Online Movie Ticket Booking

CONTENTS

1 PROBLEM STATEMENT

Movie ticket booking system is SPA (Single Page Application) for booking tickets in selected locations.
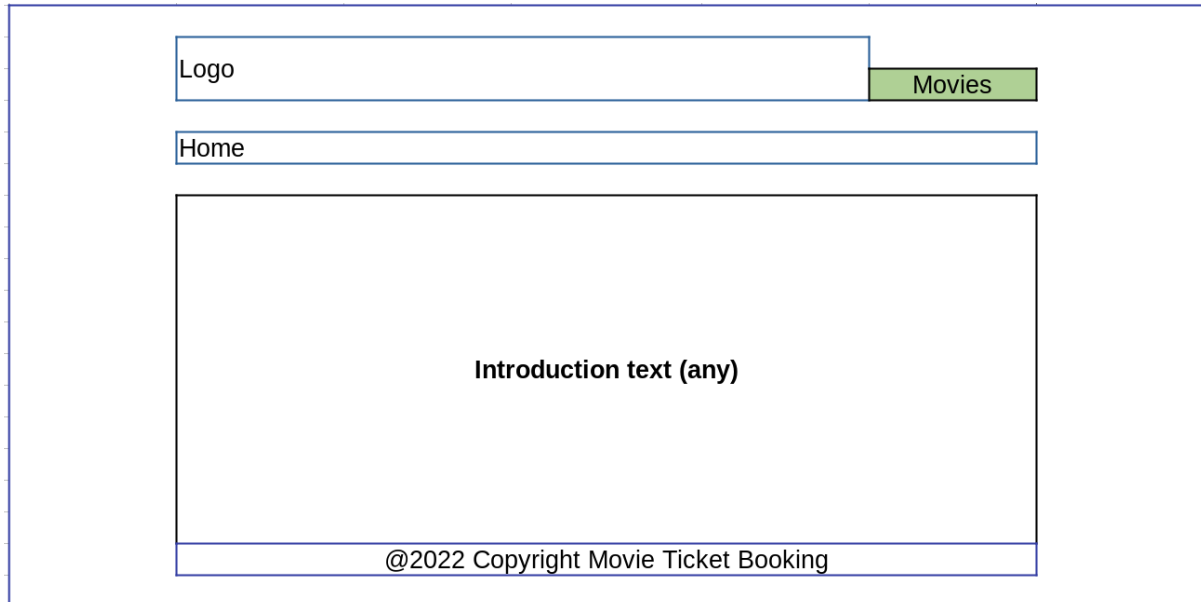
The core modules are:

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

Movies

Superman is flying

Ratings: 3.4
Genre fantasy
Length 119min
ReleasedDate 28/12/2012
Language Telugu
ShowTime 21:00
Price 300

Batman is watching

Ratings: 3.9
Genre horror
Length 124min
ReleasedDate 14/7/2014
Language English
ShowTime 15:00
Price 370

2.3 BOOK SHOW

Book Show

Details

Phone

Email

Seat
1
Price (Per Seat)
370
Price (Total)
370
Submit

3 BUSINESS-REQUIREMENT:

As an application developer, develop the Movie Ticket Booking 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. 2. All Movies should contain below details: ◆ Title ◆ Image ◆ Ratings ◆ Genre ◆ Length (minutes) ◆ Released Date ◆ Language 3. Click on movie image to book ticket for same movie. 4. A valild message should be visible if no movie is available.
US_03	Book Show	As a user I should be able to 1. Insert below details: ◆ Mobile number (required) ◆ Email-ID (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 Submit 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 CardComponent as the child component to show each movie.
2. Movies should be visible as card view. (Do not use table)
3. Movie should be visible only If "available" property value is true
4. Without selecting movie, book show page should not be visible.
5. On the book show page load, input focus must come to first input field.
6. You should be able to press "TAB" key and "SHIFT + TAB" to navigate from top field to bottom field and vice-versa.
7. Movies details must be fetched from /src/assets/movie-db.js file.
8. On successful booking, show a confirmation message - "`Your booking is confirmed`".
9. **On successful booking, booking information must get saved in /data-source/movie-db.json using axios 'POST' call.**
   Following fields must be stored for booking info:
   1. **Email**
   2. **Phone**
   3. **Seat**
   4. **Price**
   5. **totalprice**
10. **For above purpose a fake-json server is configured.**
11. On booking failure, show error message - "`Your ticket is not booked`"

Example JSON for reference of fields to be used:

```
/src/assets/movie-db.js
export const db = {
 languages: [
   { name: 'hindi', checked: true },
   { name: 'english', checked: true },
   { name: 'telugu', checked: true },
   { name: 'punjabi', checked: true }
 ],
 genres: [
   { name: 'action', checked: true },
   { name: 'fantasy', checked: true },
   { name: 'romance', checked: true },
   { name: 'horror', checked: true }
 ],
 movies: [
  {
    id: 1,
    title: 'Superman is flying',
    image:
```

```
            'https://i.picsum.photos/id/650/300/200.jpg?hmac=nuRnzrByBYKU6fsTl1xwz3MIpNrNjb0
            PjLJA5ypiOqU',
        ratings: 3.4,
        genre: 'fantasy',
        length: 119,
        releasedDate: '28/12/2012',
        language: 'Telugu',
        showTime: '21:00',
        price: 300,
        available: true
    },
    {
        id: 2,
        title: 'Batman is watching',
        image:

            'https://i.picsum.photos/id/650/300/200.jpg?hmac=nuRnzrByBYKU6fsTl1xwz3MIpNrNjb0
            PjLJA5ypiOqU',
        ratings: 3.9,
        genre: 'horror',
        length: 124,
        releasedDate: '14/7/2014',
        language: 'English',
        showTime: '15:00',
        price: 370,
        available: true
    }
  ]
}
```

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 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. 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.

7. 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

8. You can follow series of command to setup React 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:8080/8081 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 test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min

9. You may also run "npm run jest" while developing the solution to re-factor the code to pass the test-cases.

10. 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.

11. 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.