**DST PROJECT REPORT**

**SHORTEST & QUICKEST FLIGHT ROUTES**

*An innovative project*

Submitted by:

*Dipra Chaudhry (2K19/IT/046)*

*Gargi (2K19/IT/048)*

*Ishita Lather(2K19/IT/063)*

under the guidance of

**Ms Swati Sharda**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of

Engineering) Bawana Road, Delhi-

110042

DECEMBER, 2020

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

**CANDIDATE’S DECLARATION**

We solemnly declare that the project report titled **Shortest & Quickest flight routes** is based on our own work carried out during the course of our study under the supervision of **Ms Swati Sharda.**

We assert the statements made and conclusions drawn are an outcome of our research work. We further certify that

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.
4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Place: Delhi

Date: 2-12-2020

Dipra Chaudhry

Gargi

Ishita Lather

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

**CERTIFICATE**

This is to certify that *Dipra Chaudhry , Gargi & Ishita Lather*, Roll No : 2K19/IT/046 , 2K19/IT/048 & 2K19/IT/063 , Information Technology, Delhi Technological University, Delhi have successfully completed the project work entitled “**Shortest & Quickest flight routes**” as the 3rd semester Innovative project under the guidance of

*Ms Swati Sharda.*

Place: Delhi **Ms Swati Sharda**

Date: 2-12-2020 Supervisor

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of

Engineering) Bawana

Road, Delhi-110042

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to our supervisor **Ms Swati Sharda** for her invaluable guidance and suggestions throughout the course of this project on “**Shortest & Quickest flight routes**”.

We are also very thankful to our college to have given us this opportunity to do this wonderful project.

**ABSTRACT**

The project “**Shortest & Quickest flight routes**” is a website. This website gives displays the shortest & the quickest flight routes for the user entered origin and destination. These list of flights is displayed in increasing order i.e. the cheapest flights would be displayed at the topmost position in the list. Similarly the rest of the flights would be displayed accordingly.

The shortest & quickest flight routes are calculated using the Dijkstra’s algorithm. While calculating the shortest route the weights of the graph(generated in the backend) would be the distances between the origin and destination airports. And while calculating the quickest route the weights of the graph (generated in the backend) would be the duration i.e. the time taken to fly from the origin airport to the destination airport.

When the user enters the origin & destination , travel dates(from the calendar) and number of passengers in the home page of the website and then hits enter. A new page appears on the screen , and by default the *quickest* flight routes are displayed on the screen. The user can click on the *shortest* button to view the flights flying on the shortest paths.

The user can press the *back* button to go back to the home page of the website wherein he/she can enter a new origin & destination and get the results accordingly.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| Abstract | 5 |
| Table of Contents | 6 |
| Chapter 1 : Introduction | 7 |
| Chapter 2 : About the Dataset | 8 |
| Chapter 3 : About the algorithm | 9 |
| Chapter 4 : Features & Functionalities | 10 |
| Chapter 5 : Methodology   * How does it work? * Flowchart | 12 |
| Chapter 6 : Results & Analysis | 13 |
| Chapter 7 : Conclusion | 19 |
| Chapter 8 : Scope | 20 |

**INTRODUCTION**

Less than 20 percent of the world’s population has set foot on an airplane, but that’s changing dramatically. Asia is home to nine of the world’s ten most popular flights, and while air travel is booming around the world, it’s expected to soar exponentially in the region.

That brings both unprecedented opportunities and challenges, the growth will generate countless new flights & flight routes.

Choosing the right flight route becomes a very cumbersome task.

The aim of this project is to make a single platform that can be used to get the list of shortest and least time consuming flights between the desired origin & destination after comparing all the possible routes connecting the airports (origin airport & destination airport)

**ABOUT THE DATASET**

Two datasets (json files) have been made in the backend.

* One for the airport data (origin , destination , distance , duration)
* Other for the flight data (flight no , organization , origin code , destination code , departure time , price)

1. Seven airport have been taken into consideration while doing this project.
2. Over 100 flights have been added in the flight data json



DEL : Delhi

HKG : Hong Kong

KUL : Kuala Lumpur

SYD : Sydney

SIN : Singapore

BKK : Bangkok

JKT : Jakarta

**DIJKSTRA’S ALGORITHM**

Dijkstra’s algorithm gives the single-source shortest path in a weighted directed graph.

It is a generalized version of the BFS traversal.

Dijkstra algorithm maintains a set S of vertices in a graph having V vertices and E edges, whose shortest path weights have already been defined. It repeatedly selects the remaining vertices with minimum weight and adds them to set S, and relaxes the other vertices.

Steps :

1. We have to identify the source node at first.
2. Mark all the nodes unvisited and store them in a queue.
3. We initialize the source node with distance zero and all the other nodes with infinity.
4. Since the algorithm follows a greedy approach we select the unvisited node with minimum weight.
5. The current node is marked as a visited and simultaneously remove it from the unvisited set that we had maintained.
6. The algorithm stops if all the nodes have been visited or the list becomes empty. Otherwise we iterate over all the nodes and repeat the above two steps.
7. In the end finally we return the shortest path as the result of the algorithm.

**FEATURES & FUNCTIONALITIES**

Features :

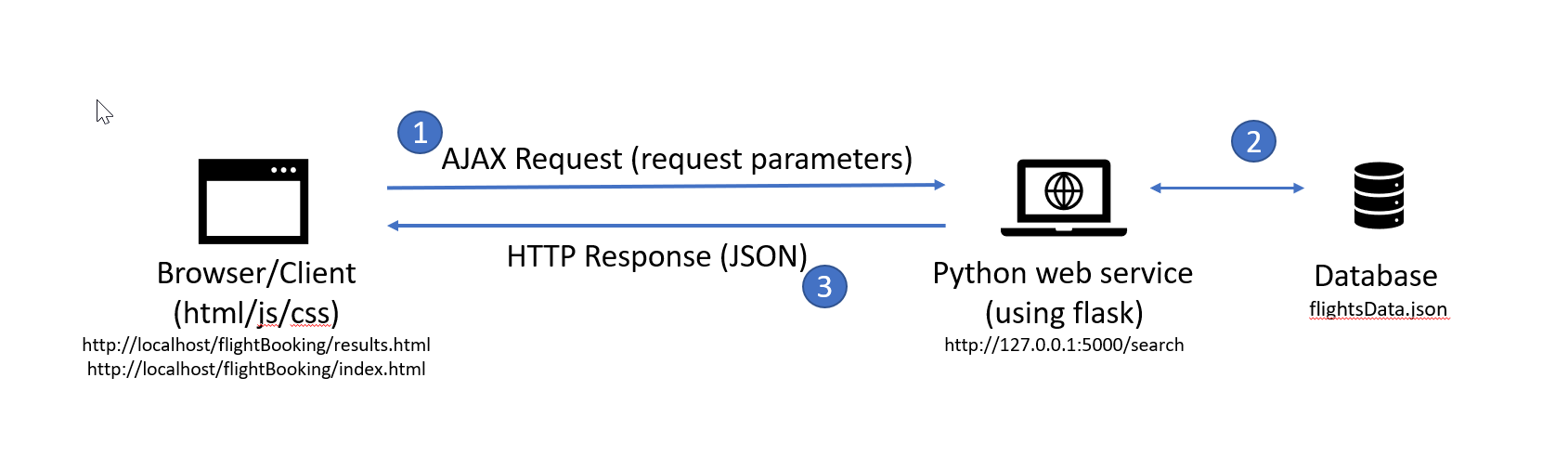
* User can view the ***quickest*** connecting flights(duration-wise) between any two locations.
* User can view the ***shortest*** connecting flights(distance-wise) between any two locations.
* All the flights appear in a sorted fashion. The cheapest ones at the top.
* Buttons can be used to easily switch from one particular view to the other

