

Name: Esha Khadeeja C

Roll No:19

Exp. no:20

Date:08/05/2024

Database Application Design

Aim: To design a database application using any front-end tool for any problem selected. The application constructed should have five or more tables.

Database Application: Hotel Management System

Front-End used: Java Swing

Swing is a Java Foundation Class library and an extension of the abstract window toolkit. Java swing offers much-improved functionality over AWT, new components, expanded components features and excellent event handling with drag and drop support.

Main features of Swing class are:

- **Lightweight components:** Swing components have their own view supported by java's look and feel classes.
- **Pluggable look and feel:** This feature enable the user to switch the look and feel of Swing components without restarting an application. The swing library supports components' look and feels that remain same across all platforms wherever the program runs.
- **Highly customizable:** Swing controls can be customized in a very easy way as visual appearance is independent of the internal representation.
- **Rich Controls:** Swing provides a rich set of advanced controls like Tree TabbedPane, slider, Colorpicker and table controls.

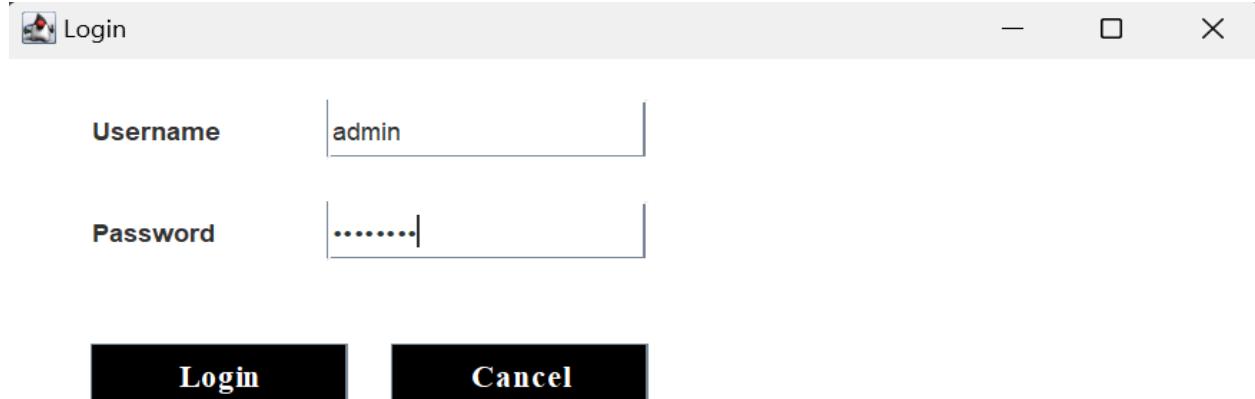
Back-end used: MySQL

MySQL is a popular open-source relational database management system(RDBMS). MySQL is fast, reliable, scalable and easy to use.

It was originally developed to handle large databases quickly and has been used in highly demanding production environments for many years.

MySQL's key benefits include performance, reliability, availability, security, flexibility and so on.

OUTPUT



A screenshot of a Windows-style login window titled "Login". The window has standard minimize, maximize, and close buttons at the top right. It contains two text input fields: "Username" with the value "admin" and "Password" with several dots indicating the password. Below the inputs are two large, dark rectangular buttons labeled "Login" and "Cancel".

Username	admin
Password

Login **Cancel**

Program code:

Login.java:

Login Page:

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.table.DefaultTableModel;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

public class Login extends JFrame implements ActionListener {

    JLabel l1, l2;
    JTextField t1;
    JPasswordField t2;
    JButton b1, b2;
    String user = "admin";
    String password = "password";

    public Login() {
        super("Login");

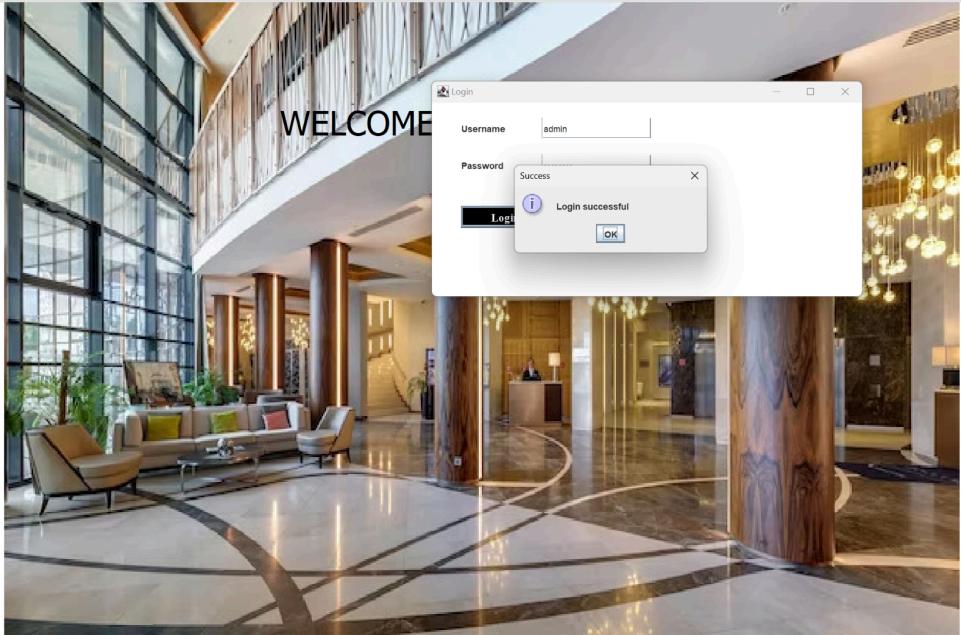
        setLayout(null);

        l1 = new JLabel("Username");
        l1.setBounds(40, 20, 100, 30);
        add(l1);

        l2 = new JLabel("Password");
        l2.setBounds(40, 70, 100, 30);
        add(l2);

        t1 = new JTextField();
        t1.setBounds(150, 20, 150, 30);
        add(t1);

        t2 = new JPasswordField();
```



```
t2.setBounds(150, 70, 150, 30);
add(t2);

b1 = new JButton("Login");
b1.setBounds(40, 140, 120, 30);
b1.setFont(new Font("serif", Font.BOLD, 15));
b1.addActionListener(this);
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);
add(b1);

b2 = new JButton("Cancel");
b2.setBounds(180, 140, 120, 30);
b2.setFont(new Font("serif", Font.BOLD, 15));
b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);
add(b2);

b2.addActionListener(this);

getContentPane().setBackground(Color.WHITE);

setVisible(true);
setSize(600, 300);
 setLocation(600, 350);
 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == b1) {
        String u = t1.getText();
        String v = new String(t2.getPassword()); // Get password as a String

