Documentation for Flask ChatGPT Integration.

The Flask ChatGPT web application enables users to interact with OpenAI's ChatGPT API by providing a user-friendly interface. Users can input questions or prompts, and the application generates AI-generated responses. It facilitates seamless communication with the AI model, allowing users to receive helpful information and engage in conversational exchanges.

Setup:

Clone the repository: git clone https://github.com/hviik/flaskchatgpt.git

dependencies are flask and openai.

```
pip install openai
```

```
pip install flask
```

Set up OpenAI API credentials by obtaining an API key from OpenAI and adding it as an environment variable named OPENAI_API_KEY.

Run the application:

```
python app.py
```

paste the above in the root folder and it should execute.

Access the web application by opening a web browser and navigating to http://localhost:5000.

Test: Ensure the Flask application is running, enter a promt in http://localhost:5000 or in your custom URL and wait for your response.

API response example:

```
{
  "message": "How does photosynthesis work?",
  "chat_response": "Photosynthesis is the process by which plants convert light energy into chemical energy. It occurs in the chloroplasts of plant cells, where chlorophyll captures sunlight and powers the synthesis of glucose and oxygen from carbon dioxide and water."
}
```

Test the ChatGPT API Integration:

Test a valid message input and verify the generated response from the ChatGPT API.

Input: "Hello"

Expected output: The response generated by the ChatGPT API.

Test an empty message input and verify the error message displayed.

Input: "" (empty string)

Expected output: An error message indicating that the message is required.

Test an API error or failed request and verify the error message displayed.

Input: Any message that causes an API error or failed request.

Expected output: An error message indicating the specific error encountered during the API call.

Test the Proper Rendering of the Homepage and Results Page:

Test the rendering of the homepage by checking the HTML elements and form.

Visit the homepage and inspect the HTML source code.

Test submitting a message from the homepage and verify the rendering of the results page.

Enter a message in the input field and submit the form.

Visit the results page and inspect the HTML source code.

Test submitting an empty message and verify the rendering of the results page with an error message.

Leave the input field empty and submit the form.

Visit the results page and inspect the HTML source code.

Note, please make sure to use your own key here to run the code as I can’t pay for the paid version of the API.