RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

ONLINE MOVIE TICKET BOOKING SYSTEM

A MINI PROJECT REPORT

SUBMITTED BY

MADHUMITHA G	231501089
LOVISHA KAROL C	231501088
KANISHKA P	231501072

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105

RAJALAKSHMI
ENGINEERING COLLEGE

2024-2025

BONAFIDE CERTIFICATE

Certified that this project report "ONLINE TICKET BOOKING SYSTEM" is the Bonafide work of "MADHUMITHA G [231501089], LOVISHA KAROL C [231501088], KANISHKA P [231501072]" who carried out this project work under my supervision.

Submitted for the Practical Examination held on _____

SIGNATURE

Mr. U. Kumaran,
Assistant Professor (SS)
AIML,
Rajalakshmi Engineering College,
(autonomous)
Thandalam, Chennai - 602 105

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The *Online Movie Ticket Booking System* is a user-friendly and efficient application designed to streamline the process of booking movie tickets for users and managing operations for cinema administrators. Developed using Java for backend functionality and MySQL for database management, the system offers a seamless and dynamic solution catering to the needs of both users and cinema operators.

The system simplifies ticket booking by allowing users to browse movie listings, select showtimes, and reserve seats through an intuitive interface. Users can also view movie details, seating availability, and ticket prices in real-time, ensuring a hassle-free booking experience. Payment integration ensures secure and efficient transactions, enhancing user satisfaction.

On the administrative side, the system provides robust tools for managing movies, showtimes, and theater seating arrangements. Cinema operators can easily add, update, or remove movie schedules, as well as monitor ticket sales and generate comprehensive reports. These reports offer valuable insights into revenue and occupancy trends, enabling better operational decision-making.

The use of Java ensures a reliable and scalable application, while MySQL supports efficient data storage and retrieval, enabling real-time updates for both users and administrators. By automating the ticket booking process, the system eliminates the need for manual bookings, reduces errors, and enhances operational efficiency.

The *Online Movie Ticket Booking System* fosters a seamless interaction between users and cinema operators, ensuring a convenient, transparent, and efficient way to enjoy movies while supporting business growth for cinemas.

TABLE OF CONTENTS

INTRODUCTION

1.1: Project Overview:

The **Movie Ticket Booking System** is a Java-based application designed to streamline and enhance the movie ticket reservation process. It provides an efficient platform for managing show schedules, seat bookings, and payment processing, significantly reducing manual effort and minimizing errors. The system includes features for user registration, movie selection, and seat reservation, as well as secure handling of ticket payments and cancellations. A user-friendly graphical interface (built using Swing or JavaFX) ensures seamless navigation for both customers and administrators. The backend integrates with relational databases like MySQL or PostgreSQL using JDBC, providing secure and reliable data storage and retrieval.

To ensure data integrity and confidentiality, the system incorporates robust security measures such as password encryption and role-based access control. Its scalable architecture allows for future enhancements, including mobile application support, integration with online payment gateways, and advanced features like personalized recommendations and analytics for understanding user preferences. This comprehensive solution caters to the modern needs of the entertainment industry, offering efficiency, accuracy, and convenience to both cinema operators and moviegoers.

1.2: Objective:

The primary objective of the **Movie Ticket Booking System** is to develop a secure, efficient, and user-friendly platform to manage the ticket reservation process for cinemas. The system aims to automate tasks such as movie scheduling, seat booking, and payment processing, thereby reducing manual intervention and minimizing errors. It seeks to enhance the customer experience by providing seamless access to movie listings, secure bookings, and timely

booking confirmations while empowering administrators with tools for effective schedule management and data analysis.

Additionally, the system is designed to ensure data integrity, confidentiality, and scalability, making it adaptable to evolving requirements of the entertainment industry. By leveraging Java's robust features, the system delivers a reliable solution that meets the operational and security needs of modern cinema operations.

MODULES:

1.Main Class:

The **Main** class serves as the entry point of the application. It initializes the system and provides a menu for users to either log in or sign up. This class primarily invokes other classes like **Login** or **SignUp** to guide the user through the system's workflow.

2.Login Class:

The **Login** class is responsible for authenticating users. It collects the username and password, verifies them against stored credentials in the database, and provides access upon successful authentication. If the credentials are incorrect, it prompts the user to try again. The class also offers an option to redirect new users to the **Sign Up** process.

3.Connection Class:

The **Connection** class manages the database connection. It establishes a secure link between the application and the database (e.g., MySQL or PostgreSQL) and provides methods for executing queries, such as validating login credentials, storing user data, or retrieving booking information.

4.SignUp Class:

The **SignUp** class handles the registration process for new users. It collects user details, such as name, contact information, and preferred account setup, and securely stores this information in the database. This class ensures all required fields are filled and validates the input before submission.

5.SignUp 2 Class:

This class extends the **SignUp** process for additional details, such as setting up user preferences (e.g., favorite genres or theatres) or verifying identity documents. It serves as an intermediary step for more detailed user registration.

6.SignUp 3 Class:

The final stage in the registration process, the **SignUp3** class confirms the user's details and stores them in the database. It provides a summary of the entered information and allows users to make corrections before final submission. Upon completion, it redirects the user to the login page.

7.Movieselection Class:

The **Movie Selection** class allows users to browse available movies and select one for booking. It retrieves movie details, such as title, showtimes, and theatre location, from the database and displays them to the user. This class ensures a user-friendly interface for choosing a movie.

8.SeatSelection Class:

The **Seat Selection** class handles the process of selecting seats for a chosen movie. It displays the available seats in a visual format and allows users to pick their preferred seats. Once selected, it updates the database to mark the seats as reserved.

9. Payment Processing Class:

The **Payment Processing** class facilitates secure payment for movie tickets. It supports multiple payment options, such as credit cards, debit cards, and digital wallets. The class ensures the entered payment details are valid and processes the transaction securely before confirming the booking.

10. Fast Booking Class:

The **Fast Booking** class offers users a quick way to book predefined packages, such as two tickets for a popular movie or family bundles. This feature allows users to skip manual selections and access popular deals quickly, enhancing convenience.

11. Booking History Class:

The Booking History class provides users with a summary of their past bookings. It retrieves ticket details, such as movie name, showtime, and seat numbers, from the database and displays them to the user. This feature helps users keep track of their previous reservations.

12. Profile Management Class:

The **Profile Management** class allows users to update their account details, such as name, contact information, and preferences. It ensures that any changes are validated and securely stored in the database. This class enhances the user experience by allowing personalization.

13. Cancellation Class:

The **Cancellation** class enables users to cancel their ticket bookings. It verifies the booking details, processes the refund (if applicable), and updates the seat availability in the database. This feature ensures flexibility and convenience for users.

14. Ticket Generation Class:

The **Ticket Generation** class generates digital tickets after successful booking. It includes details such as the movie name, showtime, theatre location, and seat number. The generated ticket can be saved or printed by the user. This class ensures an efficient and user-friendly ticketing experience.

15. Offers and Promotions Class:

The **Offers and Promotions** class showcases current deals, discounts, and promotional offers available for users. It retrieves offer details from the database and displays them, helping users save on their bookings.

CHAPTER 2: SYSTEM DESIGN

System Design is a critical phase in software development where the system architecture and components are structured and detailed. This chapter focuses on designing the Movie Ticket Booking System, detailing its structure, components, data flow, user interfaces, and interaction with databases.

2.1. System Architecture

The Movie Ticket Booking System follow a Client-Server Architecture, where the client interacts with the system through a user interface, and the server processes requests, interacts with the database, and sends responses back to the client. The architecture is divided into three main layers:

- Presentation Layer: The user interface (UI) layer where users interact with the system. This includes login screens, movie selection pages, booking confirmations, etc.
- Business Logic Layer: Contains the core functionality and logic of the system, such as processing bookings, validating payments, and managing seat availability.
- Data Layer: The database layer where user accounts, movie details, and booking information are stored and managed.

