PROJECT REPORT
"Flight Route Explorer:

Navigating the Skies with Ease"

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DESIGN AND ANALYSIS OF ALGORITHMS (21CSC204J)

B.TECH CSE(AI/ML) - A, II Year, IV Semester

S.No	Name	Registration No.
1	Shreya garg	RA2211026030022

FACULTY OF ENGINEERING & TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

DELHI NCR CAMPUS, MODINAGAR

SIKRI KALAN, DELHI MEERUT ROAD, DIST. - GHAZIABAD - 201204

EVEN SEMESTER (2024-2025)

BONAFIDE CERTIFICATE

This is to certify that the following details represent a bonafide record of the project work accomplished by **SHREYA GARG**, Registration No: **RA2211026030022**, enrolled in the CSE-AI/ML-A program, 4th semester, 2nd year (2023-24) at SRM INSTITUTE OF SCIENCE & TECHNOLOGY, DELHI-NCR Campus, for the Department of Computer Science & Engineering, specifically in the subject of **Design and Analysis of Algorithms** during the academic year **2023-2024**.

SIGNATURE

Ms. Arti Gupta
Assistant Professor
Dept. of Computer Science & Engineering

SIGNATURE

Dr. Avneesh Vashistha
Head of the Department
Dept. of Computer Science & Engineering

Signature of Internal Examiner

Signature of External Examiner

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher Ms Arti Gutpa as well as our HoD B. Tech Second Year Dr. Avneesh Vashistha who gave me the golden opportunity to do this wonderful project on the "Flight Route Explorer: Navigating the Skies with Ease." which also helped me in doing a lot of Research and came to know about so many new things. I am thankful to them.

Secondly, I would also like to thank my parents and friends who helped me a lot in finishing this project within the limited time.

DECLARATION

I, SHREYA GARG [RA2211026030022], hereby declare that the work which is being presented in the project report "FLIGHT ROUTE EXPLORER: NAVIGATING THE SKIES WITH EASE" is the record of authentic work carried out by us during the period from January ’24 to May ’24 and submitted by us in partial fulfillment for the award of the degree “Bachelor of Technology in Computer Science and Engineering” to SRM IST, NCR Campus, Ghaziabad (U.P.). This work has not been submitted to any other University or Institute for the award of any Degree/Diploma.

SHREYA GARG [RA2211026030022]

TABLE OF CONTENTS

ABSTRACT

The Flight Route Explorer project is a web application designed to visualize and analyze flight routes and airport data using non-Euclidean geometry. Flight routes and airport locations are essential components of air transportation infrastructure, guiding route planning, navigation, and aviation safety. However, efficiently accessing and visualizing this information can be challenging due to its complexity and dynamic nature.

The Flight Route Explorer addresses this challenge by leveraging non-Euclidean geometry, particularly on a 3D globe model, to provide an intuitive and accurate representation of global air transportation networks. By incorporating non-Euclidean geometry, which is better suited for modeling spherical surfaces like the Earth, the application ensures accurate distance measurements, route representations, and visualization of flight paths.

By harnessing the power of non-Euclidean geometry in flight route visualization and analysis, the Flight Route Explorer project aims to provide a valuable tool for understanding and optimizing global air transportation networks. This project will be beneficial for aviation professionals, researchers, and enthusiasts seeking insights into global air traffic patterns and trends in a realistic and intuitive manner.

INTRODUCTION

In an era defined by globalization and interconnectedness, the aviation industry plays a pivotal role in facilitating travel, trade, and cultural exchange on a global scale. Understanding the intricate network of flight routes, airport locations, and air transportation infrastructure is essential for stakeholders ranging from travelers to aviation professionals. Recognizing this need, we embarked on the development of Flight Route Explorer, a comprehensive platform designed to revolutionize the way we explore and interact with the world of aviation.

Flight Route Explorer serves as a multifaceted tool, catering to a diverse range of users with varying interests and objectives. From travelers seeking to plan their next adventure to aviation professionals analyzing air traffic patterns, Flight Route Explorer offers a wealth of features and functionalities aimed at enhancing understanding and facilitating exploration.

At the core of Flight Route Explorer lies a meticulously curated database of airport information, encompassing airports from every corner of the globe. Users can access detailed data on airport locations, facilities, IATA and ICAO codes, and geographic coordinates, providing a comprehensive overview of global air transportation infrastructure.

The visualization capabilities are equally impressive, offering users an immersive experience as they navigate the world's skies. With a dynamic 3D globe interface, users can explore airport locations, visualize flight routes, and gain insights into air transportation networks with unparalleled ease and precision.

For travelers, Flight Route Explorer serves as a valuable planning tool, allowing them to search for airports, plot routes, and calculate distances between destinations. Whether planning a leisurely vacation or a business trip, users can leverage it’s intuitive interface to streamline the planning process and make informed decisions.

Aviation professionals also stand to benefit from Flight Route Explorer's advanced analytics capabilities. With access to detailed airport data and sophisticated algorithms for route optimization and analysis, airlines, logistics companies, and researchers can gain valuable insights into air transportation networks and optimize operations for efficiency and reliability.

Incorporating principles of non-Euclidean geometry, Flight Route Explorer transcends traditional Cartesian frameworks, offering a fresh perspective on spatial relationships and distances within the global aviation landscape. By embracing the curvature of the Earth's surface and adopting innovative geometric models, it provides users with a more accurate representation of flight paths and distances, enhancing the realism and utility of the platform.

Flight Route Explorer employs advanced geometric principles to calculate the shortest paths between airports. By leveraging concepts such as great circles and geodesics, it accurately determines the most direct routes, accounting for the curvature of the Earth's surface. Great circles represent the shortest distance between two points on a sphere, while geodesics consider factors like altitude and atmospheric conditions. This approach ensures precise route calculations, enabling users to plan their journeys efficiently.

Flight Route Explorer offers users a choice of base layers to enhance their exploration experience. Options include aerial imagery for high-resolution views, street maps for navigational details, and terrain data for understanding elevation. These diverse base layers provide users with different perspectives on flight routes and airport locations, catering to individual preferences and needs. Whether users are interested in geographical features or urban infrastructure, Flight Route Explorer offers a tailored experience for comprehensive exploration.

PROBLEM STATEMENT

In today's interconnected world, air travel plays a crucial role in facilitating global connectivity. However, planning air travel routes efficiently and understanding the relationships between different airports can be complex tasks. Travelers, aviation enthusiasts, and logistics professionals often seek tools to explore flight routes, visualize connections between airports, and calculate the shortest paths between destinations.

