

# **Movie Ticket Booking System**

## **Mini Project Report - IA-3**

**Object Oriented Programming II [MCA 4221]**

**Prajwal V Naik**

**240970015**

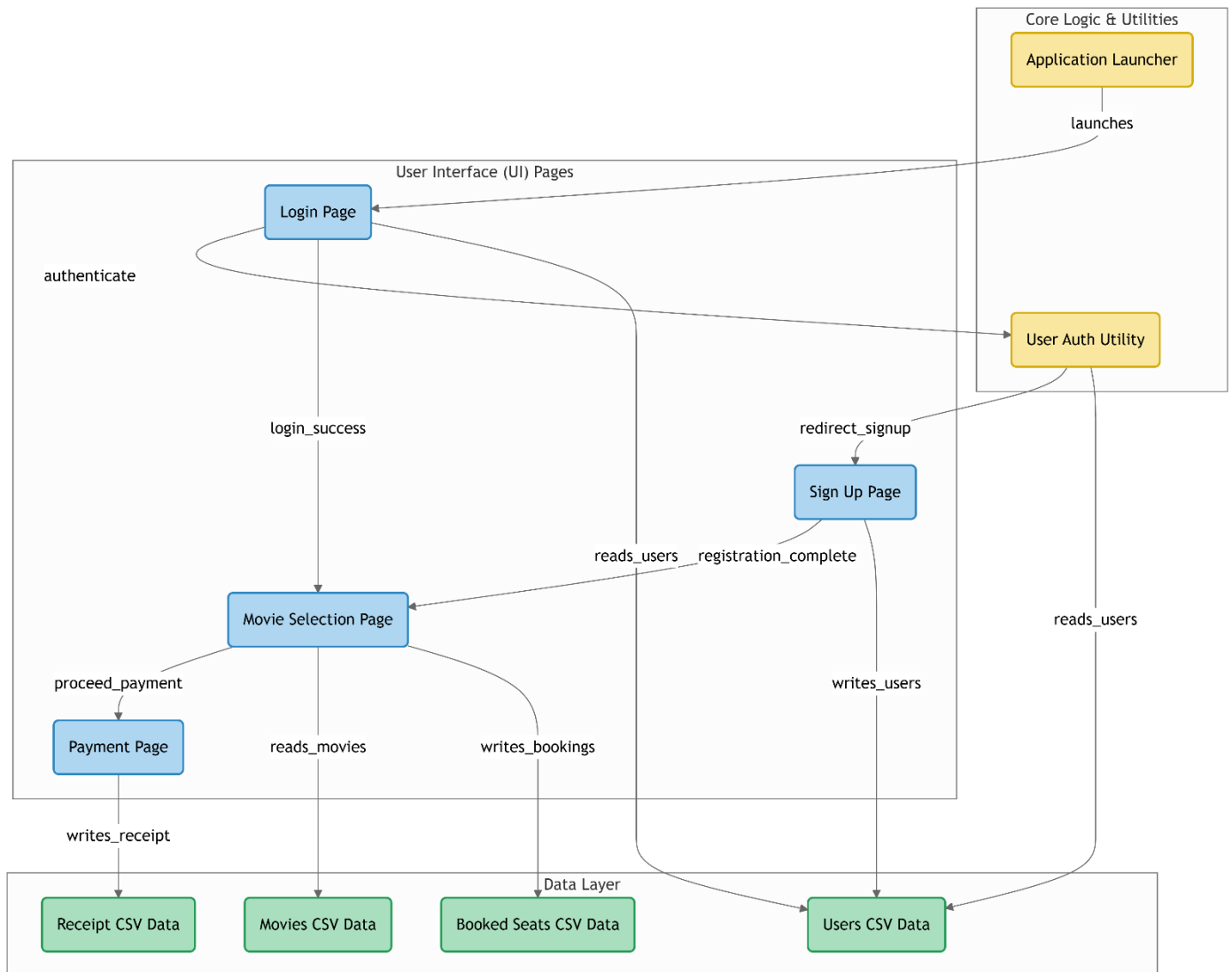
**MCA-II**

**Department of Data Science & Computer  
Applications**

# Overview

The goal of this project is to design and implement a Movie Ticket Booking System using Java Swing for GUI and CSV files for data storage. The application allows users to log in or register, select a movie from a list, choose the theater and showtime, view available seats, and book them. The booking is followed by a payment confirmation step and a final ticket confirmation. Bookings are saved to CSV files to ensure persistence.

