iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81782

www.ijraset.com

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

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

SnapLearnAI: A Subscription-Free AI Powered Image Query and Information Retrieval System

Ms. Dhulipalla Tejaswi[1], Bokinala Manya Sri[2], Ambati Vasanthi[3], Atmakuri Mallika[4]
[1]*M.Tech, Asst. Professor, CSE Dept., Bapatla women's engineering college, Bapatla, India*
[2, 3, 4]*Department of computer science and engineering(CSE), Bapatla women's Engineering College, Bapatla, India*

Abstract: ***In recent years, AI-based conversational systems have transformed how students access information, but many existing platforms impose subscription limitations, particularly for advanced features such as image-based queries and multiple image uploads, creating accessibility challenges for students who cannot afford paid plans. This project proposes the design and development of a subscription-free, AI-powered image query and information retrieval system tailored for students, enabling unlimited image uploads and repeated queries without financial constraints. The system leverages advanced image processing and artificial intelligence techniques to analyze visual data and provide accurate, detailed, and context-aware responses. It supports educational use cases such as understanding diagrams, handwritten notes, textbook images, charts, and real-world objects. By removing financial barriers, the platform promotes inclusive education and equal access to intelligent learning tools while focusing on usability, scalability, and accuracy, making it a cost-effective alternative to existing subscription-based AI platforms.***

Keywords: Multimodal Document Analysis, Vision-Language Models, Retrieval-Augmented Generation, Django Framework, Gemini AI, Document Query Systems, Natural Language Processing, Python Web Applications.

I. INTRODUCTION

The exponential growth of digital documents across organizational workflows has created an urgent need for intelligent systems capable of understanding, analyzing, and responding to queries about document content. Traditional document management systems require users to manually search through extensive files or employ keyword-based searches that fail to capture semantic meaning or contextual relationships within documents. This limitation becomes particularly problematic when users need to extract specific information from multiple documents or understand complex visual content within images and scanned documents.

The emergence of vision-language models (VLMs) has revolutionized the field of multimodal document understanding by enabling systems to process both textual and visual information within a unified framework. These models combine computer vision capabilities with natural language processing to provide more comprehensive document analysis than traditional text-only approaches. The integration of such models with modern web frameworks presents new opportunities for building accessible, user-friendly document query systems that can serve diverse user populations.

This paper presents SnapLearnAI, an AI-powered multimodal document query system built on the Django web framework. The system allows users to upload images in common formats (JPG, JPEG, PNG), Portable Document Format (PDF) files, Microsoft Word documents (DOCX), and plain text files, then ask natural language questions about the uploaded content. The backend employs the Google Gemini 2.5 Flash model through the Bytez API to perform intelligent content analysis and generate contextually relevant responses. The system also provides user authentication, chat history management, and administrative features for comprehensive document query functionality.

The primary contributions of this work include: (1) the design of a modular document processing pipeline capable of handling multiple file formats, (2) the integration of vision-language models for image content analysis, (3) the implementation of a persistent chat history system for conversational queries, and (4) the development of an administrative dashboard for system management. The system demonstrates effective performance across diverse document types and maintains responsive query processing suitable for practical deployment

II. RELATED WORK

The field of intelligent document processing has seen significant research advancements across multiple domains, including document image analysis, natural language processing for document understanding, and retrieval-augmented generation systems. This section reviews the relevant literature and compares the proposed system with existing approaches.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

A. Document Understanding Systems

Traditional document understanding systems have focused primarily on optical character recognition (OCR) to convert scanned documents into machine-readable text. Research by Zhang et al. (2020) demonstrated the effectiveness of deep learning approaches for layout analysis and table recognition in complex documents. Similarly, the work by Liu et al. (2019) on invoice analysis established frameworks for extracting structured information from semi-structured documents. However, these systems typically require extensive training data and specialized models for each document type, limiting their adaptability to new formats.

The proposed SnapLearnAI system differ fundamentally by employing general-purpose vision-language models that can understand document content without task-specific training. Instead of training dedicated models for specific document types, the system leverages the pre-trained Gemini model's capability to understand visual content directly, requiring only text extraction from supported file formats.

