Faculty of Computing (23.1)

PUSL 2024 Software Engineering 2 (24/AU/M)

Coursework Report

ABC Cinema

Group G

	Student ID (Plymouth)	Student Name (As appeared on DLE)
1	10953243	Bedde Samarasinghe (Leader)
2	10952466	Hewa Gamage
3	10953326	Yuhas Gamage
4	10952509	Amarasinghe Senevirathne
5	10903095	Liyana Gunawardana
6	10899703	Dasith Silva
7	10952758	Hiddiyadura Abeynayake
8	10820754	Herath Vimalaranga

Table of Content

Introduction

As our coursework has mentioned we have created a web-application for ‘ABC Cinema’ using Java Servlets, JSP, HTML, CSS, JavaScripts, and MYSQL Server database. In this web-application we have upgraded ‘ABC Cinema’s obsolete website to the revolutionised web-application including : Online Ticket reservation of tickets avoiding the booking collisions, Online payment handling, Visualisation of the customer feedbacks, Email payment notification (payment confirmation) service, with interactive modern user interface. We have started developing, after the designing of all the UML diagrams (Use Case diagram, Class diagram, Sequence diagram, and the ER diagram) and in below all the diagrams are explained: how this web-application’s system is designed.

Identified Processes in ABC Cinema

Customer Processes :

- **Browsing movies and showtimes.**

By clicking ‘Movies’ in navigation bar, users can browse all the available movies, and upcoming movies.

- **Creating and managing user accounts.**

By clicking ‘Login’ button in navigation bar, users can login if they have existing account in ‘ABC Cinema’ or if a new user comes to the login page, they can click ‘Register here’ button to register to the ‘ABC Cinema’. After the registration user will redirected to the ‘Login’ page.

- **Booking seats and paying for tickets.**

Users can book tickets after the selection of a movie from ‘Movies’ page. Users can select show date, show time from available times, and Number of tickets. Then clicking ‘Continue to Seat Selection’ button, will navigate the user into the seat structure map to selecta seat from available seats. By clicking ‘Proceed to Payment’ button, user will redirect to payment completion page and user can add their Visa, Master or any Credit/Debit card to Complete the payment. Then, the payment receipt will received to their registered Email address.

- **Posting feedbacks.**

By clicking, ‘Feedbacks’ in navigation bar, Users can add a feedback with their rating.

- **Viewing booked tickets.**

Users can view their booked tickets by clicking ‘My Bookings’ in navigation bar.

Admin / Manager Processes :

- **Adding / Editing and Deleting movie details.**

For a admin, We have developed an admin panel to add movies, edit movie details, and delete movies from the movie list. Those admin functions are updating real-time database’s ‘Movies’ table.

- **Ticket Management.**

In ticket management, admins can click ‘Add New Tickets’ button to add tickets to a movie declaring price, Quantity, and Status of the availability. An admins can use ‘Edit Ticket’ icon in Actions to edit the ticket details in a movie. As well as, admins can delete tickets using ‘Delete icon’ in ‘Actions’.

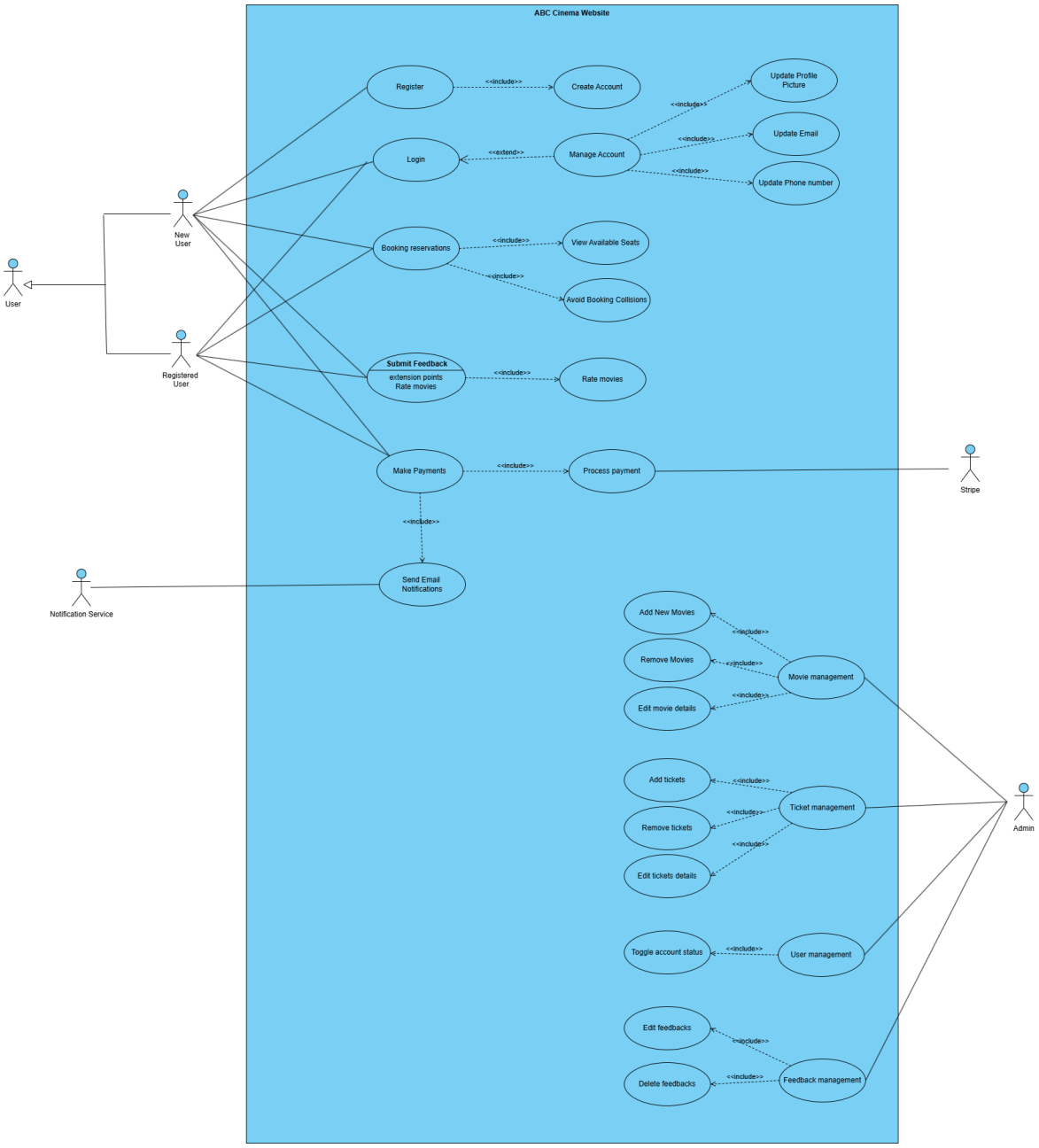
- **User Management**

In user management, admins can view all the registered users details except their passwords. An admin can change the user profiles ‘Status’ by clicking power button in ‘Actions’ column. When the user profile in ‘Inactive’ users cannot login to the system using their credentials until an admin change their status.

- **Feedback Management**

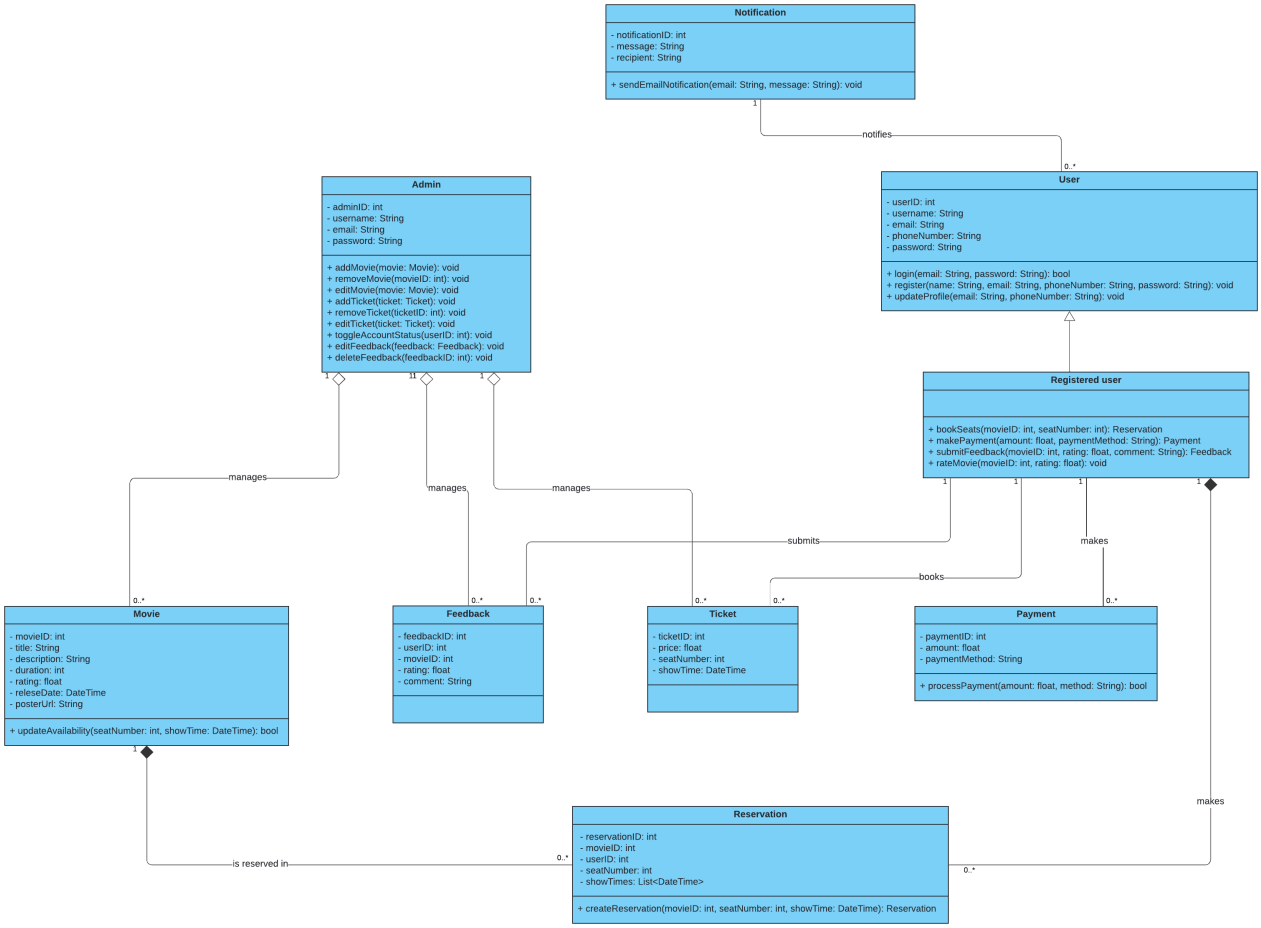
Admins can view all the user submitted feedbacks in ‘Feedback Management’ option and if there is a feedback that the ‘ABC Cinema’ want to delete or edit, we have added ‘Edit’ and ‘Delete’ option to the feedbacks.

Use Case Diagram



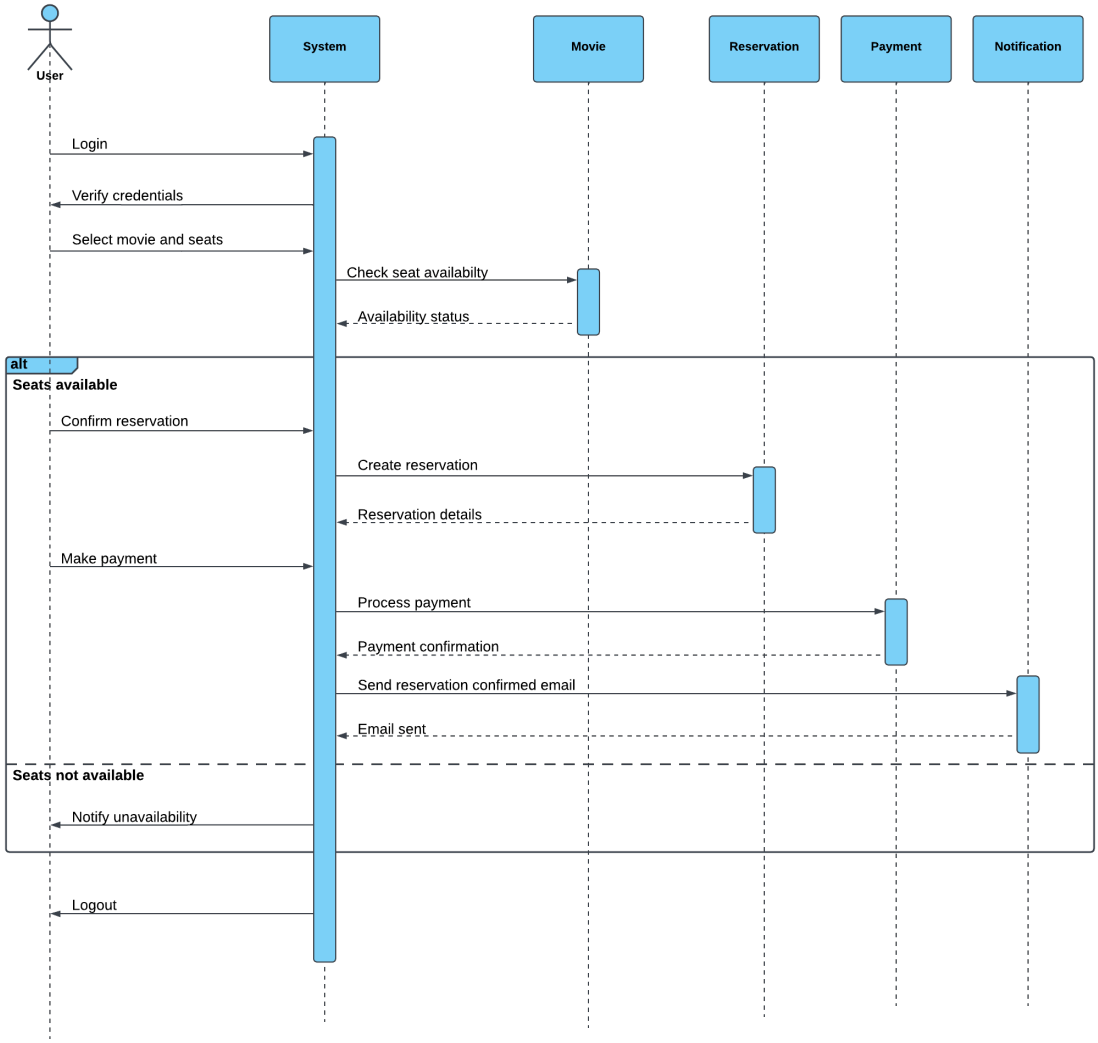
The Use Case Diagram represents core functionalities in our system. There are 4 actors. The admin manages core operations such as movies, theatres, showtimes, ticket reservations, and user feedback. Registered Users can browse movies, book tickets, make payments, provide feedback, and manage their profiles, while Guest Users may have limited access, such as browsing movies and showtimes. Notification service handle the sending notification parts and in our case Stripe is working as the payment gateway.

