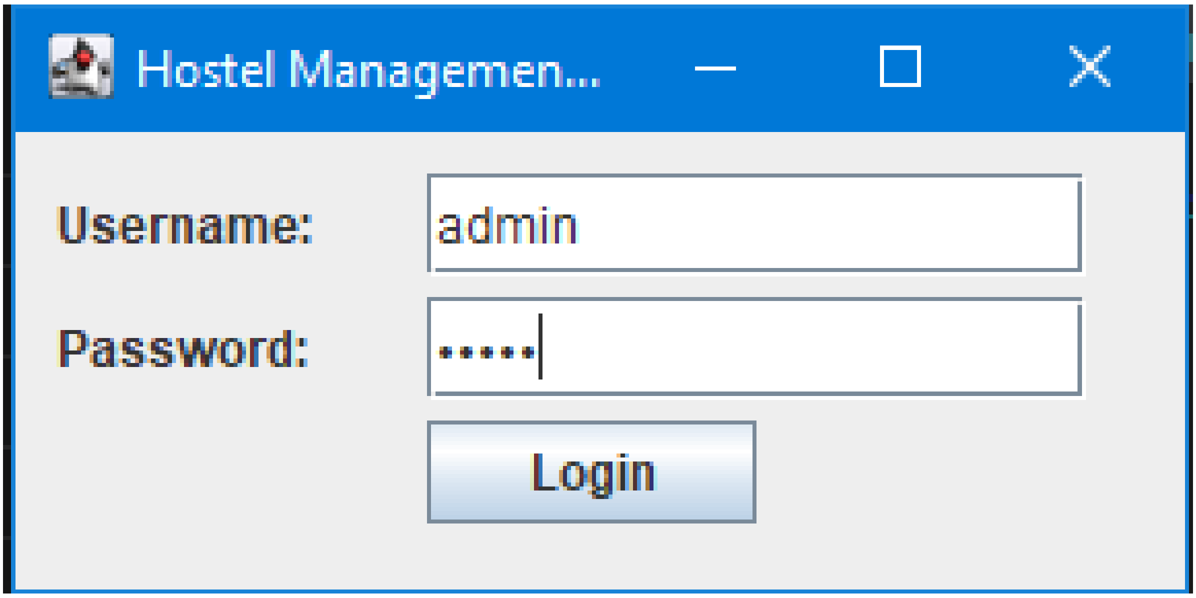
```
import java.sql.*;
import javax.swing.*;

public class RoomAvailabilityManager extends JFrame {

    private static final String DB_URL = "jdbc:mysql://localhost:3306/hostel_management";
    private static final String USER = "root";
    private static final String PASS = "777Wonders*@"; // Replace with your MySQL
password
    private static final int TOTAL_ROOMS = 5; // Define total rooms as 10

    private final JLabel availabilityLabel;

    public RoomAvailabilityManager() {
        setTitle("Room Availability");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);

        availabilityLabel = new JLabel("Checking room availability...");
        availabilityLabel.setBounds(20, 20, 250, 30);
        add(availabilityLabel);

        // Call the method to check availability after setting up the UI
        checkRoomAvailability();
    }

    private void checkRoomAvailability() {
        try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery("SELECT COUNT(*) AS student_count FROM
