



Travel Thru Air

Soumalya Kundu

Roll: 001810501033

Overview

In this assignment we need to implement a flight booking website namely "Travel thru air" using servlets as backend.

Goals:

- User can book flights on a specified date and time
- User should search flights based on time and dates
- Rest api for seamless integration between frontend and backend
- Servlets in the backend and a separate database for storing persistent informations
- Integrate and follow MVC pattern

Design Pattern

MVC pattern has been followed throughout the application and the model classes are nothing but plain old java classes. I use Vanilla javascript with JQuery for designing the frontend.

Technologies Used

Frontend: html, css, vanilla js, jquery, axios

Backend: servlets, jdbc, Gson

Database: Postgresql

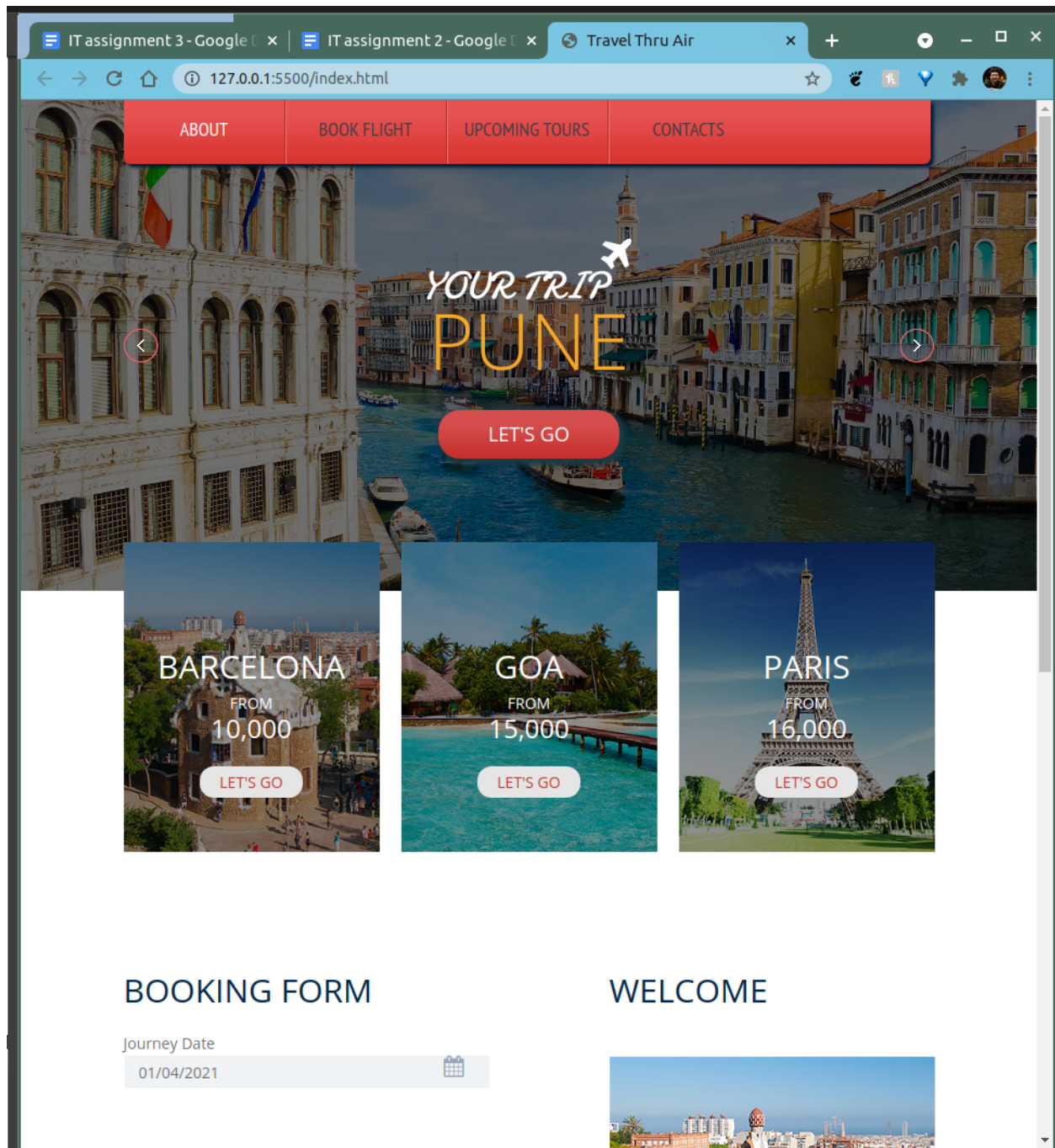
Integrations: REST api

Implementations

The home page of the application displays a list of special offers provided by the marketing department. For each offer, the flight company, departure city, arrival city, departure time, original cost, discounted cost and validity time is shown. The offer is valid only for a specified duration, after which it disappears from the home page. The GSON library is used to convert Java objects to JSON objects and vice versa.

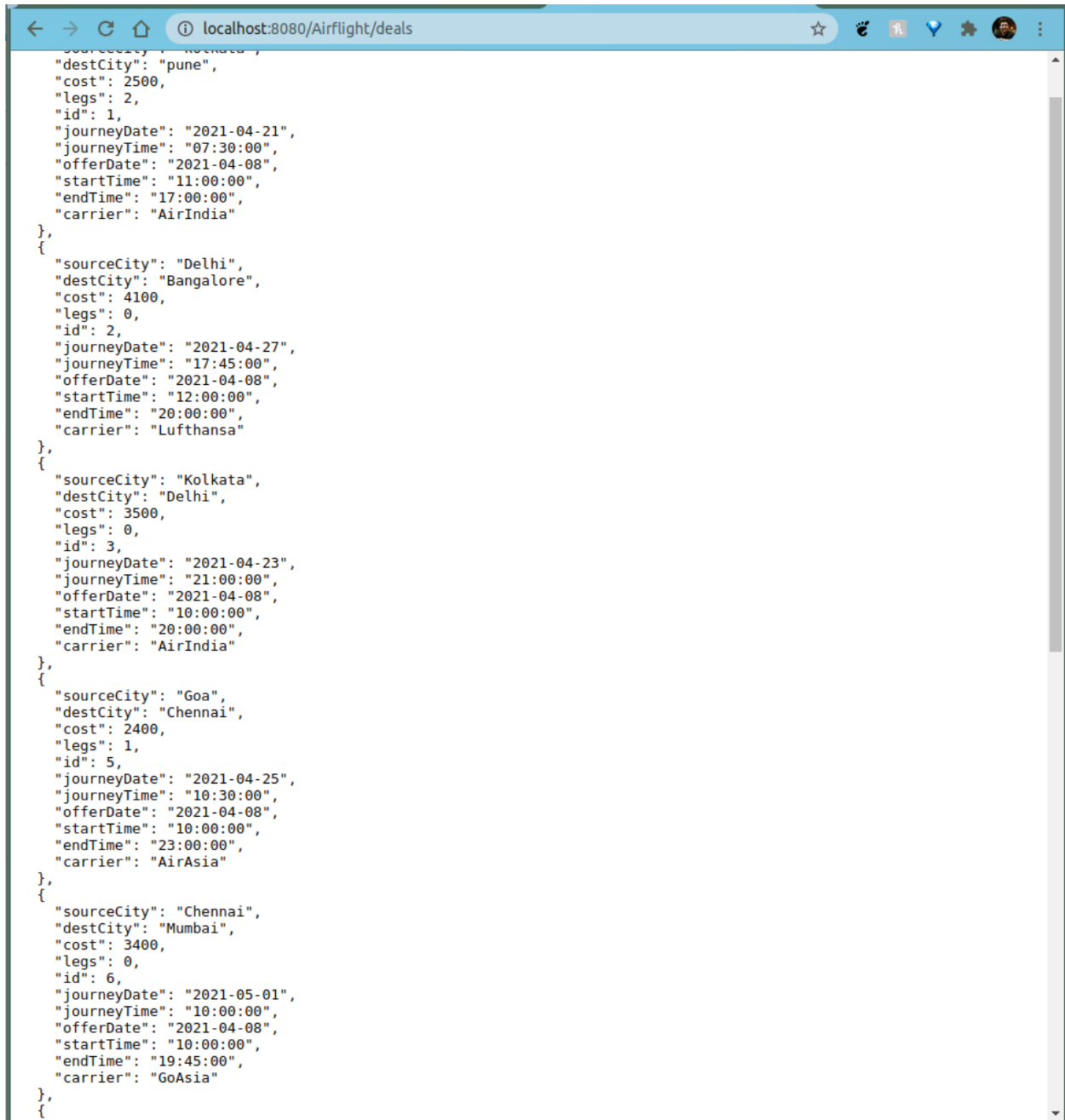
```
////////////////////////////////////  
  
//offerList.add(new Offer(5,"Kolkata","Banglore", 2500, 2, "22-04-2021", "07:30", "04-04-2021","11:00", "17:00", "GoIndia"));  
  
GsonBuilder gsonBuilder = new GsonBuilder().setPrettyPrinting();  
Gson gson = gsonBuilder.create();  
String jsonObj = gson.toJson(offerList);  
out.print(jsonObj);  
  
out.flush();  
try {  
    connection.close();  
} catch (SQLException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

Display page



Here the display page pulls data from the database through the backend and displays the deals of the day accordingly. This page also contains a booking page for users to fill out.

The exposed apis of backend from where this page pulls data looks like this. All the interaction with the backend has done with REST api using the JSON object format.



A screenshot of a web browser window displaying a JSON array of flight deals. The browser's address bar shows the URL `localhost:8080/Airflight/deals`. The JSON data is as follows:

```
[{"sourceCity": "Kolkata", "destCity": "pune", "cost": 2500, "legs": 2, "id": 1, "journeyDate": "2021-04-21", "journeyTime": "07:30:00", "offerDate": "2021-04-08", "startTime": "11:00:00", "endTime": "17:00:00", "carrier": "AirIndia"}, {"sourceCity": "Delhi", "destCity": "Bangalore", "cost": 4100, "legs": 0, "id": 2, "journeyDate": "2021-04-27", "journeyTime": "17:45:00", "offerDate": "2021-04-08", "startTime": "12:00:00", "endTime": "20:00:00", "carrier": "Lufthansa"}, {"sourceCity": "Kolkata", "destCity": "Delhi", "cost": 3500, "legs": 0, "id": 3, "journeyDate": "2021-04-23", "journeyTime": "21:00:00", "offerDate": "2021-04-08", "startTime": "10:00:00", "endTime": "20:00:00", "carrier": "AirIndia"}, {"sourceCity": "Goa", "destCity": "Chennai", "cost": 2400, "legs": 1, "id": 5, "journeyDate": "2021-04-25", "journeyTime": "10:30:00", "offerDate": "2021-04-08", "startTime": "10:00:00", "endTime": "23:00:00", "carrier": "AirAsia"}, {"sourceCity": "Chennai", "destCity": "Mumbai", "cost": 3400, "legs": 0, "id": 6, "journeyDate": "2021-05-01", "journeyTime": "10:00:00", "offerDate": "2021-04-08", "startTime": "10:00:00", "endTime": "19:45:00", "carrier": "GoAsia"}]
```

The booking form:

127.0.0.1:5500/index.html

BOOKING FORM

Journey Date
04/07/2021

Get Deals

Soumalya Kundu

India

soumalyakundu10@gmail.com

Comfort
☒ Economy ☐ Business ☐ Luxury


Boarding City:
Kolkata

Destination City:
delhi

Submit

Book Flight

WELCOME



Have the most wonderful experience flying with Travel Thru Air
Wish you the best Journey!

PRICE TO BE PAID:

₹0

Now with all the data, Book Flight button automatically redirects user to the flight search and booking page where user will get the details of all the available flights on that day and between those two cities.

127.0.0.1:5500/index-1.html?dest=delhi&source=Kolkata&date=04/07/2021

ABOUT BOOK FLIGHT UPCOMING TOURS CONTACTS

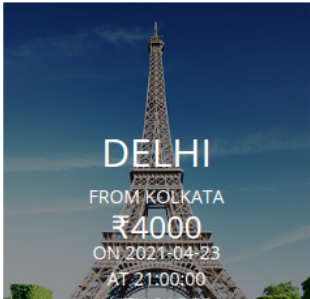
YOUR TRIP

SEARCH FORM

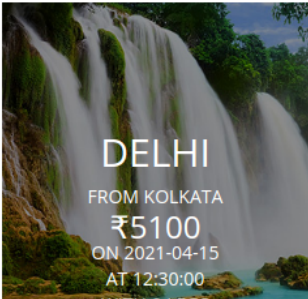
Boarding City:

Destination City:

Search



DELHI
FROM KOLKATA
₹4000
ON 2021-04-23
AT 21:00:00
WITH 1 LEG



DELHI
FROM KOLKATA
₹5100
ON 2021-04-15
AT 12:30:00
WITH 1 LEG

Here users can search flights based on time, price and/or places. Corresponding backend structure code snippet is as follows