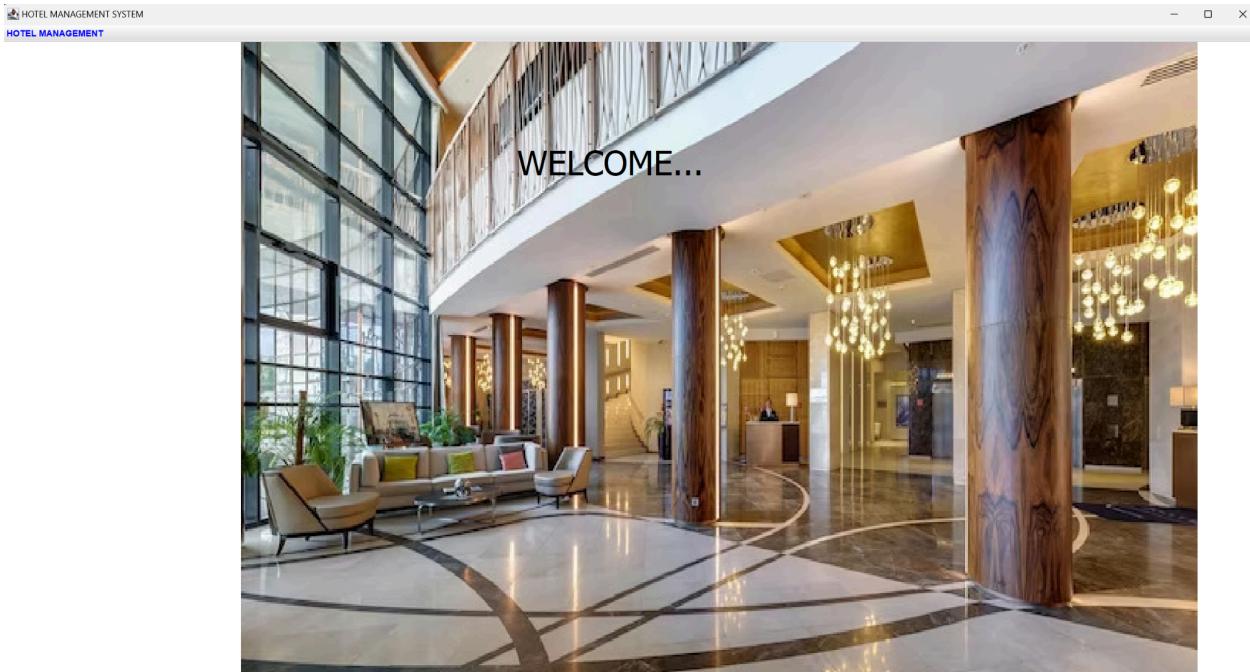
        if (u.equals(user) && v.equals(password)) {
            new Dashboard().setVisible(true);
            JOptionPane.showMessageDialog(this, "Login successful", "Success",
JOptionPane.INFORMATION_MESSAGE);
        } else {
            JOptionPane.showMessageDialog(this, "Login failed", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    } else if (ae.getSource() == b2) {
        System.exit(0); // Close the program when Cancel button is clicked
    }
}
```



```
        }
    }

public static void main(String[] arg) {
    new Login();
}
}
```

OUTPUT



Home Page:

```
class Dashboard extends JFrame implements ActionListener {  
    public Dashboard() {  
        super("HOTEL MANAGEMENT SYSTEM");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        setForeground(Color.RED);  
        setLayout(null);  
  
        ImageIcon imageIcon = new  
ImageIcon("C:\\\\Users\\\\ASUS\\\\vscode\\\\cli\\\\jdbc\\\\src\\\\hotel1.jpeg"); // Assuming the image is in  
the project directory  
        Image image = imageIcon.getImage().getScaledInstance(1300, 1000,  
Image.SCALE_DEFAULT);  
        ImageIcon scaledIcon = new ImageIcon(image);  
  
        // Create and add the JLabel with the image  
        JLabel imageLabel = new JLabel(scaledIcon);  
        imageLabel.setBounds(0, 0, 1950, 1000);  
        add(imageLabel);  
  
        JLabel AirlineManagementSystem = new JLabel("WELCOME...");  
        AirlineManagementSystem.setForeground(Color.BLACK);  
        AirlineManagementSystem.setFont(new Font("Tahoma", Font.PLAIN, 46));  
        AirlineManagementSystem.setBounds(700, 120, 1000, 85);  
        imageLabel.add(AirlineManagementSystem);  
  
        JMenuBar menuBar = new JMenuBar();  
        setJMenuBar(menuBar);  
  
        JMenu AirlineSystem = new JMenu("HOTEL MANAGEMENT");  
        AirlineSystem.setForeground(Color.BLUE);  
        menuBar.add(AirlineSystem);  
  
        JMenuItem Reception = new JMenuItem("RECEPTION");  
        Reception.addActionListener(this);  
        AirlineSystem.add(Reception);  
  
        setSize(1950, 1090);
```

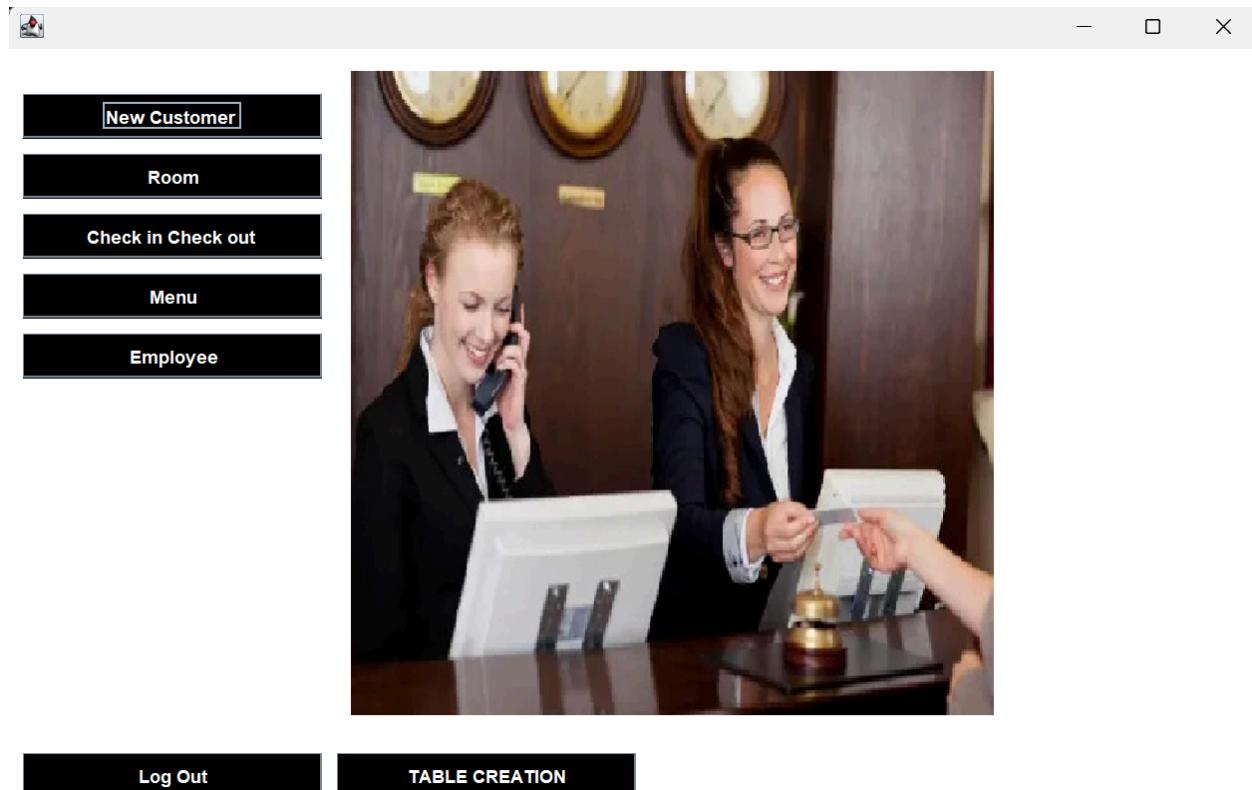
HOTEL MANAGEMENT SYSTEM
HOTEL MANAGEMENT
RECEPTION



```
setVisible(true);
getContentPane().setBackground(Color.WHITE);
}

public void actionPerformed(ActionEvent ae) {
    if (ae.getActionCommand().equals("RECEPTION")) {
        new Reception().setVisible(true);
        this.setVisible(false);
    }
}
```

OUTPUT



Reception Page:

```
class Reception extends JFrame {  
  
    public Reception() {  
        setBounds(530, 200, 850, 570);  
        JPanel contentPane = new JPanel();  
        setContentPane(contentPane);  
        contentPane.setLayout(null);  
  