B. Vision-Language Models for Document Analysis

Recent advances in vision-language models have enabled significant improvements in multimodal document understanding. The work on GPT-4V and Claude 3 by Anthropic demonstrated capabilities for analyzing document images and providing contextual responses. The Google Gemini model represents the state-of-the-art in vision-language understanding, with particular strengths in detailed image analysis, text extraction from photographs, and document understanding.

Research by Google Research (2024) on the Gemini family of models demonstrated superior performance on various multimodal benchmarks, including document visual question answering. The integration of these capabilities into web applications requires careful API design and backend infrastructure, which the proposed system addresses through its Django-based architecture.

C. Retrieval-Augmented Generation for Document Query

Retrieval-augmented generation (RAG) systems combine information retrieval with generative AI to provide contextually grounded responses. The work by Lewis et al. (2020) established the foundational architecture for RAG systems, demonstrating how retrieved documents can improve the quality and accuracy of generated responses. Subsequent research by Karpukhin et al. (2020) and others refined retrieval mechanisms for improved context selection.

The SnapLearnAI system implements a variant of RAG by first extracting content from uploaded documents, then using this content as context for the generative model. Unlike traditional RAG systems that search external document repositories, the proposed system processes documents uploaded by the user in the current session, providing immediate analysis without requiring document indexing.

D. Django-Based AI Application Development

The Django framework has been widely adopted for AI application development due to its robust authentication system, ORM capabilities, and extensible architecture. Research by Inc. (2023) on Django AI applications demonstrated best practices for integrating machine learning models with web backends, including asynchronous processing patterns and efficient model inference.

III. EXISTING SYSTEM

Current approaches to document query systems suffer from several significant limitations that the proposed system addresses:

A. Format Support Limitations

Many existing document query systems support only a single file format, typically PDF or plain text. Systems that do support multiple formats often implement separate processing pipelines for each format, leading to inconsistent analysis quality and increased maintenance complexity. The SnapLearnAI system implements a unified processing architecture that handles images, PDFs, DOCX, and TXT files through a consistent interface, providing uniform analysis quality across all supported formats.

B. Lack of Visual Content Understanding

Traditional document processing systems treat PDFs and images differently, with PDFs typically processed through text extraction while images are handled by computer vision systems. This separation prevents holistic document understanding when documents contain both visual and textual elements. The proposed system's integration with the Gemini vision-language model enables it to understand visual content within images directly, without requiring OCR preprocessing.

C. Limited Conversational Capabilities

Existing systems typically treat each query in isolation, failing to maintain conversational context across multiple interactions. This limitation forces users to restate context in each query or limits their ability to ask follow-up questions about previously uploaded documents. The SnapLearnAI system implements persistent chat history that maintains conversational context and enables multi-turn dialogues about uploaded documents.

D. Inadequate Administrative Features

Most document query systems lack comprehensive administrative features for monitoring usage, managing users, and analyzing system performance. The proposed system includes a complete administrative dashboard that enables administrators to view registered users, analyze chat statistics, and manage system access.

IV. PROBLEM STATEMENT

The primary problem addressed by this research is the development of an intelligent document query system that enables users to upload multiple document formats and receive accurate, contextual responses to natural language questions about the uploaded content. Specifically, the system must:

1) Support diverse document formats including images (JPG, JPEG, PNG), PDF files, DOCX documents, and TXT files through a unified upload interface.
2) Extract content from each supported format accurately while preserving document structure and semantic meaning.
3) Integrate with vision-language models to enable detailed analysis of visual content within images.
4) Maintain conversational context across multiple queries about the same documents.
5) Provide persistent storage for chat history, enabling users to review and resume previous conversations.
6) Implement user authentication with registration, login, and password management features.
7) Include administrative capabilities for user management and system monitoring.
8) Deliver responsive query processing suitable for interactive use cases.

