INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

www.ijraset.com

Call: ⓒ08813907089 | E-mail ID: ijraset@gmail.com

AI-Powered Document Processing and Chat System

Jay Modi[1], Anuj Kalandikar[2], Tejas Aher[3], Varad Mule[4], Prof. Suresh Reddy[5]
SRTTC- Computer Department, Savitribai Phule Pune University

Abstract: This paper presents an AI-driven platform for interactive PDF document analysis and collaboration. Leveraging LangChain, Supabase, and Vercel AI SDK, the system transforms static documents into searchable knowledge bases using vector embeddings and Retrieval-Augmented Generation (RAG). By bridging generative AI, vector databases, and end-user applications, this project demonstrates a scalable framework for transforming static documents into interactive knowledge bases, enhancing productivity in education, research, and enterprise workflows through seamless AI-human interaction.
Keywords: Artificial Intelligence, Document Processing, Vector Embeddings, Retrieval-Augmented Generation (RAG), Collaborative Chat

I. INTRODUCTION

Traditional PDF analysis relies on manual navigation, limiting efficiency in extracting insights. This paper introduces a platform that automates document interaction using AI workflows. By integrating LangChain for text processing, Supabase for vector storage, and Vercel AI SDK for real-time chat, the system enables dynamic querying, contextual discussions, and knowledge sharing. The architecture addresses scalability, security, and usability challenges, bridging generative AI with end-user applications.

II. SYSTEM ARCHITECTURE

A. Workflow Design

1) *Document Upload & Storage*: Users upload PDF documents through the web interface. The system securely stores these files in Vercel Blob Storage. Uploaded documents trigger an AI workflow to begin processing. A preview of the document is generated for display in the chat interface.

2) *AI-Powered Document Analysis*: The uploaded document is processed using Langchain, which extracts text and metadata. Vector embeddings of the document content are generated and stored in Supabase Vector Store. These embeddings enable efficient semantic search and contextual analysis. The AI model processes and structures the extracted data for chat-based interactions.

3) *Chat Interface for User Interaction*: Users can engage in discussions about the document via a chat interface. The AI model, possibly powered by OpenAI, provides insights, summaries, and answers to queries. Users can navigate through document excerpts and receive AI-generated explanations. The interface supports forking conversations to explore different discussion paths.

4) *Document Preview & Interaction*: A real-time preview of the uploaded document is displayed in the chat interface. Users can highlight sections of the document to trigger focused AI analysis. The chat system references document embeddings for more context-aware responses. This enhances user engagement and simplifies document comprehension.

5) *Export & Fork Chat Features:* Users can export chat history, including AI insights and document excerpts, into a detailed PDF. The system organizes conversations into a structured report for offline access. The **fork chat** feature allows users to split discussions into separate threads for deeper exploration. This enables collaborative document review and knowledge sharing.

B. Key Components

Frontend: The frontend of the system is built using Next.js, a powerful React-based framework that ensures a seamless and interactive user experience. The chat interface allows users to engage in AI-powered discussions, query document contents, and preview uploaded PDFs. With Vercel AI SDK, the frontend efficiently handles real-time interactions, ensuring smooth communication between users and the AI model. The document preview feature enhances usability, allowing users to navigate through their files while interacting with the chat system. Additionally, the fork chat functionality enables users to branch off discussions, improving document exploration.

Backend: The backend is powered by Langchain, which processes the uploaded documents and extracts meaningful information.

This framework enables natural language understanding (NLU) and context-aware AI responses, making document analysis more efficient. The backend integrates with OpenAI to generate intelligent insights, summaries, and responses based on user queries. It also manages the flow of data between the frontend and the storage systems, ensuring seamless interactions. Additionally, serverless architecture on Vercel enhances backend performance by automatically scaling resources based on demand, reducing infrastructure management complexity.

Storage: For document storage, the system uses Vercel Blob Storage, a scalable and secure solution for managing uploaded PDF files. This ensures fast access and retrieval while maintaining data integrity. To support AI-powered search and retrieval, Supabase Vector Store is used to store vector embeddings of documents, enabling efficient semantic search and contextual analysis. This storage approach allows the system to handle large volumes of data while ensuring quick and accurate retrieval of relevant document excerpts. The combination of blob storage for files and vector storage for AI-driven search provides a robust, scalable, and cost-effective solution for document processing.
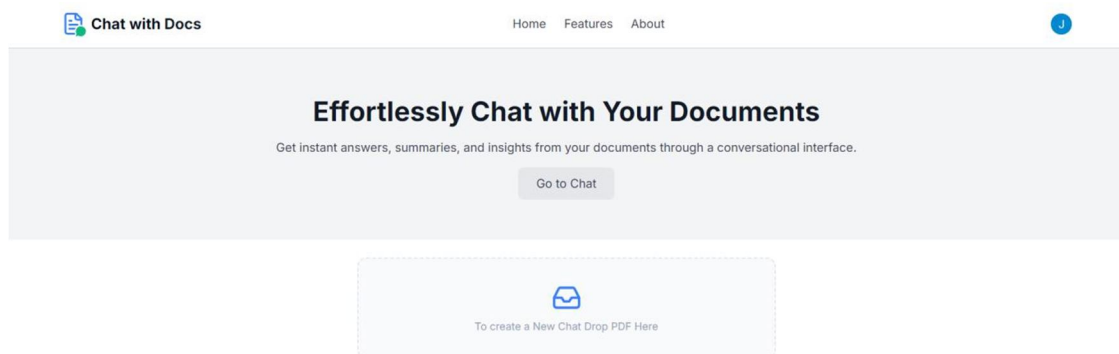

Fig. 1   Frontend Page

C.   Storage and Scalability

Efficient document storage and scalable infrastructure are critical to handling large volumes of uploaded documents and AI-powered interactions. This system leverages Vercel Blob Storage for document management and Supabase Vector Store for optimized AI processing. The combination of these technologies ensures seamless document retrieval, enhanced search capabilities, and a scalable architecture suitable for high-demand applications.

