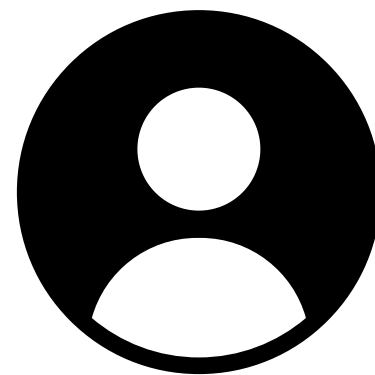
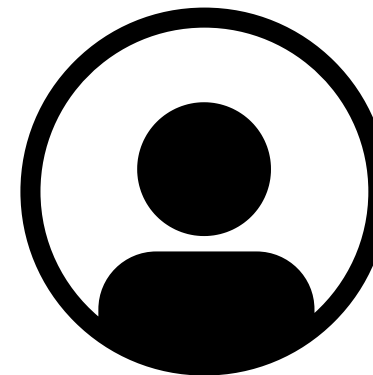


GeniusTalk

Engage in interactive conversations with greatest minds.



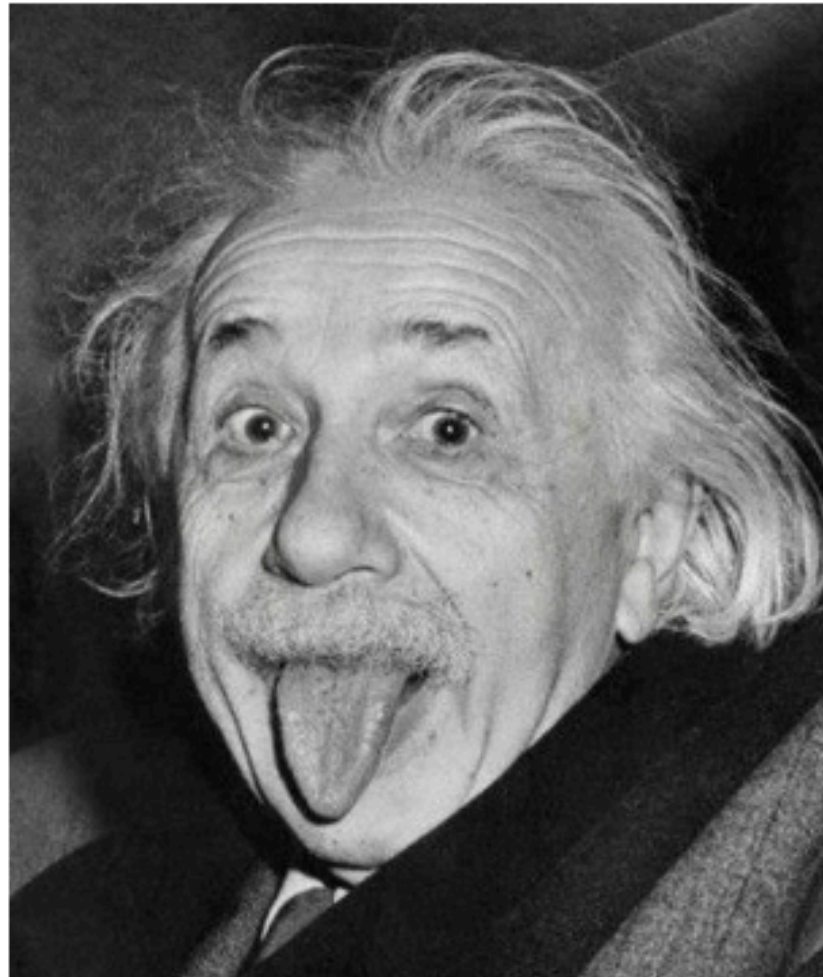
Jingxuan



Yue



Engage in meaningful conversations with history's greatest minds and artistic figures—learn, explore, and be inspired.



[Chat Now](#)

Database

Smithsonian Institution Open Access API

API documentation

Using the Services.

The Open Access API requires an API key to access the endpoints. Please register with <https://api.data.gov/signup> to get a key.

