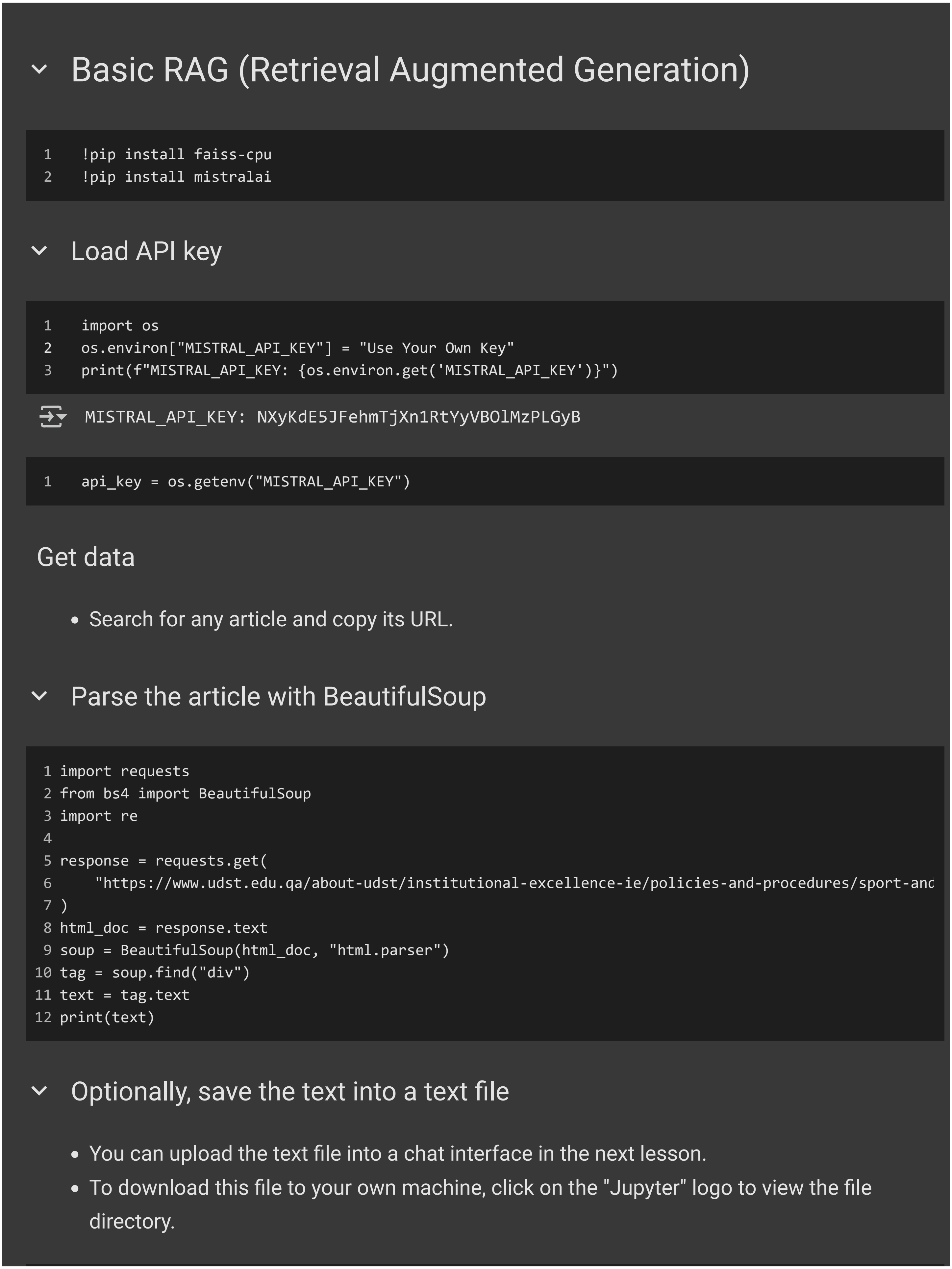


Basic RAG (Retrieval Augmented Generation)

```
!pip install faiss-cpu
!pip install mistralai
```

Load API key

```
import os
os.environ["MISTRAL_API_KEY"] = "Use Your Own Key"
print(f"MISTRAL_API_KEY: {os.environ.get('MISTRAL_API_KEY')}")
```

```
MISTRAL_API_KEY: NXyKdE5JFehmTjXn1RtYyVBOlMzPLGyB
```

```
api_key = os.getenv("MISTRAL_API_KEY")
```

Get data

- Search for any article and copy its URL.