        ImageIcon imageIcon = new  
ImageIcon("C:\\\\Users\\\\ASUS\\\\.vscode\\\\cli\\\\jdbc\\\\src\\\\ezgif-1-faaa98104a.jpg"); // Assuming the  
image is in the project directory  
        Image image = imageIcon.getImage().getScaledInstance(430, 430,  
Image.SCALE_DEFAULT);  
        ImageIcon scaledIcon = new ImageIcon(image);  
  
        // Create and add the JLabel with the image  
        JLabel imageLabel = new JLabel(scaledIcon);  
        imageLabel.setBounds(100, -120, 688, 700);  
        add(imageLabel);  
  
        JButton btnNewCustomerForm = new JButton("New Customer ");  
        btnNewCustomerForm.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent e) {  
                try {  
                    //Example: Open a new customer form  
                    Customer custom = new Customer();  
                    custom.setVisible(true);  
                    setVisible(false);  
                } catch (Exception e1) {  
                    e1.printStackTrace();  
                }  
            }  
        });  
        btnNewCustomerForm.setBounds(10, 30, 200, 30);  
        btnNewCustomerForm.setBackground(Color.BLACK);  
        btnNewCustomerForm.setForeground(Color.WHITE);  
        contentPane.add(btnNewCustomerForm);  
  
        // Add other buttons and their action listeners similarly  
  
        JButton btnNewButton = new JButton("Room");  
        btnNewButton.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent arg0) {
```



```
try{
    // Example: Open a room management interface
    Room room = new Room();
    room.setVisible(true);

    setVisible(false);
}
catch(Exception e){
    e.printStackTrace();
}

});

btnNewButton.setBounds(10, 70, 200, 30);
btnNewButton.setBackground(Color.BLACK);
btnNewButton.setForeground(Color.WHITE);
contentPane.add(btnNewButton);

JButton btnNewButton_1 = new JButton("Check in Check out ");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try{
            // Example: Open a check-in/check-out interface
            chk_in_out chk = new chk_in_out();
            chk.setVisible(true);
            setVisible(false);

        }
        catch (Exception e1){
            e1.printStackTrace();
        }
    }

});
btnNewButton_1.setBounds(10, 110, 200, 30);
btnNewButton_1.setBackground(Color.BLACK);
btnNewButton_1.setForeground(Color.WHITE);
contentPane.add(btnNewButton_1);

JButton btnNewButton_2 = new JButton("Menu");
btnNewButton_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try{

            // Example: Open a menu management interface

```



```
AddMenu menu = new AddMenu();
    menu.setVisible(true);
    setVisible(false);

}
catch (Exception e1){
    e1.printStackTrace();
}

}

});

btnNewButton_2.setBounds(10, 150, 200, 30);
btnNewButton_2.setBackground(Color.BLACK);
btnNewButton_2.setForeground(Color.WHITE);
contentPane.add(btnNewButton_2);

JButton btnNewButton_3 = new JButton("Employee");
btnNewButton_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        try{
            // Example: Open an employee management interface
            AddEmployee emp = new AddEmployee();
            emp.setVisible(true);

setVisible(false);
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }

});

btnNewButton_3.setBounds(10, 190, 200, 30);
btnNewButton_3.setBackground(Color.BLACK);
btnNewButton_3.setForeground(Color.WHITE);
contentPane.add(btnNewButton_3);

JButton btnNewButton_7 = new JButton("Log Out");
btnNewButton_7.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        try {
            //Example: Log out the user
            System.exit(1);
        setVisible(false);
    }
}
```



```

} catch (Exception e) {
    e.printStackTrace();
}
});

btnNewButton_7.setBounds(10, 470, 200, 30);
btnNewButton_7.setBackground(Color.BLACK);
btnNewButton_7.setForeground(Color.WHITE);
contentPane.add(btnNewButton_7);

JButton btnNewButton_8 = new JButton("TABLE CREATION");
btnNewButton_8.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        try {
            // Example: Log out the user
            Create_DB db=new Create_DB();
            db.createTables();
            setVisible(false);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
btnNewButton_8.setBounds(220, 470, 200, 30);
btnNewButton_8.setBackground(Color.BLACK);
btnNewButton_8.setForeground(Color.WHITE);
contentPane.add(btnNewButton_8);

setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);

addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e){

        new Dashboard().setVisible(true);
    }
});

getContentPane().setBackground(Color.WHITE);
setVisible(true);
}
}

```

OUTPUT

 Customer

Customer id	C3
Name	John
Date of Birth	2000-01-01
Room no.	A12
Phone no.	43195678

insert **delete** **display** **list of room available**

Customer Management Page:

```
class Customer extends JFrame implements ActionListener {  
  
    JLabel l1, l2, l3, l4, l5;  
    JTextField t1, t2, t3, t4, t5;  
    JButton b1, b2, b3, b4;  
    Connection conn;  
  
    public Customer() {  
        super("Customer");  
  
        setLayout(null);  
  
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);  
  
        addWindowListener(new WindowAdapter() {  
            public void windowClosing(WindowEvent e){  
  
                new Dashboard().setVisible(true);;  
            }  
        });  
  
        l1 = new JLabel("Customer id");  
        l1.setBounds(40, 20, 100, 30);  
        add(l1);  
  
        l2 = new JLabel("Name");  
        l2.setBounds(40, 70, 100, 30);  
        add(l2);  
  
        l3 = new JLabel("Date of Birth");  
        l3.setBounds(40, 120, 100, 30);  
        add(l3);  
  
        l4 = new JLabel("Room no.");  
        l4.setBounds(40, 170, 100, 30);  
        add(l4);  
  
        l5 = new JLabel("Phone no.");  
        l5.setBounds(40, 220, 100, 30);  
        add(l5);  
        t1 = new JTextField();
```

Customer Details

CID	Name	DOB	Room No	Phone No
C3	John	2000-01-01	A12	43195678

Customer

Customer id	<input type="text" value="C3"/>
Name	<input type="text" value="John"/>
Date of Birth	<input type="text" value="2000-01-01"/>
Room no.	<input type="text" value="A12"/>
Phone no.	<input type="text" value="43195678"/>

insert **delete** **display** **list of room available**

Customer Details

CID	Name	DOB	Room No	Phone No
C3	John	2000-01-01	A12	43195678

Customer

Customer id	<input type="text" value="C4"/>
Name	<input type="text" value="Rob"/>
Date of Birth	<input type="text" value="2004-01-01"/>
Room no.	<input type="text" value="A10"/>
Phone no.	<input type="text" value="43198532"/>

insert **delete** **display** **list of room available**

Message

Customer inserted successfully

OK

```
t1.setBounds(150, 20, 150, 30);
add(t1);

t2 = new JTextField();
t2.setBounds(150, 70, 150, 30);
add(t2);

t3 = new JTextField();
t3.setBounds(150, 120, 150, 30);
add(t3);

t4 = new JTextField();
t4.setBounds(150, 170, 150, 30);
add(t4);

t5 = new JTextField();
t5.setBounds(150, 220, 150, 30);
add(t5);

b1 = new JButton("insert");
b1.setBounds(40, 300, 120, 30);
b1.setFont(new Font("serif", Font.BOLD, 15));
b1.addActionListener(this);
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);
add(b1);

b2 = new JButton("delete");
b2.setBounds(180, 300, 120, 30);
b2.setFont(new Font("serif", Font.BOLD, 15));
b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);
add(b2);
b2.addActionListener(this);

b3 = new JButton("display");
b3.setBounds(320, 300, 120, 30);
b3.setFont(new Font("serif", Font.BOLD, 15));
b3.addActionListener(this);
b3.setBackground(Color.BLACK);
b3.setForeground(Color.WHITE);
add(b3);
```

Room No	Type	Status
A10	SINGLE	AVAILABLE
A11	SINGLE	AVAILABLE
A12	SUITE	BOOKED
A13	SUITE	AVAILABLE
A14	DELUXE	AVAILABLE
A15	DELUXE	AVAILABLE
A99	single	available
D23	DELUXE	AVAILABLE

Customer

Customer id: C4

Name: Rob

Date of Birth: 2004-01-01

Room no.: A10

Phone no.: 43198532

Buttons: insert, delete, display, list of room available

Room No	Type	Status
A10	SINGLE	AVAILABLE
A11	SINGLE	AVAILABLE
A12	SUITE	BOOKED
A13	SUITE	AVAILABLE
A14	DELUXE	AVAILABLE
A15	DELUXE	AVAILABLE
A99	single	available
D23	DELUXE	AVAILABLE

Customer

Customer id: C3

Name:

Date of Birth:

Room no.:

Phone no.:

Buttons: insert, delete, display, list of room available