2.2. Database Design

The database design outlines how data is stored and accessed in the Movie Ticket Booking System. We use MySQL or any relational database for storing user and booking data. Here is a simplified version of the database schema:

Tables:

- Users Table:

    - user_id (Primary Key)
    - name
    - email
    - password
    - phone
- Movies Table:
    - movie_id (Primary Key)
    - title
    - genre
    - duration
    - rating
- Showtimes Table:
    - showtime_id (Primary Key)
    - movie_id (Foreign Key referencing Movies table)
    - theater
    - date_time
- Seats Table:
    - seat_id (Primary Key)
    - showtime_id (Foreign Key referencing Showtimes table)
    - seat_number
    - status (Available/Reserved)
- Bookings Table:

- booking_id (Primary Key)
- user_id (Foreign Key referencing Users table)
- showtime_id (Foreign Key referencing Showtimes table)
- seat_id (Foreign Key referencing Seats table)
- payment_status (Paid/Pending)

Relationships:

- One user can make multiple bookings (one-to-many relationship).
- One movie can have multiple showtimes (one-to-many relationship).
- One showtime can have multiple seats (one-to-many relationship).

2.3. System Components

User Interface (UI):

The UI provides a friendly environment for users to interact with the system. It includes:

- Login Screen: A secure login interface for authentication.
- Dashboard: A central interface displaying available movies, showtimes, and booking options.
- Movie Selection Screen: Displays movies, genres, and showtimes for user selection.
- Seat Selection Screen: Provides a visual representation of available seats for selection.
- Payment Screen: Enables users to complete their payment securely.
- Booking History: Displays past and upcoming bookings.

Business Logic:

This layer contains the core logic for each feature, such as:

- Login/Authentication: Verifies user credentials and securely manages login sessions.
- Movie Management: Handles the listing and retrieval of movie and showtime details.
- Booking Processing: Manages seat selection, availability checks, and booking confirmation.
- Payment Processing: Secures payment validation and transaction completion.

Database Interaction:

- CRUD Operations: The system performs Create, Read, Update, and Delete operations on the database for user data, bookings, and movie details.
- Query Execution: SQL queries are used to retrieve movie schedules, seat availability, and user booking history.

2.4. Use Case Diagrams

Use case diagrams visually represent the system's functionalities and interactions. Key use cases include:

- User Authentication: Log in using credentials.
- Movie Selection: Browse and select a movie.
- Seat Reservation: Choose available seats for a selected showtime.
- Payment Processing: Make payments for ticket bookings.
- View Booking History: Check past and upcoming bookings.

2.5. Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) represents how data moves through the system.

Level 1 DFD:

- The user logs in by providing credentials, which the system validates against the database.
- The user selects a movie and showtime; the system retrieves available seats from the database.
- The user reserves a seat and completes payment; the system updates the booking and payment status.

Level 2 DFD:

- Detailed interactions, such as selecting a seat, processing payments, and updating seat availability, are depicted.

2.6. Sequence Diagrams

Sequence diagrams show how objects interact in a specific sequence to perform a function. Examples:

- Login Sequence: The user enters credentials, which the system validates. Upon success, the user accesses the dashboard.
- Booking Sequence: The user selects a movie, chooses seats, processes payment, and receives booking confirmation.

2.7. Security Design

Security is critical in the Movie Ticket Booking System. Key aspects include:

- Authentication: Ensuring only authorized users can access the system through secure login processes.

- Encryption: Storing sensitive information like passwords in an encrypted format.
- Session Management: Securely managing user sessions to prevent unauthorized access.
- Payment Security: Using secure payment gateways to protect transaction details.

2.8. System Flow

The system flow explains how different components of the Movie Ticket Booking System interact:

1. User Login: The system validates user credentials and grants access.
2. Dashboard: Users browse movies and showtimes.
3. Booking Execution: Users select a movie, choose seats, and make payments.
4. Ticket Generation: The system confirms the booking and provides a digital ticket.
5. Logout: Users log out after completing their tasks, securely closing the session.

CHAPTER 3: FUNCTIONAL REQUIREMENT

Functional Requirements: Movie Ticket Booking System

Functional requirements define the specific behaviours, functions, and processes that the **Movie Ticket Booking System** must support. These requirements describe the system's interactions with users and other systems, focusing on the desired capabilities and features needed to meet its objectives.

3.1. User Authentication

Description:

The system must support secure user authentication to ensure that only authorized individuals can access their accounts.

Functional Requirements:

- Users must be able to log in using a valid email and password.
- The system must authenticate the user based on the provided credentials.
- The system must enforce strong password policies (e.g., minimum length, combination of characters).
- The system should allow users to reset their password if they forget it.
- After successful login, users should be directed to their dashboard.
- The system should lock the account after multiple failed login attempts, requiring manual intervention or a secure process to unlock it.

3.2. Movie and Showtime Management

Description:

The system must allow users to browse and select movies and available showtimes.

Functional Requirements:

- Users should be able to view the list of currently showing movies, including their details (e.g., title, genre, rating, duration).
- The system must allow users to filter movies by genre, language, or rating.
- Users must be able to select a preferred showtime for a chosen movie.
- The system must display the showtimes in chronological order with theatre details.

3.3. Seat Selection

Description:

The system must provide an interactive interface for users to select available seats.

Functional Requirements:

- Users must be able to view a visual representation of the theater seating arrangement, including available and reserved seats.
- The system must allow users to select one or more seats from the available options.
- The system should automatically reserve the selected seats temporarily during the booking process to prevent conflicts.

3.4. Ticket Booking

Description:

The system must allow users to book tickets for selected movies and showtimes.

Functional Requirements:

- Users must be able to confirm their seat selection and proceed with booking.
- The system should calculate the total cost based on the number of tickets and ticket type (e.g., adult, child, senior).
- Users must be able to view a summary of their booking, including movie name, showtime, seat numbers, and total price, before confirming.
- The system must update the seat status to "reserved" or "booked" upon successful payment.

3.5. Payment Processing

Description:

The system must support secure and seamless payment processing for ticket bookings.

Functional Requirements:

- Users must be able to choose from multiple payment options, such as credit/debit cards, mobile wallets, or net banking.
- The system should validate payment details before processing the transaction.
- Upon successful payment, users should receive a confirmation of their booking.
- Failed payment transactions must not confirm the booking, and the reserved seats should be released.

3.6. Booking History

Description:

The system must allow users to view their past and upcoming bookings.

Functional Requirements:

- Users must be able to view all their bookings, including movie details, showtime, seat numbers, and payment status.
- The system should allow users to filter bookings by date or status (upcoming/completed).
- Users should be able to retrieve their e-tickets from the booking history for entry to the theatre

3.7. Notifications

Description:

The system must notify users about important booking-related activities.

Functional Requirements:

- Users must receive email or SMS notifications for successful bookings, including booking details and a unique booking ID.
- The system should notify users of upcoming showtimes.
- In case of cancellation or changes to the showtime, the system must notify affected users promptly.

3.8. Cancellation and Refunds

Description:

The system must support ticket cancellation and initiate refunds as per the cancellation policy.

Functional Requirements:

- Users should be able to cancel bookings before the showtime, subject to the theatre's cancellation policy.
- The system must calculate the refund amount based on the cancellation policy and initiate the refund process.
- The system should update the seat status to "available" upon successful cancellation.

3.9. Reporting and Analytics

Description:
The system must provide reporting features for administrators to monitor bookings and revenue.

Functional Requirements:

- Administrators must be able to generate reports on movie-wise or theater-wise ticket sales.
- The system should provide daily, weekly, and monthly revenue reports.
- Reports must include details such as total bookings, revenue, and cancellation trends.
- Users should be able to download reports in formats such as CSV or PDF.

3.10. Security and Session Management

Description:
The system must ensure the security of user data and manage sessions effectively.

Functional Requirements:

- All sensitive user data, including passwords and payment information, must be securely stored using encryption.

- The system should log users out after a period of inactivity to prevent unauthorized access.
- Users must explicitly log out to end their session, and the system should securely close all active sessions.

3.11. Backup and Recovery

Description:

The system must support backup and recovery mechanisms to prevent data loss.

Functional Requirements:

- The system should perform regular backups of movie schedules, user data, and booking records.
- In the event of a system failure, data should be restored from the most recent backup with minimal downtime.
- The backup process must ensure transaction integrity to avoid discrepancies.

3.12. Error Handling and Validation

Description:

The system must validate all inputs and handle errors gracefully.

Functional Requirements:

- The system must validate user inputs, such as email formats, payment details, and seat selections.
- The system should display clear error messages for invalid inputs (e.g., "Selected seats are no longer available").
- Unexpected errors (e.g., payment gateway failures) must be handled gracefully, and users should be informed of the issue.

