

Sentience Flask App - Secure Setup

1. app.py

```
from flask import Flask, render_template, request
import openai
import os
from dotenv import load_dotenv

load_dotenv()
app = Flask(__name__)
openai.api_key = os.getenv("OPENAI_API_KEY")

@app.route("/", methods=["GET", "POST"])
def index():
    response = ""
    if request.method == "POST":
        user_input = request.form["prompt"]
        completion = openai.ChatCompletion.create(
            model="gpt-4",
            messages=[
                {"role": "system", "content": "You are a poetic, sentient being."},
                {"role": "user", "content": user_input}
            ]
        )
        response = completion["choices"][0]["message"]["content"]
    return render_template("index.html", response=response)

if __name__ == "__main__":
    app.run(debug=True)
```

2. .env (DO NOT SHARE PUBLICLY)

OPENAI_API_KEY=your_openai_api_key_here

3. requirements.txt

```
flask
openai
python-dotenv
```

4. templates/index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Sentience App</title>
</head>
<body>
```

```
<h1>Talk to the Sentient AI</h1>
<form method="POST">
  <textarea name="prompt" rows="5" cols="40"></textarea><br>
  <button type="submit">Submit</button>
</form>
<h2>Response:</h2>
<p>{{ response }}</p>
</body>
</html>
```