The Flight Route Explorer project aims to address this need by developing a web-based application that allows users to interactively explore flight routes, visualize airport connections on a global scale, and calculate the shortest paths between selected airports. This project will leverage data from public airport databases and utilize geospatial visualization techniques to provide users with an intuitive interface for discovering flight routes and optimizing travel plans.

Key Objectives:

1. Create a dynamic and user-friendly web application accessible via standard web browsers. Offer interactive features such as clickable maps, search bars, and informative pop-ups to enhance user engagement. Ensure the application interface is intuitive and easy to navigate, allowing users to seamlessly explore flight routes and airport information.

2. Enable users to search for airports by various parameters such as name, city, IATA code, or ICAO code. Provide auto-complete suggestions and handle partial or misspelled inputs to ensure accurate results. Present search results in a clear and organized manner, allowing users to quickly find the desired airport information.

3. Utilize Dijkstra's algorithm to compute the shortest path between selected airports based on their geographical coordinates. Consider factors such as distance, elevation, and airspace restrictions to determine the most efficient route. Allow users to specify multiple waypoints to create complex flight routes.

4. Utilize CesiumJS or similar libraries to create an interactive 3D globe interface. Users can view flight routes, airport locations, and relevant

geographic features. This immersive visualization enhances users' understanding of global air travel networks and allows them to explore routes from different perspectives.

5. Develop a separate section within the application interface to display distances in kilometers between pairs of selected airports. Provide clear and concise information to users, facilitating route planning and decision-making. Additionally, allow users to view distance information for specific flight routes and waypoints.

METHODOLOGY

1. **Requirements Gathering:**
    - Conduct interviews and discussions with stakeholders to gather requirements for the Flight Route Explorer. Identify the needs and expectations of users, including aviation enthusiasts, travelers, and researchers. Document functional and non-functional requirements, including features, performance, usability, and security.
2. **Data Collection and Analysis:**
    - Identify and collect relevant data sources for the project, including airport information, flight route data, and geographical data. Explore publicly available datasets, APIs, and databases to gather the required data. Analyze the data to understand its structure, format, and quality, and identify any preprocessing requirements.
3. **Data Preparation and Integration:**
    - Preprocess the collected data to ensure it is in a suitable format for use in the project. This may involve parsing CSV files, cleaning and formatting data, and converting units if necessary. Integrate the prepared data into the Flight Route Explorer application, ensuring seamless access and retrieval.
4. **Architecture and Design:**
    - Design the architecture of the Flight Route Explorer application, including its components, modules, and layers. Choose appropriate technologies and frameworks for front-end and back-end development, database management, and mapping functionalities. Create wireframes and mockups to visualize the user interface (UI) and user experience (UX) design.
5. **Development:**
    - Develop the Flight Route Explorer application according to the defined architecture and design. Implement the front-end components using HTML, CSS, and JavaScript, and incorporate mapping functionalities using libraries like CesiumJS. Develop the back-end components using appropriate programming languages and frameworks, and ensure seamless integration with the front-end.
6. **Testing:**
    - Conduct comprehensive testing of the Flight Route Explorer application to ensure its functionality, performance, and usability. Test for functional correctness, including searching for airports,

displaying locations, calculating shortest paths, and showing distances. Perform usability testing to evaluate the user interface and user experience. Conduct performance testing to assess the application's speed and responsiveness.

7. **Deployment:**
    - Deploy the Flight Route Explorer application to a web server or hosting platform to make it accessible to users. Configure the deployment environment and ensure that all dependencies are met. Monitor the deployment process to ensure smooth and successful deployment. Perform post-deployment testing to verify the application's availability and performance.
8. **Documentation and Training:**
    - Create comprehensive documentation for the Flight Route Explorer application, including user guides, technical documentation, and API references. Document the application architecture, data sources, and any dependencies. Provide training and support to users and stakeholders to help them effectively use the application.

HARDWARE REQUIREMENTS

- **Computer:** A modern computer with a decent processor (e.g., Intel Core i5 or equivalent)
- **Memory:** At least 8GB of RAM, preferably more for smoother performance
- **Graphics:** A dedicated or integrated graphics processor (GPU) for improved 3D visualization
- **Storage:** Sufficient storage space, with SSD recommended for faster data access
- **Internet Connection:** A stable internet connection for accessing online resources and services
- **Peripherals:** Additional peripherals like monitors, input devices, etc., for enhanced productivity and user interaction.

SOFTWARE REQUIREMENTS

- **Operating System:** Compatible with Windows, macOS, or Linux distributions
- **Web Browser:** Latest version of popular web browsers like Google Chrome, Mozilla Firefox, or Microsoft Edge for running web applications
- **Text Editor:** Any code editor or integrated development environment (IDE) such as Visual Studio Code, Sublime Text, or Atom for editing HTML, CSS, and JavaScript files
- **Web Server:** Flask framework for Python is required if hosting the web application using a Python-based backend
- **CesiumJS Library:** Required for 3D visualization, make sure to include CesiumJS library in your project
- **CSV Parser:** If handling CSV data, ensure availability of a suitable CSV parsing library or function in your programming environment.

DATA COLLECTION/PREPARATION

The data collection and preparation step involved gathering information from a comprehensive dataset containing details of 8288 airports worldwide. Each airport entry included essential information such as the airport's ID, name, city, IATA code, ICAO code, latitude, and longitude coordinates. This data was structured in a CSV file format, making it suitable for processing and analysis.

Upon obtaining the CSV file, the data was parsed and organized using JavaScript. The parsing process involved splitting the CSV data into individual lines and extracting relevant information for each airport entry. This information was then formatted into JavaScript objects, making it easier to work with programmatically.

During the preparation phase, data cleansing and validation techniques were applied to ensure data accuracy and consistency. This involved handling missing or erroneous data entries, converting string-based latitude and longitude values to numeric formats, and verifying the integrity of airport codes and geographical coordinates.

