Movie Ticket booking System

Course name = DAC (KP)

S.N.	NAME	PRN No.
1.	SWAPNIL JADHAV	200250120035
2.	SAGAR PATIL	200250120062
3.	SHUBHAM MISHRA	200250120095
4.	TEJAS GUPTA	200250120100

What Project Does:

1. Customer :
    - Customers’ data will be stored in a database in the “User” table.
    - User will login with the help of Email ID and Password

2. Homepage :
    - List of the movie will be shown on the home page, whose show is available.
    - User can select the movie with his choice.

3. After clicking on any movie :
    a. The list of dates of the movie will be shown to the user which he has selected.
    b. after selecting a particular date, all the shows of that day with theater names will appear.
    c. A user can select a suitable theatre with the particular show timing.

4. Theatre Seat:
    a. After selecting the particular show, Seat layout of that particular screen will be shown.
    b. Users can choose any seat he wants to reserve.
    c. The price along with no. of seats selected by the user will be shown until he buys the tickets.

5. Book Ticket :
    a. After selection of the seat , the user can book the ticket by clicking the Book ticket button.
    b. After the booking process, booked seats will be shown as occupied and no new user can book those occupied tickets.

Take one scenario and explain what happens in presentation

1. Imagine a user wants to book a seat in xyz theatre for an abc movie.
2. After selecting a particular show , angular will send the show id to the backend.
3. Then in the Spring Boot , Seat Controller GetListOfSeats() method is called.
4. The GetListOfSeats() will call the service layer.
5. In the service, we get all the seat lists from the database of a particular show.
6. And we return that list List<MySeat> to the controller.
7. Controller GetListOfSeats will return the List<MySeat> to the angular.
8. Then User will see the seat layout of the particular show whichever he selected.
9. User will select the seats which he wants, click on the book.
10. Angular will send the list of selected seats to the spring boot controller selectSeat() method.
11. selectSeat() method will call the service layer, change the value of that selected seat from the database. Make isBooked to TRUE from FALSE.
12. And making the isSeatAvailable TRUE, seats selected by the user will be booked.
13. After change into the database, In the angular side , seats will be occupied and no other user can book the same ticket.

Problems faced and how we overcome them.

1.
    a. First difficulty we faced was when we had to put movie posters with the movie details in the movie cards present on the home page.
    b. We were planning to grab images from the database but we were facing a problem.
    c. Then we searched the ways of inserting the images in the database on the web then we came across the idea of putting the image address link in the database with which we can fetch later in the front end.

2.
    a. Then we faced another problem when we were working on the seat layout page.
    b. List of objects are coming from the Spring Boot , but the difficult task is to link those objects with the seat layout with row and column pattern.
    c. Then we convert that 1-D array into the 2-D array, and iterate them with the double for loop to achieve a realistic seat structure , so that the user can understand the position of the seat.
    d. And we also perform operations with the classes to for display seat is Available , Occupied or Selected.
    e. For step d, we again created one array , and linked them with seats, and displayed if seats are Available , Occupied or Selected.
    f. For solving the Seat layout structure page problem , we had to deal with array push and splice operations. And also deal with the CSS classes.

.

LEARNINGS DURING THE PROJECT

1. Because of this project, we learnt how to bind Angular, Spring boot and MySQL together.

2. We learnt how to work together as a team.

3. We learned to find the root cause of any bug and how to solve it.

4. Also learned how to use bootstrap css library to make our web app more elegant.

BLANK PAGE