PyCon SriLanka

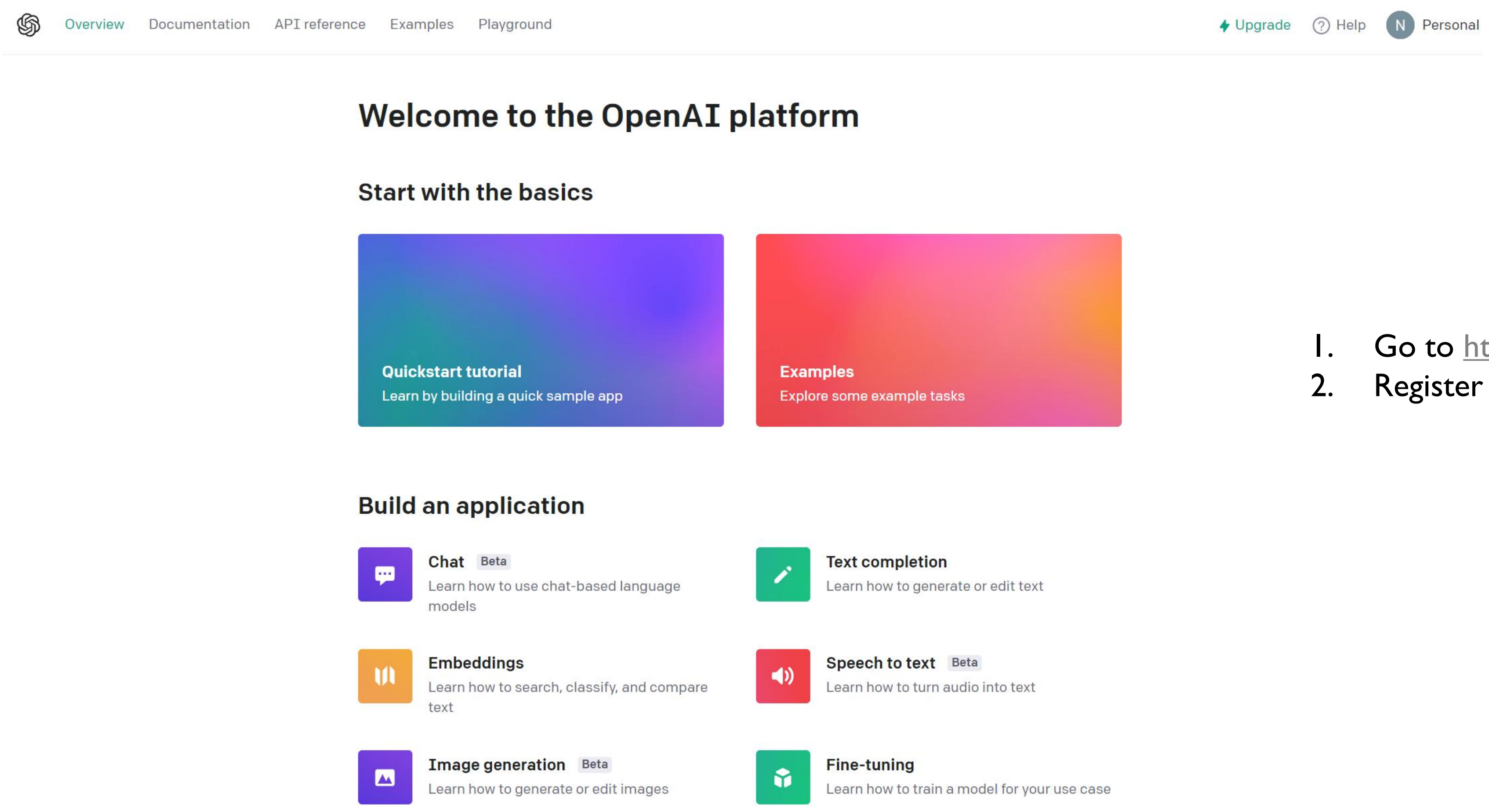
CREATE A (STELLAR!) PROFILE SUMMARIZER WEBAPP USING OPENAI API, AND STREAMLIT

NABANITA ROY
DATA SCIENTIST @ EY

INTRODUCTION TO CHATGPT

- ChatGPT is the state-of-the-art chatbot developed by OpenAI and backed by Microsoft
- It is a massive language model built on GPT3.5 model architecture
- ChatGPT is a sibling model to InstructGPT
- Can be utilized for multiple NLP tasks like classification, summarization, keyword extraction and so on.
- Number of parameters: 175 billion
- Size of Training Data: ~570 GB
