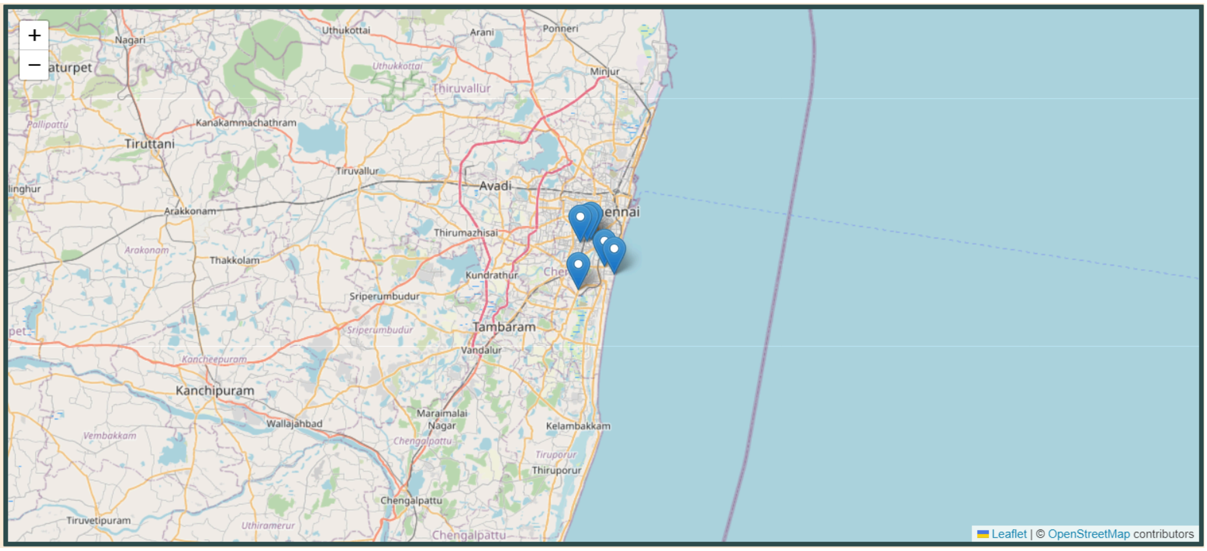
Interactive Map App (GitHub Repo)

Project Overview

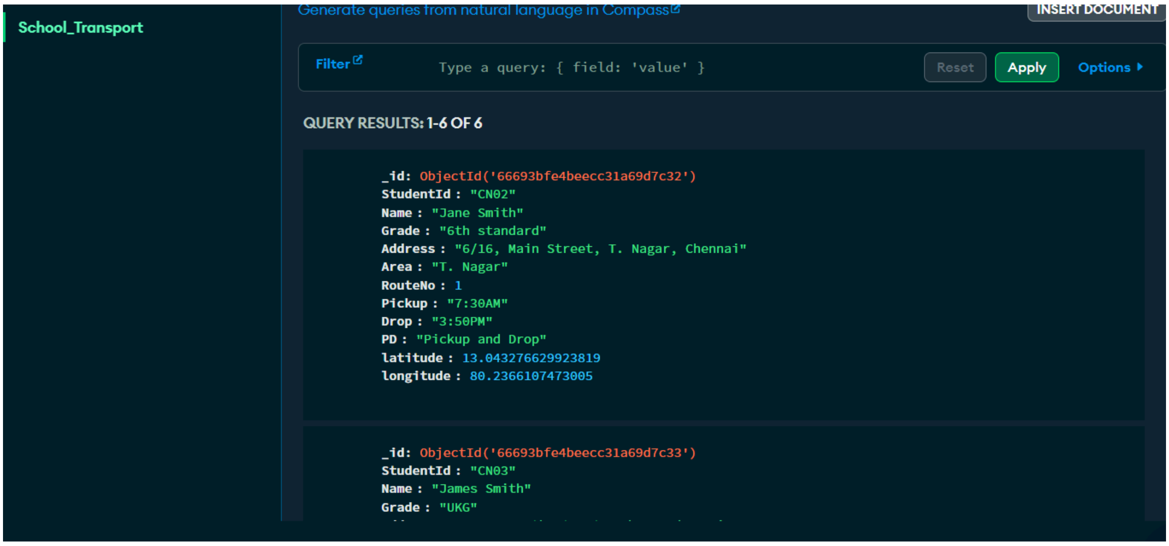
This project is an **interactive web application** designed to **assist school bus drivers in efficiently locating and managing student pickup points and route planning**. By utilizing a **map interface**, drivers can **search for student information by name, grade, student ID, address, and area**. The application **visualizes student locations on a map**, allowing drivers to easily navigate and **access relevant details** for each student.

