

Movie Ticket Booking System

Mini Project Report - IA-3

Object Oriented Programming - II [MCA 4221]

MCA-II Semester

Radhika

240970132

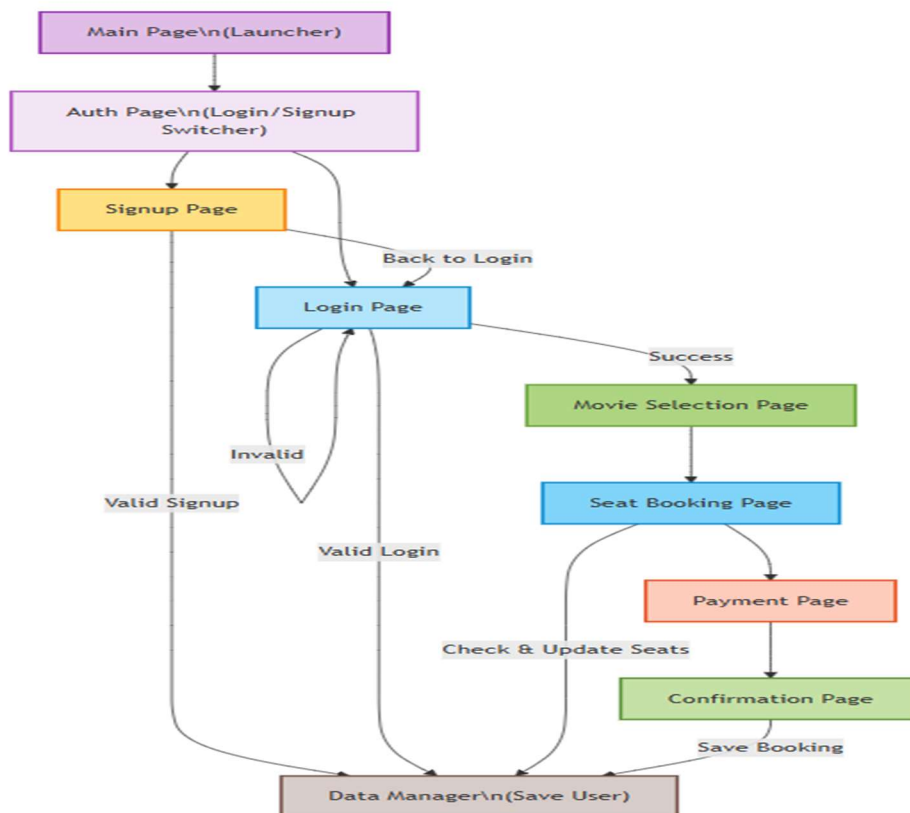
MCA – II Section -A

***Department of Data Science & Computer
Applications***

Overview

This project aims to design and implement a Movie Ticket Booking System using Java Swing. The system provides users with a user-friendly GUI interface to log in, browse movies, select showtimes and seats, make payments, and receive booking confirmations. It uses event-driven programming concepts and Swing components to provide a smooth and interactive experience.

Flow Diagram



Swing Components Used

- **JFrame** : Main window for each page like Login, Signup, Booking, and Payment.
- **JPanel** : Container for organizing UI components inside the frame.
- **JLabel** : Displays text labels such as titles and field names.
- **TextField / JPasswordField** : Used for user input like username and password.
- **JButton** : Performs actions when clicked, like Login, Signup, or Pay.
- **JOptionPane** : Shows popup messages for alerts and confirmations.
- **GridBagLayout / GridLayout** : Layout managers for arranging components.
- **TitledBorder** : Adds a titled border to group and label panel sections.

Event Handling

The application uses ActionListeners for handling button clicks:

- **Login Button** : Checks credentials and navigates to movie selection.
- **Signup Button**: Registers new users after password validation.
- **Seat Buttons**: Toggle between selected and unselected using color indicators.
- **Book Button**: Proceeds to payment only if at least one seat is selected.
- **Pay Now Button**: Confirms payment, saves booking data, and opens confirmation.