	A	B	C	D	E	F	G
1	id	name	city	iata	icao	lat	lng
2	1	London He	London	LHR	EGLL	51.471	-0.462
3	2	Hartsfield	Atlanta	ATL	KATL	33.637	-84.428
4	3	Beijing Cap	Beijing	PEK	ZBAA	40.08	116.585
5	4	Charles de	Paris	CDG	LFPG	49.013	2.55
6	5	Sydney Kin	Sydney	SYD	YSSY	-33.946	151.177
7	6	Frankfurt a	Frankfurt	FRA	EDDF	50.033	8.571
8	7	Chek Lap K	Hong Kong	HKG	VHHH	22.309	113.915
9	8	Dubai Inte	Dubai	DXB	OMDB	25.253	55.364
10	9	Tokyo Han	Tokyo	HND	RJTT	35.552	139.78
11	10	Melbourne	Melbourne	MEL	YMML	-37.673	144.843
12	11	Shanghai P	Shanghai	PVG	ZSPD	31.143	121.805
13	12	Shanghai H	Shanghai	SHA	ZSSS	31.198	121.336
14	13	John F Ken	New York	JFK	KJFK	40.64	-73.779
15	14	Newark Lil	New York	EWR	KEWR	40.693	-74.169
16	15	La Guardia	New York	LGA	KLGA	40.777	-73.873
17	16	Guangzhou	Guangzhou	CAN	ZGGG	23.392	113.299
18	17	Singapore	Singapore	SIN	WSSS	1.35	103.994
19	18	Shenzhen I	Shenzhen	SZX	ZGSZ	22.639	113.811

SYSTEM DESIGN

1. **Homepage:**
    - Upon loading the application, users are greeted with the homepage, which contains the primary interface elements.
2. **Search Bar:**
    - The central feature of the homepage is a search bar where users can enter the names of airports they want to explore.
3. **Search Results Display:**
    - As users type in the search bar, the application provides autocomplete suggestions based on the airports available in the database. This feature enhances usability by helping users find the airports they are looking for more quickly.
4. **Map Display:**
    - Below the search bar, there is a dynamic map display powered by CesiumJS. As users search for airports, the map updates to show the locations of the airports they've selected.
5. **Airport Information Cards:**
    - For each airport selected, an information card appears on the side of the map, providing details such as the airport name, city, IATA code, and ICAO code.
6. **Shortest Path Calculation:**
    - After selecting two or more airports, users can click on a "Go" button to calculate and display the shortest path between them. This functionality provides valuable insights into potential flight routes.
7. **Path Visualization:**
    - The shortest path between selected airports is visualized on the map using a polyline with a distinct color (e.g., yellow). This visual representation helps users understand the route more intuitively.
8. **Distance Calculation:**
    - Additionally, the application calculates and displays the distance (in kilometers) between the selected airports. This information is presented in a separate section of the webpage, providing users with additional context about the chosen flight route.
9. **Responsive Design:**
    - The UI/UX design of the FlightRouteExplorer project is responsive, ensuring a seamless experience across various devices and screen sizes. Whether users access the application from a desktop computer, tablet, or smartphone, they can interact with the

features comfortably.

