pip install flask flask-cors openai

from flask import Flask, request, jsonify

from flask\_cors import CORS

import openai

app = Flask(\_\_name\_\_)

CORS(app)

# OpenAI API key

openai.api\_key = "your\_openai\_api\_key"

@app.route('/generate-avatar', methods=['POST'])

def generate\_avatar():

"""

Generate a spiritual avatar description based on user's input.

"""

data = request.json

spiritual\_goals = data.get('spiritual\_goals', '')

qualities = data.get('qualities', '')

connection\_type = data.get('connection\_type', '')

# Validate input

if not spiritual\_goals or not qualities or not connection\_type:

return jsonify({"error": "Invalid input. Provide all required fields."}), 400

try:

# Call OpenAI API for avatar description

response = openai.ChatCompletion.create(

model="gpt-4",

messages=[

{"role": "system", "content": "You are a guide creating spiritual avatars based on user preferences."},

{"role": "user", "content": f"Create an avatar for someone seeking {spiritual\_goals} with qualities of {qualities} who wants a {connection\_type} connection."}

],

)

avatar\_description = response['choices'][0]['message']['content']

return jsonify({"avatar\_description": avatar\_description})

except Exception as e:

return jsonify({"error": str(e)}), 500

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

{

"spiritual\_goals": "peace and mindfulness",

"qualities": "compassion and resilience",

"connection\_type": "mentorship"

}

{

"avatar\_description": "Your avatar is a serene figure bathed in golden light, embodying peace and mindfulness. They carry a staff of wisdom and wear a flowing cloak adorned with symbols of compassion and resilience. They offer gentle mentorship through their words and presence."

}

async function fetchAvatar() {

const response = await fetch('http://localhost:5000/generate-avatar', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({

spiritual\_goals: "peace and mindfulness",

qualities: "compassion and resilience",

connection\_type: "mentorship",

}),

});

const data = await response.json();

console.log(data.avatar\_description);

}

@app.route('/generate-avatar-image', methods=['POST'])

def generate\_avatar\_image():

data = request.json

avatar\_description = data.get('avatar\_description', '')

if not avatar\_description:

return jsonify({"error": "Avatar description is required."}), 400

try:

# Call OpenAI API for image generation

response = openai.Image.create(

prompt=avatar\_description,

n=1,

size="1024x1024"

)

image\_url = response['data'][0]['url']

return jsonify({"image\_url": image\_url})

except Exception as e:

return jsonify({"error": str(e)}), 500

<img src="avatar\_image\_url" alt="Spiritual Avatar" />