CONVERT ENTERPRISE PDFS INTO SEARCHABLE KNOWLEDGE

Ismail Sk	11601022070
Supratim Nag	11601022064
Sanchari Biswas	11601022049

Introduction

In modern enterprises, critical information is distributed across thousands of PDF documents, including technical manuals, policy files, research papers, compliance documents, audit reports, and scanned letters. Although these documents hold valuable institutional knowledge, most of them remain unstructured, inconsistently formatted, and difficult to search. Traditional keyword-based search systems often fail to retrieve accurate information because they cannot understand document context, layout structures, tables, figures, or scanned text. As a result, employees spend considerable time manually searching through large PDF repositories, leading to reduced productivity and inefficient knowledge management.

With the rapid advancement of Generative AI and Retrieval-Augmented Generation (RAG) systems, organizations now have the opportunity to transition from basic keyword search to semantic, meaning-based document retrieval. This project addresses the need for an intelligent system that can transform unstructured and scanned PDFs into structured, searchable knowledge. The objective is to design a robust pipeline capable of accurately extracting text, tables, images, and charts from PDF documents and organizing them in a format optimized for efficient retrieval.

By integrating OCR for scanned content, meaningful text chunking, table extraction into a NoSQL database, and vector-based semantic search for contextual accuracy, the proposed system aims to significantly reduce information lookup time and enhance enterprise-wide knowledge accessibility. Ultimately, the project seeks to enable instant information discovery, remove bottlenecks associated with manual document search, and establish a strong foundation for future AI-powered enterprise knowledge assistants.

Abstract

Enterprise organizations store vast volumes of information in PDF formats such as manuals, policies, reports, technical documentation, and scanned records. However, retrieving relevant information from these unstructured documents is time-consuming and inefficient due to the lack of semantic search capabilities. This project aims to develop a system that converts unstructured and scanned PDF documents into structured and searchable knowledge.

The proposed solution integrates Optical Character Recognition (OCR), intelligent document segmentation, table extraction, and vector-based semantic search. Text is divided into meaningful sections without disrupting chapter boundaries, tables are extracted and stored in a NoSQL database for precise queries, and the processed text is embedded into a vector database to enable context-aware information retrieval. Additionally, images and charts are translated into textual descriptions to make multimedia content searchable.

The project is being developed as part of the **Intel® Unnati Industrial Training Program**, and is currently in progress. Upon completion, the system is expected to deliver an end-to-end pipeline that transforms enterprise PDFs into a searchable knowledge base along with a search interface for efficient retrieval. This approach aims to reduce manual document exploration time, improve organizational knowledge accessibility, and provide a scalable foundation for enterprise-grade information management.

Problems

- Modern enterprises store critical knowledge in thousands of PDFs (manuals, policies, research papers, audits, scanned documents).
- Most PDFs are **unstructured**, inconsistently formatted, and difficult to search.
- Traditional **keyword-based search** fails to understand:
- Context and semantics

I. Layout and structure

I. Tables, figures, charts

II. Scanned or image-based text

- Employees spend significant time manually searching documents → **low productivity** and **inefficient knowledge management**.

