To set up and run the application locally, follow these steps:

Frontend Setup:

1. Clone the repository to your local machine.
2. Navigate to the frontend directory in your terminal.
3. Open the `index.html` file in your preferred code editor.
4. Update the API key in the `script.js` file with your OpenWeatherMap API key.

Backend Setup:

1. Install Python if you haven't already.
2. Navigate to the backend directory in your terminal.
3. Create a virtual environment: `python -m venv venv`.
4. Activate the virtual environment:
   - On Windows: `venv\Scripts\activate`
   - On macOS/Linux: `source venv/bin/activate`
5. Install the required packages: `pip install -r requirements.txt`.
6. Set up your PostgreSQL database and update the database credentials in the `app.py` file.
7. Run the Flask application: `python app.py`.

Running the Application:

1. Open a web browser and navigate to `http://localhost:5000`.
2. You should see the weather dashboard with temperature, wind info, and humidity info.
3. The dashboard will update every 30 seconds with new weather information.

API Endpoints:

- `/weather_dashboard`: Endpoint to fetch weather information from OpenWeatherMap API.
- `/login`: Endpoint to authenticate users using session-based authentication.
- `/register`: Endpoint to register users using user authentication.

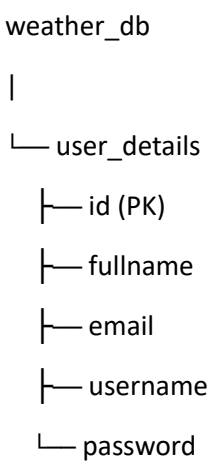
WebSocket Integration:

- The application uses long polling to update the dashboard with real-time weather information.

Database Schema:

- The application uses PostgreSQL for data storage. The database schema includes tables for storing user information.

Below is an Entity-Relationship Diagram (ERD) for the weather_db database with the user_details table:



In this ERD:

weather_db is the database containing the user_details table.

user_details table has columns id, fullname, email, username, and password.

id is the primary key (PK) of the user_details table.