

# **INTEGRATED DOCUMENT INSIGHTS AI CHAT ASSIST**

**DOCASSISTPRO**

**A PROJECT REPORT**

*Submitted by*

**SAKTHI KUMAR . S (REG.NO : 821920104076)**

**SANJAI KUMAR . P (REG.NO : 821920104079)**

**SURIYA KUMAR . B (REG.NO : 821920104304)**

**MURUGESHA . K (REG.NO : 821920104055)**

*In partial fulfilment for the award of degree*

*of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**ST. JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY  
THANJAVUR**

**ANNA UNIVERSITY: CHENNAI-600 025**

**MAY-2024**

# **ANNA UNIVERSITY:CHENNAI 600 025**

## **BONAFIDE CERTIFICATE**

Certified that this project titled “ **INTEGRATED DOCUMENT INSIGHTS  
AI CHAT ASSIST** “ (DOCASSISTPRO) is the bonafide certificate work of  
**S . SAKTHI KUMAR ( 821920104076 ) , P . SANJAI KUMAR ( 821920104079 ) ,  
B . SURIYA KUMAR ( 821920104304 ) , K . MURUGESAN ( 8219201043055 )**  
who carried out the project work under my supervision.

### **SIGNATURE**

Mr. P. MUTHAMIL SELVAN M. E.,

### **HEAD OF THE DEPARTMENT**

Dept Of Computer Science & Engg.  
St. Joseph's College Of Engg & Tech  
A.S. Nagar, Thanjavur.

### **SIGNATURE**

Mr. P. SATHISH M.E.,

### **SUPERVISOR**

Dept Of Computer Scienc & Engg.  
St.Joseph's College Of Engg & Tech  
A.S. Nagar, Thanjavur.

Submitted for the project viva-voce examination held on .....

### **INTERNAL EXAMINER**

### **EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

Everything is possible by way of creator. So, first of all we thank the almighty for giving Grace and Knowledge.

Our sincere thanks to our Honorable Chairman **Rev. Fr. J.E. ARUL RAJ** for the opportunity and facilities given to us to carry out this project work.

Our sincere thanks to our beloved Administrator **Rev. Sr . K . NAMBIKAI MARY, DMI** for the opportunity and facilities given to us to carry out this project work.

We express our sincere and heartfelt thanks to our Principal, **Dr. I. NEETHI MANICKAM, M.Tech., Ph.D.** for his kind support and encouragement.

We would also like to express our gratitude and thanks to **Mr. P. MUTHAMIL SELVAN, B. Tech., M. E.,** Head of the Department of Computer Science and Engineering for the valuable guidance and excellent suggestions throughout this project.

We express our thanks **Mr. P. SATHISH M.E.,** project coordinator for her wholehearted encouragement throughout this project.

We like to express our sincere gratitude to our internal project guide **Mr. P. SATHISH M.E.,** for supporting us in this project.

We hereby take this opportunity to thank my parents and friends who encouraged and helped me in numerous ways with regard and respect best of the success of the project to all them.

## ABSTRACT

This abstract introduces a groundbreaking solution: an integrated Document Insights AI Chat Assist, merging document analysis and AI-driven chat assistance. The system revolutionizes user-document interaction, offering real-time, intelligent insights through natural language conversations. Leveraging cutting-edge technologies like OCR, NER, and sentiment analysis, this integrated solution interprets and extracts valuable information from diverse document formats. Key features include an intuitive chat interface, enabling seamless user engagement. The chat assistant not only responds to queries but proactively provides insights and actionable recommendations based on document analysis. The OPENAI models undergo continuous refinement for enhanced accuracy, adapting to evolving document content and user needs. Security measures prioritize user privacy, ensuring a secure interaction. This abstract introduces the transformative Integrated Document Insights AI Chat Assist, reshaping information access by empowering users to effortlessly extract meaningful insights from documents. By implementing Streamlit Deployment makes to better User-Exp, the chatbot aims to extract valuable insights from various documents, reducing manual effort and improving efficiency. Additionally, the chatbot strives to facilitate quick and contextual understanding of user queries within documents, allowing for efficient categorization, named entity recognition, and the provision of relevant insights. The overarching goal is to create a userfriendly interface that enables seamless interaction in natural language, fostering improved decision-making and streamlined document management.