Solution

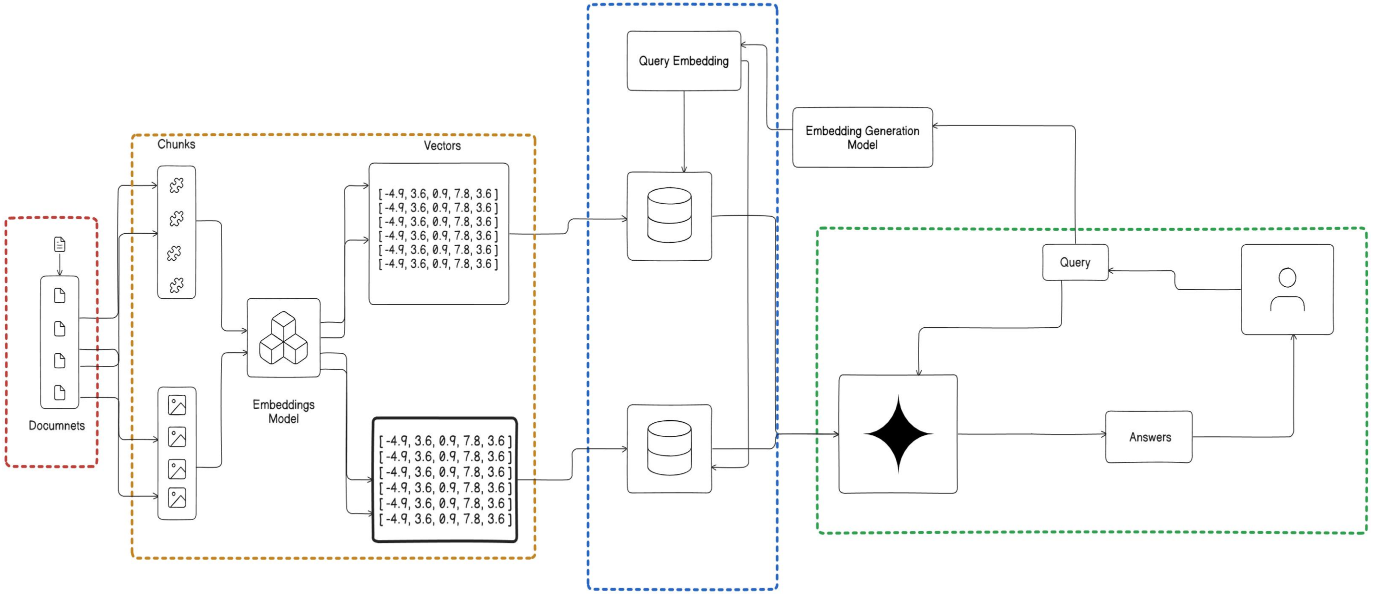
End-to-End Intelligent PDF Knowledge Processing Pipeline

- **OCR Integration** for scanned or image-based PDFs
- **Meaningful text chunking** for better semantic understanding
- **Table extraction** for structured storage and quick access
- **Vector embeddings stored in Pinecone** for fast semantic retrieval
- **RAG-based retrieval** for accurate, context-aware responses

RAG

1. **Document Collection** – Gather your documents or text sources.
2. **Chunking –** Split the documents into smaller pieces (chunks), usually sentences or paragraphs. (One chunk = many tokens.)
3. **Embedding –** Convert each chunk into a vector of numbers that represents its meaning.
4. **Vector Store –** Store all these embeddings in a database (FAISS, Chroma, Pinecone, etc.) for fast similarity search.
5. **Retrieve –** Convert the user query into an embedding, then find the chunk(s) whose embeddings are closest (e.g., using cosine similarity or Euclidean distance).
6. **Combine & Query LLM –** Feed the query along with the retrieved chunks into an LLM, which generates the final answer.

Workflow Diagram



Methodology

A. Initialization
Loads secure configuration variables (API keys, DB credentials) and connects all core components like the vector DB, embedding model, and LLM.

B. Authentication
Handles secure user login with hashed passwords and token-based access control to maintain privacy and auditability.

C. Document Processing
Extracts text/images from PDFs, applies OCR when needed, creates meaningful chunks, generates embeddings, and stores them with metadata in a vector database for semantic search.

D. Semantic Retrieval (RAG)
User queries are embedded and matched with relevant chunks. Retrieved context is sent to the LLM so responses are grounded in the organization's documents.

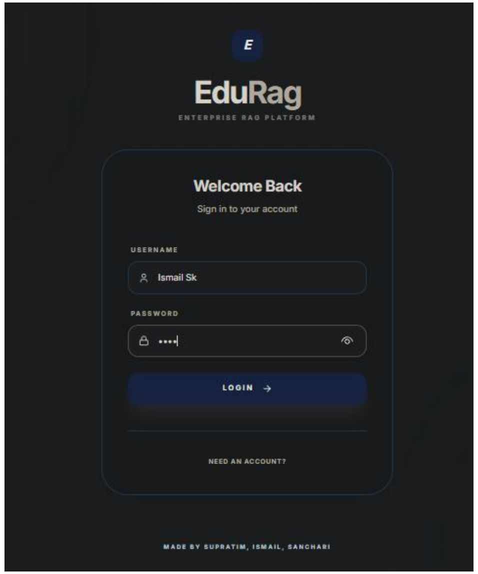
E. Response Delivery
The system returns the final answer along with source references for transparency and traceability.

