JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, Noida

Department of CSE & IT



Bachelor of Technology, \_\_Semester

**TRAVEL AGENCY**

2019-2020

2

**Ankita Bansal Batch B1 17103009**

**Srishti Chaudhary Batch B1 17103028**

**Snigdha Das Batch B1 17103033**

Submitted To:

**Deepti Singh**

**INTRODUCTION**

**Travel Agency in C++** is a system that stores the travellers travel data in the database and the system can also add, delete, and edit user information in database. Here the user can add new trip in the database and edit or delete the trip route as their need.

This project uses Dijikstra’s Algorithm to help the customer find the shortest path from the source to the destination city amongst the various routes available. This will be beneficial as it will save the customer their time and money.

The project also uses interval tree algorithm. This algorithm comes handy when the customer has to find the tickets available during a certain period of time.

***ALGORITHMS USED :-***

1. **Dijikstra’s Algorithm**: is an [algorithm](https://en.wikipedia.org/wiki/Algorithm) for finding the [shortest paths](https://en.wikipedia.org/wiki/Shortest_path_problem) between [nodes](https://en.wikipedia.org/wiki/Vertex_(graph_theory)) in a [graph](https://en.wikipedia.org/wiki/Graph_(abstract_data_type)), which may represent, for example, road networks.
2. **Interval Tree**: is a [tree data structure](https://en.wikipedia.org/wiki/Tree_(data_structure)) to hold [intervals](https://en.wikipedia.org/wiki/Interval_(mathematics)). Specifically, it allows one to efficiently find all intervals that overlap with any given interval or point. It is oftenused for windowing queries, for instance, to find all roads on a computerized map inside a rectangular viewport, or to find all visible elements inside a three-dimensional scene.
3. [**Travelling Salesman Problem (TSP)**](https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/): Given a set of cities and distance between every pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns back to the starting point.