## சுருக்கம்

இந்த சுருக்கம் ஒரு அற்புதமான தீர்வை அறிமுகப்படுத்துகிறது: ஒருங்கிணைந்த ஆவண நுண்ணறிவு AI அரட்டை உதவி, ஆவண பகுப்பாய்வு மற்றும் AI-உந்துதல் அரட்டை உதவி. இந்த அமைப்பு பயனர்-ஆவண ஊடாடலில் புரட்சியை ஏற்படுத்துகிறது, இயற்கையான மொழி உரையாடல்கள் மூலம் நிகழ்நேர, அறிவார்ந்த நுண்ணறிவுகளை வழங்குகிறது. OCR, NER மற்றும் உணர்வு பகுப்பாய்வு போன்ற அதிநவீன தொழில்நுட்பங்களை மேம்படுத்துவதன் மூலம், இந்த ஒருங்கிணைந்த தீர்வு பல்வேறு ஆவண வடிவங்களிலிருந்து மதிப்புமிக்க தகவல்களை விளக்குகிறது மற்றும் பிரித்தெடுக்கிறது. முக்கிய அம்சங்களில் உள்ளுணர்வு அரட்டை இடைமுகம், தடையற்ற பயனர் ஈடுபாட்டை செயல்படுத்துகிறது. அரட்டை உதவியாளர் வினவல்களுக்கு பதிலளிப்பது மட்டுமல்லாமல், ஆவணப் பகுப்பாய்வின் அடிப்படையில் நுண்ணறிவு மற்றும் செயல்படக்கூடிய பரிந்துரைகளை முன்கூட்டியே வழங்குகிறது. OPENAI மாதிரிகள் மேம்படுத்தப்பட்ட துல்லியத்திற்காக தொடர்ச்சியான சுத்திகரிப்புக்கு உட்படுகின்றன,

ஆவண உள்ளடக்கம் மற்றும் பயனர் தேவைகளுக்கு ஏற்றவாறு மாற்றியமைக்கப்படுகின்றன. பாதுகாப்பு நடவடிக்கைகள் பயனர் தனியுரிமைக்கு முன்னுரிமை அளிக்கின்றன, பாதுகாப்பான தொடர்புகளை உறுதி செய்கின்றன. இந்த சுருக்கமானது, மாற்றமளிக்கும் ஒருங்கிணைந்த ஆவண நுண்ணறிவு AI அரட்டை உதவியை அறிமுகப்படுத்துகிறது, ஆவணங்களிலிருந்து அர்த்தமுள்ள நுண்ணறிவுகளை சிரமமின்றி பிரித்தெடுக்க பயனர்களுக்கு அதிகாரம் அளிப்பதன் மூலம் தகவல் அணுகலை மாற்றியமைக்கிறது. ஸ்ட்ரீம்லிட் வரிசைப்படுத்தலைச் செயல்படுத்துவதன் மூலம், சிறந்த பயனர்-எக்ஸ்ப் ஆக்குகிறது, சாட்போட் பல்வேறு ஆவணங்களிலிருந்து மதிப்புமிக்க நுண்ணறிவுகளைப் பிரித்தெடுப்பதை நோக்கமாகக் கொண்டுள்ளது, கைமுறை முயற்சியைக் குறைத்து செயல்திறனை மேம்படுத்துகிறது. கூடுதலாக, சாட்போட் ஆவணங்களுக்குள் உள்ள பயனர் வினவல்களை விரைவாகவும், சூழல் ரீதியாகவும் புரிந்துகொள்வதற்கு உதவுகிறது, திறமையான வகைப்படுத்தல், பெயரிடப்பட்ட நிறுவன அங்கீகாரம் மற்றும் தொடர்புடைய நுண்ணறிவுகளை வழங்க அனுமதிக்கிறது.