F. Project Status
Core RAG pipeline is completed. Ongoing work includes UI development, broader dataset testing, OCR improvements, and performance optimization.

Results

The system has been partially implemented and tested, focusing on text extraction, embedding generation, vector storage, and semantic retrieval. The backend RAG workflow works correctly, with accurate context retrieval, grounded LLM responses, and stable API performance. A basic Streamlit interface allows PDF uploads and query testing, successfully displaying answers with relevant context. While the core pipeline functions well, image and table extraction, detailed performance benchmarking, and the final production UI are still under development.

Results


E

EduRag

ENTERPRISE RAG PLATFORM

Welcome Back

Sign in to your account

USERNAME

Ismail Sk

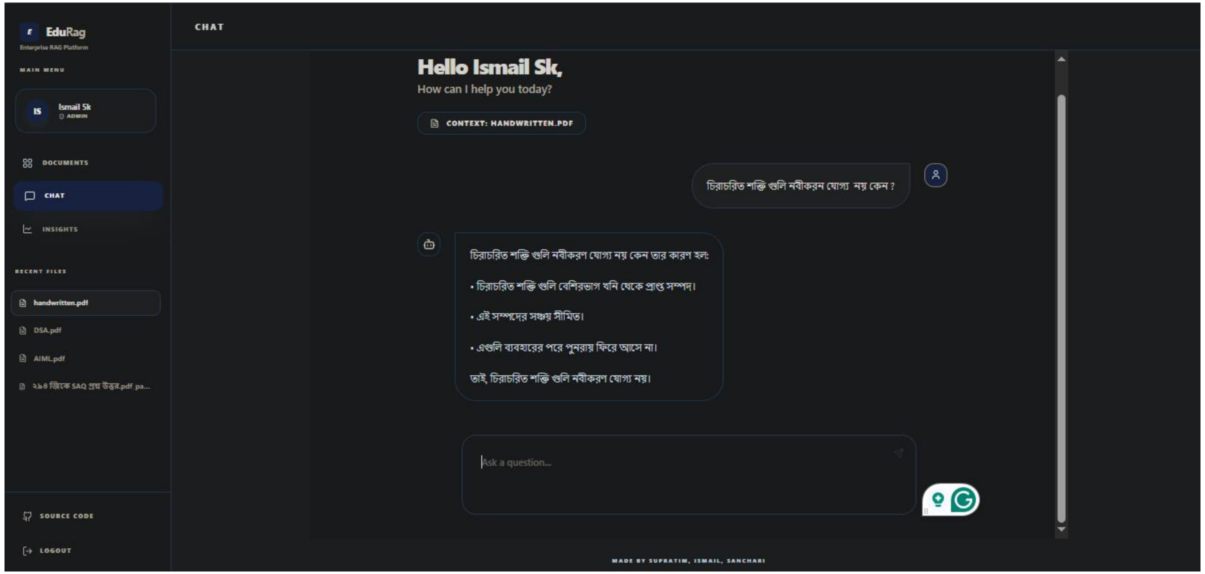
PASSWORD

••••

MADE BY SUPRATIM, ISMAIL, SANCHARI

Website Login

Results



Hello Ismail Sk,

How can I help you today?

চিরাচরিত শক্তি গুলি নবীকরন যোগ্য নয় কেন ?