## Flow Diagram



## Swing Components

- **JFrame:** Main window container for all pages.
- **JPanel:** Organizes layout into sections.
- **JLabel:** Displays text like titles and field labels.
- **TextField / JPasswordField:** For user input in login/signup.
- **JButton:** Triggers actions (login, signup, book seats, pay).
- **JComboBox:** Dropdowns for selecting movies and theaters.
- **JTextArea:** For displaying movie descriptions.
- **JOptionPane:** For pop-up messages, confirmations, and alerts.
- **JScrollPane:** Scrollable container for text area.

## Event Handling

The system is event-driven using ActionListener interfaces. Each button or interactive component triggers an action:

- **Login Button:** Validates credentials from users.csv
- **Signup Button:** Appends new user to users.csv
- **Movie ComboBox:** Filters and displays related theaters and descriptions.
- **Seat Buttons:** Toggles selection of seats; disables booked ones.
- **Book Button:** Confirms seat selection, stores in booked\_seats.csv
- **Pay Button:** Saves the booking record to receipt.csv and confirms ticket.

## Program Code

LoginPage.java

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

public class LoginPage extends JFrame {

    private JTextField usernameField;
    private JPasswordField passwordField;

    public LoginPage() {
        setTitle("Book My Ticket - Login");
        setSize(400, 250);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);

        JPanel panel = new JPanel();
        panel.setLayout(null);

        JLabel titleLabel = new JLabel("LogIn");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
        titleLabel.setBounds(150, 20, 100, 30);
        panel.add(titleLabel);

        JLabel userLabel = new JLabel("Username:");
        userLabel.setBounds(50, 70, 100, 25);
```

```
panel.add(userLabel);
```

```
usernameField = new JTextField();  
usernameField.setBounds(150, 70, 180, 25);  
panel.add(usernameField);
```

```
JLabel passLabel = new JLabel("Password:");  
passLabel.setBounds(50, 110, 100, 25);  
panel.add(passLabel);
```

```
passwordField = new JPasswordField();  
passwordField.setBounds(150, 110, 180, 25);  
panel.add(passwordField);
```

```
JButton loginButton = new JButton("LogIn");  
loginButton.setBounds(150, 150, 90, 30);  
panel.add(loginButton);
```

```
JButton signUpButton = new JButton("SignUp");  
signUpButton.setBounds(250, 150, 90, 30);  
panel.add(signUpButton);
```

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

```
loginButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        String username = usernameField.getText().trim();  
        String password = new String(passwordField.getPassword()).trim();
```

```

        if (AuthUtil.validateLogin(username, password)) {
            JOptionPane.showMessageDialog(null, "Logged in Successfully");
            dispose();
            new MovieSelectionPage(username);
        } else {
            JOptionPane.showMessageDialog(null, "Check your username or
password is correct?");
        }
    }
});
signUpButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
        new SignUpPage();
    }
});
}

public static void main(String[] args) {
    new LoginPage();
}
}

```

### SignUpPage.java

```

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

public class SignUpPage extends JFrame {

```

```
private JTextField usernameField;
private JPasswordField passwordField;
private JPasswordField confirmPasswordField;

public SignUpPage() {
    setTitle("Book My Ticket - Sign Up");
    setSize(400, 300);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    setResizable(false);

    JPanel panel = new JPanel();
    panel.setLayout(null);

    JLabel titleLabel = new JLabel("SignUp");
    titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
    titleLabel.setBounds(140, 20, 150, 30);
    panel.add(titleLabel);

    JLabel userLabel = new JLabel("Username:");
    userLabel.setBounds(50, 70, 100, 25);
    panel.add(userLabel);

    usernameField = new JTextField();
    usernameField.setBounds(160, 70, 180, 25);
    panel.add(usernameField);

    JLabel passLabel = new JLabel("Password:");
    passLabel.setBounds(50, 110, 100, 25);
```