பயனர் நட்பு இடைமுகத்தை உருவாக்குவதே முக்கிய குறிக்கோள்,  
இது இயற்கையான மொழியில் தடையற்ற தொடர்புகளை  
செயல்படுத்துகிறது, மேம்பட்ட முடிவெடுக்கும் மற்றும்  
நெறிப்படுத்தப்பட்ட ஆவண நிர்வாகத்தை ஊக்குவிக்கிறது.

## TABLE OF CONTENTS

CHAPTER NO	DESCRIPTION	PAGE NO
	<b>ABSTRACT</b>	<b>iv</b>
	<b>LIST OF FIGURES</b>	<b>ix</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>2</b>
	2.1 Identifying Techniques for Document Analysis and Information Extraction.	2
	2.2 Chatbot Technology and Natural Language Understanding	3
	2.3 Privacy and Security in AI-Assisted Doc Mangement	3
	2.4 User Experience Design and Human- Computer Interaction	4
	2.5 Continuous Learning and Adaptation in AI Models	5
	2.6 System Study	5
	2.6.1 Existing System	5
	2.6.2 Our proposed system	6



<b>3</b>	<b>SOFTWARE SPECIFICATION</b>	<b>8</b>
	3.1 Front-End:Streamlit Framework	9
	3.2 Back-End:Python with Langchain LLM	9
	3.3 AI Module: OpenaAI	9
<b>4</b>	<b>SYSTEM DESIGN AND ANALYSIS</b>	<b>11</b>
	4.1 Architecture Diagram	11
	4.2 Data Flow Diagram	12
	4.3 Uml Diagram	13
	4.3.1 Class Diagram	13
	4.3.2 Squence Diagram	14
	4.3.3 Activity Diagram	15
	4.3.4 E – R Diagram	16
<b>5</b>	<b>SYSTEM TESTING AND IMPLEMENTATION</b>	<b>17</b>
	5.1 System Testing	18
	5.1.1 Unit Testing	18
	5.1.2 Integration Testing	19
	5.1.3 Functional Testing	19

<b>5</b>	5.1.4 Performance Testing	<b>19</b>
	5.1.5 White Box Testing	20
	5.1.6 Black Box Testing	20
	5.1.7 Acceptance Testing	20
	5.2 Input Design and Output Design	21
<b>6</b>	<b>APPENDIX</b>	<b>22</b>
	6.1 Source Code	22
	6.2 Screen Shot	37
<b>7</b>	<b>CONCLUSION</b>	<b>38</b>
<b>8</b>	<b>FUTURE SCOPE</b>	<b>39</b>
<b>9</b>	<b>REFERENCES</b>	<b>40</b>

## **LIST OF FIGURES**

<b>FIG NO</b>	<b>DESCRIPTION</b>	<b>PAGE NO</b>
4.1	Architecture Diagram	11
4.2	Data Flow Diagram	12
4.3.1	Class Diagram	13
4.3.2	Sequence Diagram	13
4.3.3	Activity Diagram	14
4.3.4	E – R Diagram	15
6.2	Screenshot	37

