# Angular 201

## **Airline App**

**M1050919**

Tezi

**Readme:**

**Data is hosted on JSON server:**

A data file db.json has been included in the src folder to host on the server.

You just have to run **json-server -p 5000 --watch db.json**

90% I have used Angular Material so before running the app just run

**ng add @angular/material**

**run npm i to install node modules**

now you can run **ng serve**

A Build Airline Check-In, In-Flight and ancillary management app which comprises two sections

1. Airline Staff
2. Admin

Tech used:

1. Angular (Typescript)
2. HTML, CSS, SCSS
3. JSON Server for backend

The features are based on role-based authorizations and demonstrated as following below:

The roles would be based on the authorization as mentioned before. After Signing in, the user would be directed to the role-based section depending upon the credentials.

Facebook Social Login is used as third-party authorization for the users(so performance might fluctuate)

**Airline-Staff:**

After successful signing in, user will be directed to airline staff dashboard.

There are two options check-in and In-flight.

When the user checks-in to the check-in section, the seats maps are displayed for passengers with the color coded according to the passenger data such as booked, wheelchair, infants etc.

User (staff) can filter passengers based on check-in or not, infant, and wheelchair. Also, staff can change the check-in status of the passenger along with their seats by using change status and change seat.

When the staff navigates to the in-flight section, the seat maps with color coded are displayed for the passengers differentiated based on special meal required or not. The staff can add services, shopping requests, and change the meal preference for the passengers.

**Admin:**

When the user signs in using an admin credentials (I have checked in sign in services file), the app will be directed to an admin dashboard comprising flight details and its respective passengers.

In Admin you will have two options manage passengers and manage ancillary services.

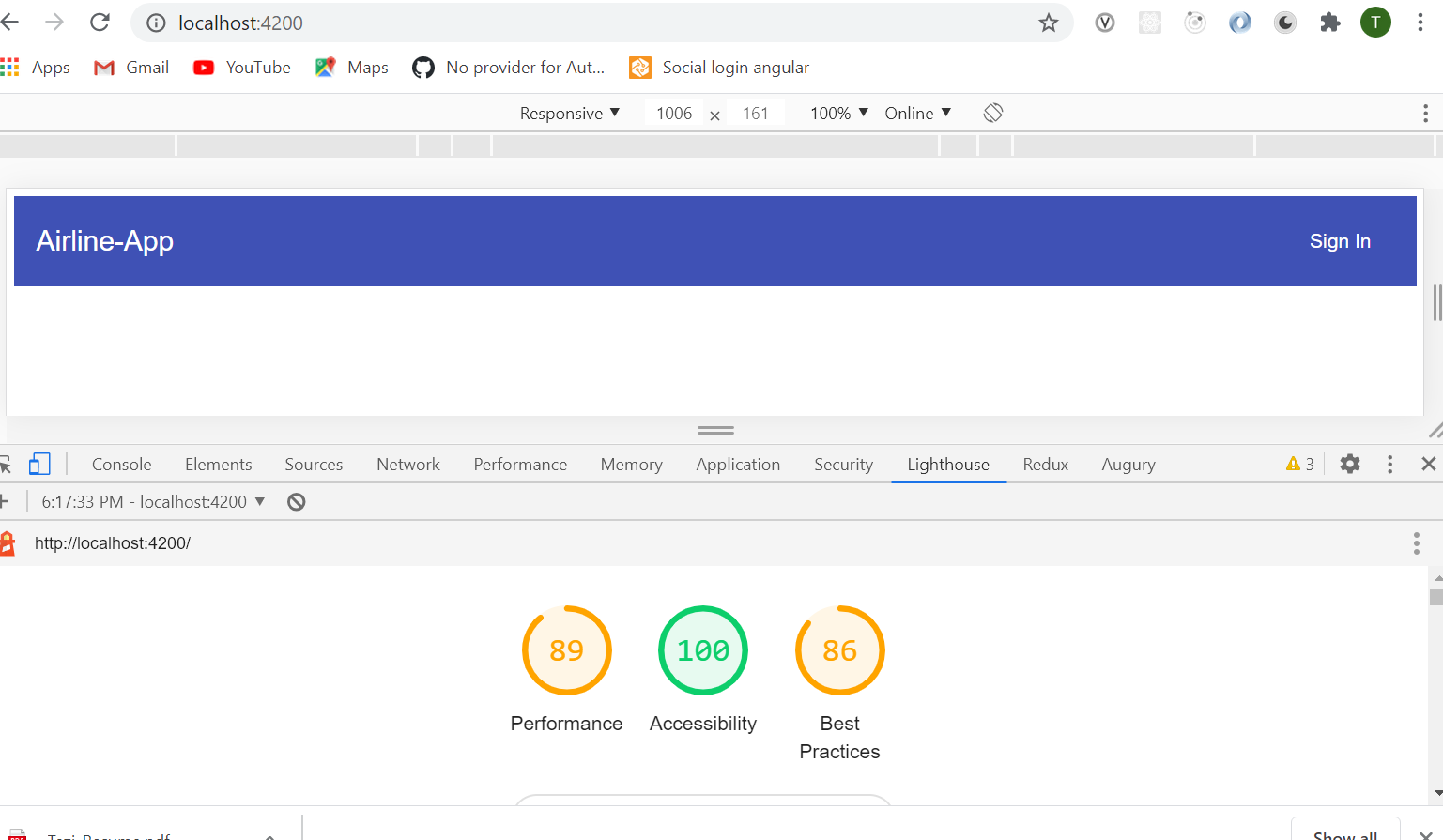
If user(admin) chooses manage passenger then Admin can add/edit passengers, filter passengers based on missing defined mandatory requirements like passport, address etc.