Functionalities :

* The origin & destination is entered by the user
* Followed by entering the desired one-way travel and return dates(optional)
* After pressing the *search* button , by default the *quickest* flight routes are displayed in a sorted manner.
  + After pressing the search button , Dijkstra’s algorithm is invoked and it finds the least time consuming path.
  + In this case the weights of the directed graphs generated at the backend are the duration / the time taken to fly from one airport to the other.
* If the user clicks on *shortest* button
  + Dijkstra’s algorithm is invoked and it finds the shortest path on the basis of distance
  + In this case the weights of the directed graphs generated at the backend are the distance from one airport to the other.
  + The user can click on the back button to go back to the home page where he/she can search for the quickest/shortest flight routes for some new origin & destination



**METHODOLOGY**



1)In the home page , the user enters the origin , destination , travel date and hits the search button.

2)After which he/she selects the desired display-option by clicking on either the *shortest/quickest* button

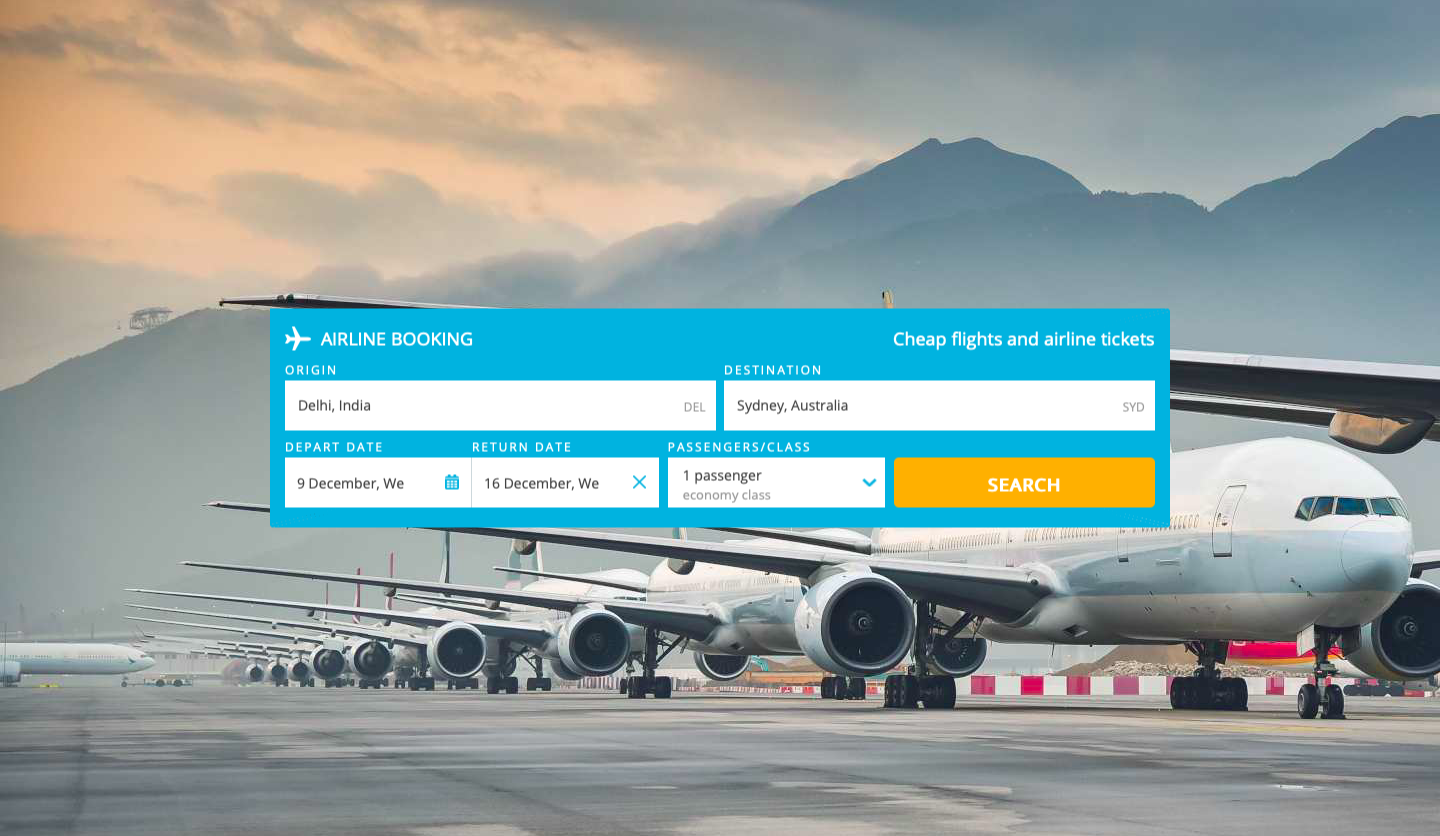
3)An ajax request is made to the backend(app.py) using Flask.

4)In the backend , the graph is loaded and then the Dijkstra’s algorithm is called on this graph for the user entered origin and destination(the weights of the loaded graph would be either the distance/duration and this is decided by the display-option entered by the user).

