Simon Tesfatsion

SFBU RAG CHAT BOT

Data from

Text Pdf YouTube Website

GItHUB LInk

Google Slides Link

OVERVIEW

Introduction

- This project demonstrates a Retrieval-Augmented Generation (RAG) chatbot using LangChain and OpenAI, designed to answer questions based on uploaded documents.

- Functionality:
  - Allows users to upload various document types (PDF, text, web pages, etc.)
  - Answers user questions using only the provided document context.
  - Displays conversations in a chat-like interface for a seamless experience.

Initialize Environment

- Environment Setup:
  - Loaded environment variables using dotenv for secure storage of API keys.
  - Utilized Streamlit secrets for OpenAI API key access.
- Library Imports:
  - Core Libraries: LangChain modules, OpenAI API, Streamlit, Chroma.
  - LangChain Components: ChatOpenAI, ChatPromptTemplate, StrOutputParser.
  - Custom RAG Components: Vector Store and document loaders.
- OpenAI Client Initialization:
  - Created an OpenAI client for text completions.
  - Set up the language model (gpt-3.5-turbo) as the default model in session_state for conversational consistency.

Data Loading and Vector Store Creation

- Supported Document Types:
  - PDF, Text, Web, Wikipedia, YouTube transcriptions.
- Data Loading Process:
  - Utilized LangChain’s document loaders to ingest different document types.
  - Split each document into manageable chunks using RecursiveCharacterTextSplitter for efficient vector storage.
- Creating Vector Embeddings:
  - Used OpenAI embeddings (text-embedding-3-large) for creating document embeddings.
- Vector Store (Chroma):
  - Stored embeddings in a Chroma database, enabling quick retrieval of contextually relevant chunks during user queries.

Retrieval

- Retrieval Function:
  - Implemented a retriever using LangChain's vector_store.as_retriever() to retrieve top k (e.g., 3) relevant document chunks based on user questions.
- How It Works:
  - The retriever matches user input with the most relevant chunks in the vector store, serving as the context for the chatbot.
- Why Retrieval Matters:
  - Ensures that the chatbot provides accurate, document-based answers rather than generic responses.
  - Maximizes relevance by narrowing down information from extensive documents.

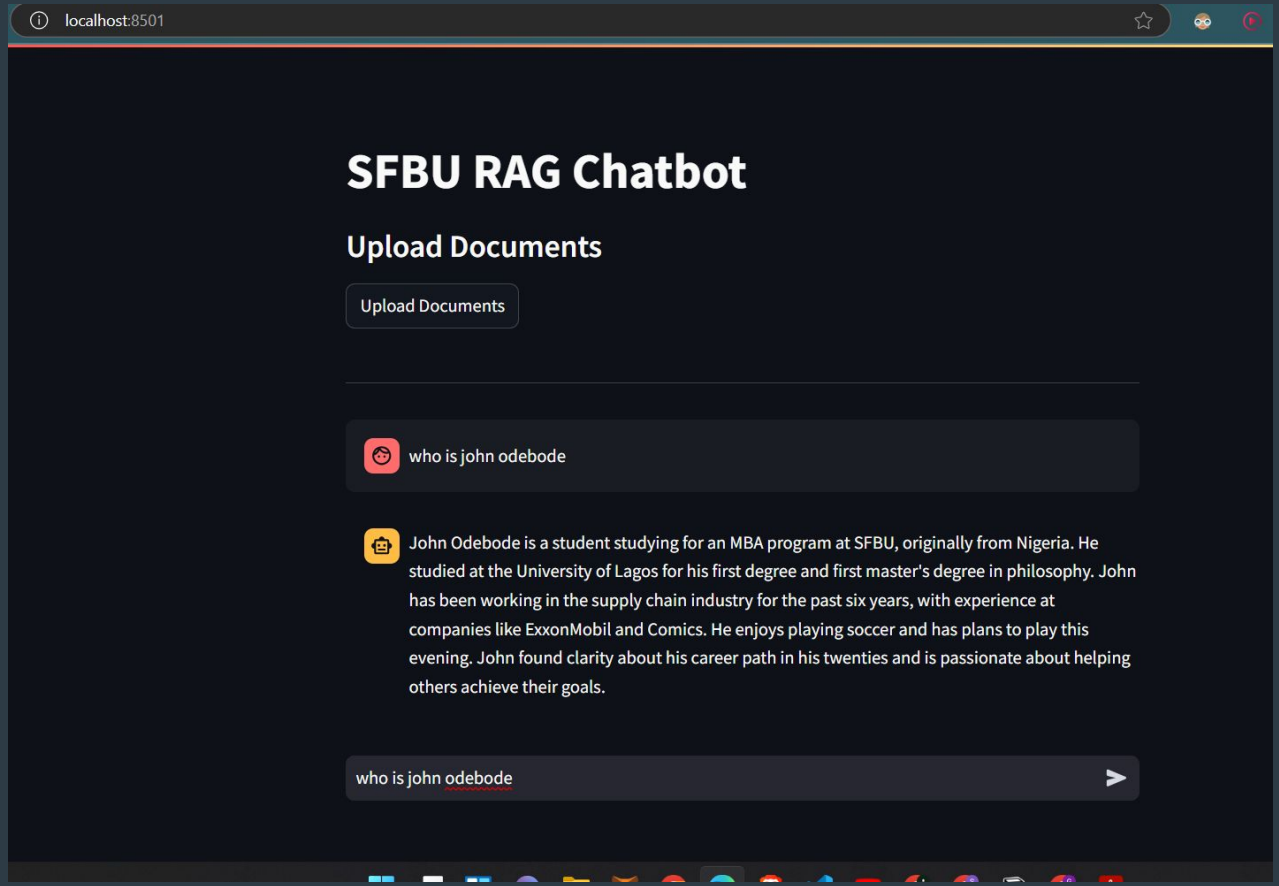
Chains

- System and User Prompts:
  - Defined a system_prompt for contextual guidance, instructing the assistant to answer based only on the retrieved document chunks.
  - Set up a user_prompt to dynamically integrate user questions.
- Chain Composition:
  - Utilized ChatPromptTemplate to format prompts.
  - Constructed the retrieval chain using RunnableMap to link the retriever and prompt together.
- Purpose of Chains:
  - Chains streamline the flow of information: retrieving relevant context, generating a response, and formatting the output.
- Workflow:
  - The retrieval chain collects context, and then the model uses this context to answer user questions, simulating a natural conversational response.

Output Parser

- Output Parsing Setup:
  - Implemented StrOutputParser to parse responses from the language model.
- Role of the Parser:
  - Ensures output is clean, concise, and formatted for display in the chat interface.
- Integration with Chain:
  - Connected the output parser in the final chain step to handle the raw response from the language model, making it suitable for chat display.
- Benefit:
  - Improves response readability and maintains consistency in user experience.

Demo:


localhost:8501
SFBU RAG Chatbot
Upload Documents
Upload Documents
who is john odebode
John Odebode is a student studying for an MBA program at SFBU, originally from Nigeria. He studied at the University of Lagos for his first degree and first master's degree in philosophy. John has been working in the supply chain industry for the past six years, with experience at companies like ExxonMobil and Comics. He enjoys playing soccer and has plans to play this evening. John found clarity about his career path in his twenties and is passionate about helping others achieve their goals.
who is john odebode

Project Reference Materials

GitHub Link :

https://github.com/Montegan/SFBU-RAG-APP?tab=readme-ov-file

Google Slides Link :

https://docs.google.com/presentation/d/1D3gPBKbaR9--UqdRLO7JgVn-6QEWncEqOkCwaKzAodM/edit?usp=sharing

Thank You

Simon Tesfatsion