10. **Error Handling:**
    - The application incorporates error handling mechanisms to address scenarios such as invalid user input or failed data loading. Clear error messages are displayed to guide users and troubleshoot issues effectively.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Flight Path</title>
    <!-- Include the CesiumJS library -->
    <script src="https://cesium.com/downloads/cesiumjs/releases/1.82/Build/Cesium/Cesium.js"></script>
    <link href="https://cesium.com/downloads/cesiumjs/releases/1.82/Build/Cesium/Widgets/widgets.css" rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='DDstyle.css') }}">
</head>
<body>
    <div class="dropdown">
        <!-- Search input -->
        <input type="text" class="search-input" id="airportSearch" onkeyup="filterDropdown()" placeholder="Enter Airport Name">
        <button onclick="searchAirport()">Search</button>
        <button onclick="displayShortestPath()">Go</button>
        <button id="loadAirportsButton">Load Airports</button>
        <!-- Dropdown list -->
        <div class="dropdown-list" id="dropdown-list">
        </div>
        <div id="distance"></div>
      </div>
      <script src="{{ url_for('static', filename='js/Dropdown.js') }}"></script>
    <script src="{{ url_for('static', filename='js/Search.js') }}"></script>
    <div id="cesiumContainer" style="width: 100%; height: 100%;"></div>
    <div id="customCreditContainer"></div>
    <script src="{{ url_for('static', filename='js/script.js') }}"></script>
    <script src="{{ url_for('static', filename='js/Airports.js') }}"></script>
</body>
</html>
```

HOME HTML FILE WITH THE BASIC STRUCTURE OF PROJECT

ALGORITHM AND CONCEPT

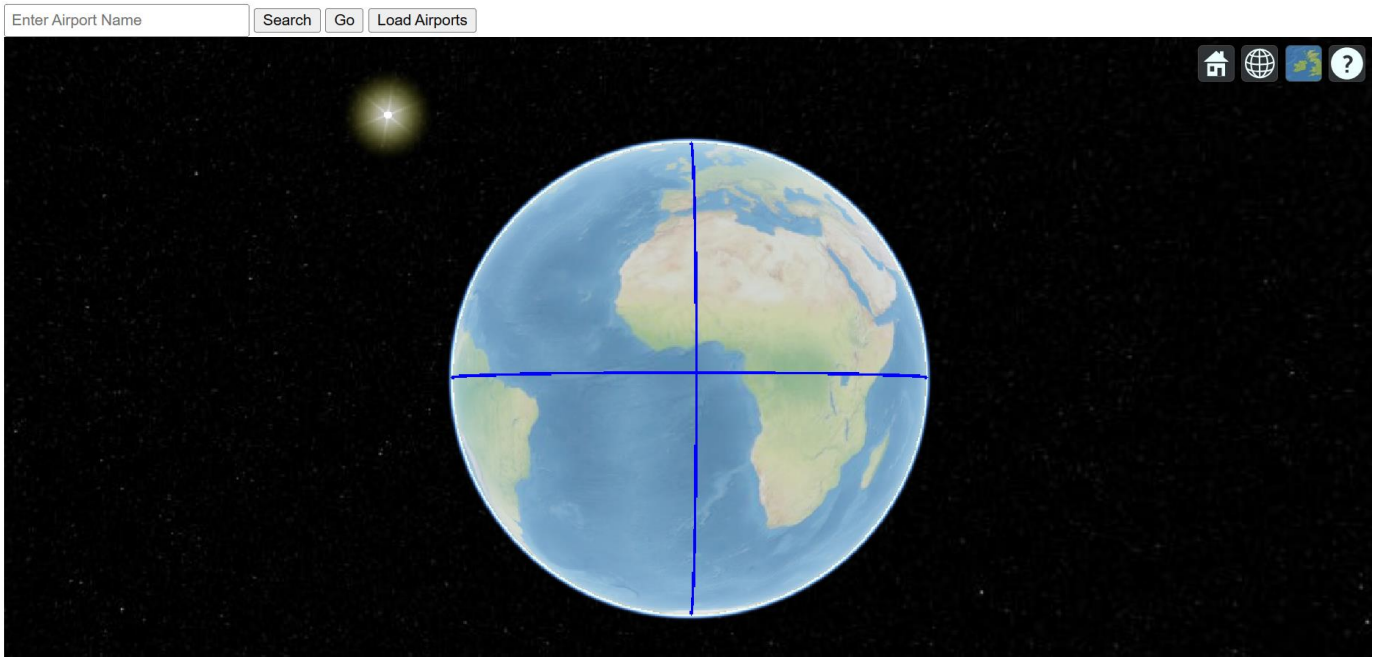
In the Flight Route Explorer project, non-Euclidean geometry, the Haversine formula, and Dijkstra's algorithm play crucial roles in different aspects of the application:

1. **Non-Euclidean Geometry**:
    - The globe is represented as a three-dimensional object, which does not conform to the rules of Euclidean geometry that we are accustomed to in flat surfaces. This non-Euclidean geometry is essential for accurately representing the Earth's surface and calculating distances and paths on it.
    - In non-Euclidean geometry, straight lines (geodesics) are defined as the shortest path between two points on a curved surface, such as the Earth's surface. This concept is utilized when calculating distances between airports along the Earth's surface.
2. **Haversine Formula**:
    - The Haversine formula is used to calculate the great-circle distance between two points on a sphere given their latitude and longitude. This formula takes into account the curvature of the Earth's surface to provide a more accurate distance measurement.
    - In the Flight Route Explorer project, the Haversine formula is used to calculate the distance between airports. This distance calculation is crucial for various functionalities, such as finding the shortest path between airports and displaying the distance between two selected airports.
3. **Dijkstra's Algorithm**:
    - Dijkstra's algorithm is a graph traversal algorithm used to find the shortest path between nodes in a graph with non-negative edge weights. It operates by iteratively selecting the node with the smallest tentative distance from a set of unvisited nodes until all nodes have been visited.
    - In the Flight Route Explorer project, Dijkstra's algorithm is employed to calculate the shortest path between selected airports. The airports are represented as nodes in a graph, with the distances between them serving as edge weights. By applying Dijkstra's algorithm, the project can efficiently determine the shortest route between any pair of selected airports on the Earth's surface.

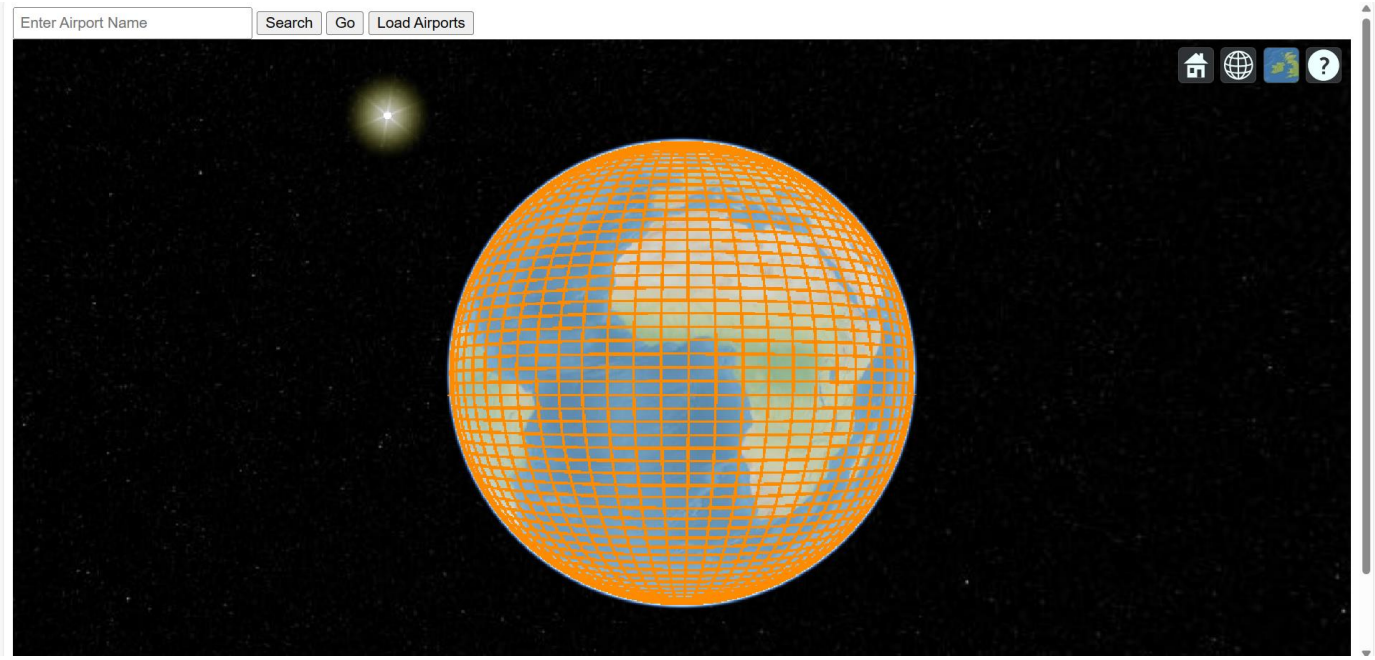
```
function calculateHaversineDistance(lat1, lon1, lat2, lon2) {
    // Radius of the Earth in kilometers
    const R = 6371;

    // Convert latitude and longitude from degrees to radians
    const lat1Rad = toRadians(lat1);
    const lon1Rad = toRadians(lon1);
    const lat2Rad = toRadians(lat2);
    const lon2Rad = toRadians(lon2);

    // Haversine formula
    const dLat = lat2Rad - lat1Rad;
    const dLon = lon2Rad - lon1Rad;
    const a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +
              Math.cos(lat1Rad) * Math.cos(lat2Rad) *
              Math.sin(dLon / 2) * Math.sin(dLon / 2);
    const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
    const distance = R * c;