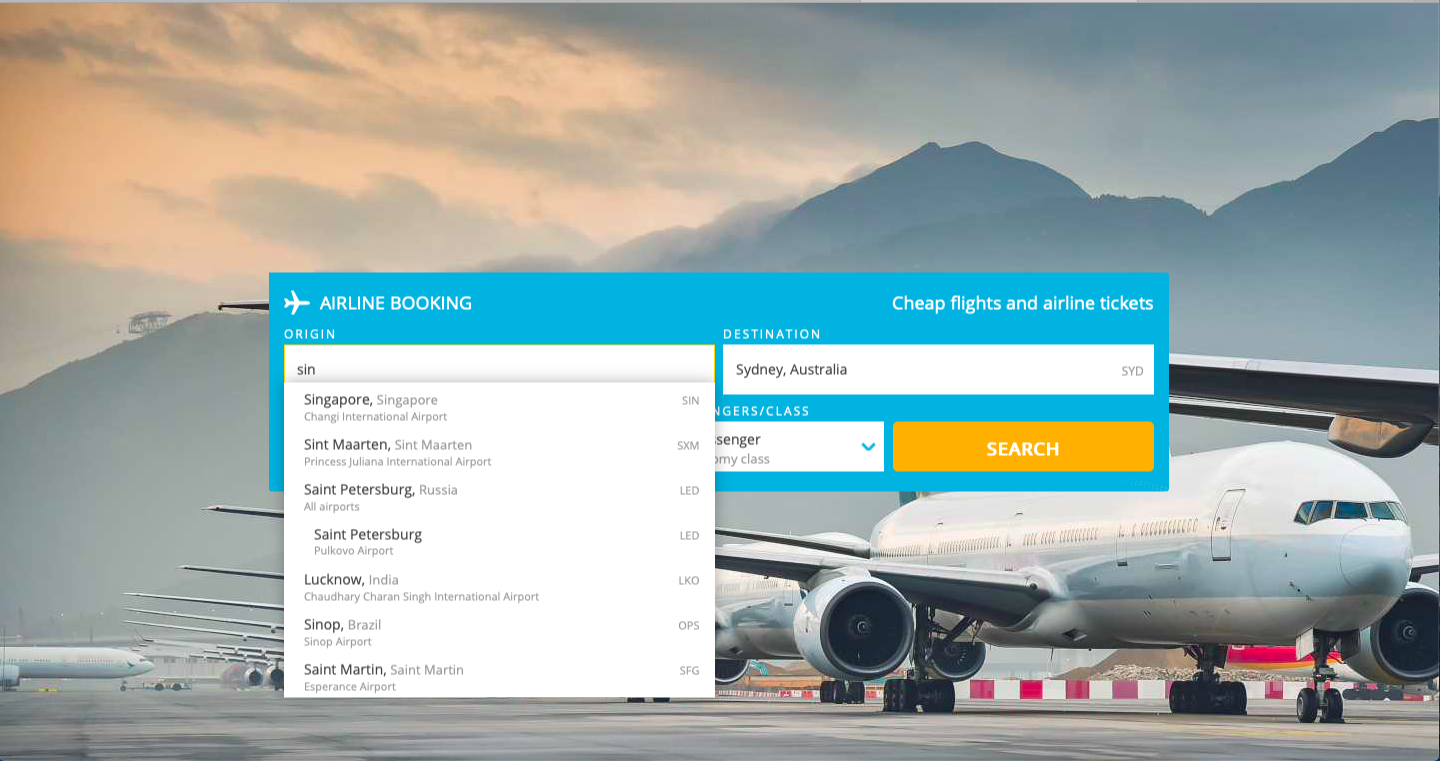
5)Then a json response is sent from the backend to result.html

and the route is displayed on the screen.

