



CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE • INDIA

Java Project

On

Online Movie Ticket Booking

Submitted by:

E.C. Greeshma [2347115]

Keerthana.c [2347124]

Visesh Agarwal [2347164]

2 MCA A
CHRIST UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Introduction

The online movie ticket booking system developed using Java, JavaScript, MySQL and NetBeans. The system allows customers to conveniently book movie tickets online by selecting a movie, show timing, theater and number of tickets.

System Analysis and Design

The key requirements gathered for the system are:

- Home page for search, login and exit.
- User login for registered customers
- Browsing movies and show timings
- Selecting theater location
- Booking desired number of tickets
- Payment gateway integration
- Generating ticket confirmation

The system was designed with a MySQL database backend to store customer, movie scheduling and ticket details. The front-end UI was implemented in Java.

Key design aspects:

- ER diagram for database schema design
- Class diagrams for entities like Customer, Payment, Booking, Theater etc.
- Flowcharts for booking workflow and payment integration
- Wireframes and mockups for web page design

Implementation

The system was implemented using Java and utilized the following core concepts:

- Java Beans - Used for entities like Customer, Movie, Payment etc. Getter/setter methods provided encapsulation.
- Java Swing - Used for creating UI components like buttons, text fields, menu etc. Provided reusable UI elements.
- Multithreading - Used for asynchronous tasks like payment integration to improve responsiveness.
- Packages - Classes organized into logical packages like utils, ui, models etc. for modularity.

- Collections – collections concept used for names in combo box.
- Inheritance - Booking and Payment classes inherited from parent Ticket class to reduce duplicate code.
- OOP - Followed principles like abstraction, encapsulation and polymorphism for reusable code.
- Buffer Class - Used StringBuffer for building dynamic strings efficiently in booking logic.

Key classes/files developed:

- Home.java for Home page
- Customer.java for customer entity
- Login.java for handling user authentication
- Booking.java manages ticket booking logic
- Payment.java integrates payment gateway APIs
- register.java new user to register
- receipt.java conformation of booking details