CHAPTER 4: SYSTEM IMPLEMENTATION

System Implementation is the phase where the actual coding and construction of the system take place. This chapter outlines how the **Movie Ticket Booking System** is developed, including the technologies used, the system architecture, programming languages, and the steps followed to transform the functional requirements into a working system.

4.1. Overview of System Implementation

The **Movie Ticket Booking System** is implemented using **Java** as the primary programming language and **MySQL** for database management. The system is designed as a standalone desktop application where users interact via a graphical user interface (GUI) built using **Java Swing**. A modular approach is followed, with each functionality—such as user authentication, seat selection, ticket booking, and payment processing—developed as separate modules or classes.

The system backend interacts with the database to manage movie schedules, user data, seat availability, and booking history. The code is organized using the **Model-View-Controller (MVC)** design pattern to ensure separation of concerns, scalability, and ease of maintenance.

1.HTML - Structure of the Front-End

HTML (HyperText Markup Language) forms the backbone of the user interface, defining the structure of the pages. The HTML code outlines the different sections such as the login page, dashboard, attendance marking form, and reports.

2. CANCEL.JAVA

```
public class CANCEL extends javax.swing.JFrame {

    /**
     * Creates new form CANCEL
     */
    public CANCEL() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
```

```
        jLabel1 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 204)));

        jLabel1.setBackground(new java.awt.Color(255, 255, 255));
        jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 36)); //
NOI18N
        jLabel1.setForeground(new java.awt.Color(0, 0, 51));
        jLabel1.setText("YOUR BOOKING IS CANCELLED");

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(96, 96, 96)
                .addComponent(jLabel1)
                .addContainerGap(96, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(
```

```
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
                .addContainerGap(46, Short.MAX_VALUE)
                .addComponent(jLabel1)
                .addGap(38, 38, 38))
        );

        jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(204, 0, 204)));

        jLabel2.setFont(new java.awt.Font("Algerian", 1, 24)); // NOI18N
        jLabel2.setForeground(new java.awt.Color(0, 0, 255));
        jLabel2.setText("PLEASE  VIST AGAIN");

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(195, 195, 195)
                .addComponent(jLabel2)
                .addContainerGap(282, Short.MAX_VALUE))
        );
```

```
        jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(40, 40, 40)
                .addComponent(jLabel2)
                .addContainerGap(44, Short.MAX_VALUE))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(169, 169, 169)
                        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(195, 195, 195)
                        .addComponent(jPanel2,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap(37, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(71, 71, 71)
                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(79, 79, 79)
                .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(60, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
```

```
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(CANCEL.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(CANCEL.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(CANCEL.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(CANCEL.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new CANCEL().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    // End of variables declaration//GEN-END:variables
}
```

3) cardselection.java

```
import javax.swing.ButtonGroup;
public class cardselection extends javax.swing.JFrame {
 public cardselection() {
        initComponents();
        groupButton();
    }
    private void groupButton() {

        ButtonGroup bg1= new ButtonGroup();
        bg1.add(jRadioButton1);
        bg1.add(jRadioButton2);
        bg1.add(jRadioButton3);
       bg1.add(jRadioButton4);



    }
    String a;String b;String c;String e;String f;
            int N1;
    public cardselection(String p1,String p2,String p3,String p4,int p5,String p6){
    initComponents();
    this.a=p1;
    this.b=p2;
    this.c=p3;
    this.e=p4;
    this.N1=p5;
    this.f=p6;
```

```
    jLabel6.setText(p4);
    jLabel7.setText(p2);
    jLabel8.setText(p1);
    jLabel9.setText(p6);
    jLabel10.setText(String.valueOf(p5));
    jLabel12.setText(p3);
    }

    cardselection(String a, String b, String c, String e, String f, int N1) {
        throw new UnsupportedOperationException("Not supported yet."); //To
change body of generated methods, choose Tools | Templates.
    }
        @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jRadioButton2 = new javax.swing.JRadioButton();
        jRadioButton3 = new javax.swing.JRadioButton();
        jRadioButton4 = new javax.swing.JRadioButton();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jRadioButton1 = new javax.swing.JRadioButton();
        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
```

```
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        jLabel12 = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
"PAYMENT TYPE",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Times New Roman", 1, 36), new java.awt.Color(0, 0, 255))); //
NOI18N

        jRadioButton2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jRadioButton2.setText("Debit card");

        jRadioButton3.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jRadioButton3.setText("NET BANKING");

        jRadioButton4.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
```

```
        jRadioButton4.setText("PAYTM WALLET");
        jRadioButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jRadioButton4ActionPerformed(evt);
            }
        });

        jButton1.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jButton1.setText("Make Payment");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jButton2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jButton2.setText("Go Back");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jRadioButton1.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jRadioButton1.setText("Credit card");
        jRadioButton1.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jRadioButton1ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(85, 85, 85)
            .addComponent(jButton1)
            .addGap(82, 82, 82)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addComponent(jRadioButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 143,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jRadioButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 167,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jRadioButton3)
                .addComponent(jRadioButton4))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
```

```
301, Short.MAX_VALUE)
                .addComponent(jButton2)
                .addGap(89, 89, 89))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jRadioButton1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(18, 18, 18)
                .addComponent(jRadioButton2)
                .addGap(18, 18, 18)
                .addComponent(jRadioButton3)
                .addGap(18, 18, 18)
                .addComponent(jRadioButton4)
                .addGap(40, 40, 40))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                    .addComponent(jButton2)
```

```
                .addComponent(jButton1))
            .addContainerGap())
        );

        jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
"CONFIRM DETAILS",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Times New Roman", 1, 36), new java.awt.Color(204, 0, 0))); //
NOI18N

        jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel1.setText("MOVIE:");

        jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel2.setText("THEATER:");

        jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel3.setText("DATE:");

        jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel4.setText("TIME:");

        jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
```

```
        jLabel5.setText("NO OF TICKETS:");

        jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel6.setText("jLabel6");

        jLabel7.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel7.setText("jLabel7");

        jLabel8.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel8.setText("jLabel8");

        jLabel9.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel9.setText("jLabel9");

        jLabel10.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel10.setText("jLabel10");

        jLabel11.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel11.setText("FARE:");

        jLabel12.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel12.setText("jLabel12");
```

```
        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(jPanel2Layout.createSequentialGroup()
               .addGap(62, 62, 62)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                     .addGroup(jPanel2Layout.createSequentialGroup()
                        .addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
                        .addComponent(jLabel6))
                     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                           .addComponent(jLabel1)
                           .addComponent(jLabel11))
                        .addGap(29, 29, 29)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
```

```
ment.LEADING)
                    .addComponent(jLabel10)
                    .addComponent(jLabel8))
                .addGap(52, 52, 52)))
            .addGap(185, 185, 185)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addComponent(jLabel2)
                    .addGap(47, 47, 47)
                    .addComponent(jLabel7))
                .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addComponent(jLabel3)
                    .addComponent(jLabel4))
                .addGap(32, 32, 32)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.TRAILING)
                    .addComponent(jLabel9)
                    .addComponent(jLabel12))))
            .addContainerGap(221, Short.MAX_VALUE))
        );
        jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
```

```
ING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(42, 42, 42)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(jLabel6)
                .addComponent(jLabel7)
                .addComponent(jLabel5))
            .addGap(54, 54, 54)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                .addComponent(jLabel8)
                .addComponent(jLabel3)
                .addComponent(jLabel9))
            .addComponent(jLabel1))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
53, Short.MAX_VALUE)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel10)
                    .addComponent(jLabel11))
                .addGap(26, 26, 26))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel12)
                    .addComponent(jLabel4))
                .addGap(37, 37, 37))))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(38, 38, 38)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap(21, 21, 21)
            .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
16, Short.MAX_VALUE)
            .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
        );
```

```
        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void jRadioButton4ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jRadioButton4ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_jRadioButton4ActionPerformed

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
this.setVisible(false);       new payment(a,b,c,e,N1,f).setVisible(true); // TODO
add your handling code here:

    }//GEN-LAST:event_jButton1ActionPerformed

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton2ActionPerformed
this.setVisible(false);       new recepit().setVisible(true); // TODO add your
handling code here:

    }//GEN-LAST:event_jButton2ActionPerformed

    private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jRadioButton1ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_jRadioButton1ActionPerformed

    /**
     * @param args the command line arguments
```

```
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(cardselection.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(cardselection.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(cardselection.class.getName()).log(java.util.
```

```
logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(cardselection.class.getName()).log(java.util.
logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new cardselection().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
```

```
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JRadioButton jRadioButton3;
    private javax.swing.JRadioButton jRadioButton4;
    // End of variables declaration//GEN-END:variables
}
```