Class Diagram



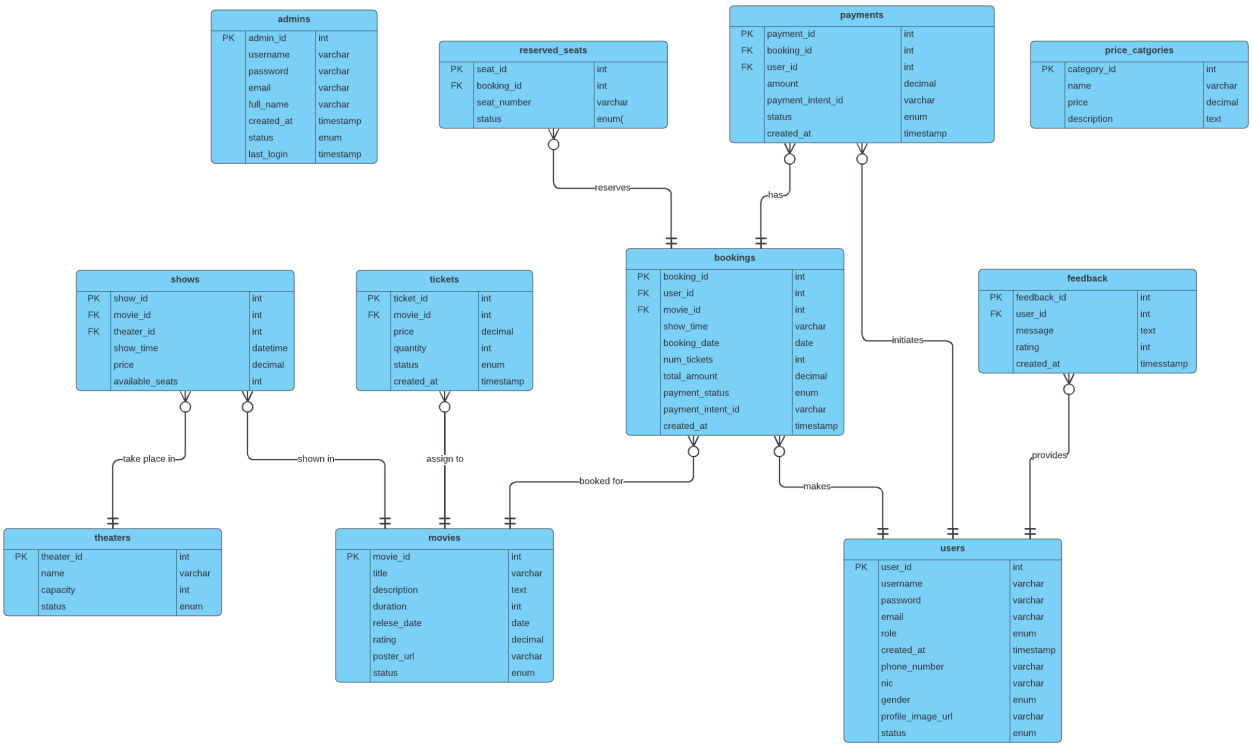
This class diagram represents the structure of your system. It's like a blueprint to the system. It identifies the key classes, which include Admin, User, Movie, Theater, Show, Booking, Payment, Ticket, ReservedSeat, and Feedback, each encapsulating relevant attributes and methods. System interactions and operations are managed by the Admin and User classes, with the Admin in charge of system control and the User in charge of reservations, payments, and feedback. ReservedSeat keeps track of assigned seating, while Booking, Payment, and Ticket classes manage reservations, payments, and ticketing procedures. The links between processes are established via relationships like dependencies and associations between these classes, which guarantee a coherent and useful system architecture.

Sequence Diagram for Booking a Reservation



The user-system interaction during the movie ticket booking process is represented in the sequence diagram. The system verifies login information and seat availability once the user chooses a movie, showtime, and seats. The Payment Gateway (in our case Stripe) is used to handle the user's payment when seats availability has been confirmed. When the payment is successful, the system completes the reservation and emails the user a confirmation along with the ticket. This ensures a smooth and well-organized reservation procedure.

ER Diagram



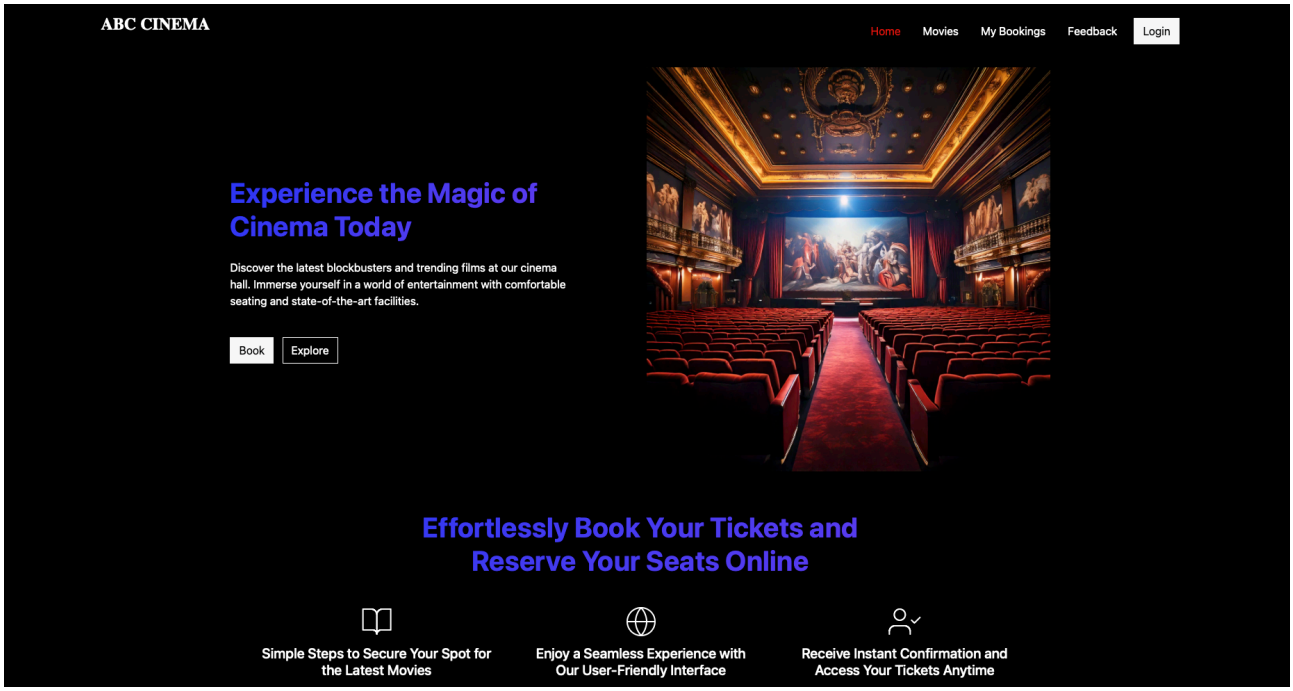
The database structure of our system, which includes things like admins, users, movies, theatres, shows, bookings, payments, tickets, reserved seats, and feedback, is represented in the ER Diagram. It describes the characteristics and connections between various entities, including the fact that Movies include details about the film, theaters store details about the movie theater, shows link Movies to Theaters for certain showtimes, and Admins and Users store login credentials. Reserved seats ensure seat availability, payments keep track of financial transactions, bookings monitor reservations, and feedback keeps track of user feedback. To ensure data consistency throughout the system, foreign keys connect Users, Movies, and Bookings.

Samples of Developed System (Screenshots)

Navigation Bar

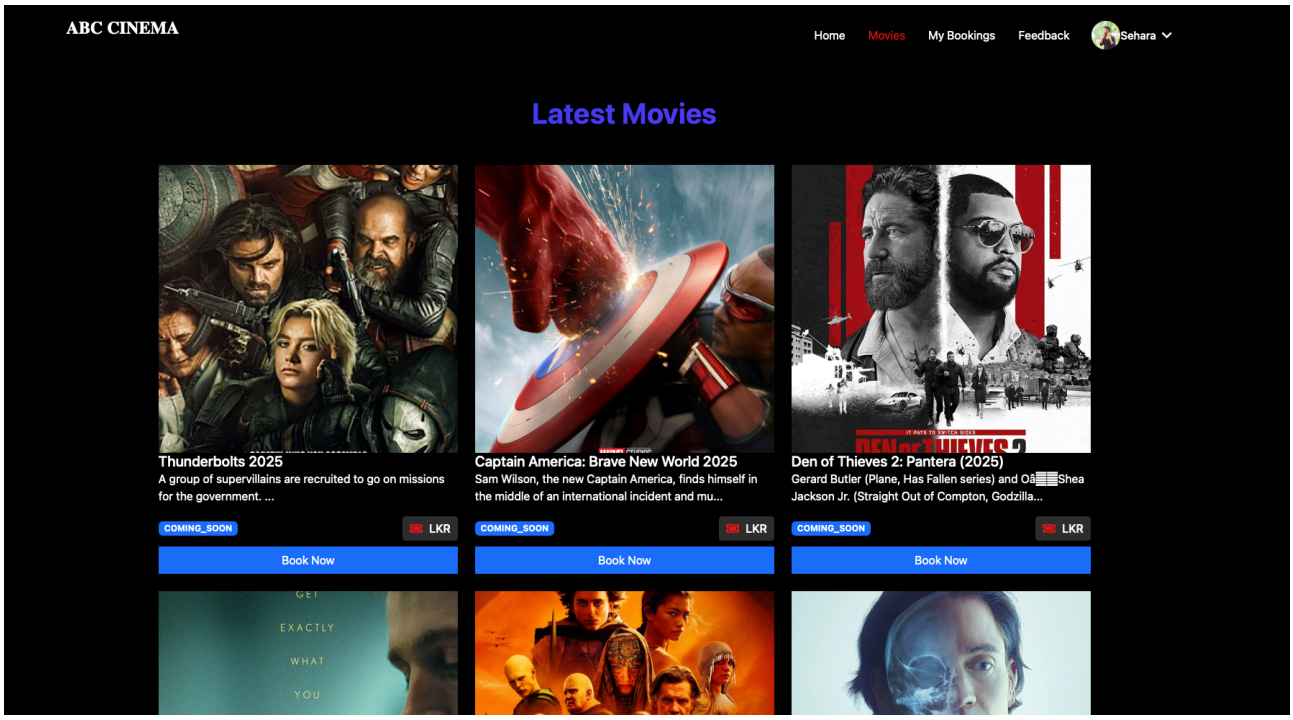
In our web application the navigation bar is designed to provide the users with clear access to the core functionalities in our system. The navigation bar includes sections such as home, movies, my bookings, feedback a login. After the user logs in to the system the login button in the navigation button replaces with profile image of the user along with the name of the user. Clicking on this shows a dropdown list including my profile, which allows users to edit their profiles, my bookings lead the users to my booking page and the logout button to log out from the system.

Home Page



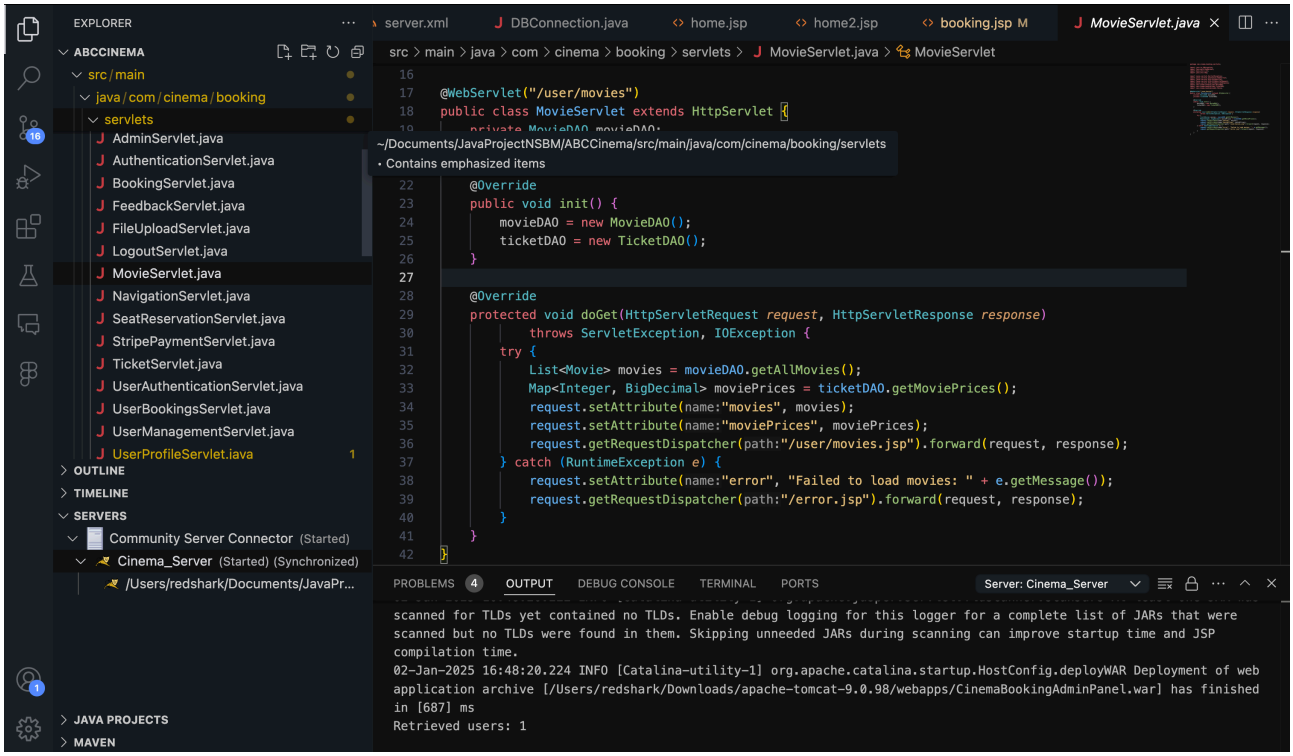
This acts as the main entry point of the website. This leads users to the landing page of our web application. This page includes descriptions of the cinema hall and details about the web applications functions like easy booking process. Unregistered users can view only this page and can register to the system using the login button and the book button. Unless they can't access the other pages.

Movies Page



This page represents the movies catalog of our cinema hall. This includes upcoming movies and now showing movies. Users can book tickets according to their choice of movies on this page. Users have to login or register to the system to see this page. The book now button in the movie cards leads user to the ticket booking page.