Program Code

Main.java

```
public class Main {  
    public static void main(String[] args) {  
        DataManager.loadData();  
        new LoginPage();  
    }  
}
```

LoginPage.java

```
import javax.swing.*.*;  
import java.awt.*.*;  
  
public class LoginPage extends JFrame {  
    public LoginPage() {  
        setTitle("Login");  
        setSize(400, 200);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        setLocationRelativeTo(null);  
  
        JLabel userLabel = new JLabel("Username:");  
        JTextField userField = new JTextField(15);  
        JLabel passLabel = new JLabel("Password:");  
        JPasswordField passField = new JPasswordField(15);  
        JButton loginBtn = new JButton("Login");  
        JButton signupBtn = new JButton("Signup");  
  
        setLayout(new GridBagLayout());  
    }  
}
```

```

GridBagConstraints gbc = new GridBagConstraints();
gbc.insets = new Insets(5, 5, 5, 5);

gbc.gridx = 0; gbc.gridy = 0; add(userLabel, gbc);
gbc.gridx = 1; add(userField, gbc);
gbc.gridx = 0; gbc.gridy = 1; add(passLabel, gbc);
gbc.gridx = 1; add(passField, gbc);
gbc.gridx = 0; gbc.gridy = 2; add(loginBtn, gbc);
gbc.gridx = 1; add(signupBtn, gbc);

setVisible(true);

loginBtn.addActionListener(e -> {
    String user = userField.getText();
    String pass = new String(passField.getPassword());
    if (DataManager.users.containsKey(user) &&
DataManager.users.get(user).equals(pass)) {
        DataManager.currentUser = user;
        dispose();
        new MovieSelectionPage();
    } else {
        JOptionPane.showMessageDialog(this, "Invalid login.");
    }
});

signupBtn.addActionListener(e -> {
    dispose();
    new SignupPage();
});
}

```

```
}
```

signupPage.java

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

public class SignupPage extends JFrame {
    public SignupPage() {
        setTitle("Signup");
        setSize(400, 300);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        gbc.fill = GridBagConstraints.HORIZONTAL;

        JLabel titleLabel = new JLabel("Create Account");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 18));
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.gridwidth = 2;
        panel.add(titleLabel, gbc);

        JLabel userLabel = new JLabel("Username:");
        JTextField userField = new JTextField(15);
        userField.setFont(new Font("Arial", Font.PLAIN, 14));
        gbc.gridy++;
```

```
gbc.gridwidth = 1;  
panel.add(userLabel, gbc);  
gbc.gridx = 1;  
panel.add(userField, gbc);
```

```
JLabel passLabel = new JLabel("Password:");  
JPasswordField passField = new JPasswordField(15);  
passField.setFont(new Font("Arial", Font.PLAIN, 14));  
gbc.gridx = 0;  
gbc.gridy++;  
panel.add(passLabel, gbc);  
gbc.gridx = 1;  
panel.add(passField, gbc);
```

```
JLabel confirmLabel = new JLabel("Confirm Password:");  
JPasswordField confirmField = new JPasswordField(15);  
confirmField.setFont(new Font("Arial", Font.PLAIN, 14));  
gbc.gridx = 0;  
gbc.gridy++;  
panel.add(confirmLabel, gbc);  
gbc.gridx = 1;  
panel.add(confirmField, gbc);
```

```
JButton signupBtn = new JButton("Signup");  
signupBtn.setFont(new Font("Arial", Font.BOLD, 14));  
gbc.gridx = 0;  
gbc.gridy++;  
gbc.gridwidth = 2;  
panel.add(signupBtn, gbc);
```

```
add(panel);
setVisible(true);

signupBtn.addActionListener(e -> {
    String username = userField.getText().trim();
    String password = new String(passField.getPassword());
    String confirmPassword = new
String(confirmField.getPassword());

    if (username.isEmpty() || password.isEmpty() ||
confirmPassword.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Please fill in all
fields.");
        return;
    }

    if (!password.equals(confirmPassword)) {
        JOptionPane.showMessageDialog(this, "Passwords do not
match.");
        return;
    }

    if (DataManager.users.containsKey(username)) {
        JOptionPane.showMessageDialog(this, "Username already
exists.");
        return;
    }

    DataManager.users.put(username, password);
    DataManager.saveData();
}
```



```

        JOptionPane.showMessageDialog(this, "Signup successful.
Please login.");
        dispose();
        new LoginPage();
    });
}
}

```

Auth.java

```

import java.io.*;
import java.util.HashMap;

public class Auth {
    private static final String FILE_NAME = "users.dat";
    protected static HashMap<String, User> userDB = loadUsers();

    @SuppressWarnings("unchecked")
    private static HashMap<String, User> loadUsers() {
        try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(FILE_NAME))) {
            return (HashMap<String, User>) in.readObject();
        } catch (Exception e) {
            return new HashMap<>();
        }
    }

    private static void saveUsers() {

```

```
try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(FILE_NAME))) {
    out.writeObject(userDB);
} catch (IOException e) {
    e.printStackTrace();
}
}
```

// Register a new user