```
panel.add(passLabel);
```

```
passwordField = new JPasswordField();  
passwordField.setBounds(160, 110, 180, 25);  
panel.add(passwordField);
```

```
JLabel confirmLabel = new JLabel("Confirm Password:");  
confirmLabel.setBounds(50, 150, 150, 25);  
panel.add(confirmLabel);
```

```
confirmPasswordField = new JPasswordField();  
confirmPasswordField.setBounds(160, 150, 180, 25);  
panel.add(confirmPasswordField);
```

```
JButton signUpButton = new JButton("SignIn");  
signUpButton.setBounds(150, 190, 100, 30);  
panel.add(signUpButton);
```

```
JButton backButton = new JButton("LogIn");  
backButton.setBounds(260, 190, 80, 30);  
panel.add(backButton);
```

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

```
signUpButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        String username = usernameField.getText().trim();  
        String password = new String(passwordField.getPassword()).trim();
```



```

String confirmPassword = new
String(confirmPasswordField.getPassword()).trim();

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

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

    boolean success = AuthUtil.signUp(username, password);
    if (success) {
        JOptionPane.showMessageDialog(null, "Registration successful!
Please login again");
        dispose();
        new LoginPage();
    } else {
        JOptionPane.showMessageDialog(null, "User already exists or
Something went wrong");
    }
}

});

backButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

```
        dispose();
        new LoginPage();
    }
});
}
```

```
public static void main(String[] args) {
    new SignUpPage();
}
}
```

AuthUtil.java

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

```
public class AuthUtil {
```

```
    private static final String FILE_NAME = "users.csv";
```

```
    public static boolean validateLogin(String username, String password) {
        try (BufferedReader br = new BufferedReader(new
        FileReader(FILE_NAME))) {
```

```
            String line;
```

```
            while ((line = br.readLine()) != null) {
```

```
                String[] values = line.split(",");
```

```
                if (values.length >= 2) {
```

```
                    if (values[0].trim().equals(username) &&
values[1].trim().equals(password)) {
```

```
                        return true;
```

```
                    }
```

```
                }
```

```
            }
```

```

    } catch (IOException e) {
        System.out.println("Error reading users.csv: " + e.getMessage());
    }
    return false;
}

public static boolean signUp(String username, String password) {
    if (validateLogin(username, password)) {
        return false;
    }

    try (FileWriter fw = new FileWriter(FILE_NAME, true);
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter out = new PrintWriter(bw)) {

        out.println(username + "," + password);
        return true;

    } catch (IOException e) {
        System.out.println("Error writing to users.csv: " + e.getMessage());
    }
    return false;
}
}