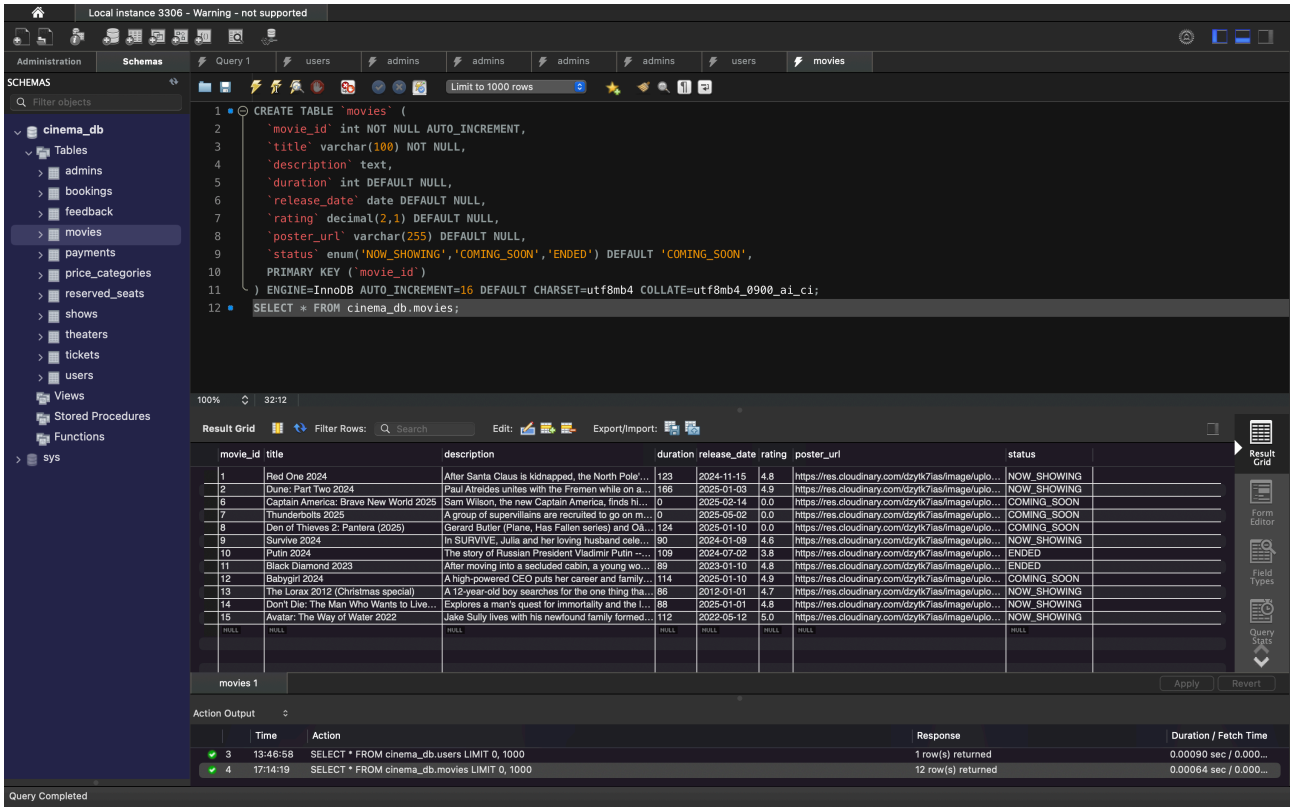
Movie Page Servlet



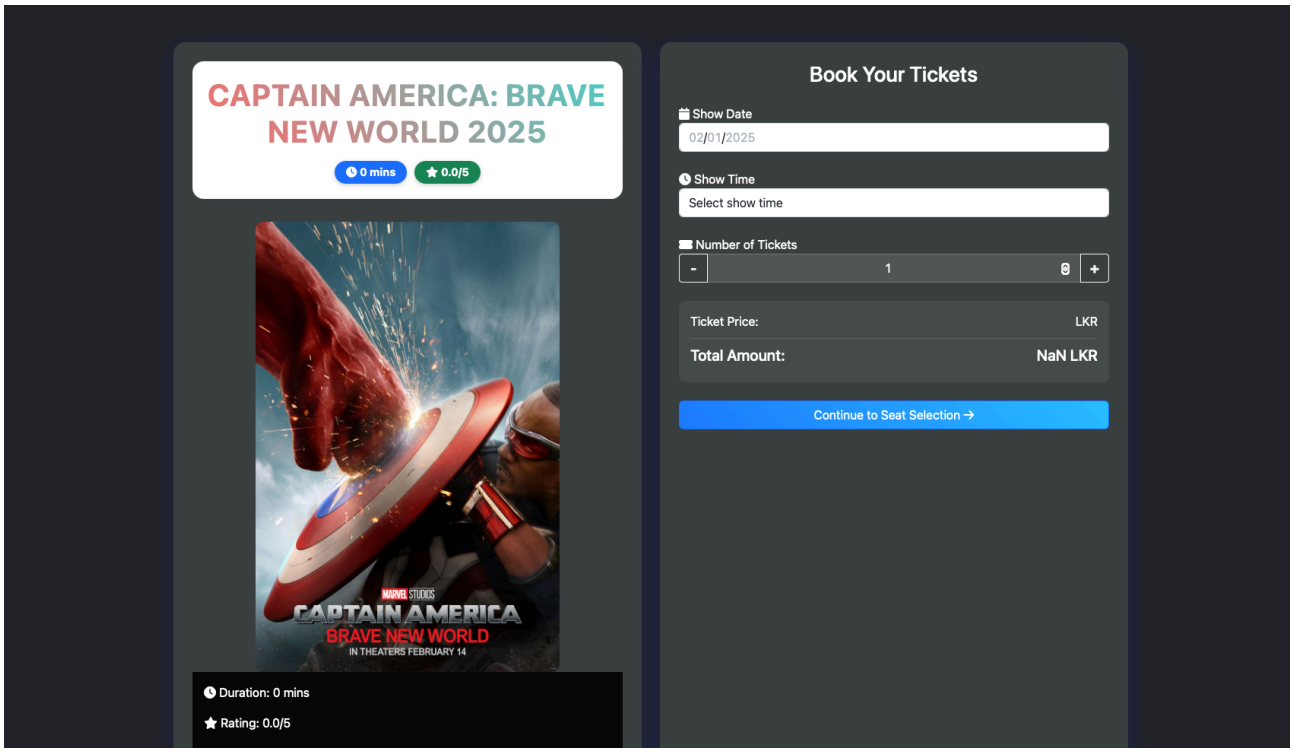
The **MovieServlet.java** class is a part of the cinema booking system and it is designed to handle requests related to Retrieving and displaying a list of movies available for booking.

Movie Page SQL Queries

In above picture, that's the SQL query that creates the movies table and the below 'Result Grid' shows the all available movies that we have added in Web-application.

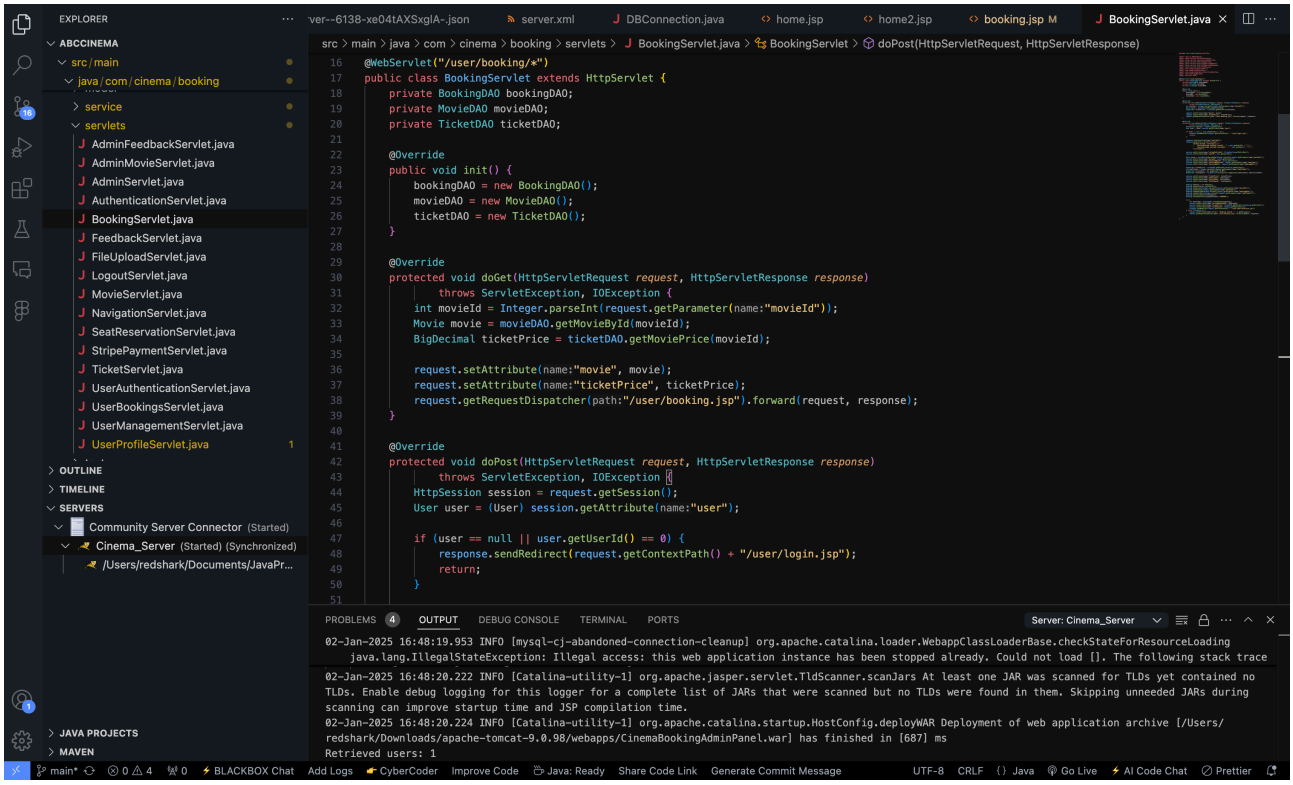


Booking Page



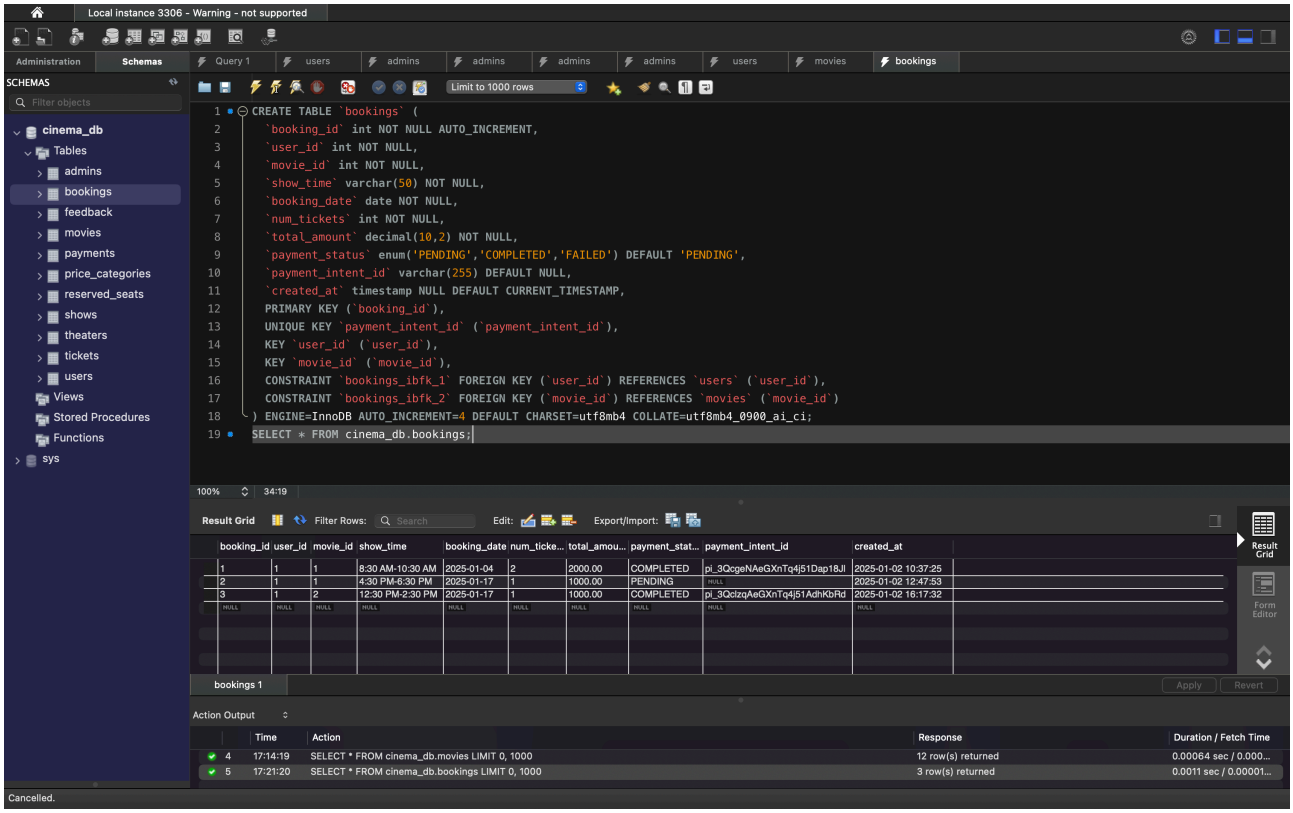
This page allows users to book their ticket for the selected movie. In this page user can choose the show date, show time and number of tickets. This page shows the price of the tickets according to the user's input. Then clicking the Continue to Seat Selection button leads user to the seat's selection page.

Booking Servlet



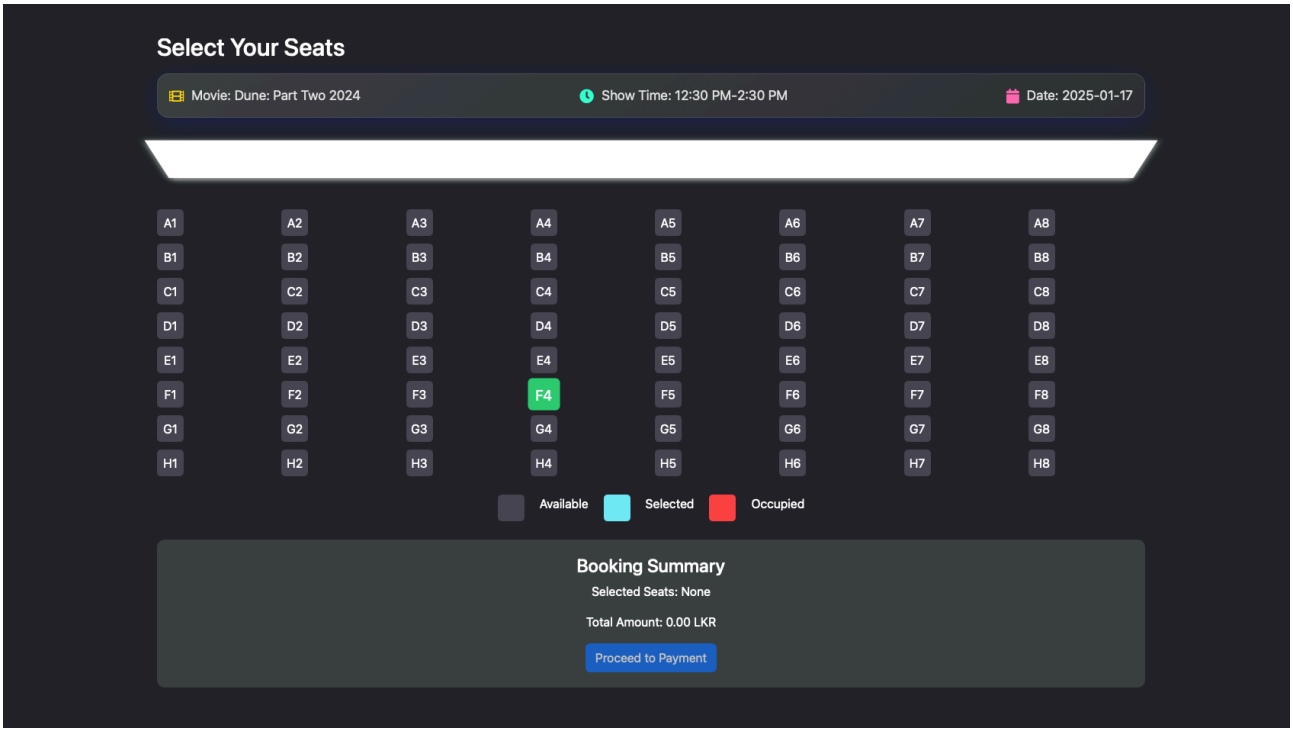
The **BookingServlet** is a core component of the 'ABC Cinema' ticket booking system that handles user requests related to booking movies. It facilitates the process of selecting movies, choosing showtimes, specifying the number of tickets, and initiating the booking process.

Booking SQL Queries



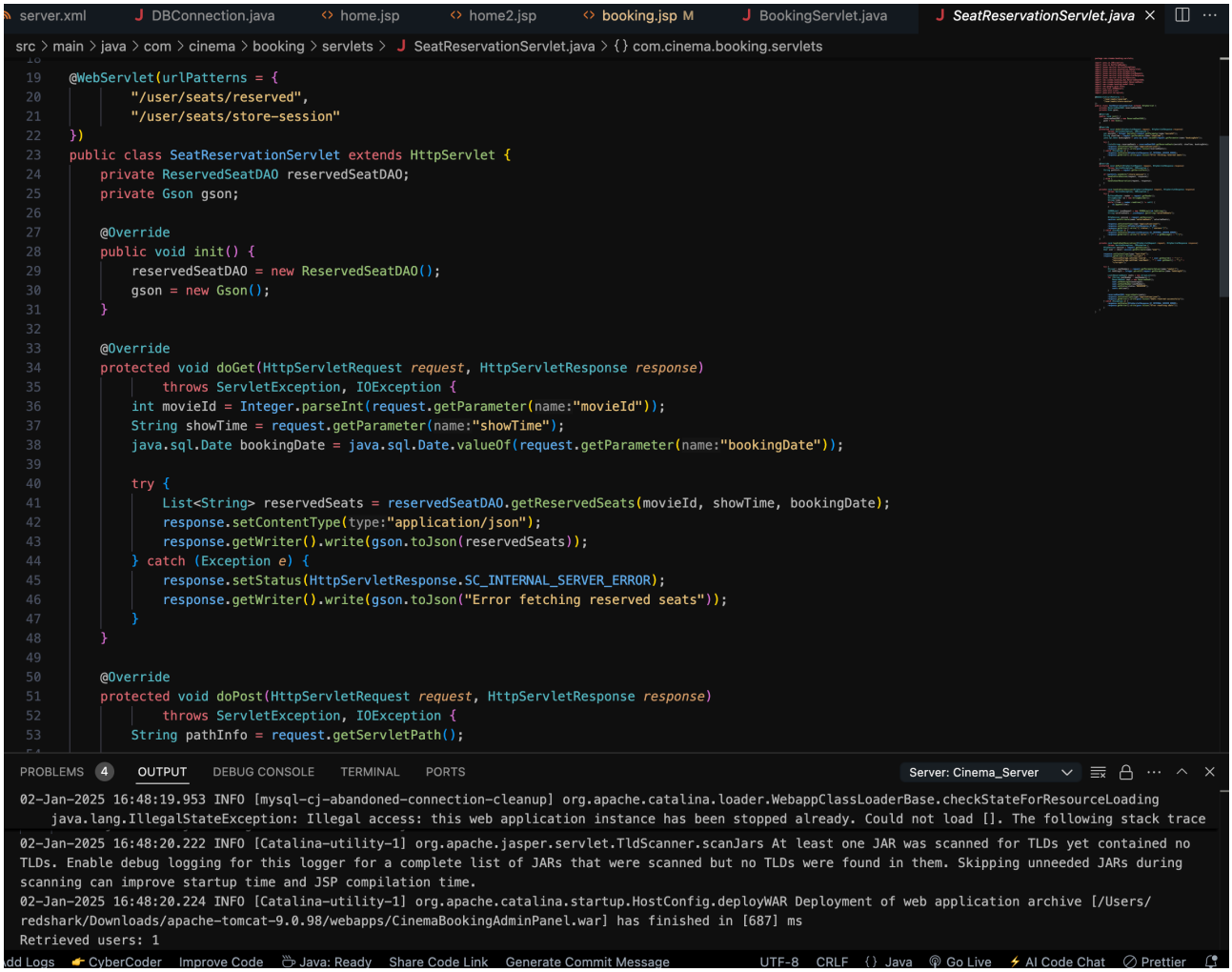
The above SQL query show the database table that holds booking details of the users.

