Weather Assistant

1. Project Description

Weather Assistant is a simple application built with Python and the Reflex framework. It aims to provide users with real-time weather information and clothing suggestions based on current weather conditions. Users can enter a city name to check current weather conditions and receive clothing recommendations based on the weather and temperature. Additionally, the application includes a wardrobe management feature, allowing users to add, edit, delete, and view wardrobe items.

2. User Stories

In our fast-paced lives, making daily wardrobe decisions can be a real challenge, especially when the weather is unpredictable. It's common for people to forget to wear a raincoat on a rainy day or carry sunglasses for bright sunny days. There's a growing need for quick access to current weather information coupled with tailored clothing recommendations based on the weather. However, it's hard to find an app that fulfills both these needs effectively. To address this issue, we've developed the "Weather Assistant" app. This innovative application seamlessly integrates weather forecasting with wardrobe management, providing not only accurate weather updates but also personalized clothing suggestions.

3. Tools

- Frontend: Reflex for the responsive and interactive UI. https://reflex.dev/
- Backend: Python for server-side logic
- APIs: OpenWeatherMap for weather data. https://openweathermap.org/current#name
- Database: SQLModel for database operations. SQLite for storing and managing wardrobe data.

4. Challenges & Solutions

- Obtaining real-time weather data was a challenge. We addressed this issue by utilizing the OpenWeatherMap API, ensuring accurate and timely weather updates.
- Developing a user-friendly interface was essential, so we selected the Reflex framework. As Reflex is quite new and specialized, and our team was inexperienced with it, we faced multiple technical issues initially. To address these challenges, we carefully reviewed Reflex's official documentation and utilized the AI support tool available on their website. This approach helped us successfully build our website.
- Database Management: Effectively managing the wardrobe database, particularly handling CRUD (Create, Read, Update, Delete) operations, was challenging. We opted for SQLModel and SQLite supported by the Reflex framework to build a simple database. However, there are still some issues to be resolved in fetching the latest data from the database.

5. Roadmap for Future Development

- Enhanced Clothing Management: Automatically adds clothes to the wardrobe through photo uploads and image recognition API. Generates clothing images using AI like Midjourney for inspiration when the wardrobe lacks suitable options.
- Weather Forecast Visualization: Introduce graphical representations of weather forecasts.
- Automatic City Detection: Automatically detect and display the user's weather information based on their location.
- Input Validation: Implement robust validation for user inputs.
- Multi-language Support: Expand the app's reach by supporting multiple languages.
- Social Sharing Feature: Allow users to share their outfit choices on social media platforms.
- Integration with Retail Platforms: Facilitate direct purchases of recommended items via retail platform integrations.