Parse the article with BeautifulSoup

```
import requests
from bs4 import BeautifulSoup
import re

response = requests.get(
    "https://www.udst.edu.qa/about-udst/institutional-excellence-ie/policies-and-procedures/sport-and
)
html_doc = response.text
soup = BeautifulSoup(html_doc, "html.parser")
tag = soup.find("div")
text = tag.text
print(text)
```

Optionally, save the text into a text file

- You can upload the text file into a chat interface in the next lesson.
- To download this file to your own machine, click on the "Jupyter" logo to view the file directory.

```
file_name = "Text.txt"
with open(file_name, 'w') as file:
    file.write(text)
```

Chunking

```
chunk_size = 512
chunks = [text[i : i + chunk_size] for i in range(0, len(text), chunk_size)]
```

```
len(chunks)
```

```
62
```

Get embeddings of the chunks

```
import os
from mistralai import Mistral

def get_text_embedding(list_txt_chunks):
    client = Mistral(api_key=api_key)
    embeddings_batch_response = client.embeddings.create(model="mistral-embed",
                                                         inputs=list_txt_chunks)
    return embeddings_batch_response.data
```

```
import numpy as np

text_embeddings = get_text_embedding(chunks)
```

```
len(text_embeddings)
```

```
62
```

```
len(text_embeddings[0].embedding)
```

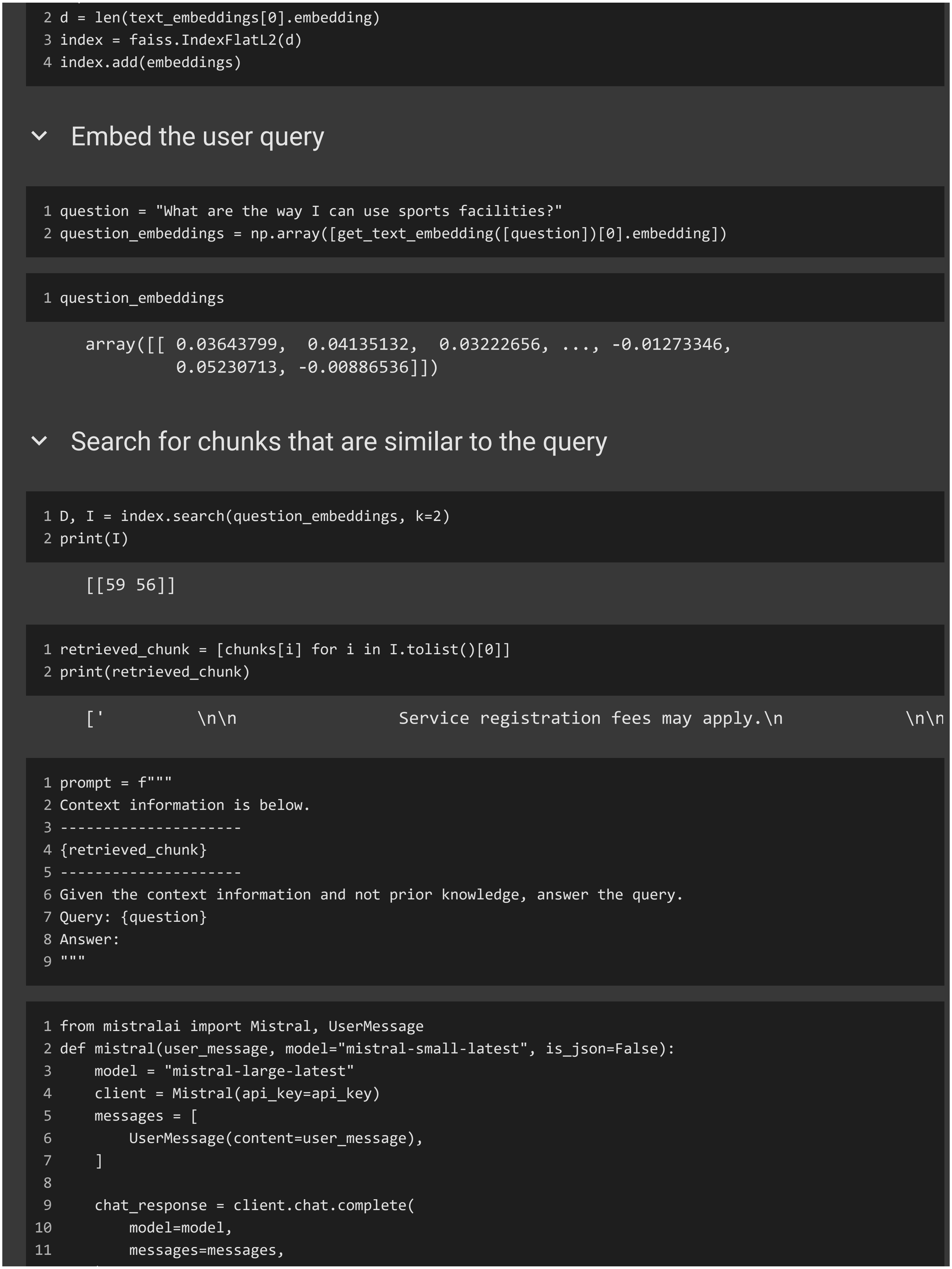
```
1024
```

```
embeddings = np.array([text_embeddings[i].embedding for i in range(len(text_embeddings))])
```

