Movie Ticket

IIHT

Time To Complete: 10 to 12 hr

CONTENTS

1 Problem Statement

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

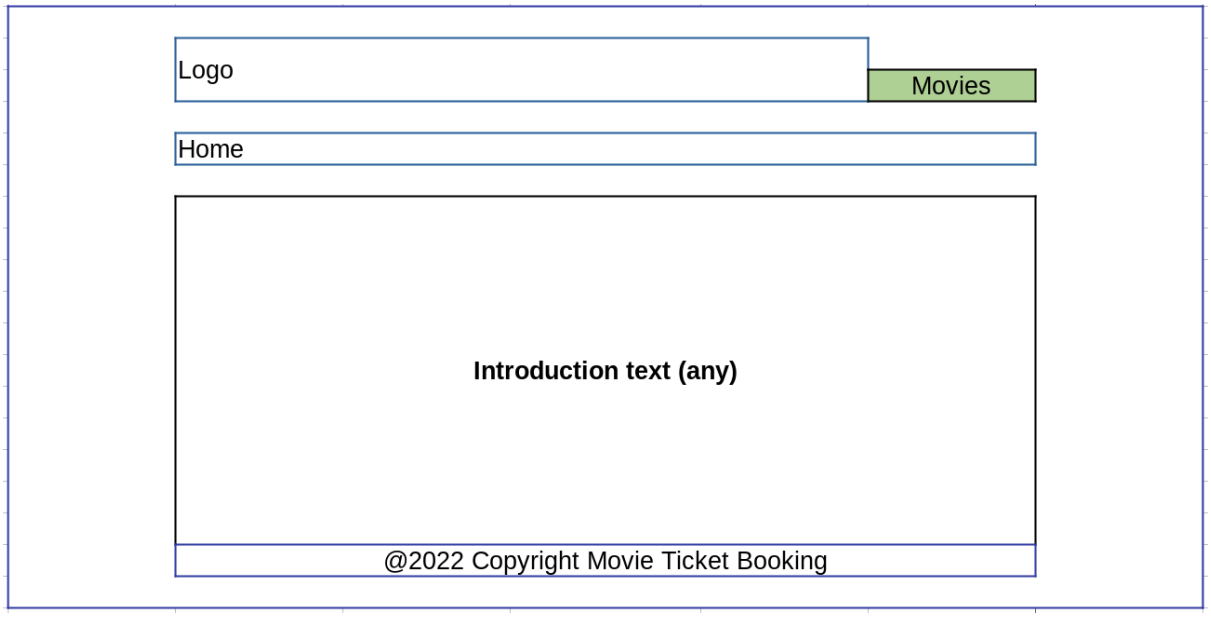
The core modules are:

1. Welcome Page
2. Select Movie
3. Book Show

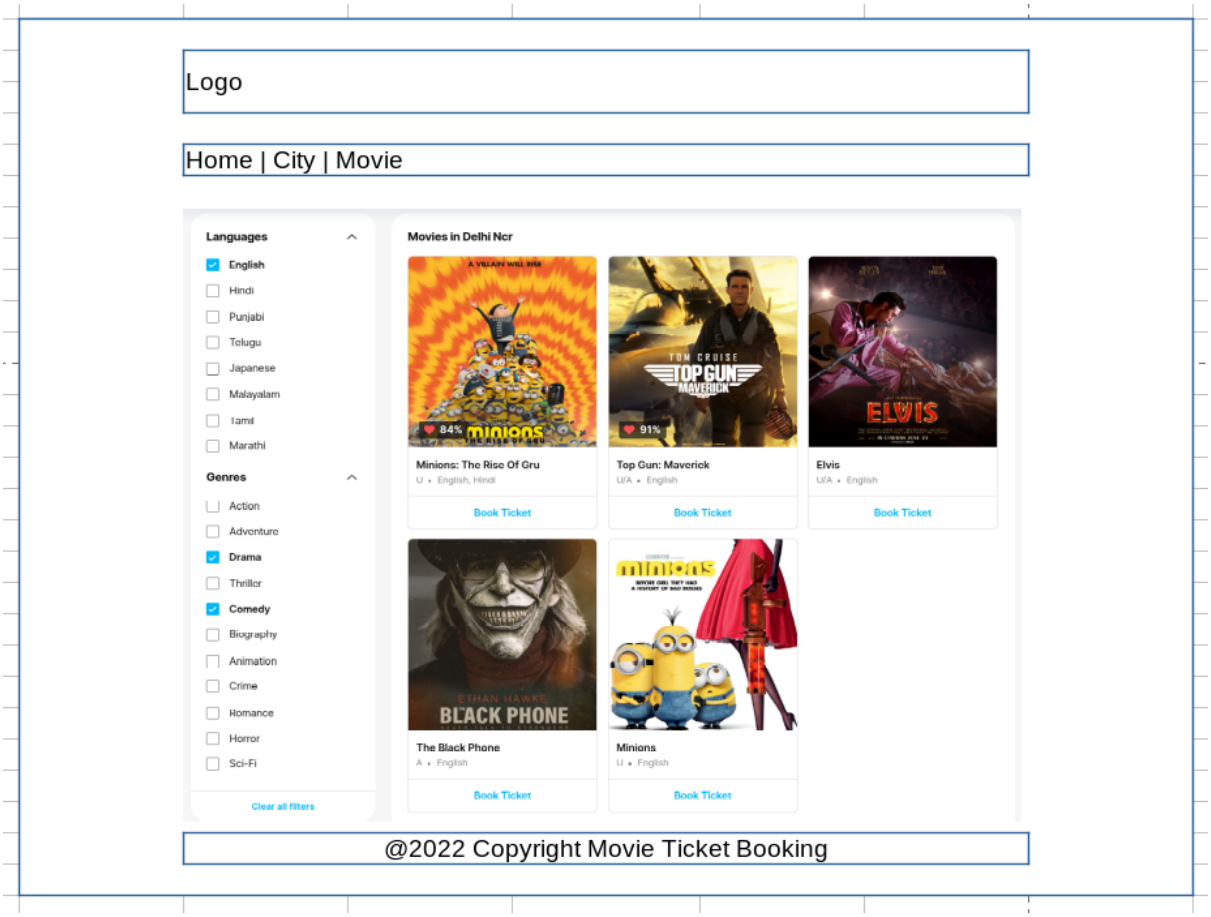
2 Proposed Wireframe

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

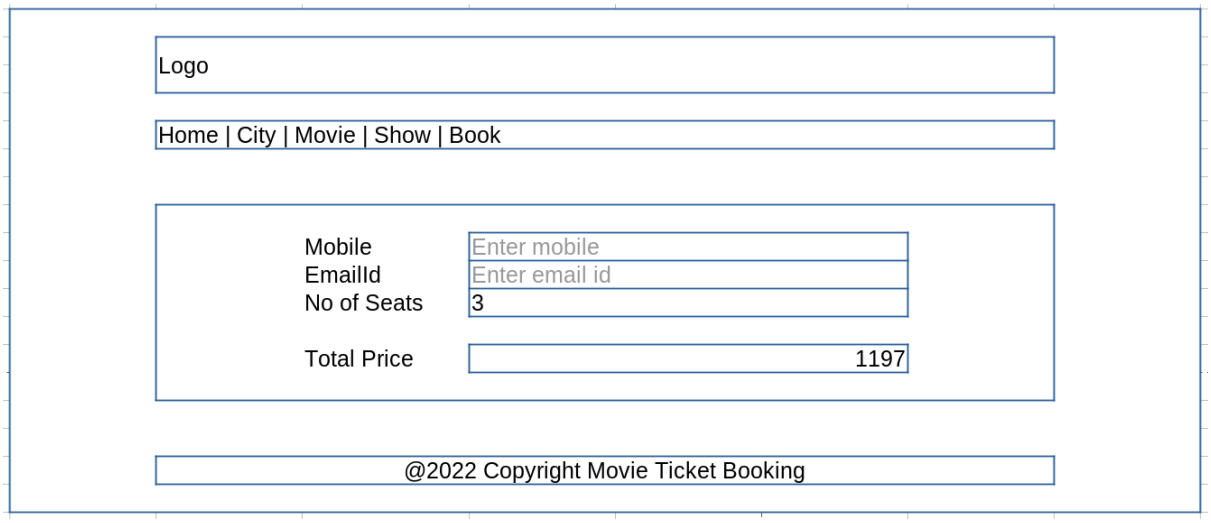
2.1 Welcome page



2.2 Select Movie



2.3 Book Show



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 any where.
US_02	Select Movie	As a user I should be able to 1. View all the movies available in selected city. 2. All Movie should contain below details: ◆ Title ◆ Image ◆ Ratings ◆ Genre ◆ Lenght (minutes) ◆ ReleasedDate ◆ Language ◆ Book button 3. Click on book button to book ticket for same movie. 4. A valild message should be visible if no movie is available in selected city.
US_03	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) 2. Click on Book button to confirm booking. 3. On successful booking, show a confirmation message - ”`Your booking is confirmed`” 4. On booking failure, show error message - “`Your ticket is not booked`”

4 CONSTRAINTS

1. Create header component for logo and menu.
2. Create Footer component with `content as shown in wireframes.`
3. `Create Breadcrumb component to be used in each page to`
    1. provide links back to each previous page the user navigated through
    2. show the user's current location
4. On the book show page load, input focus must come to first input field.
5. You should be able to press “TAB” key and “SHIFT + TAB” to navigate from top field to bottom field and vice-versa.
6. Cities, movies details must be fetched via fake-rest API from movie-ticket.json.
7. On click of “Book” button, user details must be saved via fake-rest API in movie-ticket.json.
8. Fake rest api is implemented with json-server.

Example JSON for reference of fields to be used:

movie-db.json


```
{
    “movies”: [
        {
            "id": 1,
            "title": "Superman is flying",
            "image": "url",
            "ratings": 3.4,
            "genre": "fantasy",
            "lenght": 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. **Open the test link received.**
2. **Use email, first name and last name to login to assessment. In case you have to login again, you must use the same email, first name and last name.**
3. **Click on "Start Assessment"**
4. **Click on "Open IDE". Do not click on "Submit Workspace" until you want to make the submission.**
5. **IDE will open in a new tab.**
6. **It will take 3 min for IDE to load. It is advisable to wait for 3 min.**
7. **Once IDE is loaded, you can open the terminal (nodejs) by clicking Terminal -> Open Terminal In Specific Container -> Nodejs terminal**
8. **Change to your project by using command "cd <ProjectName>". You can type the first 2 characters of the project-name and press the TAB key to autofill the project-name.**
9. **<ProjectName> will be mix of "FirstNameLastName-Unique-Hash-Code"**
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. **This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.**
    Note: The application will not run in the local browser
12. **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**

13. **You may also run "npm run jest" while developing the solution to re-factor the code to pass the test-cases.**
14. **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 The Workspace" after you are done with code.**
15. **You may close all windows, once the workspace is submitted!**