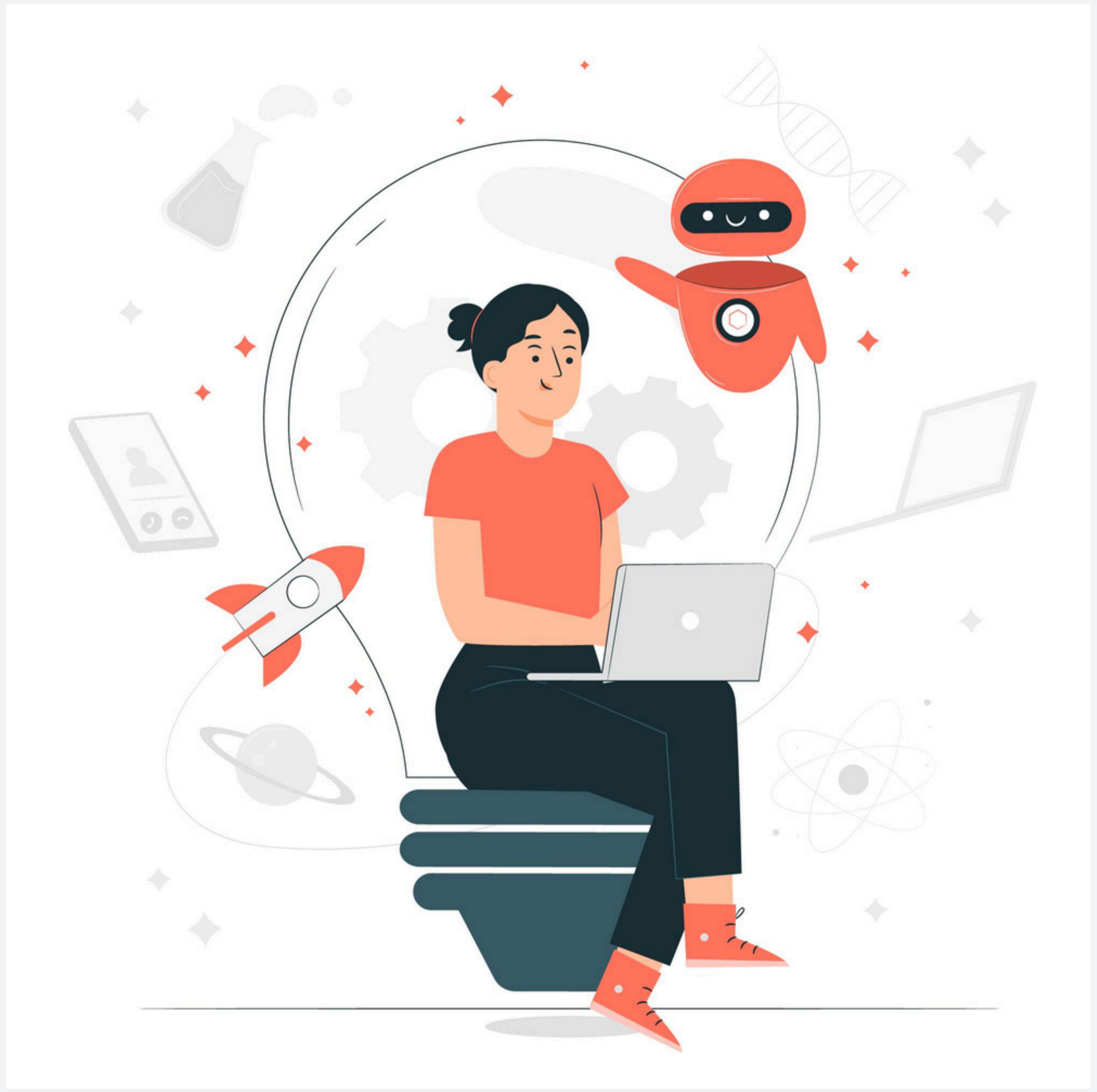
Build your own Chatbot

By Integrating LLM with Streamlit

Like | Share | Subscribe

Content

Getting started



What is Ollama?

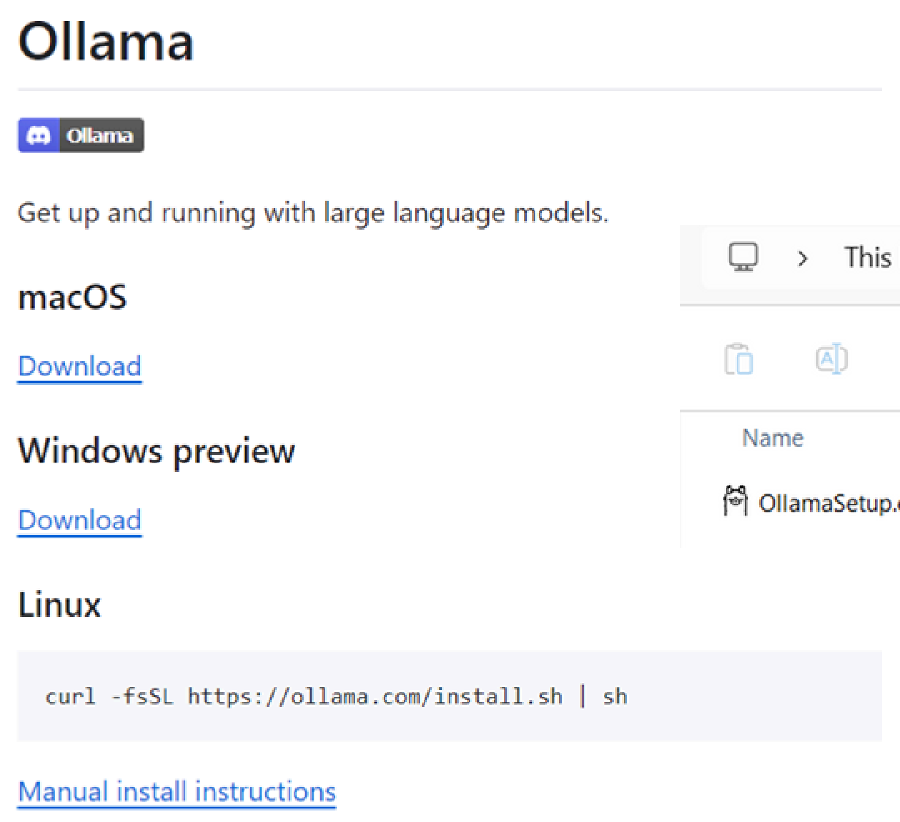
Ollama is a tool that allows users to run open-source large language models (LLMs) locally on their machines. It supports a range of models, such as Llama 3, Gemma, Mistral, and other open-source options. With its straightforward API, Ollama facilitates the creation, execution, and management of language models. Designed for simplicity and flexibility, it makes running LLMs on a local setup easy and efficient.

What is Streamlit?

Streamlit is an open-source library for building web applications using Python. Tailored for data science and machine learning, it's ideal for creating interactive dashboards and applications, including chatbots. Streamlit enables users to easily share these applications with others, offering a simple and effective way to showcase data and models.

Ollama - Download & Installation

Step 1	**Visit the Ollama Github Page:** https://github.com/Ollama/llama3
	Download Ollama: Choose the appropriate version for macOS, Windows, or Linux.
Step 2	**Run the Installer:** Execute OllamaSetup.exe and follow the prompts to complete the installation.
	Installation Directory: Ollama will be installed in C:\Users\Programs



Ollama

Get up and running with large language models.

macOS

Download

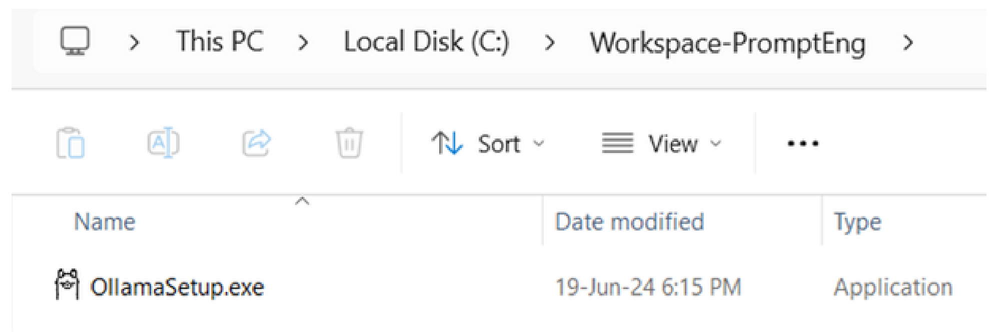
Windows preview

Download

Linux

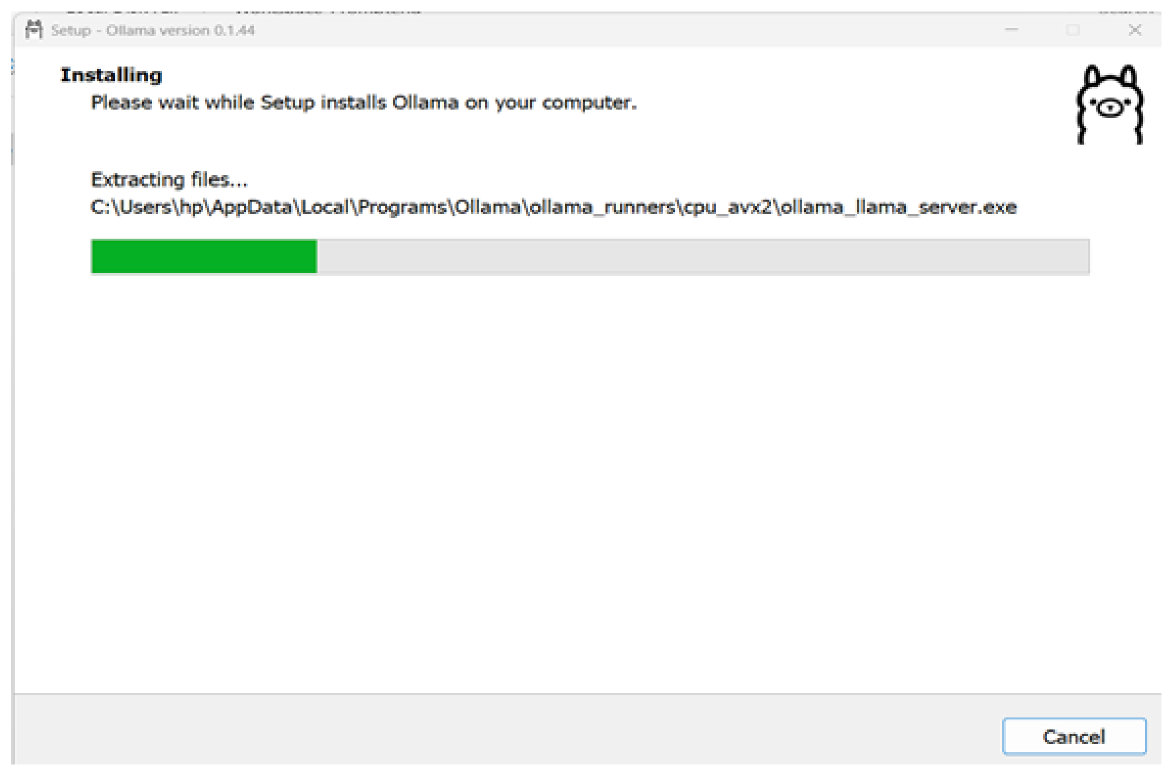
```
curl -fsSL https://ollama.com/install.sh | sh
```

Manual install instructions



This PC > Local Disk (C:) > Workspace-PromptEng >

Name	Date modified	Type
OllamaSetup.exe	19-Jun-24 6:15 PM	Application



Setup - Ollama version 0.1.44

Installing

Please wait while Setup installs Ollama on your computer.