```

b4 = new JButton("list of room available");
b4.setBounds(460, 300, 300, 30);
b4.setFont(new Font("serif", Font.BOLD, 15));
b4.addActionListener(this);
b4.setBackground(Color.BLACK);
b4.setForeground(Color.WHITE);
add(b4);

getContentPane().setBackground(Color.WHITE);

setVisible(true);
setSize(800, 400);
setLocation(400, 200);

// Connect to the database
try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root",
"d1e2v3a4");
} catch (SQLException e) {
    System.out.println("Failed to connect to the database.");
    e.printStackTrace();
}
}

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == b1) {
        insertCustomer();
    } else if (ae.getSource() == b2) {
        deleteCustomer();
    } else if (ae.getSource() == b3) {
        displayCustomers();
    } else if (ae.getSource() == b4) {
        displayRoomAvailability();
    }
}

public void insertCustomer() {
    try {
        PreparedStatement pstmt = conn.prepareStatement("INSERT INTO
CUSTOMER(CID,CNAME,DOB,ROOM_NO,PH_NO) VALUES(?,?,?,?,?,?)");
        pstmt.setString(1, t1.getText());
        pstmt.setString(2, t2.getText());
        pstmt.setString(3, t3.getText());
        pstmt.setString(4, t4.getText());
    }
}

```

Customer

Customer id	C3
Name	<input type="text"/>
Date of Birth	<input type="text"/>
Room no.	<input type="text"/>
Phone no.	<input type="text"/>

Message X

i Customer deleted successfully **OK**

insert **delete** **display** **list of room available**

Customer Details

CID	Name	DOB	Room No	Phone No
C4	Rob	2004-01-01	A10	43198532

Customer

Customer id	C3
Name	<input type="text"/>
Date of Birth	<input type="text"/>
Room no.	<input type="text"/>
Phone no.	<input type="text"/>

insert **delete** **display** **list of room available**

```

        pstmt.setString(5, t5.getText());
        pstmt.executeUpdate();

        // Update room status to booked
        updateRoomStatus(t4.getText(), "BOOKED");

        JOptionPane.showMessageDialog(this, "Customer inserted successfully");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

public void deleteCustomer() {
    try {
        PreparedStatement pstmt = conn.prepareStatement("DELETE FROM CUSTOMER
WHERE CID = ?");

        pstmt.setString(1, t1.getText());
        int rowsAffected = pstmt.executeUpdate();
        if (rowsAffected > 0) {
            // Update room status to available
            updateRoomStatus(t4.getText(), "AVAILABLE");
            JOptionPane.showMessageDialog(this, "Customer deleted successfully");
        } else {
            JOptionPane.showMessageDialog(this, "Customer with ID " + t1.getText() + " not
found", "Error", JOptionPane.ERROR_MESSAGE);
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

public void updateRoomStatus(String roomNo, String status) {
    try {
        PreparedStatement pstmt = conn.prepareStatement("UPDATE ROOM SET STATUS
= ? WHERE ROOM_NO = ?");

        pstmt.setString(1, status);
        pstmt.setString(2, roomNo);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Error updating room status: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```



```
        }

    }

public void displayCustomers() {
    try {
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM CUSTOMER");

        // Create a DefaultTableModel to hold your data
        DefaultTableModel model = new DefaultTableModel();
        model.addColumn("CID");
        model.addColumn("Name");
        model.addColumn("DOB");
        model.addColumn("Room No");
        model.addColumn("Phone No");

        // Populate the model with data from the ResultSet
        while (rs.next()) {
            Object[] row = {
                rs.getString(1),
                rs.getString(2),
                rs.getString(3),
                rs.getString(4),
                rs.getString(5)
            };
            model.addRow(row);
        }

        // Create a JTable with the model
        JTable table = new JTable(model);

        // Wrap the table in a scroll pane
        JScrollPane scrollPane = new JScrollPane(table);

        // Create a JFrame to hold the scroll pane
        JFrame frame = new JFrame("Customer Details");
        frame.add(scrollPane);
        frame.pack();
        frame.setVisible(true);
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}
```



```
}

public void displayRoomAvailability() {
    try {
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT ROOM_NO, ROOM_TYPE, STATUS
FROM ROOM");

        // Create a DefaultTableModel to hold your data
        DefaultTableModel model = new DefaultTableModel();
        model.addColumn("Room No");
        model.addColumn("Type");
        model.addColumn("Status");

        // Populate the model with data from the ResultSet
        while (rs.next()) {
            Object[] row = {
                rs.getString(1),
                rs.getString(2),
                rs.getString(3)
            };
            model.addRow(row);
        }

        // Create a JTable with the model
        JTable table = new JTable(model);

        // Wrap the table in a scroll pane
        JScrollPane scrollPane = new JScrollPane(table);

        // Create a JFrame to hold the scroll pane
        JFrame frame = new JFrame("Room Availability");
        frame.add(scrollPane);
        frame.pack();
        frame.setVisible(true);
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

}
```

OUTPUT

Room Details

Room No.	Type	Price	Status
A10	SINGLE	5000.0	AVAILABLE
A11	SINGLE	5000.0	AVAILABLE
A12	SUITE	7000.0	BOOKED
A13	SUITE	7000.0	AVAILABLE
A14	DELUXE	10000.0	AVAILABLE
A15	DELUXE	10000.0	AVAILABLE
A99	single	9000.0	available
D23	DELUXE	10000.0	AVAILABLE

Room

Room no.	
Type	
Price	
Status	

insert display delete

Room Management Page:

```
class Room extends JFrame implements ActionListener {  
  
    JLabel l1, l2, l3, l4;  
    JTextField t1, t2, t3, t4;  
    JButton b1, b2, b3;  
    Connection conn;  
  
    public Room() {  
        super("Room");  
  
        setLayout(null);  
  
        setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);  
  
        addWindowListener(new WindowAdapter() {  
            public void windowClosing(WindowEvent e){  
  
                new Dashboard().setVisible(true);  
            }  
        });  
  
        l1 = new JLabel("Room no.");  
        l1.setBounds(40, 20, 100, 30);  
        add(l1);  
  
        l2 = new JLabel("Type");  
        l2.setBounds(40, 70, 100, 30);  
        add(l2);  
  
        l3 = new JLabel("Price");  
        l3.setBounds(40, 120, 100, 30);  
        add(l3);  
  
        l4 = new JLabel("Status");  
        l4.setBounds(40, 170, 100, 30);  
        add(l4);  
  
        t1 = new JTextField();  
        t1.setBounds(150, 20, 150, 30);  
        add(t1);  
  
        t2 = new JTextField();  
        t2.setBounds(150, 70, 150, 30);
```

Room Details

Room No.	Type	Price	Status
A10	SINGLE	5000.0	AVAILABLE
A11	SINGLE	5000.0	AVAILABLE
A12	SUITE	7000.0	BOOKED
A13	SUITE	7000.0	AVAILABLE
A14	DELUXE	10000.0	AVAILABLE
A15	DELUXE	10000.0	AVAILABLE
A99	single	9000.0	available
D23	DELUXE	10000.0	AVAILABLE

