Technical Documentation for ChatGPT Web Application

- ## Purpose and functionality

  Using the OpenAI GPT-3.5 language model, the ChatGPT web Application, a Flask-based web application, enables users to ask questions and receive responses. Users can enter their queries into the application's straightforward user interface and send them to the server. The user sees the response on the results page once the server performs an API call to the GPT-3.5 model to produce it.

- ## Setting up and Running the Application

  Step 1: Clone the Repository

```
$ git clone https://github.com/karan1974/flask_chatgpt.git
$ cd flask_chatgpt
```

Step 2: Install Dependencies

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

Step 3: Set OpenAI API Key

Obtain an API key from OpenAI API.

Apply for API access by going to the https://platform.openai.com/account/api-keys and clicking the "**Create new secret key**" button.

Set the API key as an environment variable:

```
$ export OPENAI_API_KEY="your-api-key"
```

Step 4: Run the application:

```
python app.py
```

The application should now be running locally at http://127.0.0.1:5000/.

- **API Calls and Responses**

1. API Call: ‘POST/results’

   Request Body:

   ```
   {
     "message": "What is the capital of India?"
   }
   ```

   Response Body:

   ```
   {
     "message": "What is the capital of India?",
     "answer": "The capital of India is New Delhi."
   }
   ```

   This API call is made to the ‘/results’ endpoint with the user’s question as the payload. The response includes the original question and the generated answer.

2. API Call: ‘POST/results’

   Request Body:

```
{
  "message": "What is the capital of India?"
}
```

Response Body:

```
{
  "message": "What is the capital of India?",
  "answer": "The capital of India is New Delhi."
}
```

This API call asks about the full form of computer, and the response provides a full form of computer.