If user(admin) chooses manage ancillary services then Admin can manage services as well like ancillary services, shopping items and special meals per flight.

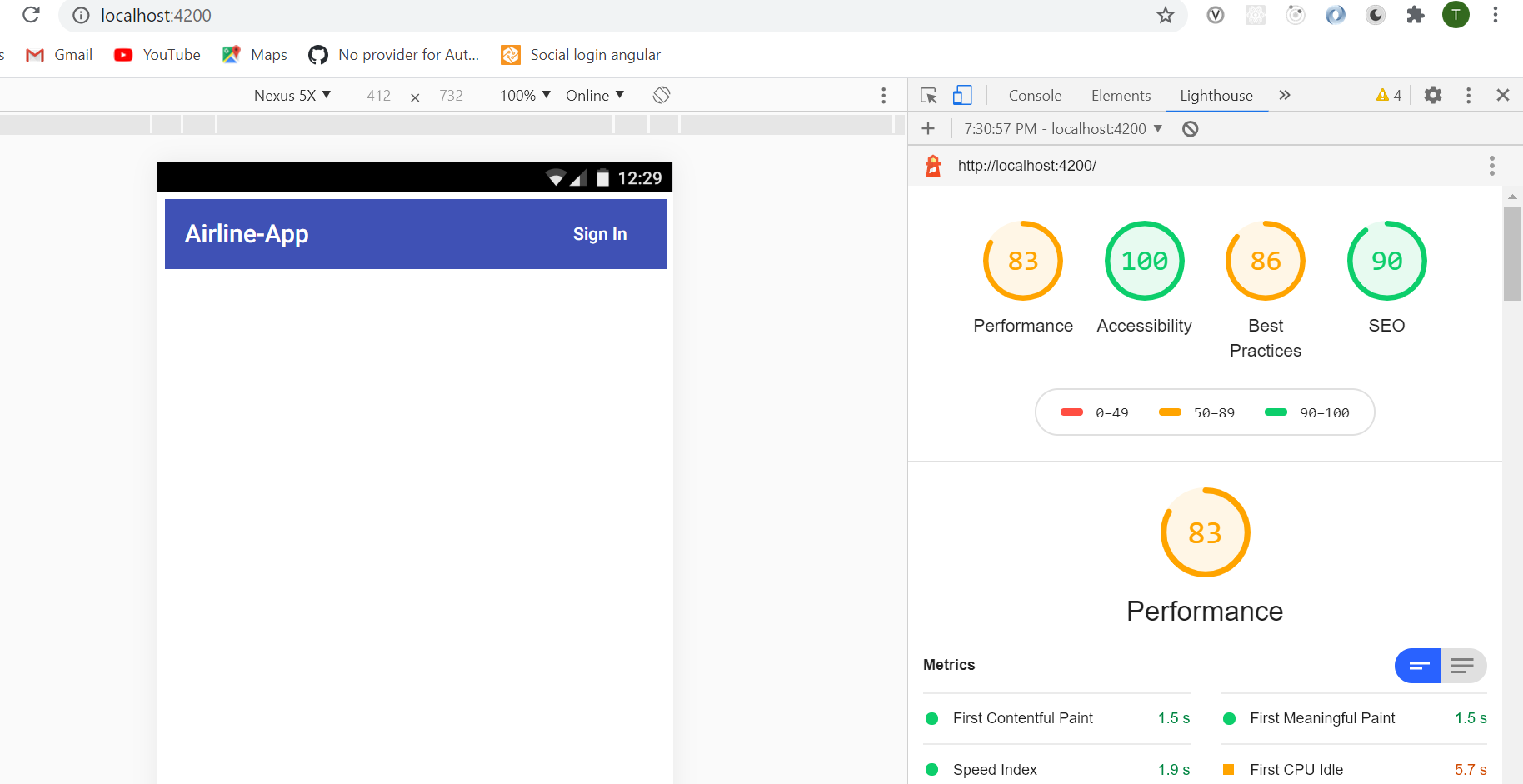
As mentioned before, admin can add/update and delete various services per flight.

**Light house report:**

Below is the screenshot of performance report of the web app Desktop

****

For Mobile View

**