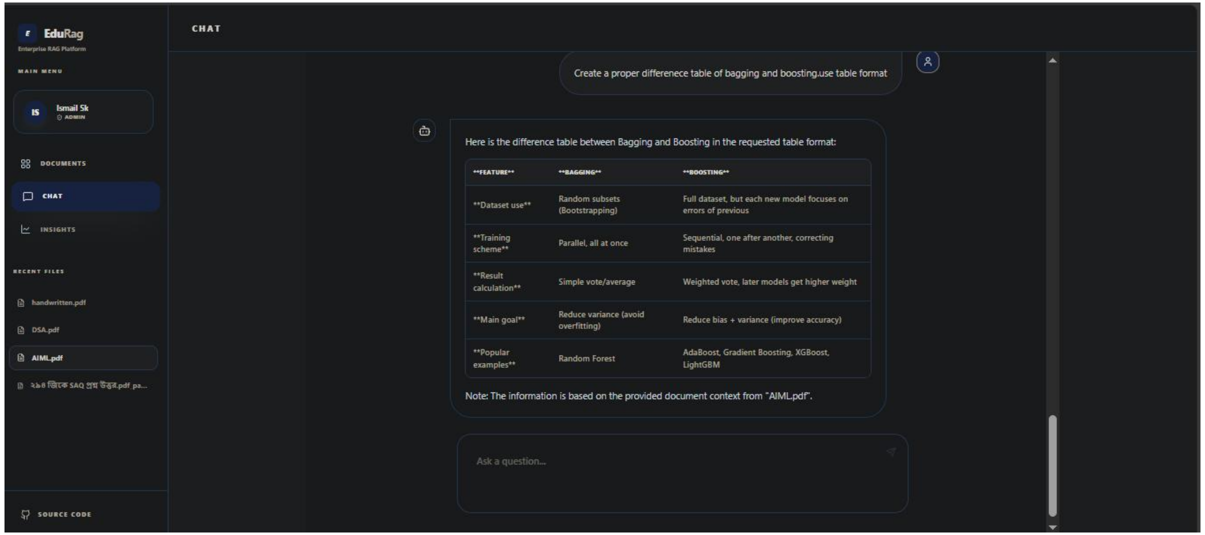
চিরাচরিত শক্তি গুলি নবীকরণ যোগ্য নয় কেন তার কারণ হল:

- চিরাচরিত শক্তি গুলি বেশিরভাগ খনি থেকে প্রাপ্ত সম্পদ।
- এই সম্পদের সঞ্চয় সীমিত।
- এগুলি ব্যবহারের পরে পুনরায় ফিরে আসে না।

তাই, চিরাচরিত শক্তি গুলি নবীকরণ যোগ্য নয়।

Multilingual Chatbot’s Reply

Results



Create a proper differenece table of bagging and boosting.use table format

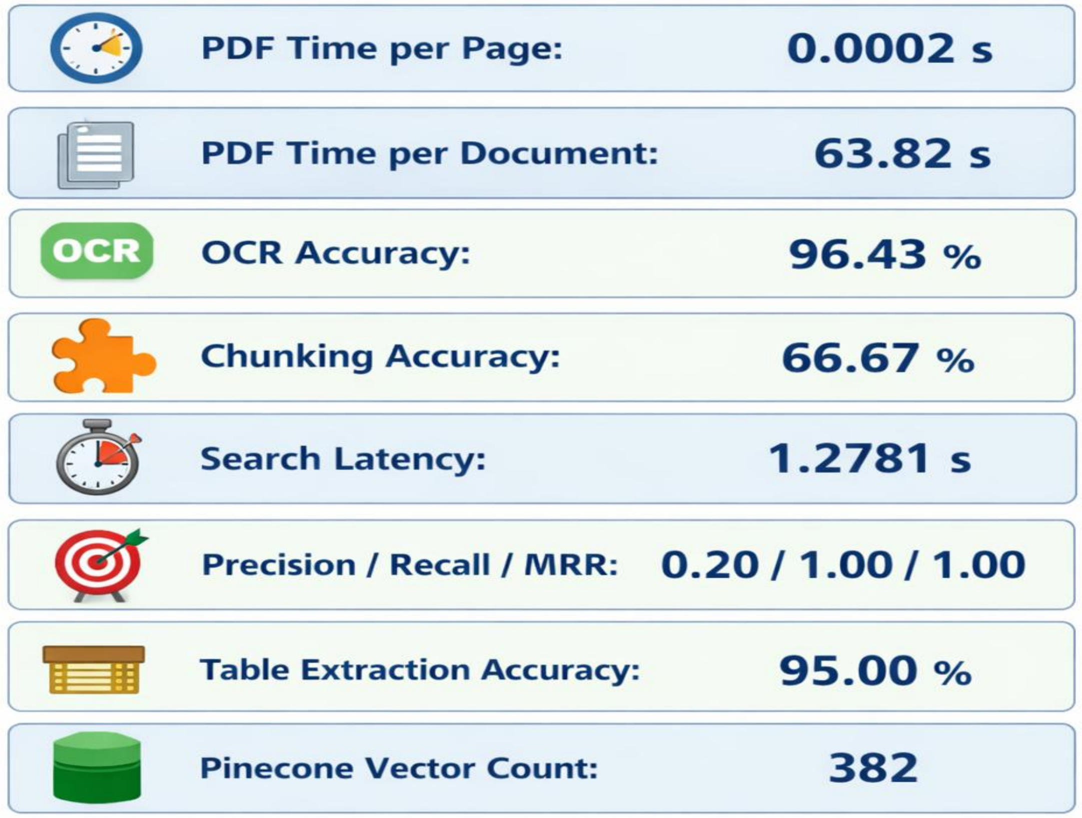
Here is the difference table between Bagging and Boosting in the requested table format:

FEATURE	**BAGGING**	**BOOSTING**
Dataset use	Random subsets (Bootstrapping)	Full dataset, but each new model focuses on errors of previous
Training scheme	Parallel, all at once	Sequential, one after another, correcting mistakes
Result calculation	Simple vote/average	Weighted vote, later models get higher weight
Main goal	Reduce variance (avoid overfitting)	Reduce bias + variance (improve accuracy)
Popular examples	Random Forest	AdaBoost, Gradient Boosting, XGBoost, LightGBM

Note: The information is based on the provided document context from "AIML.pdf".

Structured Tabular Output

Results



PDF Time per Page:	0.0002 s
PDF Time per Document:	63.82 s
OCR Accuracy:	96.43 %
Chunking Accuracy:	66.67 %
Search Latency:	1.2781 s
Precision / Recall / MRR:	0.20 / 1.00 / 1.00
Table Extraction Accuracy:	95.00 %
Pinecone Vector Count:	382

Future Work

- Performance benchmarking modules for relevance accuracy, retrieval latency, scalability, and memory efficiency
- Advanced security features: fine-grained access control, audit logs, API rate limiting, and credential protection
- Containerized deployment using Docker for cloud and on-premise environments
- Collaboration & analytics: usage dashboards, popular queries, and cross-document insights
- Future enhancement: support for handwritten and low-quality scans using image enhancement and handwriting-aware OCR

—Conclusion —

This project shows that enterprise PDFs can be converted into a structured, semantically searchable knowledge base using a RAG pipeline. The system successfully handles text extraction, chunking, embeddings, vector storage, and LLM-based responses, enabling semantic search instead of simple keyword matching. Tests through Postman and a Streamlit prototype confirm accurate retrieval and grounded answers. Although the current version supports only text-based retrieval, the results demonstrate strong potential for improving document accessibility. With future additions like multimodal extraction, multilingual support, and a complete UI, the system can evolve into a fully enterprise-ready knowledge management solution.

REFERENCES

[1] LangChain Documentation. Accessed: Jan. 2025. [Online]. Available: https://python.langchain.com
[2] Pinecone Documentation. Accessed: Jan. 2025. [Online]. Available: https://www.pinecone.io
[3] FastAPI Documentation. Accessed: Jan. 2025. [Online]. Available: https://fastapi.tiangolo.com
[4] Streamlit Documentation. Accessed: Jan. 2025. [Online]. Available: https://streamlit.io
[5] Google Generative AI (Gemini) API Documentation. Accessed: Jan. 2025. [Online]. Available: https://ai.google.dev
[6] Groq API Documentation. Accessed: Jan. 2025. [Online]. Available: https://console.groq.com/docs
[7] PyMuPDF Documentation. Accessed: Jan. 2025. [Online]. Available: https://pymupdf.readthedocs.io
[8] Tesseract OCR Documentation. Accessed: Jan. 2025. [Online]. Available: https://github.com/tesseract-ocr

Thanks!