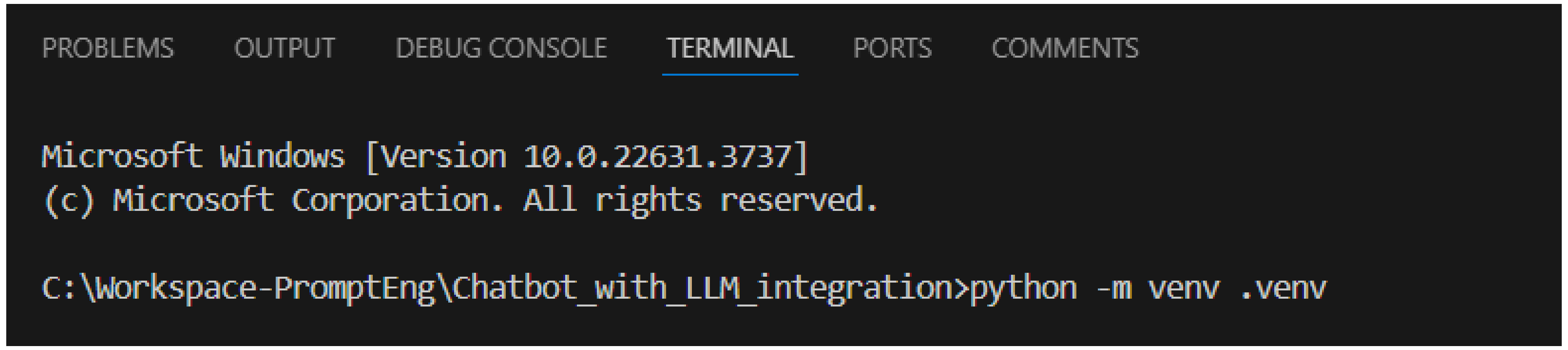
Extracting files...
C:\Users\hp\AppData\Local\Programs\Ollama\ollama_runners\cpu_avx2\ollama_llama_server.exe

Setting up Environment

STEP 1

Open a terminal, go to your project directory, and create a virtual environment using command

python -m venv .venv

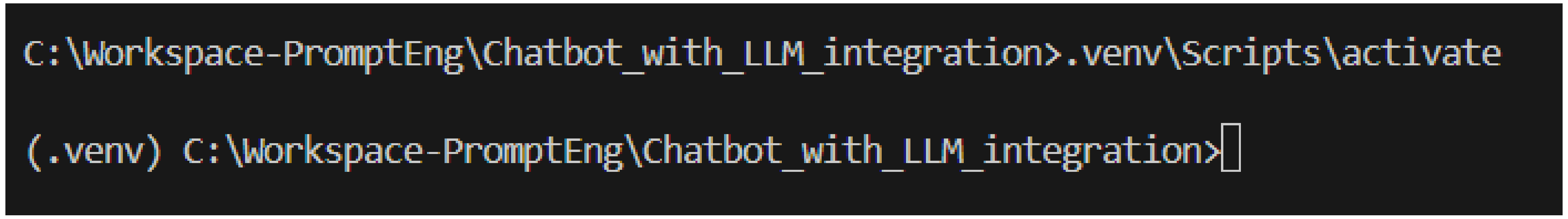


```
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

C:\Workspace-PromptEng\Chatbot_with_LLM_integration>python -m venv .venv
```

STEP 2

Activate the virtual environment

.venv\Scripts\activate



```
C:\Workspace-PromptEng\Chatbot_with_LLM_integration>.venv\Scripts\activate

(.venv) C:\Workspace-PromptEng\Chatbot_with_LLM_integration>
```

Setting up Environment

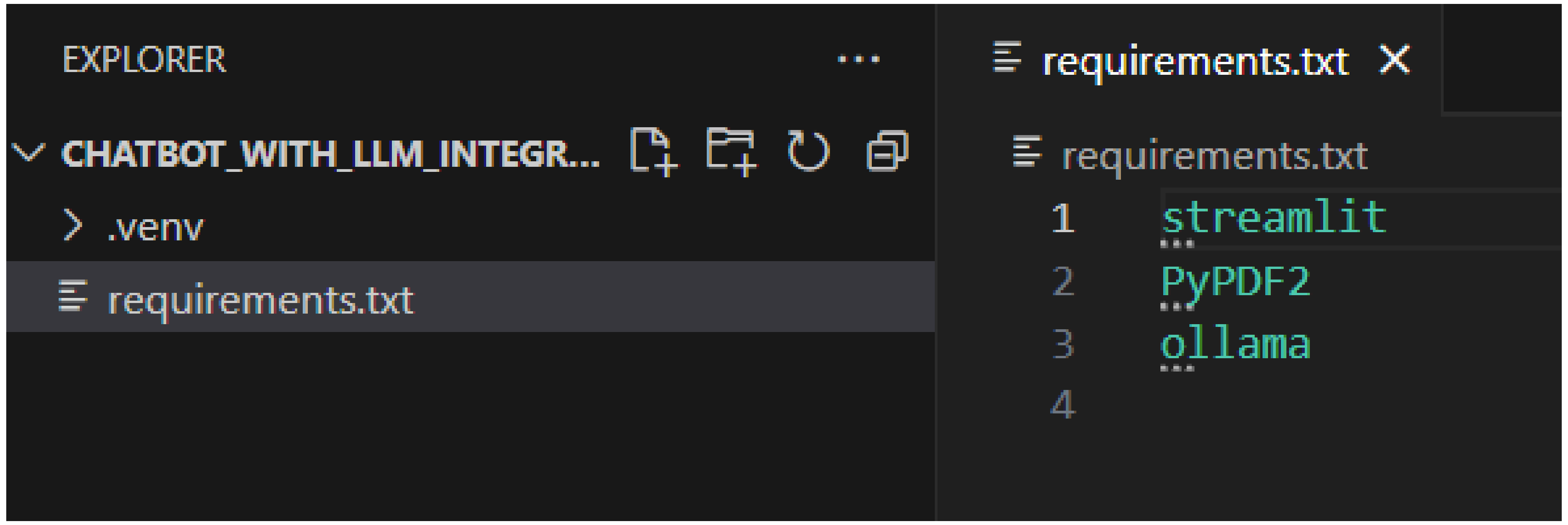
STEP 3

Now, create a **requirements.txt** file and paste the following content: streamlit
PyPDF2
ollama

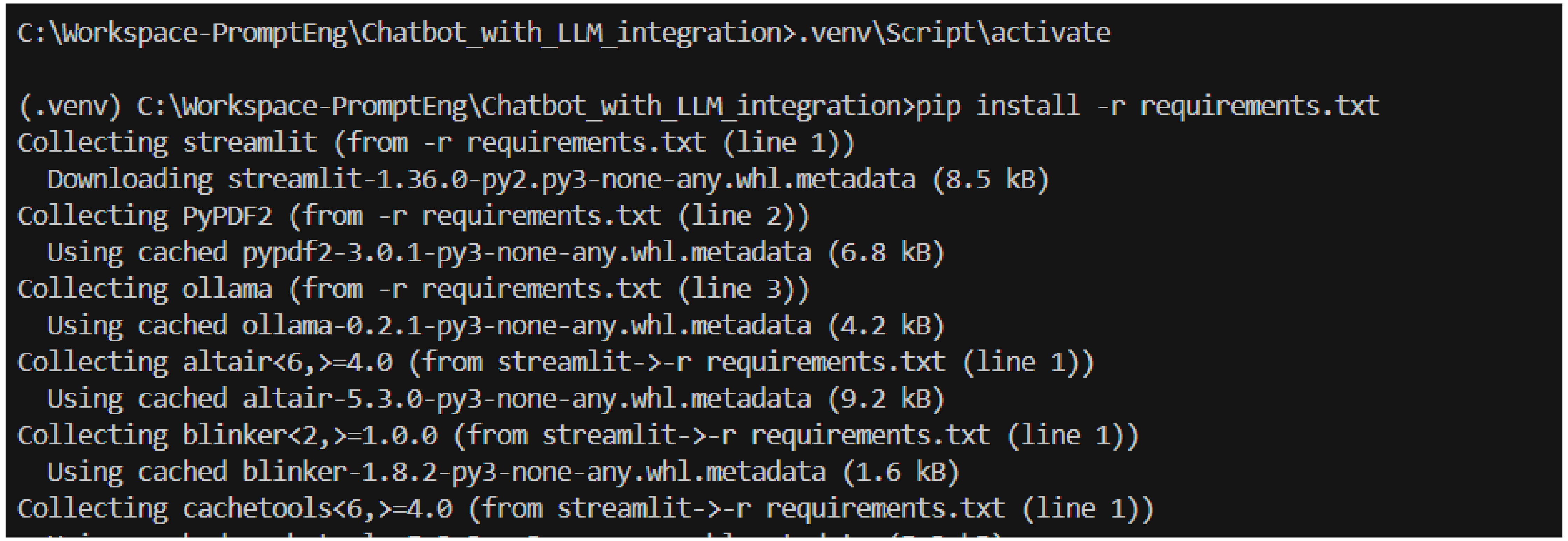
And then run:

pip install -r requirments.txt

(make sure your .venv is in active mode)

