Online Movie Ticket Booking

IIHT

Time To Complete: 2 hr

Contents

1 Problem Statement

Movie ticket booking system is SPA (Single Page Application) for booking tickets in selected locations.

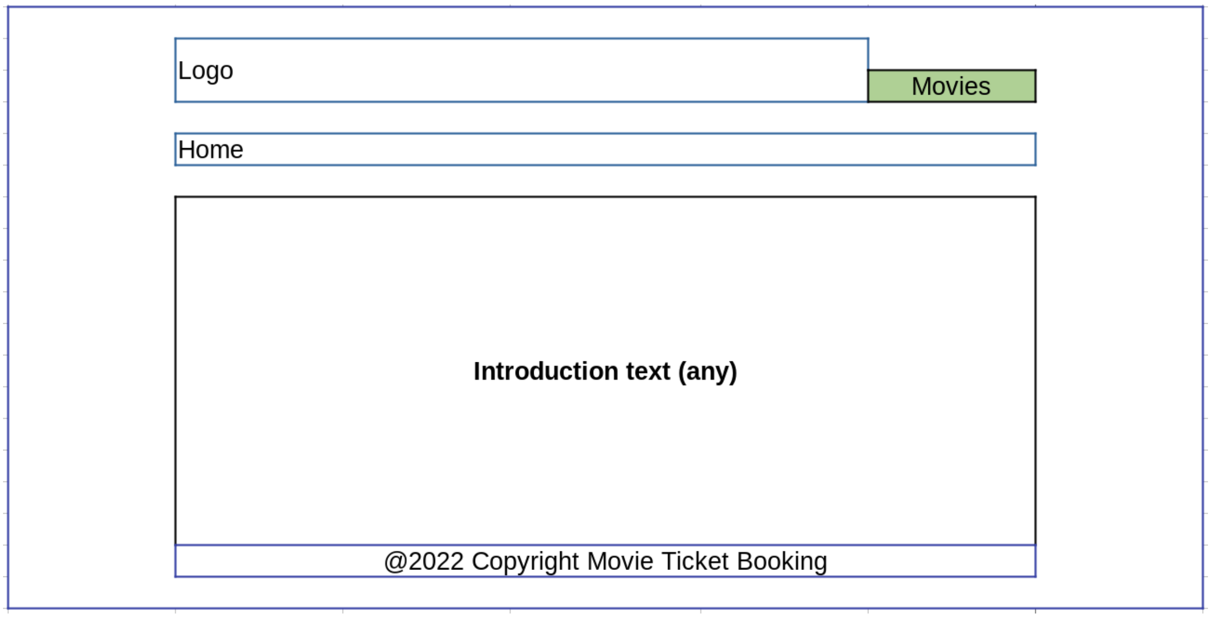
The core modules are:

1. Welcome Page
2. Select Movie
3. Select Show
4. Confirm Show

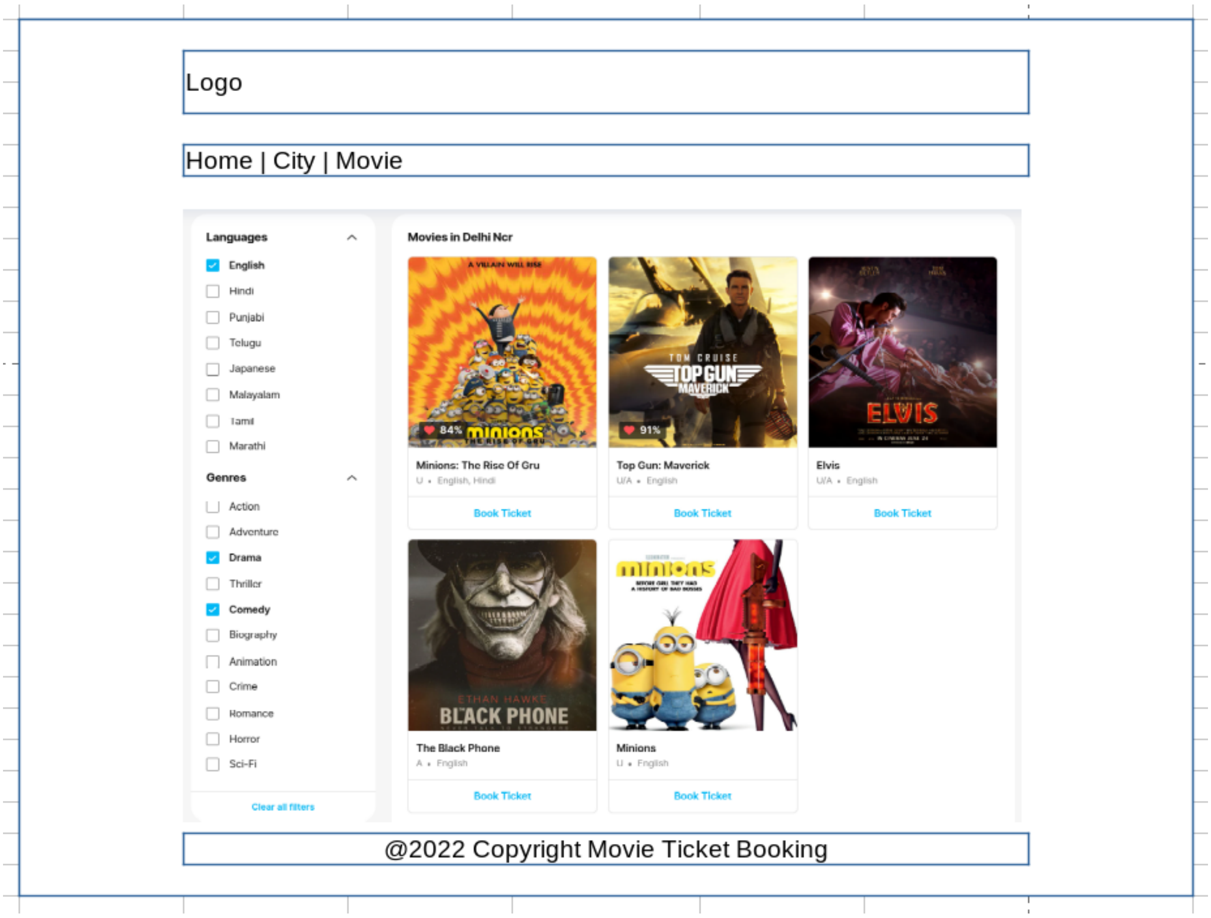
2 Proposed Wireframe

UI needs improvisation and modification as per given use case and to make test cases passed.

2.1 Welcome page



2.2 Select Movie



Logo

Home | City | Movie

Languages

- ☑ English
- □ Hindi
- □ Punjabi
- □ Telugu
- □ Japanese
- □ Malayalam
- □ Tamil
- □ Marathi

Genres

- □ Action
- □ Adventure
- ☑ Drama
- □ Thriller
- ☑ Comedy
- □ Biography
- □ Animation
- □ Crime
- □ Romance
- □ Horror
- □ Sci-Fi

