Financial Document Analyzer Using AI

Arya Dalal[1], Eshika Wandre[2], Aryan Kankaria[3], Sankalp Sharma[4]

[1]–[4] Department of Artificial Intelligence, Vishwakarma University, Pune

---***---

Abstract - Financial reports are typically lengthy, information-heavy, and challenging to interpret without automated assistance. To address this, we created an application that simplifies the review of such documents by processing them automatically. The system accepts PDF files, converts them into smaller text segments, and applies advanced language-model techniques to enable more meaningful search and understanding. These segmented representations are stored in a vector database (ChromaDB) for rapid semantic retrieval, while key extracted values are saved in MongoDB to support structured queries. A Streamlit interface provides an interactive environment where users can converse with the documents, browse stored information, and perform audits assisted by an AI model. The solution reduces the time required to navigate extensive financial files and offers a practical demonstration of how modern AI can enhance real-world document analysis workflows.

Key Words: Generative AI, Financial Documents, Semantic Search, LLM, ChromaDB, MongoDB.

1.INTRODUCTION

Annual reports, audit statements, and similar financial records contain large amounts of detailed information. Manually searching these documents for specific data points can be slow and exhausting. With the growth of digital financial archives, there is an increased need for tools that can process such material automatically and provide rapid access to relevant insights.

This project presents a system designed to simplify interaction with financial documents. Rather than requiring users to manually read lengthy PDFs, the application ingests the files, divides the extracted text into manageable segments, and processes each using modern AI models. These segments are stored as embeddings in a vector database to facilitate meaning-based retrieval instead of relying only on keyword matches. Relevant structured information extracted from the documents is stored in MongoDB for organised and editable access.

The central aim is to bring together advances in natural-language processing and practical storage tools to create an efficient and user-friendly solution. Features such as conversational querying, automated extraction, and an interactive interface collectively reduce manual workload and make financial document analysis more efficient and reliable

2. LITERATURE SURVEY

Research on financial document automation has accelerated as organisations increasingly rely on digital reporting. Amri et al. describe approaches in which generative AI models transform dense financial narratives into concise, meaningful summaries, demonstrating the potential of LLMs to assist with expert-level review tasks.

Yue and colleagues investigate techniques for processing long and complex documents by leveraging document chunking combined with embedding-based retrieval. Their findings indicate that dividing documents into granular pieces can enhance both speed and accuracy. Bailey and Schmidt discuss how AI tools support financial analysis by assisting professionals in locating significant indicators more efficiently.

Industrial platforms—such as those created by V7 Labs, cr.ai, and Datategy’s papAI—illustrate how AI is being applied in real business environments to automate table extraction, semantic search, and document classification. Although many available systems focus on individual tasks like summarisation or extraction, fewer integrate conversational search, semantic retrieval, structured data handling, and audit support within a single application. This gap underpins the motivation for the present work.

3. PROBLEM STATEMENT

Financial reports are typically long, unstructured, and filled with critical information that users must locate manually. This manual effort can consume significant time and often requires repeated scanning of the same content. Tools that offer only basic PDF extraction or keyword search are limited because they cannot interpret context or meaning, resulting in inefficient document navigation.

There is a lack of a comprehensive platform that can ingest financial documents, interpret their contents, store extracted data in an organised format, and allow users to interact with the information conversationally. This project aims to design a system that reduces manual labour, increases accuracy, and provides an integrated approach to exploring financial information using modern AI and database technologies.

4. PROJECT REQUIREMENT SPECIFICATION

The development of the financial document analysis system requires both software tools and hardware resources that support efficient processing of text, embeddings, and database operations. The specifications listed below outline the essential components needed to implement and run the system smoothly.

4.1 Software Requirements

Python: Core programming language used to implement PDF handling, embedding generation, and interface integration.

OpenAI API: Provides language-model capabilities for semantic search, summarisation, and auditing assistance.

ChromaDB: Vector storage system responsible for holding embeddings and enabling semantic similarity search.

MongoDB Atlas: Cloud-based database for storing and managing structured financial data extracted from documents.

Streamlit: Web-based framework that supports an interactive, user-friendly interface for all system features.

4.2 Hardware Requirements

RAM: Minimum of 8 GB to execute PDF processing, embedding generation, and database operations smoothly.

Processor: Intel i5 or equivalent to handle concurrent tasks such as API communication, parsing, and retrieval.