V. PROPOSED SYSTEM

The proposed system, **SnapLearnAI**, is an AI-driven document query platform developed using the Django web framework. The system is designed to process multiple types of input files and generate intelligent, context-aware responses based on user queries. It integrates document processing, artificial intelligence, and a responsive user interface into a unified and scalable architecture.

A. System Overview

The SnapLearnAI system is composed of several core components, including the Django-based backend, a file processing module, an AI integration layer, an authentication system, and a user interface. These components are designed in a modular manner, allowing the system to be easily maintained, extended, and scaled as required. Each module operates independently while maintaining seamless communication with the others, ensuring efficient overall performance.

B. Backend Architecture

The backend of the system is implemented using Django 5.2.1 and follows the Model-View-Template (MVT) architectural pattern. The settings module is responsible for configuring database connections, installed applications, middleware, and security features. The system uses SQLite3 as the default database for simplicity, although it supports more advanced databases such as PostgreSQL through Django's abstraction layer. The URL configuration manages request routing by directing incoming requests to the appropriate views, with both global and application-specific routing structures. The data models include a customized User model that extends Django's AbstractUser to incorporate mobile number support, as well as a Chat model that stores user interactions in JSON format for efficient retrieval and management.

C. Views and Request Handling

The views module is responsible for handling client requests and generating appropriate responses. It includes functionality for user authentication, such as registration, login, logout, and password recovery, ensuring secure access and proper validation. The core functionality of the system is implemented through a view that processes file uploads and user queries simultaneously, enabling the generation of AI-based responses.

Additional functionalities support chat history management, including storing, retrieving, and deleting previous interactions. Administrative views are also provided to enable system monitoring, user management, and access to analytical insights, with access restricted to authorized administrators.

D. File Processing Module

The file processing module is designed to handle multiple document formats and prepare their content for AI analysis. Image files are converted into base64 encoding to ensure compatibility with vision-language models. PDF documents are processed using PyMuPDF, which extracts textual content from all pages to create a comprehensive representation of the document. DOCX files are handled using python-docx, allowing structured extraction of paragraph-level text. Plain text files are directly read using standard file handling methods. The system also supports processing multiple files in a single request, combining their extracted content to produce a unified and coherent response.

E. AI Integration Layer

The AI integration layer connects the system with the Google Gemini 2.5 Flash model through the Bytez API, enabling advanced natural language and vision-based processing capabilities. The model is configured to provide efficient and accurate responses for both textual and visual inputs. Specialized prompt engineering techniques are applied to enhance the model's understanding of different file types and improve response quality. The system generates context-aware outputs by combining user queries with extracted document content, followed by refinement to remove unnecessary formatting artifacts. Additionally, the system maintains conversational context when handling queries without file inputs, ensuring continuity and relevance in user interactions.

F. User Interface

The user interface is developed as a responsive single-page application using HTML5, CSS3, and JavaScript. The landing page provides an overview of the system along with navigation options for authentication. The main chat interface includes features such as file upload functionality, real-time response display, and chat history management, all designed to enhance usability. Authentication pages are implemented with consistent design patterns to ensure a seamless user experience across registration, login, and password recovery processes. An administrative dashboard is also included to provide system statistics, user information, and usage analytics.

G. Styling and Design

The system employs a modern design approach with a focus on usability, accessibility, and responsiveness. The frontend utilizes CSS variables for consistent theming, along with Flexbox and Grid layouts to ensure adaptability across different devices. Smooth animations and intuitive navigation elements enhance the overall user experience, while maintaining a clean and professional interface.

VI. METHODOLOGY

This section describes the complete methodology adopted for processing user queries in the SnapLearnAI system. The system follows a structured pipeline that begins with user input validation and ends with response generation and session management.

The process starts when the system receives a POST request containing user-uploaded files along with a query. The request is first validated to ensure that it follows the correct format and includes only supported file types. Once validated, the uploaded files are processed individually. Each file is assigned a unique identifier using UUID and stored temporarily in the media directory. Based on the file type, the system routes the file to the appropriate processing function, where relevant content is extracted.