Seat Selection Page



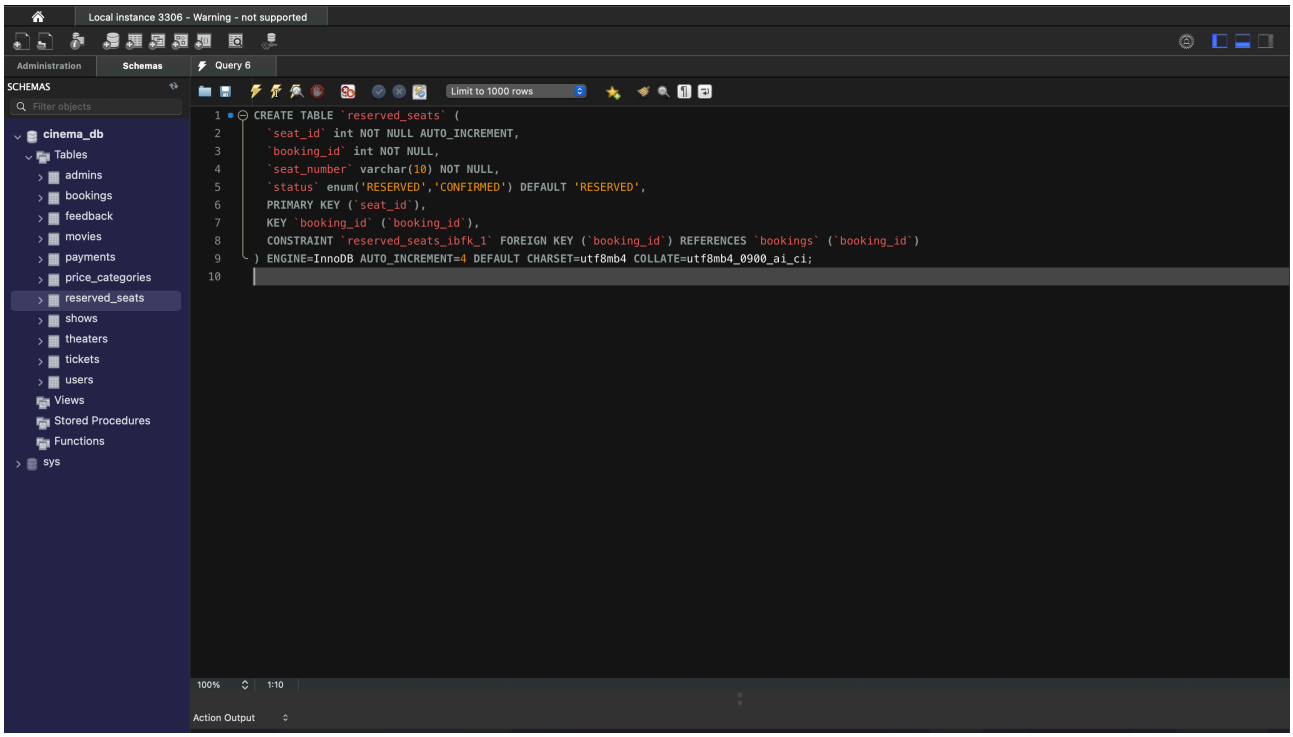
This page shows the users how the screen and the seats are in the cinema hall. And by clicking on the seat numbers the user can select the seat for only the number of seats that user have entered before. If a previously logged user has booked seats before the current user, those seats represent in red color and the current user cannot book those seats. This avoids the seats collision. After selecting the seats, the user can provide the payment by clicking the Proceed to Payment button.

SeatReservation Servlet



The **SeatReservationServlet** is responsible for managing seat reservations in the cinema booking system. It provides functionalities to fetch reserved seats, reservation for new seats, and store seat reservation related data in the user session.

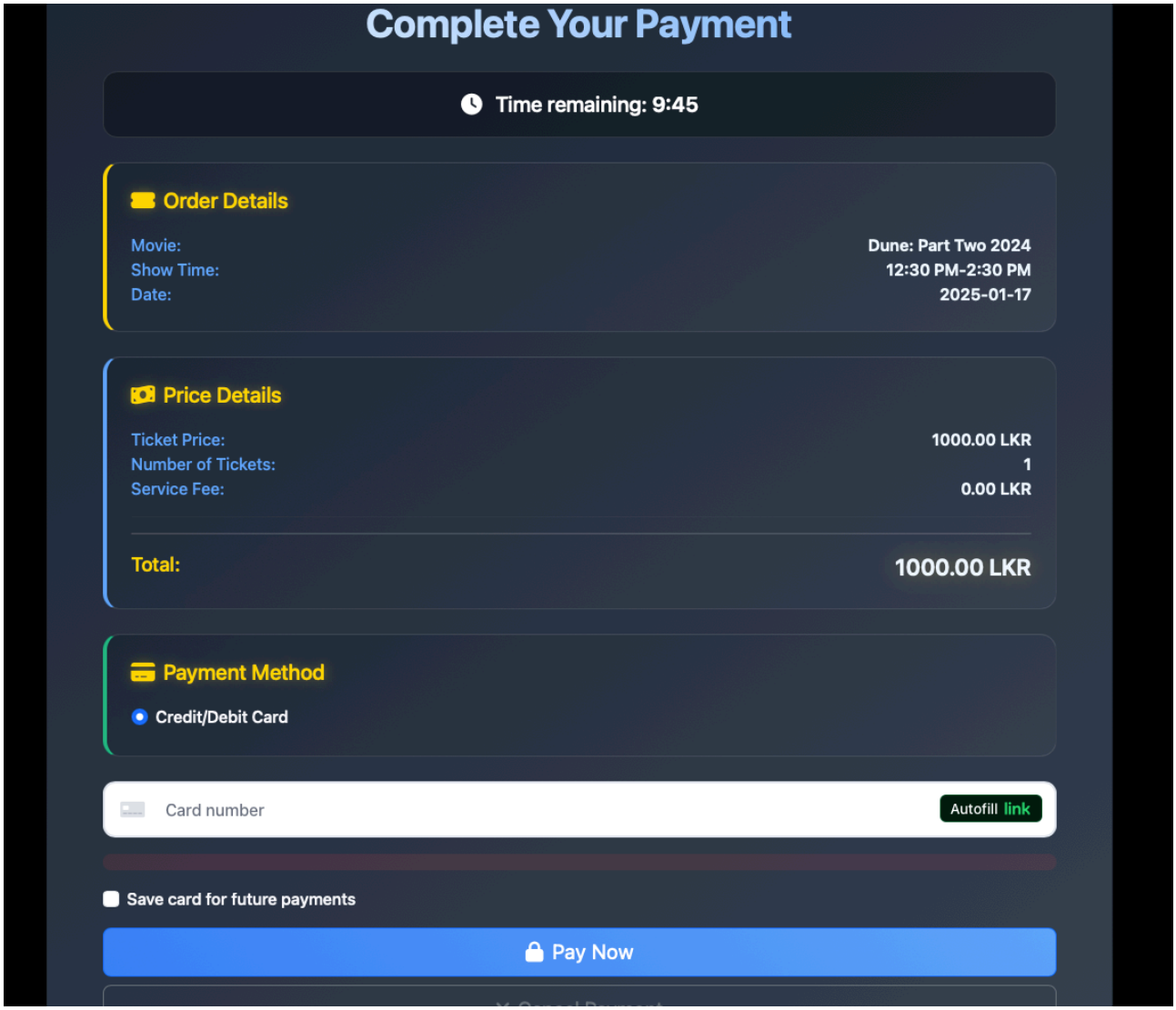
Reserved Seats Table Query



This table holds the data about reserved seats and when user requests a seat reservation servlet checks if the availability from this table.

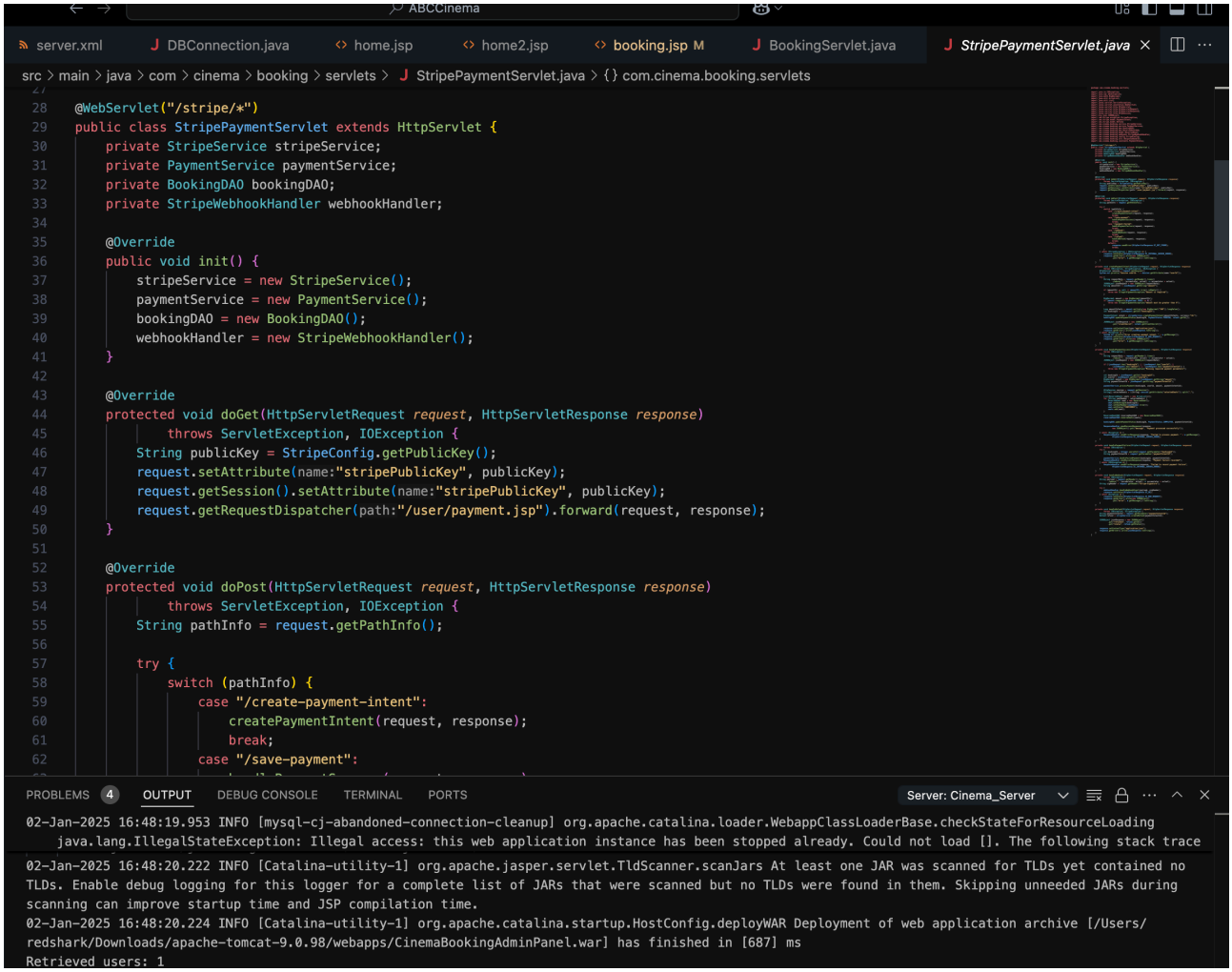
Payment Page

The payment sandbox that we have used is **Stripe** and it provides test keys for testing purposes. In above page users can provide their credit/debit card details and complete the payment procedure.

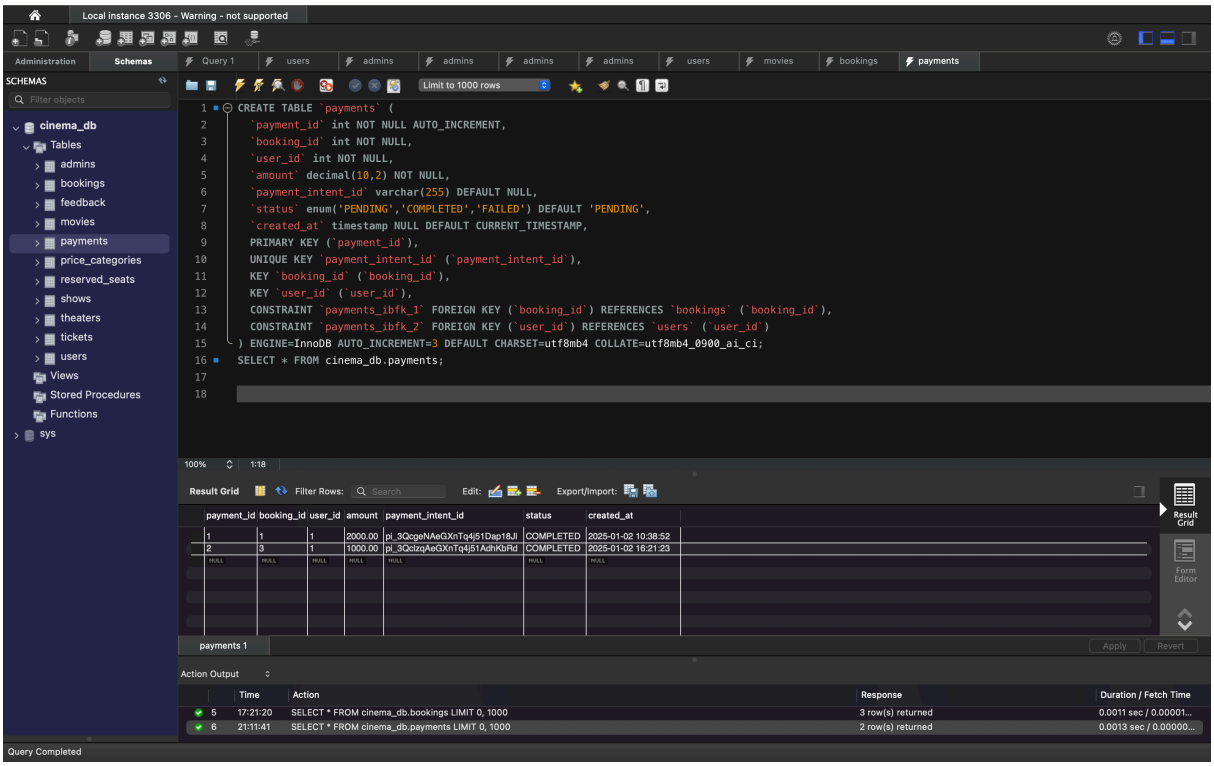

Complete Your Payment
Time remaining: 9:45
Order Details
Movie:
Dune: Part Two 2024
Show Time:
12:30 PM-2:30 PM
Date:
2025-01-17
Price Details
Ticket Price:
1000.00 LKR
Number of Tickets:
1
Service Fee:
0.00 LKR
Total:
1000.00 LKR
Payment Method
Credit/Debit Card
Card number
Autofill link
Save card for future payments
Pay Now

StripePayment Servlet

For the ABC Cinema web application, StripePaymentServlet handles Stripe payment processes. It processes both GET and POST requests, manages payment activities through various endpoints, and initialises required services. Creating payment intents, managing Stripe webhooks, logging unsuccessful payments, processing successful payments, and starting refunds are some of the essential features. It refreshes the database with booking and payment statuses and stores temporary information using session data. The servlet handles mistakes gently and guarantees safe and effective payment integration.