- Usage: Content Generation, Language Translation, Summarization, Text Completion, QA, Classification, Named Entity Recognition, and many more

GETTING STARTED WITH OPENAI CHATCOMPLETION API



Welcome to the OpenAI platform

Start with the basics

Quickstart tutorial
Learn by building a quick sample app

Examples
Explore some example tasks

Build an application

Chat Beta
Learn how to use chat-based language models

Text completion
Learn how to generate or edit text

Embeddings
Learn how to search, classify, and compare text

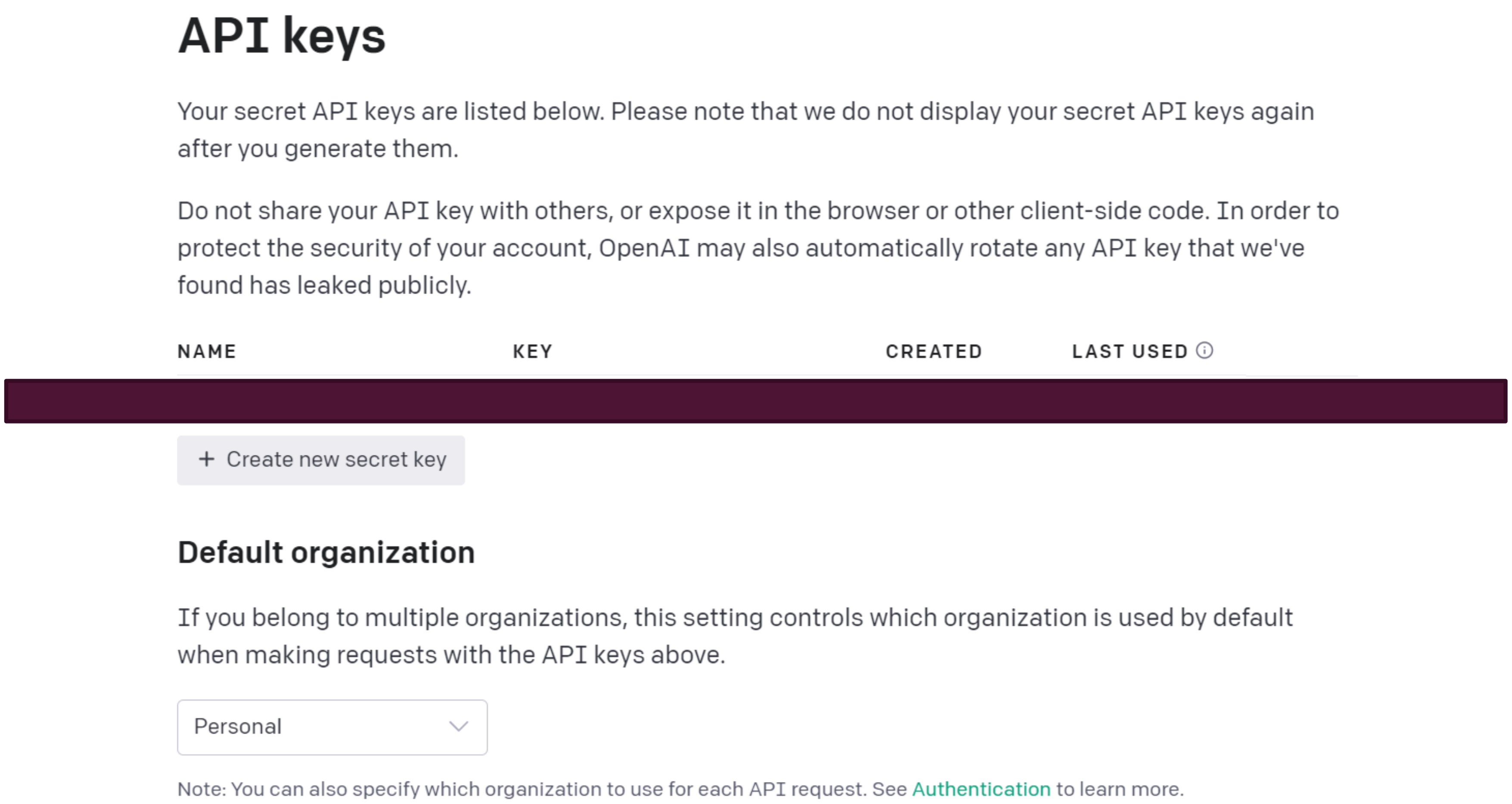
Speech to text Beta
Learn how to turn audio into text