```
public static boolean register(String username, String password) {
    if (userDB.containsKey(username)) return false;
    userDB.put(username, new User(username, password));
    saveUsers();
    return true;
}
```

// Login and return the User object

```
public static User login(String username, String password) {
    User user = userDB.get(username);
    if (user != null && user.getPassword().equals(password)) {
        return user;
    }
    return null;
}
```

// Save changes to a specific user (after booking)

```
public static void saveCurrentUser(User user) {
    userDB.put(user.getUsername(), user);
    saveUsers();
}
```

```
    public static HashMap<String, User> getAllUsers() {  
        return userDB;  
    }  
}
```

MovieSelectionPage.java

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
  
public class MovieSelectionPage extends JFrame {  
    public MovieSelectionPage() {  
        setTitle("Select Movie");  
        setSize(600, 500);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
        setLocationRelativeTo(null);  
  
        JPanel panel = new JPanel(new GridBagLayout());  
        GridBagConstraints gbc = new GridBagConstraints();  
        gbc.insets = new Insets(10, 10, 10, 10);  
        gbc.anchor = GridBagConstraints.WEST;  
  
        String[] movies = {  
            "Bhaahubali",  
            "KGF",  
            "Interstellar",  
            "The Dark Knight",  
            "Parasite"  
        }  
    }  
}
```

```
};
```

```
String[] times = {  
    "7:00 AM",  
    "1:00 PM",  
    "4:00 PM",  
    "7.00 PM",  
};
```

```
String[] theaters = {  
    "PVR Cinemas",  
    "Bharath Cinemas",  
    "INOX",  
    "Cinepolies",  
    "Cini Galaxy"  
};
```

```
JComboBox<String> movieBox = new JComboBox<>(movies);  
JComboBox<String> timeBox = new JComboBox<>(times);  
JComboBox<String> theaterBox = new JComboBox<>(theaters);  
JTextArea descriptionArea = new JTextArea(4, 30);  
descriptionArea.setLineWrap(true);  
descriptionArea.setWrapStyleWord(true);  
descriptionArea.setEditable(false);
```

```
JButton nextBtn = new JButton("Next");
```

```
gbc.gridx = 0;  
gbc.gridy = 0;  
panel.add(new JLabel("Select Movie:"), gbc);
```

```
gbc.gridx = 1;  
panel.add(movieBox, gbc);
```

```
gbc.gridx = 0;  
gbc.gridy++;  
panel.add(new JLabel("Select Time:"), gbc);  
gbc.gridx = 1;  
panel.add(timeBox, gbc);
```

```
gbc.gridx = 0;  
gbc.gridy++;  
panel.add(new JLabel("Select Theater:"), gbc);  
gbc.gridx = 1;  
panel.add(theaterBox, gbc);
```

```
gbc.gridx = 0;  
gbc.gridy++;  
panel.add(new JLabel("Movie Description:"), gbc);  
gbc.gridx = 1;  
panel.add(new JScrollPane(descriptionArea), gbc);
```

```
gbc.gridx = 1;  
gbc.gridy++;  
panel.add(nextBtn, gbc);
```

```
add(panel);  
setVisible(true);
```

```
movieBox.addActionListener(e -> {  
    String selectedMovie = (String) movieBox.getSelectedItem();
```

```
        switch (selectedMovie) {
            case "Bhaahubali":
                descriptionArea.setText("Superheroes unite to defeat
Thanos in this epic Marvel movie.");
                break;
            case "KGF":
                descriptionArea.setText("A mind-bending thriller where
dreams within dreams blur reality.");
                break;
            case "Interstellar":
                descriptionArea.setText("Astronauts explore a wormhole in
search of a new home for humanity.");
                break;
            case "The Dark Knight":
                descriptionArea.setText("Batman faces the Joker in one of
the best superhero films ever.");
                break;
            case "Parasite":
                descriptionArea.setText("A poor family infiltrates a wealthy
household in this dark comedy thriller.");
                break;
            default:
                descriptionArea.setText("");
        }
    });
```

```
nextBtn.addActionListener(e -> {
    String selectedMovie = (String) movieBox.getSelectedItemAt();
    String selectedTime = (String) timeBox.getSelectedItemAt();
    String selectedTheater = (String) theaterBox.getSelectedItemAt();
```

```

        if (selectedMovie != null && selectedTime != null &&
selectedTheater != null) {
            String fullMovieInfo = selectedMovie + " - " + selectedTime +
" @ " + selectedTheater;
            dispose();
            new SeatBookingPage(fullMovieInfo);
        } else {
            JOptionPane.showMessageDialog(this, "Please select a movie,
time, and theater.");
        }
    });
}
}