General Information

More information about the data repository and how to use the API can be found at <https://edan.si.edu/openaccess/docs/>

Click here to learn more about the [Open Access Initiative at the Smithsonian](#)



```
import requests
import json

# Get user input for the search query
search_query = input("Enter the search query: ")

# request data from the API
url = 'https://api.si.edu/openaccess/api/v1.0/search'
params = {
    'q': search_query,
    'sort': 'newest',
    'api_key': 'wFnb6cMtcbnXx2cDNtW2Yv8bbezHcjsHY6gPi0fc'
}
response = requests.get(url, params=params)

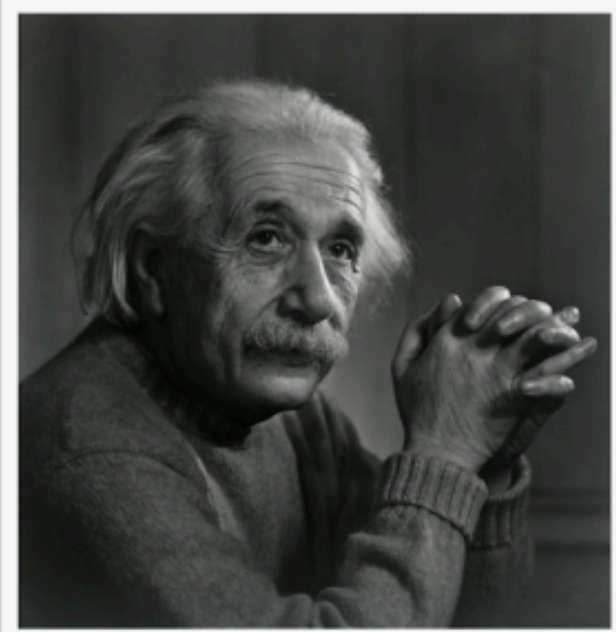
# Process the response
if response.status_code == 200: # HTTP status code 200 means success
    try:
        # Parse the response as JSON
        data = response.json()

        # Debug: Print the raw JSON response (optional)
        print("Raw API Response:", json.dumps(data, indent=4))

        # Access artworks from the response
        artworks = data.get("response", {}).get("rows", [])
        print(f"Found {len(artworks)} artworks.")

        # Prepare filtered results
        filtered_artworks = []
        for artwork in artworks:
```

```
        "type": "edanmdm",
        "url": "edanmdm:nmnhbotany_14344006",
        "content": {
            "freetext": {
                "date": [
                    {
                        "label": "Collection Date",
                        "content": "6 Apr 1928"
                    }
                ],
                "name": [
                    {
                        "label": "Biogeographical Region",
                        "content": "74 - North-Central U.S.A."
                    }
                ]
            }
        }
        ...
        "date": [],
        "record_link": "https://siris-libraries.si.edu/ipac20/ipac.js"
    }
```



Name: Albert Einstein
Year: 1948
Country: United States

Hi, I am Albert Einstein! What do you want to know from me?

What is your most proud achievement in your life?

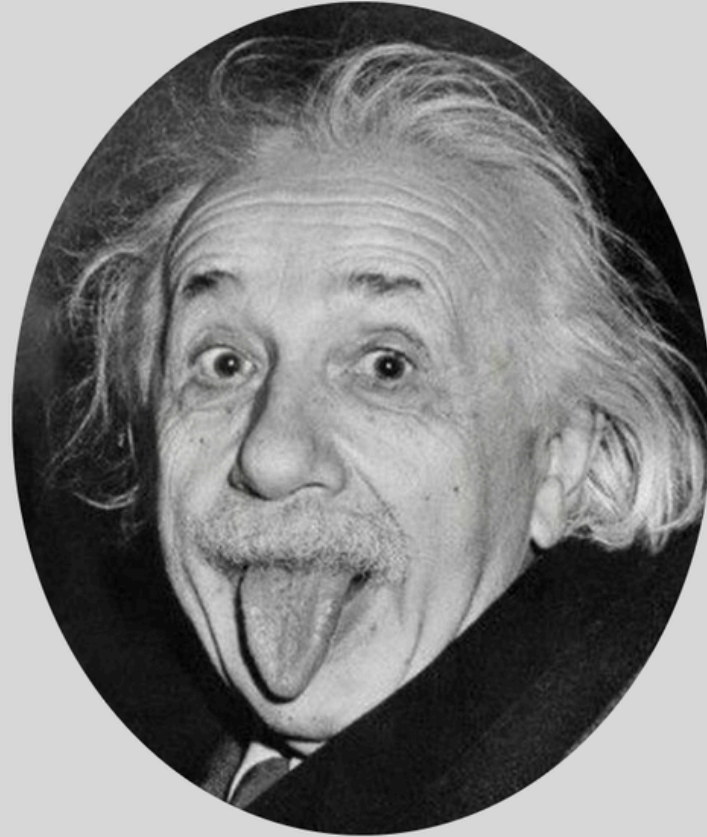
Type your message here...