Room

Room no.	<input type="text" value="A100"/>
Type	<input type="text" value="SUIT"/>
Price	<input type="text" value="7000"/>
Status	<input type="text" value="AVAILABLE"/>

insert
display
delete

Room

Room no.	<input type="text" value="A100"/>
Type	<input type="text" value="SUIT"/>
Price	<input type="text" value="7000"/>
Status	<input type="text" value="AVAILABLE"/>

insert
display
delete

Message

i Room inserted successfully

OK

```
add(t2);

t3 = new JTextField();
t3.setBounds(150, 120, 150, 30);
add(t3);

t4 = new JTextField();
t4.setBounds(150, 170, 150, 30);
add(t4);

b1 = new JButton("insert");
b1.setBounds(40, 250, 120, 30);
b1.setFont(new Font("serif", Font.BOLD, 15));
b1.addActionListener(this);
b1.setBackground(Color.BLACK);
b1.setForeground(Color.WHITE);
add(b1);

b2 = new JButton("display");
b2.setBounds(180, 250, 120, 30);
b2.setFont(new Font("serif", Font.BOLD, 15));
b2.setBackground(Color.BLACK);
b2.setForeground(Color.WHITE);
add(b2);
b2.addActionListener(this);

b3 = new JButton("delete");
b3.setBounds(320, 250, 120, 30);
b3.setFont(new Font("serif", Font.BOLD, 15));
b3.setBackground(Color.BLACK);
b3.setForeground(Color.WHITE);
add(b3);
b3.addActionListener(this);

getContentPane().setBackground(Color.WHITE);

setVisible(true);
setSize(600, 300);
 setLocation(600, 350);

// Connect to the database
try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root",
    "d1e2v3a4");
```

Room Details

Room No.	Type	Price	Status
A10	SINGLE	5000.0	AVAILABLE
A100	SUIT	7000.0	AVAILABLE
A11	SINGLE	5000.0	AVAILABLE
A12	SUITE	7000.0	BOOKED
A13	SUITE	7000.0	AVAILABLE
A14	DELUXE	10000.0	AVAILABLE
A15	DELUXE	10000.0	AVAILABLE
A99	single	9000.0	available
D23	DELUXE	10000.0	AVAILABLE

Room

Room no.	A100
Type	SUIT
Price	7000
Status	AVAILABLE

insert display delete

Room

Room no.	A100
Type	SUIT
Price	7000
Status	AVAILABLE

Message

 Room deleted successfully

OK

insert display delete

```

} catch (SQLException e) {
    System.out.println("Failed to connect to the database.");
    e.printStackTrace();
}
}

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == b1) {
        insertRoom();
    } else if (ae.getSource() == b2) {
        displayRooms();
    }
    else if (ae.getSource() == b3) {
        deleteRoom();
    }
}

public void deleteRoom(){
    try {
        PreparedStatement pstmt = conn.prepareStatement("DELETE FROM ROOM
WHERE ROOM_NO=?");
        pstmt.setString(1, t1.getText());

        pstmt.executeUpdate();
        JOptionPane.showMessageDialog(this, "Room deleted successfully");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

public void insertRoom() {
    try {
        PreparedStatement pstmt = conn.prepareStatement("INSERT INTO
ROOM(ROOM_NO,ROOM_TYPE,PRICE,STATUS) VALUES(?, ?, ?, ?)");
        pstmt.setString(1, t1.getText());
        pstmt.setString(2, t2.getText());
        pstmt.setDouble(3, Double.parseDouble(t3.getText()));
        pstmt.setString(4, t4.getText());
        pstmt.executeUpdate();
        JOptionPane.showMessageDialog(this, "Room inserted successfully");
    } catch (SQLException e) {

```



```
JOptionPane.showMessageDialog(this, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
}

}

public void displayRooms() {
try {
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM ROOM");

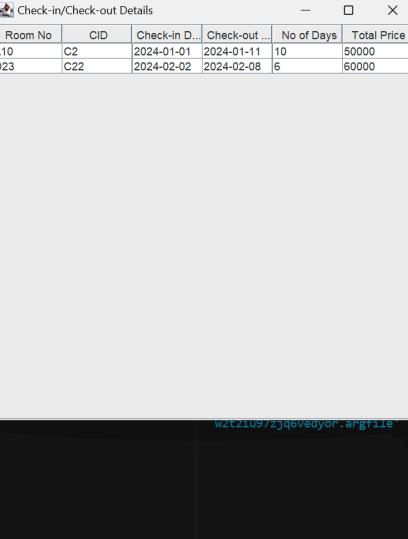
// Create a DefaultTableModel to hold your data
DefaultTableModel model = new DefaultTableModel();
model.addColumn("Room No");
model.addColumn("Type");
model.addColumn("Price");
model.addColumn("Status");

// Populate the model with data from the ResultSet
while (rs.next()) {
Object[] row = {
rs.getString(1),
rs.getString(2),
rs.getDouble(3),
rs.getString(4)
};
model.addRow(row);
}
// Create a JTable with the model
JTable table = new JTable(model);

// Wrap the table in a scroll pane
JScrollPane scrollPane = new JScrollPane(table);

// Create a JFrame to hold the scroll pane
JFrame frame = new JFrame("Room Details");
frame.add(scrollPane);
frame.pack();
frame.setVisible(true);
} catch (SQLException e) {
JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
}
}
```

OUTPUT



ADD CHECKIN DETAILS

ROOM_NO	<input type="text"/>
CID	<input type="text"/>
CheckInDate	<input type="text"/>
No Of days	<input type="text"/>

SAVE **DELETE** **DISPLAY ALL** **BILL**

Check-in Check-out Page:

```
class chk_in_out extends JFrame{ //Third Frame

    Connection conn; // Declare connection at the class level
    Statement s;
    public void conn() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root",
"d1e2v3a4");
            System.out.println("connected");
        } catch (Exception e) {
            System.out.println(e);
        }
    }

    JTextField textField,textField_1,textField_2,textField_3,textField_4,textField_5,textField_6;

    public chk_in_out(){

        setDefaultCloseOperation(JFrame.DO NOTHING_ON_CLOSE);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e){

                new Dashboard().setVisible(true);
            }
        });
        getContentPane().setForeground(Color.BLUE);
        getContentPane().setBackground(Color.WHITE);
        setTitle("CHECK IN CHECK OUT DETAILS");

        setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
        setSize(778,486);
        getContentPane().setLayout(null);
        JLabel roomno = new JLabel("ROOM_NO");
        roomno.setFont(new Font("Tahoma", Font.PLAIN, 17));
        roomno.setBounds(60, 30, 150, 27);
        add(roomno);
        setVisible(true);

        textField = new JTextField();
        textField.setBounds(200, 30, 150, 27);
```

Check-in/Check-out Details

Room No	CID	Check-in D...	Check-out ...	No of Days	Total Price
A10	C2	2024-01-01	2024-01-11	10	50000
D23	C22	2024-02-02	2024-02-08	6	60000

CHECK IN CHECK OUT DETAILS

ROOM_NO

CID

CheckInDate

Message

No Of days

Value Added

SAVE
DELETE
DISPLAY ALL
BILL

Check-in/Check-out Details

Room No	CID	Check-in D...	Check-out ...	No of Days	Total Price
A10	C2	2024-01-01	2024-01-11	10	50000
D23	C22	2024-02-02	2024-02-08	6	60000
A13	C1	2024-07-02	2024-07-04	2	14000