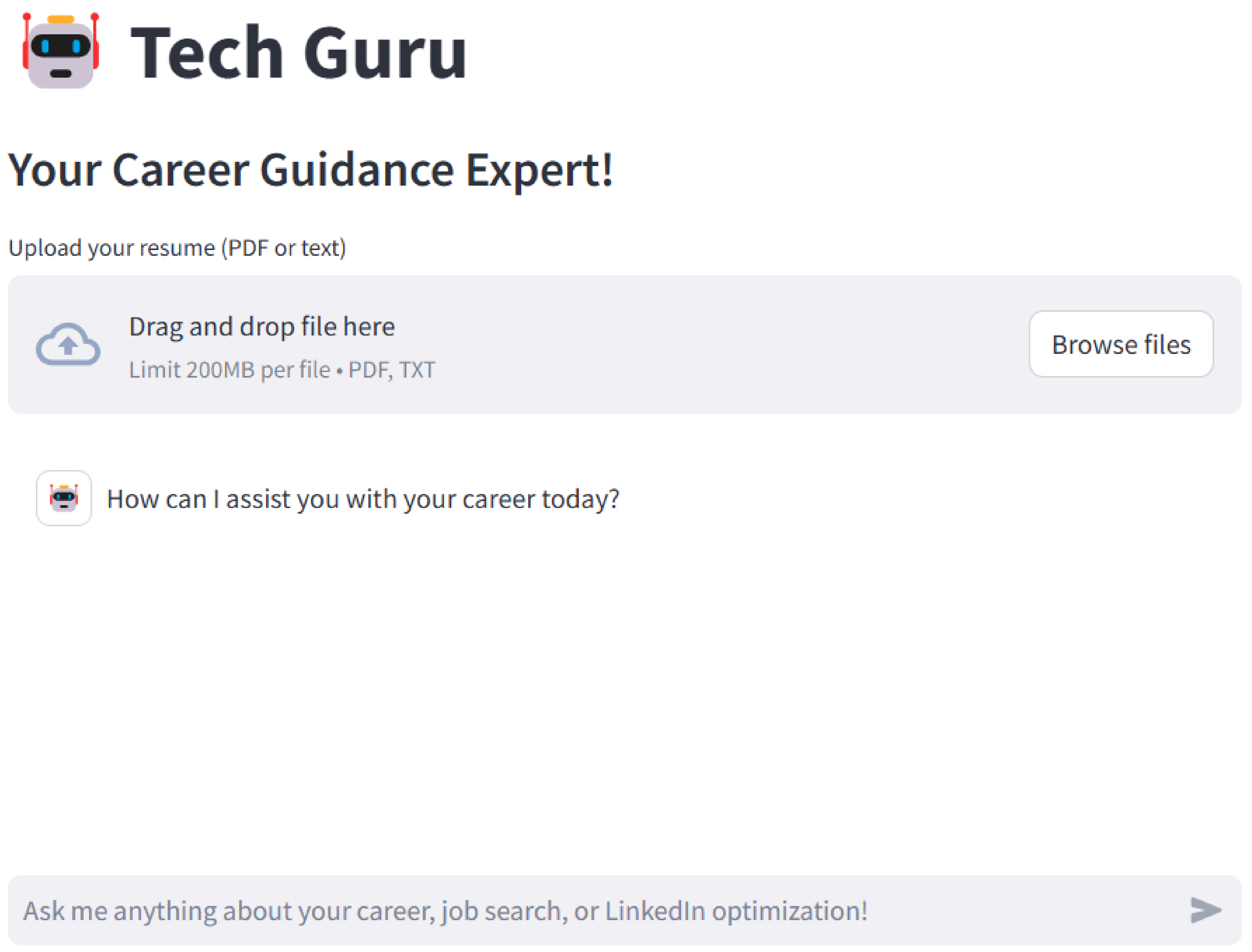


```
streamlit
PyPDF2
ollama
```



```
C:\Workspace-PromptEng\Chatbot_with_LLM_integration>.venv\Script\activate

(.venv) C:\Workspace-PromptEng\Chatbot_with_LLM_integration>pip install -r requirements.txt
Collecting streamlit (from -r requirements.txt (line 1))
  Downloading streamlit-1.36.0-py2.py3-none-any.whl.metadata (8.5 kB)
Collecting PyPDF2 (from -r requirements.txt (line 2))
  Using cached pypdf2-3.0.1-py3-none-any.whl.metadata (6.8 kB)
Collecting ollama (from -r requirements.txt (line 3))
  Using cached ollama-0.2.1-py3-none-any.whl.metadata (4.2 kB)
Collecting altair<6,>=4.0 (from streamlit->-r requirements.txt (line 1))
  Using cached altair-5.3.0-py3-none-any.whl.metadata (9.2 kB)
Collecting blinker<2,>=1.0.0 (from streamlit->-r requirements.txt (line 1))
  Using cached blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting cachetools<6,>=4.0 (from streamlit->-r requirements.txt (line 1))
```

Let's start Building our Chatbot !



Tech Guru

Your Career Guidance Expert!

Upload your resume (PDF or text)

Drag and drop file here

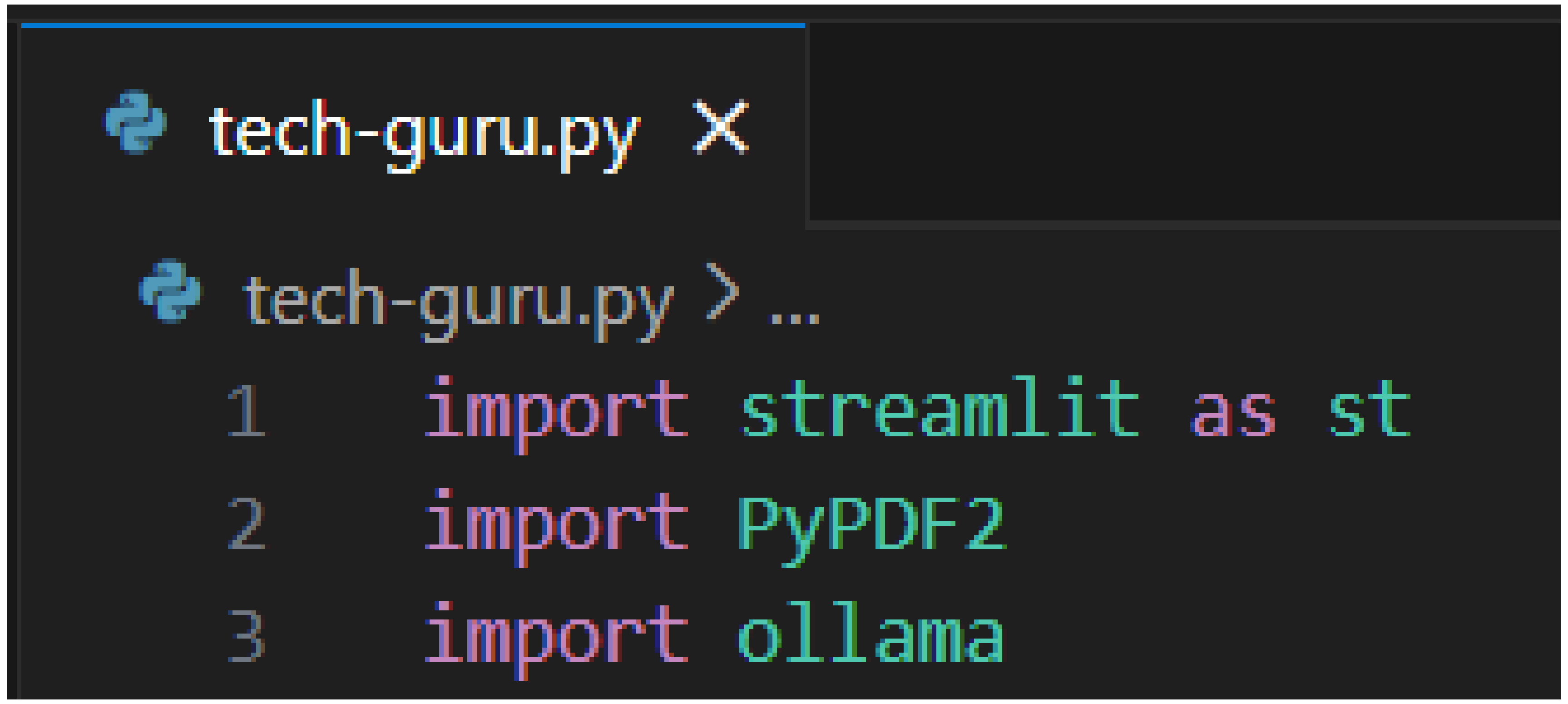
Limit 200MB per file • PDF, TXT

Browse files

How can I assist you with your career today?

Ask me anything about your career, job search, or LinkedIn optimization!

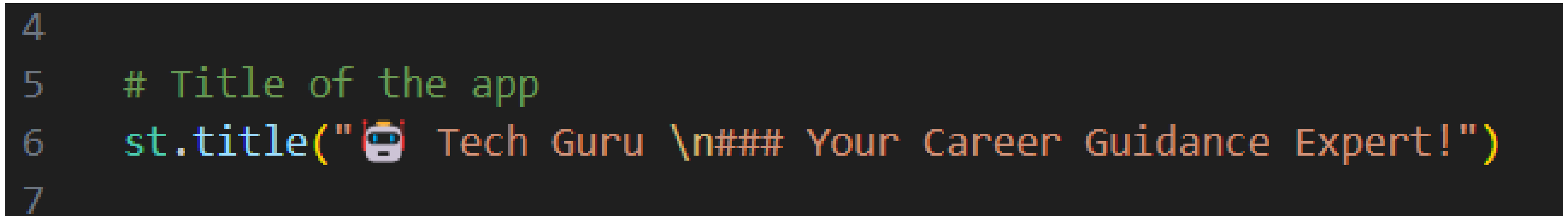
Step1: Firstly, create a python file: "tech-guru.py" (you can give any name) and import streamlit, ollama, PyPDF2


tech-guru.py

tech-guru.py > ...

