

## Online Movie Ticket Booking

IIHT

Time To Complete: 2 hr

## CONTENTS

---

1 Problem Statement	2
2 Proposed Wireframe	3
2.1 Welcome page	3
2.2 Select Movie	4
2.3 Book Show	4
3 Business-Requirement:	5
4 Constraints	6
5 Mandatory Assessment Guidelines	7

# 1 PROBLEM STATEMENT

---

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

The core modules are:

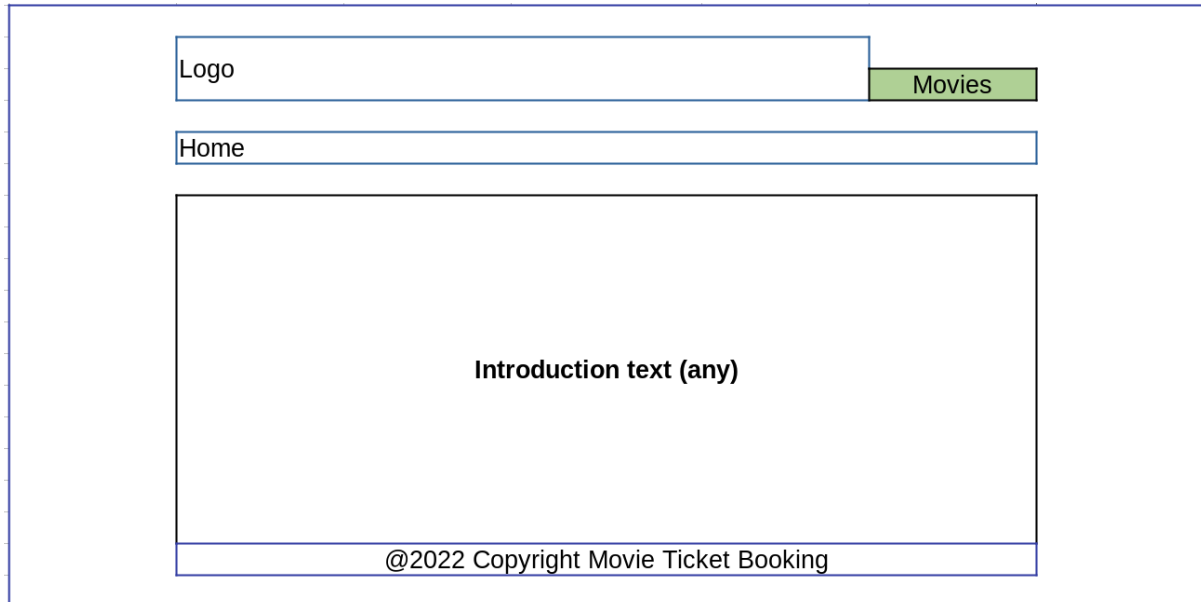
1. Welcome Page
2. Select Movie
3. Book Show

## 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

Logo

Home | City | Movie

Languages


☒ English  
☐ Hindi  
☐ Punjabi  
☐ Telugu  
☐ Japanese  
☐ Malayalam  
☐ Tamil  
☐ Marathi

Genres


☐ Action  
☐ Adventure  
☒ Drama  
☒ 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 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 with below guidelines:

User Story #	User Story Name	User Story
US_01	Welcome Page	<p>As a user I should be able to visit the welcome page as default page.</p> <p>Acceptance criteria:</p> <ol style="list-style-type: none"><li>1. User can click Movies button given in menu bar.</li><li>2. User can click on logo to come on landing page from anywhere.</li></ol>
US_02	Select Movie	<p>As a user I should be able to</p> <ol style="list-style-type: none"><li>1. View all the movies available.</li><li>2. All Movies 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>◆ Continue button</li></ul></li><li>3. Click on Continue button to book ticket for same movie.</li><li>4. A valid message should be visible if no movie is available.</li></ol>
US_03	Book Show	<p>As a user I should be able to</p> <ol style="list-style-type: none"><li>1. Insert below details:<ul style="list-style-type: none"><li>◆ Mobile number (required)</li><li>◆ Email-ID (required)</li><li>◆ No of seats (required, minimum should be 1)</li><li>◆ Price per seat (auto selected and not editable)</li><li>◆ Total price (auto calculated and not editable)</li></ul></li><li>2. Click on Book button to confirm booking.</li><li>3. On successful booking, show a confirmation message - <b>"Your booking is confirmed"</b></li><li>4. On booking failure, show error message - <b>"Your ticket is not booked"</b></li></ol>

## 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. 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.ts file.
8. On booking failure, a message should be visible on console – "Booking failed"
9. On successful booking, show ticket details on a same page. Ticket details must contain
  1. A valid email id
  2. a valid mobile number
  3. total number of seats
  4. total price

Example JSON for reference of fields to be used:

```
/src/assets/movie-db.ts  
export const db = {  
  "movies": [  
    {  
      "id": 1,  
      "title": "Superman is flying",  
      "image": "url",  
      "ratings": 3.4,  
      "genre": "fantasy",  
      "length": 119,  
      "releasedDate": "2012-12-28",  
      "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 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 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**
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.