MOVIE TICKET BOOKING SYSTEM

A PROJECT REPORT

Submitted by

SUBASH R (2303811720521052)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

INFORMATION TECHNOLOGY

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOVEMBER- 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

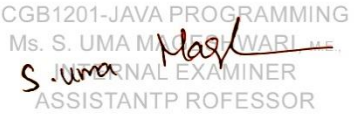
SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"MOVIE TICKET BOOKING SYSTEM"** is the bonafide work of **SUBASH R (2023811720521052)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE	**SIGNATURE**
Dr. C. Shyamala, M.E.,Ph.D.,	Ms. S. Uma Mageshwari, M.E.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	ASSISTANT PROFESSOR
Department of IT	Department of CSE
K.Ramakrishnan College of Technology (Autonomous)	K.Ramakrishnan College of Technology (Autonomous)
Samayapuram–621112.	Samayapuram–621112.

Submitted for the viva-voce examination held on 04.12.2024



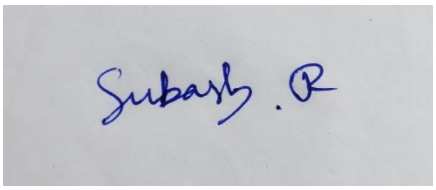
INTERNAL EXAMINER	EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **"MOVIE TICKET BOOKING SYSTEM"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING.**

.

Signature



SUBASH R

Place: Samayapuram

Date:04.12.2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.c

I whole heartily thanks to **Dr. C. SHYAMALA, M.E.,Ph.D.,** Head of the department, **INFORMATION TECHNOLOGY** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Ms. S. UMA MAGESHWARI, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

The Movie Ticket Booking System is an intuitive application designed to streamline the process of booking movie tickets. This project leverages Java Swing and AWT to provide a user-friendly graphical interface, enabling users to log in securely and book tickets for their favorite movies. The system includes a login module that ensures only authorized users can access the application. Once logged in, users can view a list of available movies, select a desired movie, specify the number of tickets, and confirm their booking. The application provides instant feedback and confirmation, making the ticket-booking experience seamless and efficient.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Movie Ticket Booking System is an intuitive application designed to streamline the process of booking movie tickets. This project leverages Java Swing and AWT to provide a user-friendly graphical interface, enabling users to log in securely and book tickets for their favorite movies. The system includes a login module that ensures only authorized users can access the application. Once logged in, users can view a list of available movies, select a desired movie, specify the number of tickets, and confirm their booking. The application provides instant feedback and confirmation, making the ticket-booking experience seamless and efficient.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1Objective

The primary objective of the Movie Ticket Booking System is to provide a user-friendly, efficient, and secure platform for booking movie tickets. This system aims to streamline the ticket reservation process by allowing users to log in, select movies, choose ticket quantities, and confirm bookings seamlessly. It also demonstrates the practical use of Java programming concepts, including GUI development with Swing and AWT, to create a functional and intuitive application.

1.2Overview

The Movie Ticket Booking System is a desktop application designed to simplify the process of reserving movie tickets. Developed using Java with Swing and AWT, it features a secure login system to ensure authorized access. After logging in, users can view a list of available movies, select a desired movie, specify the number of tickets, and confirm their booking. The application provides an intuitive graphical interface and real-time feedback to enhance the user experience. This system demonstrates the integration of object-oriented programming principles with graphical user interface design to create an efficient and user-friendly solution for ticket reservations.

1.3 Java Programming Concepts

The Movie Ticket Booking System utilizes several core Java programming concepts to function efficiently. It employs classes and objects to represent distinct components of the application, such as LoginFrame and BookingFrame. Encapsulation is used to protect user data, such as login credentials and ticket booking details, ensuring controlled access. The system follows event-driven programming by responding to user actions, like button clicks, using event listeners (ActionListener).