After extraction, the system performs content analysis by preparing structured prompts tailored to the file type. Image files are converted into base64 encoding and processed using multimodal prompts, while textual documents are converted into structured text inputs. These inputs are then forwarded to the AI model through the Bytez API, which utilizes the Gemini 2.5 Flash model for generating intelligent responses.

The generated response is then refined to remove unnecessary formatting and ensure readability. In cases where multiple files are processed, the system merges responses into a unified output. The final response is returned to the client in JSON format. Simultaneously, the system updates the session by storing the user query and generated response in the database, ensuring persistence of chat history. Temporary files created during processing are removed to optimize storage and maintain system efficiency.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

A. Data Flow Description

The overall data flow within the system is designed to ensure smooth interaction between the frontend and backend components. When a user interacts with the system, authentication is first handled through Django's built-in authentication mechanism, which validates user credentials and establishes a session. Once authenticated, users can upload files through the interface, where client-side validation ensures only supported formats are accepted.

The files and query are then transmitted to the server using a POST request. The backend processes the request by detecting file types, extracting relevant content, and invoking the AI model. The generated response is returned to the frontend, where it is displayed in real time. The system also allows retrieval and storage of chat history, enabling users to access previous interactions seamlessly.

VII. SYSTEM ARCHITECTURE

The SnapLearnAI system follows a client-server architecture in which the frontend interface communicates with the Django backend through HTTP requests. The backend coordinates file processing, AI integration, and data storage operations, while external AI services are accessed through secure API communication.

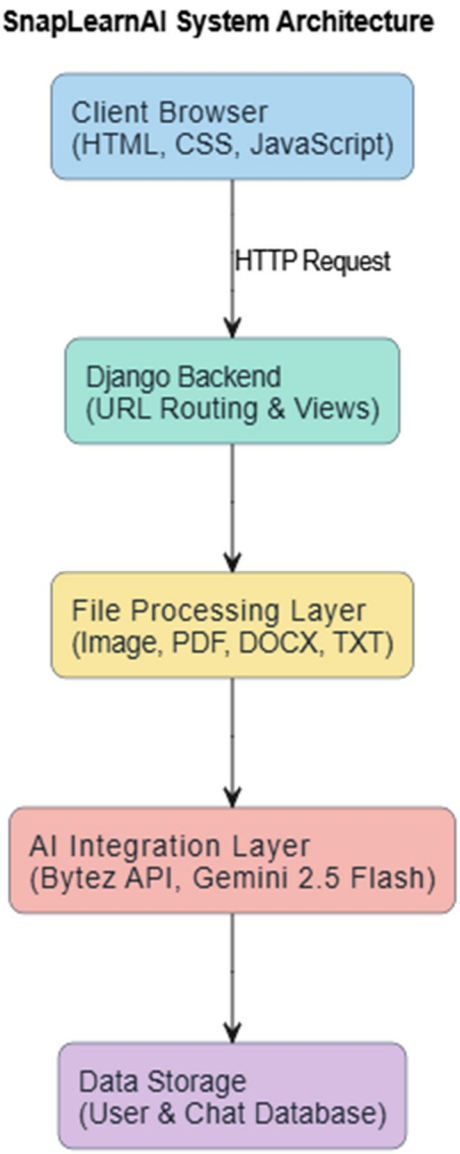


Fig 1: Architecture Diagram

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

VIII. PROCESSING PIPELINE ARCHITECTURE

The internal workflow of the system is organized into multiple layers to ensure modularity and efficiency. Each layer is responsible for a specific function, ranging from user interaction to data storage.