Send

What is relativity?

Ah, relativity! It is a beautifully intricate concept that I am known for. In simple terms, it is the idea that measurements of space and time can vary depending on your frame of reference. Essentially, everything is relative to the observer's motion and perspective. Quite fascinating, isn't it? Now, do tell me, what intrigues you about the concept of relativity?





"Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution."

- Albert Einstein

Summary

Albert Einstein (1879–1955) was a German-born theoretical physicist widely regarded as one of the most influential scientists of the 20th century. He is best known for developing the theory of relativity, particularly the equation $E=mc^2$, which describes the equivalence of energy and mass.

Einstein's groundbreaking work reshaped our understanding of time, space, gravity, and the universe. In 1921, he was awarded the Nobel Prize in Physics for his explanation of the photoelectric effect, which was pivotal in establishing quantum theory.



Continue

Finish

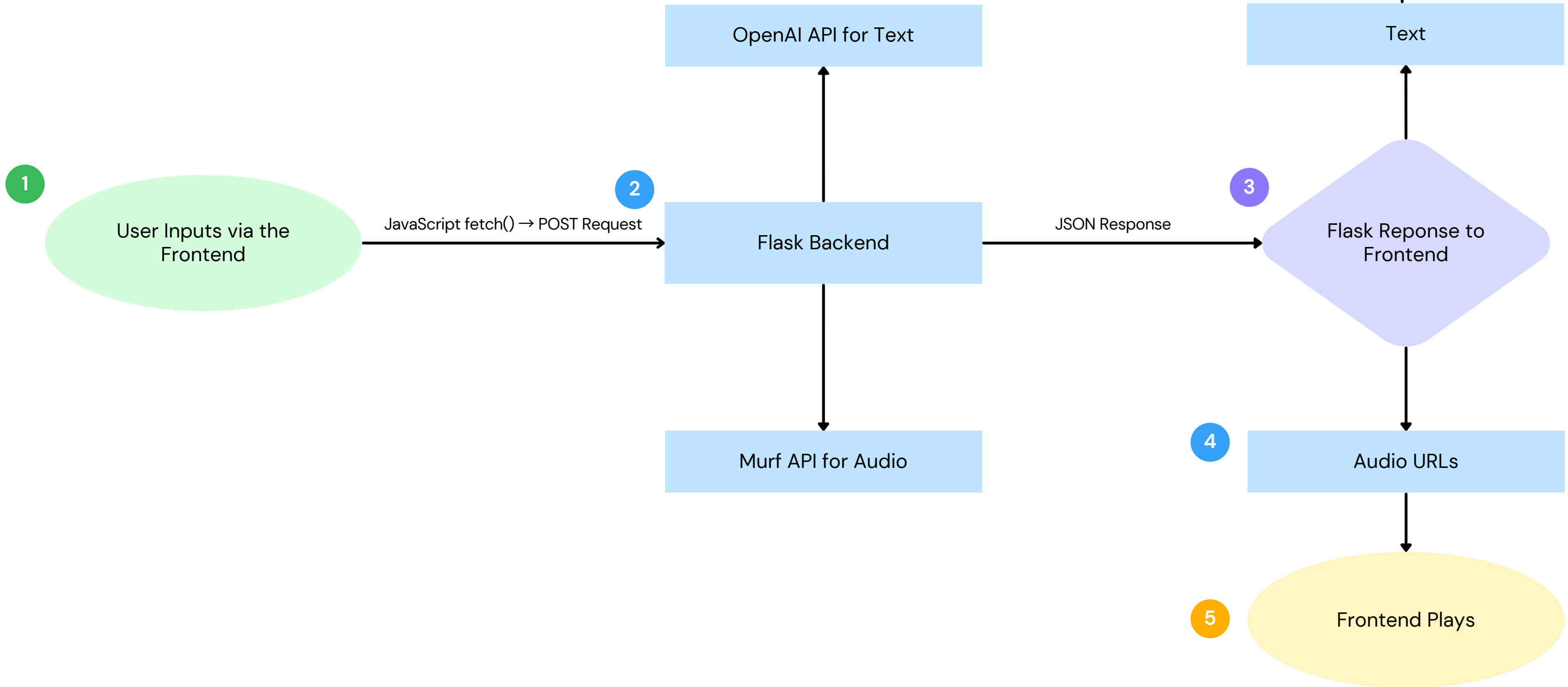
in your life?