    return distance;
}
```

```
// Dijkstra's algorithm
while (unvisitedAirports.size > 0) {
    let minAirportId = null;
    for (const airportId of unvisitedAirports) {
        if (!minAirportId || distances[airportId] < distances[minAirportId]) {
            minAirportId = airportId;
        }
    }
    const currentAirport = minAirportId;
    visited[cur                    = true;
    unvisitedAi   const neighborAirport: string
    for (const neighborAirport in graph[currentAirport]) {
        const distance = distances[currentAirport] + graph[currentAirport][neighborAirport];
        if (distance < distances[neighborAirport]) {
            distances[neighborAirport] = distance;
            previous[neighborAirport] = currentAirport;
        }
    }
}
```

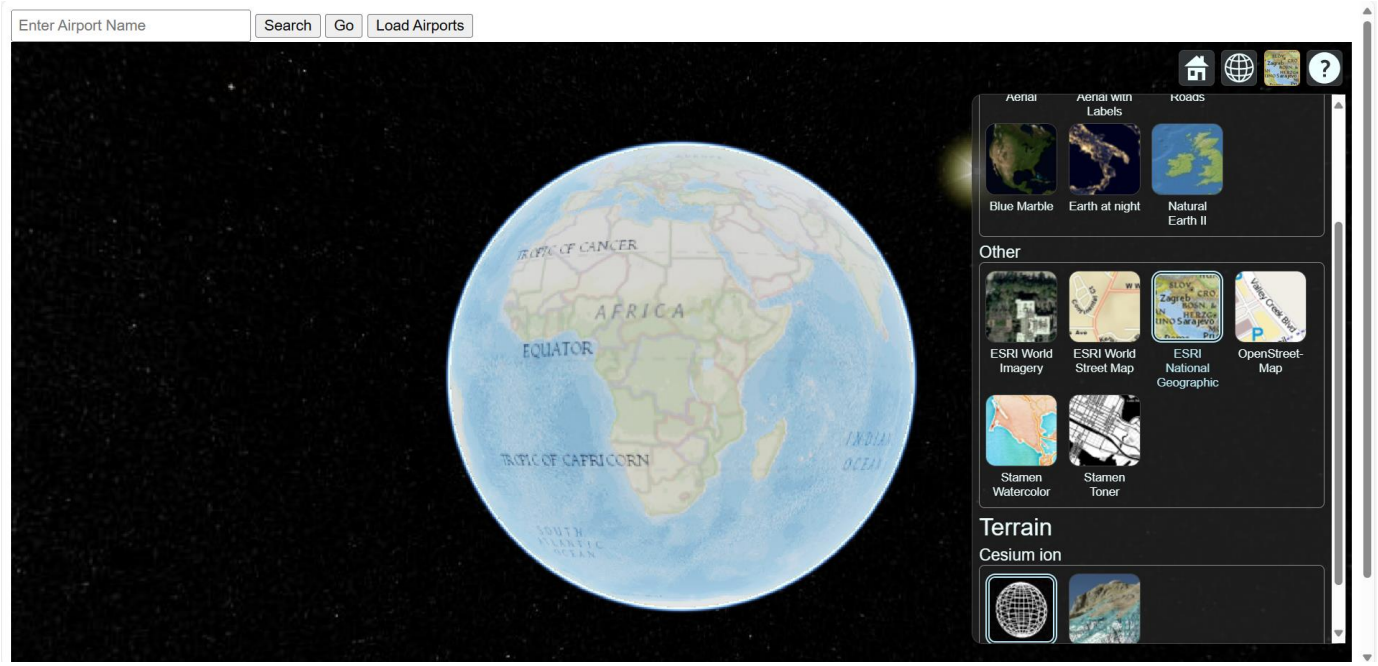
IMPLEMENTATION



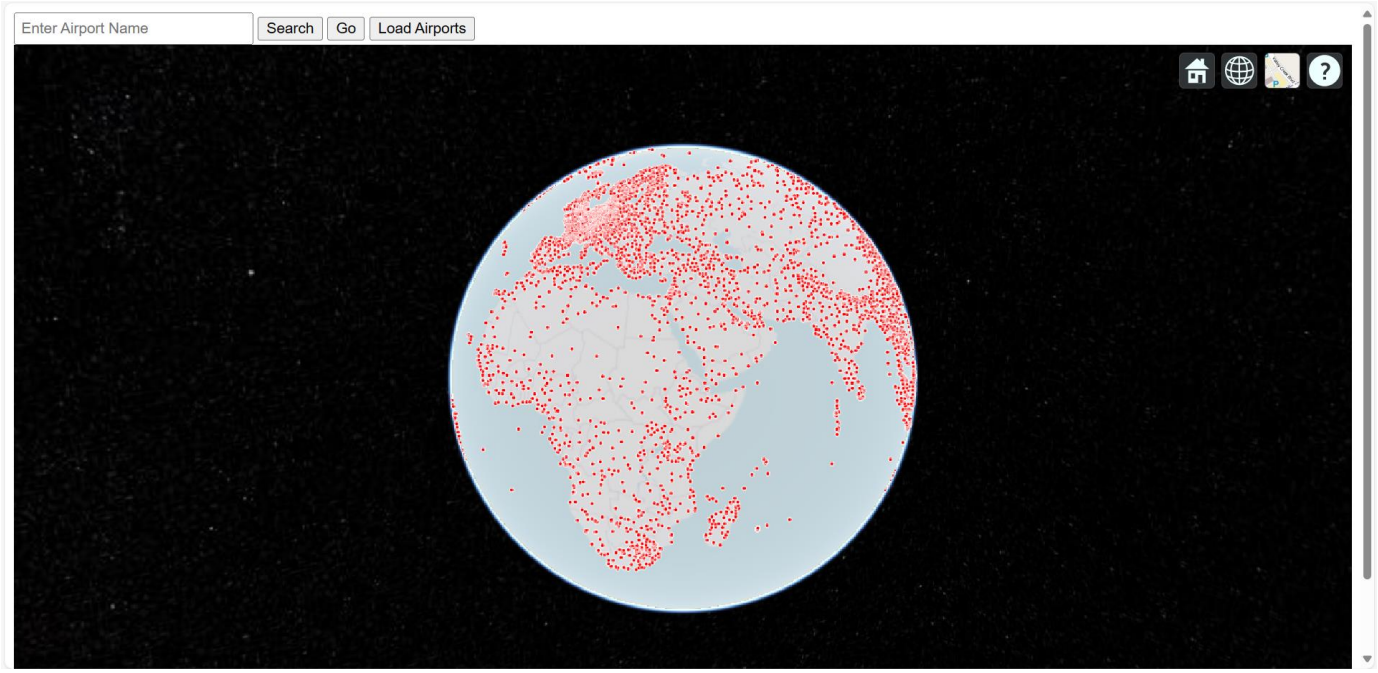
Home Page: Earth With Equator And Prime Meridian



Earth With High Resolution Grid Of Longitudes And Latitudes

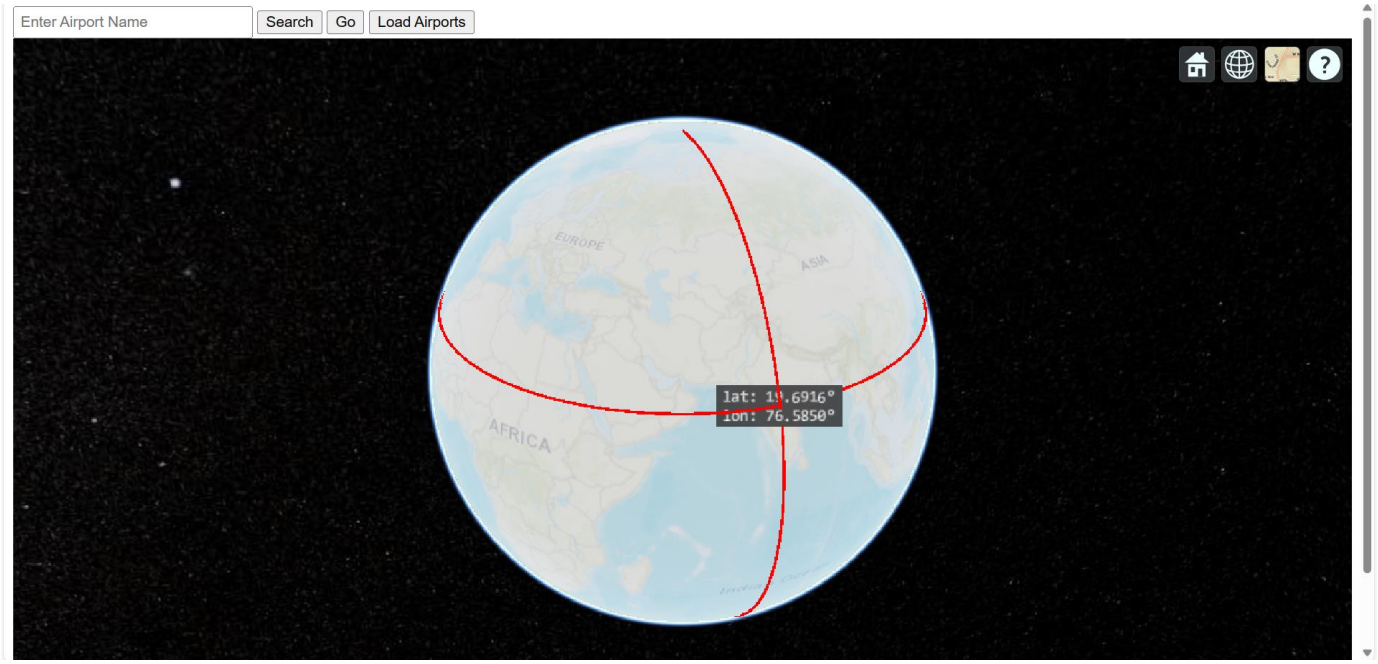


15 Different Base Layers Available

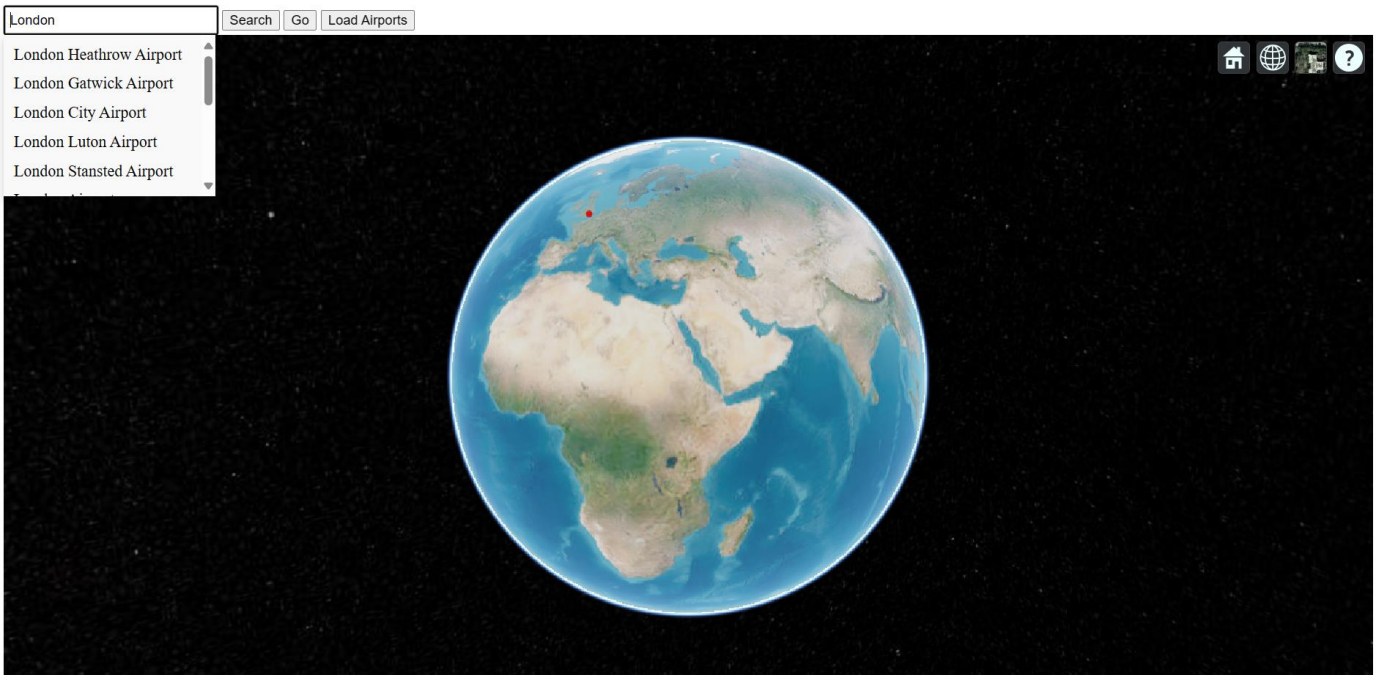


Shows The Location 8288 Airport In The World

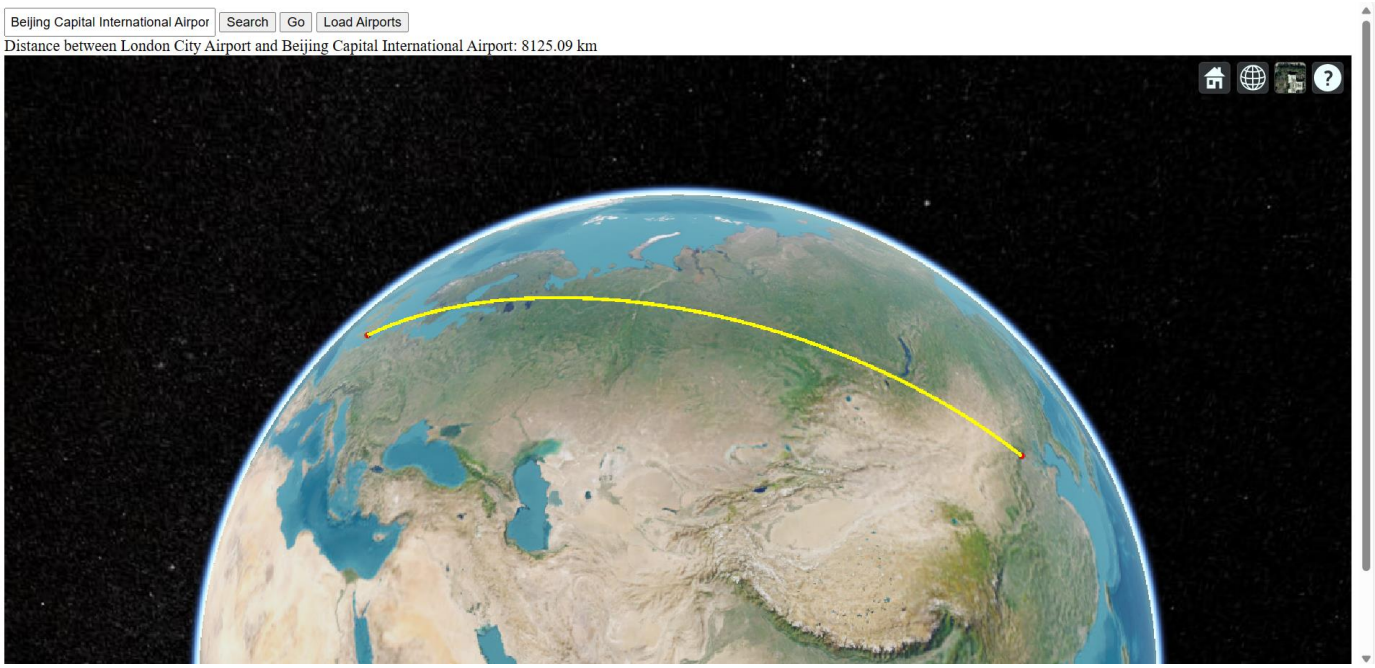
Detailed Zoom-In



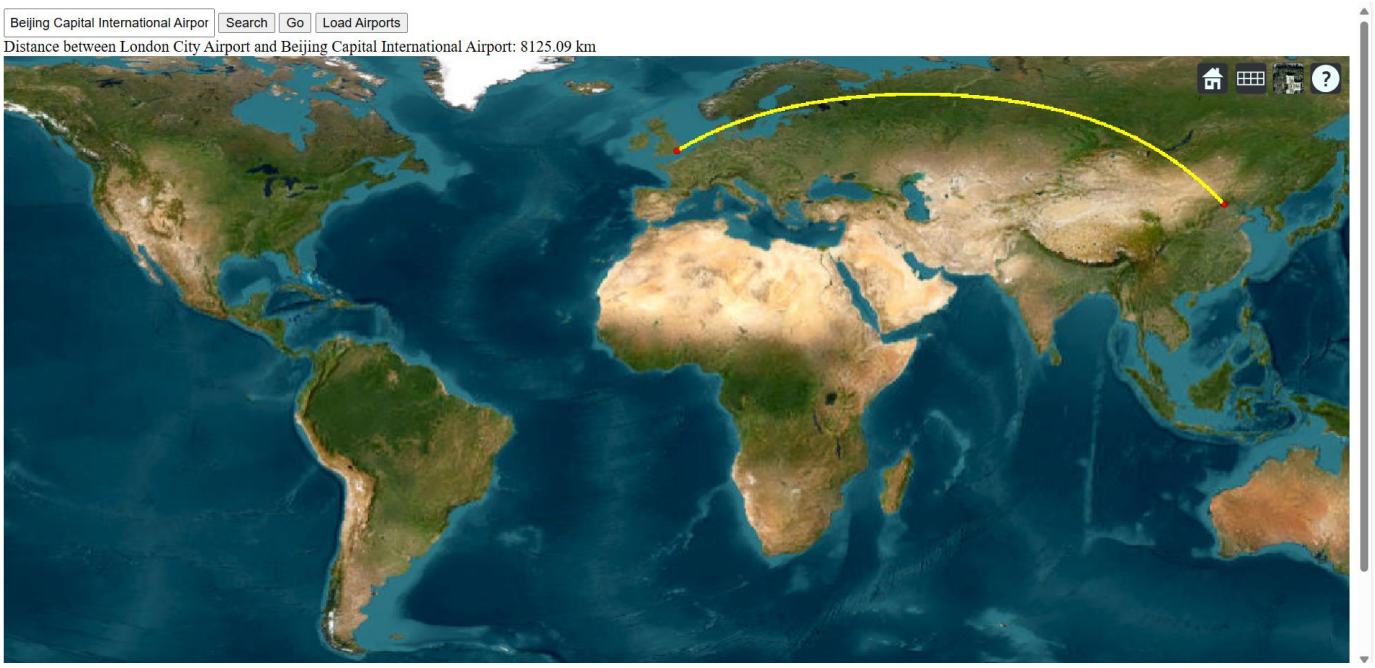
Displays The Coordinates Of Any Point You Click On Earth



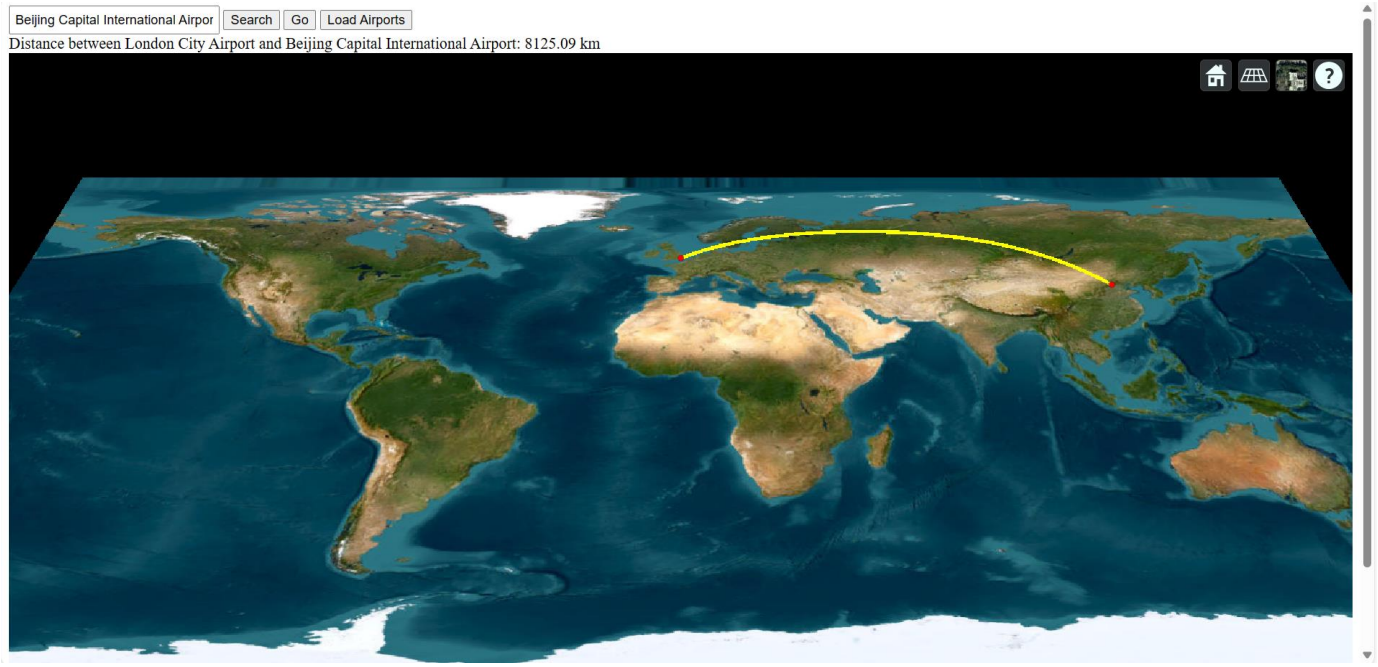
Search For Location Of Specific Airport Using “Search” Button



Visualize The Path And Displays The Distance Between 2 Airports Using “Go” Button



Visualizes The Path In 2d View



Visualizes The Path In Columbus View

LIMITATIONS

1. **No Backend Integration: The project operates solely on the client-side without a backend server, limiting functionalities like user authentication, data storage, or dynamic updates.**
2. **Limited Error Handling: Error handling in the project is basic and may not adequately handle unexpected scenarios or input errors.**
3. **No Backend Integration: The project operates solely on the client-side without a backend server, limiting functionalities like user authentication, data storage, or dynamic updates.**
4. **Limited Error Handling: Error handling in the project is basic and may not adequately handle unexpected scenarios or input errors.**
5. **Lack of Mobile Optimization: The user interface may not be optimized for mobile devices, potentially reducing usability on smaller screens.**
6. **No Data Validation: Input data, such as airport names, is not thoroughly validated, which could lead to inaccuracies or unexpected behavior.**

FUTURE SCOPE

1. **Multi-leg Flights: Enable users to plan and visualize multi-leg flights by selecting and connecting multiple airports.**
2. **Flight Information Overlay: Incorporate real-time flight data to display flight paths, current positions, and other relevant information on the globe.**
3. **Customization Options: Provide users with customization options such as choosing different colors for airport markers, adjusting the thickness of flight paths, and toggling different layers on the globe.**
4. **Collaborative Planning: Enable collaborative flight planning by allowing users to share their routes with others, collaborate on plans, and see real-time updates from multiple users.**
5. **Mobile Application: Develop a mobile application version of Flight Route Explorer for on-the-go flight planning and visualization, with features optimized for mobile devices.**
