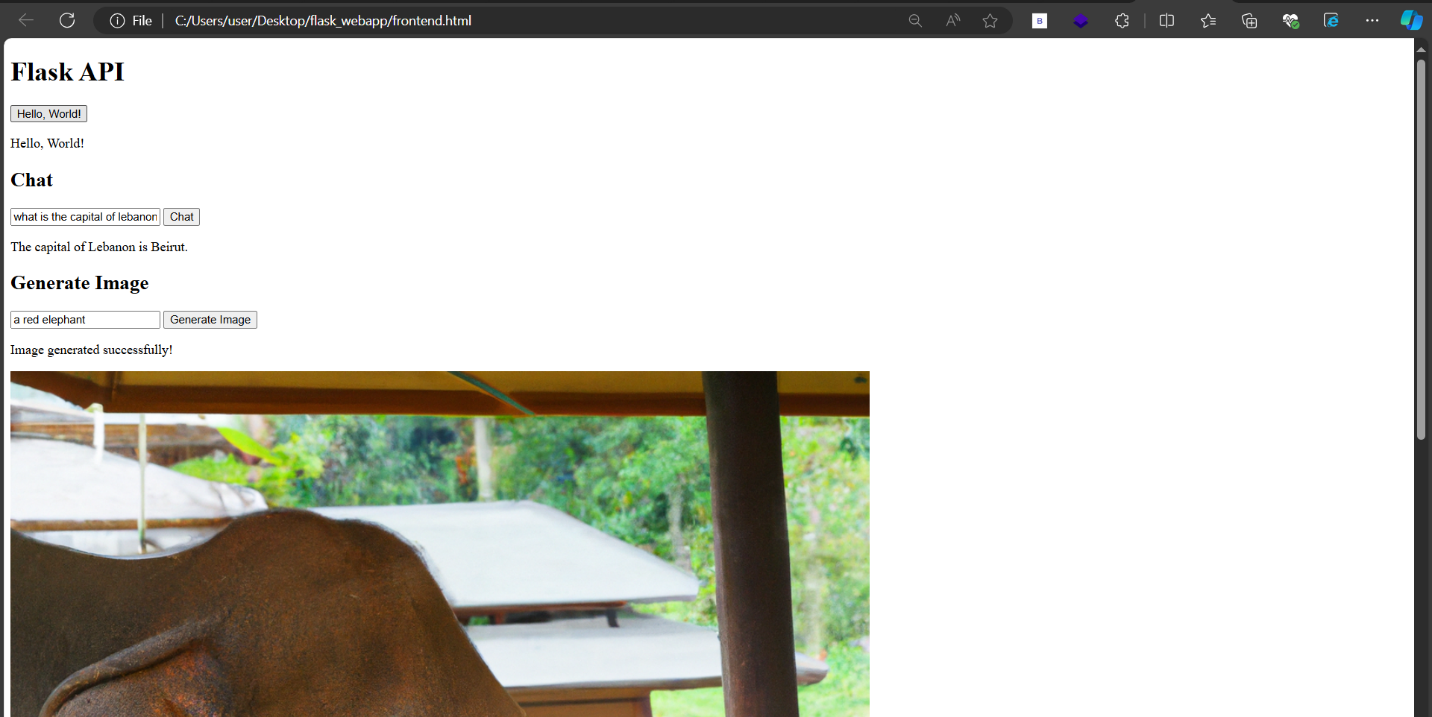
**Python CF Assessment**

# Building and Running the Docker Container:

1. Prerequisites:
   * Docker Desktop installed on your local machine.
   * OpenAI API key.
   * An API key for authorization can be generated by generate\_api\_key.py script
2. Build the Docker Image: docker build -t flask-openai-app .
3. Run the Docker Container: docker run -p 5000:5000 -e OPENAI\_API\_KEY=your\_openai\_api\_key -e API\_KEY=your\_generated\_api\_key flask-openai-app
4. Open frontend.html file and click on hello world button to test GET request and by clicking on it, it response Hello World! Appear under the button
5. To test POST/chat request, write what you want in text field then click chat button. And the response will appear under it
6. Also, to test POST/generateimage request, write what you want into text field and click generate image and it will response by image.



Note: OpenAI key needed to generate chat and image

# Instructions for Interacting with the Flask Application:

Endpoints

* GET /

This endpoint will returns “hello world” message to confirm the application is running and authorized

Request:

* Method: GET
* URL: http://localhost:5000/
* Headers:
* x-api-key: Your generated API key

Response:

* Status: 200 OK
* Body:

json

{

"response": "Hello, World!"

}

* POST /chat

This endpoint accepts a JSON input with a user message and returns a response from openai’s chat completion API.

Request:

* Method: POST
* URL: http://localhost:5000/chat
* Headers:
* x-api-key: Your generated API key
* Content-Type: application/json

Body:

json

{

"content": "What is the capital of Lebanon?"

}

Response:

* Status: 200 OK
* Body:

json

{

"response": "The capital of Lebanon is Beirut."

}

* POST /generateImage

This endpoint accepts a JSON input with a prompt and returns an image generated by openai.

Request:

* Method: POST
* URL: http://localhost:5000/generateImage
* Headers:
* x-api-key: Your generated API key
* Content-Type: application/json
* response-type: Either base64 or image

Body:

json

{

"content": "A red elephant"

}

Response:

If response-type is base64:

json

{

"base64": "<image\_base64\_data>",

"version": "0.1.0"

}

If response-type is image: The image will be directly returned.

# Prerequisites and Dependencies:

## Prerequisites:

* docker
* openai key
* python

Dependencies:

* flask web framework
* openai
* flask-cors
* swagger UI

# Examples of Requests and Expected Responses:

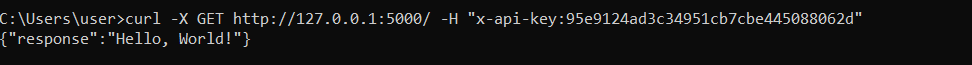
## GET Request

**Request**:

curl -X GET http://127.0.0.1:5000/ -H "x-api-key:95e9124ad3c34951cb7cbe445088062d”

**reponse**:

{"response":"Hello, World!"}



Or you can open frontend.html file and click on hello world button

## POST /chat Request

**Request**:

curl -X POST http://127.0.0.1:5000/chat \

-H "x-api-key: 95e9124ad3c34951cb7cbe445088062d \

-H "Content-Type: application/json" \

-d "{\"content\": \"What is the capital of Lebanon?\"}"

Or by frontend.html file

**Response**:

{ "response": "The capital of Lebanon is Beirut."}

## POST /generateImage Request

**Request**:

curl -X POST http://127.0.0.1:5000/generateImage \

-H "x-api-key: 95e9124ad3c34951cb7cbe445088062d " \

-H "Content-Type: application/json" \

-H "response-type: base64" \

-d "{\"content\": \"A red elephant\"}"

Or by frontend.html file

**Response:**

**{ "base64": "<image\_base64\_data>",**

**"version": "0.1.0"}**