hostelites")) {

            if (rs.next()) {
                int studentCount = rs.getInt("student_count");
                if (studentCount >= TOTAL_ROOMS) {
                    availabilityLabel.setText("Status: Full");
                } else {
                    availabilityLabel.setText("Status: Vacant");
                }
            }
        } catch (SQLException e) {
            // Displaying specific error details in a dialog
            JOptionPane.showMessageDialog(this, "Error checking availability: " +
e.getMessage(), "Database Error", JOptionPane.ERROR_MESSAGE);
            System.out.println();
        }
    }
}
```

3.6 STATUS MANAGER

```
import javax.swing.*;
import java.sql.*;

public class StatusManager extends JFrame {

    private static final String DB_URL = "jdbc:mysql://localhost:3306/hostel_management";
    private static final String USER = "root";
    private static final String PASS = "777Wonders*@"; // Replace with your MySQL
password

    public StatusManager() {
        setTitle("Check Student/Employee Status");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setLayout(null);

        JButton checkLivingButton = new JButton("Check Living Students");
        checkLivingButton.setBounds(10, 10, 200, 25);
        add(checkLivingButton);

        JButton checkLeavedButton = new JButton("Check Leaved Students");
        checkLeavedButton.setBounds(10, 50, 200, 25);
        add(checkLeavedButton);

        checkLivingButton.addActionListener(e -> checkStatus("Living"));
        checkLeavedButton.addActionListener(e -> checkStatus("Leaved"));
    }

    private void checkStatus(String status) {
        try (Connection conn = DriverManager.getConnection(DB_URL, USER, PASS);
             PreparedStatement pstmt = conn.prepareStatement("SELECT * FROM hostelites
WHERE status = ?")) {
            pstmt.setString(1, status);
            ResultSet rs = pstmt.executeQuery();
            StringBuilder result = new StringBuilder();

            while (rs.next()) {