CHECK IN CHECK OUT DETAILS

ROOM_NO

CID

CheckInDate

No Of days

SAVE
DELETE
DISPLAY ALL
BILL

```
add(textField);

JButton Next = new JButton("SAVE");
Next.setBounds(100, 420, 150, 30);
Next.setBackground(Color.BLACK);
Next.setForeground(Color.WHITE);
add(Next);

JButton Delete = new JButton("DELETE");
Delete.setBounds( 350, 420, 150, 30);
Delete.setBackground(Color.BLACK);
Delete.setForeground(Color.WHITE);
add(Delete);

JButton Display = new JButton("DISPLAY ALL");
Display.setBounds( 600, 420, 150, 30);
Display.setBackground(Color.BLACK);
Display.setForeground(Color.WHITE);
add(Display);

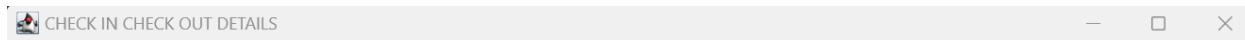
JButton Search = new JButton("BILL");
Search.setBounds( 600, 460, 150, 30);
Search.setBackground(Color.BLACK);
Search.setForeground(Color.WHITE);
add(Search);

JLabel CID = new JLabel("CID");
CID.setFont(new Font("Tahoma", Font.PLAIN, 17));
CID.setBounds(60, 80, 150, 27);
add(CID);

textField_1 = new JTextField();
textField_1.setBounds(200, 80, 150, 27);
add(textField_1);

JLabel cid = new JLabel("CheckInDate");
cid.setFont(new Font("Tahoma", Font.PLAIN, 17));
cid.setBounds(60, 140, 150, 27);
add(cid);

textField_2 = new JTextField();
textField_2.setBounds(200, 140, 150, 27);
add(textField_2);
```



ADD CHECKIN DETAILS

ROOM_NO

A13

CID

C1

CheckInDate

2024-07-02

No Of days

2

Message



ENTRY Deleted

OK

SAVE

DELETE

DISPLAY ALL

BILL

Check-in/Check-out Details					
Room No	CID	Check-in D...	Check-out ...	No of Days	Total Price
A10	C2	2024-01-01	2024-01-11	10	50000
D23	C22	2024-02-02	2024-02-08	6	60000

Check-in/Check-out Details

ROOM_NO

A13

CID

C1

CheckInDate

2024-07-02

No Of days

2

ADD CHECKIN DETAILS

SAVE

DELETE

DISPLAY ALL

BILL

```

//JLabel cod = new JLabel("CheckOutDate");
// cod.setFont(new Font("Tahoma", Font.PLAIN, 17));
// cod.setBounds(60, 220, 150, 27);
// add(cod);

// textField_3 = new JTextField();
// textField_3.setBounds(200, 220, 150, 27);
// add(textField_3);

JLabel nod = new JLabel("No Of days");
nod.setFont(new Font("Tahoma", Font.PLAIN, 17));
nod.setBounds(60, 250, 150, 27);
add(nod);

textField_4 = new JTextField();
textField_4.setBounds(200, 250, 150, 27);
add(textField_4);

setVisible(true);

JLabel Addcheck = new JLabel("ADD CHECKIN DETAILS");
Addcheck.setForeground(Color.BLUE);
Addcheck.setFont(new Font("Tahoma", Font.PLAIN, 31));
Addcheck.setBounds(450, 24, 442, 35);
add(Addcheck);

Next.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        String roomno = textField.getText();
        String CID = textField_1.getText();
        String cid = textField_2.getText();
        String nod = textField_4.getText();

        try {
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root", "d1e2v3a4");

            // Use prepared statement to prevent SQL injection
            PreparedStatement ps = conn.prepareStatement("SELECT price FROM room
WHERE room_no=?");
            ps.setString(1, roomno);

```

BILL

Room No	CID	Check-in D...	Check-out ...	No of Days	Total Price
D23	C22	2024-02-02	2024-02-08	6	60000

CHECK IN CHECK OUT DETAILS

ROOM_NO

CID

CheckInDate

No Of days

ADD CHECKIN DETAILS

SAVE **DELETE** **DISPLAY ALL**
BILL

```

ResultSet rs = ps.executeQuery();

// Check if ResultSet has any rows
if (rs.next()) {
    int price = rs.getInt("price");
    int total = price * Integer.parseInt(nod);

    // Calculate the checkout date by adding number of days to check-in date
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Date checkInDate = sdf.parse(cid);
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(checkInDate);
    calendar.add(Calendar.DAY_OF_MONTH, Integer.parseInt(nod));
    Date checkOutDate = calendar.getTime();
    String cod = sdf.format(checkOutDate);

    // Use prepared statement to prevent SQL injection
    PreparedStatement psInsert = conn.prepareStatement("INSERT INTO
CheckIN_CheckOUT VALUES (?, ?, ?, ?, ?, ?, ?)");
    psInsert.setString(1, roomno);
    psInsert.setString(2, CID);
    psInsert.setString(3, cid);
    psInsert.setString(4, cod); // Set the calculated checkout date
    psInsert.setString(5, nod);
    psInsert.setInt(6, total);

    // Execute the insert statement
    psInsert.executeUpdate();

    JOptionPane.showMessageDialog(null, "Value Added");

    //chk_in_out chk = new chk_in_out();
    //chk.setVisible(true);
    //setVisible(false); // Assuming you want to hide the window after adding the
value
} else {
    JOptionPane.showMessageDialog(null, "Room not found", "Error",
JOptionPane.ERROR_MESSAGE);
}
} catch (SQLException | NumberFormatException | java.text.ParseException e) {
    JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    e.printStackTrace();
}

```



```

        }
    });

Delete.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae){
        //String roomno = textField.getText();
        String CID = textField_1.getText();
        // String cid = textField_2.getText();
        // String cod = textField_3.getText();
        // String nod = textField_4.getText();

        try {
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root", "d1e2v3a4");
            Statement s = conn.createStatement();
            String str = "DELETE from CheckIN_CheckOUT where CID='"+CID+"'";
            s.executeUpdate(str);
            JOptionPane.showMessageDialog(chk_in_out.this,"ENTRY Deleted");
            //setVisible(false);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

Display.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        try {
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root", "d1e2v3a4");
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM checkin_checkout");

            // Create a DefaultTableModel to hold your data
            DefaultTableModel model = new DefaultTableModel();
            model.addColumn("Room No");
            model.addColumn("CID");
            model.addColumn("Check-in Date");
            model.addColumn("Check-out Date");
            model.addColumn("No of Days");

```



```

model.addColumn("Total Price");

// Populate the model with data from the ResultSet
while (rs.next()) {
    Object[] row = {
        rs.getString(1),
        rs.getString(2),
        rs.getString(3),
        rs.getString(4),
        rs.getString(5),
        rs.getString(6)
    };
    model.addRow(row);
}

// Create a JTable with the model
JTable table = new JTable(model);

// Wrap the table in a scroll pane
 JScrollPane scrollPane = new JScrollPane(table);

// Create a JFrame to hold the scroll pane
 JFrame frame = new JFrame("Check-in/Check-out Details");
 frame.add(scrollPane);
 frame.pack();
 frame.setVisible(true);

} catch (SQLException e) {
    JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
}
}

};

Search.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        try {
            String CID = textField_1.getText();

            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root", "d1e2v3a4");
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM checkin_checkout where
CID = "+CID+"");

```



```

// Create a DefaultTableModel to hold your data
DefaultTableModel model = new DefaultTableModel();
model.addColumn("Room No");
model.addColumn("CID");
model.addColumn("Check-in Date");
model.addColumn("Check-out Date");
model.addColumn("No of Days");
model.addColumn("Total Price");

// Populate the model with data from the ResultSet
while (rs.next()) {
    Object[] row = {
        rs.getString(1),
        rs.getString(2),
        rs.getString(3),
        rs.getString(4),
        rs.getString(5),
        rs.getString(6)
    };
    model.addRow(row);
}

// Create a JTable with the model
JTable table = new JTable(model);
// Wrap the table in a scroll pane
 JScrollPane scrollPane = new JScrollPane(table);

// Create a JFrame to hold the scroll pane
JFrame frame = new JFrame("BILL");
frame.add(scrollPane);
frame.pack();
frame.setVisible(true);
} catch (SQLException e) {
    JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
}
}

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(900,600);
setVisible(true);
 setLocation(530,200);
}
}
}