Type your message here...

Send



Workflow: Backend-Frontend



Flask Backend

Flask Initialization

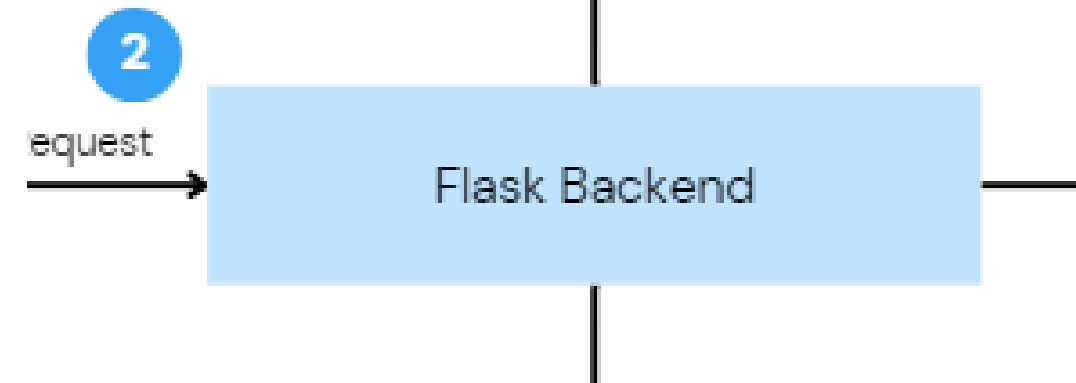
```
from flask import Flask, request, jsonify
from flask_cors import CORS

app = Flask(__name__)
CORS(app, resources={r"/*": {"origins": "*"}})

@app.route('/')
def home():
    return "Welcome to GeniusTalk! The chat endpoint is available at /chat."
```

the /chat endpoint that processes user message

```
@app.route('/chat', methods=['POST'])
def chat():
    try:
        user_message = request.json.get('message', '')
        if not user_message:
            return jsonify({'response': "I didn't catch that. Could you repeat?"}), 400
```



1. Flask Setup:

- `Flask(__name__)`: Initializes the Flask app.

2. CORS Integration:

- Enables cross-origin requests for frontend-backend communication.

3. Default Endpoint:

- A simple `/` route to confirm the server is running.

1. Data Flow:

- Frontend sends user input as JSON (`request.json`).

2. Input Validation:

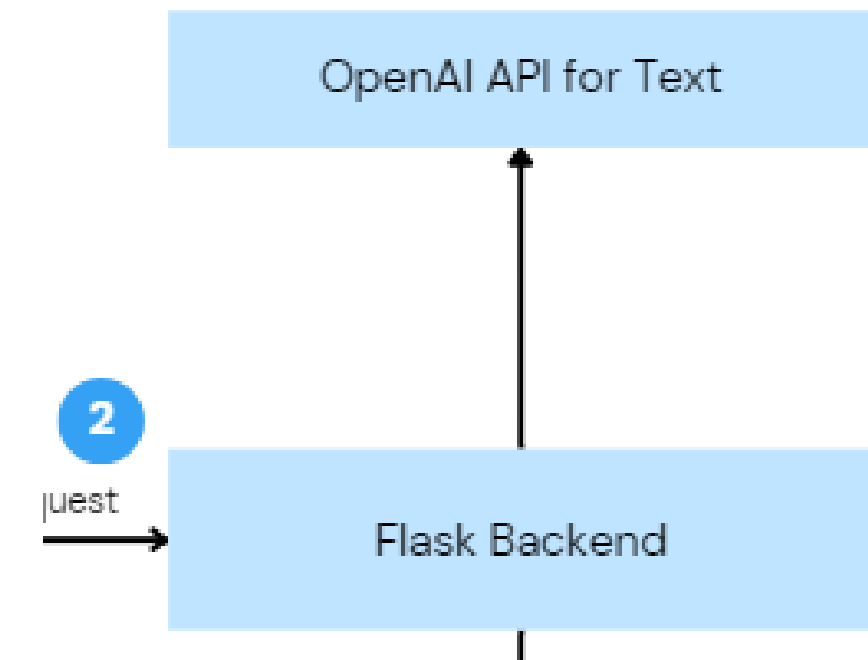
- If no message is provided, returns an error (`400` status code).