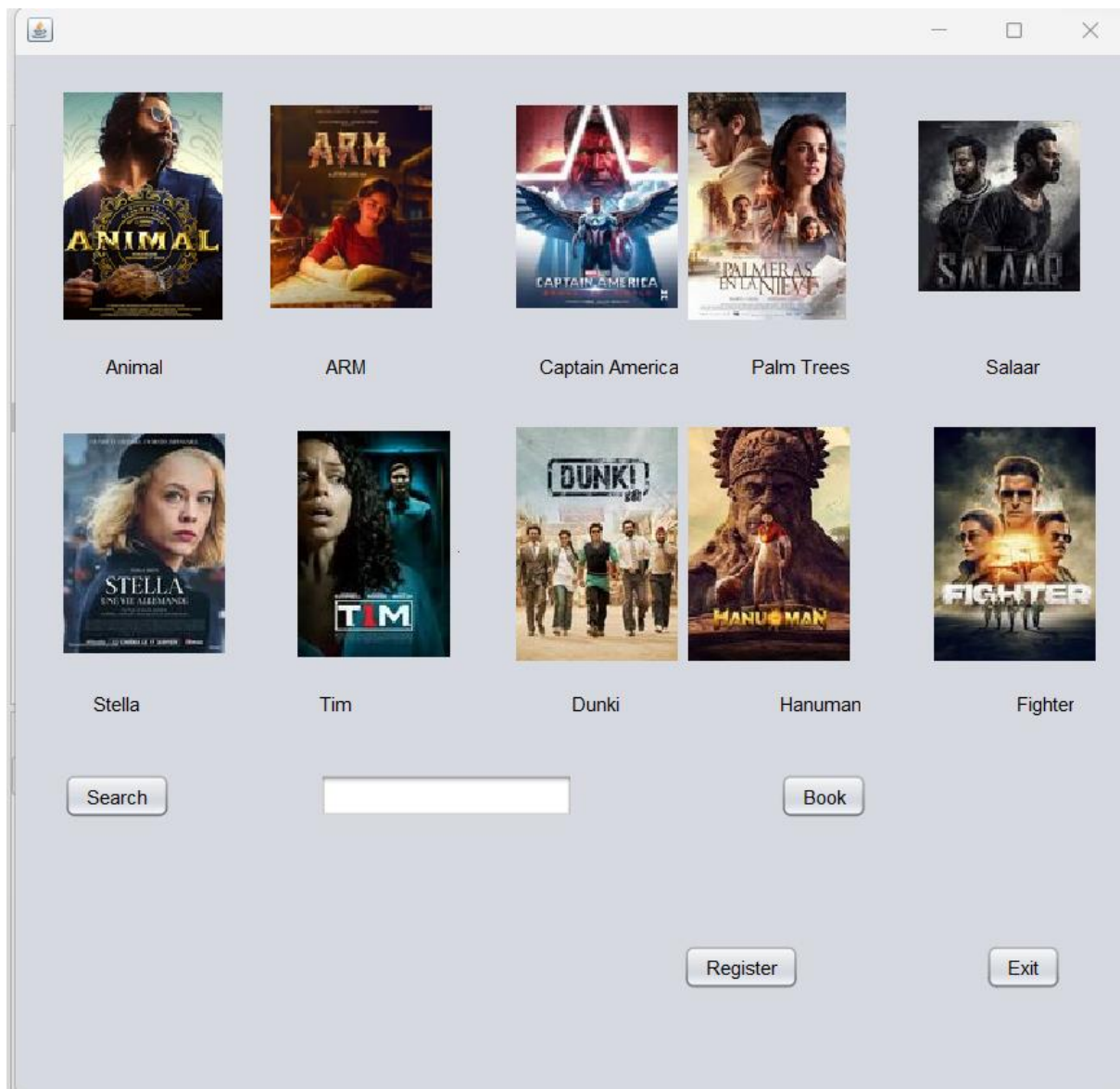
Testing and Results

The system was tested with different use cases covering:

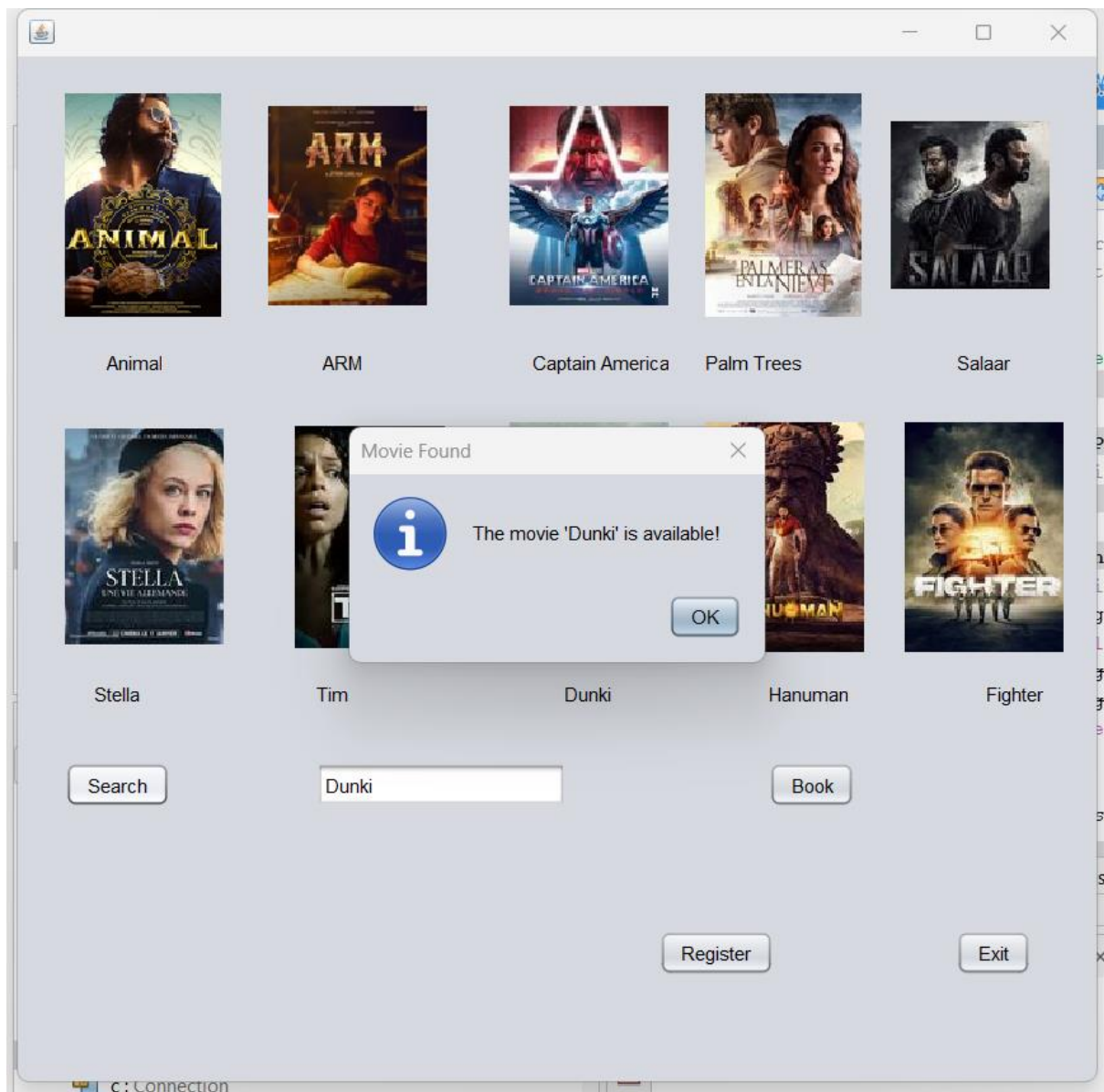
- Home page user can login and search for particular movie
- Login with valid and invalid credentials
- Browsing movie catalog and theater lists
- Booking tickets for different customers
- Cancelling selected tickets

Sample screenshots are attached showing the key flows.