```

OUTPUT

 ADD EMPLOYEE DETAILS

— □ ×

NAME

EID

Designation

DOB

ADD EMPLOYEE DETAILS

SAVE **DELETE** **DISPLAY**

SEARCH EID

Employee Management Page:

```
class AddEmployee extends JFrame { // Third Frame

    Connection conn; // Declare connection at the class level

    public void conn() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root",
"d1e2v3a4");
            Statement s = conn.createStatement();
            System.out.println("connected");
        } catch (Exception e) {
            System.out.println(e);
        }
    }

    JTextField textField, textField_1, textField_2, textField_3, textField_4, textField_5,
textField_6;

    public AddEmployee() {

        setDefaultCloseOperation(JFrame.DO NOTHING ON CLOSE);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e){

                new Dashboard().setVisible(true);
            }
        });
        getContentPane().setForeground(Color.BLUE);
        getContentPane().setBackground(Color.WHITE);
        setTitle("ADD EMPLOYEE DETAILS");

        setDefaultCloseOperation(JFrame.HIDE ON CLOSE);
        setSize(778, 486);
        getContentPane().setLayout(null);

        JLabel name = new JLabel("NAME");
        name.setFont(new Font("Tahoma", Font.PLAIN, 17));
        name.setBounds(60, 30, 150, 27);
        add(name);
```

Name	EID	Designation	DOB
abc	E45	job	0204-09-08
john	E59	job	1999-01-01

ADD EMPLOYEE DETAILS

NAME

EID

Designation

DOB

SAVE **DELETE** **DISPLAY** **SEARCH EID**

```
textField = new JTextField();
textField.setBounds(200, 30, 150, 27);
add(textField);

JButton Next = new JButton("SAVE");
Next.setBounds(100, 420, 150, 30);
Next.setBackground(Color.BLACK);
Next.setForeground(Color.WHITE);
add(Next);

JButton Delete = new JButton("DELETE");
Delete.setBounds(350, 420, 150, 30);
Delete.setBackground(Color.BLACK);
Delete.setForeground(Color.WHITE);
add(Delete);

JButton Display = new JButton("DISPLAY");
Display.setBounds(600, 420, 150, 30);
Display.setBackground(Color.BLACK);
Display.setForeground(Color.WHITE);
add(Display);

JButton Search = new JButton("SEARCH EID");
Search.setBounds(600, 460, 150, 30);
Search.setBackground(Color.BLACK);
Search.setForeground(Color.WHITE);
add(Search);

JLabel EID = new JLabel("EID");
EID.setFont(new Font("Tahoma", Font.PLAIN, 17));
EID.setBounds(60, 80, 150, 27);
add(EID);

textField_1 = new JTextField();
textField_1.setBounds(200, 80, 150, 27);
add(textField_1);

JLabel Designation = new JLabel("Designation");
Designation.setFont(new Font("Tahoma", Font.PLAIN, 17));
Designation.setBounds(60, 140, 150, 27);
add(Designation);

textField_2 = new JTextField();
textField_2.setBounds(200, 140, 150, 27)
```

ADD EMPLOYEE DETAILS

NAME	<input type="text" value="rob"/>
EID	<input type="text" value="E1"/>
Designation	<input type="text" value="manager"/>
DOB	<input type="text" value="1998-01-01"/>

ADD EMPLOYEE DETAILS

Message

i Employee Added

OK

SAVE **DELETE** **DISPLAY** **SEARCH EID**

Employee Details

Name	EID	Designation	DOB
rob	E1	manager	1998-01-01
abc	E45	job	0204-09-08
john	E69	job	1999-01-01

ADD EMPLOYEE DETAILS

NAME	<input type="text" value="rob"/>
EID	<input type="text" value="E1"/>
Designation	<input type="text" value="manager"/>
DOB	<input type="text" value="1998-01-01"/>

ADD EMPLOYEE DETAILS

w2t2iu97zjq6vedyor.argvfile'

SAVE **DELETE** **DISPLAY** **SEARCH EID**

```

add(textField_2);

JLabel DOB = new JLabel("DOB");
DOB.setFont(new Font("Tahoma", Font.PLAIN, 17));
DOB.setBounds(60, 220, 150, 27);
add(DOB);

textField_3 = new JTextField();
textField_3.setBounds(200, 220, 150, 27);
add(textField_3);

setVisible(true);
JLabel AddPassengers = new JLabel("ADD EMPLOYEE DETAILS");
AddPassengers.setForeground(Color.BLUE);
AddPassengers.setFont(new Font("Tahoma", Font.PLAIN, 31));
AddPassengers.setBounds(450, 24, 442, 35);
add(AddPassengers);

Next.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        String name = textField.getText();
        String EID = textField_1.getText();
        String Designation = textField_2.getText();
        String DOB = textField_3.getText();

        try {
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hotel_db", "root", "d1e2v3a4");
            Statement s = conn.createStatement();
            String str = "INSERT INTO employee values( " + name + ", " + EID + ", " +
Designation + "," + DOB + ")";
            s.executeUpdate(str);
            JOptionPane.showMessageDialog(AddEmployee.this, "Employee Added");
            //setVisible(false);
            //AddEmployee emp = new AddEmployee();
            //emp.setVisible(true);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

```

ADD EMPLOYEE DETAILS

NAME	<input type="text"/>
EID	<input type="text" value="E45"/>
Designation	<input type="text"/>
DOB	<input type="text"/>

Message

Employee Deleted

OK

SAVE
DELETE
DISPLAY

SEARCH EID

Employee Details

Name	EID	Designation	DOB
rob	E1	manager	1999-01-01
john	E59	job	1999-01-01

ADD EMPLOYEE DETAILS

NAME	<input type="text"/>
EID	<input type="text" value="E45"/>
Designation	<input type="text"/>
DOB	<input type="text"/>

SAVE
DELETE
DISPLAY

SEARCH EID

```
Delete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        String EID = textField_1.getText();

        try {
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hotel_db", "root", "d1e2v3a4");
            Statement s = conn.createStatement();
            String str = "DELETE from employee where EID='" + EID + "'";

            s.executeUpdate(str);
            JOptionPane.showMessageDialog(AddEmployee.this, "Employee Deleted");
            // setVisible(false);
            // AddEmployee emp = new AddEmployee();
            // emp.setVisible(true);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

Display.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        try {
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hotel_db", "root", "d1e2v3a4");
            Statement s = conn.createStatement();
            ResultSet rs = s.executeQuery("SELECT * FROM EMPLOYEE");

            // Create a DefaultTableModel to hold your data
            DefaultTableModel model = new DefaultTableModel();
            model.addColumn("Name");
            model.addColumn("EID");
            model.addColumn("Designation");
            model.addColumn("DOB");

            // Populate the model with data from the ResultSet
            while (rs.next()) {
                Object[] row = {
                    rs.getString(1),
                    rs.getString(2),
                    rs.getString(3),
                    rs.getString(4)
                }
            }
        }
    }
});
```

Name	EID	Designation	DOB
rob	E1	manager	1998-01-01

ADD EMPLOYEE DETAILS

NAME

EID E1

Designation

DOB

SAVE **DELETE** **DISPLAY** **SEARCH EID**

```

    };

        model.addRow(row);
    }

    // Create a JTable with the model
    JTable table = new JTable(model);

    // Wrap the table in a scroll pane
    JScrollPane scrollPane = new JScrollPane(table);

    // Create a JFrame to hold the scroll pane
    JFrame frame = new JFrame("Employee Details");
    frame.add(scrollPane);
    frame.pack();
    frame.setVisible(true);

} catch (SQLException e) {
    JOptionPane.showMessageDialog(AddEmployee.this, "Error: " + e.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
}
}

});

Search.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        try {

            String EID = textField_1.getText();

            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hotel_db", "root", "d1e2v3a4");
            Statement s = conn.createStatement();
            ResultSet rs = s.executeQuery("SELECT * FROM EMPLOYEE where EID =
""+EID+"");

            // Create a DefaultTableModel to hold your data
            DefaultTableModel model = new DefaultTableModel();
            model.addColumn("Name");
            model.addColumn("EID");
            model.addColumn("Designation");
            model.addColumn("DOB");

            // Populate the model with data from the ResultSet