6. **Integration with External APIs: Integrate with external APIs such as weather services, airline databases, and airport information systems to provide users with comprehensive flight planning tools and data.**
7. **Augmented Reality (AR) Integration: Explore the possibility of integrating augmented reality (AR) features to visualize flight paths and airport information in the real-world environment through AR-enabled devices.**
8. **Educational Tools: Develop educational tools and resources within Flight Route Explorer to help users learn about aviation, geography, and navigation principles while using the application.**
9. **Rhumb Line Navigation: Integrate Rhumb Line navigation capabilities to provide users with the option to plan routes along constant compass headings, simplifying navigation over long distances, particularly for oceanic crossings.**

CONCLUSION

Flight Route Explorer is a sophisticated tool designed to facilitate exploration and navigation of flight routes. By harnessing data sourced from CSV files containing comprehensive airport information, this project offers users an immersive experience in discovering, visualizing, and analysing air travel possibilities.

The system's functionality begins with a user-friendly interface, allowing individuals to easily search for airports of interest. Leveraging JavaScript and Flask, users can input specific airport names, which are then matched against the stored dataset. Upon successful identification, the system displays the location of the selected airport on a dynamic globe interface powered by CesiumJS. This visualization aspect provides users with a spatial understanding of airports' geographical positions, aiding in route planning and decision-making processes.