4)login.java

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

public class login extends javax.swing.JFrame {

    private Timer timer;
    private int timeRemaining = 180; // 3 minutes in seconds

    public login() {
        initComponents();
        startTimer(); // Start the timer when the login window is displayed
    }

    @SuppressWarnings("unchecked")
    private void initComponents() {
```

```
        jPanel1 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        submit = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jPasswordField1 = new javax.swing.JPasswordField();
        timerLabel = new javax.swing.JLabel(); // Label for showing the timer
        jLabel1 = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

        jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel2.setText("USERNAME");

        jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel3.setText("PASSWORD");

        jTextField1.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        submit.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
```

```
NOI18N
        submit.setText("LOGIN");
        submit.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                submitActionPerformed(evt);
            }
        });

        jButton2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jButton2.setText("NEW REGISTER");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jPasswordField1.setFont(new java.awt.Font("Times New Roman", 1, 24));
// NOI18N

        timerLabel.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
Timer label
        timerLabel.setText("Time remaining: 03:00"); // Default text for 3 minutes

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
```

```
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(21, 21, 21)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addComponent(jLabel3)
                                .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 174,
javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addGap(59, 59, 59)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                .addComponent(jTextField1,
javax.swing.GroupLayout.DEFAULT_SIZE, 160, Short.MAX_VALUE)
                                .addComponent(jPasswordField1))
                            .addGap(0, 0, Short.MAX_VALUE))
                        .addGroup(jPanel1Layout.createSequentialGroup()
                            .addComponent(submit)
                            .addGap(85, 85, 85)
                            .addComponent(jButton2)
                            .addContainerGap(112, Short.MAX_VALUE))))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
```

```
jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addComponent(timerLabel)
                    .addGap(162, 162, 162))
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                    .addContainerGap(70, Short.MAX_VALUE)
                    .addComponent(timerLabel) // Add the timer label
                    .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                        .addComponent(jLabel2)
                        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(41, 41, 41)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                        .addComponent(jLabel3)
```

```
                        .addComponent(jPasswordField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGap(61, 61, 61)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(submit)
                        .addComponent(jButton2))
                    .addGap(45, 45, 45))
        );

        jLabel1.setFont(new java.awt.Font("Algerian", 1, 36)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(51, 51, 255));
        jLabel1.setText("Online Movie Ticket Booking");

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(137, 137, 137)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel1)
```

```
                                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addContainerGap(175, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                            .addContainerGap(57, Short.MAX_VALUE)
                            .addComponent(jLabel1)
                            .addGap(45, 45, 45)
                            .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(69, 69, 69))
        );

        pack();
    }

    private void submitActionPerformed(java.awt.event.ActionEvent evt) {
        stopTimer(); // Stop the timer upon login attempt
        workWithDatabase();
    }
```

```
    private void workWithDatabase() {
        Connection c = null;
        Statement s = null;
        ResultSet rs = null;

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            c =
DriverManager.getConnection("jdbc:mysql://localhost:3306/UserRegistrationD
B", "root", "Janet_88");
            s = c.createStatement();

            String username = jTextField1.getText();
            String password = new String(jPasswordField1.getPassword());

            if (username.isEmpty() || password.isEmpty()) {
                JOptionPane.showMessageDialog(this, "Please enter valid details");
                return;
            }

            String query = "SELECT name, password FROM register";
            rs = s.executeQuery(query);

            boolean isValidUser = false;

            while (rs.next()) {
                String dbUsername = rs.getString("name");
                String dbPassword = rs.getString("password");
```

```
            if (username.equals(dbUsername) && password.equals(dbPassword))
{
                isValidUser = true;
                break;
            }
        }

        if (isValidUser) {
            JOptionPane.showMessageDialog(this, "Logged in successfully!");
            this.setVisible(false); // Hide login window
            new movie().setVisible(true); // Open movie selection screen
        } else {
            JOptionPane.showMessageDialog(this, "Invalid username or
password.");
        }
    } catch (Exception e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Database error: " +
e.getMessage());
    } finally {
        try {
            if (rs != null) rs.close();
            if (s != null) s.close();
            if (c != null) c.close();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

```
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        stopTimer(); // Stop the timer upon navigating to the registration page
        this.setVisible(false);
        new register().setVisible(true);
    }

    private void startTimer() {
        timer = new Timer(1000, new ActionListener() { // Update every second
            @Override
            public void actionPerformed(ActionEvent e) {
                timeRemaining--;
                int minutes = timeRemaining / 60;
                int seconds = timeRemaining % 60;
                timerLabel.setText(String.format("Time remaining: %02d:%02d",
minutes, seconds));

                if (timeRemaining <= 0) {
                    timer.stop();
                    JOptionPane.showMessageDialog(null, "Session expired. Please
restart the application.");
                    System.exit(0); // Exit the application
                }
            }
        });
        timer.start();
    }

    private void stopTimer() {
```

```
        if (timer != null) {
            timer.stop(); // Stop the timer to prevent session expiration
        }
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(() -> new login().setVisible(true));
    }

    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPasswordField jPasswordField1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JButton submit;
    private javax.swing.JLabel timerLabel; // Timer label
}
```

5)movie.java

```
public class movie extends javax.swing.JFrame {

    /**
     * Creates new form movie
     */
    public movie() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jLabel6 = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jComboBox1 = new javax.swing.JComboBox<>();
        jComboBox2 = new javax.swing.JComboBox<>();
        jComboBox3 = new javax.swing.JComboBox<>();
        jComboBox4 = new javax.swing.JComboBox<>();
```

```
        jLabel5 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
"MOVIE DETAILS",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Algerian", 1, 36), new java.awt.Color(0, 204, 204))); // NOI18N
        jPanel1.setFont(new java.awt.Font("Times New Roman", 1, 36)); //
NOI18N

        jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel1.setText("SELECT MOVIE:");

        jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel2.setText("SELECT  THEATRE:");

        jComboBox1.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "Robo 2", "sahoo", "arvandi sametha" }));
```

```
        jComboBox2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jComboBox2.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "srivishnu", "galaxy", "asscars", "PVR" }));

        jComboBox3.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jComboBox3.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "7:00AM", "11:00AM", "2:00PM", "6:00PM", "9:00PM" }));

        jComboBox4.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jComboBox4.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "11-11-2018", "12-11-2018", "10-11-2018", "09-11-2018" }));
        jComboBox4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jComboBox4ActionPerformed(evt);
            }
        });

        jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel5.setText("SELECT DATE:");

        jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel3.setText("SELECT TIME:");
```

```
        jTextField1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // 
NOI18N

        jButton1.setFont(new java.awt.Font("Algerian", 1, 24)); // NOI18N
        jButton1.setText("SUBMIT");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 24)); // 
NOI18N
        jLabel4.setText("NO OF TICKETS:");

        javax.swing.GroupLayout jPanel1Layout = new 
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(31, 31, 31)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(190, 190, 190)
                        .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 131,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel4)
                        .addGap(74, 74, 74)
                        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 45,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap(206, Short.MAX_VALUE))
            .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addComponent(jLabel3)
                    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 203,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 251,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel5,
javax.swing.GroupLayout.PREFERRED_SIZE, 210,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)
                    .addComponent(jComboBox1, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jComboBox2, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jComboBox4, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jComboBox3, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGap(60, 60, 60))))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(58, 58, 58)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(48, 48, 48)
                    .addComponent(jLabel2))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(37, 37, 37)
                    .addComponent(jComboBox2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(34, 34, 34)
                    .addComponent(jLabel5))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(23, 23, 23)
                    .addComponent(jComboBox4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(35, 35, 35)
```

```
                    .addComponent(jLabel3))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(25, 25, 25)
                    .addComponent(jComboBox3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
39, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addComponent(jLabel4,
javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jTextField1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(40, 40, 40)
            .addComponent(jButton1)
            .addContainerGap())
        );

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
```