Store in a vector databsae

- In this lab, you'll use Faiss

```
import faiss
```



```
d = len(text_embeddings[0].embedding)
index = faiss.IndexFlatL2(d)
index.add(embeddings)
```

Embed the user query

```
question = "What are the way I can use sports facilities?"
question_embeddings = np.array([get_text_embedding([question])[0].embedding])
```

```
question_embeddings
```

```
array([[ 0.03643799,  0.04135132,  0.03222656, ..., -0.01273346,
         0.05230713, -0.00886536]])
```

Search for chunks that are similar to the query

```
D, I = index.search(question_embeddings, k=2)
print(I)
```

```
[[59 56]]
```

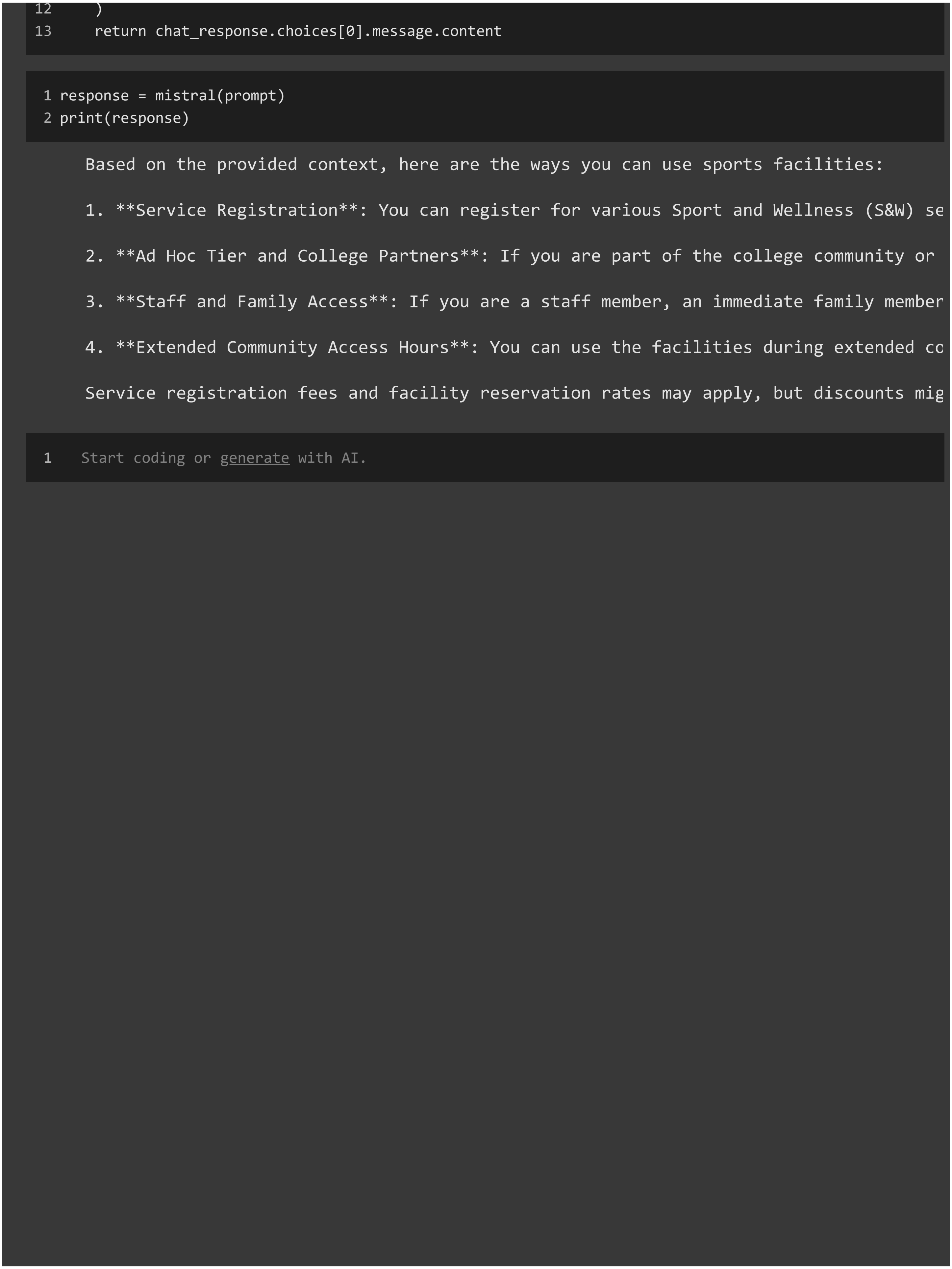
```
retrieved_chunk = [chunks[i] for i in I.tolist()[0]]
print(retrieved_chunk)
```

```
['         \n\n                  Service registration fees may apply.\n             \n\n
```

```
prompt = f"""
Context information is below.
---------------------
{retrieved_chunk}
---------------------
Given the context information and not prior knowledge, answer the query.
Query: {question}
Answer:
"""
```

```
from mistralai import Mistral, UserMessage
def mistral(user_message, model="mistral-small-latest", is_json=False):
    model = "mistral-large-latest"
    client = Mistral(api_key=api_key)
    messages = [
        UserMessage(content=user_message),
    ]

    chat_response = client.chat.complete(
        model=model,
        messages=messages,
```



```
    )
    return chat_response.choices[0].message.content
```

```
response = mistral(prompt)
print(response)
```

```
Based on the provided context, here are the ways you can use sports facilities:

1. **Service Registration**: You can register for various Sport and Wellness (S&W) se

2. **Ad Hoc Tier and College Partners**: If you are part of the college community or 

3. **Staff and Family Access**: If you are a staff member, an immediate family member

4. **Extended Community Access Hours**: You can use the facilities during extended co

Service registration fees and facility reservation rates may apply, but discounts mig
```

```
Start coding or generate with AI.
```