                String name = rs.getString("name");
                String rollNumber = rs.getString("roll_number");
                result.append("Name: ").append(name).append(", Roll No: ").append(rollNumber).append("\n");
            }

            if (result.length() == 0) {
                JOptionPane.showMessageDialog(this, "No students found.");
            } else {
                JOptionPane.showMessageDialog(this, result.toString());
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

SNAPSHOTS

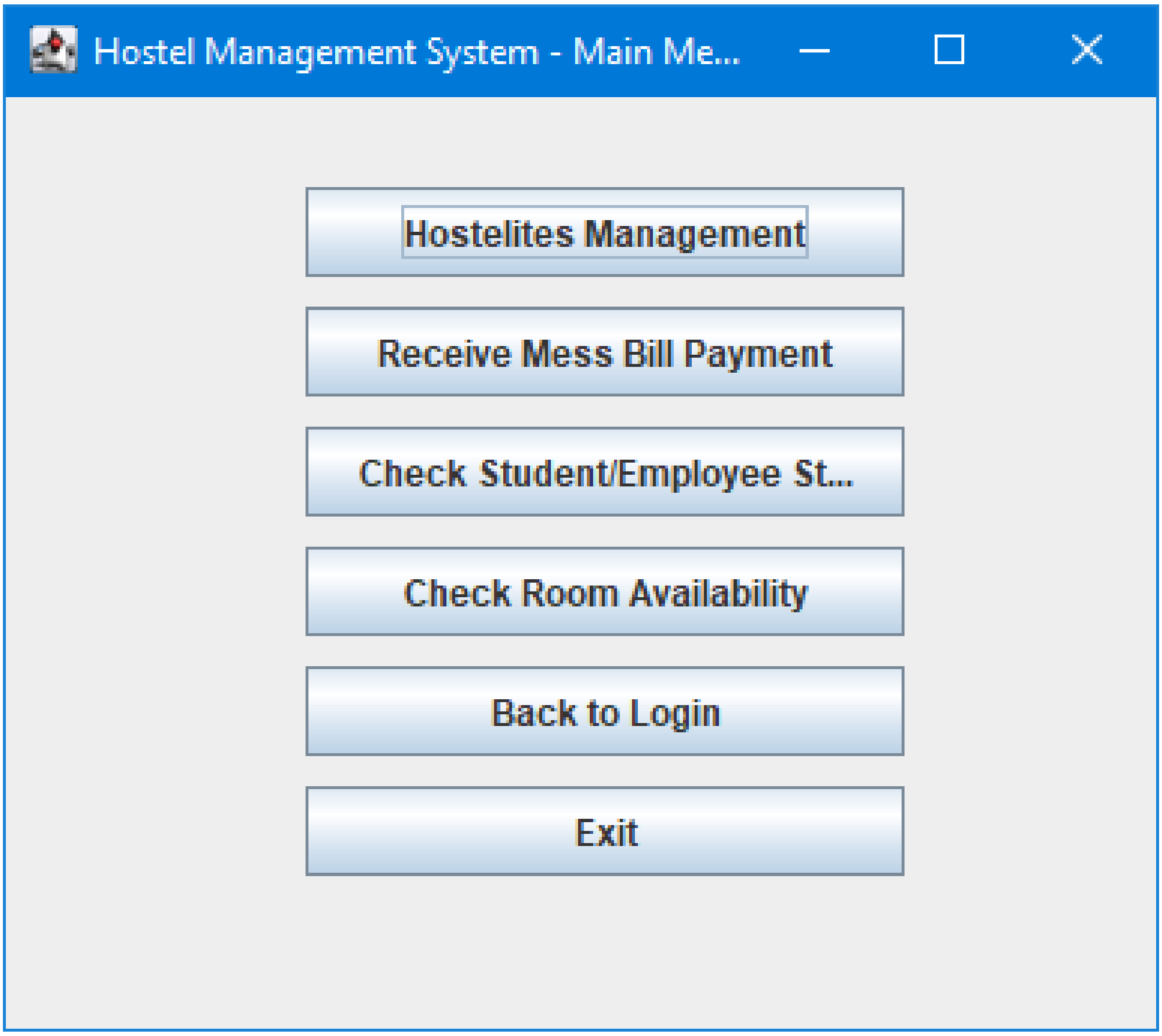
4.1 LOGIN PAGE



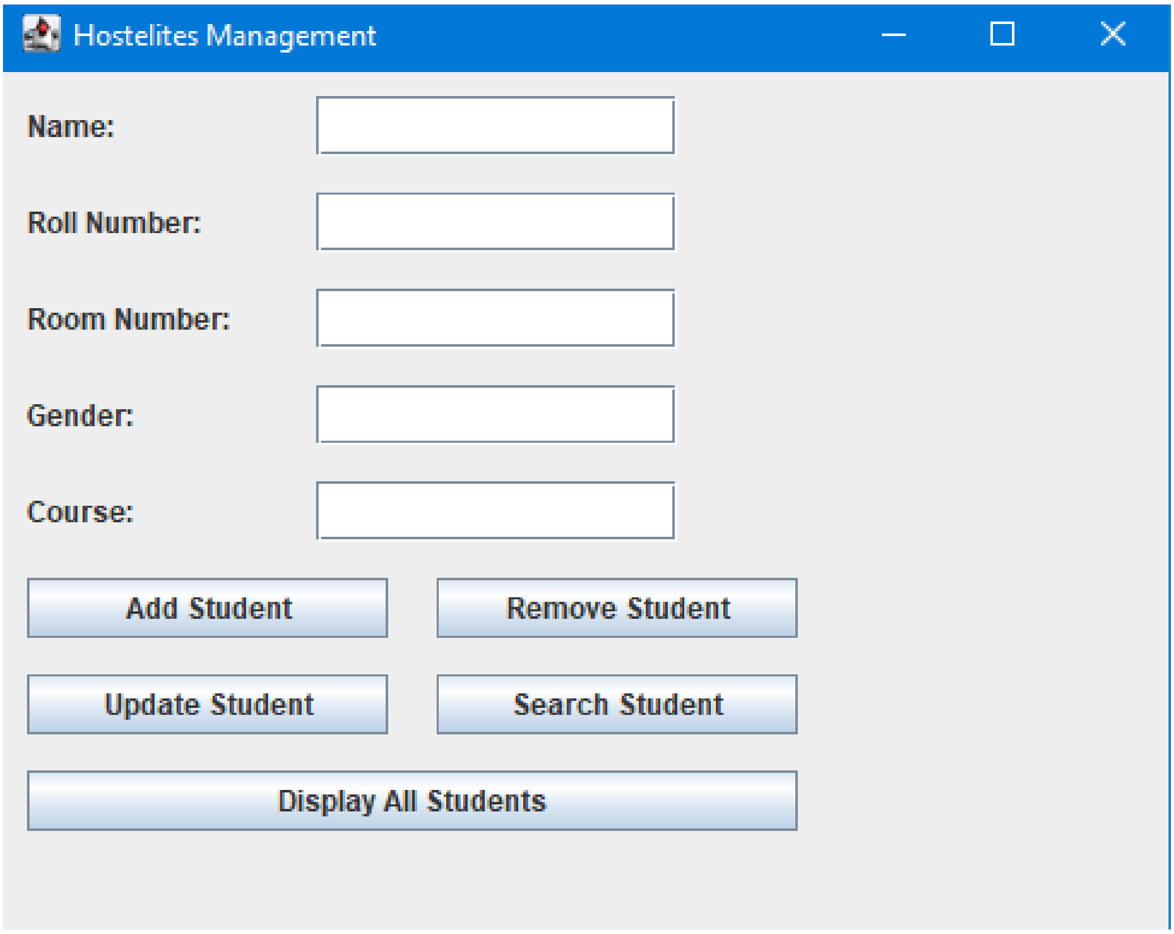
Username: admin

Password: •••••

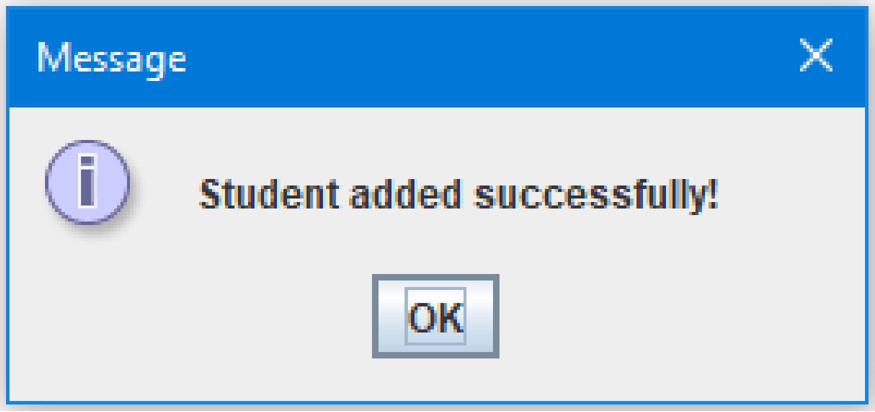
4.2 HOSTEL MANAGEMENT DASHBOARD


Hostel Management System - Main Me...
Hostelites Management
Receive Mess Bill Payment
Check Student/Employee St...
Check Room Availability
Back to Login
Exit

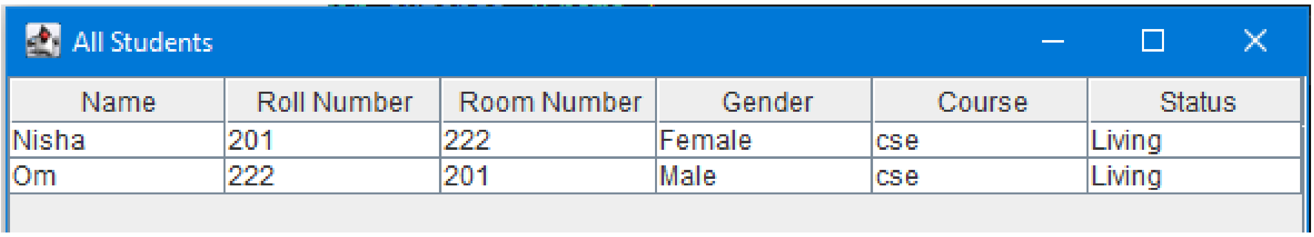
4.3 MANAGER FILLUP



4.4 STUDENT DETAILS

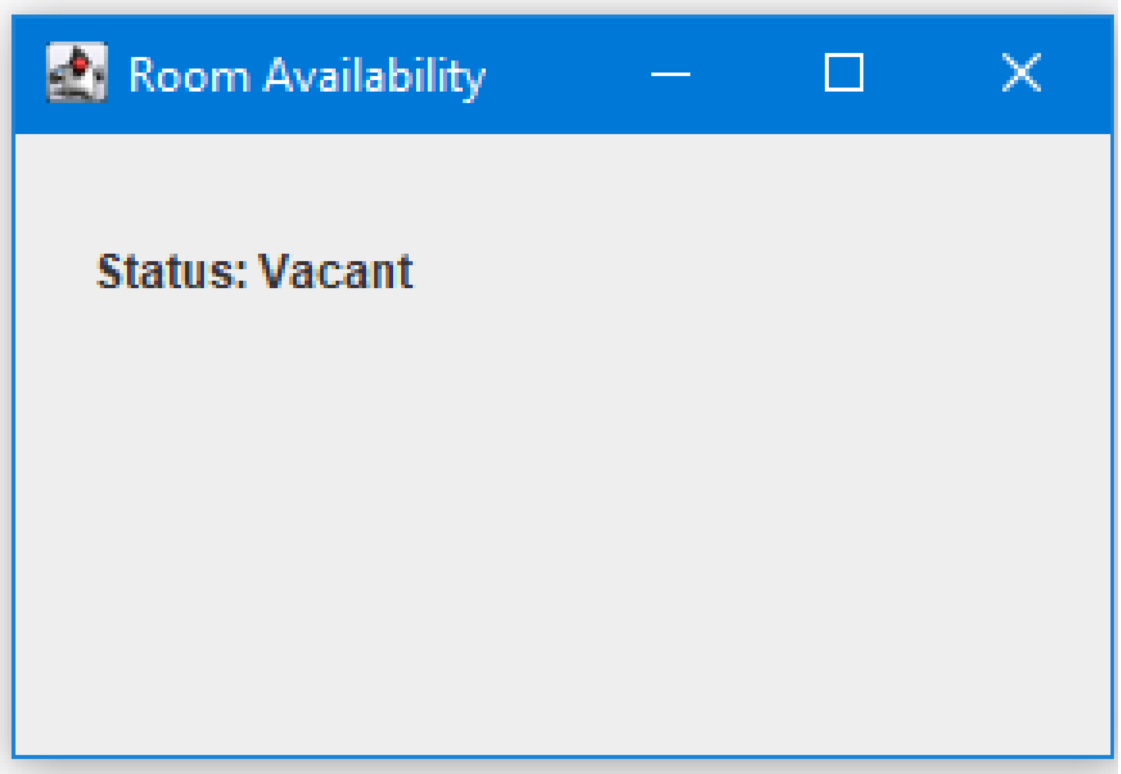


Student added successfully!