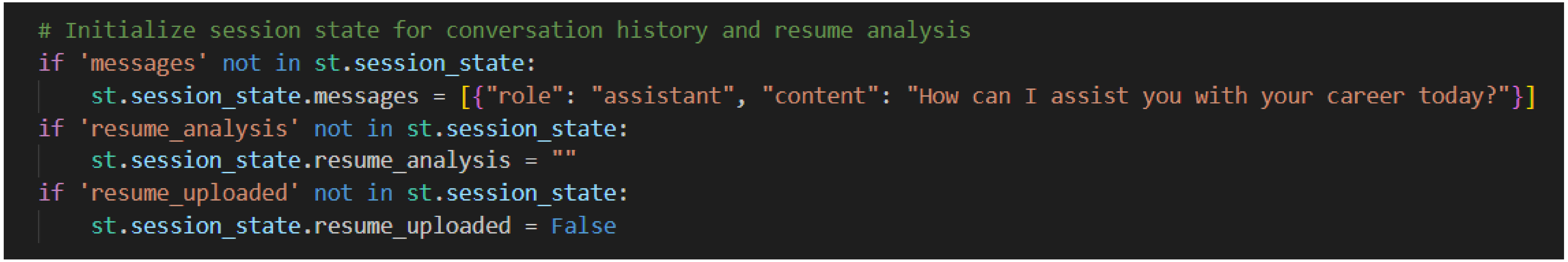
```python
import streamlit as st
import PyPDF2
import ollama
```

Step2: Then give the title of the app using st.title



```python

# Title of the app
st.title("🤖 Tech Guru \n### Your Career Guidance Expert!")

```

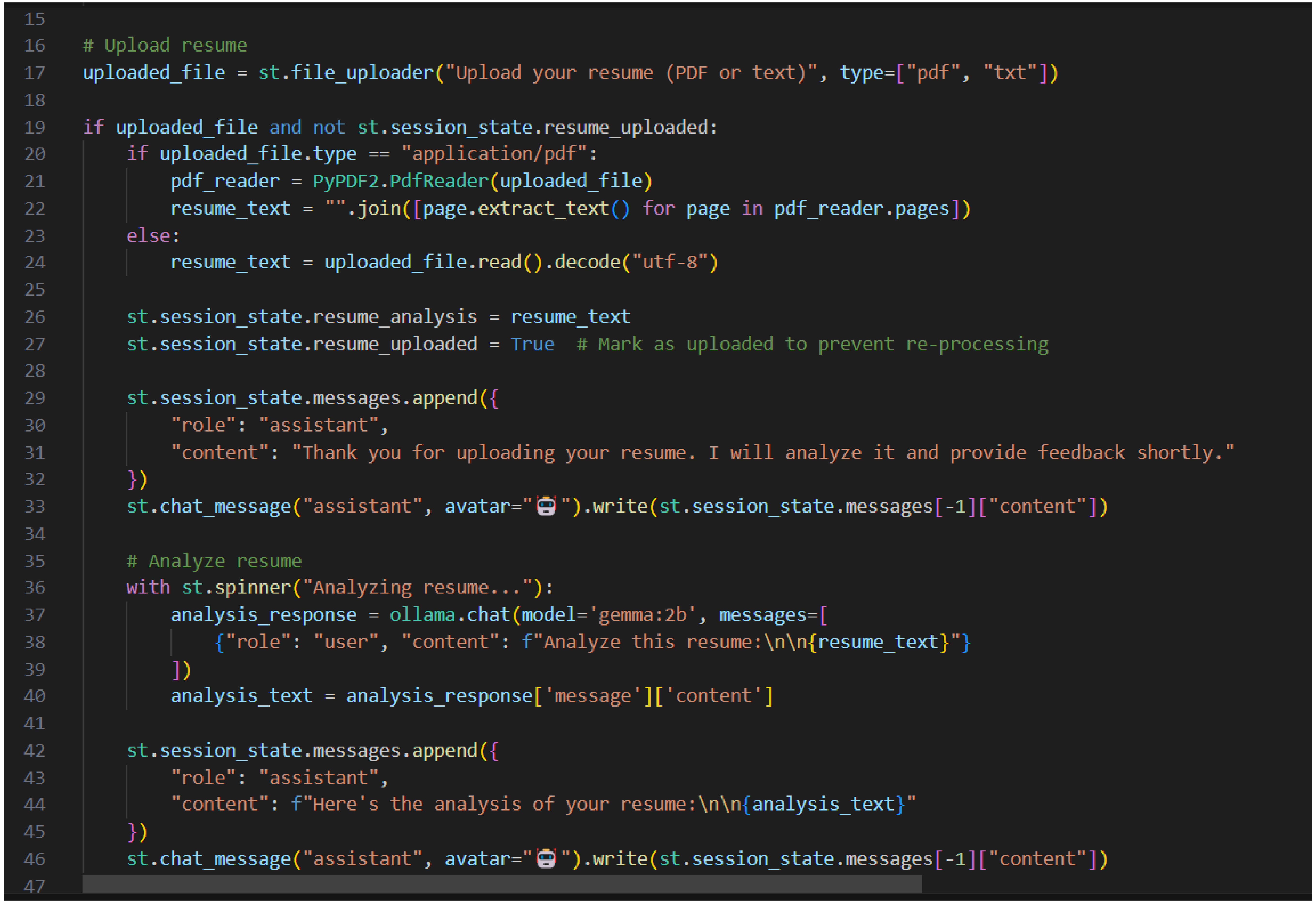
Step3: Initialize session state for conversation history (messages), the resume analysis text (resume analysis), and a flag indicating whether a resume has been uploaded (resume_uploaded)



```python
# Initialize session state for conversation history and resume analysis
if 'messages' not in st.session_state:
    st.session_state.messages = [{"role": "assistant", "content": "How can I assist you with your career today?"}]
if 'resume_analysis' not in st.session_state:
    st.session_state.resume_analysis = ""
if 'resume_uploaded' not in st.session_state:
    st.session_state.resume_uploaded = False
```

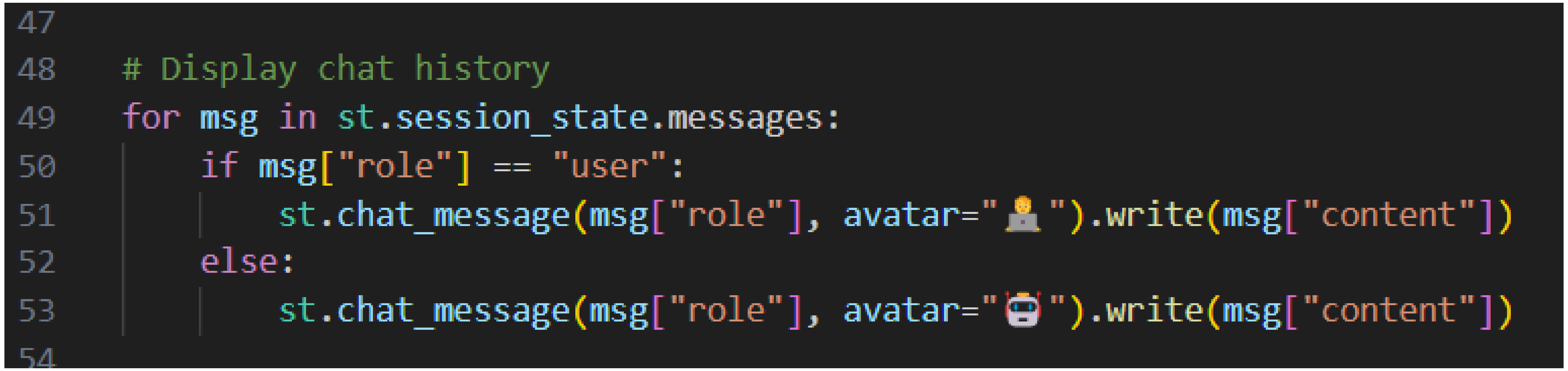
1. messages: To store and manage the chat history between the user and the assistant.
2. resume_analysis: To store the text extracted from the user's uploaded resume.
3. resume_uploaded: A flag to track whether a resume has been uploaded, preventing redundant processing.

Step 4: Resume Upload and Analysis



```python

# Upload resume
uploaded_file = st.file_uploader("Upload your resume (PDF or text)", type=["pdf", "txt"])

if uploaded_file and not st.session_state.resume_uploaded:
    if uploaded_file.type == "application/pdf":
        pdf_reader = PyPDF2.PdfReader(uploaded_file)
        resume_text = "".join([page.extract_text() for page in pdf_reader.pages])
    else:
        resume_text = uploaded_file.read().decode("utf-8")

    st.session_state.resume_analysis = resume_text
    st.session_state.resume_uploaded = True  # Mark as uploaded to prevent re-processing

    st.session_state.messages.append({
        "role": "assistant",
        "content": "Thank you for uploading your resume. I will analyze it and provide feedback shortly."
    })
    st.chat_message("assistant", avatar="🤖").write(st.session_state.messages[-1]["content"])

    # Analyze resume
    with st.spinner("Analyzing resume..."):
        analysis_response = ollama.chat(model='gemma:2b', messages=[
            {"role": "user", "content": f"Analyze this resume:\n\n{resume_text}"}
        ])
        analysis_text = analysis_response['message']['content']

    st.session_state.messages.append({
        "role": "assistant",
        "content": f"Here's the analysis of your resume:\n\n{analysis_text}"
    })
    st.chat_message("assistant", avatar="🤖").write(st.session_state.messages[-1]["content"])