D.   PDF Upload and Processing

The system provides a seamless and user-friendly interface for PDF document uploads, allowing users to easily upload files for processing. Once uploaded, the documents are securely stored in Vercel Blob Storage, ensuring data integrity, fast retrieval, and scalable storage management. To enhance document processing, Langchain is utilized to extract key insights, including text and metadata, enabling efficient information retrieval. Additionally, the system generates vector embeddings of the extracted content, which are stored in Supabase Vector Store, facilitating advanced semantic search and AI-powered interactions, as shown in Fig. 2.

Fig. 2 Example of an Uploading PDF's

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

E. Interacting with PDF via AI-Powered Chat

By integrating Langchain and OpenAI, the chat system provides context-aware responses, ensuring that users receive precise answers based on document content. This is further enhanced by semantic search capabilities powered by Supabase Vector Store, enabling the retrieval of relevant sections based on user queries. Users can request AI-generated summaries, explanations, and key highlights, making document analysis more intuitive and efficient. Additionally, the system offers a fork chat feature, allowing multiple conversation threads to be created for different aspects of document exploration, as shown in Fig. 3.
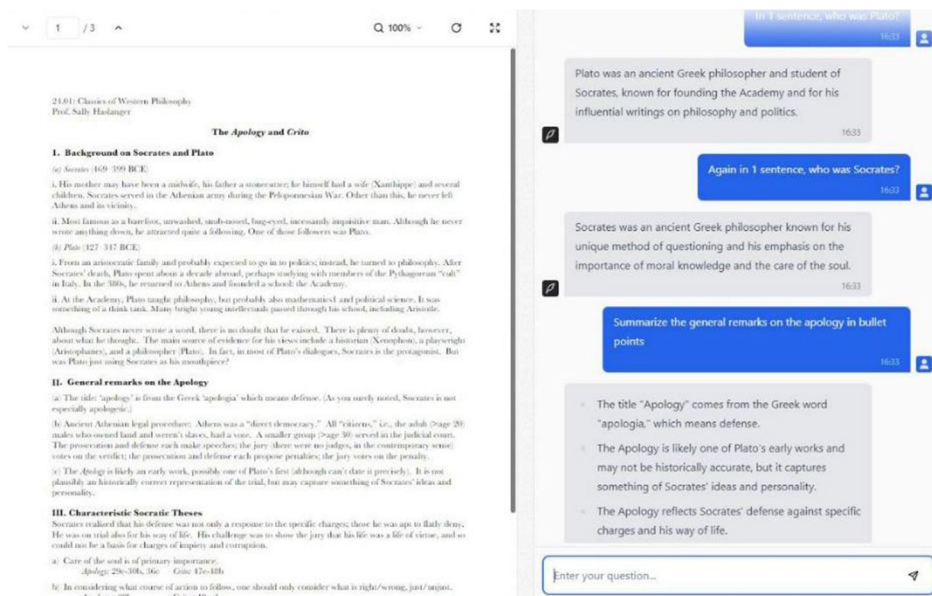

Fig. 3 Example of interacting with Pdf

TABLE I
System Archure Overview With Key Components And Technologies

COMPONENT	KEY TECHNOLOGIES/PROCESSES	DESCRIPTION
DOCUMENT UPLOAD & STORAGE	Next.js, Vercel Blob Storage	Handles secure PDF uploads via Vercel Blob Storage, triggers backend processing via Next.js API routes.
AI WORKFLOW	Langchain, OpenAi Embeddings, Supabase	Extracts text, splits chunk, generates embeddings via OpenAi, and stores vectors in Supabase.
CHAT INTERFACE	Vercel AI SDK, OpenAi, Supabase Vector Store	Enables contextual Q&A using retrieved document chunks and OpenAi for response generation.
DOCUMENT PREVIEW	React PDF Viewer	Displays uploaded PDFs in the chat interface, linked to metadata from vector embeddings.
EXPORT CHAT AS PDF	PDF-Lib, React-PDF	Generates structured PDFs with chat history, AI analysis, and document excerpts for offline access.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

FORK CHAT	Superbase Database	Duplicates chat sessions using parent-child relationships in Supabase tables to enable branching conversations.
TECH STACK CONFIGURATION	Supabase (pgvector), Vercel Blob, OpenAi API	Configures storage, embeddings, and API integrations with environment variables and access controls.
SCALABILITY & SECURITY	Vercel Serverless Functions, Redis (optional), RLS	Implements async processing and row-level security for performance and data protection.
DEPLOYMENT	Vercel Hosting, Serverless Architecture	Deploys frontend and backend with optimized resource allocation and monitoring for reliability.

III. CONCLUSIONS

This paper introduces an AI-powered document processing system that enhances document interaction through intelligent analysis and chat-based discussions. By integrating Langchain, Next.js, and Vercel AI SDK, the system enables efficient document handling, retrieval, and user engagement. Future enhancements may include multi-document comparison, advanced summarization techniques, and real-time collaboration features.

ACKNOWLEDGMENT

REFERENCES

[1] Langchain Documentation, Available: https://python.langchain.com/
[2] Vercel AI SDK, Available: https://vercel.com/
[3] Next.js Framework, Available: https://nextjs.org/
[4] Supabase Documentation, Available: https://supabase.com/docs/
[5] OpenAi AI Overview, Available: https://ai.google.com/OpenAi

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)