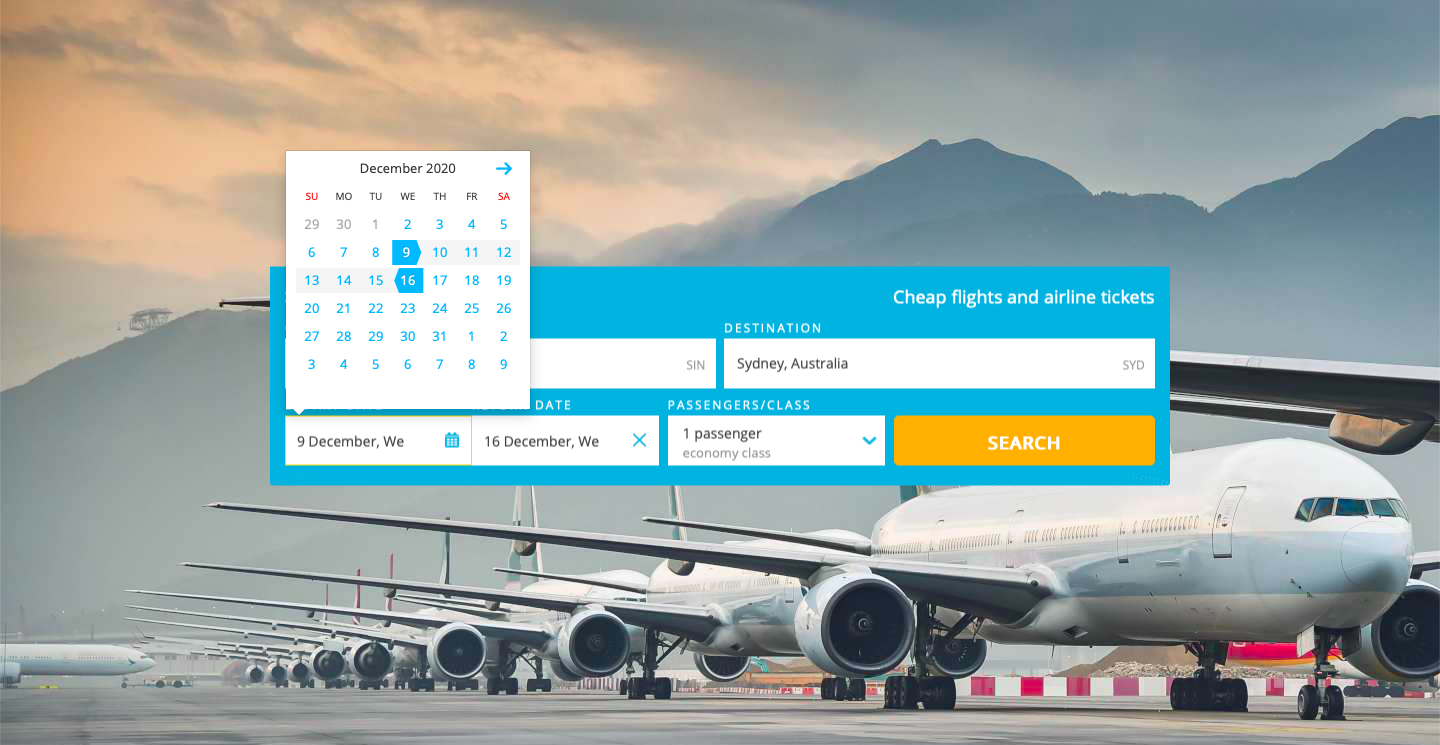
**RESULT**

****

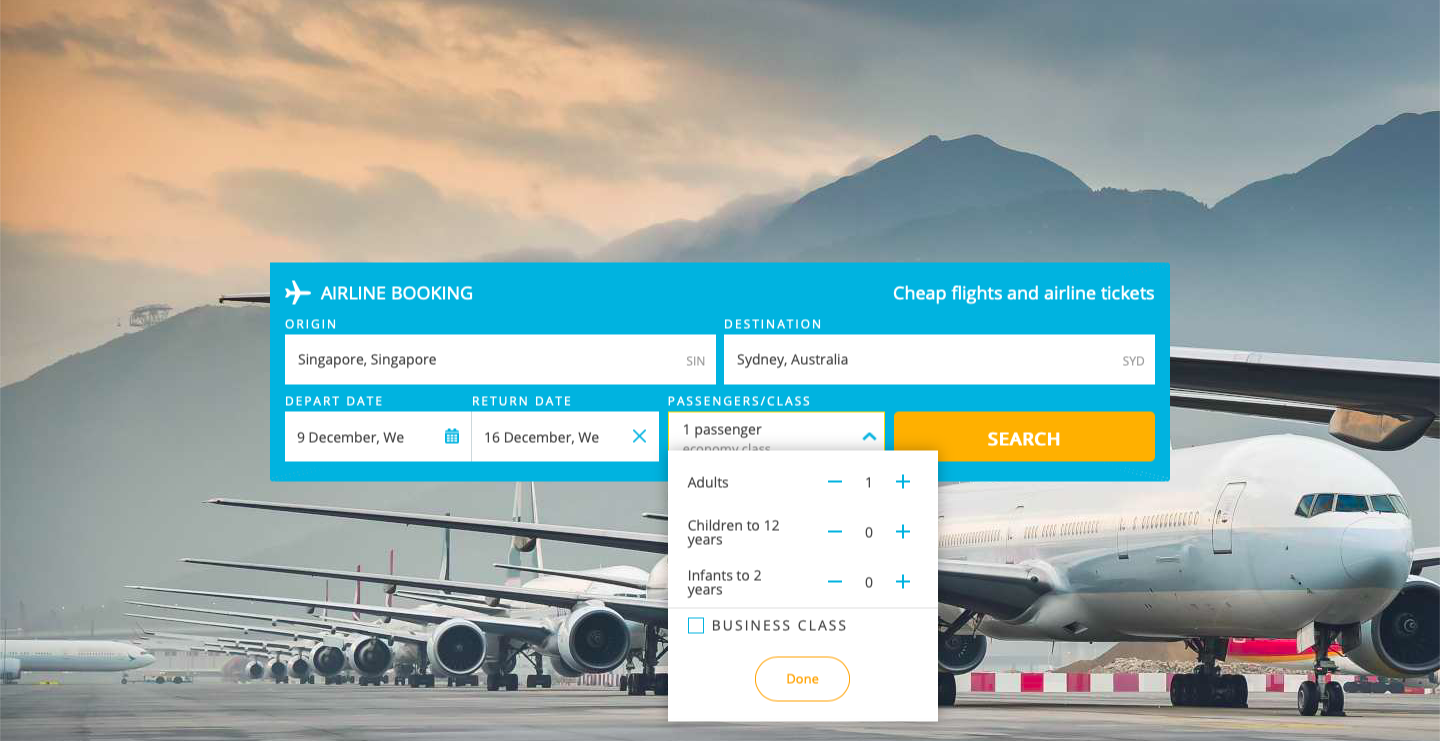
Enter origin & destination



Pick the desired dates

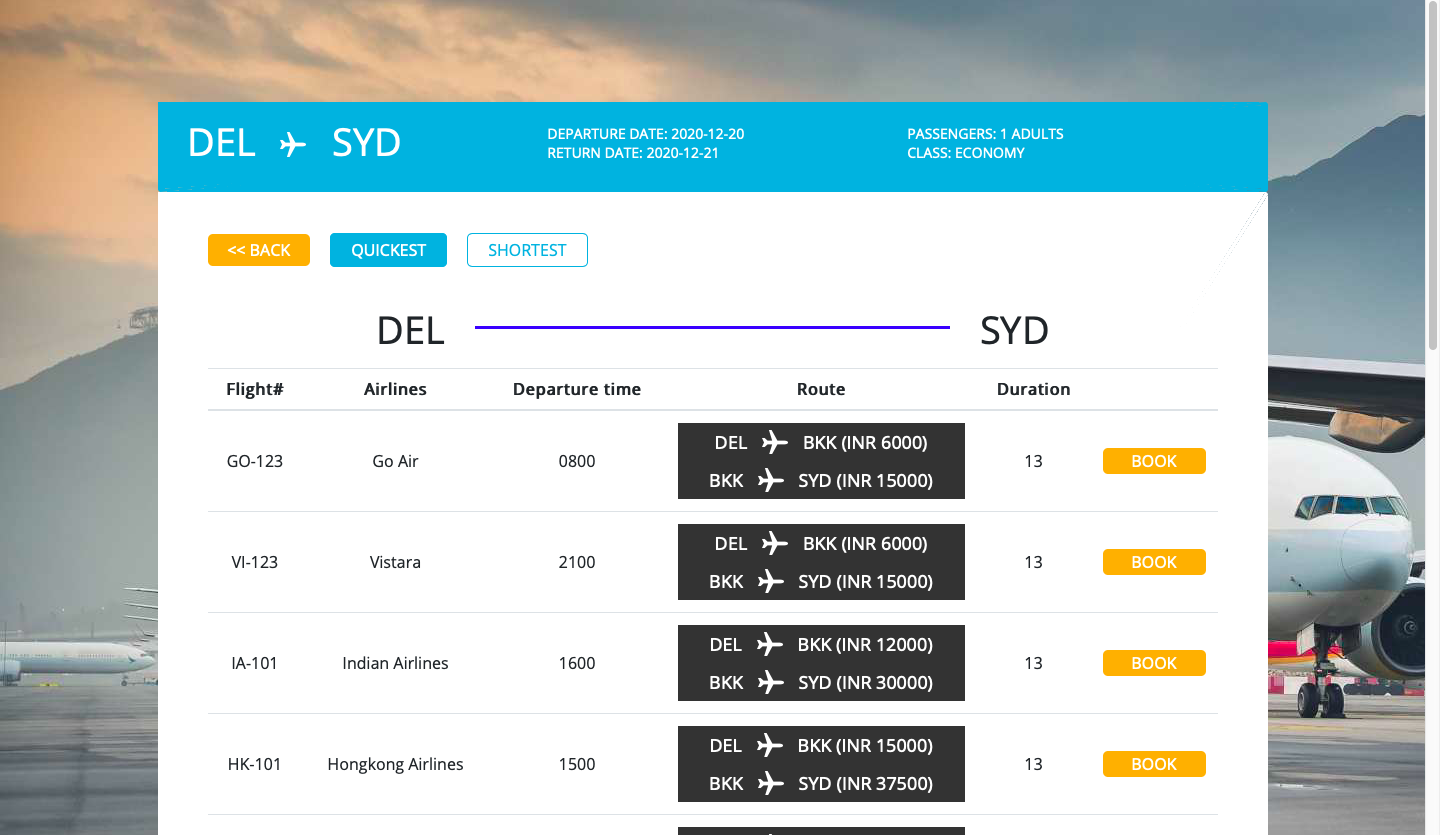


Enter the number of passengers & and pick the class

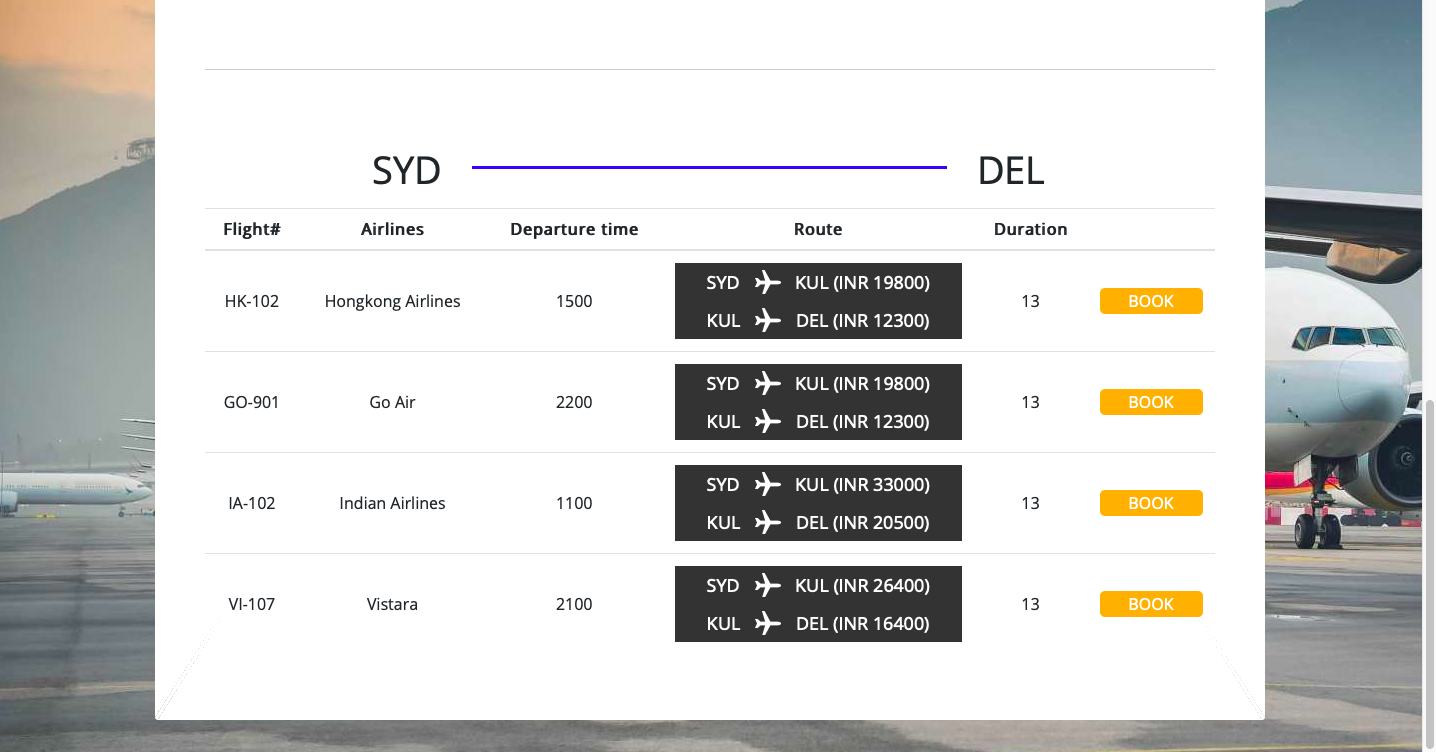