Image generation Beta
Learn how to generate or edit images

Fine-tuning
Learn how to train a model for your use case

1. Go to https://openai.com/
2. Register for the API and generate API key

GETTING STARTED WITH OPENAI CHATCOMPLETION API



API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED ⓘ
[redacted]	[redacted]	[redacted]	[redacted]

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal

Note: You can also specify which organization to use for each API request. See Authentication to learn more.

PROMPT ENGINEERING FOR TEXT GENERATION

- Prompts are communications/instructions given to an LLM to produce a suitable response
- Prompt Engineering is the art of constructing the prompts efficiently to guide the LLM to produce the desired responses

Why Prompt Engineering?

- Enable developing solutions by using LLMs
- Test and Evaluate LLMs

Model Parameters

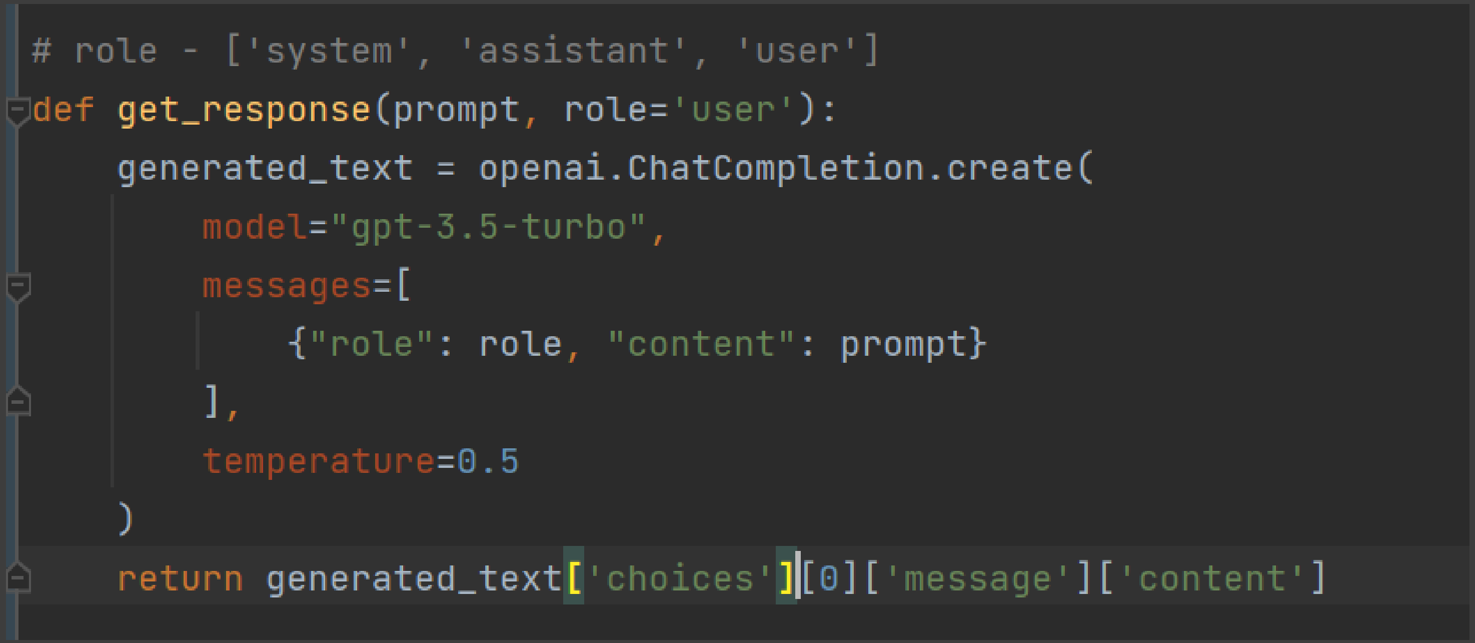
temperature

top_p

n

max_tokens

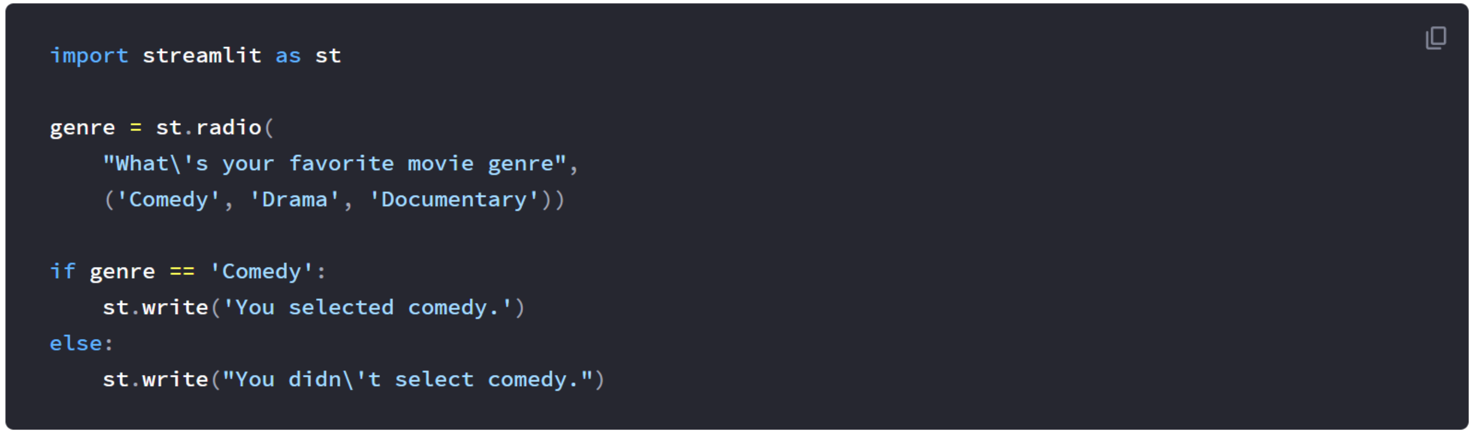
CHATCOMPLETION API IN PYTHON



```python
# role - ['system', 'assistant', 'user']
def get_response(prompt, role='user'):
    generated_text = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": role, "content": prompt}
        ],
        temperature=0.5
    )
    return generated_text['choices'][0]['message']['content']
```