```

### MovieSelectionPage.java

```

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;
import java.io.*.*;

```

```
import java.util.List;
import java.util.ArrayList;
import java.util.Set;
import java.util.HashSet;
import java.util.Map;
import java.util.HashMap;
import java.util.Arrays;
```

```
public class MovieSelectionPage extends JFrame {
```

```
    private String username;
```

```
    private JComboBox<String> movieDropdown;
```

```
    private JComboBox<String> theaterDropdown;
```

```
    private JTextArea movieDescriptionArea;
```

```
    private JPanel seatPanel;
```

```
    private JButton[][] seats;
```

```
    private final int rows = 5, cols = 5;
```

```
    private List<Movie> movieList = new ArrayList<>();
```

```
    private Map<String, Set<String>> bookedSeatsMap = new HashMap<>();
```

```
    public MovieSelectionPage(String username) {
```

```
        this.username = username;
```

```
        setTitle("Book My Ticket - Select Your Happiness");
```

```
        setSize(700, 600);
```

```
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        setLocationRelativeTo(null);
```

```
setLayout(new BorderLayout());
```

```
loadMoviesFromCSV("movies.csv");
```

```
loadBookedSeats();
```

```
JPanel topPanel = new JPanel(new GridLayout(3, 2, 10, 10));
```

```
topPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
```

```
movieDropdown = new  
JComboBox<>(getUniqueMovieTitles().toArray(new String[0]));
```

```
theaterDropdown = new JComboBox<>();
```

```
movieDescriptionArea = new JTextArea(3, 30);
```

```
movieDescriptionArea.setEditable(false);
```

```
movieDescriptionArea.setLineWrap(true);
```

```
movieDescriptionArea.setWrapStyleWord(true);
```

```
topPanel.add(new JLabel("Select Movie:"));
```

```
topPanel.add(movieDropdown);
```

```
topPanel.add(new JLabel("Select Theater:"));
```

```
topPanel.add(theaterDropdown);
```

```
topPanel.add(new JLabel("Description:"));
```

```
topPanel.add(new JScrollPane(movieDescriptionArea));
```

```
add(topPanel, BorderLayout.NORTH);
```

```
seatPanel = new JPanel(new GridLayout(rows, cols, 5, 5));
```

```
seats = new JButton[rows][cols];
```

```
add(seatPanel, BorderLayout.CENTER);
```

```
JButton bookButton = new JButton("Book Seats");  
bookButton.setPreferredSize(new Dimension(150, 40));  
add(bookButton, BorderLayout.SOUTH);
```

```
movieDropdown.addActionListener(e -> updateTheaterDropdown());  
theaterDropdown.addActionListener(e -> updateSeatLayout());
```

```
bookButton.addActionListener(e -> handleBooking());
```

```
movieDropdown.setSelectedIndex(0);  
updateTheaterDropdown();
```

```
setVisible(true);
```

```
}
```

```
private void loadMoviesFromCSV(String filename) {  
    try (BufferedReader br = new BufferedReader(new FileReader(filename))) {  
        String line;  
        br.readLine();  
        while ((line = br.readLine()) != null) {  
            String[] parts = line.split(",", 4);  
            if (parts.length == 4) {  
                movieList.add(new Movie(parts[0].trim(), parts[1].trim(),  
parts[2].trim(), parts[3].trim()));  
            }  
        }  
    } catch (IOException e) {  
        JOptionPane.showMessageDialog(this, "Error loading movie data!!!");  
    }  
}
```

```
}
```

```
private void loadBookedSeats() {  
    File file = new File("booked_seats.csv");  
    if (!file.exists()) return;  
  
    try (BufferedReader br = new BufferedReader(new FileReader(file))) {  
        String line;  
        while ((line = br.readLine()) != null) {  
            String[] parts = line.split(",", 2);  
            if (parts.length == 2) {  
                String key = parts[0];  
                String[] seats = parts[1].split(" ");  
                bookedSeatsMap.putIfAbsent(key, new HashSet<>());  
                bookedSeatsMap.get(key).addAll(Arrays.asList(seats));  
            }  
        }  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

```
private Set<String> getUniqueMovieTitles() {  
    Set<String> titles = new HashSet<>();  
    for (Movie m : movieList) titles.add(m.title);  
    return titles;  
}
```

```
private void updateTheaterDropdown() {
```

```
theaterDropdown.removeAllItems();
String selectedMovie = (String) movieDropdown.getSelectedItem();
for (Movie m : movieList) {
    if (m.title.equals(selectedMovie)) {
        theaterDropdown.addItem(m.theater + " (" + m.showtime + ")");
    }
}
```

```
for (Movie m : movieList) {
    if (m.title.equals(selectedMovie)) {
        movieDescriptionArea.setText(m.description);
        break;
    }
}
```

```
if (theaterDropdown.getItemCount() > 0)
    theaterDropdown.setSelectedIndex(0);
}
```

```
private void updateSeatLayout() {
    seatPanel.removeAll();
```

```
String movie = (String) movieDropdown.getSelectedItem();
String theaterInfo = (String) theaterDropdown.getSelectedItem();
if (movie == null || theaterInfo == null) return;
```

```
String theater = theaterInfo.split("\\(")[0];
String showtime = theaterInfo.split("\\(")[1].replace(")", "");
String bookingKey = movie + "_" + theater + "_" + showtime;
```



```
Set<String> booked = bookedSeatsMap.getOrDefault(bookingKey, new  
HashSet<>());
```

```
for (int i = 0; i < rows; i++) {  
    for (int j = 0; j < cols; j++) {  
        String seatCode = (i + 1) + "" + (char) ('A' + j);  
        JButton seat = new JButton(seatCode);  
        if (booked.contains(seatCode)) {  
            seat.setBackground(Color.RED);  
            seat.setEnabled(false);  
        } else {  
            seat.setBackground(Color.GREEN);  
            seat.addActionListener(new SeatClickHandler(seat));  
        }  
        seats[i][j] = seat;  
        seatPanel.add(seat);  
    }  
}
```