```

SeatBookingPage.java

```

import javax.swing.*;
import java.awt.*;
import java.util.*;

```

```

public class SeatBookingPage extends JFrame {
    private Set<String> selected = new HashSet<>();

    public SeatBookingPage(String movie) {
        setTitle("Select Seats - " + movie);
        setLayout(new BorderLayout());

        JPanel seatPanel = new JPanel(new GridLayout(5, 6, 5, 5));
        Set<String> allBooked = new HashSet<>();
        for (String key : DataManager.bookings.keySet()) {

```

```
    if (key.endsWith(movie)) {  
        allBooked.addAll(DataManager.bookings.get(key));  
    }  
}
```

```
for (int i = 1; i <= 30; i++) {  
    String seat = "S" + i;  
    JButton btn = new JButton(seat);  
    if (allBooked.contains(seat)) {  
        btn.setBackground(Color.RED);  
        btn.setEnabled(false);  
    } else {  
        btn.setBackground(Color.GREEN);  
        btn.addActionListener(e -> {  
            if (selected.contains(seat)) {  
                selected.remove(seat);  
                btn.setBackground(Color.GREEN);  
            } else {  
                selected.add(seat);  
                btn.setBackground(Color.YELLOW);  
            }  
        });  
    }  
    seatPanel.add(btn);  
}
```

```
JButton bookBtn = new JButton("Book");  
bookBtn.addActionListener(e -> {  
    if (selected.isEmpty()) {
```



```
        JOptionPane.showMessageDialog(this, "Select at least one
seat.");
        return;
    }
    dispose();
    new PaymentPage(movie, selected);
});

JPanel bottomPanel = new JPanel();
bottomPanel.add(bookBtn);

add(seatPanel, BorderLayout.CENTER);
add(bottomPanel, BorderLayout.SOUTH);

setSize(400, 350);
setLocationRelativeTo(null);
setVisible(true);
}
}
```

PaymentPage.java

```
import javax.swing.*;
import java.awt.*;
import java.util.Set;

public class PaymentPage extends JFrame {
    public PaymentPage(String movie, Set<String> seats) {
        setTitle("Payment");
        setSize(400, 300);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
```

```
setLocationRelativeTo(null);
```

```
JPanel panel = new JPanel(new GridBagLayout());  
panel.setBorder(BorderFactory.createTitledBorder(  
    BorderFactory.createLineBorder(Color.GRAY, 1),  
    "Proceed to Payment to Confirm Your Seat",  
    javax.swing.border.TitledBorder.CENTER,  
    javax.swing.border.TitledBorder.TOP,  
    new Font("Arial", Font.BOLD, 16),  
    Color.BLACK  
));
```

```
GridBagConstraints gbc = new GridBagConstraints();  
gbc.insets = new Insets(10, 10, 10, 10);  
gbc.anchor = GridBagConstraints.WEST;
```

```
String username = DataManager.currentUser;  
int totalAmount = seats.size() * 150;
```

```
JLabel userLabel = new JLabel("Username: " + username);  
userLabel.setFont(new Font("Arial", Font.BOLD, 16));
```

```
JLabel totalLabel = new JLabel("Amount to Pay: ₹" +  
totalAmount);  
totalLabel.setFont(new Font("Arial", Font.BOLD, 16));
```

```
JButton payBtn = new JButton("Pay Now");  
payBtn.setFont(new Font("Arial", Font.BOLD, 14));
```

```
gbc.gridx = 0;
```

```

gbc.gridy = 0;
gbc.gridwidth = 2;
panel.add(userLabel, gbc);

gbc.gridy++;
panel.add(totalLabel, gbc);

gbc.gridy++;
panel.add(payBtn, gbc);

add(panel);
setVisible(true);

payBtn.addActionListener(e -> {

    String key = DataManager.currentUser + "-" + movie;
    DataManager.bookings.put(key, seats);
    DataManager.saveData();

    JOptionPane.showMessageDialog(this, "Payment successful!
Booking confirmed.");
    dispose();
    new ConfirmationPage(movie, seats);
});
}
}

```

```

confirmationPage.java
// ConfirmationPage.java
import javax.swing.*;