Clear all filters

Movies in Delhi Ncr

Minions: The Rise Of Gru
U • English, Hindi
Book Ticket

Top Gun: Maverick
U/A • English
Book Ticket

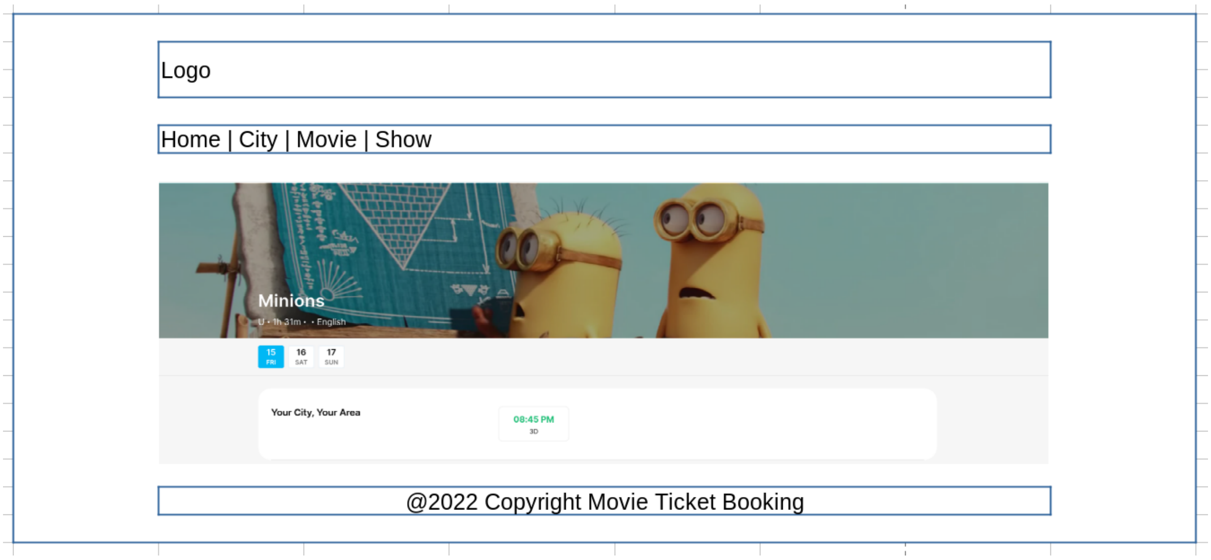
Elvis
U/A • English
Book Ticket

The Black Phone
A • English
Book Ticket

Minions
U • English
Book Ticket

@2022 Copyright Movie Ticket Booking

2.3 Select Show



Logo

Home | City | Movie | Show

Minions
U • 1h 31m • • English

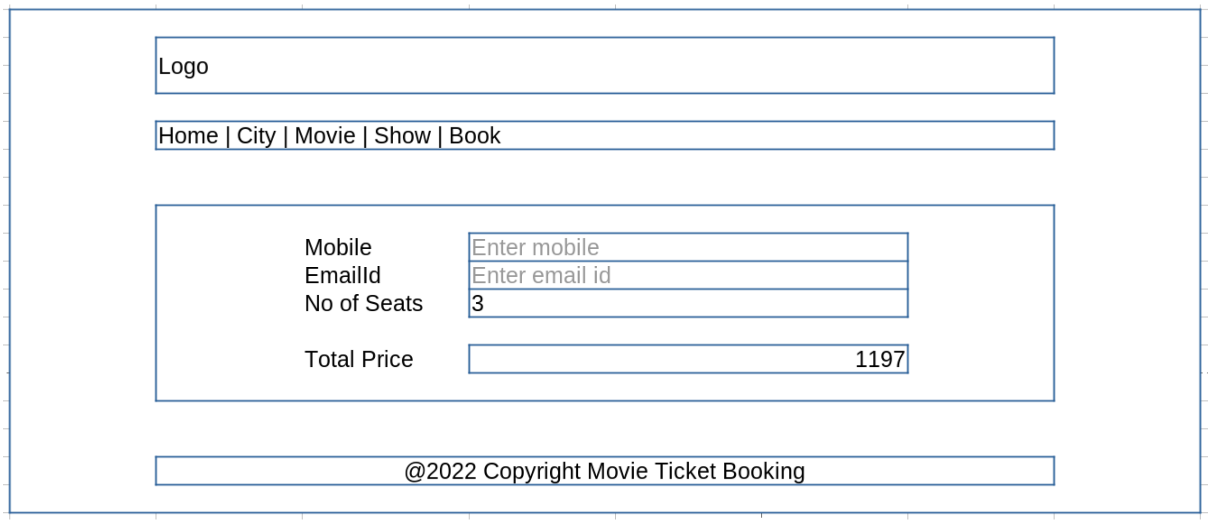
15 FRI | 16 SAT | 17 SUN

Your City, Your Area

08:45 PM
3D

@2022 Copyright Movie Ticket Booking

2.4 Book Show



Logo

Home | City | Movie | Show | Book

Mobile	Enter mobile
EmailId	Enter email id
No of Seats	3
Total Price	1197

@2022 Copyright Movie Ticket Booking

3 BUSINESS-REQUIREMENT:

As an application developer, develop the Movie Ticket Booking App (Single Page App) with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	As a user I should be able to visit the welcome page as default page. Acceptance criteria: 1. User can click Movies button given in menu bar. 2. User can click on logo to come on landing page from anywhere.
US_02	Select Movie	As a user I should be able to 1. View all the movies available in the selected city. 2. All Movie should be visible in card view with below details: ◆ Title ◆ Image ◆ Ratings ◆ Genre ◆ Length (minutes) ◆ Released Date ◆ Language ◆ Book button 3. Click on any movie to continue to book a ticket for the same movie. 4. A valid message should be visible if no movie is available in the selected city.
US_03	Select Show	As a user I should be able to 1. View all the shows available for the selected movie. 2. All the shows should contain below details: ◆ Show Time ◆ Continue button 3. Click on the continue button to select show and book tickets for the same show. 4. A valid message should be visible if no show is available in the selected movie.
US_04	Book Show	As a user I should be able to 1. Insert below details: ◆ Mobile number (required) ◆ Emailid (required) ◆ No of seats (required, minimum should be 1) ◆ Price per seat (auto selected and not editable) ◆ Total price (auto calculated and not editable)

US_05	Confirm Booking	2. Click on the Book button to confirm booking. As a user I should be able to 1. see all the movie ticket details before saved in db via rest api. 2. Confirm ticket to book. 1. On successful booking show message - **"Your booking is confirmed, Please be on time"** 2. On booking failure show message - **"Your booking failed, please try again"** 3. POST booked ticket to fake rest api.

4 Constriants

1. On the book show page load, input focus must come to the first input field.
1. You should be able to press the “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.
1. Cities, movies details must be fetched via fake-rest API from movie-ticket.json.
1. On click of “Book” button, user details must be saved via fake-rest API in movie-ticket.json.
1. Fake rest api is implemented with json-server.

Example JSON for reference of fields to be used:

movie-db.json

```
{
        “cities”: [
            { "id": "1", "name": "Mumbai", "state": "Maharashtra" },
            { "id": "2", "name": "Delhi", "state": "Delhi" },
            { "id": "3", "name": "Bengaluru", "state": "Karnataka" }
        ],
        “movies”: [
            {
                    "id": 1,
                    "title": "Superman is flying",
                    "image":
        "https://i.picsum.photos/id/650/300/200.jpg?hmac=nuRnzrByBYKU6fsTl1xwz3MIpNrNjb
        0PjLJA5ypiOqU",
                    "ratings": 3.4,
                    "genre": "fantasy",
                    "length": 119,
                "releasedDate": "28/12/2012",
                "language": "English",
                "showTime": "21:00",
                “price”: 300
            }
        ],
        “tickets”: [
            { “id”: 1, “mobile”: “9898989898”, “email”: “demo@gmail.com”, “seats”: 4,
        “movieId”: 1, “paid”: 1200 }
        ]
}
```

5 Mandatory Assessment Guidelines

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to**
   Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.
3. **This editor Auto Saves the code.**
4. **If the you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**

6. **You can follow series of command to setup Angular environment once you are in your project-name folder:**
   a. **npm install -> Will install all dependencies -> takes 10 to 15 min**

b. npm run start -> To compile and deploy the project in browser. You can press <Ctrl> key while clicking on localhost:4200 to open project in browser -> takes 2 to 3 min

c. npm run json-server -> to deploy fake rest api created with json-server -> takes 10 to 15 seconds

d. npm run jest -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min

7. You may also run "npm run jest" while developing the solution to re-factor the code to pass the test-cases.
8. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.
9. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.
10. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.
11. To test any UI based application the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.