

Internet Technologies Lab Report
Assignment 3

Md Sahil
BCSE IV
Roll-001710501029

1 Problem Statement

Implement a web application for “Travel Thru Air” using servlets to support the following two use cases

- A list of current special deals must appear on the home page. Each special deal must display the departure city, the arrival city, and the cost. These special deals are set up by the marketing department and change during the day, so it can’t be static. Special deals are only good for a limited amount of time.
- A user may search for flights, given a departure city, time and an arrival city. The results must display the departure city, the arrival city, the total cost, and how many legs the flight will have. State and explain why and where you have used design patterns.

2 Design & Implementation

2.1 Directory structure

```
travelthruair/  
|-- WEB-INF  
|   |-- classes  
|   |   |-- Makefile  
|   |   |-- api  
|   |   |   |-- ApiService.class  
|   |   |   |-- ApiService.java  
|   |   |-- model  
|   |       |-- Deal.class  
|   |       |-- Deal.java  
|   |       |-- Deals.class  
|   |       |-- Deals.java  
|   |-- lib  
|   |   |-- gson-2.8.6.jar  
|   |   |-- servlet-api.jar  
|   |-- web.xml  
|-- index.html  
-- js  
    |-- app.js
```

6 directories, 12 files

2.2 Compilation

Compilation is done using the *Makefile*. In order to compile the java classes, run the following command while in the **travelthruair/WEB-INF/classes** directory

```
$ make
```

2.3 Design

The application is build as a RESTfull service. The client side makes API calls to the server and the server responds accordingly.

The api calls can be made with the url `<hostname>/travelthruair/api/deals`.

Filtering of flights can be done using query params. The availabe query params are *arrivalCity*, *departureCity*, *arrivalDate*, *departureDate*.

Example api call with query params:

```
<hostname>/travelthruair/api/deals?arrivalCity=Kolkata&departureDate=2021-02-07&
```

Returned json data:

```
[
  {
    "departureCity": "Kolkata",
    "arrivalCity": "Delhi",
    "cost": 3000,
    "arrivalDate": {
      "year": 2021,
      "month": 2,
      "day": 8
    },
    "departureDate": {
      "year": 2021,
      "month": 2,
      "day": 7
    }
  }
]
```

The following error checks have been implemented:

- Arrival and departure city can't be the same
- Departure date and arrival date cannot be before the current date
- Arrival Date cannot be before departure date

The frontend consist of a simple form that makes calls to this api in order to fetch the available flights. Javascript functions are used to fetch the flight details and fill it in the HTML elements. The *get deals* button returns all the upcomming available flights

2.4 MVC

The backend consists of 2 packages the **api** package and the **model** package The **api** package consists of the servlet classes and acts as the *Controller*. The model package consists of two classes. The **Deal** class is a simple DTO (Data Transfer Object) class. Each object of the **Deal** class corresponds to a flight deal. The **Deals** class acts as the DAO (Data Access Object) class and encapsulated the methods that returns the set of **Deal** objects accorting to the search params and also performs error checking. The **model** package acts as the *Model* in the MVC pattern. Finally the HTML frontend that displays the contents fetched by the Javascript functions acts as the *View*