**Capstone Project- Flight Booking**

**Problem Statement**

Infy Airlines wants to automate the process of flight ticket booking and management. The following functionalities are required to be implemented.

1. Book flight tickets and update the number of seats for the same flight
2. View and Delete flight bookings

Project contains two subfolders named **FB\_UI** and **FBwebservice.**

* **FBwebservice** folder contains the **implemented** server-side code
* **FB\_UI**folder contains the **client-side code** which **needs to be implemented**

Follow the steps mentioned in the software and project setup section below before starting to implement the solution.

**List of Software Required:**

* Node JS
* Angular
* Visual Studio Code IDE

Note: Follow the steps mentioned at the beginning of the course to install the required software.

**Project Implementation:**

**Frontend Setup:**

1. Use **npm install** command to download all the required packages.

2. Start the angular application using **ng serve**.

**Backend Setup:**

1. Use **npm run server** to start the json-server at 1020 which will act as the backend for this application.

NB: Hit all these commands on the project folder where we have src folder, package.json etc. Please ensure to keep both FE and BE running while in use.

**Implementations to be done:**

**FB\_UI** contains the following artifacts:

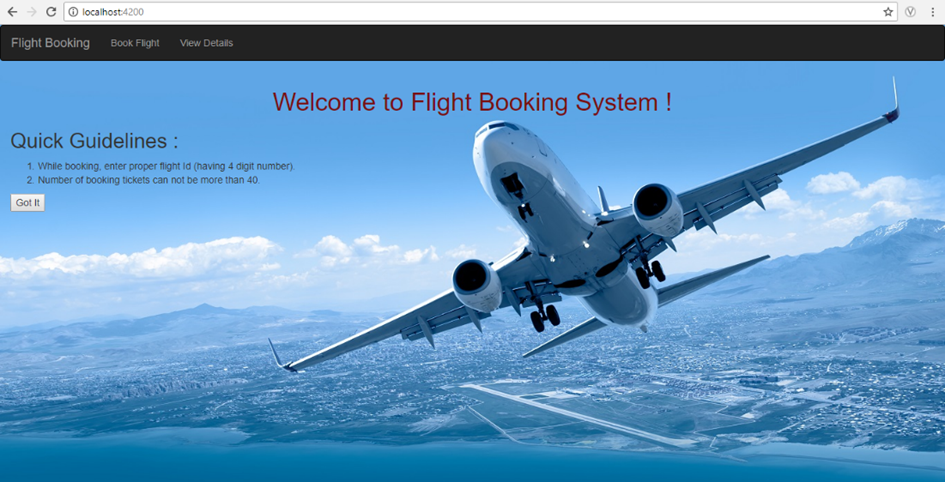
| **File Name** | **Description** |
| --- | --- |
| app/book-flight/book-flight.service.ts | **Partially Implemented** |
| app/app-routing.module.ts |
| app/book-flight/book-flight.component.ts |
| app/book-flight/book-flight.component.html |
| app/book-flight/view-details.component.ts |
| app/book-flight/view-details.component.html |
| app/book-flight/view-details.service.ts |
| app/app.component.html |
| app/shared/Flight.ts | **Fully Implemented** |
| app/shared/FlightBooking.ts |
| app/app.module.ts |

**Routing:**

|  |  |
| --- | --- |
| **Route Path** | **Component** |
| /book | BookFlightComponent |
| /view | ViewDetailsComponent |
| /\* | BookFlightComponent |

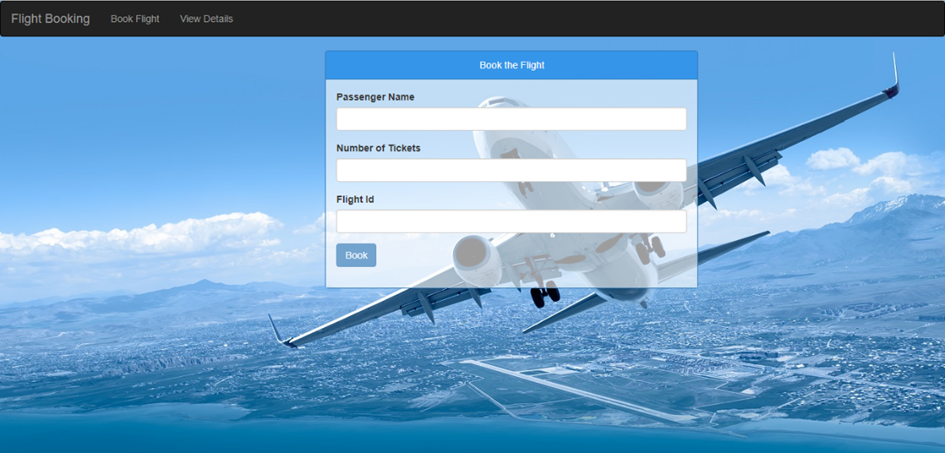
**Home Page:**

This must be coded in **AppComponent**. The below view must be displayed:



* The navigational bar should have three links that will redirect to appropriate components, as a Single Page Application without refresh.
* On click of the "Got it" Button, the 'Quick Guidelines' should disappear and display the Book Flight view as shown below:

**Book Flight:** This must be coded in **BookFlightComponent**.



**Flight Booking Component:**

**Properties:**

* errorMessage:string
* successMessage:string
* flights:Flights[]
* bookingForm:FormGroup

The form control names under the form group are:

* passengerName
* noOfTickets
* flightId

**Validations:**

* passengerName – required
* noOfTickets – required and minimum 1
* flightId – required. It must be 7 characters long. The first three characters must be Capital letters, followed by a hyphen and followed by three numbers. Ex: IND-105. Use custom validator ‘validateFlight’.

