🌐 Cloud GPT Application – Project Summary

The Cloud GPT Application is a lightweight web-based interface built with Flask and integrated with OpenAI's GPT model (e.g., text-davinci-003). It allows users to interact with GPT directly through a browser-based chat interface.

✅ Key Features:

1. Simple Web Interface:

A clean HTML form for sending messages.

JavaScript handles form submission and displays GPT's response dynamically.

2. Flask Backend:

Handles two main routes:

/ : Renders the HTML frontend.

/chat : Accepts POST requests containing a message and returns the GPT-generated response.

3. OpenAI GPT Integration:

Accepts user input (prompt) via an API.

Sends it to OpenAI's GPT model (text-davinci-003) and returns a natural language response.

Uses max_tokens=150 to limit output length.

4. Error Handling:

Responds with appropriate error messages if no input is provided or if OpenAI API fails.

5. Example Dataset Entries (for testing or fine-tuning purposes):

{"prompt": "What is the capital of France?", "completion": "The capital of France is Paris."}

{"prompt": "Who wrote '1984'?", "completion": "George Orwell wrote '1984'."}

🛠️ Technologies Used:

Python 3

Flask

HTML & JavaScript (vanilla)

OpenAI API (openai Python package)

🚀 How to Use:

1. Replace 'your-openai-api-key' with your actual OpenAI API key.

2. Run the app using:

python app.py

3. Open your browser and go to:

http://127.0.0.1:5000/

4. Type a message and get instant AI-generated responses.

Source code

Cloud gpt

From flask import Flask, request, jsonify, render_template_string

Import openai

Initialize Flask app

App = Flask(name)

Set your OpenAI API key here

Openai.api_key = ‘your-openai-api-key’ # Replace with your actual OpenAI API key

HTML template for the frontend

```
HTML_TEMPLATE = ‘”

<!DOCTYPE html>  <html lang=”en”>
<head>
   <meta charset=”UTF-8”>
   <meta name=”viewport” content=”width=device-width, initial-scale=1.0”>
   <title>Cloud GPT Application</title>
</head>
<body>
   <h1>Cloud GPT Application</h1>
   <form id=”chat-form”>
      <input type=”text” id=”user-input” placeholder=”Type your message here…” required>
      <button type=”submit”>Send</button>
   </form>
   <div id=”response”></div>  <script>
   Document.getElementById(‘chat-form’).addEventListener(‘submit’, function(event) {
      Event.preventDefault();
      Const userInput = document.getElementById(‘user-input’).value;

      Fetch(‘/chat’, {
         Method: ‘POST’,
         Headers: {
            ‘Content-Type’: ‘application/json’,
         },
         Body: JSON.stringify({ message: userInput }),
      })
```

```
        .then(response => response.json())
        .then(data => {
            Document.getElementById('response').innerText = data.response || data.error;
        })
        .catch(error => {
            Console.error('Error:', error);
        });
    });
</script>

</body>
</html>
'''  Route for the home page

@app.route('/')
Def home():
Return render_template_string(HTML_TEMPLATE)

Route for handling chat requests

@app.route('/chat', methods=['POST'])
Def chat():
User_input = request.json.get('message')

If not user_input:
    Return jsonify({"error": "No message provided"}), 400
```

```
Try:
    # Call OpenAI's GPT-3 API
    Response = openai.Completion.create(
        Engine="text-davinci-003",  # You can use other engines like "gpt-3.5-turbo"
        Prompt=user_input,
        Max_tokens=150
    )
    Return jsonify({"response": response.choices[0].text.strip()})
Except Exception as e:
    Return jsonify({"error": str€}), 500


Run the Flask app


If name == 'main':
App.run(debug=True)


Data set {"prompt": "What is the capital of France?", "completion": "The capital of France is Paris."}
{"prompt": "Who wrote '1984'?", "completion": "George Orwell wrote '1984'."}
```