INTRODUCTION TO STREAMLIT

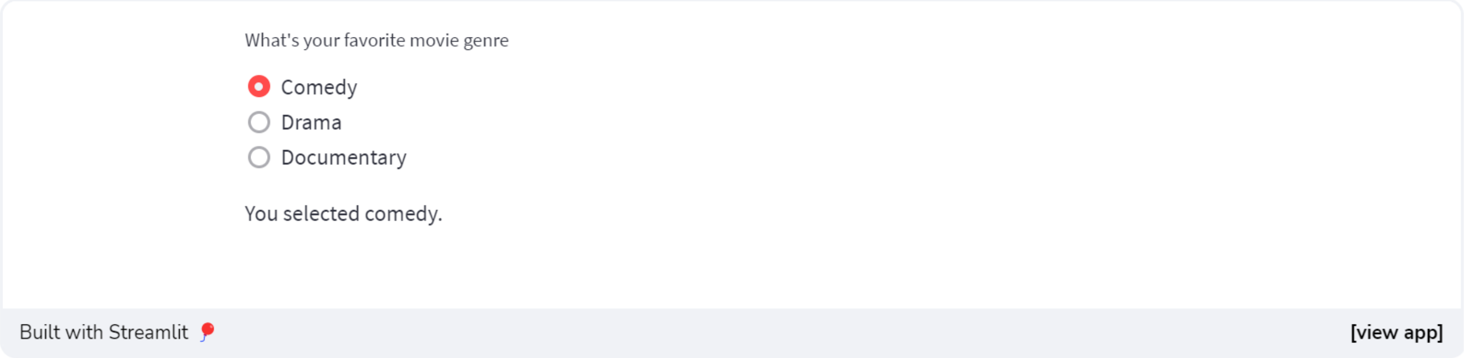
- Quickly build web applications using Python
- Documentation on the features includes useful examples



```python
import streamlit as st

genre = st.radio(
    "What\'s your favorite movie genre",
    ('Comedy', 'Drama', 'Documentary'))

if genre == 'Comedy':
    st.write('You selected comedy.')
else:
    st.write("You didn\'t select comedy.")
```

