From flask import Flask, request, jsonify, render\_template\_string

Import openai

Import os

# Initialize Flask app

App = Flask(\_\_name\_\_)

# Set your OpenAI API key (ensure it’s set as an environment variable)

Openai.api\_key = os.getenv(“OPENAI\_API\_KEY”)

# HTML Template for Web Interface

Html\_template = “””

<!DOCTYPE html>

<html>

<head>

<title>AI Customer Support Chatbot</title>

<style>

Body { font-family: Arial, sans-serif; margin: 20px; background-color: #f5f5f5; }

#chatbox { border: 1px solid #ccc; background: #fff; padding: 15px; height: 400px; overflow-y: scroll; }

#userInput { width: 80%; padding: 10px; }

Button { padding: 10px; }

P { margin: 5px 0; }

</style>

</head>

<body>

<h2>Customer Support Chatbot</h2>

<div id=”chatbox”></div>

<input type=”text” id=”userInput” placeholder=”Type your message...” />

<button onclick=”sendMessage()”>Send</button>

<script>

Async function sendMessage() {

Const input = document.getElementById(‘userInput’);

Const message = input.value;

If (!message) return;

Const chatbox = document.getElementById(‘chatbox’);

Chatbox.innerHTML += “<p><strong>You:</strong> “ + message + “</p>”;

Input.value = “”;

Const res = await fetch(“/chat”, {

Method: “POST”,

Headers: { “Content-Type”: “application/json” },

Body: JSON.stringify({ message: message })

});

Const data = await res.json();

Chatbox.innerHTML += “<p><strong>Bot:</strong> “ + data.reply + “</p>”;

Chatbox.scrollTop = chatbox.scrollHeight;

}

</script>

</body>

</html>

“””

# Route for the chat interface

@app.route(‘/’)

Def index():

Return render\_template\_string(html\_template)

# Chat endpoint for AJAX

@app.route(‘/chat’, methods=[‘POST’])

Def chat():

User\_message = request.json.get(‘message’, ‘’)

If not user\_message:

Return jsonify({‘reply’: ‘Please enter a message.’})

Try:

Response = openai.ChatCompletion.create(

Model=”gpt-4”,

Messages=[

{“role”: “system”, “content”: “You are a helpful customer support chatbot that provides professional, friendly, and accurate responses.”},

{“role”: “user”, “content”: user\_message}

]

)

Reply = response[‘choices’][0][‘message’][‘content’].strip()

Return jsonify({‘reply’: reply})

Except Exception as e:

Return jsonify({‘reply’: f”An error occurred: {str(e)}”})

# Run the Flask app

If \_\_name\_\_ == ‘\_\_main\_\_’:

App.run(debug=True)