It leverages Swing and AWT for the graphical user interface, using components such as buttons, labels, and combo boxes to facilitate user interaction. Conditional statements help validate user input, such as checking login credentials and ensuring the correct number of tickets is entered. While exception handling isn't explicitly shown, it can be used to manage runtime errors, such as invalid input. Overall, the system integrates these Java concepts to create a functional, user-friendly movie ticket booking experience.

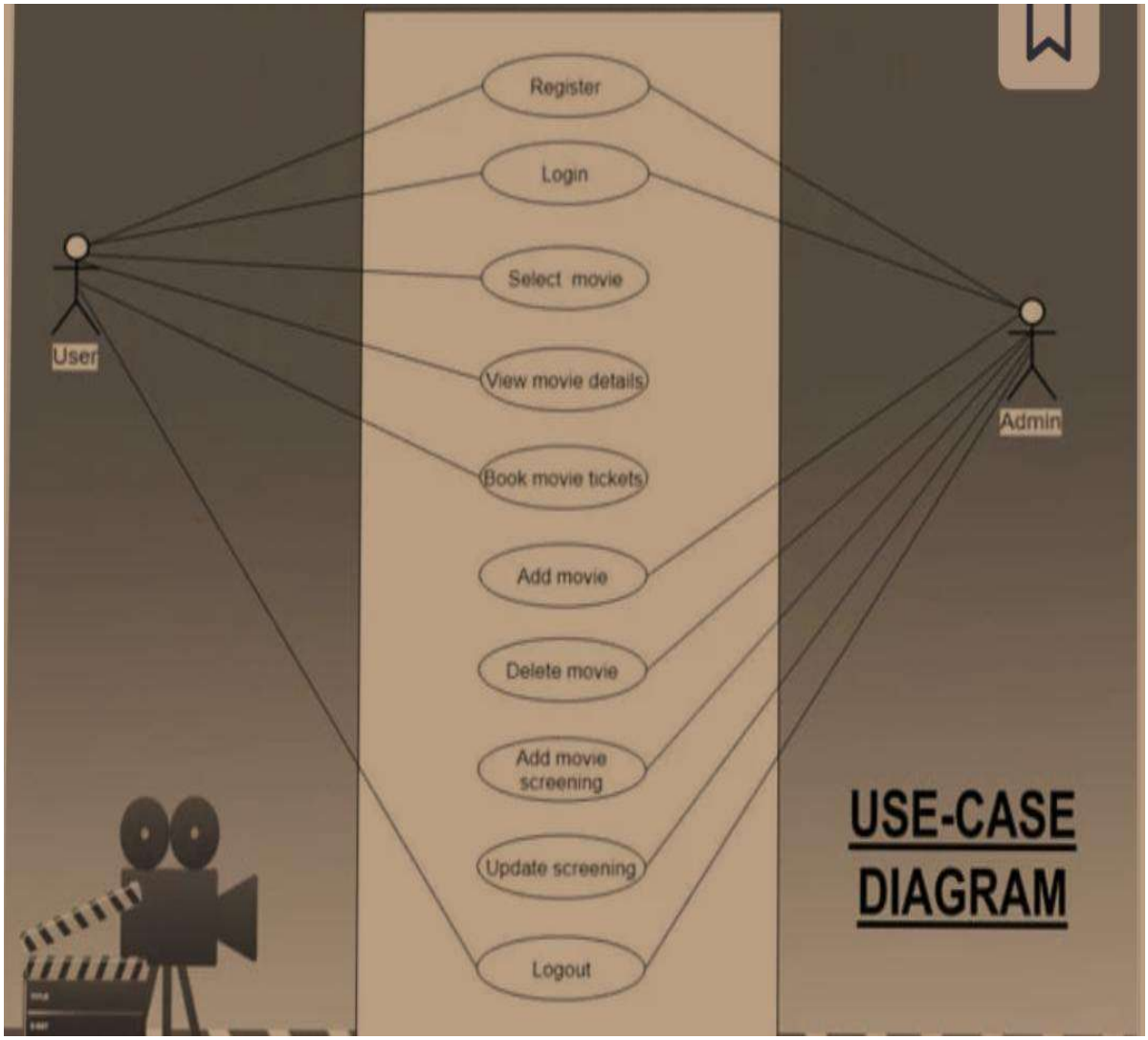
CHAPTER 2
PROJECT METHODOLOGY

2.1Proposed Work

The proposed work for the Movie Ticket Booking System aims to enhance its functionality by incorporating additional features to improve user experience and system scalability. Future improvements include database integration to store user credentials, movie listings, and booking records, allowing for a more dynamic and persistent experience. A seat selection feature will be added, enabling users to pick their preferred seats during booking. Additionally, the system will integrate a payment gateway to allow users to complete their bookings with online payments, and movie showtimes will be incorporated, enabling users to choose specific timings for their selected movies.

Furthermore, the proposed work includes user profile management, where users can create accounts, view past bookings, and manage preferences. An admin panel will be developed for administrators to manage movie listings and bookings efficiently. To improve the visual appeal, the graphical user interface (GUI) will be enhanced with modern design elements, making the system more interactive and user-friendly. These upgrades will transform the Movie Ticket Booking System into a comprehensive solution for movie reservations, suitable for real-world implementation.

2.2 Block Diagram



CHAPTER 3
MODULE DESCRIPTION

3.1 User Interface (UI) Module

This module is the first point of interaction for users. It includes the login page, where users enter their credentials, the movie selection page, where they can browse available movies, and the ticket booking page, where users choose their seats and ticket quantity.

3.2 Authentication Module

Responsible for validating user credentials. This module checks the entered username and password against stored data to authenticate the user. If authentication is successful, the system allows the user to proceed with booking. Otherwise, it prompts the user for re-entry.

