Running the Ayurvedic Recommender Project

This document provides step-by-step instructions to set up and run the Ayurvedic Recommender project locally with the new weather-based feature enabled.

Initial Setup

1. **Clone the Repository**

```
git clone https://github.com/Anindya2369/ayurveda.git

# Navigate to the project root
cd ayurveda
```

> Note: If you have already cloned the repository, you can skip this step and simply navigate to the project directory.

Overview

- **Project Structure**: The project consists of a React frontend and a Flask backend, organized in separate 'frontend' and 'back' directories within the ayurveda directory.
- **New Feature**: Weather-based Ayurvedic recommendations are integrated into the frontend (see `components/WeatherDisplay.jsx`) and supported by backend endpoints (see `routes/weather_routes.py` and `routes/recommendations_routes.py`).

Prerequisites

- **Repository**: Ensure you have cloned the repository and are in the root `ayurveda` directory before proceeding.
- **Node.js and npm**: Ensure you have Node.js (v14 or higher) and npm installed.
- **Python**: Python 3.8+ is required.
- **Package Manager**: Use `pip` (or conda if preferred) to manage Python dependencies.

Backend Setup

1. **Navigate to the Backend Directory**

   Ensure you are in the `ayurveda` directory, then navigate to the backend directory:

```
cd back
```

   All backend commands should be run from within the back directory.

> Note: It's recommended to open a dedicated terminal window for running the backend server.

2. **Create a Virtual Environment (Optional but Recommended)**

   - Using conda:

     ```
     conda create -n herbbot python=3.8 -y
     conda activate herbbot
     ```

   - Or using venv:

     ```
     python -m venv venv
     source venv/bin/activate  # On Windows use: venv\Scripts\activate
     ```

3. **Install Dependencies**

   ```
   pip install -r requirements.txt
   ```

4. **Configure Environment Variables**

   - Create a `.env` file in the back directory with the following content:

     ```
     PINECONE_API_KEY = "your_pinecone_api_key"
     OPENAI_API_KEY = "your_openai_api_key"
     SERP_API_KEY = "your_serp_api_key"
     WEATHER_API_KEY = "your_weather_api_key"  # Required for weather-based
     recommendations
     ```

5. **(Optional) Create Data Directory and Add PDF Files**

   - Create a Data directory in the backend folder and add your PDF files:

     ```
     mkdir -p Data
     # Copy your PDF files into the Data directory
     ```

6. **(Optional) Build the Knowledge Base**

   - If required for the new feature, run:

     ```
     python store_index.py
     ```

7. **Start the Backend Server**

```
python app.py
```

- The Flask backend should now be running on http://localhost:8080.

Frontend Setup

1. **Navigate to the Frontend Directory**

   Open another terminal window for the frontend commands. Ensure you are in the `ayurveda` root directory, then:

   ```
   cd frontend
   ```

   All frontend commands should be run from within the frontend directory.

2. **Install Node Dependencies**

   ```
   npm install
   ```

3. **Start the React Development Server**

   ```
   npm start
   ```

   - The React app will typically open at http://localhost:3000.

Verifying the New Feature

- **Ensure Both Services are Running**:
  - Verify that the backend is running on http://localhost:8080
  - Verify that the frontend is running on http://localhost:3000
  - Both services must be running concurrently for the application to work properly
- **Access the React Application**:
  - Open your browser and navigate to http://localhost:3000.
- **Test the Weather Feature**:
  - Click on the 'Weather' tab in the navigation bar. Verify that the app displays current weather data along with Ayurvedic recommendations based on the fetched weather information.
- **API Testing**:

  - You can directly test the weather API endpoint: http://localhost:8080/api/weather?city=Mumbai&country=IN
  - Test weather-based recommendations: http://localhost:8080/api/recommendations?dosha=pitta&city=Mumbai&country=IN

Troubleshooting

- **Terminal Setup**:
  - Ensure you have opened separate terminal windows for the backend and frontend.
  - Verify that each terminal is in the correct directory (`back` for backend, `frontend` for frontend).
- **Environment Variables**:
  - Confirm that the `.env` file exists in the `back` directory and contains all required API keys.
  - Ensure all API keys are valid and correctly formatted.
- **Data Directory**:
  - If using the knowledge base feature, verify that the `Data` directory exists in the `back` directory and contains the necessary PDF files.
- **General Issues**:
  - If issues with dependencies or API errors occur, confirm all prerequisites are installed and API keys in the `.env` file are correct.
  - Check the console logs for error messages and address them accordingly.
  - For weather-related features, ensure your WEATHER_API_KEY is valid and that you're providing valid city names in your requests.
  - If the weather feature isn't working, verify network connectivity as the application needs to make external API calls.

Additional Notes

- The weather feature integrates with the recommendations system to provide personalized Ayurvedic advice based on current weather conditions in your location.
- Weather data includes temperature, humidity, and general conditions which are mapped to Ayurvedic principles.
- The system can automatically determine the season based on weather data if not explicitly provided.

This document is designed to guide the local development setup. For full deployment instructions (e.g., Docker, EC2), please refer to the README.md file.