```
seatPanel.revalidate();  
seatPanel.repaint();  
}
```

```
private void handleBooking() {  
    String movie = (String) movieDropdown.getSelectedItem();  
    String theaterInfo = (String) theaterDropdown.getSelectedItem();  
    if (movie == null || theaterInfo == null) return;
```

```
String theater = theaterInfo.split("\\(")[0];  
String showtime = theaterInfo.split("\\(")[1].replace(")", "");  
String bookingKey = movie + "_" + theater + "_" + showtime;
```

```
StringBuilder selectedSeats = new StringBuilder();  
List<String> newBookedSeats = new ArrayList<>();
```

```
for (int i = 0; i < rows; i++) {  
    for (int j = 0; j < cols; j++) {  
        JButton btn = seats[i][j];  
        if (btn.getBackground() == Color.YELLOW) {  
            String seatCode = btn.getText();  
            selectedSeats.append(seatCode).append(" ");  
            newBookedSeats.add(seatCode);  
        }  
    }  
}
```

```
if (newBookedSeats.isEmpty()) {  
    JOptionPane.showMessageDialog(this, "No seats selected!");  
    return;  
}
```

```
int choice = JOptionPane.showConfirmDialog(this,  
    "Selected Seats: " + selectedSeats.toString() + "\nProceed to  
payment?",  
    "Confirm Booking",  
    JOptionPane.OK_CANCEL_OPTION,  
    JOptionPane.INFORMATION_MESSAGE);
```

```

if(choice == JOptionPane.OK_OPTION) {
    try (FileWriter fw = new FileWriter("booked_seats.csv", true)) {
        fw.write(bookingKey + "," + String.join(" ", newBookedSeats) + "\n");
    } catch (IOException e) {
        e.printStackTrace();
    }

    bookedSeatsMap.putIfAbsent(bookingKey, new HashSet<>());
    bookedSeatsMap.get(bookingKey).addAll(newBookedSeats);

    dispose();
    new PaymentPage(username, movie, theater, showtime,
newBookedSeats);
}
}

class SeatClickHandler implements ActionListener {
    private JButton seat;

    public SeatClickHandler(JButton seat) {
        this.seat = seat;
    }

    public void actionPerformed(ActionEvent e) {
        if (seat.getBackground() == Color.GREEN) {
            seat.setBackground(Color.YELLOW);
        } else if (seat.getBackground() == Color.YELLOW) {
            seat.setBackground(Color.GREEN);
        }
    }
}

```

```
    }  
  }  
}
```

```
class Movie {  
    String title, description, theater, showtime;  
  
    Movie(String title, String description, String theater, String showtime) {  
        this.title = title;  
        this.description = description;  
        this.theater = theater;  
        this.showtime = showtime;  
    }  
}  
  
public static void main(String[] args) {  
    new MovieSelectionPage("testuser");  
}  
}
```

### *PaymentPage.java*

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.time.LocalDateTime;  
import java.util.List;
```

```
public class PaymentPage extends JFrame {

    private String username, movie, theater, showtime;
    private List<String> selectedSeats;

    public PaymentPage(String username, String movie, String theater, String
showtime, List<String> selectedSeats) {
        this.username = username;
        this.movie = movie;
        this.theater = theater;
        this.showtime = showtime;
        this.selectedSeats = selectedSeats;

        setTitle("Book My Ticket - Pay for Happiness");
        setSize(400, 350);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);
        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
        panel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
        JLabel label = new JLabel("Proceed with payment to confirm your
booking:");
        label.setAlignmentX(Component.CENTER_ALIGNMENT);

        JLabel userLabel = new JLabel("User: " + username);
        userLabel.setAlignmentX(Component.CENTER_ALIGNMENT);

        JLabel amountLabel = new JLabel("Total Amount: ₹" + (selectedSeats.size()
* 100));
```

```

amountLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
JButton payButton = new JButton("Pay Now");
payButton.setAlignmentX(Component.CENTER_ALIGNMENT);
panel.add(label);
panel.add(Box.createRigidArea(new Dimension(0, 10)));
panel.add(userLabel);
panel.add(Box.createRigidArea(new Dimension(0, 10)));
panel.add(amountLabel);
panel.add(Box.createRigidArea(new Dimension(0, 20)));
panel.add(payButton);
add(panel);
setVisible(true);
payButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        saveReceipt();
        showConfirmation();
    }
});
}

private void saveReceipt() {
    try (FileWriter fw = new FileWriter("receipt.csv", true)) {
        String record = username + "," + movie + "," + theater + "," + showtime +
        "," +
        String.join(" ", selectedSeats) + "," + (selectedSeats.size() * 150) +
        "," +
        LocalDateTime.now();
        fw.write(record + "\n");
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