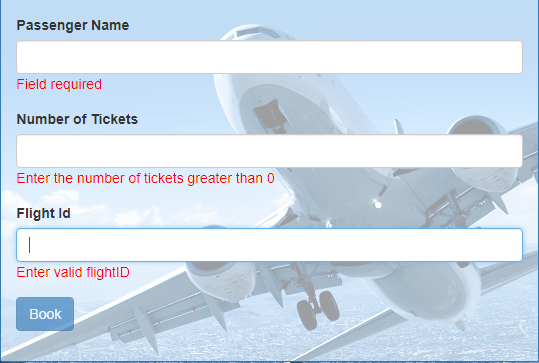
**Methods:**

**​​​​​​​ngOnInIt():**On load of the component, getAllFlights() ofbook-flight.service.ts must be invoked and handle the success and error messages. All the flight details must be stored in flights property.

**book():**Must be invoked when the Book Flight button is clicked and necessary operations must be performed. Below conditions needs to be checked and if failed, appropriate error messages should be displayed.

1. Check if flight to be booked is available. If not, display error “Flight unavailable!!”
2. Check if flight status is cancelled. If yes, display error “Flight cancelled!!”
3. Check if required number of seats are available. If not, display error "Requested number of seats unavailable"
4. Once above conditions are checked, calculate the ticket fare and invoke bookFlight() of book-flight.service.ts to book the flight. Handle the success and error messages.
5. Once flight is booked successfully, update the available seats for the flight by invoking updateFlight() of book-flight.service.ts.

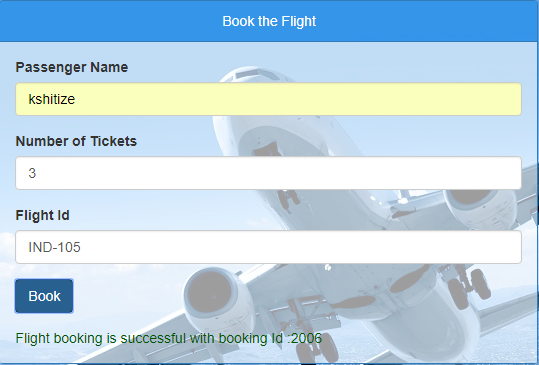
**Note:** If any form validation fails, then display error messages as shown below:



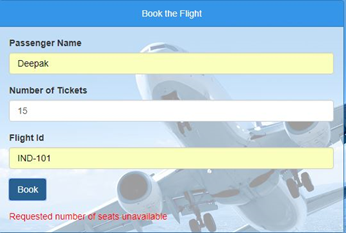
**Note:**

* All validation messages must be displayed if the original data of the field was modified.
* The span tag displaying the error message should have the ID as <formcontrolname>error

If everything is right, display the success message as shown :



If any error occurs display the error message from backend services as shown below:



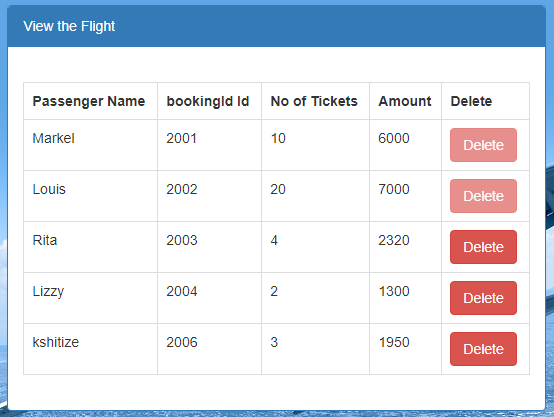
**book-flight.service.ts:**

**getAllFlights()** – must send a GET request to http://localhost:1020/flights

**bookFlight()** – must send a POST request to http://localhost:1020/bookings

**updateFlight()** – must send a PATCH request to http://localhost:1020/flights/:flightId

**View Details**: The View Details page should look like below. Implement this in **ViewDetailsComponent**



* It should display all the bookings done by the user
* The delete button should be enabled only for bookings whose total amount is less than 5000

**view-details.component.ts:**

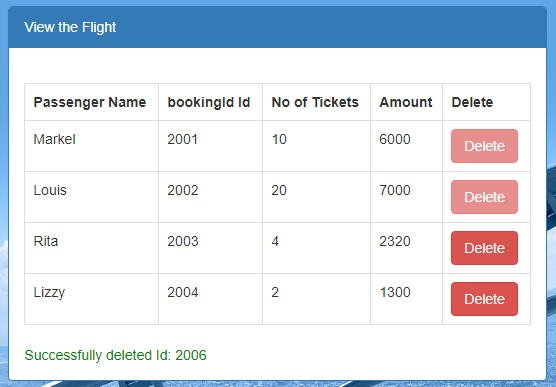
**Properties:**

* flightDetails : FlightBooking[];
* successMessage:string;
* errorMessage:string;

On page load, display the booking details by contacting the view() of view-details.service.ts. Display error messages if any.

**Methods:**

**delete(id):**Delete the booking detail by invoking delete() of view-details.service.ts and passing the **id**. Also, the booking detail must be removed from the table on the page and a success message should be displayed as shown below:



**view-details.service.ts:**

**view()**-  Performs a get request to the URL **:**http://localhost:1020/bookings

**delete()** – Performs a delete request to the URL http://localhost:1020/bookings/:id

**Instructions for Evaluation**

* Refer to the below rubric for getting the project evaluated by your SMEs from project or account

**Assessment Rubric:**

| **Component** | **Marks** |
| --- | --- |
| book-flight.component.ts | 9 |
| book-flight.component.html | 10 |
| book-flight.service.ts | 5 |
| view-flight.component.ts | 9 |
| view-flight.component.html | 8 |
| view-flight.service.ts | 5 |
| app-routing.module.ts | 4 |