Wasserstoff AI Internship Project - Resume Q&A Chatbot

Date: May 21, 2025

# Introduction

This project was completed as part of the Generative AI Internship task for Wasserstoff. The objective was to build a document-based chatbot that can analyze, understand, and respond to user queries using natural language processing techniques. The chatbot is designed to extract information from documents such as resumes and provide clear, cited responses.

# Problem Statement

The challenge was to develop an AI-powered chatbot capable of:  
- Accepting a large number of PDFs or scanned documents  
- Performing OCR for scanned content  
- Extracting and cleaning text from documents  
- Storing the content in a vector database for reuse  
- Handling user queries and returning precise, cited responses  
- Identifying common themes across documents

# Project Overview

To address the challenge, I built a local, offline chatbot system using Python and open-source tools. The system is designed to run entirely on a local machine, without the need for API keys or cloud-based services, which makes it secure and cost-effective.

# Approach

1. Uploaded a resume PDF file into the Jupyter Notebook environment.  
2. Extracted text from the PDF using PyMuPDF (fitz), with fallback support for OCR in case of scanned documents.  
3. Cleaned and split the text into manageable chunks to maintain contextual integrity.  
4. Used HuggingFace’s 'all-MiniLM-L6-v2' model to generate vector embeddings for each chunk.  
5. Stored the vectors in ChromaDB, a lightweight local vector database.  
6. Implemented a retrieval system to fetch top relevant chunks based on user questions.  
7. Used a local question-answering model (deepset/roberta-base-squad2) to generate answers from the retrieved content.  
8. Returned the answer along with confidence scores and citations pointing to the specific document chunks.

# Why This Approach

- It is fully local and does not send data to external servers.  
- It does not require any paid APIs or authentication keys.  
- It works efficiently even on machines without a GPU.  
- It fulfills all the functional requirements laid out in the internship task.

# Technologies Used

- Python 3.x  
- Jupyter Notebook  
- PyMuPDF for text extraction  
- sentence-transformers (HuggingFace)  
- ChromaDB for vector storage  
- Transformers (HuggingFace) for question-answering  
- LangChain for document management and interfacing

# Example Output

Question: What are Kaameshwar's key technical skills?  
Answer: Python, SQL, Tableau, Pandas, NumPy, Scikit-Learn, Matplotlib  
Citations: Resume\_Chunk\_2, Resume\_Chunk\_3  
Confidence: 93.27%

# Conclusion

This project demonstrates a scalable and effective way to build a document-based chatbot using open-source tools. It lays the foundation for future systems that can handle larger datasets, multi-document analysis, and deeper thematic insights. The solution is aligned with the goals of the internship and offers a reliable local alternative to cloud-based AI assistants.