Storage: At least 20 GB of free space for extracted text, temporary files, and caching needs.

Internet Connectivity: Required for accessing the OpenAI API and MongoDB Atlas services**.**

5. SYSTEM ARCHITECTURE

The system follows a modular design that reflects the actual implementation carried out during the project. Each module handles a specific stage of the financial document processing pipeline, allowing the entire system to function reliably and remain easy to maintain.

5.1 Data Ingestion Layer

Users upload PDF documents through the Streamlit interface. The system extracts the text and prepares it for further processing. This module handles file reading, text cleaning, and ensures that the extracted content is suitable for analysis.

5.2 Text Chunking and Embedding Generation

After extraction, the document text is divided into smaller segments to preserve context and improve search accuracy. Each segment is converted into an embedding using an OpenAI model. These embeddings serve as the foundation for semantic search across the document collection.

5.3 Vector Database (ChromaDB)

The generated embeddings are stored in ChromaDB, which allows fast and meaning-based retrieval of information. Whenever a user asks a question, the system searches ChromaDB to fetch the most relevant text chunks rather than relying on plain keyword matching.

5.4 Structured Data Storage (MongoDB Atlas)

Key financial information extracted from the PDFs—such as tables, values, and important summaries—is stored in MongoDB. This makes it easier to view, edit, or audit the data. MongoDB’s flexible schema supports different document formats and financial structures.

5.5 Semantic Query and Response Engine (OpenAI)

The OpenAI model forms the intelligence layer of the system. It uses the retrieved context from ChromaDB to answer user queries, generate summaries, and support auditing tasks. This allows users to interact with the system in a natural conversational manner.

5.6 User Interface (Streamlit)

All features are integrated into a Streamlit application with a clean, multi-page layout. Users can upload PDFs, perform semantic searches, inspect stored data, and run audits. The interface allows smooth interaction with both the vector database and MongoDB.

This completed architecture brings together AI models, vector search, structured storage, and an intuitive UI to streamline the analysis of financial documents.

6. RESULTS AND SYSTEM FEATURES

The implemented system successfully enables automated analysis of financial documents. PDF files can be uploaded and processed, with text accurately extracted and segmented. The embedding-based pipeline allows the system to perform semantic searches that identify relevant content even when queries do not match the original phrasing.

Structured information stored in MongoDB provides a clear and manageable way to inspect and update financial data. The AI-supported auditing tool evaluates entries for potential inconsistencies by referencing the original financial content.

The front-end interface consolidates all features, enabling users to switch easily between document chat, database browsing, and management tools. The overall system demonstrates how combining generative AI, vector search, and an intuitive interface can simplify complex document-analysis tasks.

7. CONCLUSIONS

The system developed in this project demonstrates a practical and efficient approach to analysing financial documents using modern AI tools. By combining text extraction, semantic embeddings, vector search, and structured storage, the application provides a streamlined alternative to manually reviewing long and complex financial reports. Users can ask questions, retrieve relevant information, and inspect important data without going through entire documents.

The integration of ChromaDB and MongoDB offers both flexibility and speed, enabling smooth transitions between unstructured text and structured records. The conversational interface further enhances usability by allowing users to interact with documents in a natural, intuitive way. Overall, the project shows that generative AI and vector-based search methods can significantly reduce effort, improve accuracy, and support faster decision-making when dealing with financial information.

ACKNOWLEDGEMENT

We express our sincere gratitude to our project guide, **Prof. Swati Shriyal**, for her continuous support, guidance, and encouragement throughout the development of this work. Her insights and suggestions were valuable at every stage of the project. We also thank **Vishwakarma University** for providing the resources, infrastructure, and environment that enabled us to carry out this project successfully.

REFERENCES

[1] S. Amri, R. Bani and S. Bani, “An Approach to the Analysis of Financial Documents Using Generative AI,” 2024.

[2] C. Yue, et al., “Efficient Information Extraction from Hybrid Long Documents,” *arXiv*, 2024.

[3] K. Bailey and J. Schmidt, “AI and Financial Statement Analysis: Tools and Techniques,” 2025.

[4] V7 Labs, “AI for Financial Statement Analysis,” 2025.

[5] cr.ai, “Generative AI for Financial Analysis,” 2025.

[6] Datategy, “NLP-Based Document Processing with papAI,” 2025.

[7] pdf.ai, “Top Free AI Tools for Financial Analysis,” 2025.