DeefinityWeatherMail Project

Project Overview

This project, **DeefinityWeatherMail**, was developed by **Fares Naem** as part of the job selection process for **Deefinity Digital Consulting GmbH**. The primary objective of the project is to provide a weather forecast for Berlin (Germany) for the next 14 days, including min/max temperatures, chance of precipitation, and wind velocity. The forecast is sent to registered users via email, helping marketing managers plan outdoor events without needing to manually check the weather forecast.

Features

- **User Registration**: Users can register with their name, email, and city to receive a 14-day weather forecast.
- **Email Notifications**: Automatically send a weather forecast email for Berlin to the registered users.
- **User Login and Token-Based Authentication**: Registered users can log in securely and manage their subscriptions.
- **Subscription Management**: Users can unsubscribe or resubscribe from the service at any time.
- **Daily Weather Email**: Users receive an updated weather forecast every 14 days.

Project Structure

/DeefinityWeatherMail

├── weatherMail

| ├── weatherMail_utilities.py   # Handles weather data fetching and email sending functions.

├── templates

| ├── index.html   # HTML template for the web UI.

├── static

| ├── app.js   # JavaScript for frontend interactivity.

| ├── style.css   # Styling for the web page.

```
├── main.py.txt                 # The main FastAPI application code.
├── api_test.py.txt              # Unit tests for testing the API endpoints.
├── requirements.txt              # Project dependencies.
├── users.json                  # Stores registered user information.
```

How to Run the Project

1. Clone the Repository

git clone https://github.com/your-username/DeefinityWeatherMail.git

cd DeefinityWeatherMail

2. Install Dependencies

Ensure that you have Python 3.8+ installed. Install the required dependencies using:

pip install -r requirements.txt

3. Running the Application

To start the FastAPI server, run:

uvicorn main:app --reload

The app will be available at http://localhost:8000.

4. Access the Web Interface

Navigate to http://localhost:8000 in your browser to interact with the app via the user-friendly web interface.

5. Run Unit Tests

To test the API endpoints, run:

pytest api_test.py

Key Files and Functions

1. main.py

This is the core file of the application, containing the following features:

- **User Registration**: @app.post("/register")
- **Token-Based Authentication**: @app.post("/token")

- **Subscription Management**: Users can unsubscribe or resubscribe via /unsubscribe and /resubscribe endpoints.
- **Weather Email Dispatch**: Sends emails using send_weather_forecast_email.

2. weatherMail_utilities.py

- Contains helper functions to fetch weather data and send it via email.

3. index.html

- The main user interface for registration, login, and subscription management.

4. app.js

- JavaScript logic for form submissions (register, login, unsubscribe, resubscribe).

APIs

- **POST /register**: Register a new user.
- **POST /token**: Login and receive a JWT token.
- **GET /me**: Retrieve current user details.
- **POST /unsubscribe**: Unsubscribe from the weather service.
- **POST /resubscribe**: Resubscribe to the weather service.

License

This project is licensed under the MIT License.

Contact

For any issues or suggestions, feel free to reach out to **Fares Naem** (naemfares@gmail.com).