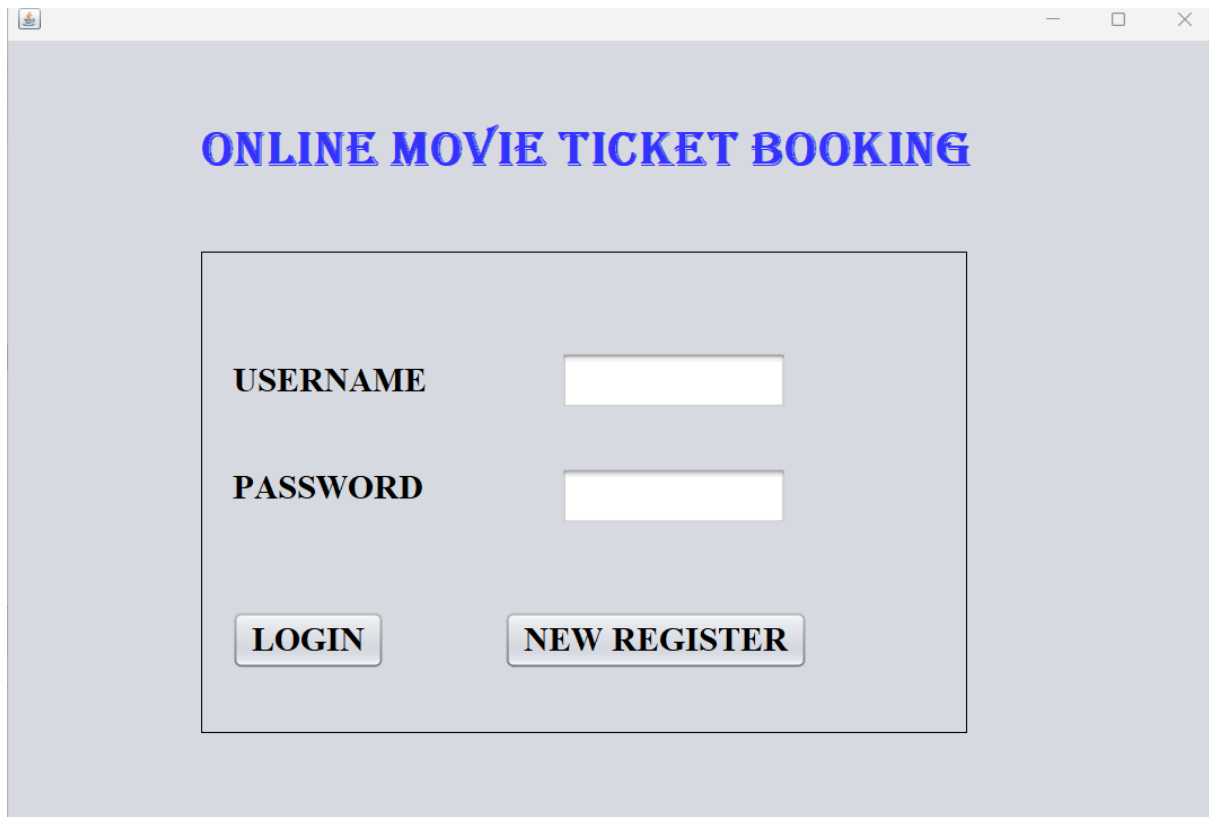
Home Page:



Searching a movie in Home page:



Login page:



The image shows a web browser window with a light gray background. At the top, the title bar contains a small icon on the left and standard window controls (minimize, maximize, close) on the right. The main content area has a light gray background. In the center, the text **ONLINE MOVIE TICKET BOOKING** is displayed in a large, bold, blue, serif font. Below this title, there is a white rectangular box with a thin black border. Inside this box, the labels **USERNAME** and **PASSWORD** are positioned to the left of two white input fields. Below the input fields, there are two buttons: **LOGIN** on the left and **NEW REGISTER** on the right. Both buttons have a light gray background and a thin black border.