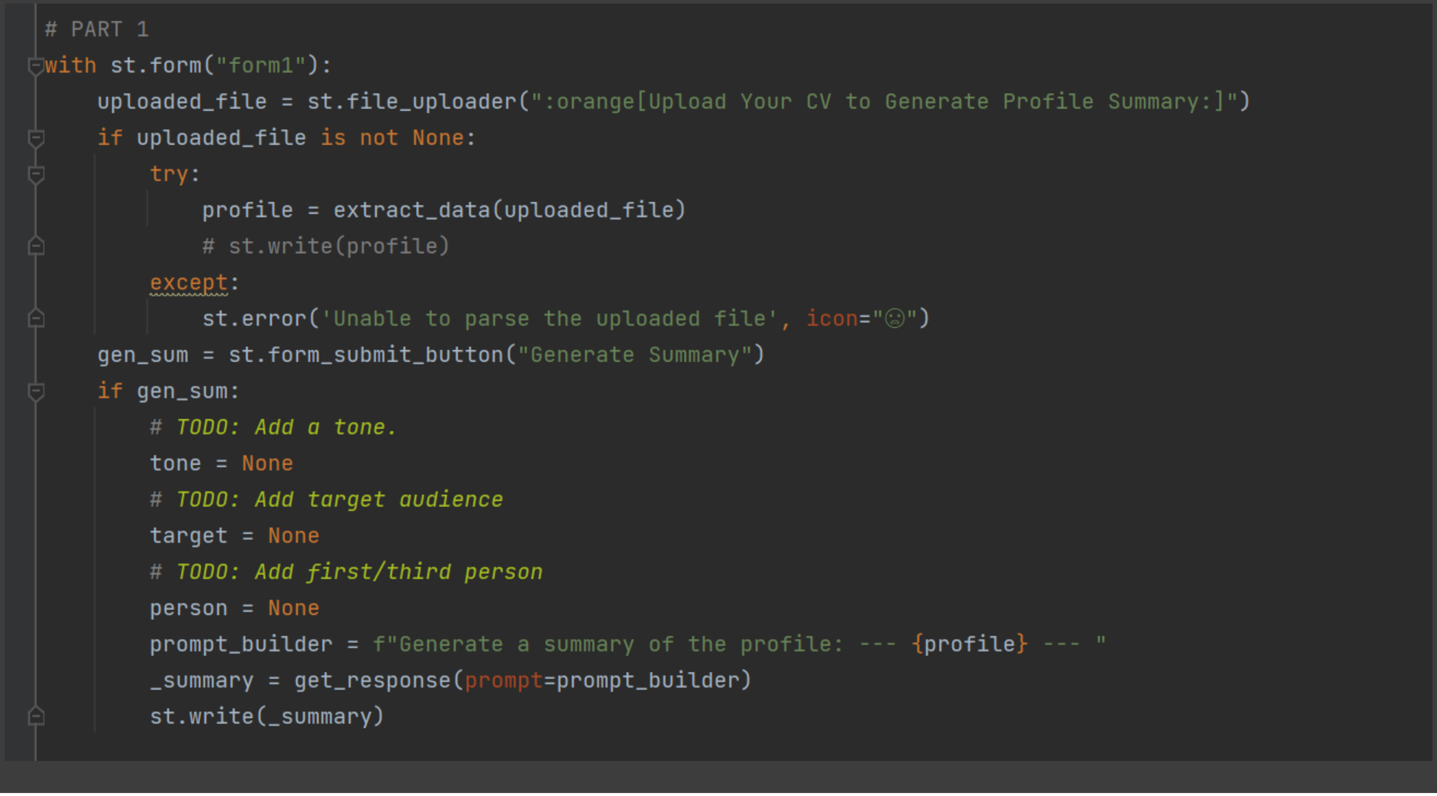


What's your favorite movie genre

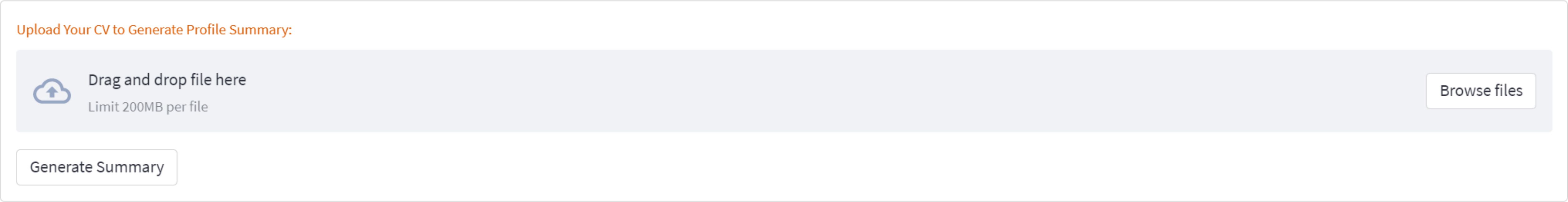
● Comedy
○ Drama
○ Documentary

You selected comedy.