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(137, 137, 137)
                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(209, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(27, 27, 27))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
```

```
    workWithDatabase();
}
 public void workWithDatabase()
{
     String a=(String) jComboBox1.getSelectedItem();
   String b=(String) jComboBox2.getSelectedItem();
    String e=(String) jComboBox3.getSelectedItem();
    String f=(String) jComboBox4.getSelectedItem();
   String d=jTextField1.getText();


     Connection c=null;
     Statement  s=null;

     ResultSet rs=null;
     int flag=0;

    //if(!a.equals(""))
     String N=jTextField1.getText();
    int N1=Integer.parseInt(N);
    int N2=N1;
     N1*=100;
   //new recepit(a,b,e,d,N1,f).setVisible(true);

  if(N2<10 && N2>0)

     {
        this.setVisible(false);  new recepit(a,b,e,d,N1,f).setVisible(true);
```

```
        try
   {
      Class.forName("com.mysql.jdbc.Driver");

c=DriverManager.getConnection("jdbc:mysql://localhost/java_dbmovies","root
","");
       s=c.createStatement();
        b=(String) jComboBox2.getSelectedItem();
       e=(String) jComboBox3.getSelectedItem();
      String q1=b;
       String q2=e;
       rs=s.executeQuery("select tickets from table3 where
theatre="+"'"+q1+"'"+" and shows="+"'"+q2+"'");
      String bid=jTextField1.getText();
      int id=Integer.parseInt(bid);
      rs=s.executeQuery("select tickets from table3 where theatre="+"'"+q1+"'"+"
and shows="+"'"+q2+"'");
       while(rs.next())
     {
    String id1=rs.getString("tickets");
     int id2=Integer.parseInt(id1);


     id2=id2-N2;


     s.executeUpdate("Update table3 set tickets="+id2+" where
theatre="+"'"+q1+"'"+" and shows="+"'"+q2+"'");
    }
    //new recepit(a,b,e,d).setVisible(true);
```

```
        while(rs.next())
        {
              String tickets1=rs.getString("tickets");
              String q3 = tickets1;

              //jLabel5 = new javax.swing.JLabel("tickets available:"+"  "+q3);

         }


       //rs.close;
       //s.close;
       //c.close;
    }
         catch(SQLException | ClassNotFoundException e1)
         {
            System.out.println(e1);
         }









         }

// TODO add your handling code here:
    }//GEN-LAST:event_jButton1ActionPerformed

    private void jComboBox4ActionPerformed(java.awt.event.ActionEvent evt)
```

```
{//GEN-FIRST:event_jComboBox4ActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_jComboBox4ActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(movie.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(movie.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(movie.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(movie.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new movie().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JComboBox<String> jComboBox1;
    private javax.swing.JComboBox<String> jComboBox2;
    private javax.swing.JComboBox<String> jComboBox3;
    private javax.swing.JComboBox<String> jComboBox4;
    private javax.swing.JLabel jLabel1;
```

```
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration//GEN-END:variables
}
```

6)payment.java

```
public class payment extends javax.swing.JFrame {

    /**
     * Creates new form payment
     */
    public payment() {
        initComponents();
    }
       String a;String b;String c;String e;String f;
            int N1;
    public payment(String p1,String p2,String p3,String p4,int p5,String p6){
    initComponents();
    this.a=p1;
    this.b=p2;
    this.c=p3;
    this.e=p4;
    this.N1=p5;
```

```
    this.f=p6;
    jLabel10.setText(p4);
    jLabel14.setText(p2);
    jLabel12.setText(p1);
    jLabel15.setText(p6);
    jLabel13.setText(String.valueOf(p5));
    jLabel16.setText(p3);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jButton1 = new javax.swing.JButton();
        jComboBox1 = new javax.swing.JComboBox<>();
        jComboBox2 = new javax.swing.JComboBox<>();
        jPasswordField1 = new javax.swing.JPasswordField();
        jLabel1 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        jTextField2 = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
```

```
        jLabel4 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        jLabel12 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();
        jLabel14 = new javax.swing.JLabel();
        jLabel15 = new javax.swing.JLabel();
        jLabel16 = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
"CARD DETAILS",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Algerian", 1, 36), new java.awt.Color(0, 204, 204))); // NOI18N
        jPanel1.setForeground(new java.awt.Color(0, 102, 102));

        jButton1.setFont(new java.awt.Font("Algerian", 1, 24)); // NOI18N
        jButton1.setText("proceed");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton1ActionPerformed(evt);
        }
    });

    jComboBox1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
    jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12" }));

    jComboBox2.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
    jComboBox2.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "2018", "2019", "2020", "2021", "2022" }));

    jPasswordField1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N

    jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
    jLabel1.setText("Card number");

    jTextField1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N

    jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
    jLabel2.setText("Card holder name");
```

```
        jTextField2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel3.setText("CVV");

        jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel4.setText("EXP");

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(330, 330, 330)
                        .addComponent(jButton1))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.TRAILING, false)
                        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
```

```
jPanel1Layout.createSequentialGroup()
                .addGap(248, 248, 248)
                .addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jPasswordField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 41,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
42, Short.MAX_VALUE)
                .addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
                .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jComboBox2,
javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel1Layout.createSequentialGroup()
                .addGap(246, 246, 246)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
```

```
ment.LEADING, false)
                    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 231,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 238,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField1,
javax.swing.GroupLayout.DEFAULT_SIZE, 352, Short.MAX_VALUE)
                    .addComponent(jTextField2)))))
            .addContainerGap(361, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
```

```
            .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 21,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(27, 27, 27)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.BASELINE)
                .addComponent(jLabel3)
                .addComponent(jPasswordField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel4)
                .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jComboBox2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(33, 33, 33)
            .addComponent(jButton1)
            .addGap(39, 39, 39))
```

```
        );

        jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
"CONFORM BOOKING",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Algerian", 1, 36), new java.awt.Color(0, 0, 255))); // NOI18N

        jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel5.setText("NO OF TICKETS:");

        jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel6.setText("MOVIE:");

        jLabel11.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel11.setText("FARE:");

        jLabel7.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel7.setText("THEATER:");

        jLabel8.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel8.setText("DATE:");

        jLabel9.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
```

```
NOI18N
        jLabel9.setText("TIME:");

        jLabel10.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel10.setText("jLabel6");

        jLabel12.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel12.setText("jLabel8");

        jLabel13.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel13.setText("jLabel10");

        jLabel14.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel14.setText("jLabel7");

        jLabel15.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel15.setText("jLabel9");

        jLabel16.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel16.setText("jLabel12");

        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
```

```
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(48, 48, 48)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addComponent(jLabel5)
                        .addGap(37, 37, 37)
                        .addComponent(jLabel10)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel6)
                            .addComponent(jLabel11))
                        .addGap(43, 43, 43)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel12)
                            .addComponent(jLabel13))
```

```
                    .addGap(237, 237, 237)))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addComponent(jLabel7)
                    .addGap(44, 44, 44)
                    .addComponent(jLabel14))
                .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                        .addComponent(jLabel8)
                        .addComponent(jLabel9))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(jPanel2Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jLabel16))
                        .addGroup(jPanel2Layout.createSequentialGroup()
                            .addGap(96, 96, 96)
                            .addComponent(jLabel15)))))
            .addContainerGap(188, Short.MAX_VALUE))
        );
        jPanel2Layout.setVerticalGroup(
```

```
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
               .addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel7)
                    .addComponent(jLabel14)))
            .addGroup(jPanel2Layout.createSequentialGroup()
               .addGap(25, 25, 25)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel5)
                    .addComponent(jLabel10))))
          .addGap(29, 29, 29)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel6)
            .addComponent(jLabel12)
            .addComponent(jLabel8)
            .addComponent(jLabel15))
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(26, 26, 26)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel11)
                        .addComponent(jLabel13))
                    .addGap(0, 0, Short.MAX_VALUE))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(34, 34, 34)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel9)
                        .addComponent(jLabel16))
                    .addGap(37, 65, Short.MAX_VALUE))))
        );

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(66, 66, 66)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(43, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap(22, Short.MAX_VALUE)
            .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(40, 40, 40)
            .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 385,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
        );
```