```

```
import java.awt.*;
import java.util.Set;

public class ConfirmationPage extends JFrame {
    public ConfirmationPage(String movie, Set<String> seats) {
        setTitle("Booking Confirmed");
        setSize(500, 400);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel(new GridBagLayout());
        panel.setBorder(BorderFactory.createTitledBorder(
            BorderFactory.createLineBorder(Color.GRAY, 1),
            "Booking Confirmation",
            javax.swing.border.TitledBorder.CENTER,
            javax.swing.border.TitledBorder.TOP,
            new Font("Arial", Font.BOLD, 16),
            Color.BLACK
        ));

        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        gbc.anchor = GridBagConstraints.WEST;

        String username = DataManager.currentUser;
        int amount = seats.size() * 150;

        JLabel headingLabel = new JLabel("Booking Confirmed");
        headingLabel.setFont(new Font("Arial", Font.BOLD, 18));
```

```
JLabel userLabel = new JLabel("User: " + username);
JLabel movieLabel = new JLabel("Movie: " + movie);
JLabel seatsLabel = new JLabel("Seats: " + String.join(", ", seats));
JLabel amountLabel = new JLabel("Total Paid: ₹" + amount);
JLabel confirmLabel = new JLabel("Your booking is confirmed!");
```

```
// Set all fonts to bold
Font boldFont = new Font("Arial", Font.BOLD, 16);
userLabel.setFont(boldFont);
movieLabel.setFont(boldFont);
seatsLabel.setFont(boldFont);
amountLabel.setFont(boldFont);
confirmLabel.setFont(boldFont);
```

```
JButton bookAgainBtn = new JButton("Book Another Ticket");
JButton exitBtn = new JButton("Exit");
```

```
bookAgainBtn.setFont(new Font("Arial", Font.BOLD, 14));
exitBtn.setFont(new Font("Arial", Font.BOLD, 14));
```

```
gbc.gridx = 0;
gbc.gridy = 0;
gbc.gridwidth = 2;
panel.add(headingLabel, gbc);
```

```
gbc.gridy++;
panel.add(userLabel, gbc);
```

```
gbc.gridy++;
panel.add(movieLabel, gbc);
```

```
gbc.gridy++;
panel.add(seatsLabel, gbc);

gbc.gridy++;
panel.add(amountLabel, gbc);

gbc.gridy++;
panel.add(confirmLabel, gbc);

gbc.gridy++;
panel.add(bookAgainBtn, gbc);

gbc.gridy++;
panel.add(exitBtn, gbc);

add(panel);
setVisible(true);

bookAgainBtn.addActionListener(e -> {
    dispose();
    new MovieSelectionPage();
});

exitBtn.addActionListener(e -> System.exit(0));
}
}
```

DataManager.java

```
import java.io.*;
```

```

import java.util.*;

public class DataManager {
    public static Map<String, String> users = new HashMap<>();
    public static Map<String, Set<String>> bookings = new
HashMap<>();
    public static String currentUser = "";

    private static final String DATA_FILE = "data.dat";

    @SuppressWarnings("unchecked")
    public static void loadData() {
        File file = new File(DATA_FILE);
        if (!file.exists()) return;

        try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(file))) {
            users = (Map<String, String>) in.readObject();
            bookings = (Map<String, Set<String>>) in.readObject();
        } catch (Exception e) {
            System.out.println("Error loading data: " + e.getMessage());
            users = new HashMap<>();
            bookings = new HashMap<>();
        }
    }

    public static void saveData() {
        try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(DATA_FILE))) {
            out.writeObject(users);

```

```
        out.writeObject(bookings);
    } catch (IOException e) {
        System.out.println("Error saving data: " + e.getMessage());
    }
}
}
```