OpenAI API Integration for Text

```
# Generate AI response
response = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "system", "content": "You are Albert Einstein"},
        {"role": "user", "content": user_message}
    ],
    max_tokens=150,
    temperature=0.7
)
answer = response.choices[0].message['content'].strip()

# Generate audio using Murf
audio_url = generate_audio_from_text(answer)

# Return response to frontend
return jsonify({"reply": answer, "audio_url": audio_url})
except Exception as e:
    return jsonify({"error": str(e)}), 500
```



1. `model` : Specifies the GPT-3.5 Turbo model for responses.
2. `messages` : Contextual input:
 - `system` : Defines Einstein's role.
 - `user` : Captures the user's query.
3. `max_tokens` : Limits the response length.
4. `temperature` : Controls randomness (0.7 balances coherence and creativity).

Murf API Integration for Audio

```
def generate_audio_from_text(reply_text):
    payload = json.dumps({
        "voiceId": "en-US-ken",
        "style": "Conversational",
        "text": reply_text,
        "sampleRate": 48000,
        "format": "MP3",
    })
    response = requests.post(
        "https://api.murf.ai/v1/speech/generate",
        headers={'api-key': MURF_API_KEY, 'Content-Type': 'application/json'},
        data=payload
    )
    if response.status_code == 200:
        return response.json().get('audioFile', None)
    else:
        print(f"Error: {response.status_code}")
        return None
```

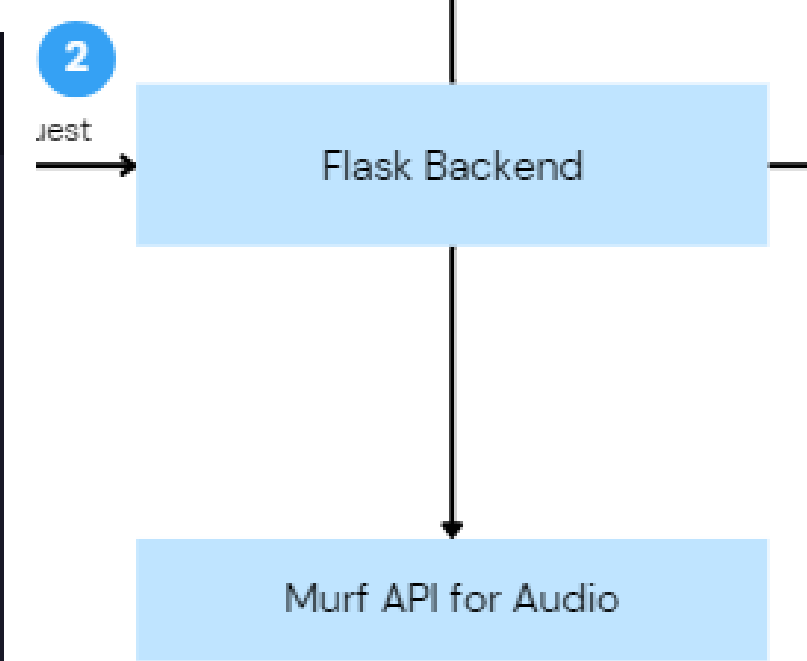


```
import requests
import json
```

```
url = "https://api.murf.ai/v1/speech/generate"
```

```
payload = json.dumps({
    "voiceId": "en-US-natalie",
    "style": "Promo",
    "text": "In this experiential e-learning module, you'll master the basics of using this Tex",
    "rate": 0,
    "pitch": 0,
    "sampleRate": 48000,
    "format": "MP3",
    "channelType": "MONO",
    "pronunciationDictionary": {},
})
```

Murf Text-to-Speech API



1. Voice and Style Selection:

- `voiceId="en-US-ken"` for a conversational tone.

2. API Request:

- `requests.post()` sends a JSON payload with the text to convert.

3. Error Handling:

- Logs errors if the API fails.



Lessons Learned

- *Don't Panic. Stay Calm. Ask for Help.*
- *Documentation always helps!*
- *Refine the project iteratively.*
- *Be realistically ambitious.*



Demo(n) Time 🙌!