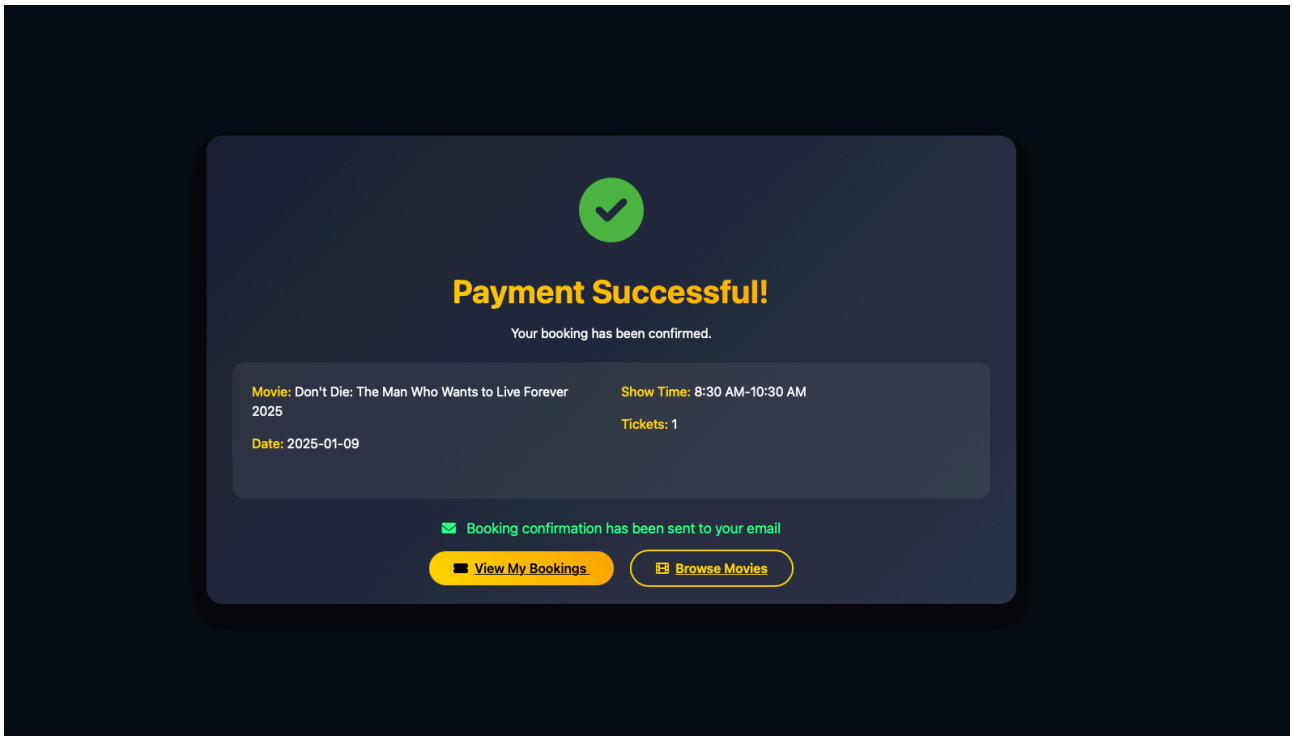
Payments Table Query



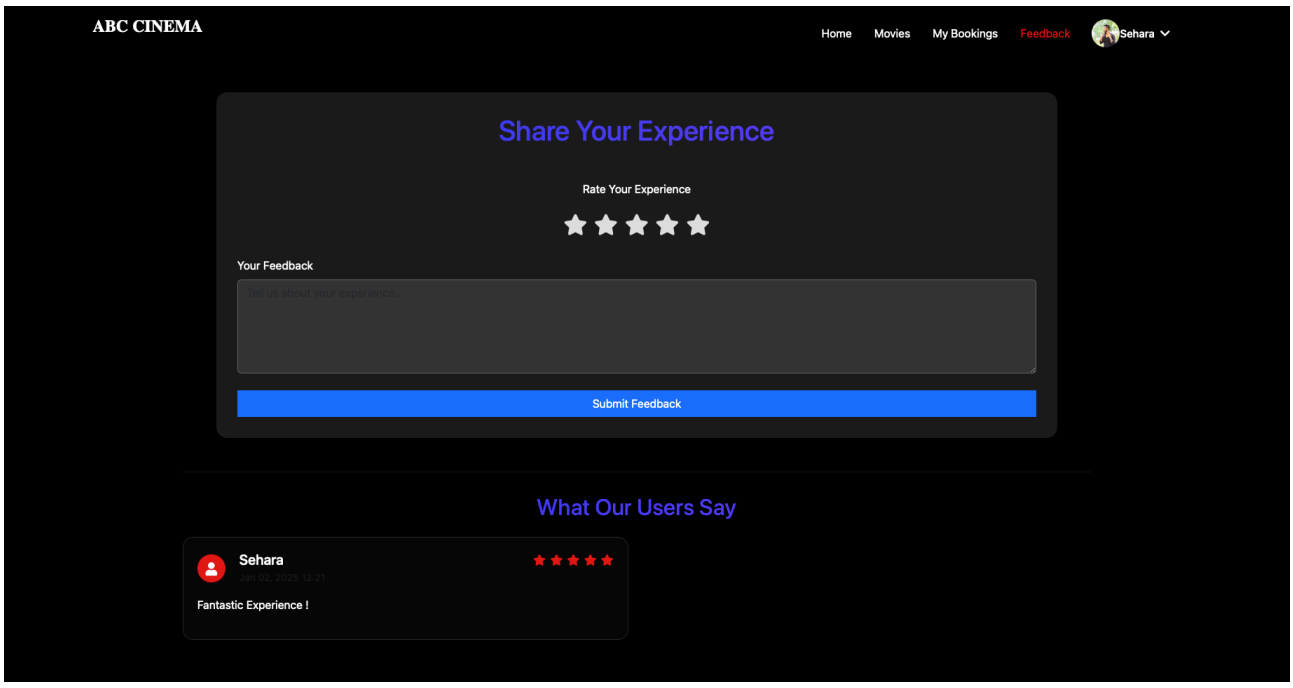
The above ‘Payments’ table holds the data that stored from the user payment sessions.

Payment Success Page

The above page popups when the payment is completed and the Ticket Reservation receipt will be send to the user’s registered email address and by clicking on the ‘View my bookings’ users can directly view their bookings from Web-Application. And clicking on the ‘Browse Movies’ users will redirected to the movie page.



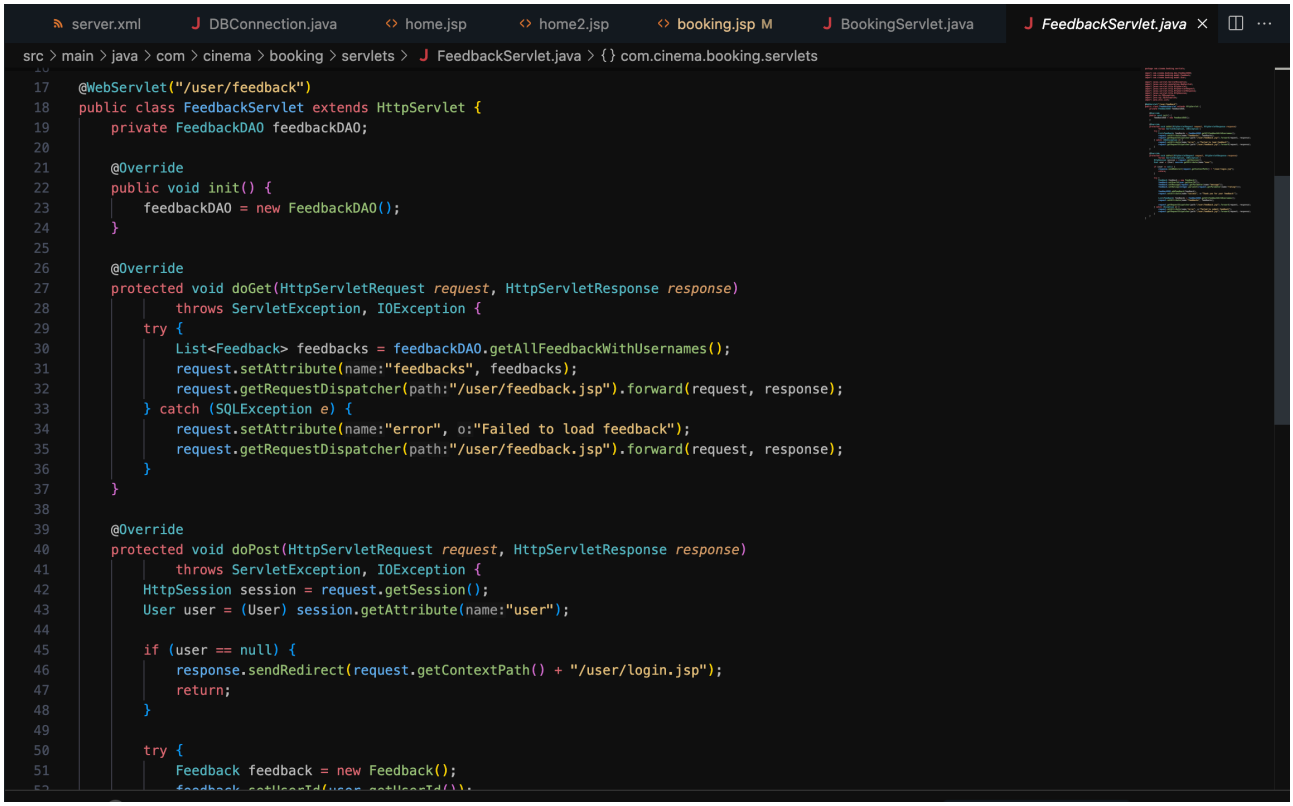
Feedbacks Page



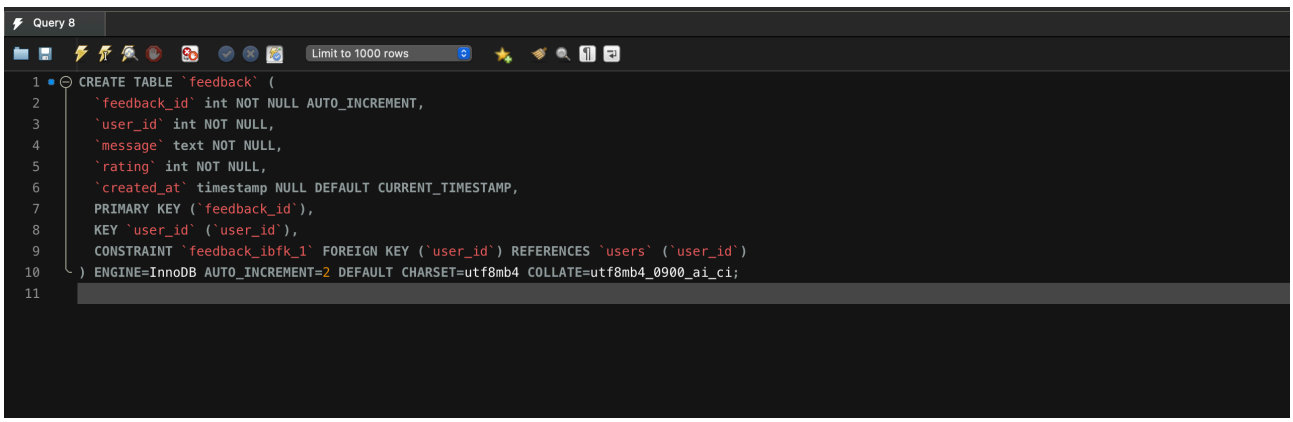
Users can click on ‘Feedbacks’ button in navigation bar and it will redirects user to the above page. First users have to click on star icons to rate the ‘ABC Cinema’ and they can post about their thoughts by writing them below and clicking the ‘Submit Feedback’ button. The submitted feedbacks are shown below the feedback card.

Feedback Servlet

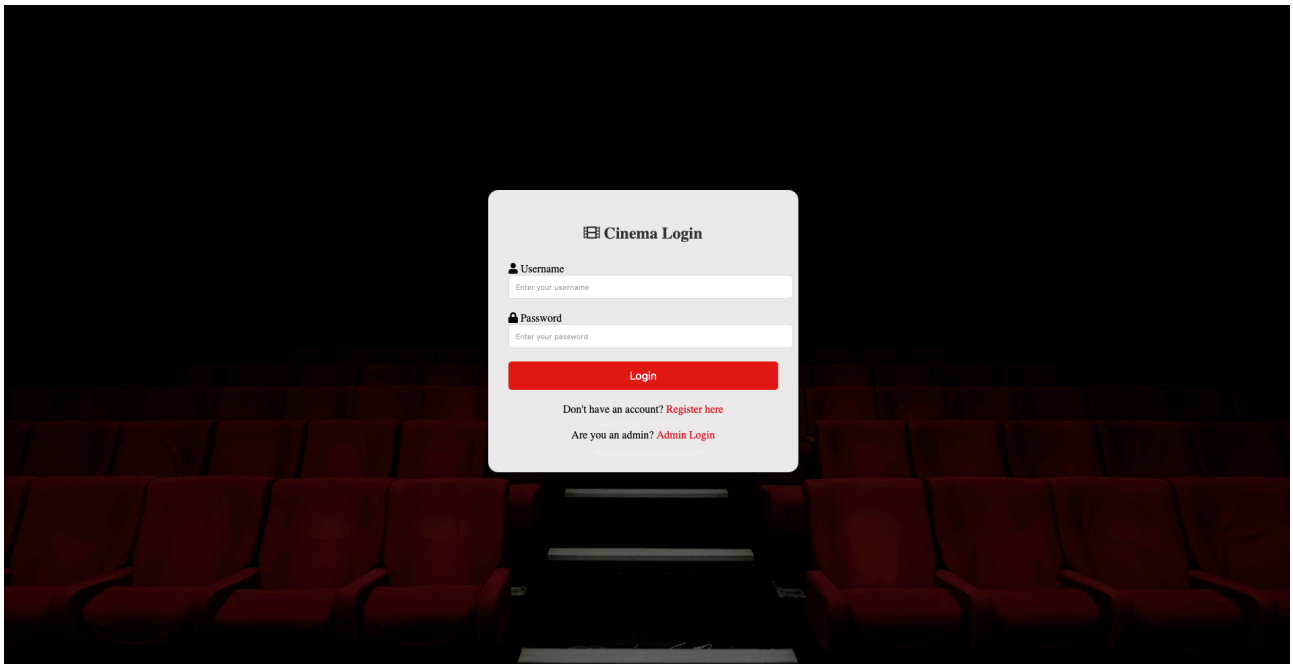
The below feedback servlet is used to display all the feedbacks in a user-friendly style using comments and associated usernames. And also the ‘Feedback Servlet’ captures users’ inputs and saved them to the database. This will only allows logged in users to submit a feedback.



Feedbacks Table Query



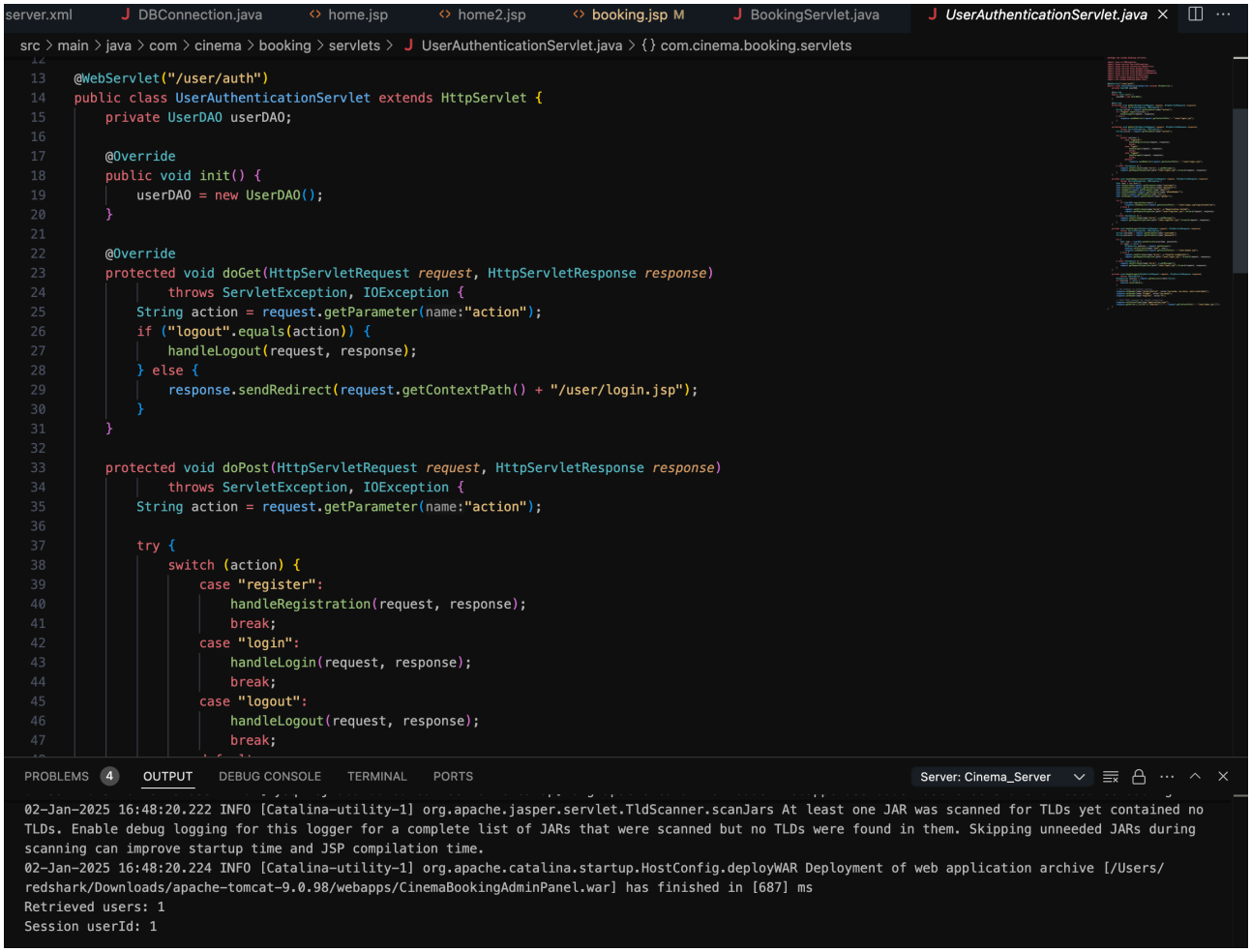
Login Page



The above page is the login page of ‘ABC Cinema’ and users have to enter their registered username and password to login to the Web-Application and if a new user comes they have to click on ‘Register here’ link and it will redirects in to the register page and after the registration they will redirect back to the login page. Then they can login using their credentials.