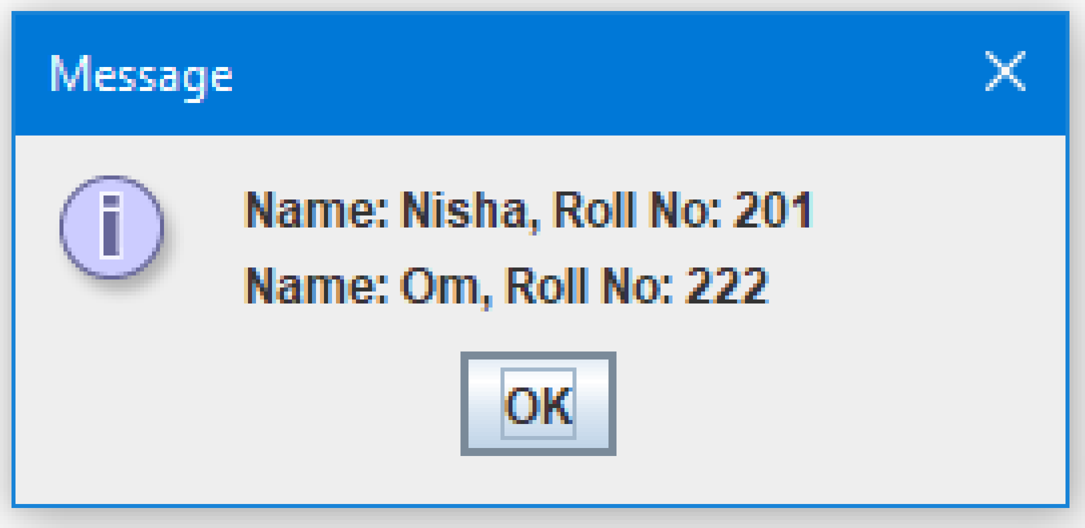
Name	Roll Number	Room Number	Gender	Course	Status
Nisha	201	222	Female	cse	Living
Om	222	201	Male	cse	Living

4.5 ROOM AVAILABILITY



Status: Vacant

4.6 STUDENTS STATUS



Name: Nisha, Roll No: 201

Name: Om, Roll No: 222

CONCLUSION

At the end of this program we have seen addition, updating, deleting, viewing and other features related to hostel management. This program is a simple and 'easier to manage' type for a manager of hostel. The program is done in Visual Studio Code along with JDBC and jar frame in order to present the application window.

6. REFERENCES

- **Java Swing Documentation**: offers comprehensive details on building graphical user interfaces in Java.

  - Java Swing on Oracle's Java Docs

- **MySQL Java Connector Documentation**: provides official guidelines for connecting Java applications to MySQL databases.

  - MySQL Connector/J Documentation

- **JDBC (Java Database Connectivity)** (Oracle): This documentation explains how to connect Java applications to a database using JDBC.

  - JDBC Basics

- **Java Platform SE API Documentation**: This provides comprehensive documentation on classes like `JFrame`, `JButton`, and `JOptionPane`.

  - Java SE API Documentation