```

}

private void showConfirmation() {
    String seatList = String.join(", ", selectedSeats);
    String message = "Booking Confirmed!\n\n" +
        "User: " + username + "\n" +
        "Movie: " + movie + "\n" +
        "Theater: " + theater + "\n" +
        "Showtime: " + showtime + "\n" +
        "Seats: " + seatList + "\n" +
        "Amount Paid: ₹" + (selectedSeats.size() * 150) + "\n\n" +
        "Enjoy your movie!";

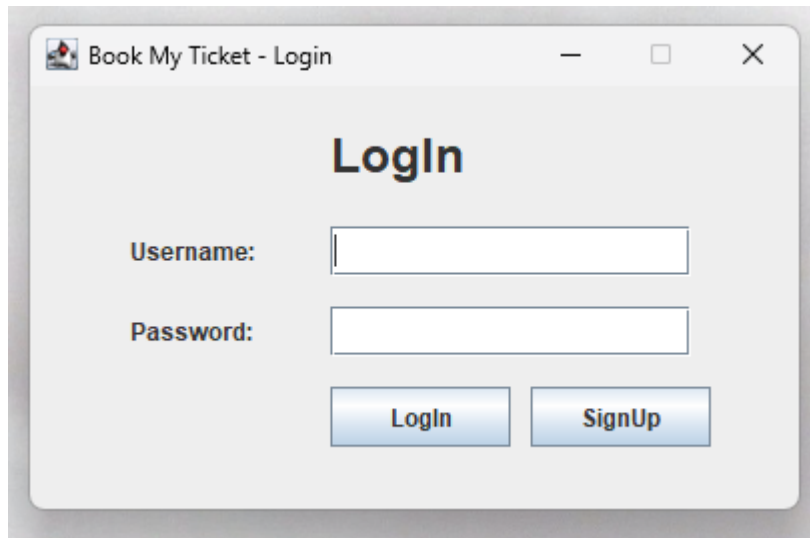
    int option = JOptionPane.showOptionDialog(null, message, "Ticket
Confirmation",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.INFORMATION_MESSAGE, null,
        new String[]{"Book Another Ticket", "Exit"}, "Book Another Ticket");
    if(option == 0) {
        dispose();
        new MovieSelectionPage(username);
    } else {
        JOptionPane.showMessageDialog(null, "Thank you for using Movie
Booking App!!!");
        System.exit(0);
    }
}

