To set up and run flight tracking website project with the existing frontend and backend code, you'll need to follow an installation and execution process that involves setting up both environments, installing dependencies, and ensuring both parts communicate correctly. Below is a step-by-step guide based on the files provided

### Setting Up the Backend

1. Navigate to the Backend Directory:

Assuming your backend code resides in a separate directory (e.g., `flight-tracker-backend`), open a terminal and navigate to this directory.

```bash

cd path/to/flight-tracker-backend

```

2. \*\*Install Dependencies:\*\*

Your backend uses Node.js with several dependencies like `express`, `mongoose`, `cors`, `axios`, `bcryptjs`, `jsonwebtoken`, and `dotenv`. Install these by running:

```bash

npm install

```

This command installs all dependencies listed in your `package.json` file.

3. \*\*Environment Variables:\*\*

The backend requires setting up environment variables for database connections, API keys, and other sensitive information. Create a `.env` file in the root of your backend directory with the following variables (customize the values according to your setup):

```

PORT=5000

MONGODB\_URI=your\_mongodb\_uri

AVIATION\_EDGE\_API\_KEY=your\_api\_key

TWILIO\_ACCOUNT\_SID=your\_twilio\_sid

TWILIO\_AUTH\_TOKEN=your\_twilio\_token

TWILIO\_PHONE\_NUMBER=your\_twilio\_phone\_number

JWT\_SECRET=your\_jwt\_secret

```

4. \*\*Start the Backend Server:\*\*

With dependencies installed and environment variables set, start the server by running:

```bash

npm start

```

Or, if you're using `nodemon` for live reloading:

```bash

nodemon app.js

```

### Setting Up the Frontend

1. \*\*Navigate to the Frontend Directory:\*\*

Similarly, navigate to your frontend directory (e.g., `flight-tracker-frontend`) in a separate terminal window.

```bash

cd path/to/flight-tracker-frontend

```

2. \*\*Install Dependencies:\*\*

The frontend uses React along with libraries like `@mui/material`, `@chakra-ui/react`, and `react-router-dom`. Install all necessary dependencies by running:

```bash

npm install

```

This command reads your `package.json` file and installs the required packages.

3. \*\*Environment Variables for Frontend:\*\*

If your frontend needs to communicate with the backend on a specific port or domain, ensure you set the correct API base URL. You might not need a `.env` file for the frontend if you're using proxy settings or if the backend API URL is hardcoded. However, if needed, create a `.env` file in your frontend directory:

```

REACT\_APP\_API\_BASE\_URL=http://localhost:5000

```

Adjust the port number if your backend runs on a different port.

4. \*\*Start the Frontend Development Server:\*\*

Once all dependencies are installed, you can start the React development server by running:

```bash

npm start

```

This command will launch your frontend application, usually accessible at `http://localhost:3000`.

For Operation Search Flightno

1) Search FlightIata no in the search bar and click then it will give the output of flight details

2)For Airport Timetable, click timetable on navbar then type iata code of airport it will give departure and arrival details

3)For Searchbyroute,enter dep and arrival airport iata code and search it

it will give results of Flights travelling through these airport destinations.

4) For the Future schedule or flight history enter airport iata code of preferred place and date which should be one from future.

5)Airport Search gives airport details,Search airportname,cityname,airportcode or anything its available in homepage after search details will be shown.