Key Features

- **Map Display**: The application features a dynamic map using Leaflet.js to visualize student locations.

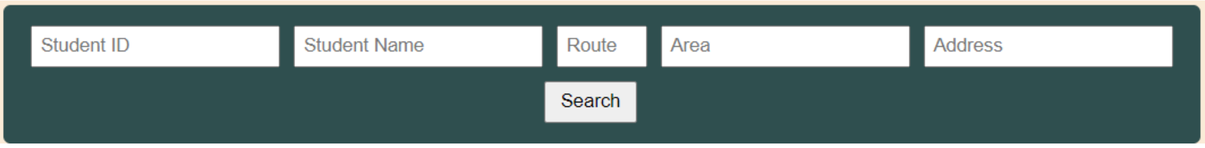


- **MongoDB Atlas**: The student data is stored securely on a MongoDB Atlas Database, thus ensuring that only authorized members can access the data.

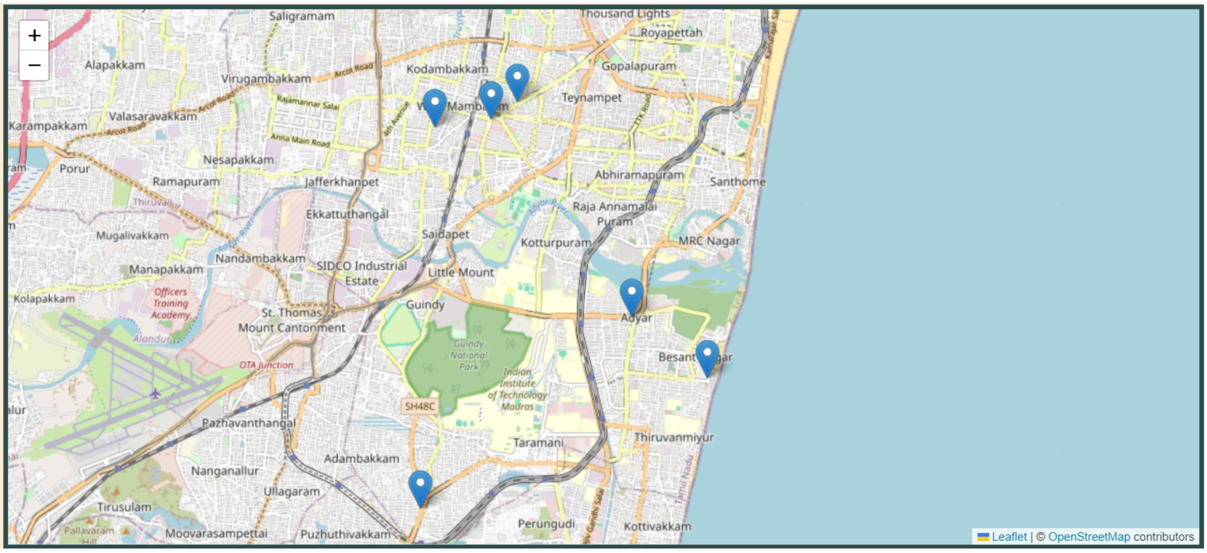


```
Type a query: { field: 'value' }

QUERY RESULTS: 1-6 OF 6

_id: ObjectId('66693bfe4beecc31a69d7c32')
StudentId : "CN02"
Name : "Jane Smith"
Grade : "6th standard"
Address : "6/16, Main Street, T. Nagar, Chennai"
Area : "T. Nagar"
RouteNo : 1
Pickup : "7:30AM"
Drop : "3:50PM"
PD : "Pickup and Drop"
latitude : 13.043276629923819
longitude : 80.2366107473005

_id: ObjectId('66693bfe4beecc31a69d7c33')
StudentId : "CN03"
Name : "James Smith"
Grade : "UKG"
```

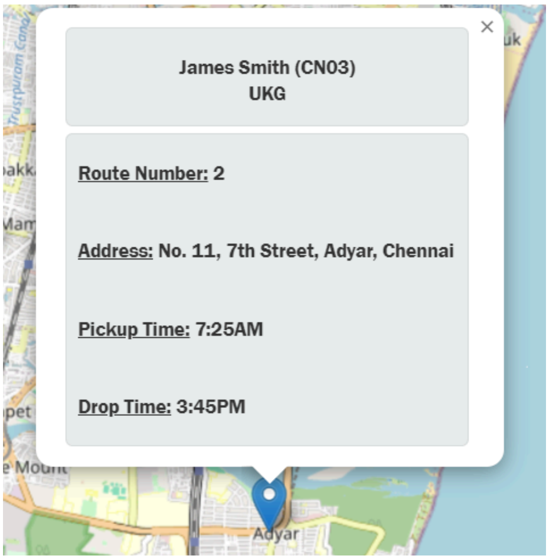
- **Search Functionality**: Drivers can search for students by name, grade, student ID, address, and area.



- **Marker Placement**: Each student's address is marked on the map with markers.



- **Interactive Pop-ups**: Clicking on a marker reveals a popup with detailed student information.


×

James Smith (CN03)
UKG

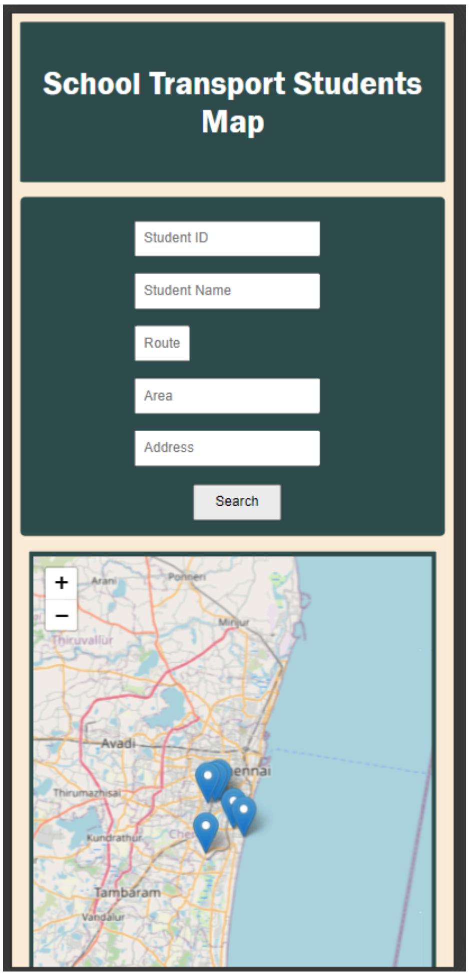
Route Number: 2

Address: No. 11, 7th Street, Adyar, Chennai

Pickup Time: 7:25AM

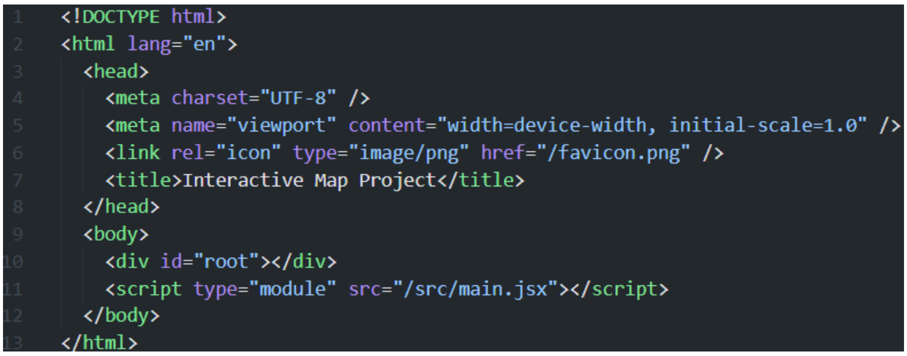
Drop Time: 3:45PM

- **Responsive Design**: The map and user interface are optimized for both desktop and mobile devices.