If a admin comes to the login page they have separate login interface for admin panel. Admin panel login page explanation can be seen below after the user login interface explanation.

User Login handling Servlet



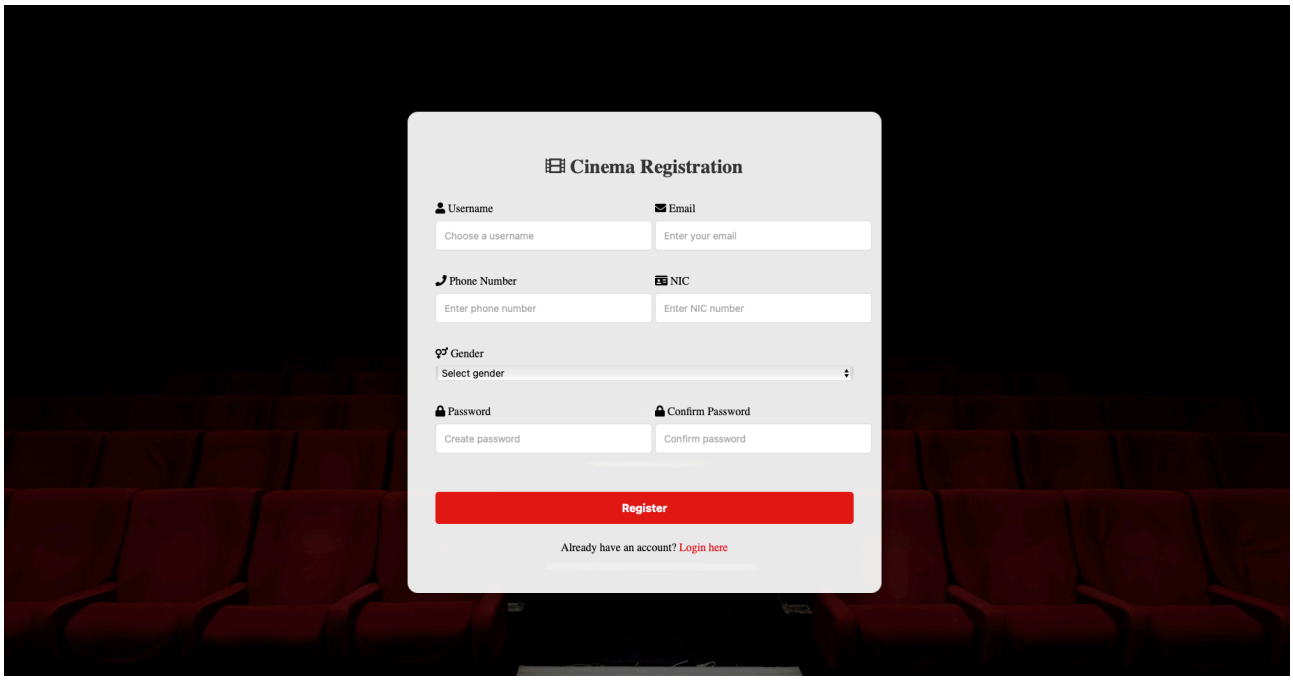
For the web application, the UserAuthenticationServlet manages user authentication tasks including registration, login, and logout. The action parameter determines the steps it takes while processing HTTP GET and POST requests. Depending on the activity, the doPost method leads to the registration, login, or logout methods. The UserDAO class is used to gather and store user data for registration. If credentials are good, the servlet creates a session after authenticating the user for login. The response contains headers to avoid caching and a JSON redirect for a seamless transition in the event of a logout, which invalidates the session. Setting error messages and sending requests to the relevant JSP pages are how errors are handled.

Users Table Query

The below ‘users’ table stores all users details and the users’ passwords are hashed because of security reasons.

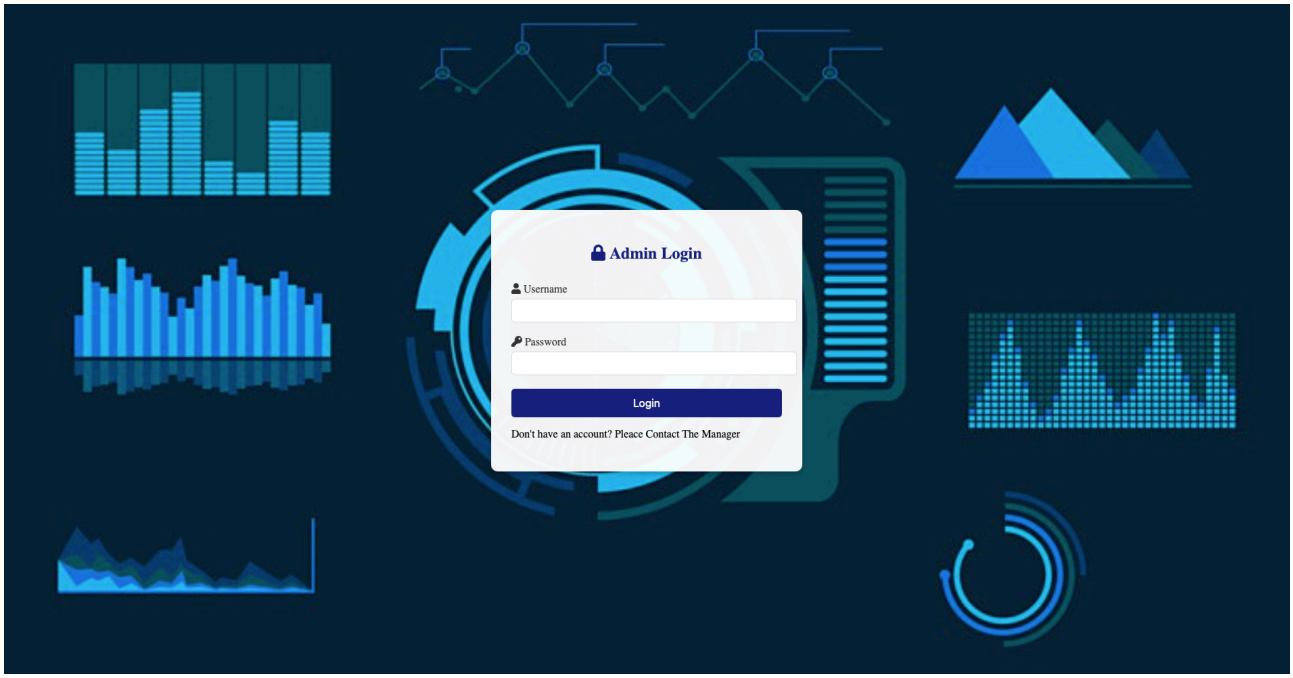
```sql
CREATE TABLE `users` (
  `user_id` int NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(100) NOT NULL,
  `role` enum('ADMIN','USER') DEFAULT 'USER',
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `phone_number` varchar(15) NOT NULL,
  `nic` varchar(10) NOT NULL,
  `gender` enum('MALE','FEMALE','OTHER') NOT NULL,
  `profile_image_url` varchar(255) DEFAULT NULL,
  `status` enum('active','inactive') DEFAULT 'active',
  PRIMARY KEY (`user_id`),
  UNIQUE KEY `username` (`username`),
  UNIQUE KEY `email` (`email`),
  UNIQUE KEY `nic` (`nic`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
SELECT * FROM cinema_db.users;
```

User Register Page



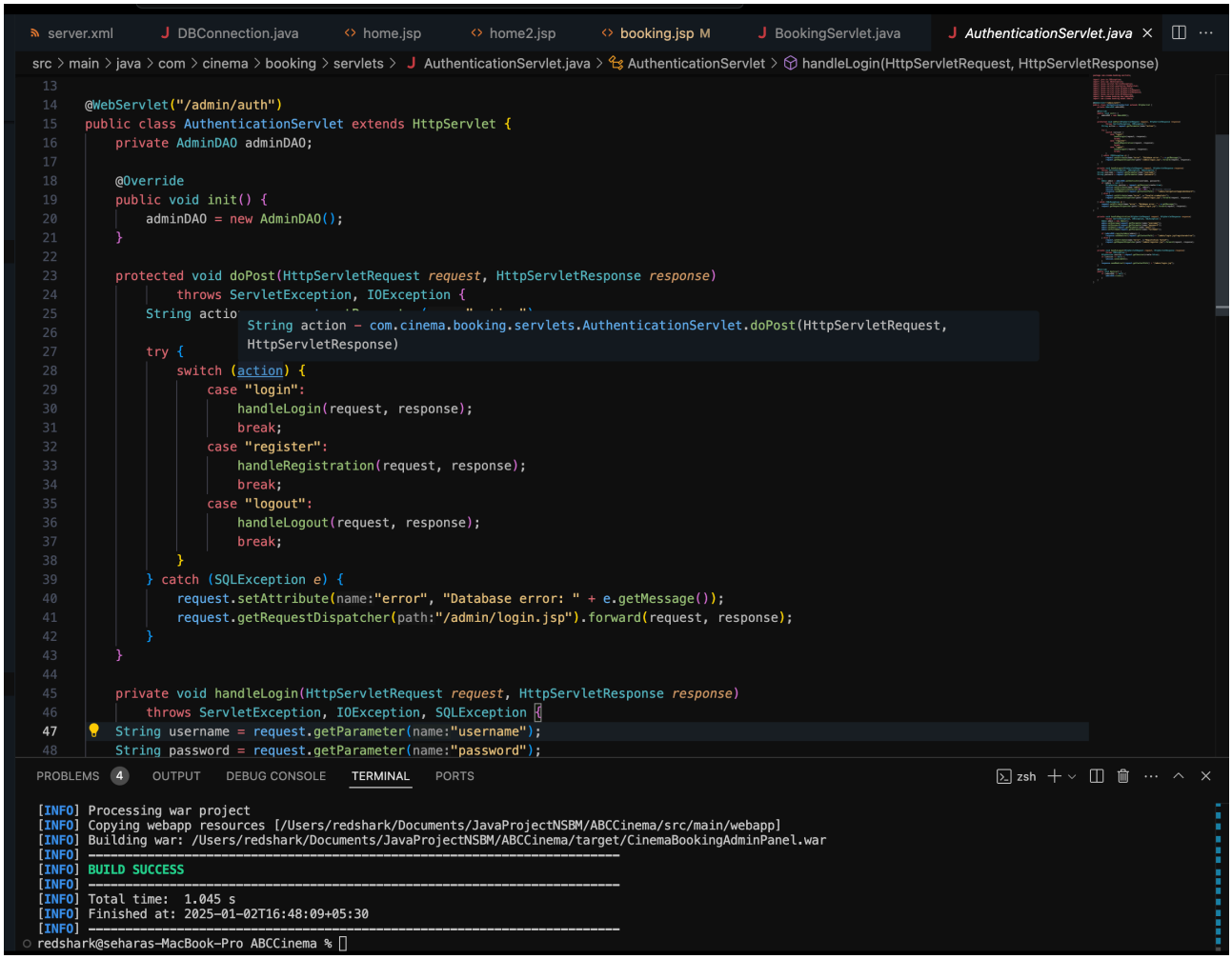
In the above user registration page, new coming users can create an account of ‘ABC Cinema’. At first user have to enter specific user name and their email. Then the users have to enter their local telephone number and National Identity Card number. Then, from the gender dropdown menu they have to select the gender and enter secure password for web application. After that by clicking the Register button, if given the informations are valid, the user will redirect to the login page.

Admin Login Page



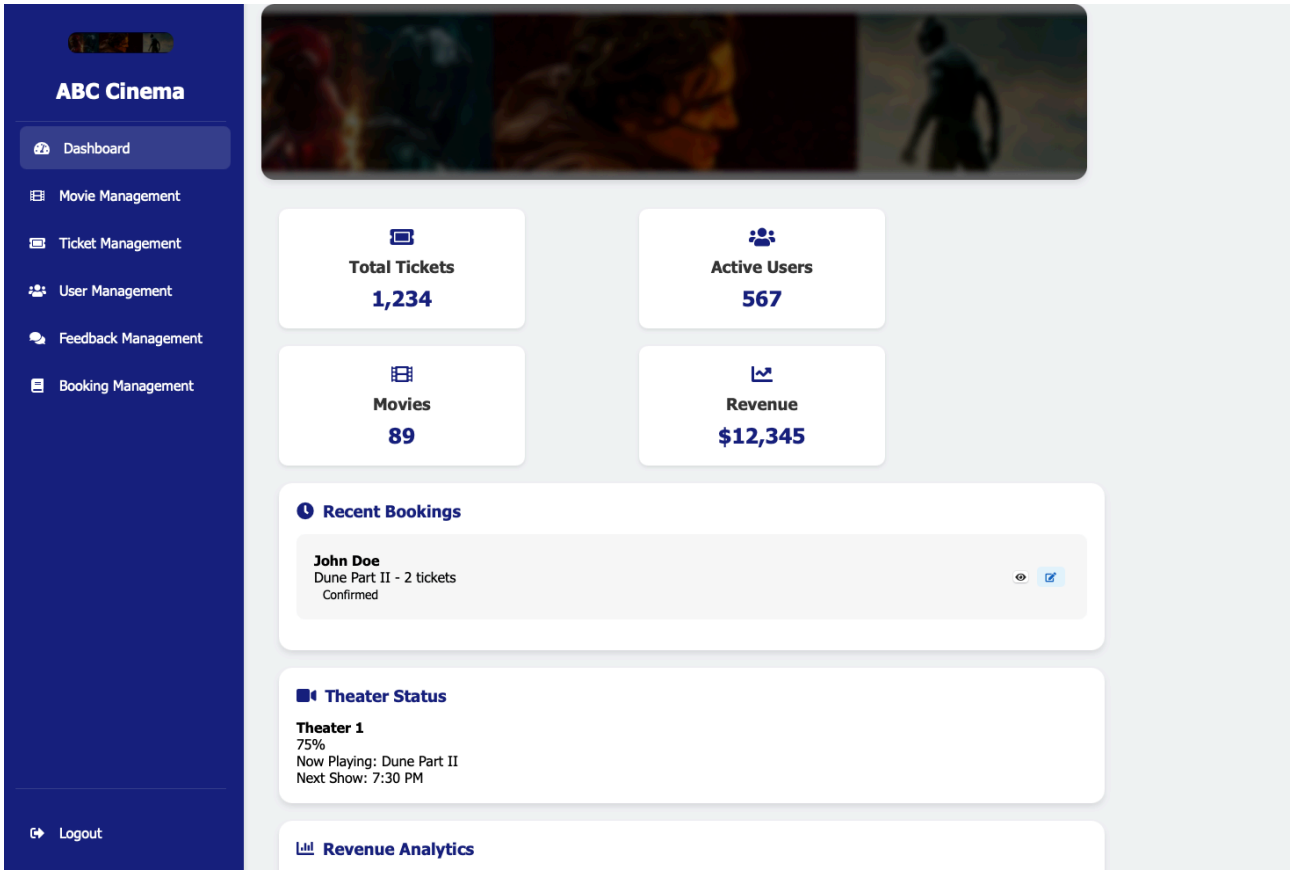
In the above admin login page, admin’s have to log in to the system via the given credentials that have given by the company (ABC Cinema). The ‘Login’ button will redirects to the ‘Admin Dashboard’ that will explained below.