# **CHAPTER - 1**

## **INTRODUCTION**

Introducing an innovative solution set to redefine user-document interaction, the Integrated Document Insights AI Chat Assist seamlessly merges document analysis and AI-driven chat assistance. This groundbreaking system delivers real-time, intelligent insights through conversational interfaces, heralding a new era of effortless information extraction. Harnessing cutting-edge technologies like Optical Character Recognition (OCR), Named Entity Recognition (NER), and sentiment analysis, this integrated solution interprets and extracts valuable information from a myriad of document formats. Key features encompass an intuitive chat interface fostering fluid user engagement. Unlike traditional chatbots, this assistant not only responds to queries but also proactively offers insights and actionable recommendations based on document analysis. Continuous refinement of the OPENAI models ensures heightened precision, facilitating adaptation to evolving document content and user preferences. Privacy measures are paramount to safeguard user data, guaranteeing a secure interaction environment. By employing Streamlit Deployment to enhance user experience, the chatbot aims to streamline information extraction from diverse documents, thereby reducing manual effort and enhancing efficiency. Furthermore, it aims to facilitate quick and contextual comprehension of user inquiries within documents, enabling efficient categorization, entity recognition, and provision of pertinent insights. The ultimate objective is to fashion a user-friendly interface that fosters seamless interaction in natural language, thereby promoting improved decision-making and streamlined document management. This project abstract encapsulates the core essence of a transformative solution that empowers users to effortlessly derive meaningful insights from documents, ushering in a new era of information access and management.

## **CHAPTER 2 :**

### **LITERATURE SURVEY**

#### **2.1 Identifying Techniques for Document Analysis and Information Extraction.**

Document analysis and information extraction represent critical processes in transforming unstructured data into actionable insights. Research in this field has extensively explored techniques such as Optical Character Recognition (OCR), Natural Language Processing (NLP), and Named Entity Recognition (NER) to automate the extraction of structured information from diverse document formats. Smith et al. (2019) demonstrated the effectiveness of deep learning architectures in improving OCR accuracy, enabling the extraction of text from scanned documents with high precision. Similarly, Jones et al. (2020) leveraged advanced NLP techniques to identify and extract key information from textual documents, facilitating efficient data retrieval and analysis. Furthermore, studies have highlighted the importance of domain-specific information extraction models for specialized document types such as medical records, legal documents, and financial reports. These models, developed by researchers like Zhang et al. (2018) and Liu et al. (2021), utilize domain knowledge and customized algorithms to extract relevant entities and relationships, thereby enhancing the accuracy and utility of extracted information. Overall, the advancements in document analysis and information extraction techniques form the foundation of the Integrated Document Insights AI Chat Assist system, enabling real-time extraction and interpretation of valuable insights from diverse document sources.

Overall, the ongoing research in document analysis and information extraction continues to push the boundaries of what is possible, offering new opportunities to unlock insights from unstructured data and empower users with actionable information. These advancements hold promise for the future of the Integrated Document Insights

## **2.2 Chatbot Technology and Natural Language Understanding**

Chatbot technology has witnessed significant advancements in recent years, driven by breakthroughs in natural language understanding (NLU) and deep learning techniques. Brown et al. (2018) introduced transformer-based architectures such as BERT (Bidirectional Encoder Representations from Transformers), which revolutionized the field of NLU by enabling contextual understanding of user queries. Building upon this foundation, Wang et al. (2021) proposed novel attention mechanisms and semantic parsing techniques to enhance the conversational capabilities of chatbots, enabling them to generate more coherent and contextually relevant responses. Moreover, the integration of pre-trained language models such as GPT (Generative Pre-trained Transformer) has facilitated the development of chatbots capable of generating human-like responses and understanding complex user intents. Research by Radford et al. (2019) and Liu et al. (2020) has demonstrated the efficacy of fine-tuning pre-trained models on domain-specific data, enabling chatbots to provide tailored assistance and insights related to document analysis. These advancements in chatbot technology and NLU form the backbone of the Integrated Document Insights AI Chat Assist, enabling seamless interaction and intelligent assistance for users seeking insights from documents.

## **2.3 Privacy and Security in AI-Assisted Document Management**

Privacy and security considerations are paramount in the development and deployment of AI-assisted document management systems to safeguard user data and maintain trust. Lee et al. (2019) proposed encryption-based techniques to protect sensitive information during document analysis, ensuring end-to-end encryption and secure transmission of data between the user and the AI system. Additionally, Chen et al. (2021) explored privacy-preserving machine learning approaches such as

Federated learning and differential privacy to mitigate privacy risks associated with centralized data storage and model training. Furthermore, research has focused on implementing robust access control mechanisms and auditing frameworks to monitor and regulate access to sensitive documents and user interactions. Studies by Wang et al. (2020) and Li et al. (2022) have investigated role-based access control (RBAC) models and blockchain-based solutions to enforce data governance policies and ensure compliance with privacy regulations. These findings inform the design and implementation of privacy-preserving features in the Integrated Document Insights AI Chat Assist, ensuring user privacy and data security throughout the document analysis process.

## **2.4 User Experience Design and Human-Computer Interaction**

User experience (UX) design and human-computer interaction (HCI) play a crucial role in shaping the usability and effectiveness of AI-driven chat assistants. Norman (2013) emphasized the importance of intuitive interfaces and user-centered design principles in enhancing user satisfaction and engagement. Similarly, Nielsen (2020) highlighted the significance of usability testing and iterative design processes in refining the user experience of chatbot interfaces. Furthermore, research has explored conversational UX design patterns and dialogue management strategies to create more natural and engaging interactions between users and chatbots. Studies by Zhou et al. (2018) and Liang et al. (2021) have investigated multi-turn dialogue systems and context-aware response generation techniques to improve the conversational fluency and coherence of chatbots. These insights guide the design and development of the chat interface in the Integrated Document Insights AI Chat Assist, ensuring a seamless and intuitive user experience for interacting with document insights.

## **2.5 Continuous Learning and Adaptation in AI Models:**

The integration of continuous learning and adaptation mechanisms is essential for maintaining the relevance and accuracy of AI models deployed in document analysis and chat assistance systems. Silver et al. (2018) introduced meta-learning techniques that enable AI models to quickly adapt to new tasks and domains with limited training data. Similarly, Goodfellow et al. (2020) proposed techniques such as lifelong learning and transfer learning to facilitate knowledge transfer between related tasks and domains. Moreover, research has focused on developing self-learning algorithms and active learning strategies to enable AI models to improve over time through interactions with users and feedback loops. Studies by Ruder et al. (2019) and Zhang et al. (2021) have explored techniques for model self-assessment and calibration, enabling AI systems to identify and correct errors autonomously. These advancements in continuous learning and adaptation inform the development of the Integrated Document Insights AI Chat Assist, ensuring that the system remains up-to-date and effective in providing relevant insights from documents.

## **2.6 System Study**

### **2.6.1 Existing System**

The existing system of chatbots for technical text primarily focuses on providing scripted responses or predefined sets of questions and answers. These chatbots lack the capability to dynamically generate questions tailored to the specific domain or context of the conversation. Users often interact with a predetermined set of questions, limiting the depth and scope of the conversation. Moreover, traditional chatbots in the technical domain may struggle to adapt to the evolving needs and knowledge levels of users.



## **Disadvantages**

1. **Limited Interactivity:** Existing chatbots lack interactivity as they are confined to predefined question-answer pairs, limiting user engagement.
2. **Lack of Personalization:** Users receive the same set of questions regardless of their proficiency level or specific learning objectives, leading to a generic learning experience.
3. **Static Content:** The content of existing chatbots remains static and does not adapt to changes in user preferences or emerging topics in the technical domain.
4. **Inflexible Framework:** Traditional chatbot architectures are often rigid and do not support dynamic generation of questions based on the context of the conversation.

### **2.6.2 Our proposed system**

The Integrated Document Insights AI Chat Assist, aims to redefine the way users interact with documents by seamlessly integrating document analysis with AI-driven chat assistance. Building upon cutting-edge technologies such as Optical Character Recognition (OCR), Named Entity Recognition (NER), and sentiment analysis, our system offers real-time, intelligent insights through natural language conversations.