STREAMLIT CODE

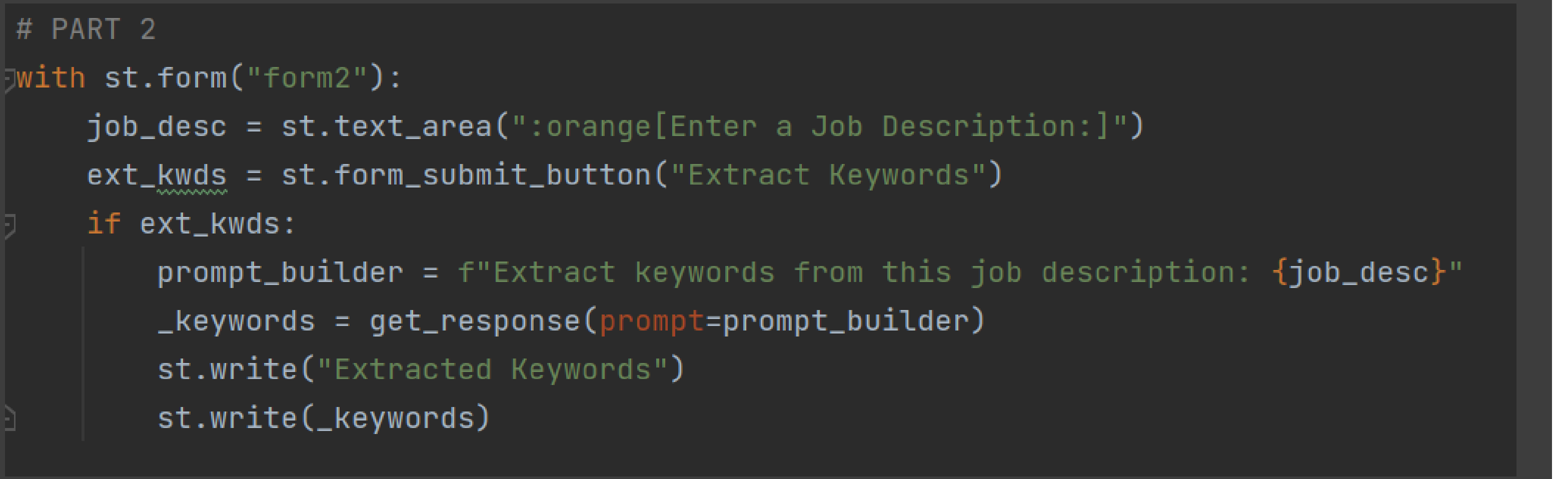


```
# PART 1
with st.form("form1"):
    uploaded_file = st.file_uploader(":orange[Upload Your CV to Generate Profile Summary:]")
    if uploaded_file is not None:
        try:
            profile = extract_data(uploaded_file)
            # st.write(profile)
        except:
            st.error('Unable to parse the uploaded file', icon="😬")
    gen_sum = st.form_submit_button("Generate Summary")
    if gen_sum:
        # TODO: Add a tone.
        tone = None
        # TODO: Add target audience
        target = None
        # TODO: Add first/third person
        person = None
        prompt_builder = f"Generate a summary of the profile: --- {profile} --- "
        _summary = get_response(prompt=prompt_builder)
        st.write(_summary)
```

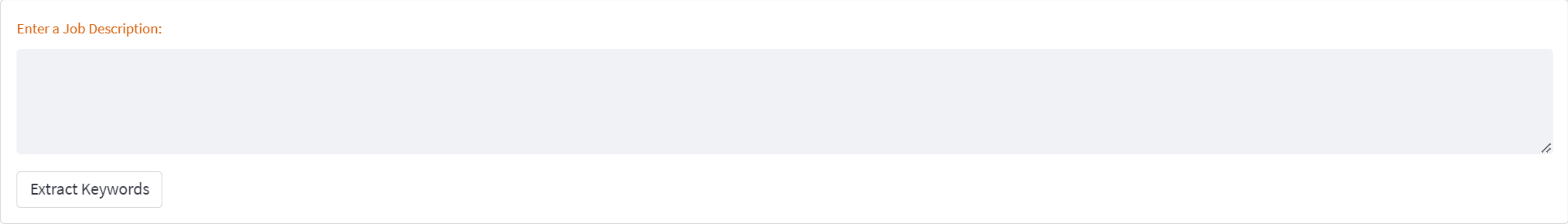


Upload Your CV to Generate Profile Summary:

STREAMLIT CODE



```
# PART 2
with st.form("form2"):
    job_desc = st.text_area(":orange[Enter a Job Description:]")
    ext_kwds = st.form_submit_button("Extract Keywords")
    if ext_kwds:
        prompt_builder = f"Extract keywords from this job description: {job_desc}"
        _keywords = get_response(prompt=prompt_builder)
        st.write("Extracted Keywords")
        st.write(_keywords)
```



Enter a Job Description:

DEMO

References:

1. https://openai.com/
2. https://github.com/dair-ai/Prompt-Engineering-Guide/blob/main/lecture/Prompt-Engineering-Lecture-Elvis.pdf
3. https://streamlit.io/
4. Slides: https://github.com/royn5618/Talks_Resources
5. GitHub Repository: https://github.com/royn5618/ProfileSummaryGeneratorChatGPT

Thank You

Questions?