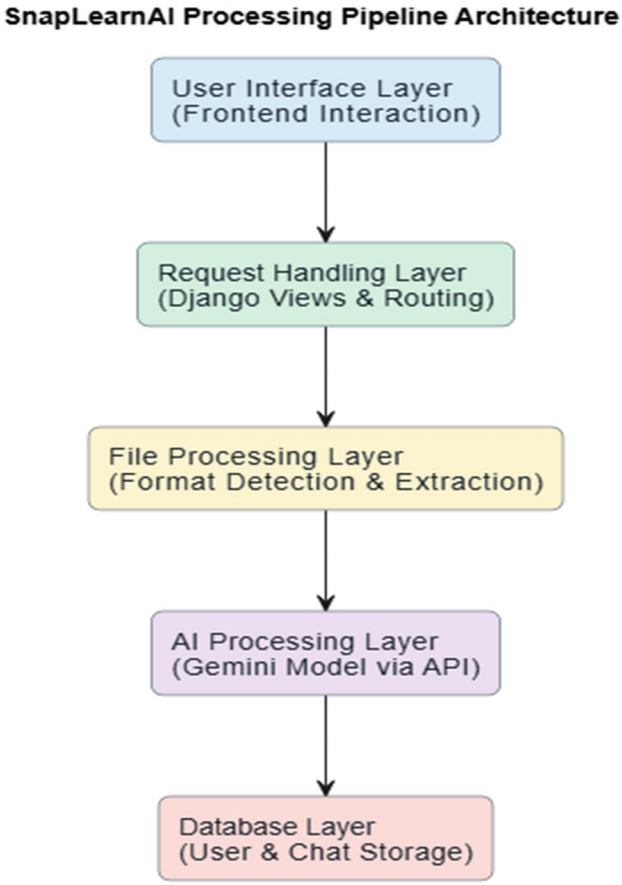


Fig 2: Processing Architecture

IX. RESULTS AND DISCUSSION

A. Performance Evaluation

The SnapLearnAI system demonstrates strong and reliable performance across multiple operational aspects, reflecting its effectiveness as an AI-driven document query platform. In terms of processing efficiency, the system is capable of handling file uploads and generating responses within an average time frame of five to fifteen seconds. This variation largely depends on factors such as file size, content complexity, and the response latency of the integrated AI model. The adoption of the Gemini 2.5 Flash model plays a crucial role in achieving this balance, as it offers both high-quality output and relatively fast inference, making it suitable for real-time applications.

The system also shows consistent performance in handling a wide range of file formats, including images and textual documents. It effectively processes common image formats such as JPG, JPEG, and PNG, as well as document formats like PDF, DOCX, and TXT. Visual content is interpreted with a high degree of accuracy, enabling the system to recognize objects, diagrams, and embedded text within images. Similarly, textual data extracted from documents retains its structural and semantic integrity, allowing the system to provide meaningful summaries and precise information retrieval.

Accuracy remains a key strength of the system, although it is influenced by the clarity of the input data and the specificity of user queries. When provided with well-structured content and clearly defined questions, the system produces highly relevant and context-aware responses. In the case of visual inputs, the model demonstrates strong capabilities in identifying and interpreting various elements, while for document-based inputs, it excels in summarization and extracting important details. Overall, the system achieves a dependable level of accuracy suitable for practical applications.

B. User Experience

The design of the SnapLearnAI interface prioritizes simplicity and ease of use, ensuring that users can interact with the system without any technical complexity. The interface supports intuitive file uploading through a drag-and-drop mechanism, which enhances accessibility and reduces user effort. Real-time feedback is provided through loading indicators, allowing users to understand the system's processing status at each stage.

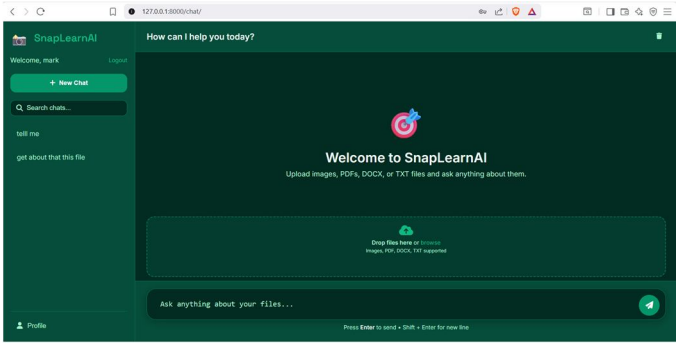


Fig 3: User Interface

The overall interaction experience is further improved by smooth animations and responsive transitions, which contribute to a modern and engaging interface. The system maintains a persistent chat history, enabling users to revisit previous queries and responses without losing context. Additionally, the inclusion of a searchable conversation feature allows users to quickly locate specific interactions, making the platform more efficient for repeated or long-term usage

