IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.15160

AI Powered PDF Chat Application

H S Shreyas[1], Vishvanath A G[2]

Department of MCA, BIT,

K.R. Road, V.V. Pura, Bangalore, India[1,2]

Abstract: Extracting meaningful information from large PDF documents remains a significant challenge for students, researchers, and professionals, as traditional document reading and keyword-based search methods are time-consuming and inefficient. Existing tools often lack semantic understanding, contextual awareness, and interactive capabilities, making it difficult for users to obtain precise answers from complex documents. This limitation results in reduced productivity and increased effort when analysing lengthy technical, academic, or legal PDFs.

To overcome these challenges, the AI Powered PDF Chat Application integrates Generative AI, Large Language Models (LLMs), and vector-based semantic search techniques to enable intelligent document interaction. The system processes uploaded PDF documents by extracting text, generating embeddings, and storing them in a vector database to support context-aware retrieval. When a user submits a query, the application identifies the most relevant document segments using similarity search and generates accurate, context-driven responses through an AI language model. This approach ensures that answers are grounded in the document content rather than relying on generic responses.

The application is implemented as a secure web-based platform featuring user authentication, PDF preview, interactive chat interface, conversation history management, and PDF-based chat export functionality. By combining retrieval-augmented generation with real-time user interaction, the system significantly improves information accessibility, reduces document analysis time, and enhances user comprehension. The proposed solution demonstrates how AI-driven document intelligence can transform traditional PDF reading into an efficient, interactive, and scalable knowledge retrieval experience.

I. INTRODUCTION

Efficient extraction of information from digital documents is essential in academic, professional, and research environments, yet users often struggle to obtain precise answers from large PDF files. Traditional document reading methods and keyword-based search tools lack contextual understanding and require significant manual effort, leading to reduced productivity and delayed decision-making. As document sizes grow and information complexity increases, users face difficulties in locating relevant content quickly and accurately.

The AI Powered PDF Chat Application integrates Generative AI, Large Language Models (LLMs), and semantic search techniques to enable intelligent interaction with PDF documents. Users upload PDF files and ask natural language questions, while the system retrieves relevant document context and generates accurate, document-grounded responses. By combining text extraction, embedding-based similarity search, and AI-driven answer generation, the application transforms static documents into interactive knowledge sources. This approach improves comprehension, reduces document analysis time, and enhances overall user efficiency.

By enabling conversational document querying and contextual response generation, the proposed system democratizes access to intelligent document analysis for students, professionals, and researchers without requiring specialized technical expertise.

1.1 Project Description

Document analysis and information retrieval from PDF files present a significant challenge due to the unstructured nature of content and the volume of data involved. Users such as students, researchers, and professionals often rely on manual reading or basic search functions, which fail to provide semantic understanding, contextual relevance, or precise answers. Conventional tools do not support intelligent querying, contextual retrieval, or dynamic interaction with document content, resulting in inefficient workflows and limited insight extraction.

To address these limitations, the AI Powered PDF Chat Application leverages advanced technologies such as Large Language Models and vector-based semantic search for real-time document analysis. The system processes uploaded PDFs by extracting text, generating embeddings, and storing them in a vector database to enable similarity-based retrieval. Users can interact with documents through a chat interface, receive context-aware answers, manage chat history, and export conversations as PDF reports. The application provides an end-to-end solution for intelligent document interaction, from PDF upload to accurate answer generation.

1.2 Motivation

Accessing relevant information from digital documents is a critical requirement across education, research, and industry domains. Users frequently encounter challenges such as information overload, lack of contextual search, and time-intensive document scanning, which negatively impact learning efficiency, research accuracy, and professional productivity. Even small delays in identifying key information can lead to missed insights, incorrect interpretations, and reduced decision quality.

Despite the widespread use of PDFs, existing document tools primarily focus on static viewing and keyword matching, offering limited support for semantic understanding or interactive exploration. These approaches are often reactive and insufficient for complex documents containing technical, academic, or legal information. The absence of intelligent retrieval mechanisms makes it difficult for users to verify information relevance, trace context, and understand document relationships effectively.

Recent advancements in Artificial Intelligence, particularly in Natural Language Processing and Retrieval-Augmented Generation (RAG), enable early identification of relevant content and accurate answer generation grounded in source documents. Motivated by these developments, this project aims to build an AI Powered PDF Chat Application that combines semantic retrieval with real-time conversational AI. The system enhances document accessibility, reduces cognitive load, and provides a reliable, transparent, and efficient framework for intelligent PDF-based knowledge retrieval.

II. RELATED WORK

Several studies have explored artificial intelligence–based approaches for document understanding, information retrieval, and question answering using natural language processing and machine learning techniques. Transformer-based models such as BERT, GPT, and other large language models have demonstrated strong performance in semantic text understanding, contextual reasoning, and natural language response generation. Research in document question answering systems highlights the effectiveness of embedding-based similarity search and attention mechanisms for retrieving relevant text segments from unstructured documents.

Other works focus on retrieval-augmented generation (RAG) frameworks, where vector databases such as FAISS or Pinecone are used to store document embeddings and retrieve contextually relevant content before generating answers. These systems significantly improve answer accuracy by grounding responses in source documents rather than relying solely on generative capabilities. Optical character recognition (OCR) and PDF text extraction techniques are also widely studied to handle diverse document formats and layouts.
AI-powered document interaction tools emphasize improving user productivity, reducing manual reading effort, and enabling conversational access to large document collections. However, many existing solutions address document retrieval and answer generation as separate components or lack interactive chat-based interfaces with session management and export capabilities. There remains a research gap in developing an end-to-end system that seamlessly integrates PDF ingestion, semantic retrieval, conversational AI, chat history management, and response export within a single unified platform. The proposed AI Powered PDF Chat Application addresses this gap by combining intelligent document processing, vector-based retrieval, and conversational AI into a cohesive framework for efficient and accurate PDF-based knowledge interaction.

III. METHODOLOGY

A. System Architecture Overview

The AI-Powered PDF Chat Application is implemented as a full-stack web-based system that integrates a user-friendly frontend, a backend processing layer, and an AI-driven retrieval and response engine. The system follows a modular architecture where each component is designed to perform a specific function, ensuring scalability, maintainability, and efficient data flow.

The frontend layer provides interfaces for user authentication, PDF upload, document preview, question input, and response display. The backend layer manages user sessions, file handling, text extraction, embedding generation, vector storage operations, and chat history management. The AI processing layer interacts with the Large Language Model (Google Gemini API) to generate context-aware responses based on retrieved document content.

The application employs a Retrieval-Augmented Generation (RAG) approach, where extracted document text is converted into vector embeddings and stored in a vector database. During query processing, relevant document context is retrieved and combined with the user's question before being sent to the AI engine. This architecture ensures accurate, document-specific responses while maintaining efficient performance and data consistency.

B. Document Processing and Query Customization

Users begin by uploading PDF documents through the web interface. The system validates the uploaded file and extracts textual content from the document. The extracted text is segmented into manageable chunks and converted into numerical embeddings using an embedding model. These embeddings are stored in a vector store to enable fast and accurate similarity-based retrieval.

When a user asks a question, the system converts the query into an embedding and retrieves the most relevant document chunks from the vector store. The retrieved context is dynamically combined with the user's query and sent to the AI engine for response generation. The generated response is then displayed to the user in real time.

The system maintains a session-based chat history that allows users to view previous interactions, download the conversation as a PDF, or clear stored chat data. This approach enables personalized, document-centric question answering while ensuring efficient reuse of processed document data across multiple queries.

C. Generative AI–Based Query Processing Module

The AI-Powered PDF Chat Application uses a Generative AI–based query processing module to generate accurate, document-specific responses to user questions. Instead of generating questions, this module focuses on understanding user queries and retrieving relevant information from uploaded PDF documents.

When a user submits a query, the system converts the query into a vector embedding using an embedding model compatible with the document embeddings. The generated query embedding is compared against stored document embeddings in the vector database using similarity search techniques. The most relevant document chunks are selected based on semantic similarity.

These retrieved document segments are combined with the user query and sent to the Large Language Model (Google Gemini API) for response generation. By grounding the AI response in retrieved document context, the system ensures that answers are accurate, relevant, and directly derived from the uploaded PDF content. This Retrieval-Augmented Generation (RAG) approach minimizes hallucinations and improves response reliability.

D. Document Text Extraction and Response Generation

Upon successful PDF upload, the system extracts textual content from the document using PDF parsing techniques. The extracted text is cleaned, segmented into smaller chunks, and processed for embedding generation. This preprocessing step enables efficient storage and retrieval of document information during query execution.

When a response is generated by the AI engine, it is returned to the backend and displayed to the user in real time through the web interface. Each interaction is stored as part of the session-based chat history, allowing users to revisit previous answers, export conversations as a PDF, or clear stored chat data when required. The system includes error-handling mechanisms to manage invalid PDF files, extraction failures, or AI service interruptions. User-friendly messages guide the user in case of errors, ensuring smooth and uninterrupted interaction with the application.

DOI: 10.17148/IJARCCE.2026.15160

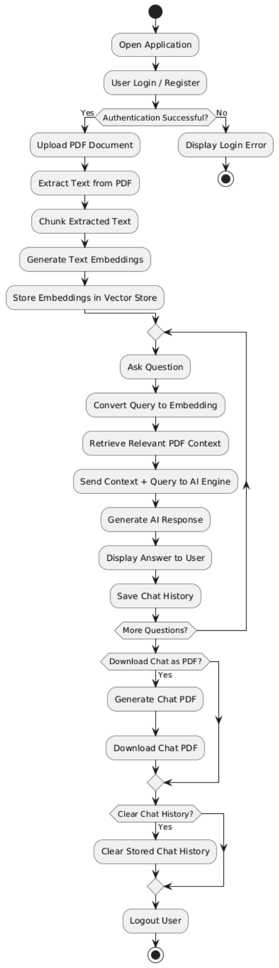


Fig. 1. Flowchart of methodology.

E. Implementation Flow

1. Initialize the system environment and authenticate users to ensure secure access to the application
2. Create a new user session by generating a unique session identifier and associating it with the uploaded document context for consistent interaction tracking.
3. Upload the PDF document through the web interface and validate the file format, size, and readability before processing.
4. Extract textual content from the uploaded PDF and preprocess it by cleaning, normalizing, and splitting the text into smaller chunks suitable for semantic analysis.
5. Apply Natural Language Processing techniques to generate vector embeddings for each text chunk, capturing semantic meaning for accurate information retrieval.
6. Store all generated embeddings securely in a vector database to support fast similarity search during user queries.
7. Process user questions by retrieving the most relevant document chunks from the vector store based on semantic similarity.
8. Generate AI-based responses using the retrieved document context to ensure answers are accurate, relevant, and grounded in the uploaded PDF content.
9. Log all user interactions, queries, responses, and session details to maintain chat history, enable session continuity, and support system optimization and analysis.

F. Hardware and Software Requirements

- Standard desktop or laptop system with a minimum of 8 GB RAM and a quad-core processor to ensure smooth PDF processing, embedding generation, and AI-based response generation

- Python 3.x runtime with Flask framework for backend development, HTML, CSS, and JavaScript for frontend interface design, SQLite for user authentication and chat history storage, FAISS for vector-based semantic retrieval, Google Gemini API for generative AI response generation, PyMuPDF (fitz) for PDF text extraction, and supporting libraries such as NumPy, Flask-Login, Flask-SQLAlchemy, and WeasyPrint for database management, authentication, and PDF export functionality..

IV. SIMULATION AND EVALUATION FRAMEWORK

This section outlines the overall system design, evaluation workflow, and performance assessment approach adopted for the proposed AI Powered PDF Chat Application**.** The framework integrates Artificial Intelligence (AI), semantic search, and web-based technologies to simulate intelligent document interaction, evaluate query–response accuracy, and generate reliable, document-grounded answers. The system is implemented as a secure web-based platform, with a Flask-based backend and HTML, CSS, and JavaScript–driven frontend, enabling real-time PDF interaction, semantic retrieval, and conversational querying. The evaluation process focuses on assessing response relevance, contextual accuracy, retrieval effectiveness, and system reliability using AI-driven analysis, ensuring consistent, objective, and scalable document intelligence for users.

A. System Architecture and Workflow

The proposed architecture is designed to support interactive PDF-based querying, semantic retrieval, and AI-driven response generation. The system ensures seamless interaction between users and AI components while maintaining consistency, scalability, and secure data handling. The major components of the system are described below:

Web-Based Document Interaction Platform:
The application provides authenticated access for users to upload PDF documents, preview document content, submit natural language queries, and view AI-generated responses in real time. The platform supports session-based chat history management, document-specific interaction, and PDF export of conversations through an intuitive user interface.

AI and Semantic Retrieval Layer:
The AI processing layer employs Large Language Models (Google Gemini API) combined with vector-based semantic search to generate accurate, context-aware responses. Extracted PDF text is converted into embeddings and stored in a FAISS vector database. When a query is submitted, relevant document segments are retrieved using similarity search and provided as contextual input to the AI model, ensuring responses are grounded strictly in document content.

Authentication and Data Management Module:
Secure user authentication and session handling are implemented using Flask-Login and database-backed credential storage. Uploaded documents, vector indices, and chat histories are managed securely to ensure data integrity, user isolation, and session continuity across multiple interactions.

Analytics and Interaction Logging Layer:
The system logs user queries, retrieved document context, AI-generated responses, and interaction timestamps. This information supports performance evaluation, response consistency verification, and future system optimization while maintaining transparency and traceability.

B. System Evaluation Setup

The evaluation framework is designed to measure the effectiveness of the AI Powered PDF Chat Application under realistic document analysis scenarios. Multiple test cases are executed using PDFs of varying size, structure, and domain to assess retrieval accuracy, response relevance, and system performance.

Document Configuration:
PDF documents from academic, technical, and informational domains are uploaded to evaluate system performance across different content structures, lengths, and complexity levels.

Query Interaction Scenarios:
Users submit diverse natural language queries, including factual questions, summary requests, and explanation-based queries, to evaluate semantic retrieval accuracy and AI response reliability under real-world usage conditions.

DOI: 10.17148/IJARCCE.2026.15160

C. Evaluation and Verification Process

Each user interaction is uniquely associated with a session identifier that links uploaded documents, query inputs, retrieved context, and AI-generated responses. As users interact with the system, all queries and responses are processed and stored securely. Users can review previous interactions, clear session history, or export conversations as PDF reports. This evaluation and verification process ensures transparent, repeatable, and trustworthy document intelligence by validating that all responses are contextually accurate, traceable to source documents, and consistently generated across multiple interaction scenarios.

D. Results and Observations

Document Query and Response Accuracy:

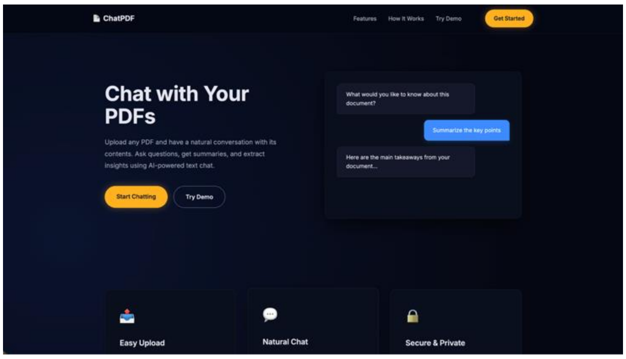
- The AI-generated responses were contextually accurate and strictly grounded in the content of the uploaded PDF documents.
- Semantic search and embedding-based retrieval successfully identified relevant document segments for a wide range of user queries, including factual, explanatory, and summary-based questions.

System Reliability and Consistency:

- PDF upload, text extraction, embedding generation, and query processing were executed without data loss or processing errors across multiple sessions.
- The system consistently retrieved relevant document context and generated responses in real time, demonstrating stable performance and reliable AI-assisted document interaction.

User Impact:

- Users were able to extract meaningful information from large PDF documents efficiently, significantly reducing manual reading and analysis time.
- The interactive chat interface, session-based history, and PDF export functionality improved user experience, document comprehension, and overall productivity.



Chat with Your PDFs

Upload any PDF and have a natural conversation with its contents. Ask questions, get summaries, and extract insights using AI-powered text chat.

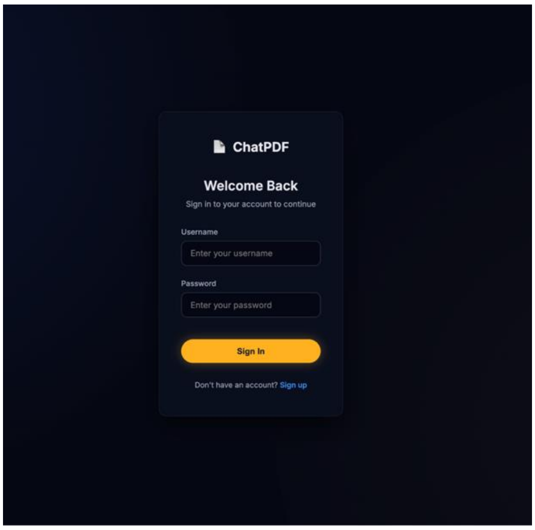
What would you like to know about this document?

Summarize the key points

Here are the main takeaways from your document...

Fig.2. Landing Page

DOI: 10.17148/IJARCCE.2026.15160


ChatPDF

Welcome Back

Sign in to your account to continue

Username

Enter your username

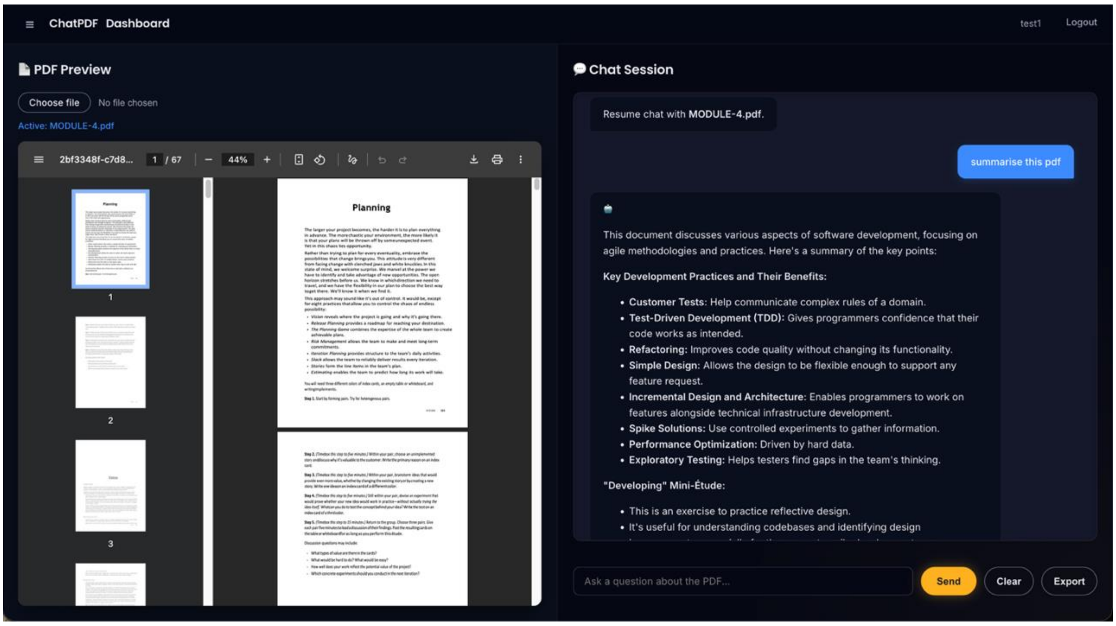
Password

Enter your password

Don't have an account? Sign up

Fig.3. Login Page

- The Login Page provides secure user authentication by allowing registered users to access the application using valid credentials.
- The PDF Interaction Page displays the document preview alongside an interactive chat interface, enabling users to view the uploaded PDF and simultaneously ask questions related to its content.



Active: MODULE-4.pdf

2bf3348f-c7d8... 1 / 67 44%

Planning

Resume chat with **MODULE-4.pdf**.

summarise this pdf

This document discusses various aspects of software development, focusing on agile methodologies and practices. Here's a summary of the key points:

Key Development Practices and Their Benefits:

- **Customer Tests:** Help communicate complex rules of a domain.
- **Test-Driven Development (TDD):** Gives programmers confidence that their code works as intended.
- **Refactoring:** Improves code quality without changing its functionality.
- **Simple Design:** Allows the design to be flexible enough to support any feature request.
- **Incremental Design and Architecture:** Enables programmers to work on features alongside technical infrastructure development.
- **Spike Solutions:** Use controlled experiments to gather information.
- **Performance Optimization:** Driven by hard data.
- **Exploratory Testing:** Helps testers find gaps in the team's thinking.

"Developing" Mini-Étude:

- This is an exercise to practice reflective design.
- It's useful for understanding codebases and identifying design

Ask a question about the PDF...

Fig. 4. PDF Integration Page

V.RESULTS AND DISCUSSION

The experimental evaluation of the proposed AI Powered PDF Chat Application demonstrates its effectiveness in enabling intelligent document interaction through semantic retrieval and AI-driven response generation. Multiple PDF documents of varying size and complexity were used to evaluate system performance under realistic academic and professional document analysis scenarios.

The results indicate that the retrieval-augmented generation mechanism consistently produced accurate and context-aware responses grounded in the uploaded PDF content. By combining vector-based similarity search with a large language model, the system successfully identified relevant document segments and generated precise answers aligned

DOI: 10.17148/IJARCCE.2026.15160

with user queries. Compared to traditional keyword-based search methods, the AI-driven approach significantly improved information relevance and reduced the effort required to locate specific content within large documents.

The document processing and embedding generation pipeline effectively handled PDF uploads, text extraction, and chunk-based semantic indexing. Query processing was performed in real time, enabling users to receive responses with minimal latency. The use of semantic similarity search ensured that responses were contextually accurate rather than generic, enhancing user trust and response reliability.

Additionally, the interactive chat interface and session-based history management enabled users to maintain continuous document-centric conversations. Users could resume previous chats, clear stored interactions, and export AI-generated responses as PDF reports, supporting both short-term analysis and long-term reference. The system maintained reliable storage of document embeddings and conversation data, ensuring consistency across multiple interactions with the same document.

Overall, the integrated platform demonstrated improved document comprehension, reduced analysis time, and efficient knowledge extraction from complex PDF files. The results confirm that the AI Powered PDF Chat Application provides a scalable, reliable, and user-friendly solution for intelligent document understanding while maintaining accuracy, transparency, and practical usability in real-world scenarios.

VI. CONCLUSION

This project demonstrates the feasibility and effectiveness of applying Artificial Intelligence and modern web technologies to enable intelligent interaction with PDF documents. The proposed AI Powered PDF Chat Application successfully transforms static PDF files into interactive knowledge sources by combining semantic retrieval techniques with AI-driven natural language response generation.

The integration of document text extraction, vector-based semantic search, and Retrieval-Augmented Generation allows users to ask natural language questions and receive accurate, context-aware responses grounded in the uploaded document content. Unlike traditional keyword-based search or manual document reading, the system provides precise information retrieval while significantly reducing analysis time and user effort.

Additionally, the application ensures secure user authentication, reliable document processing, and persistent chat session management. Features such as PDF preview, conversation history tracking, and chat export functionality enhance usability and support both short-term analysis and long-term reference. By maintaining document-specific embeddings and structured interaction records, the system enables consistent and transparent access to AI-generated insights.

Overall, the project validates that AI-driven document intelligence can greatly improve information accessibility, comprehension, and productivity. The AI Powered PDF Chat Application offers a scalable, reliable, and user-friendly solution for students, researchers, and professionals seeking efficient knowledge extraction from complex PDF documents.

VII. FUTURE WORK

While the proposed AI Powered PDF Chat Application effectively demonstrates intelligent document interaction using semantic retrieval and generative AI, several enhancements can be explored to extend its functionality and real-world applicability. Future work may focus on supporting additional document formats such as Word documents, scanned PDFs with Optical Character Recognition (OCR), and multi-language content to broaden system usability.

Another potential enhancement involves improving contextual reasoning by incorporating advanced embedding models and larger document chunking strategies. This can further enhance response accuracy for complex or cross-referenced queries. Integrating citation highlighting or source linking within responses could also improve transparency by clearly indicating the document sections used to generate answers.

Scalability improvements may include cloud-based deployment, distributed vector storage, and optimization for handling large document repositories and concurrent users. Additionally, future versions of the system could integrate user analytics dashboards to visualize document usage patterns and query trends.

DOI: 10.17148/IJARCCE.2026.15160

These extensions would enhance the system's robustness, scalability, and interpretability, enabling wider adoption of AI-powered document intelligence solutions across academic, research, and professional environments.

REFERENCES

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of NAACL-HLT*, 2019.

[2] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] Facebook AI Research, "FAISS: A Library for Efficient Similarity Search and Clustering of Dense Vectors," *FAISS Documentation*, 2023. [Online]. Available: https://faiss.ai

[4] Google Research, "Gemini API Documentation," 2024. [Online]. Available: https://ai.google.dev

[5] A. Vaswani et al., "Attention Is All You Need," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[6] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, 2009.

[7] Flask Development Team, "Flask: A Lightweight WSGI Web Application Framework," 2024. [Online]. Available: https://flask.palletsprojects.com

[8] Python Software Foundation, "SQLite Database Engine Documentation," 2024. [Online]. Available: https://www.sqlite.org/docs.html

[9] Mozilla Developer Network (MDN), "HTML, CSS, and JavaScript Web Standards," 2024. [Online]. Available: https://developer.mozilla.org

[10] P. G. Fitzpatrick, "PyMuPDF Documentation: PDF Text Extraction and Processing," 2024. [Online]. Available: https://pymupdf.readthedocs.io