3.3 Movie Database Module

This module stores and retrieves movie data, including the list of available movies, showtimes, and availability of seats. It acts as the backend database for movie information, providing necessary data for movie selection and showtime availability.

3.4 Booking Management Module

Handles ticket selection and seat management. Users can choose the number of tickets and select their preferred seats from the available options. This module ensures the user is only able to book available seats and correctly manages seat allocation to prevent double bookings.

3.5 Payment System Module

Manages online payment processing. This module integrates with payment gateways to allow users to make payments for their ticket bookings. It verifies successful payment transactions and ensures users are charged accurately based on their selections.

3.6 Confirmation Module

After payment confirmation, this module generates a booking confirmation. It sends a confirmation message to the user, displaying the ticket details (movie name, showtime, seat number) and prompts for any feedback or ratings on the booking experience.

3.7 Admin Panel Module

T his module is for administrators to manage the system. Admins can update movie listings, manage showtimes, monitor ticket bookings, and check payment statuses. They also have the ability to generate reports on sales and bookings for system maintenance and future improvements.

CHAPTER 4
CONCLUSION AND FUTURE SCOPE

4.1 Conclusion

The Movie Ticket Booking System is designed to provide a seamless, user-friendly experience for booking movie tickets. By integrating various modules, including user authentication, movie selection, booking management, and payment systems, the system ensures smooth operation from start to finish. The user interface is intuitive, making it easy for users to browse available movies, select showtimes, and complete bookings. The backend components, such as the movie database and payment system, ensure that the information is accurate and transactions are secure.

Additionally, the admin panel allows administrators to manage the movie listings, monitor ticket sales, and ensure that the system runs efficiently. Overall, this system aims to reduce the time and effort involved in purchasing movie tickets, making it more convenient for users. The integration of these different components ensures that the movie ticket booking process is not only efficient but also reliable and scalable, allowing for future enhancements.