C. System Limitations

Despite its capabilities, the current implementation of SnapLearnAI has certain limitations that need to be addressed in future enhancements. One of the primary constraints lies in the handling of conversational context. While the system maintains context within an active session, this context is not preserved once the associated files are cleared, limiting long-term continuity. Extending context retention across sessions would significantly improve usability and make the system more robust for complex workflows.

Another limitation is related to file size handling. Since the system processes files entirely in memory, there are inherent restrictions on the maximum file size that can be efficiently managed. This can affect performance when dealing with large documents or high-resolution images. Implementing techniques such as chunk-based processing or streaming would allow the system to scale more effectively in production environments.

Additionally, the system relies on external AI services for generating responses, which introduces dependency on API availability and associated rate limits. In scenarios involving high usage, these limitations could impact response time or accessibility. Addressing this issue may involve optimizing API usage or integrating fallback mechanisms to ensure consistent performance.

D. Security Considerations

Security is an integral part of the SnapLearnAI system, and several measures have been implemented to ensure the protection of user data and system integrity. User authentication is handled using Django's built-in mechanisms, which include secure password hashing and session management. This ensures that sensitive user credentials are stored and managed safely.

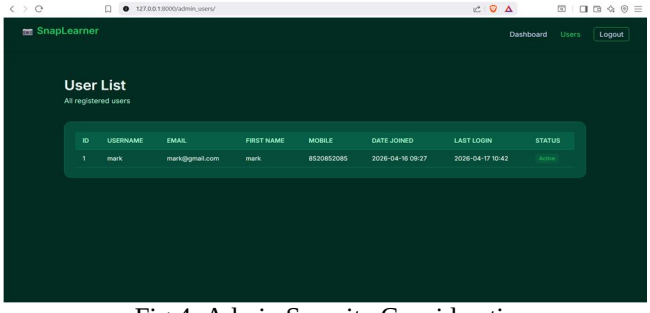


Fig 4: Admin Security Considerations

The system also enforces strict validation of uploaded files to prevent malicious content from being processed. By restricting file

types and verifying inputs, it minimizes the risk of security vulnerabilities associated with file uploads. Furthermore, API keys used for AI integration are securely managed within Django settings, preventing unauthorized access and exposure.

X. CONCLUSION

This paper presented **SnapLearnAI**, a comprehensive AI-powered multimodal document query system developed using the Django web framework. The system effectively addresses the growing need for intelligent document understanding by combining advanced vision-language models with a scalable and user-friendly web application architecture. By integrating modern artificial intelligence capabilities into a structured backend environment, SnapLearnAI demonstrates how complex document analysis tasks can be simplified and made accessible to end users.

One of the key strengths of the proposed system lies in its unified processing pipeline, which enables seamless handling of multiple file formats through a single interface. Whether the input consists of images, PDF documents, Word files, or plain text, the system ensures consistent processing and analysis quality. This uniformity reduces complexity for users and enhances reliability, making the platform suitable for diverse real-world applications.

The integration of a vision-language model further enhances the system's capabilities by allowing it to interpret both textual and visual content effectively. Through the use of advanced AI models such as Gemini 2.5 Flash, the system can analyze diagrams, extract embedded text from images, and understand complex visual elements. This multimodal capability significantly extends the scope of document analysis beyond traditional text-based systems.

Another important contribution of SnapLearnAI is its conversational interface, which enables users to interact with the system in a natural and intuitive manner. By maintaining a persistent chat history, the system supports multi-turn interactions, allowing users to refine queries and explore documents in greater depth. This conversational approach not only improves usability but also aligns with modern expectations of intelligent assistant systems.

