**Aviation Data Explorer - README**

**Project Description**

**Aviation Data Explorer** is a modern web application designed to simplify the process of retrieving and visualizing aviation-related data. It consolidates various data sources into a user-friendly interface, helping pilots, aviation enthusiasts, and air traffic simmers quickly access:

* Airport information (ICAO/FAA)
* Weather (METAR/TAF)
* Charts and approach plates
* Preferred flight routes
* Live VATSIM virtual flight activity

**Target Browsers**

The application is designed to be fully responsive and works best on the following platforms:

* Google Chrome (desktop/mobile)
* Safari (iOS)
* Firefox (desktop/mobile)
* Microsoft Edge (desktop)
* Android Chrome Browser

**Developer Manual**

Welcome to the development documentation for the **Aviation Data Explorer**. This guide is intended for developers who want to maintain or extend the project.

**Installation**

**Prerequisites:**

* Node.js (v16+ recommended)
* A modern web browser
* Internet access for API usage

**Running the Application Locally**

Since this is a static front-end application:

1. Open index.html in your preferred browser **OR**
2. Run a local server using VS Code Live Server extension or Python:

python -m http.server 8000

1. Visit http://localhost:8000 in your browser.

**Description**

Aviation Data Explorer is a comprehensive web application designed for aviation enthusiasts, pilots, and professionals. It provides easy access to consolidated aviation data including airport information, charts, preferred routes, and live virtual flight data from VATSIM. The platform serves as a one-stop solution for aviation data that is typically scattered across multiple sources.

Key Features:

* Search airports by ICAO/FAA code
* Access airport charts and diagrams
* View preferred routes between airports
* Monitor real-time VATSIM flights and controllers
* Check current weather (METAR) and forecasts (TAF)

**Target Browsers**

The application is designed to work on modern browsers across multiple platforms:

**Desktop:**

* Chrome (latest version)
* Firefox (latest version)
* Safari (latest version)
* Edge (latest version)

**Mobile:**

* iOS Safari (iOS 13+)
* Android Chrome (latest version)

The responsive design ensures proper functionality on both desktop and mobile devices.

**Link to Developer Manual**

Developer Manual

**Developer Manual**

**Installation and Setup**

**Prerequisites**

* Node.js (v14 or higher)
* npm (comes with Node.js)
* Supabase account (for database)
* AviationAPI access (for data)

**Installation Steps**

1. Clone the repository:
   * git clone https://github.com/FaiqueAmeer/Final-Project.git

cd Final-Project

1. Install dependencies:
   * npm install
2. Create a .env file in the root directory with the following variables:
   * PORT=3000
   * SUPABASE\_URL=your\_supabase\_url
   * SUPABASE\_KEY=your\_supabase\_key

**Running the Application**

**Development Mode**

1. Start the server:
   * npm start
2. Open your browser and navigate to:
   * http://localhost:3000

**Production Mode**

For production deployment, consider using:

* PM2 for process management
* Nginx as a reverse proxy
* Environment variables for configuration

**API Endpoints**

**GET /api/airport/:code**

* **Description**: Retrieves airport information
* **Parameters**:
  + code: Airport ICAO/FAA code
* **Response**:

{

"icao": "string",

"faa": "string",

"name": "string",

"city": "string",

"state": "string",

"country": "string",

"latitude": "number",

"longitude": "number",

"elevation": "number",

"tower\_frequency": "string",

"atis\_frequency": "string",

"runways": ["string"]

}

**GET /api/charts/:code**

* **Description**: Retrieves airport charts
* **Parameters**:
  + code: Airport ICAO/FAA code
* **Response**:

[

{

"chart\_name": "string",

"chart\_code": "string",

"chart\_type": "string",

"chart\_date": "string",

"pdf\_path": "string"

}

]

**GET /api/routes/:origin/:destination**

* **Description**: Retrieves preferred routes between airports
* **Parameters**:
  + origin: Origin airport code
  + destination: Destination airport code
* **Response**:

[

{

"route": "string",

"type": "string",

"min\_altitude": "number",

"max\_altitude": "number",

"route\_string": "string"

}

]

**Known Issues and Future Development**

**Known Bugs**

1. Occasionally cached data may not refresh automatically
2. Some smaller airports may not have complete chart data
3. VATSIM data may occasionally timeout during high traffic periods

**Roadmap**

1. **Short-term (next release)**:
   * Add user authentication
   * Implement favorite airports feature
   * Improve mobile responsiveness
2. **Medium-term**:
   * Add flight planning tools
   * Integrate NOTAM data
   * Implement airport weather cameras
3. **Long-term**:
   * Add 3D airport visualization
   * Implement AI-powered route suggestions
   * Develop mobile applications

**Testing**

Currently, the application has manual testing procedures. Future development should include:

1. Unit tests for API endpoints
2. Integration tests for data fetching
3. End-to-end tests for user flows