Authentication Servlet



User authentication for the admin section of the ABC Cinema web application is managed by the AuthenticationServlet. It handles requests for registration, login, and logout. The action (login, register, or logout) is identified in the doPost method, which then sends the request to the relevant handler. In order to log in, it uses the AdminDAO to authenticate the admin credentials. If it is successful, a session is started. It gathers admin information, registers the admin with AdminDAO, and sends to the login page if registration is needed. It sends to the login page and invalidates the session for logout. The servlet forwards error messages to the appropriate JSP pages in order to handle exceptions, such as database failures. The destroy method also cleans up resources.

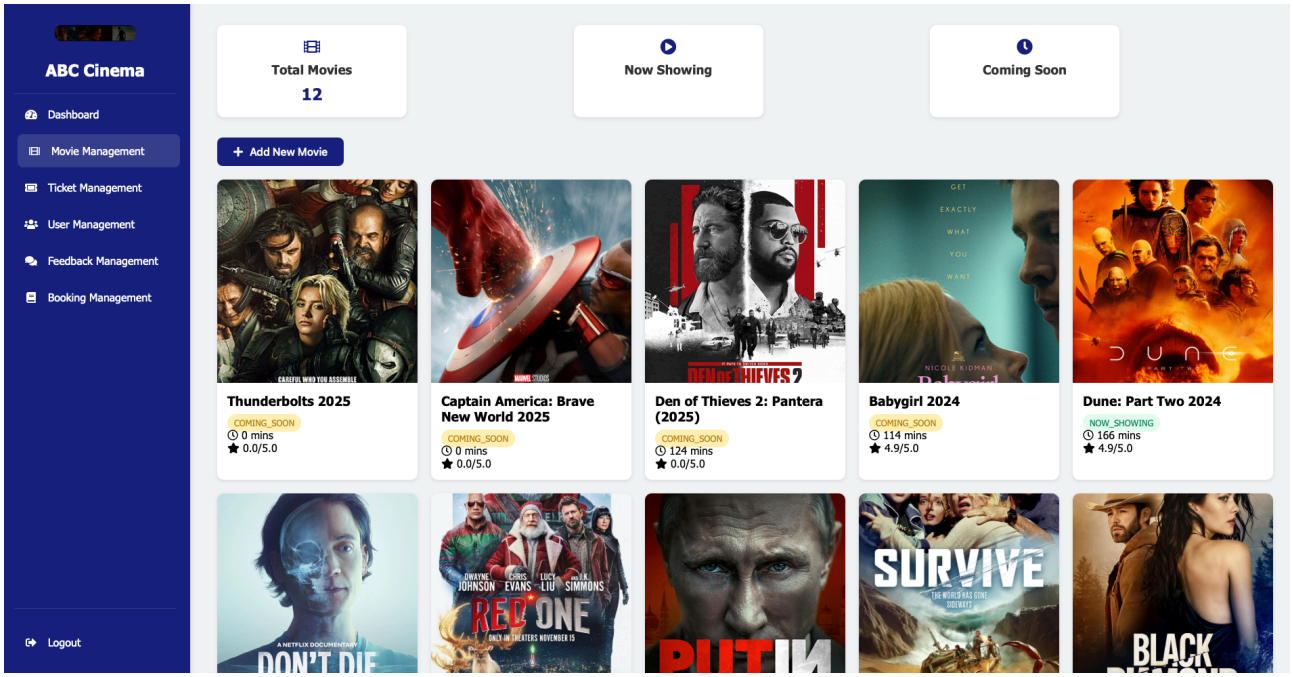
Admin Dashboard



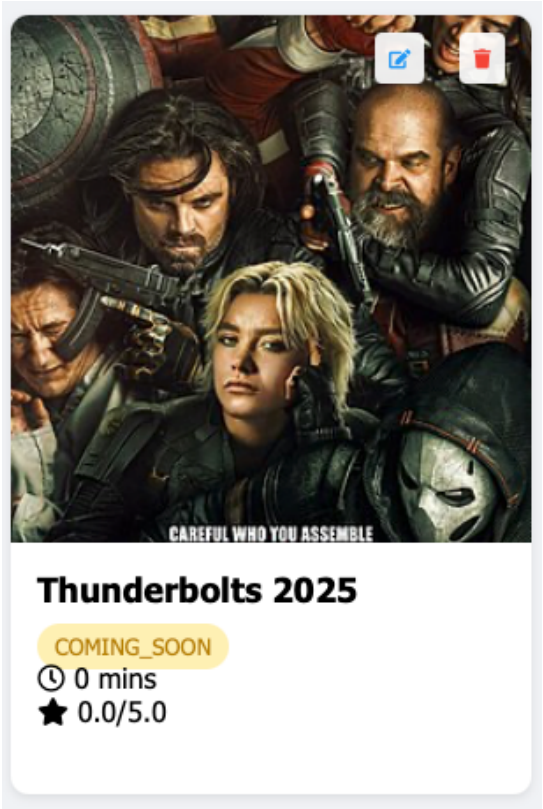
We have our admin dashboard, that the admin can do following operations:

- Movie Management
- Ticket Management
- User Management
- Feedback Management
- Booking Management

Movie Management



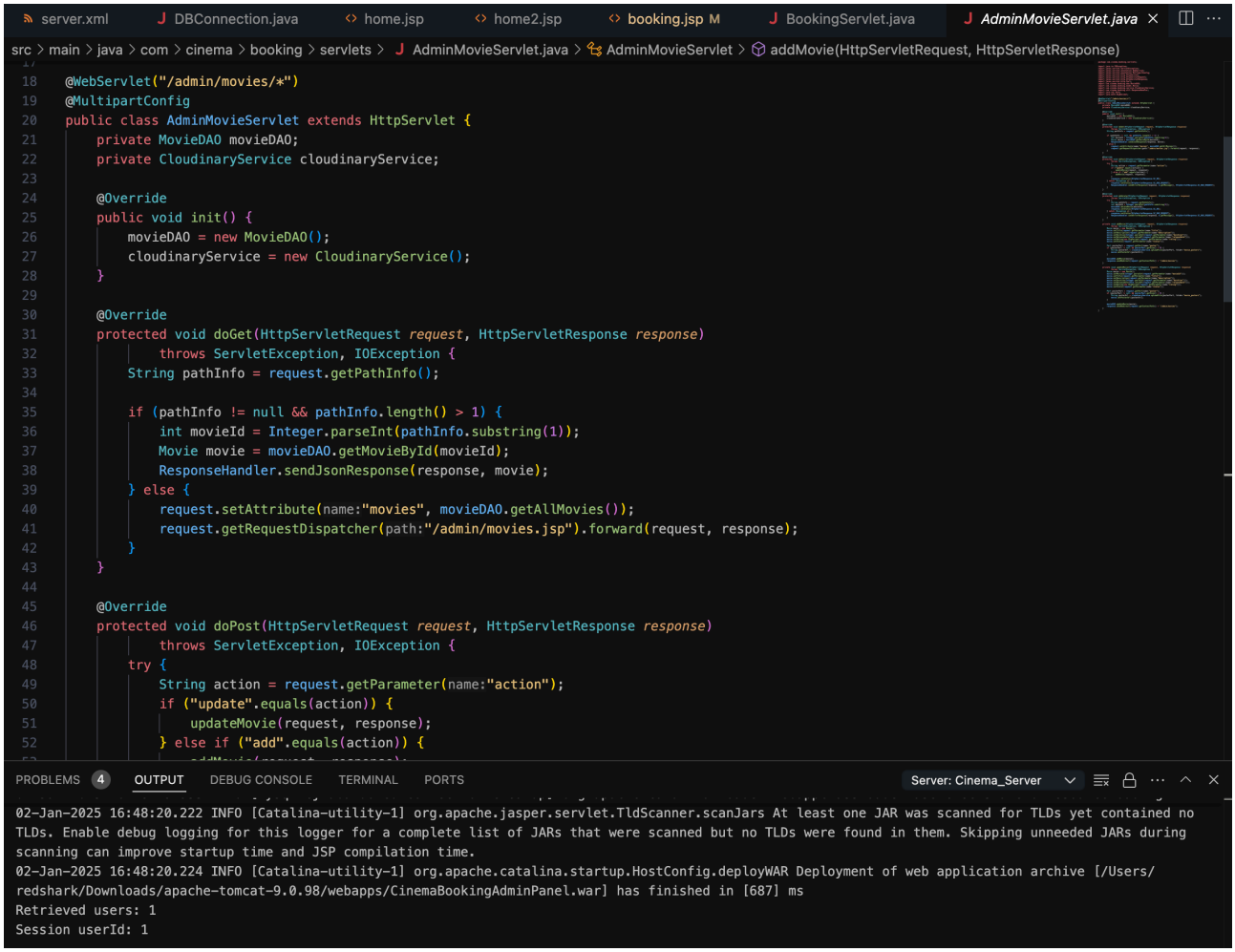
In the movie management page, admins can add new movies, edit movies, and delete available movies.



As this sample movie card, admins can use edit button to view the editing panel and by clicking the delete icon, admins can delete the movies.

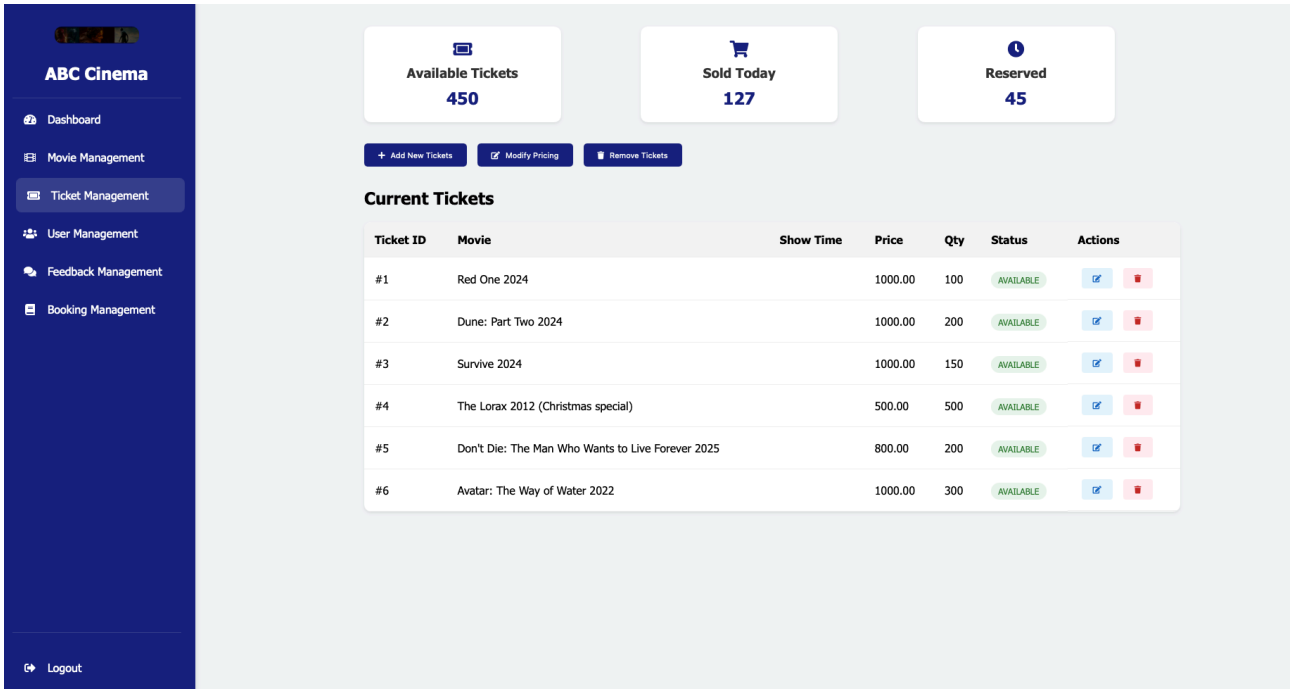
In the editing panel, admins can change the title, description, duration, rating, release date, status, and movie poster.

Admin Movie Servlet

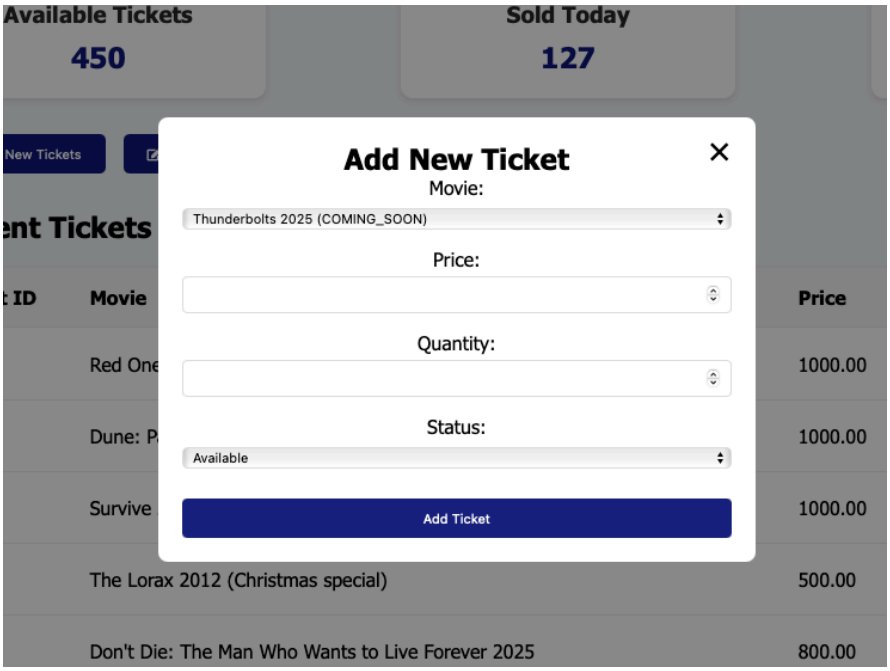


Movie management for the web application's admin section is managed by the AdminMovieServlet. It allows to view, add, edit, and remove movie details. The doGet method either lists all movies for display or uses its ID to obtain a single movie. The doPost method handles movie additions or modifications by collecting form data, using the CloudinaryService to handle file uploads (such movie posters), and storing the movie details in the database. Movies can be deleted by their ID using the doDelete method. The servlet utilises the ResponseHandler to transmit responses and communicates with the MovieDAO to carry out database operations. Additionally, it handles file uploads via multipart configuration.

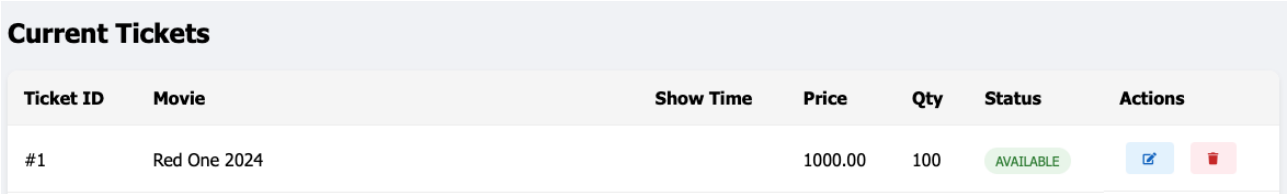
Ticket Management



The above Admin Ticket Management page allows admins to add new tickets, edit tickets and delete tickets.