**OUTPUT**

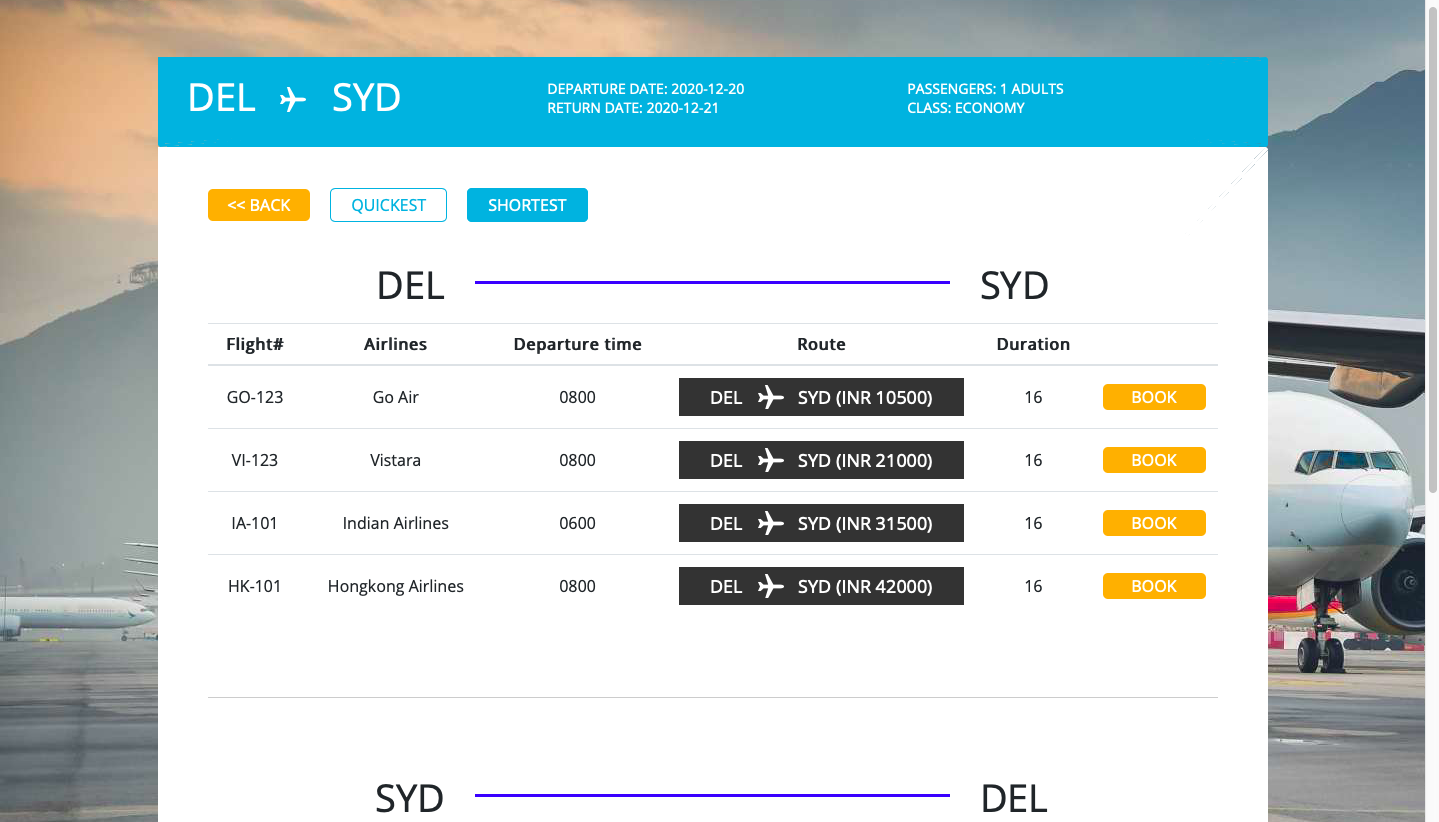
Quickest Route



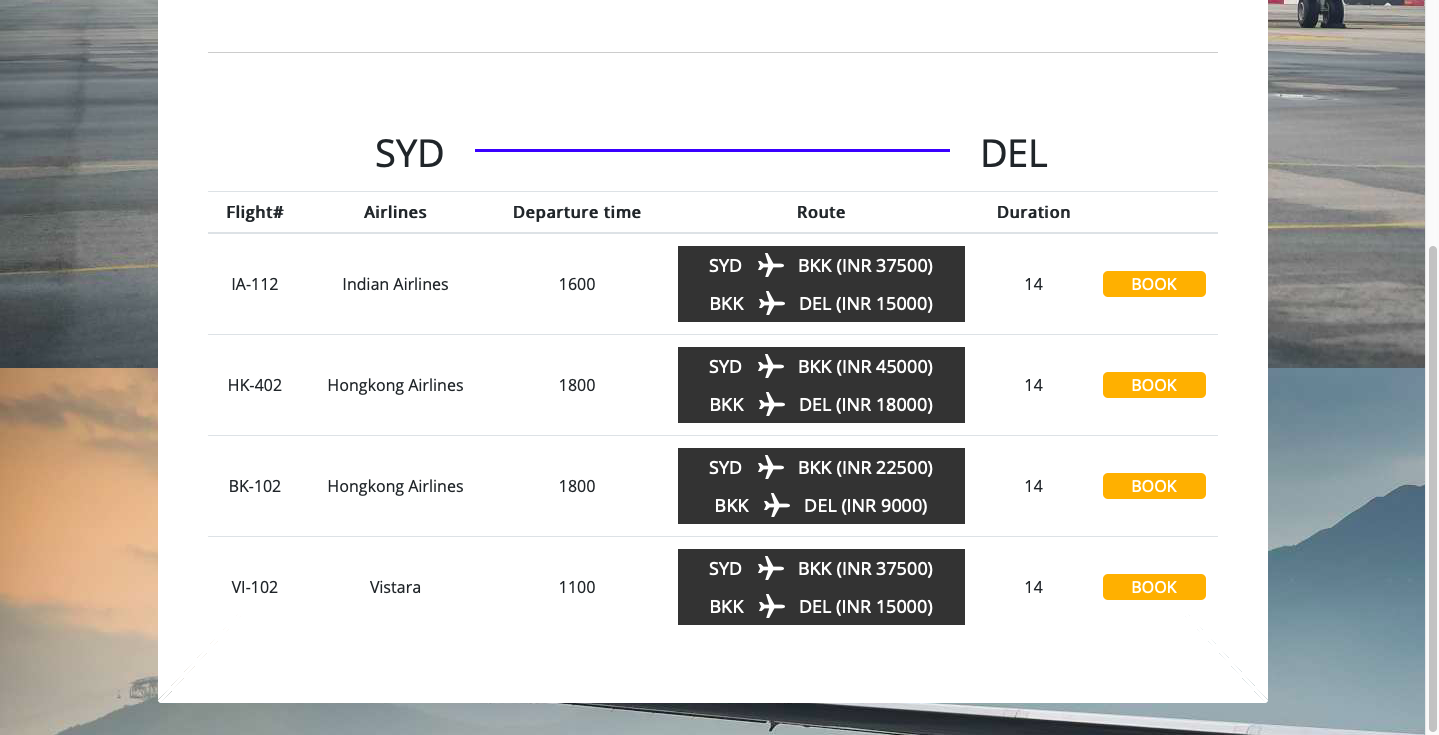
Quickest Route for return flight

****

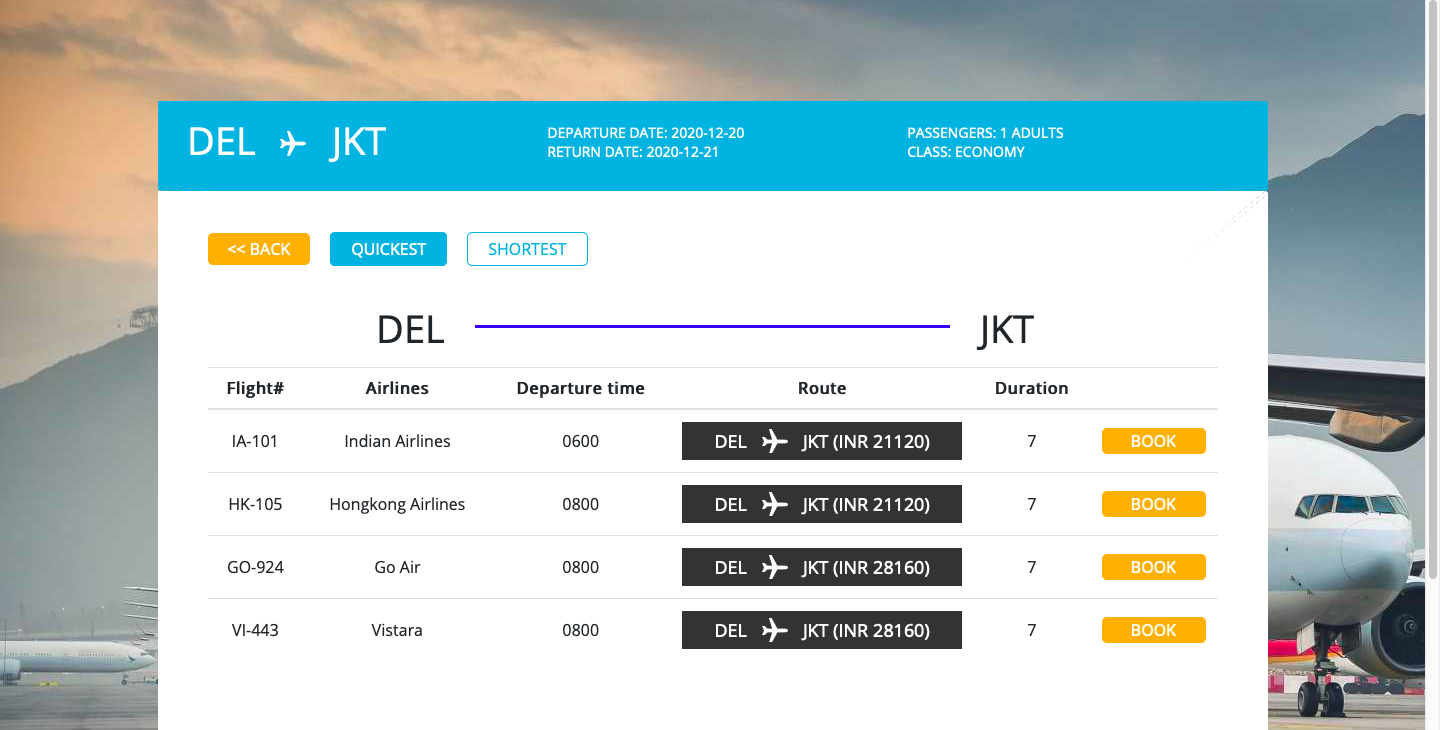
Shortest Route



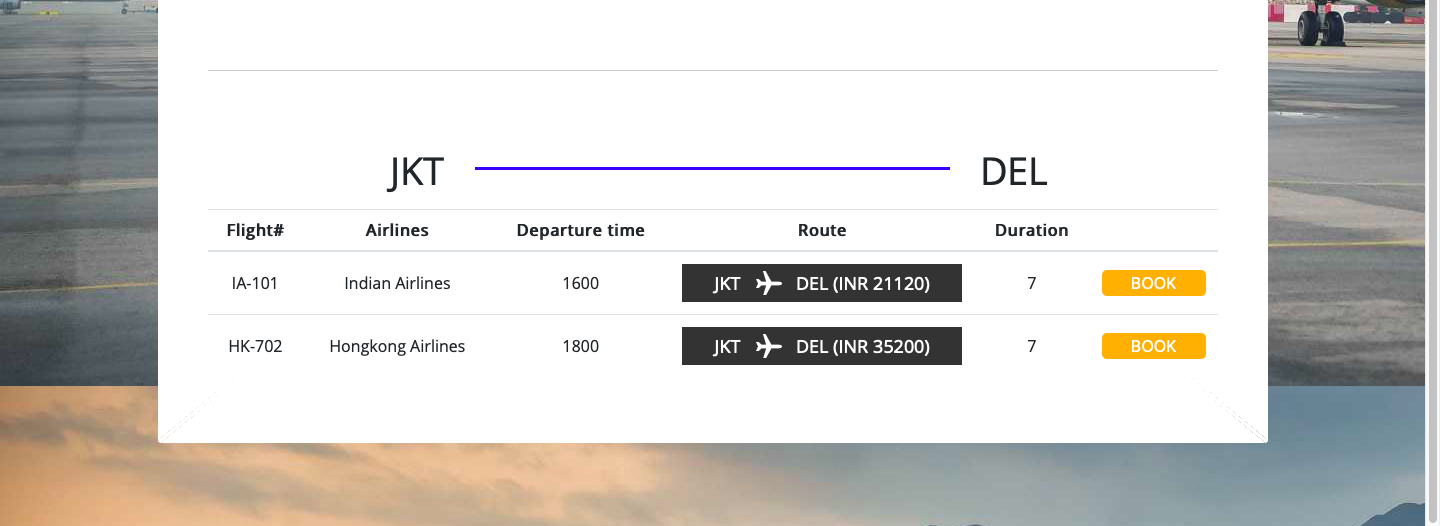
Shortest Route for return flight