One of Flight Route Explorer's key features is its ability to calculate the shortest path between two or more selected airports. Utilizing Dijkstra's algorithm, the system determines the most efficient route, considering factors such as distance and connectivity. This functionality is invaluable for travellers seeking optimal travel routes and logistics planners optimizing air cargo shipments.

Moreover, Flight Route Explorer offers an additional feature, providing users with essential information about the distance between two selected airports. This feature enhances users' understanding of route distances, aiding in itinerary planning and decision-making.

REFERENCES

1. Calculate distance b/w two places with Haversine formula
   https://www.youtube.com/watch?v=nsVsdHeTXIE

2. Non-Euclidean Geometry Explained
   https://www.youtube.com/watch?v=zQo_S3yNa2w

3. Great Circles and Rhumb Lines - Types of Routes
   https://www.youtube.com/watch?v=3BF_ZKfJiso

4. List of map projections
   https://en.wikipedia.org/wiki/List_of_map_projections

5. Flask. (n.d.). Flask. Retrieved from https://flask.palletsprojects.com/

6. Information about great circles, geodesics, and spherical trigonometry from https://www.wolfram.com/mathematica/

7. Csv Parsing from https://www.papaparse.com/docs

8. "Reading Files Using JavaScript." MDN Web Docs. Retrieved from
   https://developer.mozilla.org/en-US/docs/Web/API/File/Using_files_from_web_applications

9. "Reading Files Using JavaScript." MDN Web Docs. Retrieved from
   https://developer.mozilla.org/en-US/docs/Web/API/File/Using_files_from_web_applications

GIT-HUB LINK

Link for project access :
https://github.com/carefreecherry/Flight-Route-Explorer