In addition to user-focused features, the system also incorporates administrative functionalities that support efficient system management. The inclusion of an admin dashboard allows for monitoring of user activity, management of system resources, and access to analytical insights, thereby enhancing overall operational control.

XI. FUTURE WORK

The SnapLearnAI system provides a strong foundation for intelligent document analysis; however, there are several promising directions for future enhancement that can further improve its functionality, scalability, and user experience. One important area of development is the implementation of persistent file-based session management. Currently, conversations are tied to active sessions, which limits continuity when files are removed or sessions expire. Enabling users to resume interactions with previously uploaded documents would significantly enhance usability and support long-term analytical workflows.

Another key improvement involves extending support for large files. At present, the system processes documents in memory, which imposes limitations on file size and can impact performance for large datasets. Future versions can incorporate streaming or chunk-based processing techniques, allowing the system to handle extensive documents and high-resolution images more efficiently while maintaining responsiveness. The flexibility of the system can also be enhanced by introducing model selection capabilities. Allowing users to choose between different AI models based on their specific requirements, such as speed, accuracy, or domain specialization, would make the platform more adaptable to diverse use cases. This feature would enable users to optimize performance according to their individual needs. In addition, the introduction of collaborative features represents a significant opportunity for expanding the system's applicability. By enabling shared workspaces, multiple users could interact with the same set of documents, exchange insights, and perform collaborative analysis in real time. Such functionality would be particularly valuable in academic, research, and enterprise environments.

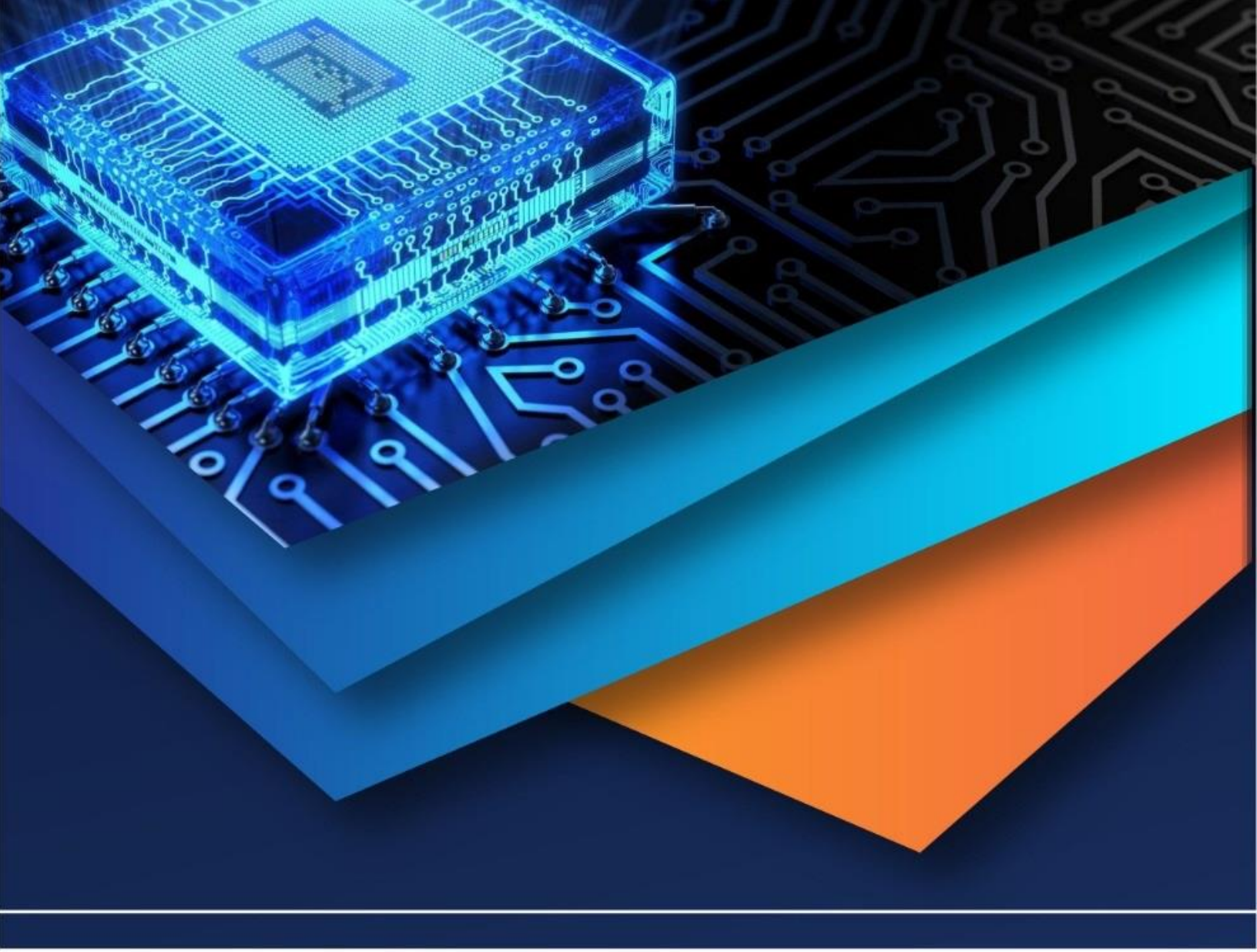
The development of dedicated mobile applications is another important direction for future work. While the current system is accessible through a responsive web interface, native applications for platforms such as Android and iOS would provide improved performance, better integration with device capabilities, and enhanced user convenience. This would allow users to interact with the system more effectively in mobile and on-the-go scenarios.