4.2 Future Scope

The future scope of the Movie Ticket Booking System can be expanded in various ways to enhance user experience and streamline operations. User profile management can be introduced to allow users to customize their preferences and view booking history. Integrating a movie rating and review system would enable users to share feedback and view movie ratings before booking. Payment gateway integrations like PayPal or card payments would ensure secure transactions, while dynamic seat selection and the ability to choose from multiple theater locations would improve convenience. Advanced search filters, special offers, and promotions could further attract users, and a notification system would keep them informed about upcoming shows or changes. Additionally, an admin panel would help manage movie schedules and bookings, and a mobile app version would enable users to book tickets on the go. AI-based recommendations could personalize movie suggestions, and social media integration would let users share their experiences. As technology advances, VR features could also be explored for immersive cinema previews. These features would not only make the system more user-friendly but also set it up for future growth and adaptability.

References:

Herbert Schildt, "Java: The Complete Reference", McGraw Hill Education, 11th Edition.

Cay S. Horstmann, "Core Java Volume I – Fundamentals", Pearson Education, 11th Edition.

E. Balagurusamy, "Programming with Java: A Primer", McGraw Hill Education, 5th Edition.

APPENDIX A

(SOURCE CODE)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class MovieTicketBookingSystem {

    static String username = "user";
    static String password = "password";
    static String loggedInUser = "";
    static JFrame frame;
    static JPanel panel;
    static CardLayout cardLayout;

    static String selectedMovie = "";
    static int selectedSeats = 0;
    static int ticketPrice = 150;  // Default ticket price per seat
    static String seatSelection = "";
    static Map<String, Integer> moviePrices = new HashMap<>();
    static ArrayList<String> bookedSeats = new ArrayList<>();
    static JLabel priceLabel;
    static JLabel paymentDetailsLabel;

    public static void main(String[] args) {
        // Initialize movie prices
```

```
        moviePrices.put("Avengers: Endgame", 250);
        moviePrices.put("Spider-Man: No Way Home", 300);
        moviePrices.put("The Dark Knight", 200);
        moviePrices.put("Inception", 180);
        moviePrices.put("Titanic", 220);
        moviePrices.put("The Lion King", 150);

        frame = new JFrame("Movie Ticket Booking");
        frame.setSize(800, 600);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        cardLayout = new CardLayout();
        panel = new JPanel(cardLayout);

        panel.add(loginPage(), "LoginPage");
        panel.add(movieListPage(), "MovieListPage");
        panel.add(seatSelectionPage(), "SeatSelectionPage");
        panel.add(paymentPage(), "PaymentPage");
        panel.add(ticketDetailsPage(), "TicketDetailsPage");

        frame.add(panel);
        frame.setVisible(true);
    }

    // Beautified Login Page
    public static JPanel loginPage() {
        JPanel loginPanel = new JPanel(new GridBagLayout());
        loginPanel.setBackground(new Color(30, 30, 30));
```

```
        JLabel titleLabel = new JLabel("Welcome to Movie Ticket
Booking");
        titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
        titleLabel.setForeground(Color.WHITE);

        JLabel usernameLabel = new JLabel("Username:");
        usernameLabel.setForeground(Color.WHITE);
        JTextField usernameField = new JTextField(15);

        JLabel passwordLabel = new JLabel("Password:");
        passwordLabel.setForeground(Color.WHITE);
        JPasswordField passwordField = new JPasswordField(15);

        JButton loginButton = new JButton("Login");
        loginButton.setBackground(new Color(0, 153, 0));
        loginButton.setForeground(Color.WHITE);

        loginButton.addActionListener(e -> {
            String enteredUsername = usernameField.getText();
            String enteredPassword = new
String(passwordField.getPassword());

            if (enteredUsername.equals(username) &&
enteredPassword.equals(password)) {
                loggedInUser = enteredUsername;
                cardLayout.show(panel, "MovieListPage");
            } else {
                JOptionPane.showMessageDialog(frame, "Invalid username or
password.");
```

```
    }
});