User.java

```
import java.io.Serializable;
```

```
public class User implements Serializable {
    private String username;
    private String password;

    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }

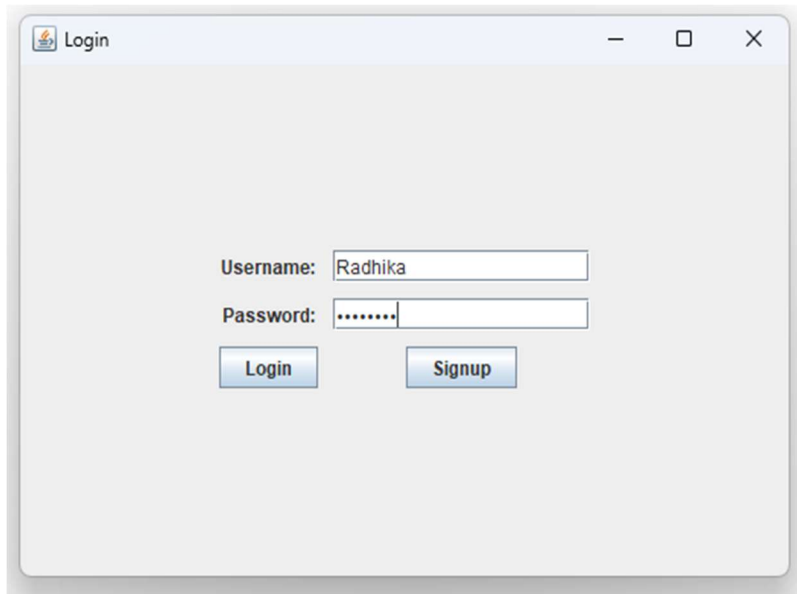
    public String getUsername() {
        return username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```


Screenshots(Output)

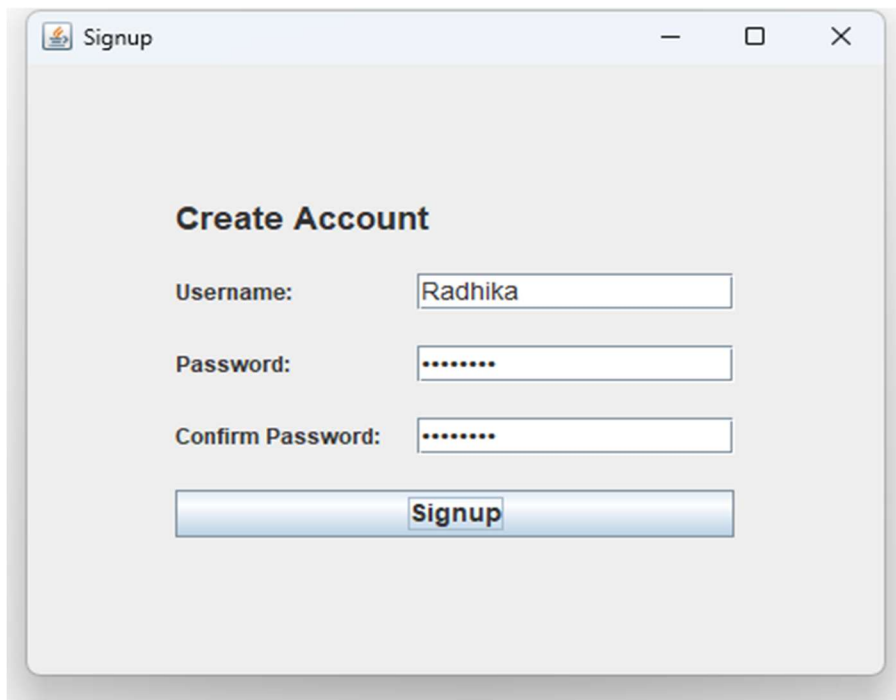
- **Login Page**



A screenshot of a web application window titled "Login". The window has a light gray background and a standard window title bar with minimize, maximize, and close buttons. The login form is centered and contains the following elements:

- A label "Username:" followed by a text input field containing the text "Radhika".
- A label "Password:" followed by a password input field with seven dots.
- Two buttons: "Login" and "Signup", both with a blue gradient and white text.

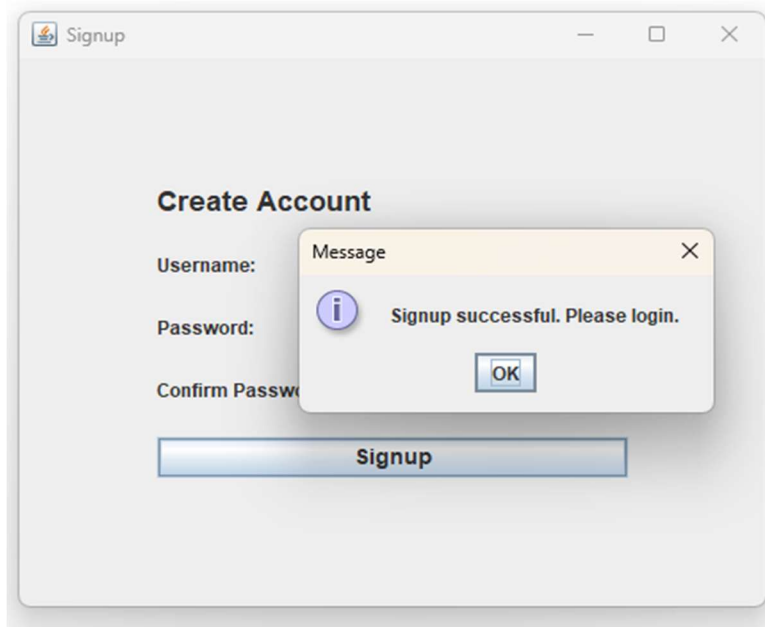
- **Signup Page**



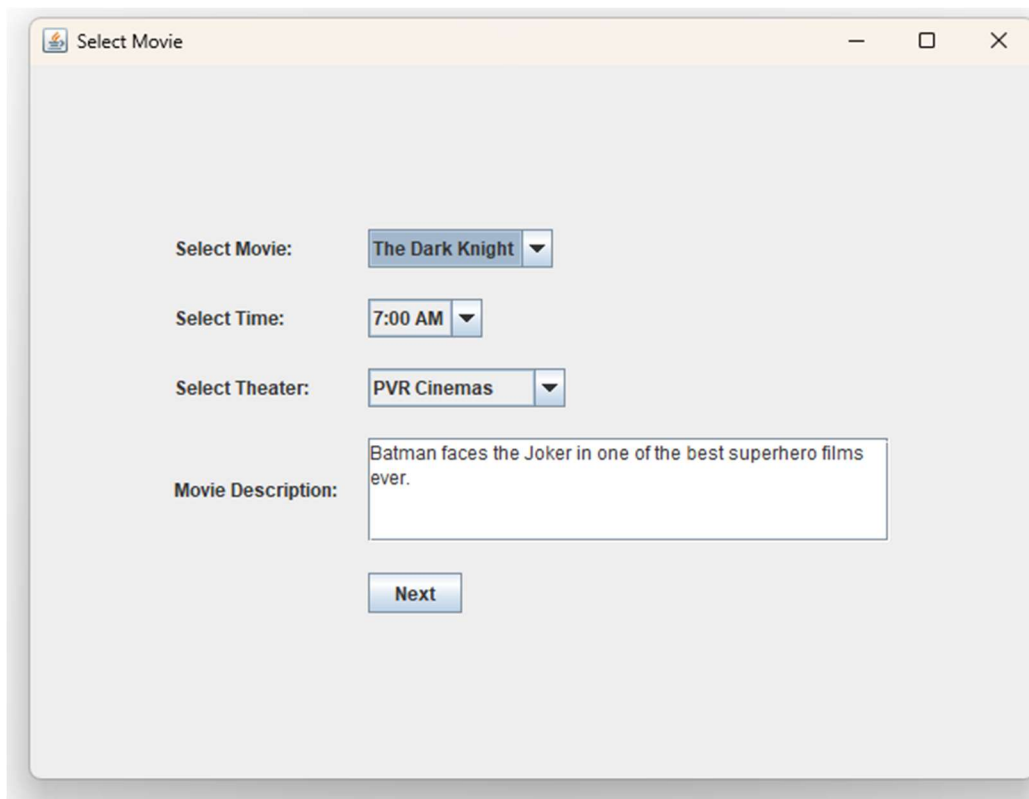
A screenshot of a web application window titled "Signup". The window has a light gray background and a standard window title bar with minimize, maximize, and close buttons. The signup form is titled "Create Account" and contains the following elements:

- A label "Username:" followed by a text input field containing the text "Radhika".
- A label "Password:" followed by a password input field with seven dots.