```



```
        while (rs.next()) {
            Object[] row = {
                rs.getString(1),
                rs.getString(2),
                rs.getString(3),
                rs.getString(4)
            };
            model.addRow(row);
        }

        // Create a JTable with the model
        JTable table = new JTable(model);

        // Wrap the table in a scroll pane
        JScrollPane scrollPane = new JScrollPane(table);

        // Create a JFrame to hold the scroll pane
        JFrame frame = new JFrame("Employee Details");
        frame.add(scrollPane);
        frame.pack();
        frame.setVisible(true);

    } catch (SQLException e) {
        JOptionPane.showMessageDialog(AddEmployee.this, "Error: " + e.getMessage(),
        "Error", JOptionPane.ERROR_MESSAGE);
    }
}

});

setSize(900, 600);
setVisible(true);
setLocation(530, 200);

}
}
```

OUTPUT

 ADD MENU DETAILS

FoodName

FoodID

FoodPrice

ADD MENU DETAILS

SAVE **DELETE** **DISPLAY**

Food Menu Page:

```
class AddMenu extends JFrame{  
  
    Connection conn; // Declare connection at the class level  
  
    public void conn() {  
        try {  
            Class.forName("com.mysql.jdbc.Driver");  
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root",  
"d1e2v3a4");  
            Statement s = conn.createStatement();  
            System.out.println("connected");  
        } catch (Exception e) {  
            System.out.println(e);  
        }  
    }  
  
    JTextField textField,textField_1,textField_2,textField_3,textField_4,textField_5,textField_6;  
  
    public AddMenu(){  
  
        getContentPane().setForeground(Color.BLUE);  
        getContentPane().setBackground(Color.WHITE);  
        setTitle("ADD MENU DETAILS");  
  
        //setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);  
        setSize(778,486);  
        getContentPane().setLayout(null);  
  
        JLabel fdn = new JLabel("FoodName");  
        fdn.setFont(new Font("Tahoma", Font.PLAIN, 17));  
        fdn.setBounds(60, 30, 150, 27);  
        add(fdn);  
  
        textField = new JTextField();  
        textField.setBounds(200, 30, 150, 27);  
        add(textField);  
  
        JButton Next = new JButton("SAVE");
```

Menu Details

FoodID	FoodName	FoodPrice
C10	biryani	130
C2	noodles	120

ADD MENU DETAILS

FoodName

FoodID

FoodPrice

SAVE **DELETE** **DISPLAY**

ADD MENU DETAILS

```
Next.setBounds(100, 420, 150, 30);
Next.setBackground(Color.BLACK);
Next.setForeground(Color.WHITE);
add(Next);

JButton Delete = new JButton("DELETE");
Delete.setBounds( 350, 420, 150, 30);
Delete.setBackground(Color.BLACK);
Delete.setForeground(Color.WHITE);
add(Delete);

JButton Display = new JButton("DISPLAY");
Display.setBounds( 600, 420, 150, 30);
Display.setBackground(Color.BLACK);
Display.setForeground(Color.WHITE);
add(Display);

JLabel FID = new JLabel("FoodID");
FID.setFont(new Font("Tahoma", Font.PLAIN, 17));
FID.setBounds(60, 80, 150, 27);
add(FID);

textField_1 = new JTextField();
textField_1.setBounds(200, 80, 150, 27);
add(textField_1);

JLabel fpz = new JLabel("FoodPrice");
fpz.setFont(new Font("Tahoma", Font.PLAIN, 17));
fpz.setBounds(60, 140, 150, 27);
add(fpz);

textField_2 = new JTextField();
textField_2.setBounds(200, 140, 150, 27);
add(textField_2);

setVisible(true);
JLabel AddPassengers = new JLabel("ADD MENU DETAILS");
AddPassengers.setForeground(Color.BLUE);
AddPassengers.setFont(new Font("Tahoma", Font.PLAIN, 31));
AddPassengers.setBounds(450, 24, 442, 35);
add(AddPassengers);

Next.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae){
```

 ADD MENU DETAILS

FoodName	<input type="text" value="Meals"/>
FoodID	<input type="text" value="C1"/>
FoodPrice	<input type="text" value="100"/>

ADD MENU DETAILS

Message X

 Menu Added

OK

SAVE DELETE DISPLAY

 Menu Details

FoodID	FoodName	FoodPrice
C1	Meals	100
C10	biryani	130
C2	noodles	120

 ADD MENU DETAILS

FoodName	<input type="text" value="Meals"/>
FoodID	<input type="text" value="C1"/>
FoodPrice	<input type="text" value="100"/>

ADD MENU DETAILS

SAVE DELETE DISPLAY

```

String fdn = textField.getText();
String FID = textField_1.getText();
String fpz = textField_2.getText();

try {
    Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root", "d1e2v3a4");
    Statement s = conn.createStatement();
    String str = "INSERT INTO Menu values( "+FID+", "+fdn+", "+fpz+")";

    s.executeUpdate(str);
    JOptionPane.showMessageDialog(AddMenu.this,"Menu Added");
    //setVisible(false);
    //new AddMenu().setVisible(true);
} catch (Exception e) {
    e.printStackTrace();
}
}

Delete.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae){
        String fdn = textField.getText();
        String FID = textField_1.getText();
        String fdz = textField_2.getText();

        try {
            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root", "d1e2v3a4");
            Statement s = conn.createStatement();
            String str = "DELETE from Menu where FoodID='"+FID+"'";

            s.executeUpdate(str);
            JOptionPane.showMessageDialog(AddMenu.this,"Menu Value Deleted");
            //setVisible(false);
            //new AddMenu().setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

Display.addActionListener(new ActionListener() {

```

 ADD MENU DETAILS

FoodName

FoodID

FoodPrice

ADD MENU DETAILS

Message X

 Menu Value Deleted

OK

SAVE DELETE DISPLAY

 Menu Details

FoodID	FoodName	FoodPrice
C10	biryani	130
C2	noodles	120

 ADD MENU DETAILS

FoodName

FoodID

FoodPrice

ADD MENU DETAILS

SAVE DELETE DISPLAY

```
public void actionPerformed(ActionEvent ae) {
    try {
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/Hotel_DB", "root", "d1e2v3a4");
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM Menu");

        // Create a DefaultTableModel to hold your data
        DefaultTableModel model = new DefaultTableModel();
        model.addColumn("FoodID");
        model.addColumn("FoodName");
        model.addColumn("FoodPrice");

        // Populate the model with data from the ResultSet
        while (rs.next()) {
            Object[] row = {
                rs.getString(1),
                rs.getString(2),
                rs.getString(3)
            };
            model.addRow(row);
        }

        // Create a JTable with the model
        JTable table = new JTable(model);

        // Wrap the table in a scroll pane
        JScrollPane scrollPane = new JScrollPane(table);

        // Create a JFrame to hold the scroll pane
        JFrame frame = new JFrame("Menu Details");
        frame.add(scrollPane);
        frame.pack();
        frame.setVisible(true);

    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
}

setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
```



```
addWindowListener(new WindowAdapter() {  
    public void windowClosing(WindowEvent e){  
  
        new Dashboard().setVisible(true);;  
    }  
});  
setSize(900,600);  
setVisible(true);  
setLocation(530,200);  
  
}  
  
}
```

Result:

Successfully implemented the project.