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

gbc.gridx = 0;
gbc.gridy = 0;
gbc.gridwidth = 2;
loginPanel.add(titleLabel, gbc);

gbc.gridwidth = 1;
gbc.gridy++;
loginPanel.add(usernameLabel, gbc);

gbc.gridx = 1;
loginPanel.add(usernameField, gbc);

gbc.gridx = 0;
gbc.gridy++;
loginPanel.add(passwordLabel, gbc);

gbc.gridx = 1;
loginPanel.add(passwordField, gbc);

gbc.gridx = 0;
gbc.gridy++;
gbc.gridwidth = 2;
loginPanel.add(loginButton, gbc);
```

```
        return loginPanel;
    }

    // Beautified Movie List Page
    public static JPanel movieListPage() {
        JPanel moviePanel = new JPanel(new BorderLayout());
        moviePanel.setBackground(new Color(45, 45, 45));

        JLabel titleLabel = new JLabel("Select a Movie",
SwingConstants.CENTER);
        titleLabel.setFont(new Font("Arial", Font.BOLD, 24));
        titleLabel.setForeground(Color.WHITE);
        titleLabel.setBorder(BorderFactory.createEmptyBorder(20, 0, 20,
0));

        JPanel gridPanel = new JPanel(new GridLayout(3, 2, 10, 10));
        gridPanel.setBackground(new Color(45, 45, 45));

        for (String movie : moviePrices.keySet()) {
            JButton movieButton = new JButton("<html><center>" + movie
+ "<br>₹" + moviePrices.get(movie) + "</center></html>");
            movieButton.setFont(new Font("Arial", Font.BOLD, 14));
            movieButton.setBackground(new Color(70, 130, 180));
            movieButton.setForeground(Color.WHITE);

            movieButton.addActionListener(e -> {
                selectedMovie = movie;
                ticketPrice = moviePrices.get(movie);  // Get the price of the
```

```
selected movie
                    cardLayout.show(panel, "SeatSelectionPage");
                });
                gridPanel.add(movieButton);
            }

            moviePanel.add(titleLabel, BorderLayout.NORTH);
            moviePanel.add(gridPanel, BorderLayout.CENTER);

            return moviePanel;
        }

        // Seat Selection Page
        public static JPanel seatSelectionPage() {
            JPanel seatPanel = new JPanel();
            seatPanel.setLayout(new GridLayout(6, 10, 5, 5));  // 60 seats in
total

            ArrayList<JButton> seatButtons = new ArrayList<>();
            for (int i = 1; i <= 60; i++) {
                String seatLabel = "Seat " + i;
                JButton seatButton = new JButton(seatLabel);
                seatButton.setBackground(Color.GREEN);
                seatButton.addActionListener(e -> {
                    if (seatButton.getBackground() == Color.GREEN) {
                        seatButton.setBackground(Color.RED);
                        bookedSeats.add(seatLabel);
                        selectedSeats++;
                    } else {
```

```
                    seatButton.setBackground(Color.GREEN);
                    bookedSeats.remove(seatLabel);
                    selectedSeats--;
                }
                updatePriceLabel();
            });
            seatButtons.add(seatButton);
            seatPanel.add(seatButton);
        }

        priceLabel = new JLabel("Total Price: ₹" + (selectedSeats *
ticketPrice));
        JButton proceedButton = new JButton("Proceed to Payment");
        proceedButton.addActionListener(e -> {
            if (selectedSeats == 0) {
                JOptionPane.showMessageDialog(frame, "Please select at least
one seat.");
            } else {
                StringBuilder seatDetails = new StringBuilder();
                for (String seat : bookedSeats) {
                    seatDetails.append(seat).append(" ");
                }