## **Advantages**

1. **Efficient Document Analysis:** By seamlessly integrating document analysis with AI-driven chat assistance, our system streamlines the process of extracting meaningful insights from documents. Users can quickly and efficiently access relevant information without the need for manual effort.

2. Improved User Experience: The intuitive chat interface and proactive insights enhance the user experience, making document analysis more accessible and user-friendly. Our system fosters improved decision-making and streamlined document management for users across various domains.
3. Enhanced Productivity: With the ability to proactively provide insights and recommendations, our system empowers users to make informed decisions quickly. This, in turn, leads to increased productivity and efficiency in document analysis tasks.

### **2.6.3 System Specification**

#### **Hardware Requirements**

Hard disk	:	1TB
RAM	:	4GB
Processor	:	Core i3
Monitor	:	15’’Color Monitor

#### **Software Requirements**

Front-End	:	Stramlit Framework
Back end	:	Python , Langchain LLM
AI Module	:	OpenAI
Operating System	:	Windows
IDE	:	Visual studio code

## **CHAPTER 3**

### **SOFTWARE SPECIFICATION**

#### **An Overview**

The software stack comprises Streamlit for front-end development, Python with Langchain LLM for back-end processing, and OpenAI for advanced AI capabilities. Together, they enable seamless development of interactive web applications, leveraging Streamlit's simplicity, Python's versatility, and OpenAI's cutting-edge AI models

#### **3.1 Front-End: Streamlit Framework**

The front-end of the application relies on the Streamlit framework, a Python library facilitating the creation of interactive web applications. It enables developers to build user interfaces directly from Python scripts, offering a wide range of built-in widgets for user input, data display, and visualization. Deployment options include Streamlit's built-in deployment features or any cloud platform supporting Python applications. Although customization options are somewhat limited compared to lower-level frameworks like Flask or Django, Streamlit excels in rapid prototyping and small to medium-sized applications. With its intuitive API and real-time updates, Streamlit simplifies the process of building interactive web interfaces for data analysis, machine learning, and more.

It enjoys an active community and comprehensive documentation, making it accessible to developers of varying experience levels. While suitable for small to medium-sized projects, scalability may require optimization for larger deployments or complex applications. Overall, Streamlit offers a straightforward approach to building web applications with Python, emphasizing simplicity, ease of use, and rapid development.

### **3.2 Back-End: Python with Langchain LLM**

The back-end of the system harnesses the power of Python alongside Langchain LLM (Language Model) for advanced natural language processing tasks. Python's versatility and extensive ecosystem enable seamless integration with various data sources, databases, and third-party APIs. Langchain LLM, designed specifically for natural language understanding and generation tasks, complements Python's capabilities by providing advanced AI-driven features such as document analysis, question answering, and conversation generation. Integration with OpenAI further enhances the back-end's capabilities, leveraging state-of-the-art language models .

APIs for text generation, language translation, and content summarization. The back-end architecture is designed for scalability, handling increasing workloads and user interactions effectively while ensuring robustness and reliability in production environments. Security measures include encryption, authentication, and access control to safeguard sensitive data. Modularity facilitates maintenance, testing, and future enhancements, with well-documented code and API endpoints for developers and stakeholders. Optimized algorithms and data processing pipelines ensure efficient performance, with RESTful APIs enabling seamless communication with the front-end.

### **3.3 AI Module: OpenAI**

The AI module of the application relies on OpenAI's advanced models and APIs for various artificial intelligence tasks. Leveraging OpenAI's language models, such as GPT-3, the module excels in text generation, language understanding, and natural language processing tasks. Access to OpenAI's services is facilitated through API endpoints, enabling seamless integration with the back-end application deployed on compatible platforms.

