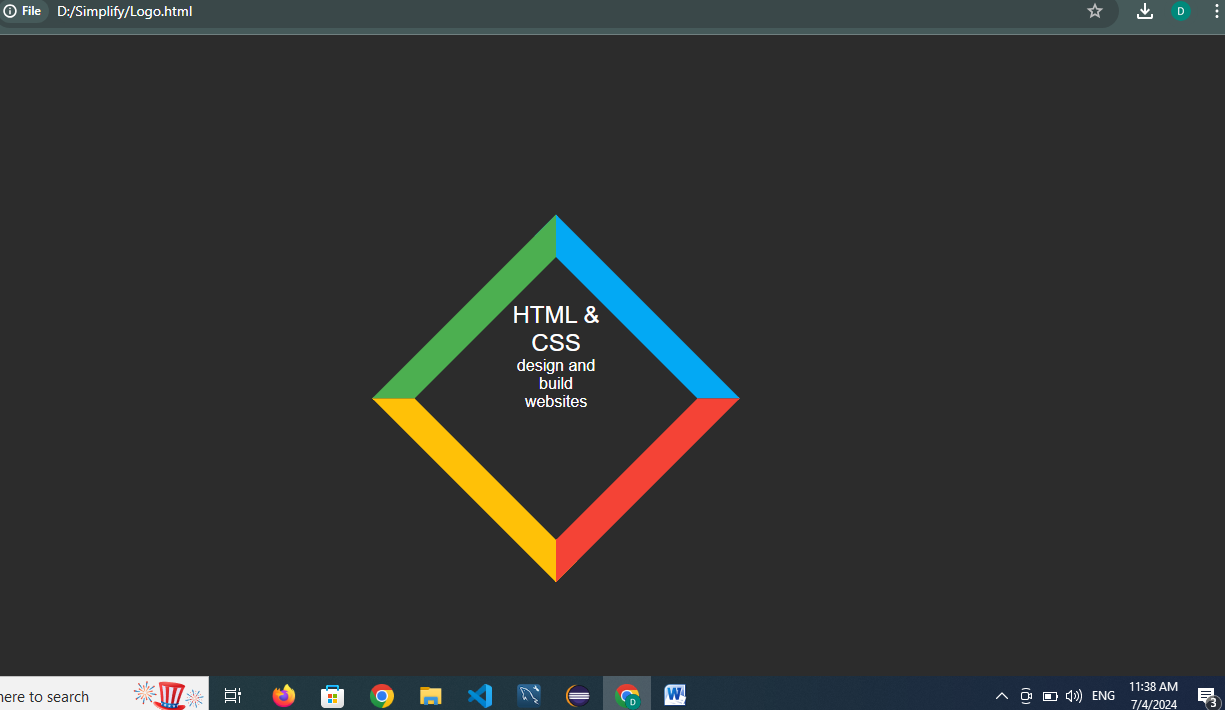
Assignment1: Logo code present in logo.html

Output: 

Assignment2 : Algorithm

1.Graph Construction:

Use a Map<String, List<String>> to represent the graph. Each key is a city, and the value is a list of cities that can be reached directly from the key city.

For each ticket, add the destination city to the adjacency list of the departure city.

2.Sort the Adjacency Lists:

To ensure that we can traverse the cities in lexical order, sort the adjacency lists of each city. This will help in visiting the cities in alphabetical order if there are multiple destinations from a city.

Depth-First Search (DFS):

3.Use DFS to explore the graph.

Start from the given starting city and recursively visit all reachable cities.

Use a stack to store the final route. This is because DFS will explore as deep as possible before backtracking, resulting in the route being stored in reverse order.

4.Reverse the Route:

Since the route stored in the stack is in reverse order of the actual travel path, pop cities from the stack to get the route in the correct order.