```
        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
        // TODO add your handling code here:
 this.setVisible(false);       new recepit1(a,b,c,e,N1,f).setVisible(true);
    }//GEN-LAST:event_jButton1ActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
```

```
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(payment.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(payment.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(payment.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(payment.class.getName()).log(java.util.logg
ing.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new payment().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
```

```
    private javax.swing.JComboBox<String> jComboBox1;
    private javax.swing.JComboBox<String> jComboBox2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPasswordField jPasswordField1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration//GEN-END:variables
}
```

7)receipt.java

```
public class recepit extends javax.swing.JFrame {

    /**
     * Creates new form recepit
     */
    public recepit() {
        initComponents();
    }
    String a;String b;String c;String e;String f;
            int N1;
    public recepit(String p1,String p2,String p3,String p4,int p5,String p6){
    initComponents();
    this.a=p1;
    this.b=p2;
    this.c=p3;
    this.e=p4;
    this.N1=p5;
    this.f=p6;
    jLabel7.setText(p4);
    jLabel8.setText(p2);
    jLabel9.setText(p1);
    jLabel10.setText(p3);
    jLabel11.setText(String.valueOf(p5));
    jLabel13.setText(p6);
    }
    /**
     * This method is called from within the constructor to initialize the form.
```

```
    * WARNING: Do NOT modify this code. The content of this method is always
    * regenerated by the Form Editor.
    */
   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
   private void initComponents() {

      jLabel3 = new javax.swing.JLabel();
      jPanel1 = new javax.swing.JPanel();
      jLabel2 = new javax.swing.JLabel();
      jLabel4 = new javax.swing.JLabel();
      jLabel5 = new javax.swing.JLabel();
      jLabel6 = new javax.swing.JLabel();
      jLabel7 = new javax.swing.JLabel();
      jLabel12 = new javax.swing.JLabel();
      jLabel13 = new javax.swing.JLabel();
      jLabel8 = new javax.swing.JLabel();
      jLabel9 = new javax.swing.JLabel();
      jLabel10 = new javax.swing.JLabel();
      jLabel11 = new javax.swing.JLabel();
      jLabel1 = new javax.swing.JLabel();
      jButton1 = new javax.swing.JButton();
      jButton2 = new javax.swing.JButton();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

      jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
```

```
"BOOKING DETAILS",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Algerian", 1, 36), new java.awt.Color(0, 0, 204))); // NOI18N

        jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel2.setText("MOVIE:");

        jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel4.setText("THEATRE:");

        jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel5.setText("FARE:");

        jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel6.setText("SHOW:");

        jLabel7.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel12.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel12.setText("DATE:");

        jLabel13.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
```

```
NOI18N

        jLabel8.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel9.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel10.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel11.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel1.setText("NO OF TICKETS:");

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(33, 33, 33)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
```

```
ment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE, 403, Short.MAX_VALUE)
                    .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE, 161,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(271, 271, 271))
            .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addComponent(jLabel1)
                    .addComponent(jLabel12,
javax.swing.GroupLayout.PREFERRED_SIZE, 99,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel6,
javax.swing.GroupLayout.PREFERRED_SIZE, 109,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel5,
javax.swing.GroupLayout.PREFERRED_SIZE, 92,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(110, 110, 110)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)
```

```
                    .addComponent(jLabel9,
javax.swing.GroupLayout.DEFAULT_SIZE, 182, Short.MAX_VALUE)
                    .addComponent(jLabel8,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel7,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel13,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel10,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel11,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
                .addGap(0, 36, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
```

```
ment.LEADING, false)
                .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel7,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGap(36, 36, 36)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.TRAILING)
                .addComponent(jLabel4)
                .addComponent(jLabel8,
javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(34, 34, 34)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)
                .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel9,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGap(24, 24, 24)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
```

```
                .addComponent(jLabel13,
javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel12))
            .addGap(28, 28, 28)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addComponent(jLabel6)
                .addComponent(jLabel10,
javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(30, 30, 30)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addComponent(jLabel5)
                .addComponent(jLabel11,
javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(39, 39, 39))
        );

        jButton1.setFont(new java.awt.Font("Algerian", 1, 24)); // NOI18N
        jButton1.setText("BOOK");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
```

```
        });

        jButton2.setFont(new java.awt.Font("Algerian", 1, 24)); // NOI18N
        jButton2.setText(" CANCEL");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(181, 181, 181)
                .addComponent(jLabel3)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(76, 76, 76)
                        .addComponent(jButton1)
                        .addGap(259, 259, 259)
                        .addComponent(jButton2))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(32, 32, 32)
```

```
                    .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap(196, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(181, 181, 181)
                .addComponent(jLabel3)
                .addContainerGap(386, Short.MAX_VALUE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addComponent(jButton1)
                    .addComponent(jButton2))
                .addGap(16, 16, 16))
        );
```

```
        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton1ActionPerformed
this.setVisible(false);         new cardselection(a,b,c,e,N1,f).setVisible(true); //
TODO add your handling code here:
    }//GEN-LAST:event_jButton1ActionPerformed

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{//GEN-FIRST:event_jButton2ActionPerformed
this.setVisible(false);         new CANCEL().setVisible(true);         // TODO add
your handling code here:
    }//GEN-LAST:event_jButton2ActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
```

```
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(recepit.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(recepit.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(recepit.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(recepit.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
```

```
            new recepit().setVisible(true);
        }
    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
// End of variables declaration//GEN-END:variables


}
```

8)receipt1.java

This Java Swing application creates a GUI for a movie ticket receipt system. It initializes components to display details like the movie, theater, show, fare, date, and ticket count, with constructors handling data binding to respective labels.

```
public class recepit1 extends javax.swing.JFrame {

    /**
     * Creates new form recepit1
     */
    public recepit1() {
        initComponents();
    }
String a;String b;String c;String e;String f;
            int N1;
    public recepit1(String p1,String p2,String p3,String p4,int p5,String p6){
    initComponents();
    this.a=p1;
    this.b=p2;
    this.c=p3;
    this.e=p4;
    this.N1=p5;
    this.f=p6;
    jLabel7.setText(p4);
    jLabel8.setText(p2);
    jLabel9.setText(p1);
    jLabel10.setText(p3);
    jLabel11.setText(String.valueOf(p5));
```

```
    jLabel13.setText(p6);
    }

    recepit1(String a, String b, String c, String d, String e, int N1) {
        throw new UnsupportedOperationException("Not supported yet."); //To
change body of generated methods, choose Tools | Templates.
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel12 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
```

```
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null,
"RECEPIT", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Algerian", 1, 36), new java.awt.Color(51, 0, 204))); // NOI18N

        jLabel4.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel4.setText("THEATRE:");

        jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel5.setText("FARE:");

        jLabel6.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel6.setText("SHOW:");

        jLabel12.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel12.setText("DATE:");

        jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
```

```
NOI18N
        jLabel2.setText("MOVIE:");

        jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel1.setText("NO OF TICKETS:");

        jLabel7.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel13.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel8.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel9.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel10.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel11.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N

        jLabel3.setFont(new java.awt.Font("Algerian", 1, 36)); // NOI18N
        jLabel3.setForeground(new java.awt.Color(0, 255, 0));
        jLabel3.setText("THANKS FOR BOOKING");
```

```
        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                            .addComponent(jLabel1)
                            .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE, 158,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 123,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel12,
javax.swing.GroupLayout.PREFERRED_SIZE, 108,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel5,
javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                    .addComponent(jLabel6,
javax.swing.GroupLayout.PREFERRED_SIZE, 108,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(219, 219, 219)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING, false)
                    .addComponent(jLabel8,
javax.swing.GroupLayout.DEFAULT_SIZE, 181, Short.MAX_VALUE)
                    .addComponent(jLabel7,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel13,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel10,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel9,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel11,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(164, 164, 164)
                .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE, 426,
javax.swing.GroupLayout.PREFERRED_SIZE)))
```

```
            .addContainerGap(298, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEAD
ING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(28, 28, 28)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addComponent(jLabel1)
                    .addComponent(jLabel7,
javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(41, 41, 41)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel4,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addGap(34, 34, 34))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel8,
javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addComponent(jLabel2)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(6, 6, 6)
                    .addComponent(jLabel13,
javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(32, 32, 32)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel12,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addGap(31, 31, 31))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel10,
javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(46, 46, 46)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addComponent(jLabel6)
```