- A label "Confirm Password:" followed by a password input field with seven dots.
- A single "Signup" button with a blue gradient and white text.

- **Signup Successful**



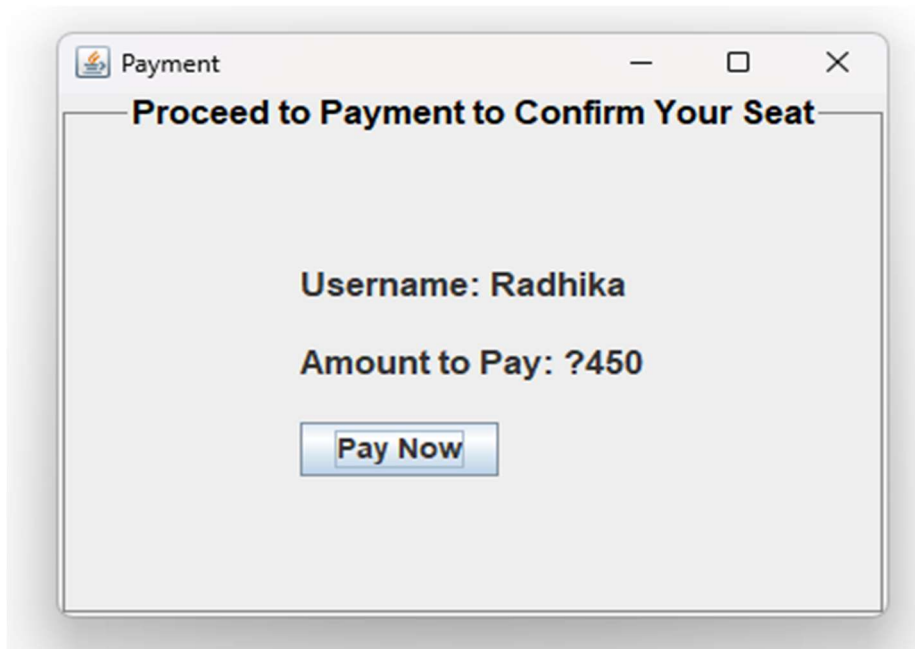
- **Ticket Booking Page**



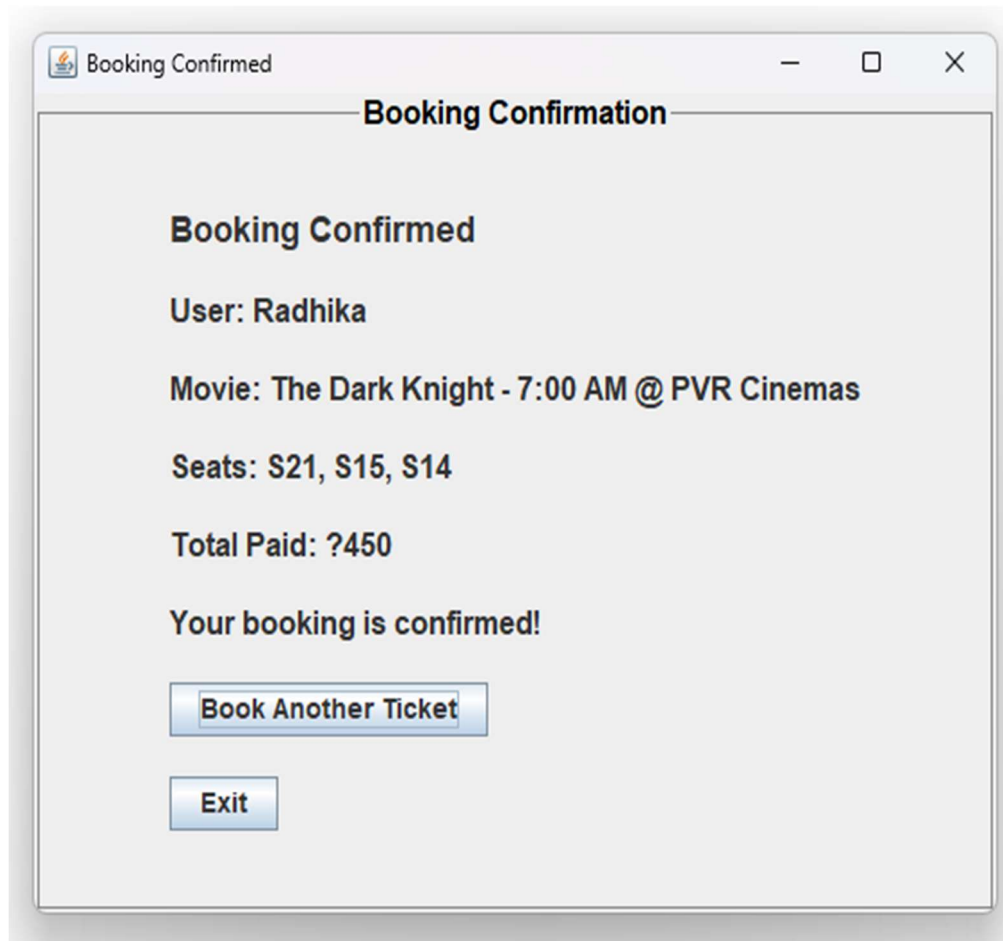
- **SeatBooking Page**



- **Payment Page**



- **Booking Confirmed**



References

- Coursera GUI App Project: <https://www.coursera.org/projects/build-java-gui-apps>
- Oracle Java Docs : <https://docs.oracle.com/javase/8/docs/api/>
- Swing Tutorials: <https://docs.oracle.com/javase/tutorial/uiswing/>
- Java File I/O and Serialization :<https://www.baeldung.com/java-io>

- **Coursera Certificate**