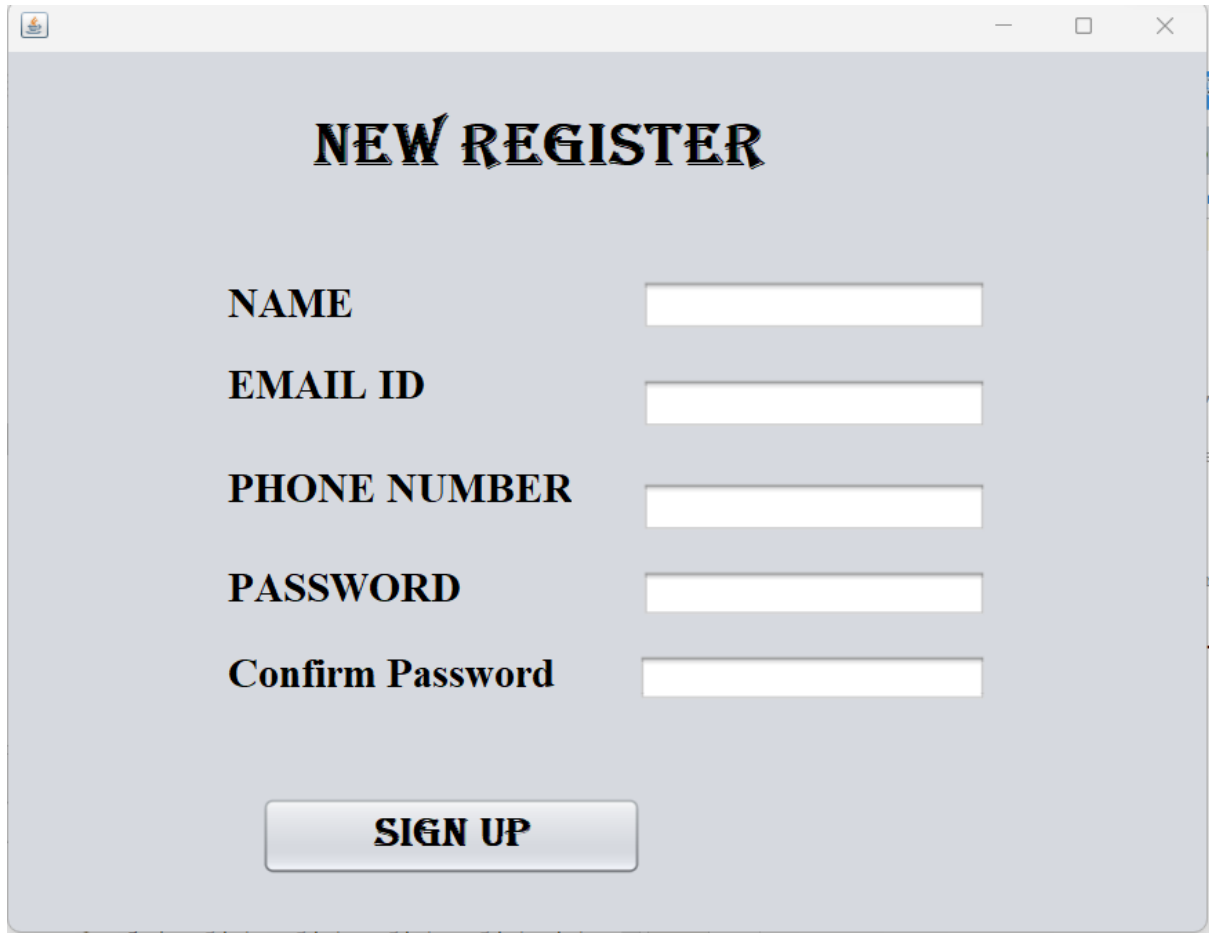
ONLINE MOVIE TICKET BOOKING

USERNAME

PASSWORD

LOGIN **NEW REGISTER**

New Register:



A screenshot of a web browser window titled "NEW REGISTER". The window has a light gray background and a standard browser title bar with minimize, maximize, and close buttons. The title "NEW REGISTER" is centered at the top in a large, bold, black serif font. Below the title, there are five input fields arranged vertically, each preceded by a label in a bold, black serif font: "NAME", "EMAIL ID", "PHONE NUMBER", "PASSWORD", and "Confirm Password". Each label is aligned to the left of its corresponding white input box. At the bottom center of the form, there is a rectangular button with a light gray gradient and the text "SIGN UP" in a bold, black serif font.

| | |
|-------------------------|--------------------------|
| NAME | <input type="text"/> |
| EMAIL ID | <input type="text"/> |
| PHONE NUMBER | <input type="text"/> |
| PASSWORD | <input type="password"/> |
| Confirm Password | <input type="password"/> |

SIGN UP

Booking details:

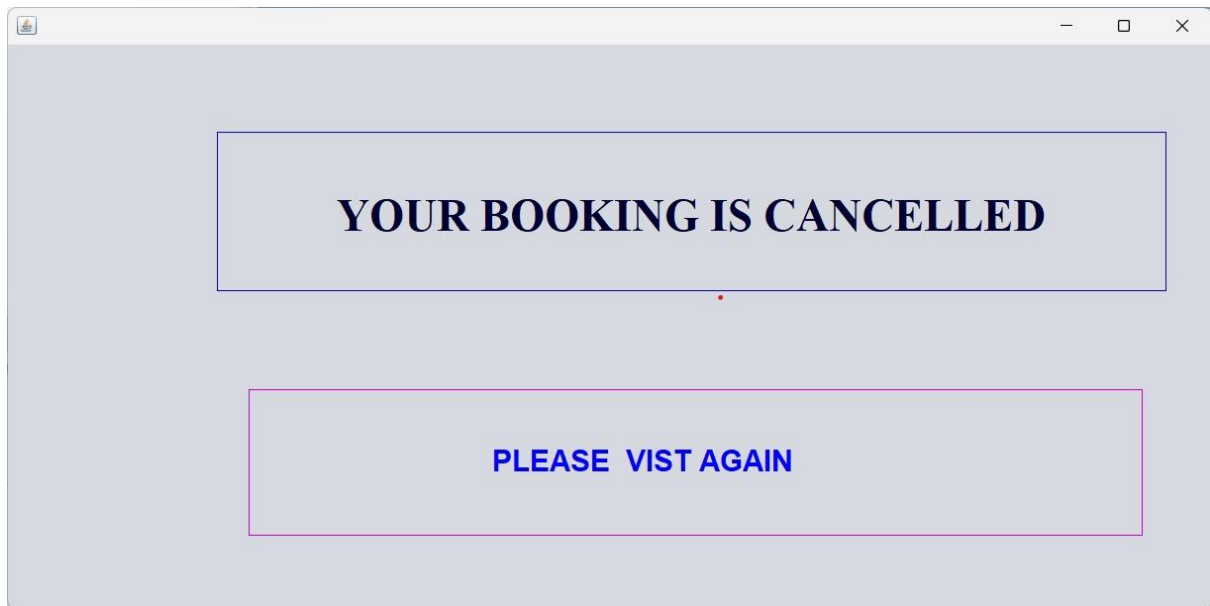


A screenshot of a web browser window titled "BOOKING DETAILS". The window has a light gray background and a standard browser title bar. The title "BOOKING DETAILS" is centered at the top in a bold, blue serif font. Below the title, there is a large, light gray rectangular box containing booking information. Inside this box, the labels and values are arranged in two columns, with labels on the left and values on the right, all in a bold, black serif font. The labels are "NO OF TICKETS:", "THEATRE:", "MOVIE:", "DATE:", "SHOW:", and "FARE:". The corresponding values are "5", "INOX", "Animal", "24-01-2024", "10:00 AM", and "1100". Below this box, at the bottom of the window, there are two buttons: "Book" on the left and "CANCEL" on the right. Both buttons have a light gray gradient and are in a bold, black serif font.

| | |
|-----------------------|-------------------|
| NO OF TICKETS: | 5 |
| THEATRE: | INOX |
| MOVIE: | Animal |
| DATE: | 24-01-2024 |
| SHOW: | 10:00 AM |
| FARE: | 1100 |

Book **CANCEL**

Cancellation page:



Movie details:

A screenshot of a web application window with a light gray background. At the top, there is a teal-colored header with the text "MOVIE DETAILS" in bold, teal, uppercase letters. Below the header, there is a white rectangular box with a thin gray border containing the booking details. The details are as follows: "SELECT MOVIE:" followed by a dropdown menu showing "Animal"; "SELECT THEATRE:" followed by a dropdown menu showing "INOX"; "SELECT DATE:" followed by a dropdown menu showing "24-01-2024"; "SELECT TIME:" followed by a dropdown menu showing "10:00 AM"; and "NO OF TICKETS:" followed by a text input field containing the number "5". At the bottom of the white box, there is a blue rectangular button with the text "SUBMIT" in bold, white, uppercase letters. The window has a standard title bar with a minimize icon, a maximize icon, and a close icon.