```
                .addComponent(jLabel9,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(54, 54, 54)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
                .addComponent(jLabel5)
                .addComponent(jLabel11,
javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(4, 4, 4)
            .addComponent(jLabel3))
        );

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
            .addGap(232, 232, 232)
            .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(201, Short.MAX_VALUE))
```

```
        );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(91, 91, 91))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the
default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
```

```
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(recepit1.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(recepit1.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(recepit1.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(recepit1.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
```

```
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new recepit1().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    // End of variables declaration//GEN-END:variables

    recepit1 getVisible(boolean b) {
        throw new UnsupportedOperationException("Not supported yet."); //To
change body of generated methods, choose Tools | Templates.
    }
}
```

9)register.java

This Java Swing program provides a registration GUI for users to input their name, email, password, and phone number. Upon clicking "SIGN UP," the data is validated and stored in a MySQL database, with error handling for database operations**.**

```
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class register extends javax.swing.JFrame {

    /**
     * Creates new form register
     */
    public register() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jTextField3 = new javax.swing.JTextField();
```

```
        jButton1 = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jTextField4 = new javax.swing.JTextField();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setPreferredSize(new java.awt.Dimension(1080, 754));
        getContentPane().setLayout(null);

        jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel1.setText("NAME");
        getContentPane().add(jLabel1);
        jLabel1.setBounds(135, 128, 110, 28);

        jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel2.setText("EMAIL ID");
        getContentPane().add(jLabel2);
        jLabel2.setBounds(135, 184, 140, 28);

        jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel3.setText("PASSWORD");
        getContentPane().add(jLabel3);
        jLabel3.setBounds(135, 309, 150, 28);

        jTextField1.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
```

```
NOI18N
        getContentPane().add(jTextField1);
        jTextField1.setBounds(343, 129, 190, 34);

        jTextField2.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        getContentPane().add(jTextField2);
        jTextField2.setBounds(343, 185, 190, 30);

        jTextField3.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        getContentPane().add(jTextField3);
        jTextField3.setBounds(343, 310, 190, 30);

        jButton1.setFont(new java.awt.Font("Algerian", 1, 24)); // NOI18N
        jButton1.setText("SIGN UP");
        jButton1.addActionListener(evt -> jButton1ActionPerformed(evt));
        getContentPane().add(jButton1);
        jButton1.setBounds(200, 380, 220, 41);

        jLabel4.setFont(new java.awt.Font("Algerian", 1, 36)); // NOI18N
        jLabel4.setText("NEW REGISTER");
        getContentPane().add(jLabel4);
        jLabel4.setBounds(160, 30, 300, 47);

        jLabel5.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        jLabel5.setText("PHONE NUMBER");
        getContentPane().add(jLabel5);
```

```
        jLabel5.setBounds(135, 250, 200, 28);

        jTextField4.setFont(new java.awt.Font("Times New Roman", 1, 24)); //
NOI18N
        getContentPane().add(jTextField4);
        jTextField4.setBounds(343, 245, 190, 30);

        pack();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String name = jTextField1.getText();
        String email = jTextField2.getText();
        String password = jTextField3.getText();
        String phone = jTextField4.getText();

        // Validate input
        if (name.isEmpty() || email.isEmpty() || password.isEmpty() ||
phone.isEmpty()) {
            JOptionPane.showMessageDialog(this, "Please fill in all fields.");
            return;
        }

        try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost/UserRegistrationDB",
"root", "Janet_88");
             PreparedStatement pstmt = conn.prepareStatement("INSERT INTO
register (name, email, password, phone) VALUES (?, ?, ?, ?)")) {
```

```
            pstmt.setString(1, name);
            pstmt.setString(2, email);
            pstmt.setString(3, password);
            pstmt.setString(4, phone);
            pstmt.executeUpdate();

            JOptionPane.showMessageDialog(this, "Registration successful!");
            this.setVisible(false);
            new login().setVisible(true);

        } catch (SQLException e) {
            Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, e);
            JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());
        }
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(() -> new register().setVisible(true));
    }