public static void main(String[] args) {
    new PaymentPage("testuser", "Dune", "PVR", "6 PM", List.of("3A",
"3B"));
}
}

```

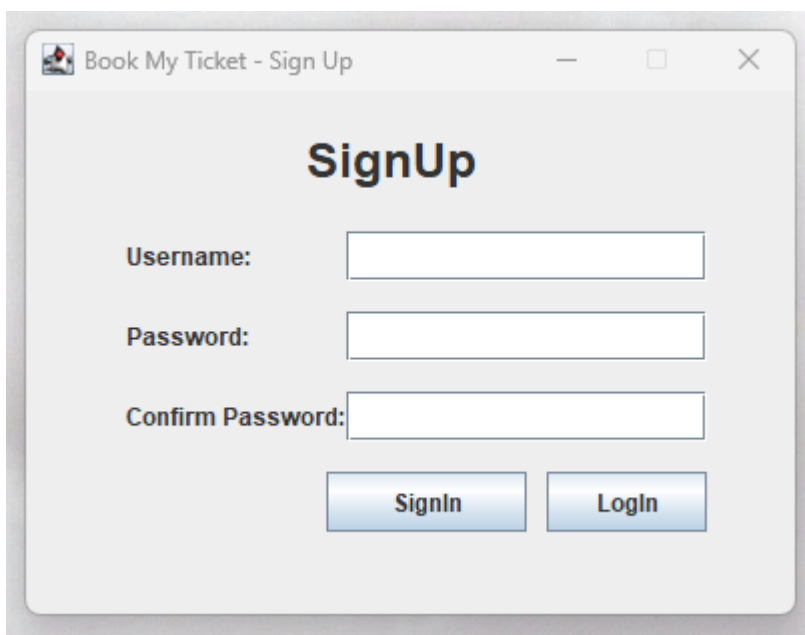
## Screenshots(Output)

- LogIn page



A screenshot of a web application window titled "Book My Ticket - Login". The window has a light gray background and a standard Windows-style title bar with minimize, maximize, and close buttons. The main heading "Login" is centered at the top in a bold, black font. Below the heading, there are two input fields: "Username:" and "Password:". Each label is followed by a white rectangular input box with a thin blue border. At the bottom of the form, there are two blue buttons with white text: "Login" and "SignUp".

- SignUp Page



A screenshot of a web application window titled "Book My Ticket - Sign Up". The window has a light gray background and a standard Windows-style title bar with minimize, maximize, and close buttons. The main heading "SignUp" is centered at the top in a bold, black font. Below the heading, there are three input fields: "Username:", "Password:", and "Confirm Password:". Each label is followed by a white rectangular input box with a thin blue border. At the bottom of the form, there are two blue buttons with white text: "SignIn" and "Login".



- Ticket Booking Page

Book My Ticket - Select Your Happiness

Select Movie: Interstellar

Select Theater: PVR (4 PM)

Description: A space-time journey through wormholes

1A	1B	1C	1D	1E
2A	2B	2C	2D	2E
3A	3B	3C	3D	3E
4A	4B	4C	4D	4E
5A	5B	5C	5D	5E

Book Seats

- Payment Page

Book My Ticket - Pay for Happiness

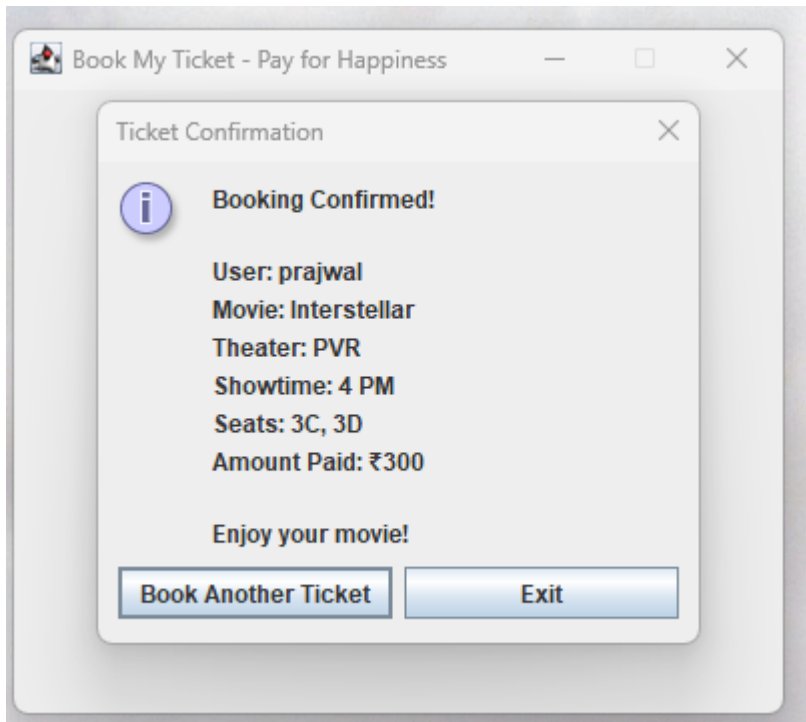
Proceed with payment to confirm your booking:

User: prajwal

Total Amount: ₹200

Pay Now

- Booking confirmed



## References

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