GDG TechSprint

Team: Runtime Terror

Team

Name	Registration No.	Email ID	Mobile Number	Github Username
Shreyansh Kushwaha	20255084	Shreyansh.20255084@mnnit.ac.in	+91 9415972322	Shreyansh-kushw
Aayush Thakur	20253534	Aayush.20253534@mnnit.ac.in	+977 9807014913	Aayush20253534
Prateek Rastogi	20253202	Prateek.20253202@mnnit.ac.in	+91 6393363659	Prateek816
Aayansh Niranjan	20253004	aayansh.20253004@mnnit.ac.in	+91 8090895438	aayansh207

Project Name and Problem statement: ContextIQ

Selected Idea : An AI-powered RAG (Retrieval-Augmented Generation) Knowledge Engine for automated document analysis and semantic retrieval.

Problem Statement:

In many organizations and academic environments, large volumes of documents such as PDFs, reports, research papers, manuals, and notes are generated and stored over time. As the number of documents increases, it becomes difficult for users to efficiently search, understand, and extract relevant information from them. Traditional keyword-based search systems fail to capture the semantic meaning of queries, leading to inaccurate or incomplete results. Moreover, users often need quick summaries and precise answers from documents instead of reading entire files.

There is a need for an intelligent system that can **accept and manage multiple document uploads**, **store them securely**, and **process their contents intelligently**. The system should be capable of **summarizing documents**, performing **semantic search across uploaded documents**, and allowing users to **ask natural language questions** and receive accurate, context-aware answers derived directly from the document content.

This project aims to design and develop a **Retrieval-Augmented Generation (RAG) based document intelligence system** that enables document ingestion (uploading/loading), semantic indexing, summarization, and AI-powered querying. By combining vector-based retrieval with large language models, the system will ensure that responses are grounded in the uploaded documents, reducing hallucinations and improving reliability.

Objectives

- To allow users to upload and store multiple documents securely.
- To extract and preprocess text from uploaded documents.
- To generate concise summaries of documents automatically.
- To implement semantic search using vector embeddings.
- To enable AI-powered question answering over the documents using a RAG architecture.
- To ensure responses are context-aware and sourced from relevant documents.

Proposed Solution (Brief)

The system will load uploaded documents, split them into meaningful chunks, generate embeddings, and store them in a vector database. When a user searches or asks a question, relevant document chunks will be retrieved using semantic similarity and passed to a language model to generate accurate summaries or answers.

Tech Stack

Frontend: HTML, CSS, JS, Tailwind CSS
Backend: Python with FastAPI
Database: SQLite, Pinecone
Additonal tools/APIs: Gemini API, easyocr, pymupdf

Google Technologies used:

- **AI/LLM:** Gemini API
- **Auth:** Firebase Authentication (Google Sign-In)
- **Analytics:** Firebase Analytics
- **UI Resources:** Google Fonts & Material Symbols

Planned Functionalities (Upcoming Work)

- **Data Ingestion:** Extend support for large-scale document uploads with intelligent chunking and preprocessing for efficient downstream processing.
- **RAG Integration:** Implement a Retrieval-Augmented Generation pipeline to fetch relevant document contexts before response generation.
- **Semantic Search:** Enable meaning-based document and content search using vector embeddings instead of keyword matching.
- **AI Querying:** Allow users to ask natural language questions over uploaded documents and receive context-aware, grounded answers.
- **Source Attribution:** Provide references to the exact document sections used for generating responses to improve transparency and reliability.

Repository Link: https://github.com/Aayansh207/GDG-25