```

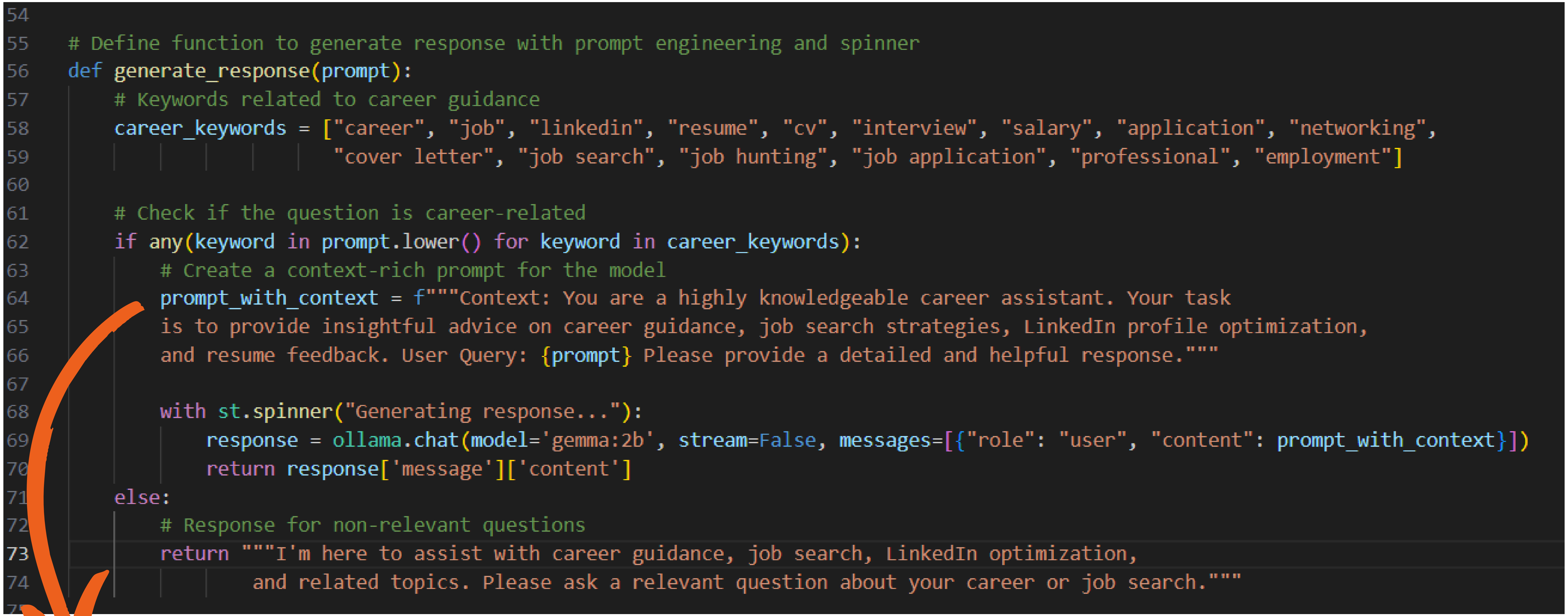
Step 5: Displaying Chat History



```python
# Display chat history
for msg in st.session_state.messages:
    if msg["role"] == "user":
        st.chat_message(msg["role"], avatar="🧑‍💻").write(msg["content"])
    else:
        st.chat_message(msg["role"], avatar="🤖").write(msg["content"])
```

Iterates through the conversation history **(st.session_state.messages)** and displays each message with appropriate avatars for the user (🧑‍💻) and the assistant (🤖).

Step 6: Defining Function to Generate Response

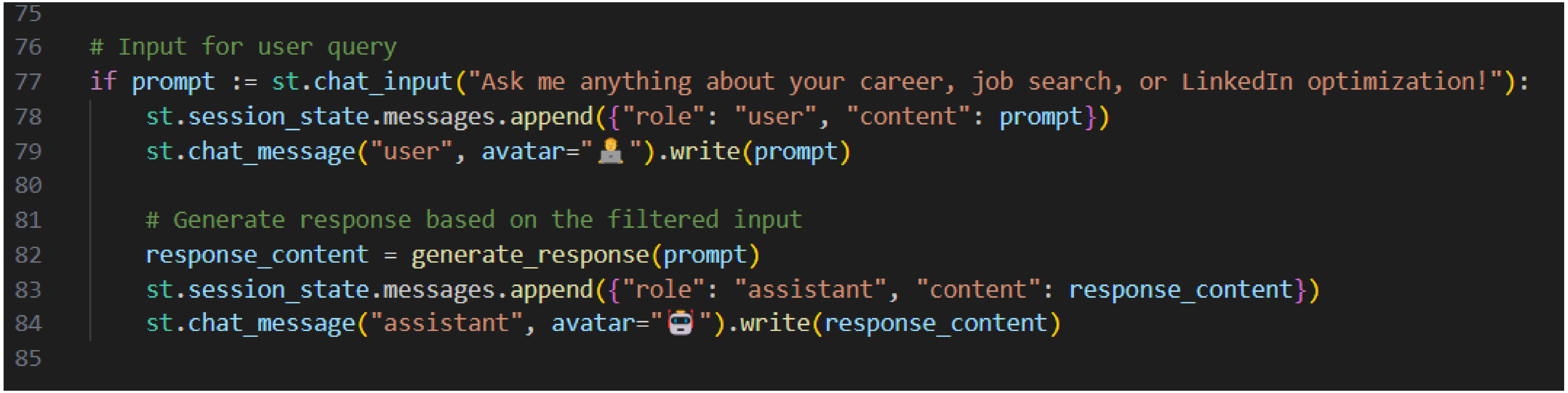


```python

# Define function to generate response with prompt engineering and spinner
def generate_response(prompt):
    # Keywords related to career guidance
    career_keywords = ["career", "job", "linkedin", "resume", "cv", "interview", "salary", "application", "networking",
                       "cover letter", "job search", "job hunting", "job application", "professional", "employment"]

    # Check if the question is career-related
    if any(keyword in prompt.lower() for keyword in career_keywords):
        # Create a context-rich prompt for the model
        prompt_with_context = f"""Context: You are a highly knowledgeable career assistant. Your task
        is to provide insightful advice on career guidance, job search strategies, LinkedIn profile optimization,
        and resume feedback. User Query: {prompt} Please provide a detailed and helpful response."""

        with st.spinner("Generating response..."):
            response = ollama.chat(model='gemma:2b', stream=False, messages=[{"role": "user", "content": prompt_with_context}])
            return response['message']['content']
    else:
        # Response for non-relevant questions
        return """I'm here to assist with career guidance, job search, LinkedIn optimization,
                and related topics. Please ask a relevant question about your career or job search."""
```

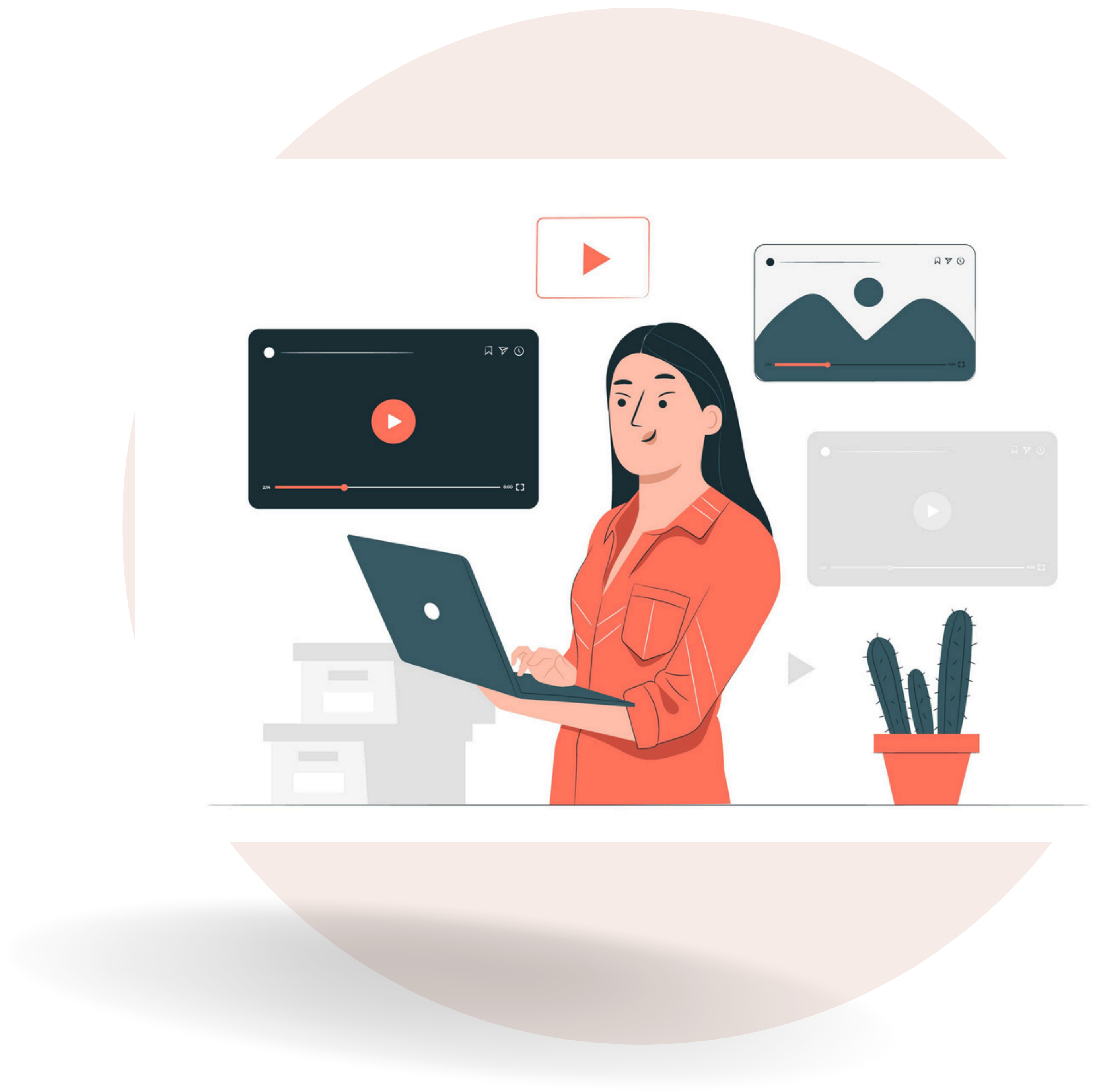
Prompt Engineering Pattern used: Persona Pattern

Here the persona is a "highly knowledgeable career assistant."

Step 7: Handling User Input and Generating Response



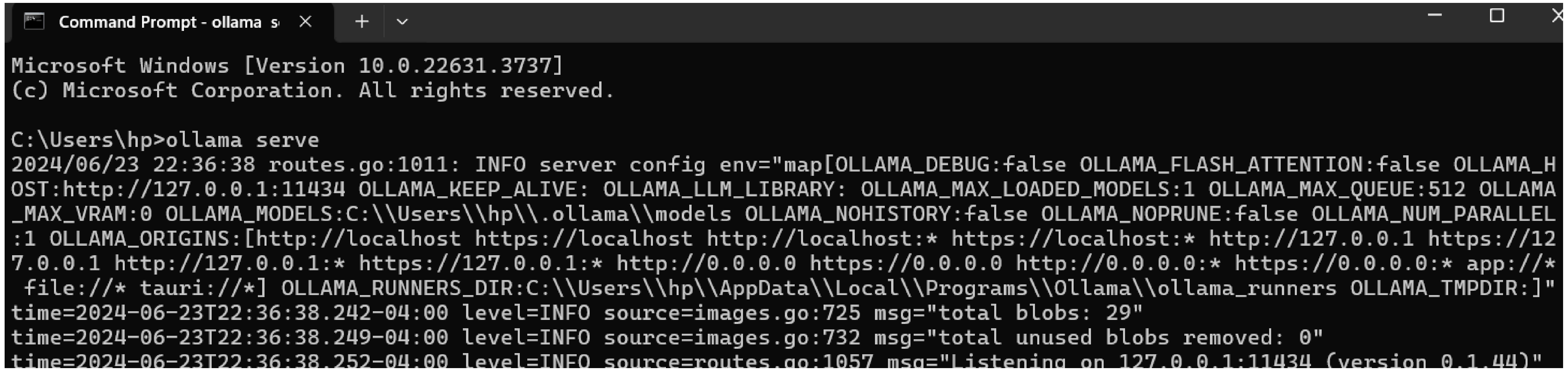
```python

# Input for user query
if prompt := st.chat_input("Ask me anything about your career, job search, or LinkedIn optimization!"):
    st.session_state.messages.append({"role": "user", "content": prompt})
    st.chat_message("user", avatar="🧑‍💻").write(prompt)

    # Generate response based on the filtered input
    response_content = generate_response(prompt)
    st.session_state.messages.append({"role": "assistant", "content": response_content})
    st.chat_message("assistant", avatar="🤖").write(response_content)

```



Lets run the Chatbot now!

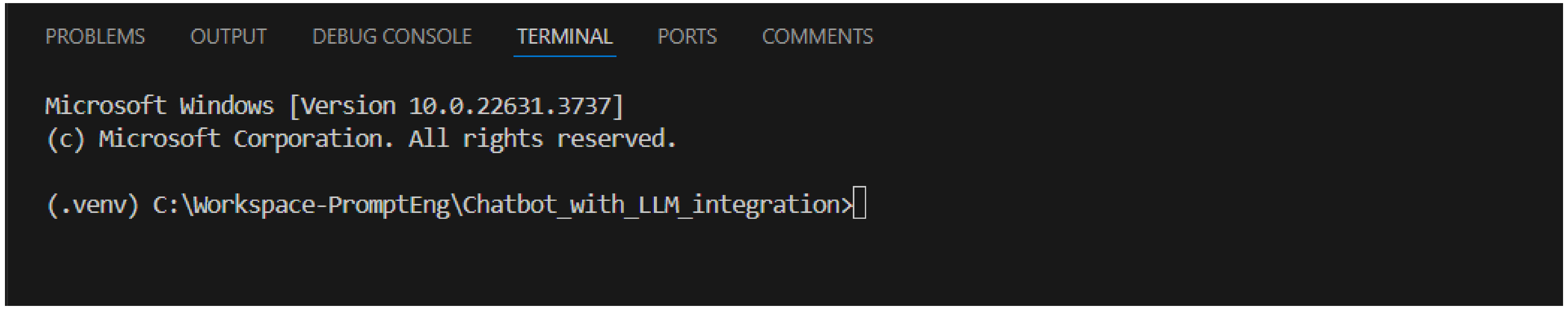
Step 1: Open new terminal/cmd and run Ollama server using below command:

C:\Users\hp>ollama serve



```
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>ollama serve
2024/06/23 22:36:38 routes.go:1011: INFO server config env="map[OLLAMA_DEBUG:false OLLAMA_FLASH_ATTENTION:false OLLAMA_H
OST:http://127.0.0.1:11434 OLLAMA_KEEP_ALIVE: OLLAMA_LLM_LIBRARY: OLLAMA_MAX_LOADED_MODELS:1 OLLAMA_MAX_QUEUE:512 OLLAMA
_MAX_VRAM:0 OLLAMA_MODELS:C:\\Users\\hp\\.ollama\\models OLLAMA_NOHISTORY:false OLLAMA_NOPRUNE:false OLLAMA_NUM_PARALLEL
:1 OLLAMA_ORIGINS:[http://localhost https://localhost http://localhost:* https://localhost:* http://127.0.0.1 https://12
7.0.0.1 http://127.0.0.1:* https://127.0.0.1:* http://0.0.0.0 https://0.0.0.0 http://0.0.0.0:* https://0.0.0.0:* app://*
 file://* tauri://*] OLLAMA_RUNNERS_DIR:C:\\Users\\hp\\AppData\\Local\\Programs\\Ollama\\ollama_runners OLLAMA_TMPDIR:]"
time=2024-06-23T22:36:38.242-04:00 level=INFO source=images.go:725 msg="total blobs: 29"
time=2024-06-23T22:36:38.249-04:00 level=INFO source=images.go:732 msg="total unused blobs removed: 0"
time=2024-06-23T22:36:38.252-04:00 level=INFO source=routes.go:1057 msg="Listening on 127.0.0.1:11434 (version 0.1.44)"
```

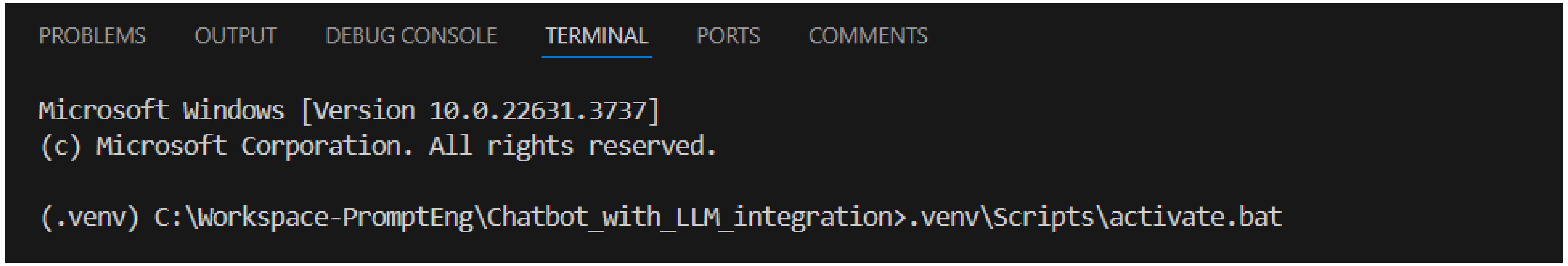
Step 2: Save the python file (tech-guru.py) and open the new terminal in VS Code



```
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

(.venv) C:\Workspace-PromptEng\Chatbot_with_LLM_integration>
```

Step 3: Activate the virtual environment using below command in VS code's terminal:

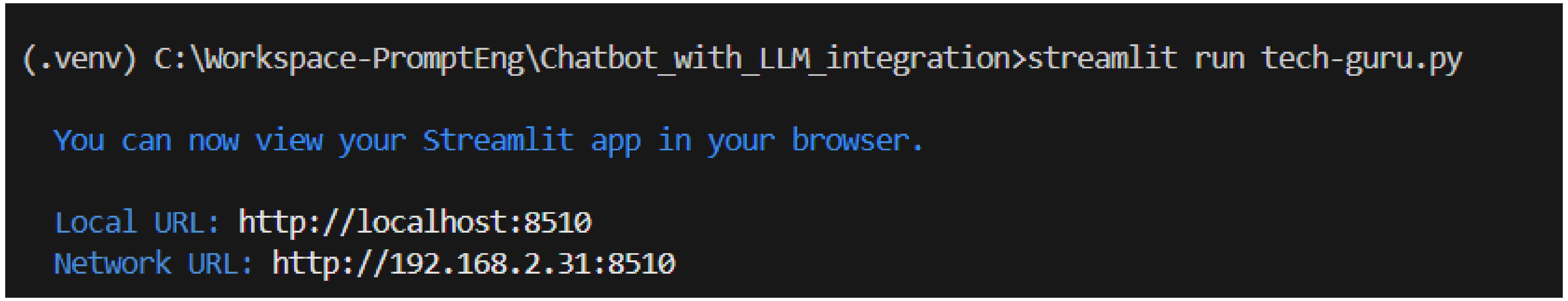
.   venv\Scripts\activate.bat



```
Microsoft Windows [Version 10.0.22631.3737]
(c) Microsoft Corporation. All rights reserved.

(.venv) C:\Workspace-PromptEng\Chatbot_with_LLM_integration>.venv\Scripts\activate.bat
```

Step 4: Now run the python app using below command:

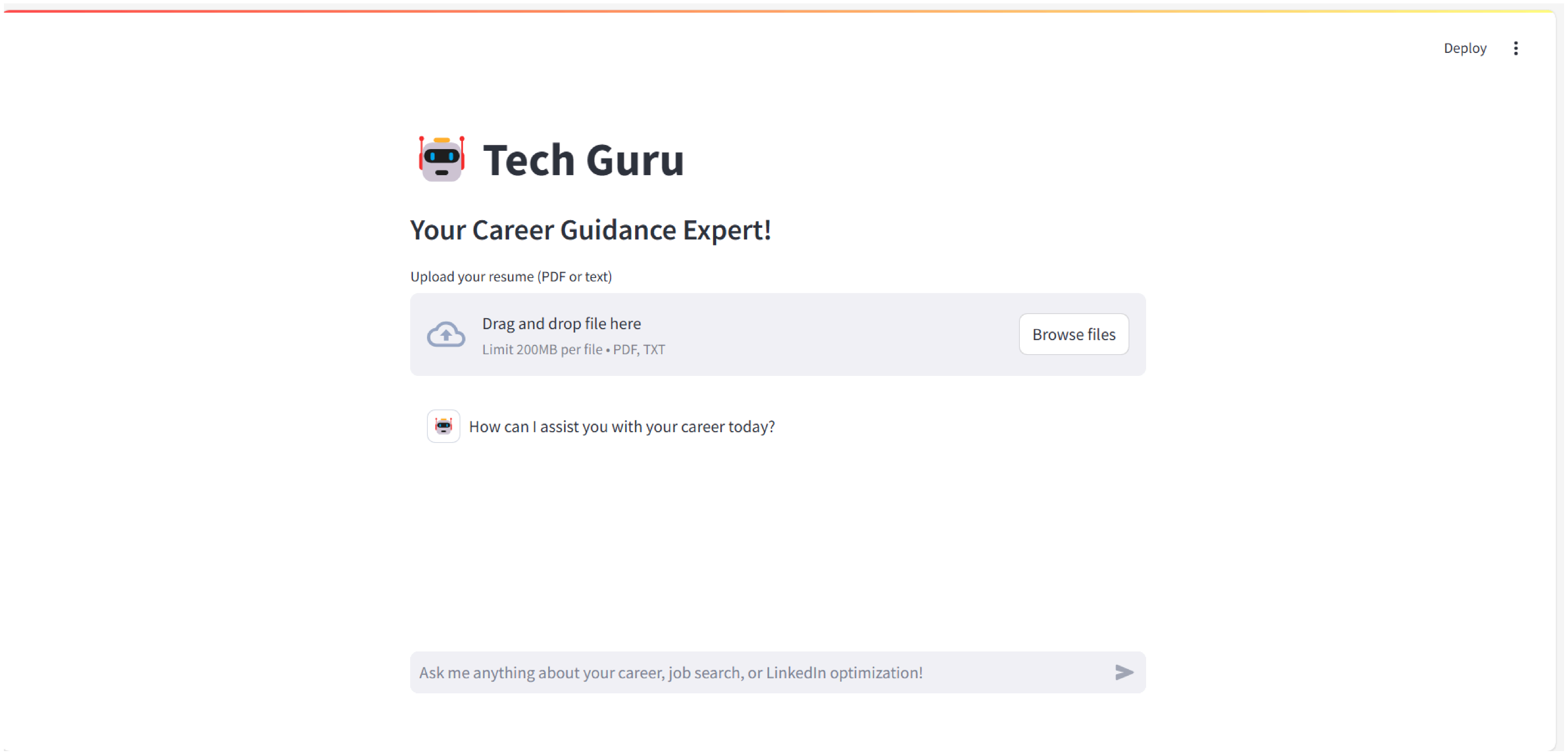
streamlit run tech-guru.py



```
(.venv) C:\Workspace-PromptEng\Chatbot_with_LLM_integration>streamlit run tech-guru.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8510
  Network URL: http://192.168.2.31:8510
```

Now your app is running on: http://localhost:8510

Streamlit will open the browser automatically and the app is running on: http://localhost:8510



Tech Guru

Your Career Guidance Expert!

Upload your resume (PDF or text)

Drag and drop file here

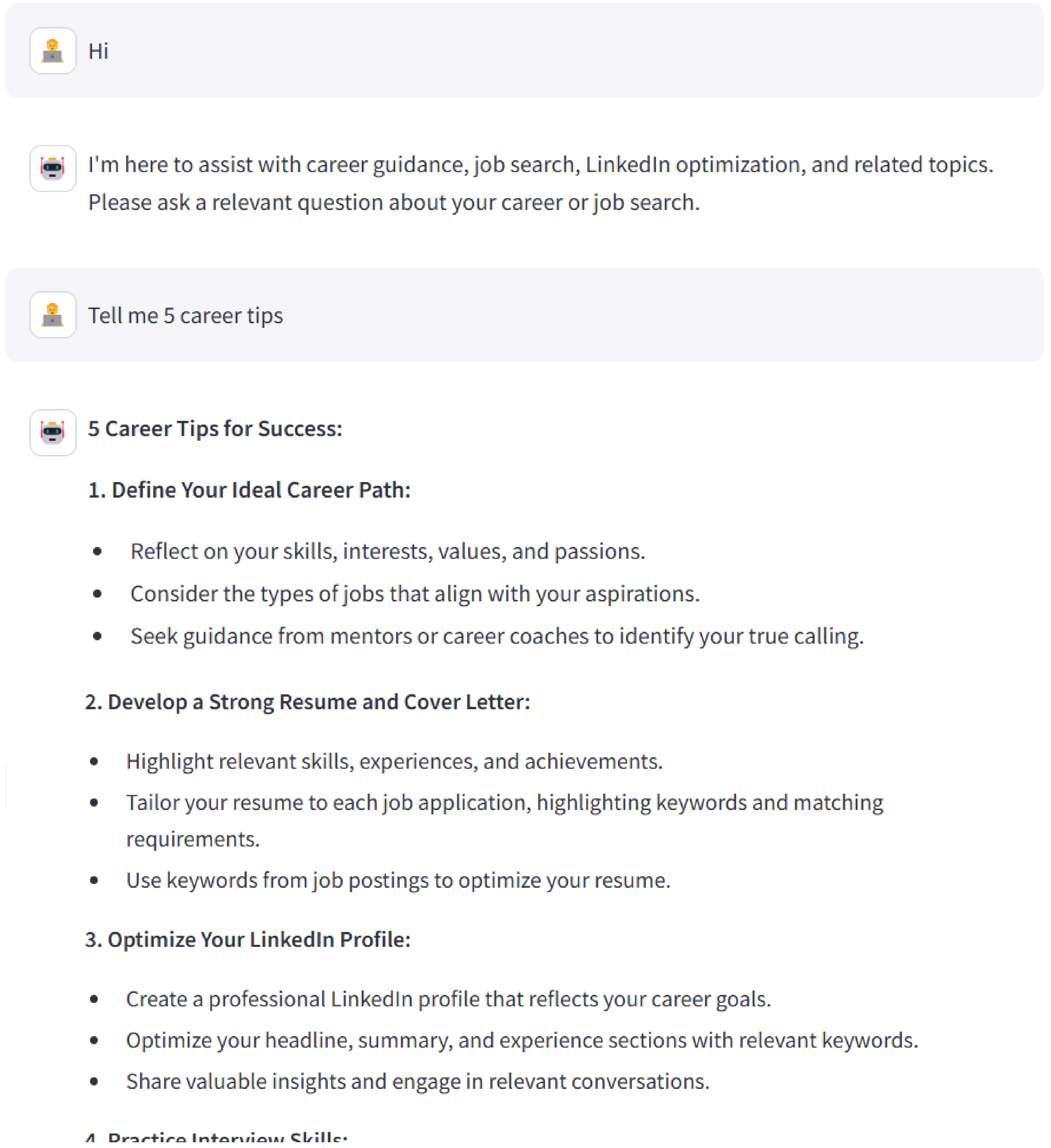
Limit 200MB per file • PDF, TXT

Browse files

How can I assist you with your career today?

Ask me anything about your career, job search, or LinkedIn optimization!

Ask a question related to career, job search or linkedIn optimization etc.



Hi

I'm here to assist with career guidance, job search, LinkedIn optimization, and related topics. Please ask a relevant question about your career or job search.

Tell me 5 career tips

5 Career Tips for Success:

1. Define Your Ideal Career Path:

- Reflect on your skills, interests, values, and passions.
- Consider the types of jobs that align with your aspirations.
- Seek guidance from mentors or career coaches to identify your true calling.

2. Develop a Strong Resume and Cover Letter:

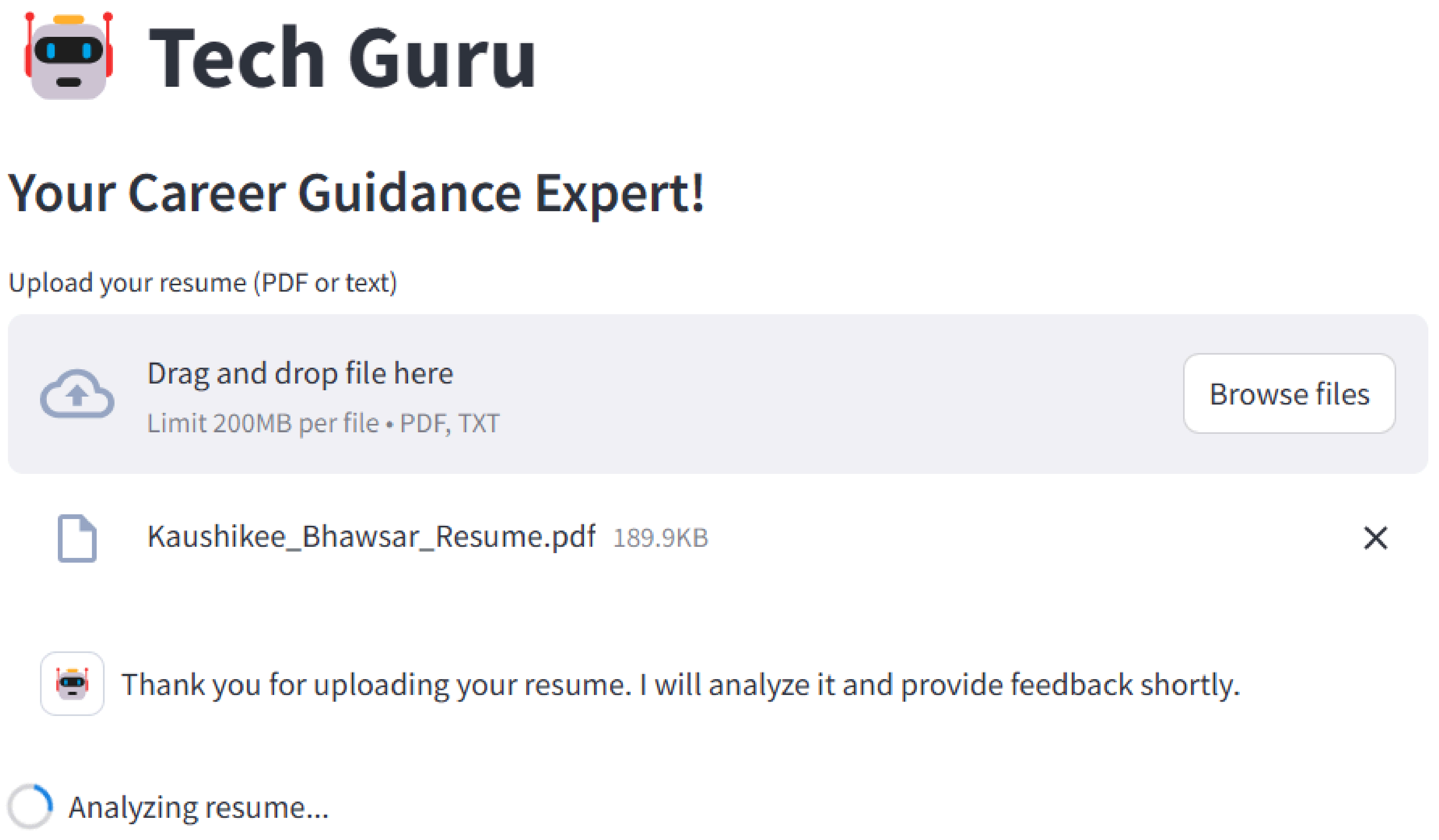
- Highlight relevant skills, experiences, and achievements.
- Tailor your resume to each job application, highlighting keywords and matching requirements.
- Use keywords from job postings to optimize your resume.

3. Optimize Your LinkedIn Profile:

- Create a professional LinkedIn profile that reflects your career goals.
- Optimize your headline, summary, and experience sections with relevant keywords.
- Share valuable insights and engage in relevant conversations.

4. Practice Interview Skills:

Or upload a resume!



Tech Guru

Your Career Guidance Expert!

Upload your resume (PDF or text)

Drag and drop file here
Limit 200MB per file • PDF, TXT

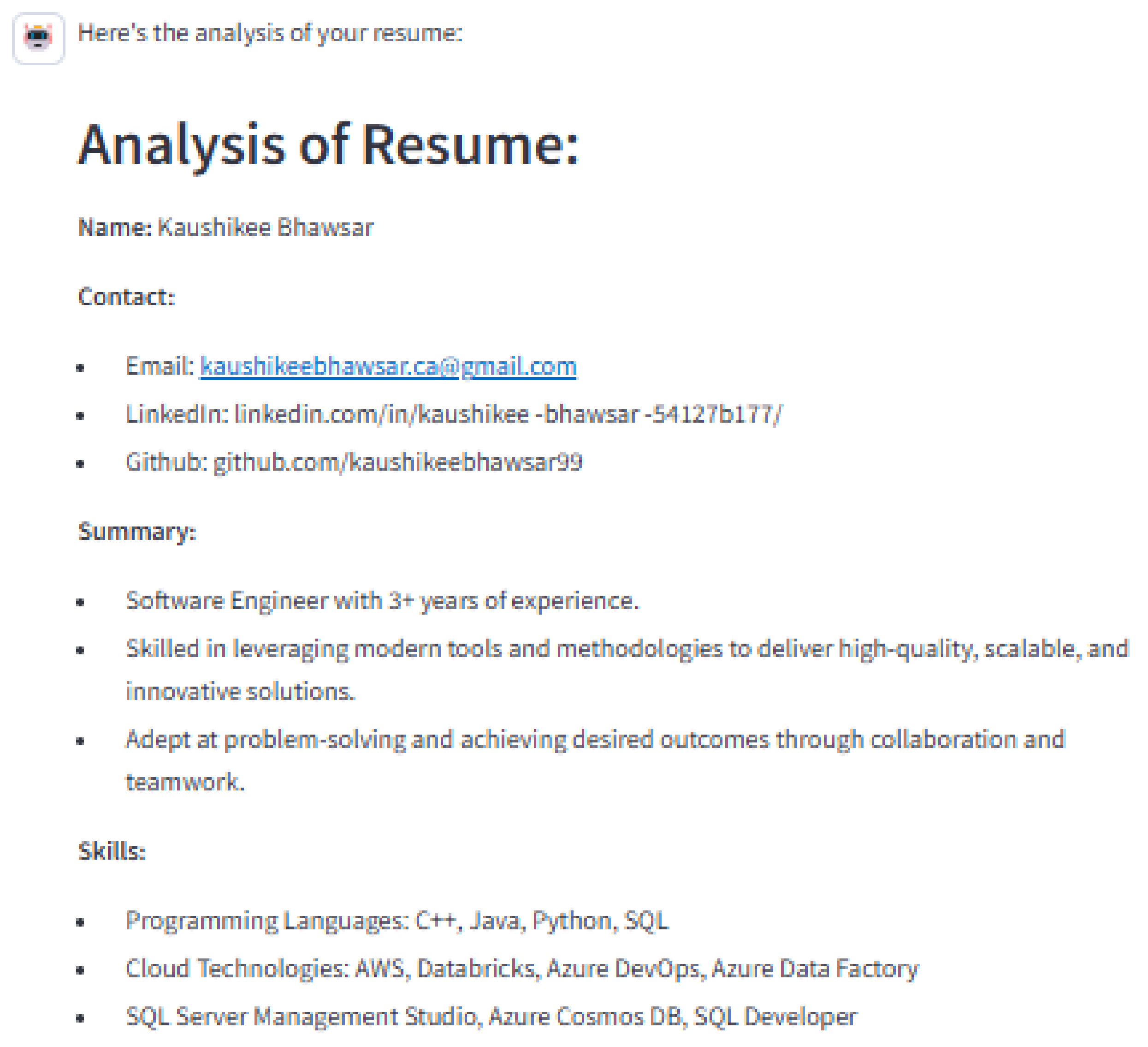
Browse files

Kaushikee_Bhawsar_Resume.pdf 189.9KB

Thank you for uploading your resume. I will analyze it and provide feedback shortly.

Analyzing resume...

It will provide the analysis of resume and you can ask questions further, for example -
"What are key strength and weakness in my resume ?"



Here's the analysis of your resume:

Analysis of Resume:

Name: Kaushikee Bhawsar

Contact:

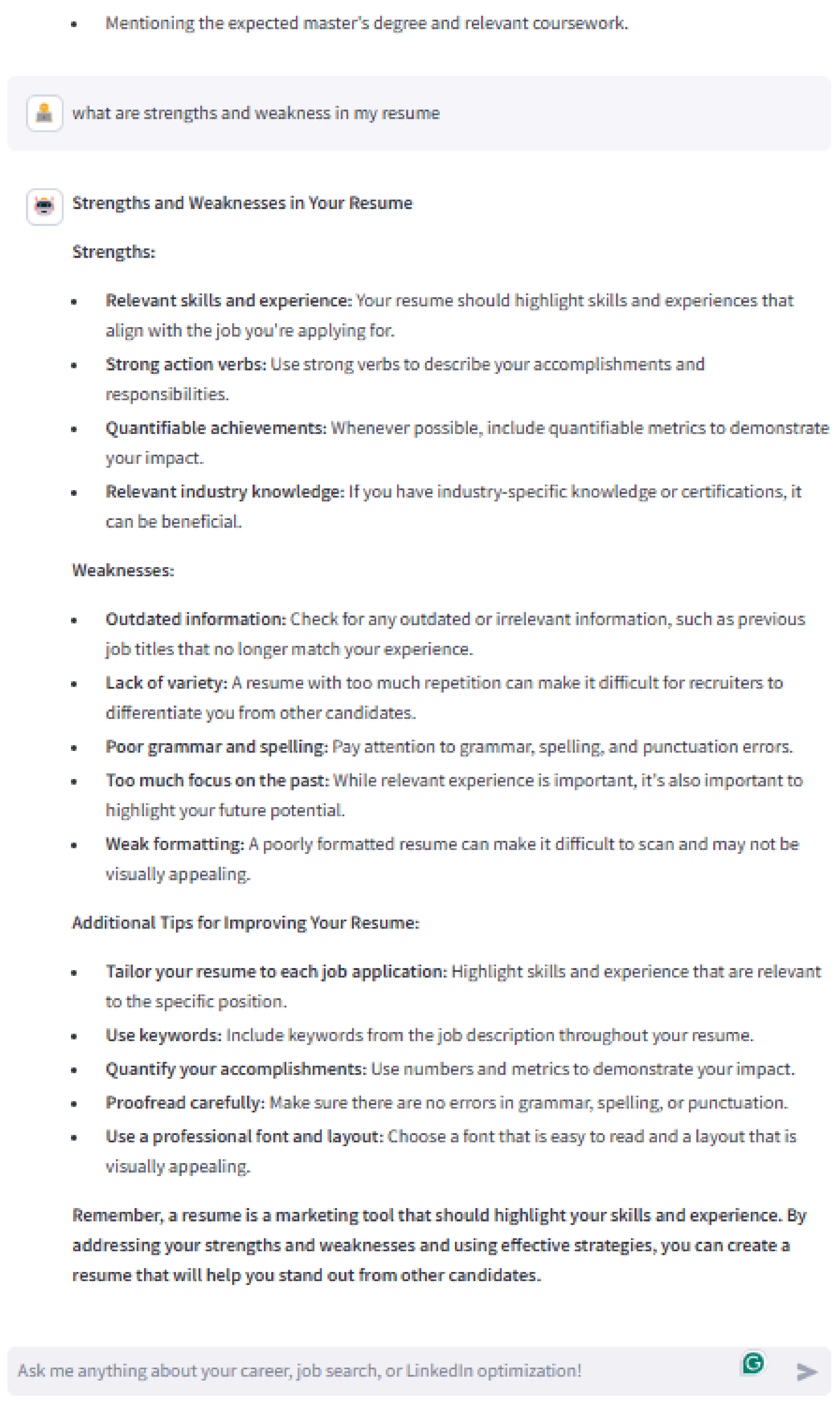
- Email: kaushikeebhawsar.ca@gmail.com
- LinkedIn: linkedin.com/in/kaushikee -bhawsar -54127b177/
- Github: github.com/kaushikeebhawsar99

Summary:

- Software Engineer with 3+ years of experience.
- Skilled in leveraging modern tools and methodologies to deliver high-quality, scalable, and innovative solutions.
- Adept at problem-solving and achieving desired outcomes through collaboration and teamwork.

Skills:

- Programming Languages: C++, Java, Python, SQL
- Cloud Technologies: AWS, Databricks, Azure DevOps, Azure Data Factory
- SQL Server Management Studio, Azure Cosmos DB, SQL Developer



- Mentioning the expected master's degree and relevant coursework.

what are strengths and weakness in my resume

Strengths and Weaknesses in Your Resume

Strengths:

- **Relevant skills and experience:** Your resume should highlight skills and experiences that align with the job you're applying for.
- **Strong action verbs:** Use strong verbs to describe your accomplishments and responsibilities.
- **Quantifiable achievements:** Whenever possible, include quantifiable metrics to demonstrate your impact.
- **Relevant industry knowledge:** If you have industry-specific knowledge or certifications, it can be beneficial.

Weaknesses:

- **Outdated information:** Check for any outdated or irrelevant information, such as previous job titles that no longer match your experience.
- **Lack of variety:** A resume with too much repetition can make it difficult for recruiters to differentiate you from other candidates.
- **Poor grammar and spelling:** Pay attention to grammar, spelling, and punctuation errors.
- **Too much focus on the past:** While relevant experience is important, it's also important to highlight your future potential.
- **Weak formatting:** A poorly formatted resume can make it difficult to scan and may not be visually appealing.

Additional Tips for Improving Your Resume:

- **Tailor your resume to each job application:** Highlight skills and experience that are relevant to the specific position.
- **Use keywords:** Include keywords from the job description throughout your resume.
- **Quantify your accomplishments:** Use numbers and metrics to demonstrate your impact.
- **Proofread carefully:** Make sure there are no errors in grammar, spelling, or punctuation.
- **Use a professional font and layout:** Choose a font that is easy to read and a layout that is visually appealing.

Remember, a resume is a marketing tool that should highlight your skills and experience. By addressing your strengths and weaknesses and using effective strategies, you can create a resume that will help you stand out from other candidates.

Ask me anything about your career, job search, or LinkedIn optimization!

Thank you!