Conform details:

CONFIRM DETAILS

NO OF TICKETS: 6

THEATER: INOX

MOVIE: Animal

DATE: 24-01-2024

FARE: 1320

TIME: 10:00 AM

PAYMENT TYPE

☐ Card Payment

☐ Internet Banking

☐ UPI

Make Payment

Go Back

Conform booking:

CONFORM BOOKING

NO OF TICKETS: 6

THEATER: INOX

MOVIE: Animal

DATE: 24-01-2024

FARE: 1320

TIME: 10:00 AM

CARD DETAILS

Proceed

Card number

8742841296321467

Card holder name

Visesh Agarwal

CVV **

EXP 01 20...

Payment page:

Select your Bank

ICICI Bank

Username

viseshagarwal

Password

Make Payment

Input

?

Enter UPI ID:

visesh@upi

Cancel

OK

Validation Result

i

Payment request sent!

OK

Input [X]

Enter UPI ID:

visesh

Cancel OK

Validation Result [X]

Invalid UPI ID!

OK

Validation:

NEW REGISTER

NAME

EMAIL

PHONE

PASSWORD

Confirm Password

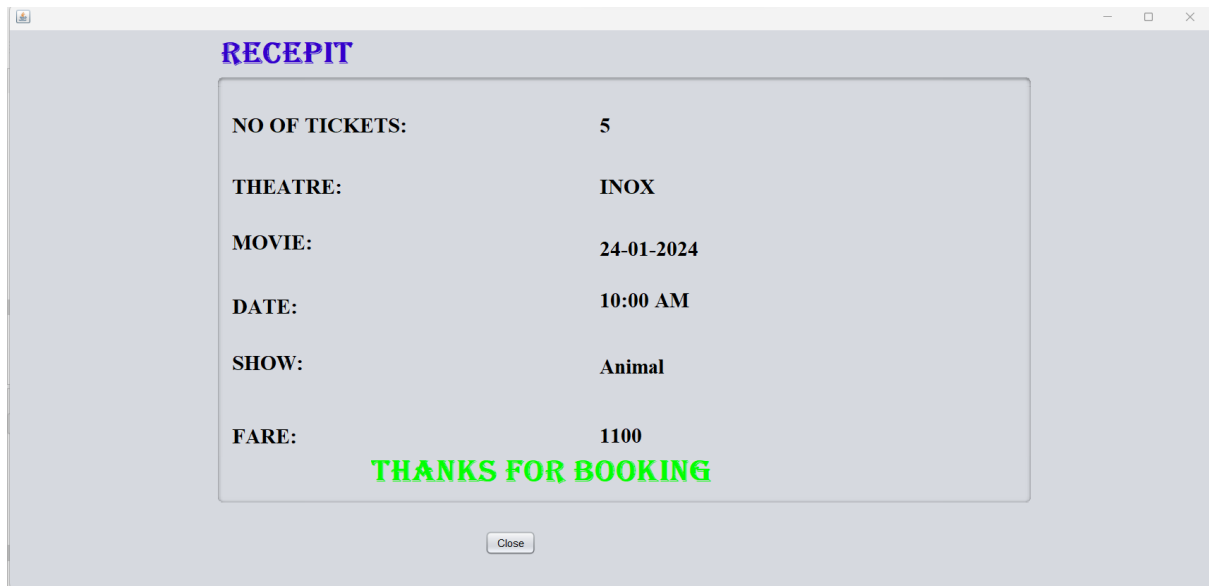
SIGN UP

Error [X]

Please fill in all fields with valid information.

OK

Receipt page:



Performance was smooth for concurrent bookings with no database locking issues. The system meets the primary booking functionalities required.

Limitation:

For enhanced user experience, further improvements like showing seat availability, booking history and live cancellation can be added.

The payment gateway can also be connected to banks for actual payments.

Conclusion:

The online movie ticket booking system was implemented successfully in Java using core Java concepts like multithreading, swing, collection, inheritance, OOP design and MySQL database, meeting the requirements for booking tickets conveniently via a web interface. The project provided good opportunity to learn web application development concepts.