```
protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
{
    res.setContentType("application/json");

    ArrayList<Flight> flightList = new ArrayList<>();
    Connection connection = null;

    try {
        Class.forName("org.postgresql.Driver");
        connection = DriverManager.getConnection("jdbc:postgresql://localhost:5432/FlightDB", "postgres", "soumalya@18");

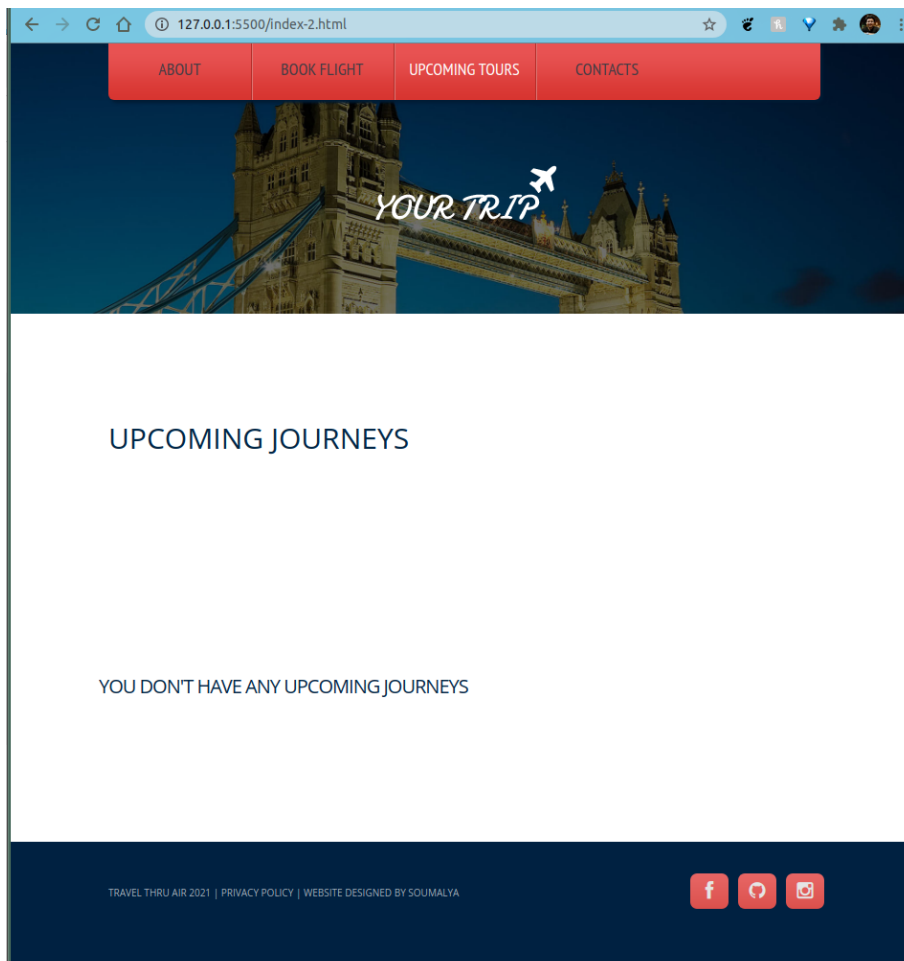
        if(connection != null)
            System.out.println("Hey hello high yoo");
        else
            System.out.println("Nopeeeeeeeeeeeeeeeee!");
    } catch (Exception e) {
        // TODO: handle exception
        System.out.println(e);
    }

    //System.out.println("Here inside of flights");
    String query = "select * from \"Flights\" where acity=${acity} and dCity = ${dCity}";

    try {
        Statement st= connection.createStatement();
        ResultSet rs = st.executeQuery(query);

        while(rs.next()) {
            Flight flight = new Flight();
            flight.convert(rs);
        }
    }
```

Upcoming Journeys



Comments

This assignment gave us an opportunity to learn many new things. I learnt how to build an application by integrating Java servlets and javascript frontend. I also learnt the usage of the GSON library and how to integrate backend and frontend using REST api. I also learnt how to use and configure database and how to perform basic CRUD commands on a database.