                paymentDetailsLabel.setText("<html>Movie: " +
selectedMovie +
                        "<br>Seats Booked: " + selectedSeats +
                        "<br>Seats: " + seatDetails.toString() +
                        "<br>Total Price: ₹" + (selectedSeats * ticketPrice) +
"</html>");
```

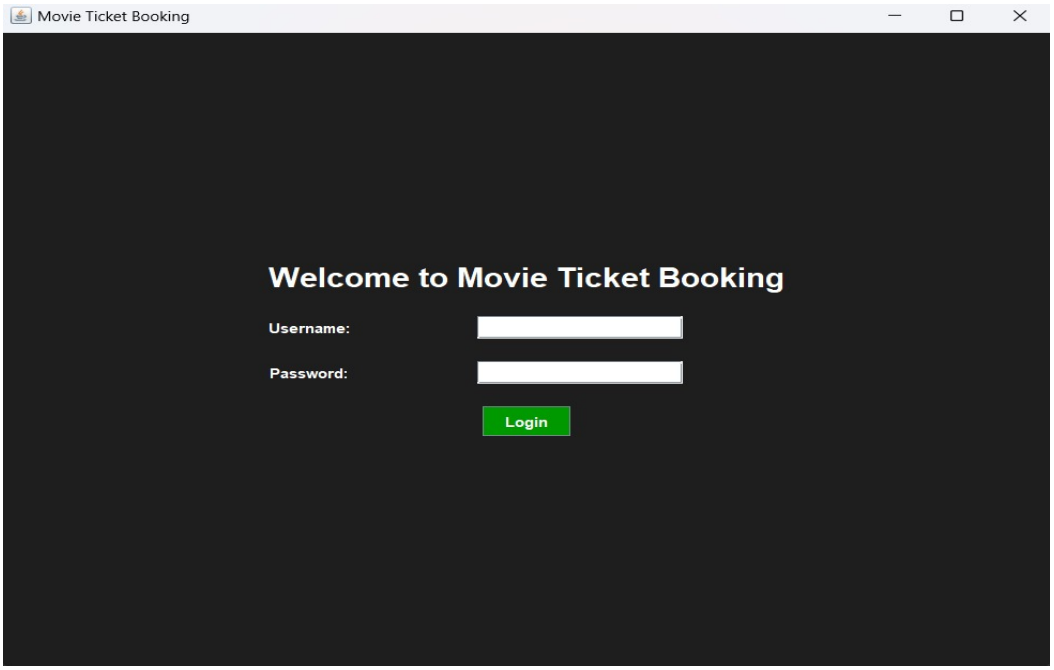
```
                cardLayout.show(panel, "PaymentPage");
            }
        });

        seatPanel.add(proceedButton);
        seatPanel.add(priceLabel);

        return seatPanel;
    }

    // Payment Page
    public static JPanel paymentPage() {
        JPanel paymentPanel = new JPanel();
        paymentPanel.setLayout(new BoxLayout(paymentPanel,
BoxLayout.Y_AXIS));

        paymentDetailsLabel = new JLabel("");
        JButton confirmButton = new JButton("Confirm Payment");

        confirmButton.addActionListener(e -> {
            JOptionPane.showMessageDialog(frame, "Payment Successful!");
            cardLayout.show(panel, "TicketDetailsPage");
        });

        paymentPanel.add(paymentDetailsLabel);
        paymentPanel.add(confirmButton);

        return paymentPanel;
    }
```

```
        // Ticket Details Page
        public static JPanel ticketDetailsPage() {
            JPanel ticketPanel = new JPanel();
            ticketPanel.setLayout(new BoxLayout(ticketPanel,
BoxLayout.Y_AXIS));

            JLabel ticketLabel = new JLabel();
            JButton backButton = new JButton("Back to Movie List");

            backButton.addActionListener(e -> {
                selectedSeats = 0;
                bookedSeats.clear();
                cardLayout.show(panel, "MovieListPage");
            });

            ticketPanel.add(ticketLabel);
            ticketPanel.add(backButton);

            return ticketPanel;
        }

        // Update price label based on the number of seats selected
        public static void updatePriceLabel() {
            priceLabel.setText("Total Price: ₹" + (selectedSeats * ticketPrice));
        }
    }
```

APPENDIX B

(SCREENSHOTS)

User Login



Movie Selection

Seat Selection

Payment Confirmation