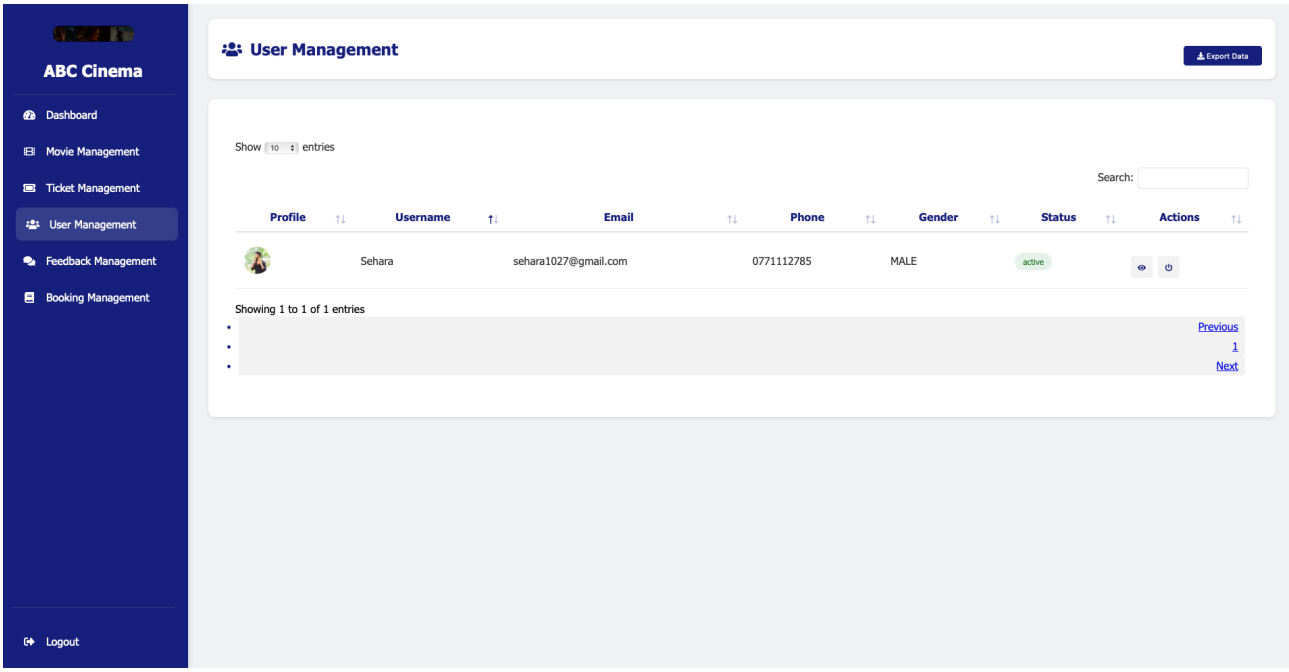
The ‘Add New Tickets’ button will open the above page and admins can add tickets to the available movies. Admins have to select the movie from the dropdown menu and after entering price and quantity of tickets, they can change the status (availability) of selected movie.



By clicking the ‘Edit Icon’ in the actions column admins can edit the added tickets’ details.

By clicking the ‘Delete Icon’ in the actions column admins can delete the added tickets.

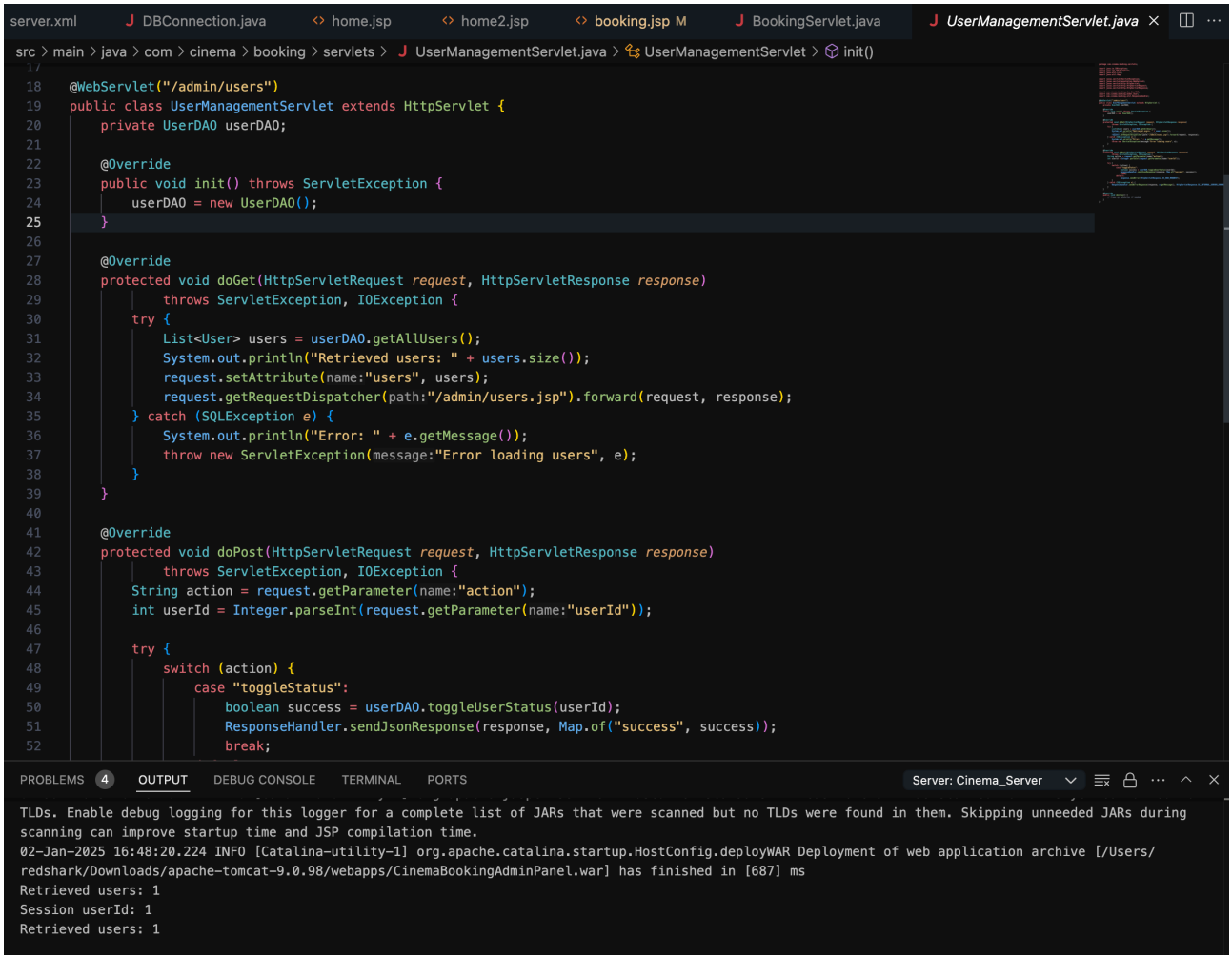
User Management



In the above admin user management page, will allow admins to view all the registered users details except the login password, because of security features. And by clicking the ‘Power button’ in actions column, admits can toggle the status of the users’ accounts.

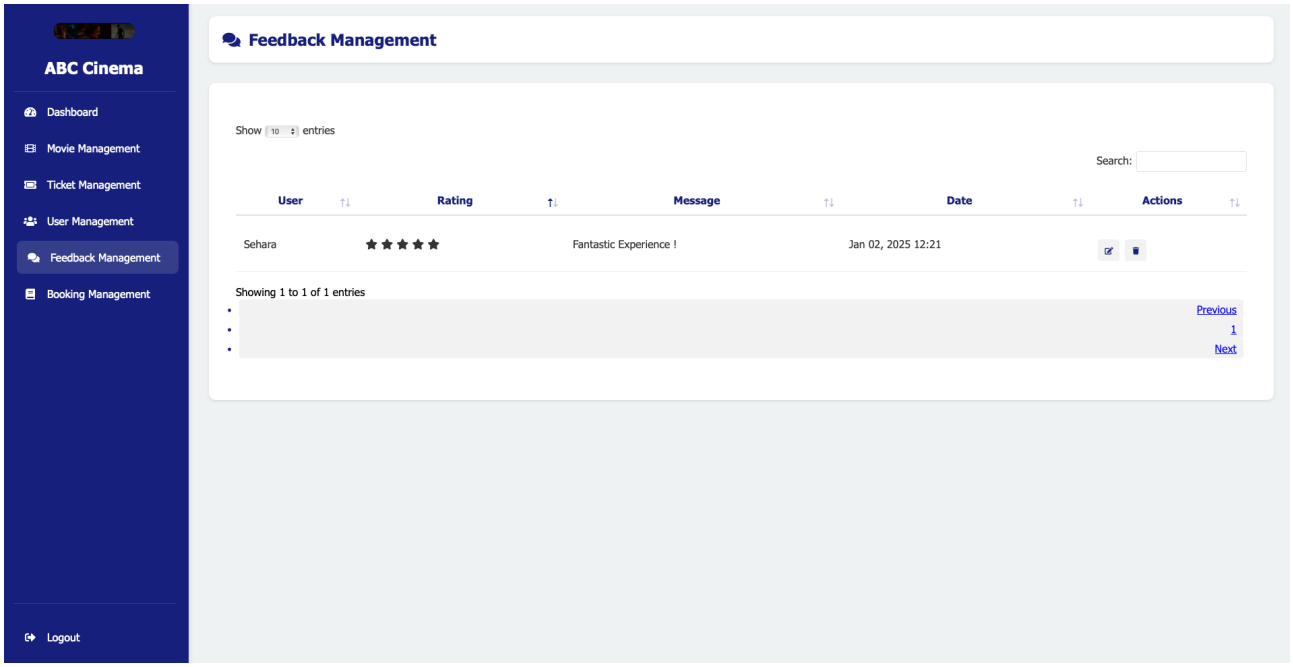
If the user’s status is active, user can login to their to the ABC Cinema and if admin made the status inactive, users can not login to the web application.

User Management Servlet



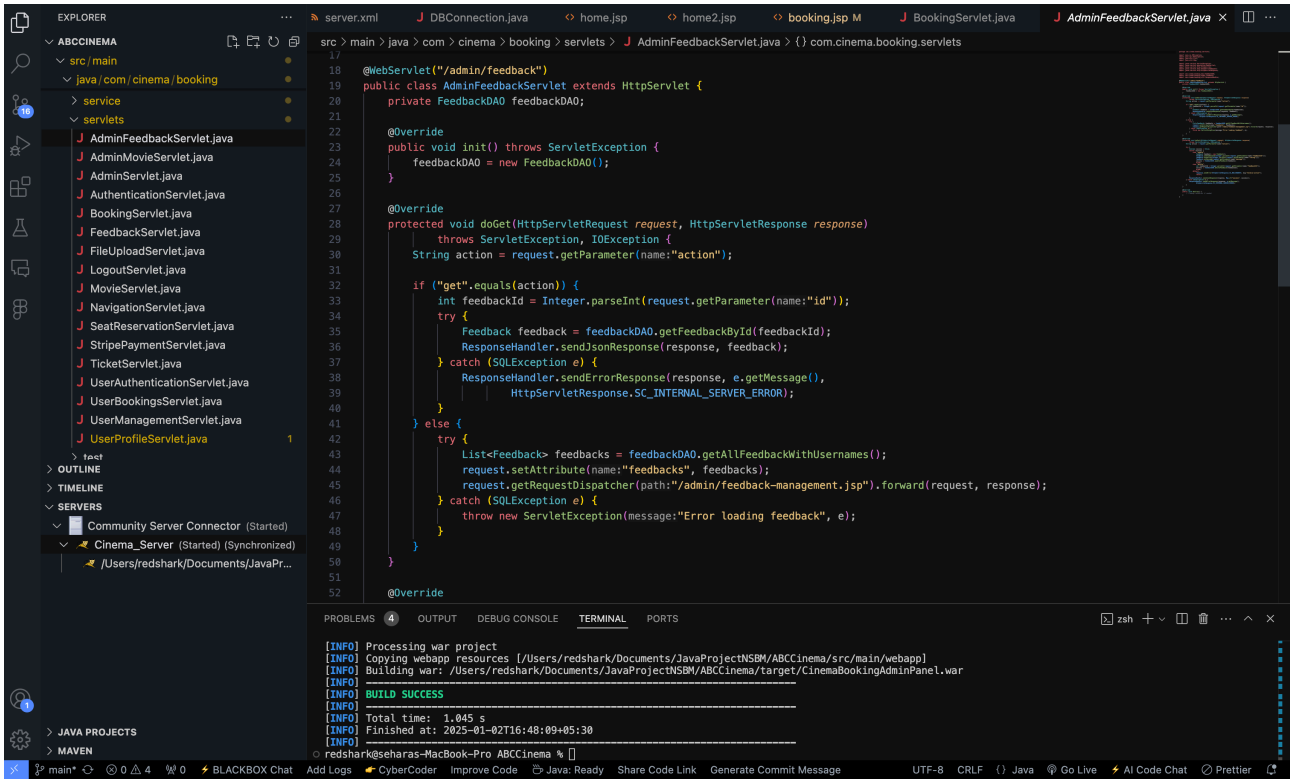
In the web application's admin part, the UserManagementServlet is in charge of overseeing user-related operations. The doGet function sends the user data to the users after retrieving a list of every user from the UserDAO.jsp page to be shown. It raises a ServletException if there is a problem fetching users. Toggling a user's status is handled by the doPost method, which calls the toggleUserStatus method of the UserDAO after receiving an action and a user ID from the request. Success or failure is indicated via a JSON response that is returned. The client receives an error response if there are any problems with the database action.

Feedback Management



In the above feedback management page, admins can view all the submitted feedbacks by users as well as their ratings for managerial decisions. By clicking the ‘Edit icon’ admins can edit the feedbacks if users posts inappropriate feedbacks. And from the delete icon admins can delete the feedbacks.

Admin Feedback Servlets



The AdminFeedbackServlet handles feedback management for the admin section of the web application. Both GET and POST requests are supported. The doGet function can either get every feedback entries and show them on the feedback management page, or it can retrieve specific feedback depending on its ID. Depending on the action indicated in the request, the doPost method manages updating and removing feedback. In the case of "update," a feedback entry is modified, whereas "delete" eliminates a feedback. The servlet utilises a ResponseHandler to provide JSON replies with the results after interacting with the FeedbackDAO to execute database operations.

Challenges

- Had to migrate from glassfish 7.0.15v and netbeans IDE to Visual Studio Code and Tomcat 9.0.98 because of port conflictions of group members.

- Had to clean the http and shutdown ports using nmap after scanning all ports of the system because testings of several servers has filled the most of the common ports.

- Had to fix merge conflicts of the GitHub to maintain the repository.

Conclusion

From this report, We have explained all the functionalities and how they will work in backend of ‘ABC Cinema’ Web-Application

References

Visual Paradigm (2024). *UML Class Diagram Tutorial*. [online] Visual-paradigm.com. Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/.

www.javatpoint.com. (n.d.). *Learn Servlet Tutorial - javatpoint*. [online] Available at: https://www.javatpoint.com/servlet-tutorial.

Stripe.com. (2024). *Manage access and API keys*. [online] Available at: https://docs.stripe.com/sandboxes/dashboard/manage-access [Accessed 2 Jan. 2025].

Cloudinary.com. (2024). *Cloudinary Java SDK Quick Start | Cloudinary*. [online] Available at: https://cloudinary.com/documentation/java_quickstart [Accessed 2 Jan. 2025].

tomcat.apache.org. (n.d.). *Apache Tomcat 9 (9.0.75) - Documentation Index*. [online] Available at: https://tomcat.apache.org/tomcat-9.0-doc/index.html.

Glassfish.org. (2018). *Eclipse GlassFish Application Deployment Guide, Release 7*. [online] Available at: https://glassfish.org/docs/latest/application-deployment-guide.html [Accessed 2 Jan. 2025].

code.visualstudio.com. (n.d.). *Getting Started with Java in Visual Studio Code*. [online] Available at: https://code.visualstudio.com/docs/java/java-tutorial.

maven.apache.org. (n.d.). *Maven – Configuring Apache Maven*. [online] Available at: https://maven.apache.org/configure.html.

Simplilearn (2020). *Java Servlets Tutorial | Java JSP Tutorial | Java Server-Side Programming For Beginners |Simplilearn*. [online] YouTube. Available at: https://www.youtube.com/watch?v=7AIjcZMo-V4 [Accessed 2 Jan. 2025].

Git (n.d.). *Git - Reference*. [online] git-scm.com. Available at: https://git-scm.com/docs.

Work Breakdown Matrix

Student ID (Plymouth)	Student Name	Contributed Tasks
10953243	Bedde Samarasinghe	UI / UX Design Front-End Development Report Writing Ticket Management Movie Management Payment Handling
10952466	Hewa Gamage	UML Diagrams Front-End Development Report Writing Ticket Management Movie Management Notification Handling
10953326	Yuhas Gamage	Testing and Debugging User Management
10952509	Amarasinghe Senevirathne	Admin Login and Register Movie Management
10903095	Liyana Gunawardana	Booking Management Admin Dashboard
10899703	Dasith Silva	Admin Dashboard Booking Management
10820754	Herath Vimalaranga	Feedback Management
10952758	Hiddiyadura Abeynayake	Feedback Management Ticket Management Admin Dashboard

GitHub Repository Link

All the projects' files, database backup file and this report is available in below GitHub repository:

https://github.com/RedsharkG/ABCCinema.git