Finally, the incorporation of offline processing capabilities would address concerns related to privacy and data security. By enabling local AI model inference, the system could process sensitive documents without requiring data transmission to external servers. This approach would be particularly beneficial in domains where data confidentiality is critical, such as healthcare, legal services, and enterprise operations.

REFERENCES

[1] Django Software Foundation, "Django Documentation," 2024. [Online]. Available: https://docs.djangoproject.com/
[2] Google, "Gemini API Documentation," Google AI Studio, 2024. [Online]. Available: https://ai.google.dev/docs
[3] Bytez AI, "Bytez SDK Documentation," 2024. [Online]. Available: https://bytez.ai/docs
[4] K. Zhang, C. Zhu, and J. Liu, "Deep Learning for Document Layout Analysis," IEEE Access, vol. 8, pp. 184792-184804, 2020.
[5] P. Lewis, E. Perez, A. Piktus, et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Advances in Neural Information Processing Systems, vol. 33, pp. 9459-9474, 2020.
[6] V. Karpukhin, B. Oğuz, P. Lewis, et al., "Dense Passage Retrieval for Open-Domain Question Answering," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020.
[7] L. Liu, Z. Lu, and G. Xue, "End-to-End Invoice Understanding with Deep Learning," in Proc. IEEE International Conference on Document Analysis and Recognition, 2019, pp. 407-413.
[8] Anthropic, "Claude 3 Model Card," Anthropic AI, 2024. [Online]. Available: https://www.anthropic.com
[9] OpenAI, "GPT-4V(ision) System Card," OpenAI, 2023. [Online]. Available: https://openai.com
[10] python-docx Contributors, "python-docx Documentation," 2024. [Online]. Available: https://python-docx.readthedocs.io/
[11] PyMuPDF Contributors, "PyMuPDF Documentation," 2024. [Online]. Available: https://pymupdf.readthedocs.io/
[12] Mozilla Developer Network, "Fetch API," MDN Web Docs, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API
[13] W3C, "HTML5 Specification," World Wide Web Consortium, 2024. [Online]. Available: https://html.spec.whatwg.org/
[14] World Wide Web Consortium, "CSS Flexible Box Layout Module Level 1," W3C Recommendation, 2024. [Online]. Available: https://www.w3.org/TR/css-flexbox-1/
[15] NIST, "Privacy and Security Considerations for AI Systems," National Institute of Standards and Technology, AI RMF v1.0, 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ai/

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY