**TASK 1**

**Example Queries and Answers:**

Ques: What skills are mentioned in the PDF?

Ans:

Answer: The skills mentioned in the PDF are:

- Object-Oriented Programming (OOPs)

- Problem Solving

- Critical Thinking

- JWT and JWKS

- Python

- Flask

- Web Scraping

- Data Analysis

- Git

- RESTful APIs

- NoSQL Databases

- Microservice Architecture

- Generative AI

- Large Language Models (LLM)

Ques: Who is the resume about?

Ans:

Answer: The resume is about Khushi Sanghrajka.

Ques: What projects has Khushi worked on?

Ans:

Answer: Based on the information provided, Khushi Sanghrajka has worked on various projects utilizing skills in Python, Generative AI, and more. She specializes in building AI-powered applications and scalable microservices, leveraging technologies such as LLMs, LangChain, and OpenAI APIs. She has experience in developing RAG pipelines and integrating third-party services, focusing on performance, security, and scalability across multiple projects. Her role as a Junior Software Developer at Kevit Technologies highlights her practical experience in these areas.

**Explanation of the pipeline**

1. Install all the necessary dependencies. Here I have used Azure AI Inference, Langchain, Chromadb, pypdf and other such libraries.

2. Make the necessary imports.

3. In the code, I have used the models hosted and provided by Github. So, in order to use them, a Github token is necessary along with the base url. This is also configured in the code.

4. The next step was to define the necessary functions for the pipeline steps which are:

-loading the pdf

-chunking the pdf

-initializing Chroma client to store embeddings

-create a chroma collection

-create embeddings using Embeddings client through Azure AI inference package (using text embedding large openai model)

-finally querying the database based on the question and generating output.

5. I implemented the load\_pdf\_content() function using the PDFReader from PyPDF. For testing, I used my resume stored on Google Drive as the input PDF.

6. After that, I use the RecursiveCharacterTextSplitter to make chunks from the content of the pdf.

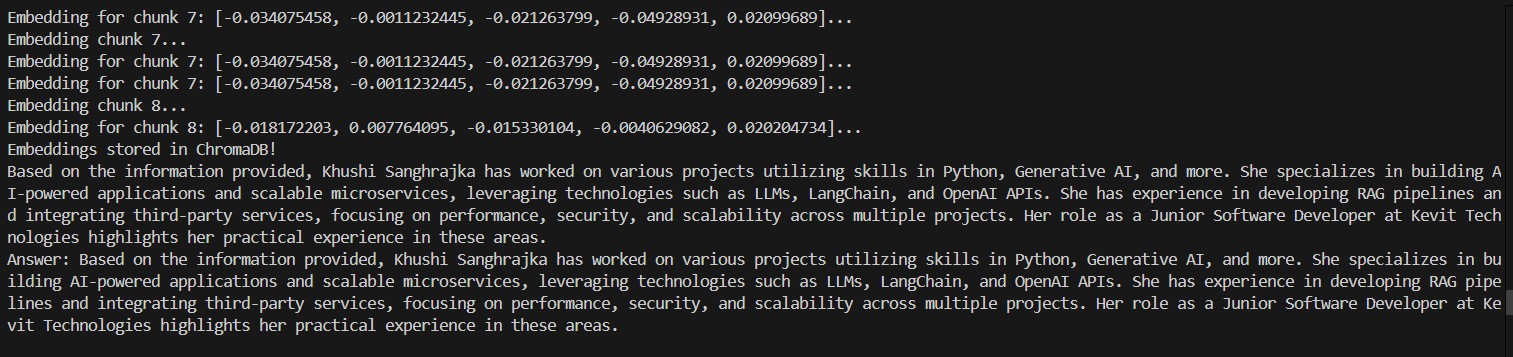
7. Next I initialize a ChromaDB client and an EmbeddingsClient using AzureKeyCredential.

8. I used the text-embedding-3-large model from Azure AI to create embeddings for each content chunk.

Additionally, I generated an embedding for the user query to efficiently match it with relevant content from the ChromaDB.

9. I used the OpenAI API with the GPT-4o model to generate a coherent response based on the retrieved information.

I have used various methodologies to demonstrate that I am adept at using any new techniques and can readily take up and face challenges.

Screenshots: 

**TASK 2**

**Examples:**

Ques: What is mentioned in the PDF?

Ans:

Answer: The document provides information about:

* Skills:
  + Object-Oriented Programming (OOPs)
  + Problem Solving
  + Critical Thinking
  + JWT and JWKS
* Awards:
  + Rising Talent of the Year at Kevit Technologies
  + Recognition for being in the Top 10% in an Internship Drive at Innomatics Research Labs
  + Gold Medalist in an English Olympiad
* Certifications:
  + Several IBM certifications in Python for Data Science, Data Science 101, Data Visualizations, and Machine Learning
  + Prompt Design in Vertex AI by Google
* Experience:
  + Position as a Junior Software Developer at Kevit Technologies
  + Achievements include:
    - Building a chat API using ChromaDB for efficient query responses
    - Implementing a RAG pipeline with MongoDB for bot training
    - Developing Azure authentication middleware
    - Integrating third-party services for SEO and plagiarism checks
    - Creating algorithms for personalized content for upselling
    - Optimizing OpenAI token usage to reduce costs while maintaining performance.

Ques: What is Khushi's experience?

Ans:

**Answer:** Khushi Sanghrajka is a Junior Software Developer with a focus on Python and Generative AI. She has a proven track record in building AI-powered applications and scalable microservices. Her expertise includes utilizing LLMs, LangChain, and OpenAI APIs to enhance efficiency and performance. Khushi is skilled in developing Retrieval-Augmented Generation (RAG) pipelines and integrating third-party services.

**Education:**

* B.E. in Computer Engineering from V.V.P. Engineering College, Gujarat Technological University, with a CGPA of 8.76/10 (2020-2024).

**Skills:**

* Python Programming
* Generative AI
* Large Language Models (LLMs)
* LangChain
* Vector Database
* NoSQL (MongoDB)
* REST APIs
* Git and GitHub
* Data Science & Analytics
* Object-Oriented Programming (OOP)
* Problem Solving and Critical Thinking
* JWT and JWKS

**Awards:**

* Rising Talent of the Year at Kevit Technologies
* Recognition for being in the Top 10% in an Internship Drive at Innomatics Research Labs
* Gold Medalist in the English Olympiad

**Certifications:**

* Python for Data Science (IBM)
* Data Science 101 (IBM)
* Data Visualizations with Python (IBM)
* Machine Learning with Python (IBM)
* Prompt Design in Vertex AI (Google)

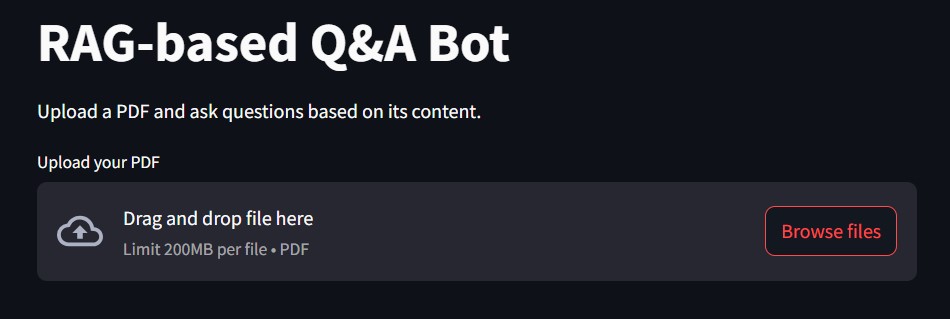
**Experience:**

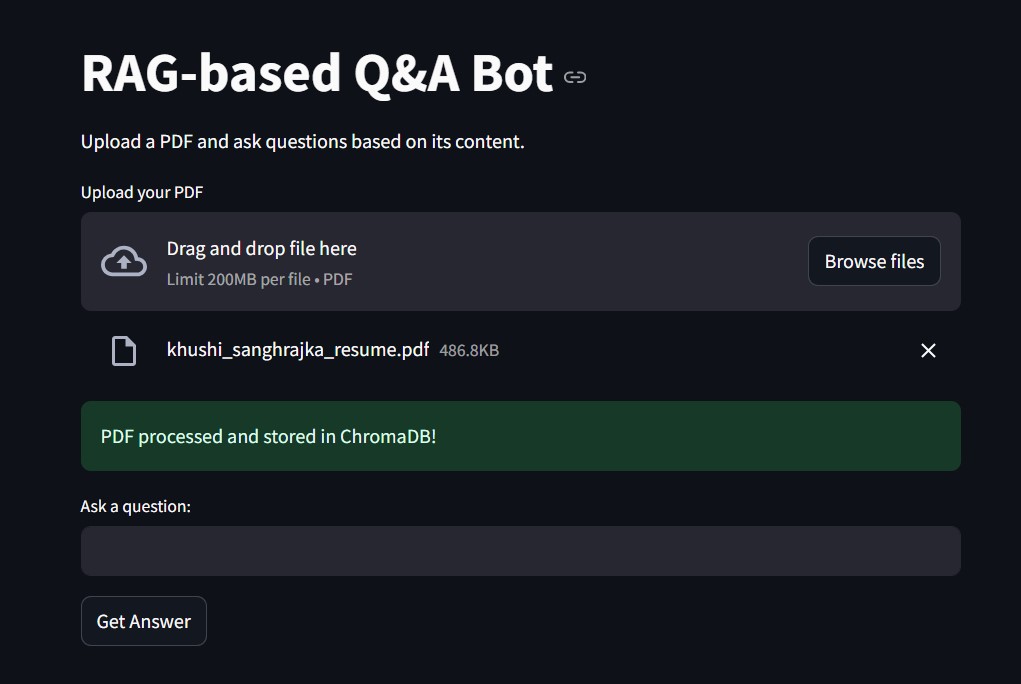
* Junior Software Developer at Kevit Technologies

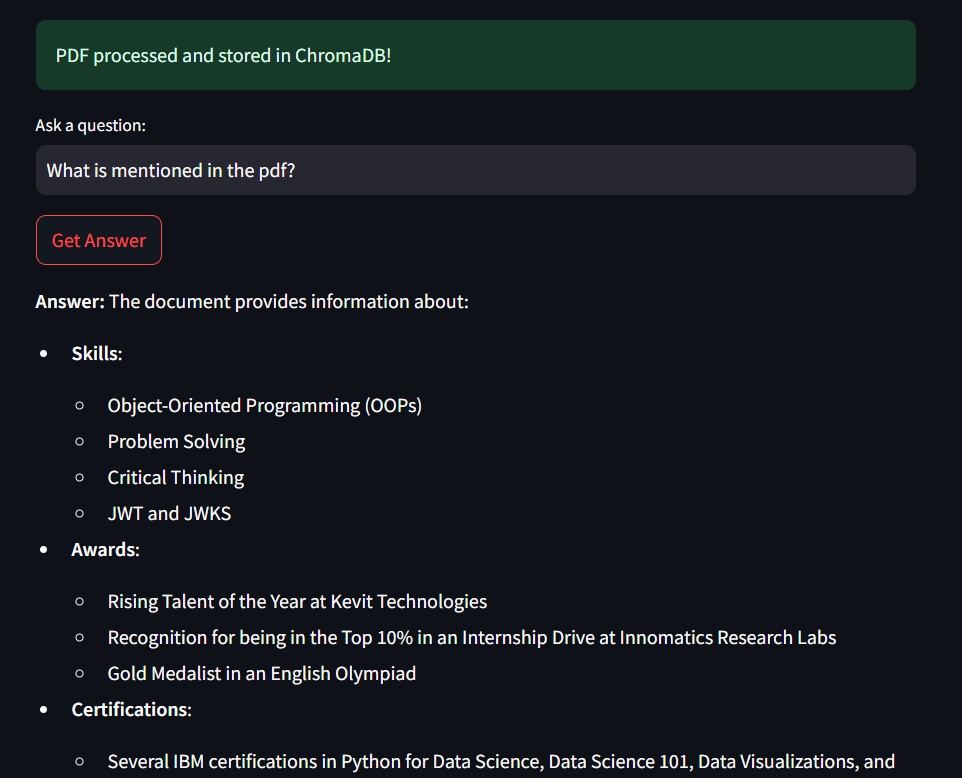
Her experience showcases a strong foundation in software development, particularly in Python and AI technologies.

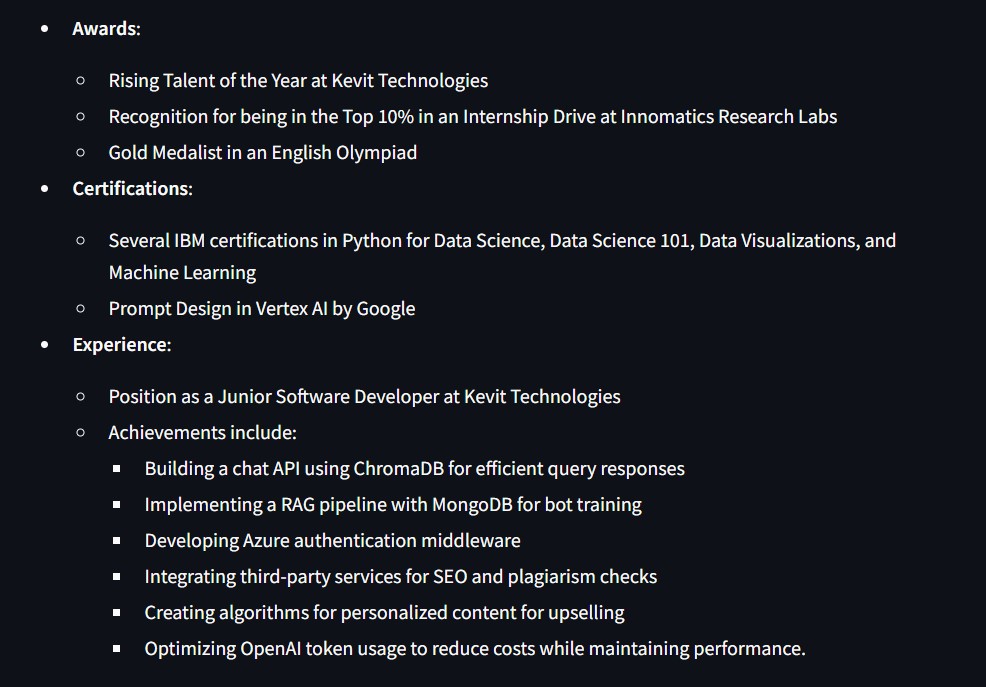
**Explanation of the pipeline**

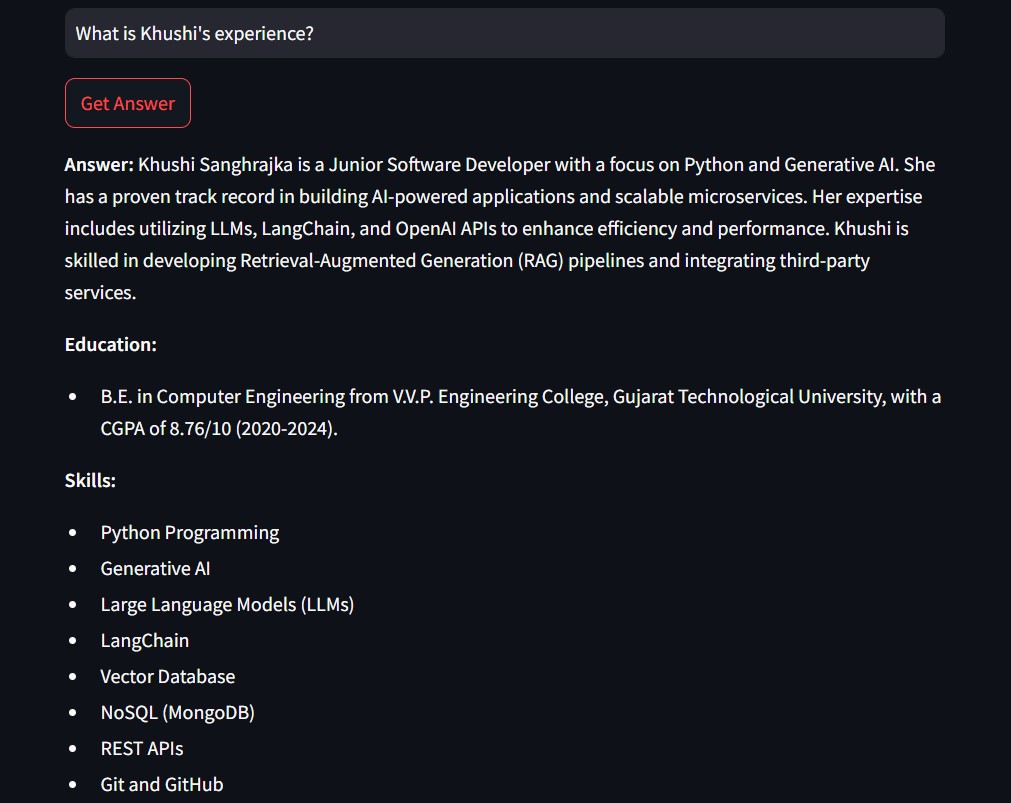
1. I installed all the necessary libraries, including Azure AI Inference, LangChain, ChromaDB, PyPDF, streamlit and other relevant packages essential for building the RAG (Retrieval-Augmented Generation) pipeline.
2. Imported all required modules and packages. Since I utilized models hosted on GitHub, the code required a GitHub token along with the base URL for authentication.
3. I used the same backend as in the first task but integrated it with streamlit for making the frontend UI part.
4. When user runs the streamlit app, he will be shown a screen where the PDF can be uploaded and questions can be asked directly from there.
5. Some screenshots are attached in the section below.









  
Due to API key security and limited use of the APIs available, I am not able to deploy the tasks publicly. To ensure that you may know about the working and functionalities, I have attached screenshots for both the tasks in the notebook as well as in this document. Also the Rate Limits play a significant role, so I can’t deploy them publicly.