    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
```

```
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
```

4.2. Tools and Technologies Used

- **Java:** Core programming language used for system development. Its object-oriented design ensures robust and flexible implementation.
- **MySQL:** Relational database management system for storing movie schedules, user data, bookings, and payment records.
- **Java Swing:** A GUI toolkit used to create an interactive user interface with buttons, labels, forms, and tables.
- **JDBC (Java Database Connectivity):** Java API used to connect the application to the MySQL database and execute SQL queries.
- **NetBeans/IntelliJ IDEA:** Integrated Development Environments (IDEs) used for coding, debugging, and running the system.

4.3. System Architecture

The Movie Ticket Booking System follows a **Client-Server Architecture**:

- **Client (Java Swing Application):** The client-side consists of a user interface where users can browse movies, select showtimes, choose seats,

and confirm bookings. The client communicates with the database through **JDBC** to fetch or update data.

- **Server (Backend/Database):** The server-side handles user requests, manages business logic (e.g., seat availability checks), and interacts with the database to store user and booking data.

The system uses a **Request-Response Model**, where the client sends requests (e.g., fetch movie schedules, reserve seats), and the server processes them and sends responses.

4.4. Database Implementation

The system uses **MySQL** to store essential data, such as movie schedules, user accounts, seat availability, and booking records.

Tables in the Database:

- **Users Table:** Stores user information, including name, email, and encrypted passwords.
- **Movies Table:** Stores movie details, including title, genre, duration, language, and ratings.
- **Showtimes Table:** Contains information about showtimes, theaters, and available seats for each movie.
- **Bookings Table:** Tracks all bookings made by users, including showtime, seat numbers, and payment status.
- **Transactions Table:** Logs all payment transactions, including amounts, timestamps, and status.

Sample SQL Queries:

- **Create User:**
- INSERT INTO users (name, email, password) VALUES ('John Doe', 'john@example.com', 'hashed_password');
- **Add Movie Showtime:**
- INSERT INTO showtimes (movie_id, theater, time, available_seats) VALUES (1, 'Theater 1', '2024-11-21 18:30:00', 100);
- **Reserve Seats:**
- UPDATE showtimes SET available_seats = available_seats - 2 WHERE showtime_id = 5 AND available_seats >= 2;
- **Fetch Booking History:**
- SELECT * FROM bookings WHERE user_id = 1 ORDER BY booking_date DESC;

4.5. Key Modules and Their Implementation

4.5.1. User Authentication (Login)

The login system validates user credentials against the **Users** table. Successful login redirects the user to the dashboard.

Steps for Implementation:

1. User inputs email and password.
2. System checks credentials against the **Users** table using a SELECT query.
3. If valid, the user is authenticated and granted access.
4. If invalid, an error message is displayed.

4.5.2. Movie and Showtime Management

Users can browse available movies and their showtimes.

Steps for Implementation:

1. Fetch movie details using a SELECT query from the **Movies** and **Showtimes** tables.
2. Display the movie list and showtimes on the GUI.
3. Users can filter movies by genre, language, or rating.

4.5.3. Seat Selection

Users select available seats for their chosen movie and showtime.

Steps for Implementation:

1. Fetch the seating layout and availability using a SELECT query.
2. Display an interactive seating chart where users can select seats.
3. Reserve selected seats temporarily during the booking process.

4.5.4. Ticket Booking

The booking module finalizes the seat reservation and processes the payment.

Steps for Implementation:

1. Confirm selected seats and calculate total cost.
2. Update seat availability using an UPDATE query.
3. Insert booking details into the **Bookings** table.
4. Log the transaction in the **Transactions** table.

4.5.5. Payment Processing

The payment system validates user payment details and confirms the transaction.

Steps for Implementation:

1. Integrate with a payment gateway to process payments securely.
2. Log payment details in the **Transactions** table.
3. Confirm the booking upon successful payment.

4.5.6. Booking History

The system displays past and upcoming bookings.

Steps for Implementation:

1. Fetch booking records from the **Bookings** table.
2. Display booking details such as movie name, showtime, and seat numbers.

4.5.7. Notifications

Users receive notifications for successful bookings and upcoming showtimes.

Steps for Implementation:

1. Send email or SMS notifications after confirming a booking.
2. Notify users of any showtime changes or cancellations.

4.6. Error Handling

Proper error handling ensures system stability:

- Use try-catch blocks to handle exceptions (e.g., database connection issues, invalid inputs).
- Display user-friendly error messages (e.g., "Seats no longer available").

4.7. Security Measures

- **Encryption:** Store passwords and payment details in encrypted form to prevent unauthorized access.
- **Session Management:** Create and terminate sessions securely for logged-in users.
- **SQL Injection Prevention:** Use parameterized queries and prepared statements.

4.8. Testing

The system undergoes extensive testing, including:

- **Unit Testing:** Validate individual modules (e.g., login, booking).
- **Integration Testing:** Ensure smooth interaction between modules (e.g., seat selection and payment).
- **User Acceptance Testing (UAT):** Verify the system meets user requirements and expectations.

By following these steps, the **Movie Ticket Booking System** is implemented as a reliable, secure, and user-friendly platform for ticket booking.

CHAPTER 6: RESULTS AND DISCUSSION

This section presents the results obtained after the implementation of the Online Movie Ticket Booking System and discusses its effectiveness in terms of performance, usability, and impact on the ticket booking process.

6.1. System Performance

The Online Movie Ticket Booking System has been tested under various scenarios, and the results show that it performs efficiently even with a large number of users and transactions. The system handles operations like searching for movies, booking tickets, and processing payments in a timely manner, ensuring a smooth user experience.

- Login and Authentication:

The login process operates seamlessly, allowing users (customers/admins) to access their respective dashboards quickly after entering valid credentials. The system employs secure password handling and encryption techniques to ensure protection against unauthorized access.

- Movie Search and Selection:

Users can quickly search for movies based on title, genre, or location, and the system provides relevant results instantly. Even with a large database of movies and theaters, search operations execute efficiently.

- Booking and Payment:

Ticket booking and payment processes are completed within a few steps, ensuring quick transactions. The system supports multiple payment gateways

and processes payments securely. Real-time updates on seat availability prevent overbooking.

- Database Operations:

SQL queries for retrieving movie data, managing seat bookings, and generating receipts are optimized to handle high user loads. The database structure minimizes redundancy, ensuring fast access and retrieval of information. The system also supports multiple concurrent users without performance issues.

6.2. User Interface and Usability

The system's front-end, developed using HTML, CSS, and JavaScript, is designed to be intuitive and visually appealing. It ensures ease of use for both first-time users and regular customers.

- User Dashboard:

Customers can easily browse through movies, view show timings, and book tickets. The interface provides clear instructions for each step, making the ticket booking process hassle-free.

- Admin Dashboard:

Admins can manage movie schedules, update ticket availability, and generate sales reports with minimal effort. The dashboard design simplifies administrative tasks and reduces complexity.

- Mobile-Friendly Design:

The system is responsive and accessible on various devices, including smartphones and tablets, allowing users to book tickets conveniently on the go.

- Reports and Data Analysis:

The system offers detailed reports on sales, user activity, and booking trends. Admins can generate monthly or weekly reports to analyze revenue and monitor user engagement.

6.3. Impact on Ticket Booking Process

The implementation of this system has significantly improved the efficiency and reliability of the movie ticket booking process. Traditional methods like in-person bookings or over-the-phone reservations were often time-consuming and prone to errors. The automated system ensures accurate and fast transactions, enhancing the overall experience for both customers and administrators.

- Efficiency:

By automating the booking process, the system saves time for both customers and theater administrators. Customers can book tickets within minutes, while admins benefit from automated updates on bookings and schedules.

- Accuracy:

The system eliminates issues such as double-booking, incorrect seat assignments, or mismanaged payments. Real-time updates on seat availability ensure error-free transactions.

- Transparency:

Customers can view seat layouts, ticket prices, and booking statuses in real-time, providing transparency and fostering trust. Admins also have clear insights into revenue and user behavior.

The Online Movie Ticket Booking System has successfully streamlined the ticket booking process, improved customer satisfaction, and reduced administrative burdens, making it a valuable tool for modern entertainment venues.

CHAPTER 7: CONCLUSION

The Online Movie Ticket Booking System is a comprehensive solution designed to streamline the process of booking movie tickets in modern theaters and multiplexes. By integrating advanced web technologies such as PHP, MySQL, and JavaScript, the system provides an efficient and user-friendly platform for both customers and administrators. The core objective of the system—automating and simplifying the ticket booking process—has been successfully achieved.

The system enables users to book tickets with ease and generate detailed receipts, significantly reducing manual effort. Customers benefit from real-time access to seat availability and booking confirmation, fostering greater convenience and transparency. The database backend ensures accurate and consistent record-keeping, while the front-end interface offers a smooth and intuitive experience.

One of the primary strengths of the system is its ability to handle bookings for multiple movies and showtimes with minimal effort. Customers can search for movies, select seats, and complete payments quickly, while the system automatically updates seat availability to prevent overbooking. Additionally, features such as bulk ticket booking and real-time notifications enhance the system's effectiveness in managing reservations across multiple theaters.

Despite the system's many advantages, there are some challenges, such as its reliance on internet connectivity and the need for user training, particularly for customers unfamiliar with online platforms. However, these challenges can be

addressed through improvements in user education, mobile application integration, and enhanced security features in future updates.

The system's implementation marks a significant step toward improving the efficiency and reliability of the ticket booking process. By automating tasks like seat selection, payment processing, and booking confirmation, the system allows administrators to focus on managing theater operations more effectively, ultimately contributing to a smoother and more satisfying customer experience.

CHAPTER 8: REFERENCES

This section lists the sources consulted and referenced during the development of the Online Movie Ticket Booking System using Java and MySQL. These resources provided essential insights into Java programming, database management, and application development.

1. Oracle Documentation. (n.d.). Java SE Documentation. Retrieved from https://docs.oracle.com/javase/

Comprehensive official documentation for the Java Standard Edition, covering core concepts and APIs used in the system development.

2. MySQL Documentation. (n.d.). MySQL Database Documentation. Retrieved from https://dev.mysql.com/doc/

Detailed official documentation for managing and interacting with the MySQL database, which supported the system's backend.

3. W3Schools. (n.d.). Java Tutorial. Retrieved from https://www.w3schools.com/java/

A beginner-friendly guide to Java programming concepts and syntax.

4. GeeksforGeeks. (n.d.). Java Programming. Retrieved from https://www.geeksforgeeks.org/java/

A useful resource for understanding object-oriented programming, JDBC, and advanced Java features applied in the system.

5. Baeldung. (n.d.). Introduction to JDBC. Retrieved from https://www.baeldung.com/java-jdbc

An essential resource for working with JDBC to establish a connection between Java and MySQL databases.

6. Deitel, P., & Deitel, H. (2017). Java: How to Program (11th Edition). Pearson.

This book provided a detailed understanding of Java concepts, data structures, and best practices for application development.

7. Stack Overflow. (n.d.). Java and MySQL Discussions and Solutions. Retrieved from https://stackoverflow.com/

A valuable platform for troubleshooting Java coding issues and optimizing database queries.

8. YouTube. (2021). Building a Java Application with MySQL. Retrieved from https://www.youtube.com/

Video tutorials offering step-by-step instructions for integrating Java with MySQL, designing the database schema, and implementing CRUD operations.

9. GitHub. (n.d.). Open-Source Java Projects with MySQL Integration. Retrieved from https://github.com/

Reviewed open-source projects to understand implementation strategies for features such as booking workflows and database operations.

10. MySQL Workbench Manual. (n.d.). Retrieved from https://dev.mysql.com/doc/workbench/en/

Used as a guide for designing the database schema, creating tables, and writing optimized queries.

RESEARCH PAPERS AND GITHUB LINKS

Research Papers:

1. http://i-rep.emu.edu.tr:8080/xmlui/handle/11129/5139

Discusses database management and application design techniques.

2. https://air.ashesi.edu.gh/items/8dbf3c7e-813f-44f9-85b7-5a4a9b575c6b

Reviews ticket booking systems and their features.

GitHub Links:

1. https://github.com/Madhu20053/DBMS.git

2. https://github.com/Kanishka-0001/MINI-PROJECT.git

3. https://github.com/LovishaaKarol/DBMS-project.git