Product-Ingredient-Agent Documentation

1. Project Overview

The Product-Ingredient-Agent is a Streamlit application designed to analyze product ingredients using Google's Gemini AI and provide insights into food and personal care products. It leverages multimodal capabilities to process product images and extract ingredient information, offering detailed analysis, health implications, and healthier alternatives. The application is built with `phidata` for agentic workflow management, `Streamlit` for the user interface, and `Tavily Search` for external information retrieval.

Key Features:

- **Example Products**: Provides pre-loaded examples for quick demonstrations and testing.
- **Image Upload**: Allows users to upload their own product images for analysis.
- **Camera Capture**: Enables direct photo capture through the application for real-time analysis.
- **AI Analysis**: Utilizes Google Gemini 2.0 Flash for advanced ingredient analysis.
- **Ingredient Insights**: Delivers detailed analysis of ingredients, potential concerns, and implications.
- **Dietary Restrictions**: Checks against major dietary restrictions (vegan, halal, kosher).
- **Nutritional Rating**: Rates nutritional value on a scale of 1-5.
- **Health Implications**: Highlights key health implications or concerns related to ingredients.
- **Healthier Alternatives**: Suggests healthier alternatives where applicable.
- **Evidence-Based Recommendations**: Provides brief, evidence-based recommendations.

2. Installation

To set up and run the Product-Ingredient-Agent application, follow these steps:

2.1. Clone the Repository

First, clone the project repository from GitHub to your local machine:

```Bash
git clone https://github.com/yourusername/Product-Ingredient-Agent.git
cd Product-Ingredient-Agent
```

2.2. Create a Virtual Environment

It is highly recommended to create a virtual environment to manage project dependencies. This ensures that the project's dependencies do not conflict with other Python projects on your system.

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

2.3. Install Dependencies

Once the virtual environment is activated, install the required Python packages using `pip`:

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

This command will install all the libraries listed in `requirements.txt`, including `streamlit`, `phidata`, `pillow`, `tavily-python`, and `google-generativeai`.

3. Configuration

Before running the application, you need to configure your API keys for Tavily Search and Google Gemini AI.

3.1. API Keys

Create a `.env` file in the root directory of your project (e.g., `Product-Ingredient-Agent/.env`) and add your API keys as follows:

```
Plain Text

TAVILY_API_KEY = your_tavily_api_key
GOOGLE_API_KEY = your_gemini_api_key
```

- **TAVILY_API_KEY**: Obtain this from the Tavily API website.
- **GOOGLE_API_KEY**: Obtain this from the Google AI Studio.

These keys are securely loaded by the application using Streamlit secrets management.

3.2. Example Images

The application uses example images for demonstration purposes. You can add your own product images to the `images/` directory. The expected structure is:

```
Plain Text

images/
├── hide_and_seek.jpg
├── bournvita.jpg
├── lays.jpg
└── shampoo.jpg
```

4. Usage

To run the Product-Ingredient-Agent application:

1. **Run the Streamlit app**:
2. **Open in Browser**: Open your web browser and navigate to `http://localhost:8501`.

3. **Analyze a Product**: Choose one of the three options to analyze a product:
    - **Select from example products**: Use pre-loaded images for quick analysis.
    - **Upload your own image**: Upload a product image from your local machine.
    - **Take a photo using your camera**: Capture a photo directly using your device's camera.

5. Project Structure

The project is organized into the following directories and files:

Plain Text

```
product-ingredient-analyzer/
├── app.py                  # Main Streamlit application logic and UI
├── constants.py            # Defines system prompts and instructions for the
AI agent
├── requirements.txt        # Lists all Python dependencies
├── images/                # Contains example product images
├── .streamlit/            # Streamlit specific configurations (e.g.,
secrets.toml)
├── .devcontainer/         # DevContainer configurations for development
environments
├── LICENSE                # Project license information
└── README.md              # Project overview and basic instructions
```

6. Dependencies

The core dependencies for this project are:

- `streamlit`: For building interactive web applications.
- `phidata`: A framework for building AI agents and workflows.
- `pillow`: Python Imaging Library, used for image processing.
- `tavily-python`: Python client for Tavily Search API.

- `google-generativeai` : Google's client library for Gemini AI.

For a complete list of dependencies, refer to the `requirements.txt` file.

7. Contributing

Contributions to the Product-Ingredient-Agent project are welcome. Please consider starring the `Phidata` repository on GitHub to support the underlying framework:

- ⭐ Phidata GitHub Repository

8. License

This project is licensed under the MIT License. See the `LICENSE` file in the project root for more details.

9. Acknowledgments

- **Phidata**: For building the Multimodal Agent framework.
- **Google Gemini AI**: For powering the ingredient analysis.
- **Streamlit**: For providing the web interface.
- **Tavily**: For search capabilities.