

# Online Movie Listing App

IIHT

Time To Complete: 2 hr

## CONTENTS

---

1 Problem Statement	2
2 Proposed Wireframe	3
2.1 View All Movies	3
3 Business-Requirement:	4
4 Constraints	4
5 Mandatory Assessment Guidelines	5

# 1 PROBLEM STATEMENT

---

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

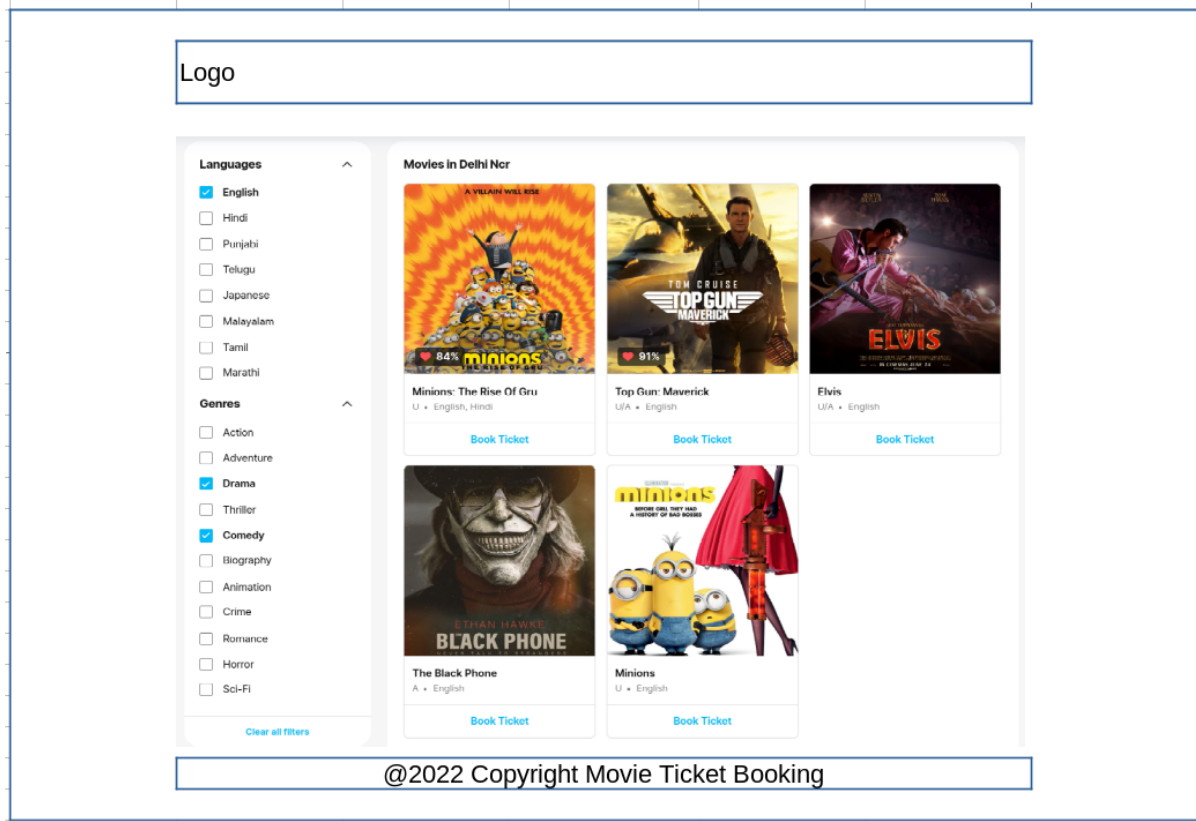
The core modules are:

1. Welcome Page
2. View Movies

## 2 PROPOSED WIREFRAME

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

### 2.1 VIEW ALL MOVIES



### 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	View All Movies	<p>As a user I should be able to</p> <ol style="list-style-type: none"><li>1. View all the movies.</li><li>2. Filter movies based on language and genre (as given in the wireframe)</li><li>3. All Movie should contain below details:<ul style="list-style-type: none"><li>◆ Title</li><li>◆ Image</li><li>◆ Ratings</li><li>◆ Genre</li><li>◆ Length (minutes)</li><li>◆ Released Date</li><li>◆ Language</li><li>◆ ShowTime</li><li>◆ Price</li><li>◆ Available</li></ul></li></ol>

### 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. Create component checkbox for showing Languages & Genres.
4. Movie should be visible only If "available" property value is true
5. Movies data must be fetched/imported from data-source/movie-db.js file.
6. Add more records from your end for exhaustive testing.

Example JSON for reference of fields to be used:

data-source/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=nuRnZrByBYKU6fsTl1xwz3MIpNrNjb0PjLJA5ypi0qU',
      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=nuRnZrByBYKU6fsTl1xwz3MIpNrNjb0PjLJA5ypi0qU',
    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. 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.
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 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.