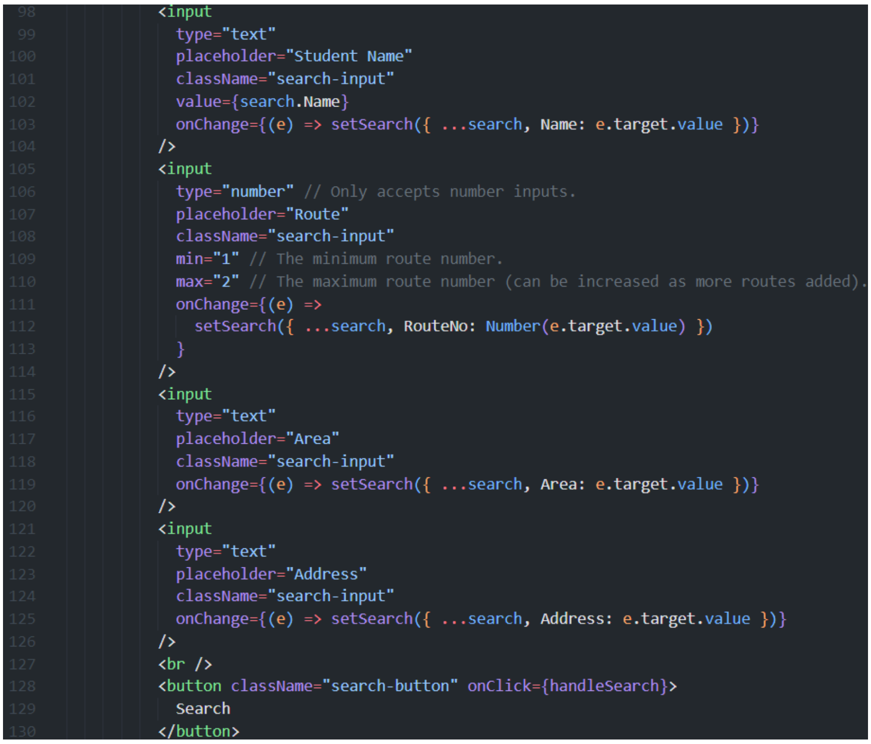
Technologies Used

- **HTML5/JSX**: For structuring the web application.



```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="icon" type="image/png" href="/favicon.png" />
    <title>Interactive Map Project</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

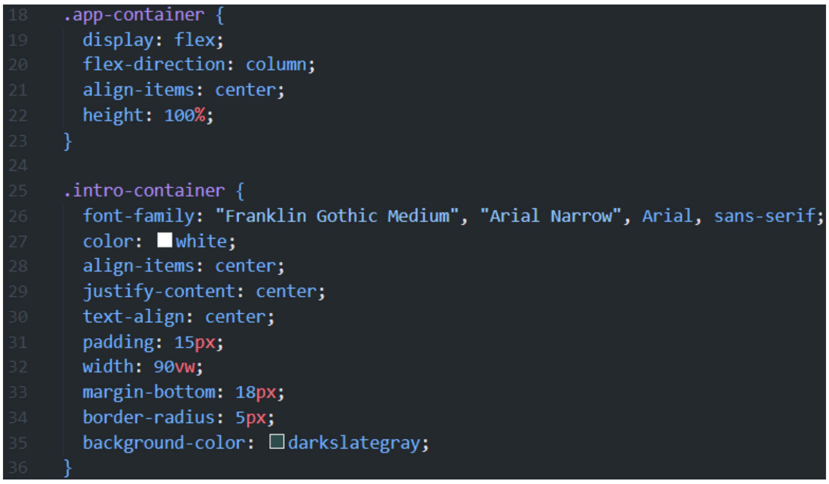
HTML



```jsx
<input
  type="text"
  placeholder="Student Name"
  className="search-input"
  value={search.Name}
  onChange={(e) => setSearch({ ...search, Name: e.target.value })}
/>
<input
  type="number" // Only accepts number inputs.
  placeholder="Route"
  className="search-input"
  min="1" // The minimum route number.
  max="2" // The maximum route number (can be increased as more routes added).
  onChange={(e) =>
    setSearch({ ...search, RouteNo: Number(e.target.value) })
  }
/>
<input
  type="text"
  placeholder="Area"
  className="search-input"
  onChange={(e) => setSearch({ ...search, Area: e.target.value })}
/>
<input
  type="text"
  placeholder="Address"
  className="search-input"
  onChange={(e) => setSearch({ ...search, Address: e.target.value })}
/>
<br />
<button className="search-button" onClick={handleSearch}>
  Search
</button>
```

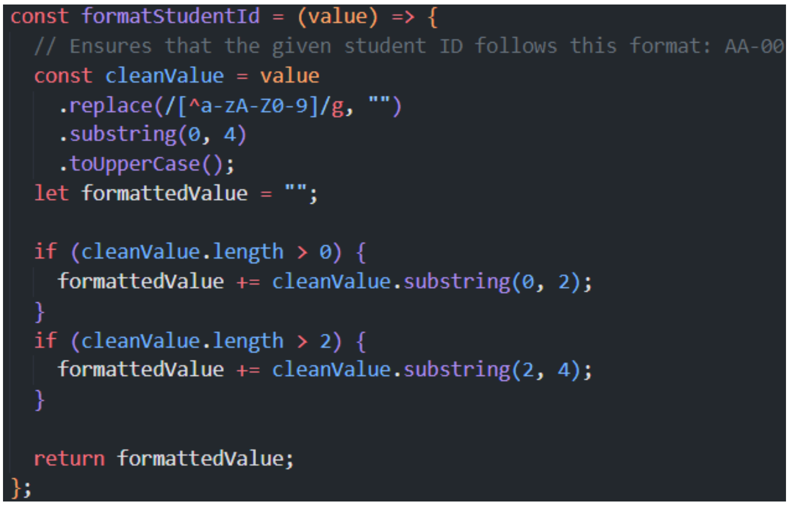
JSX

- **CSS3**: For styling and responsive design.



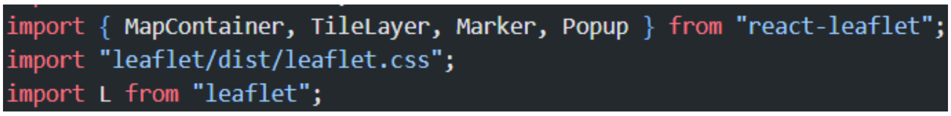
```css
.app-container {
  display: flex;
  flex-direction: column;
  align-items: center;
  height: 100%;
}

.intro-container {
  font-family: "Franklin Gothic Medium", "Arial Narrow", Arial, sans-serif;
  color: □white;
  align-items: center;
  justify-content: center;
  text-align: center;
  padding: 15px;
  width: 90vw;
  margin-bottom: 18px;
  border-radius: 5px;
  background-color: ■darkslategray;
}
```

- **JavaScript (React.js)**: For dynamic functionality and interactive features.

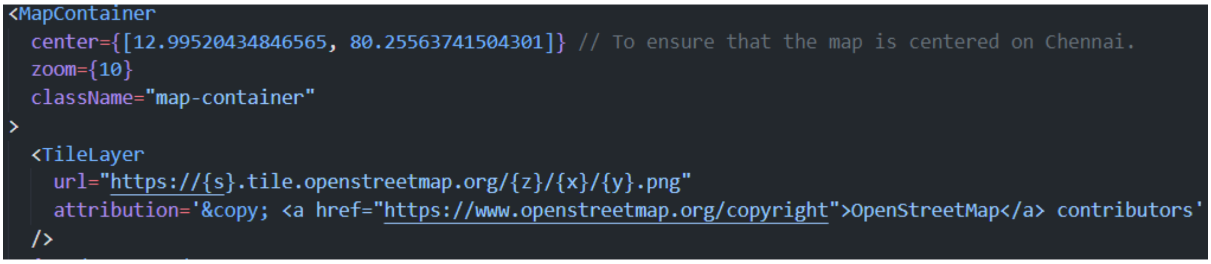


```javascript
const formatStudentId = (value) => {
  // Ensures that the given student ID follows this format: AA-00
  const cleanValue = value
    .replace(/[^a-zA-Z0-9]/g, "")
    .substring(0, 4)
    .toUpperCase();
  let formattedValue = "";

  if (cleanValue.length > 0) {
    formattedValue += cleanValue.substring(0, 2);
  }
  if (cleanValue.length > 2) {
    formattedValue += cleanValue.substring(2, 4);
  }

  return formattedValue;
};
```

- **Leaflet.js**: For map and marker features.

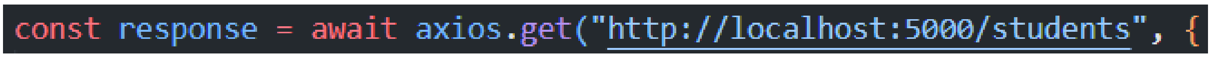


```javascript
import { MapContainer, TileLayer, Marker, Popup } from "react-leaflet";
import "leaflet/dist/leaflet.css";
import L from "leaflet";
```



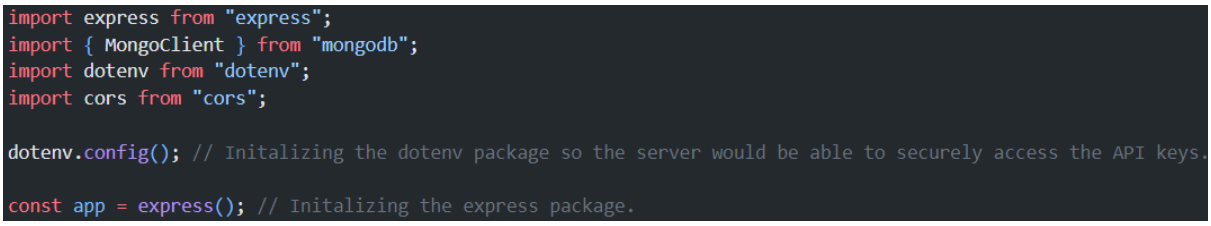
```jsx
<MapContainer
  center={[12.99520434846565, 80.25563741504301]} // To ensure that the map is centered on Chennai.
  zoom={10}
  className="map-container"
>
  <TileLayer
    url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
    attribution='&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
  />
```

- **Axios**: For handling server requests from the frontend and to fetch data from the database.



```javascript
const response = await axios.get("http://localhost:5000/students", {
```

- **Express.js**: For working with APIs within the NodeJS framework.



```javascript
import express from "express";
import { MongoClient } from "mongodb";
import dotenv from "dotenv";
import cors from "cors";

dotenv.config(); // Initalizing the dotenv package so the server would be able to securely access the API keys.

const app = express(); // Initalizing the express package.
```

Project Structure

- **App.css**: The CSS file for styling the application.
- **App.jsx**: The main JavaScript file where the application logic and interactions are implemented.
- **main.jsx**: Used in conjunction with index.html to render the App.jsx into the DOM.

Setup Instructions

1. Clone the repository from GitHub.
2. Run *npm install* in the terminal while in the main directory.
3. Run *npm run dev* in the terminal while in the main directory.
4. Navigate to *localhost:5173* in a browser.
5. Ensure you have an active internet connection for Leaflet.js to load the map tiles.

Usage Instructions

1. Open the application in a web browser.
2. Use the search bar to find students by name, grade, student ID, address, or area.
3. Click on the markers on the map to view detailed information about each student.
4. Utilize the map controls to zoom in and out or navigate to different areas.

Project Outcomes

This project provided practical experience in front-end web development, particularly in working with interactive maps and markers. The skills gained include:

- Implementing map functionality using Leaflet.js.
- Enhancing user experience through interactive elements such as popups and responsive design.
- Developing a full-stack application using HTML, CSS, and JavaScript.