****

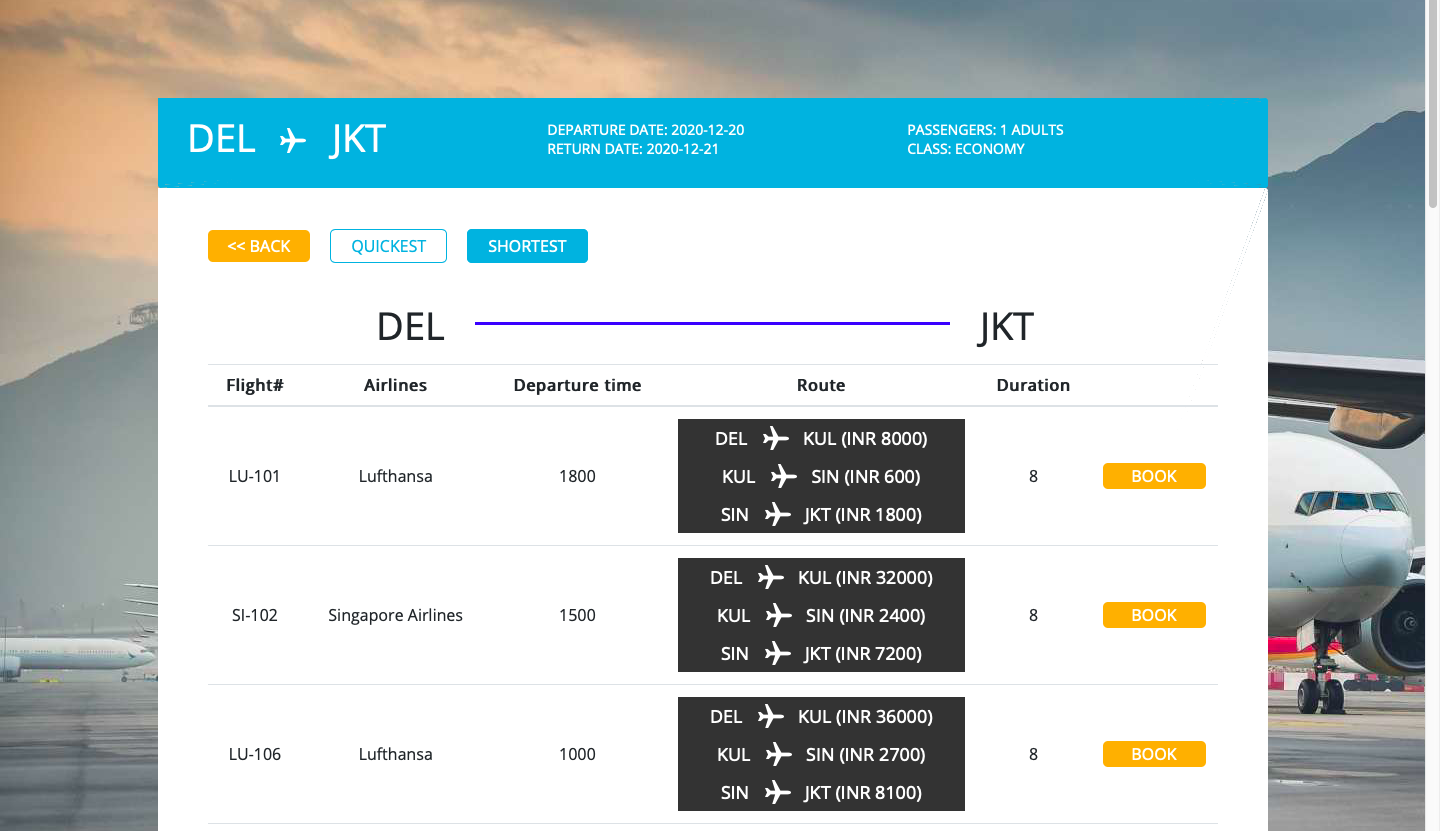
Quickest Route

****

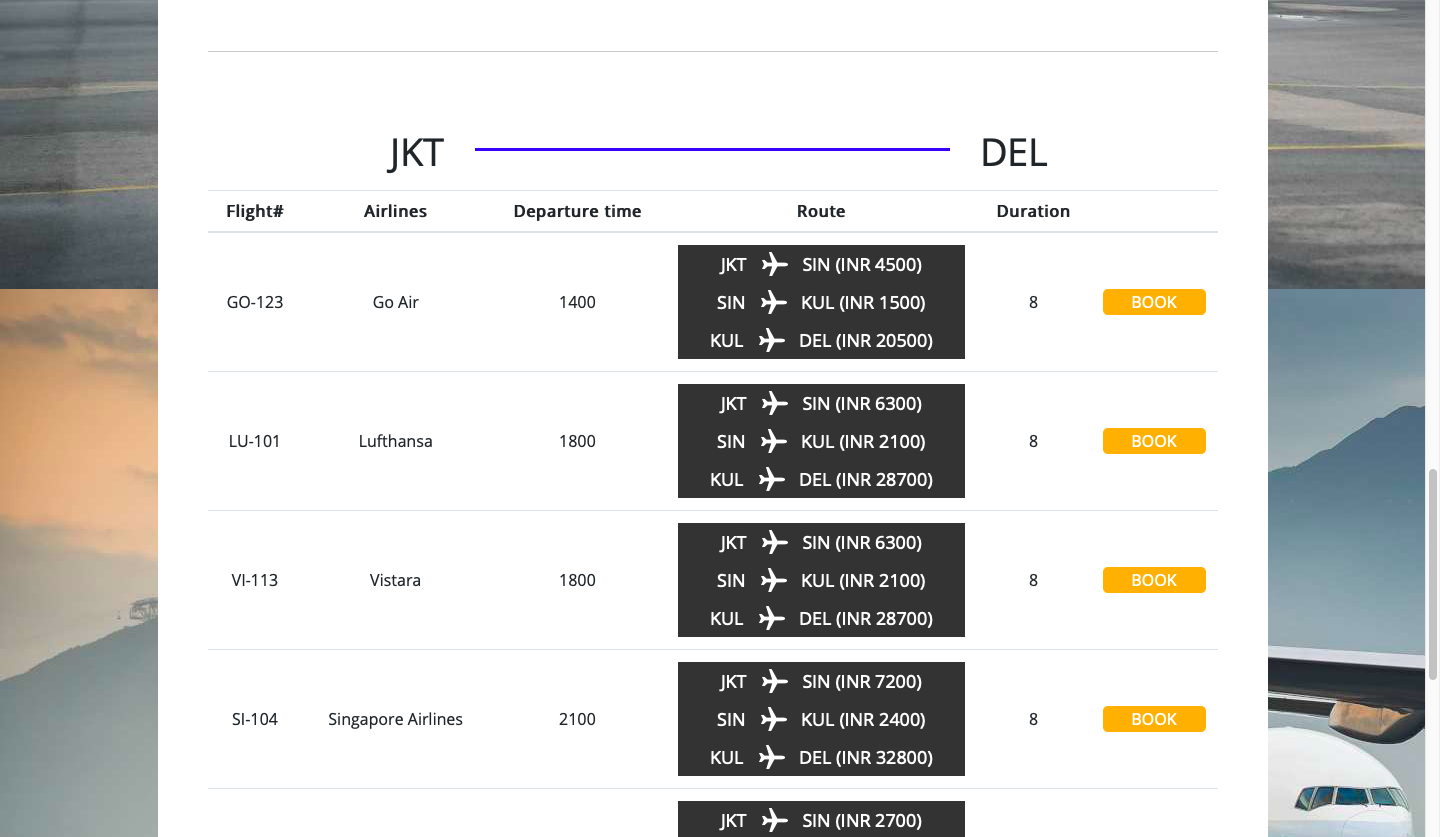
Quickest Route for return flight

****

Shortest Route

****

Shortest Route for return flight

****

**CONCLUSION**

Various data structures & web development concepts(backend-python ; frontend-html , CSS , javascript ; Flask framework) have been used in developing this website which displays the quickest & shortest flight routes in a sorted manner.

**SCOPE**

In future , this website can be converted into a meta-search engine. It would fetch the airport data(origin , destination , distance , duration) & the flight data(flight no , organization , origin code , destination code , departure time , price) directly the internet and display the shortest & quickest flight routes accordingly. This would eliminate the need of giving two separate json containing the airport and flight data.