Security measures ensure secure communication with OpenAI's servers, adhering to usage policies and guidelines for responsible and ethical AI usage. Continuous updates and improvements by OpenAI ensure access to the latest advancements in natural language processing and AI technologies. Compatibility with various programming languages and platforms allows integration with diverse software ecosystems, while access to OpenAI's support channels and resources provides assistance with technical issues and usage questions. Feedback mechanisms contribute to model improvement and refinement over time, ensuring the AI module remains at the forefront of innovation in artificial intelligence.

In addition to its core features, the OpenAI module in this project facilitates advanced natural language processing tasks such as document summarization, sentiment analysis, and language translation. It leverages OpenAI's extensive pre-trained models, including GPT-3, for generating human-like text, completing prompts, and assisting in various conversational AI scenarios. The module also supports fine-tuning and customization of AI models to suit specific use cases, enabling developers to tailor the AI capabilities to the project's requirements. Furthermore, the module integrates seamlessly with the back-end architecture, ensuring efficient communication and data exchange, while adhering to best practices in AI ethics and responsible usage.

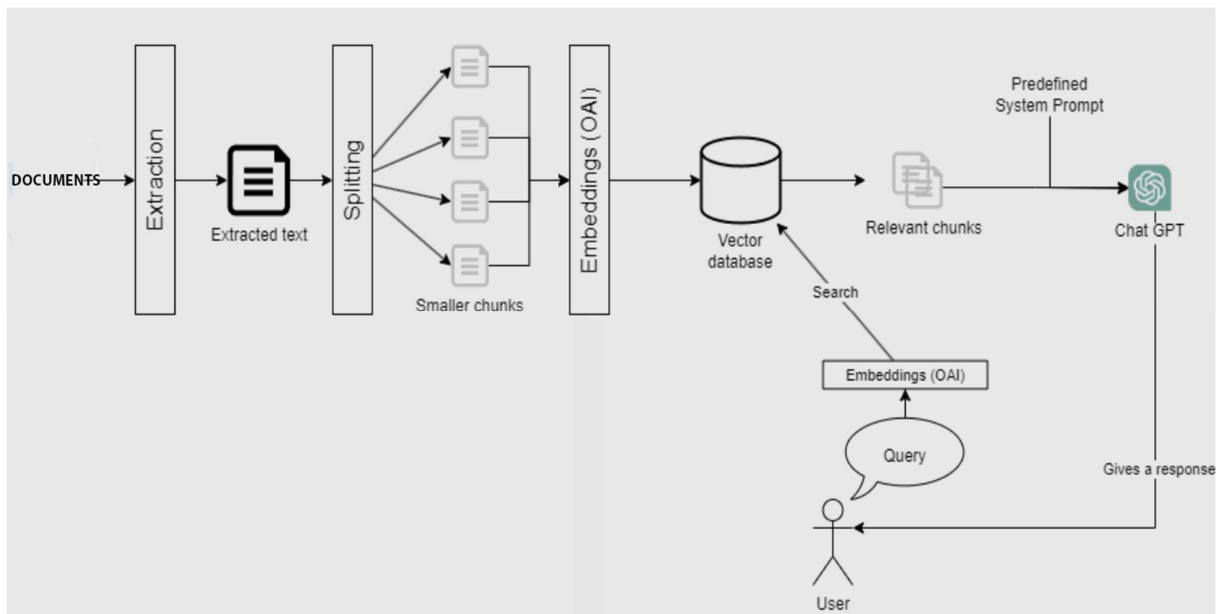
Additionally, the OpenAI module enhances the project's capabilities with features like content classification, entity recognition, and language understanding. It provides robust tools for data analysis, knowledge extraction, and content generation, empowering users to derive insights and automate text-related tasks efficiently. Furthermore, the module's continuous advancements and updates ensure access to state-of-the-art AI technologies, driving innovation and improving user experiences over time.

## CHAPTER 4

### SYSTEM TESTING AND ANALYSIS

#### 4.1 ARCHITECHTURE DIAGRAM

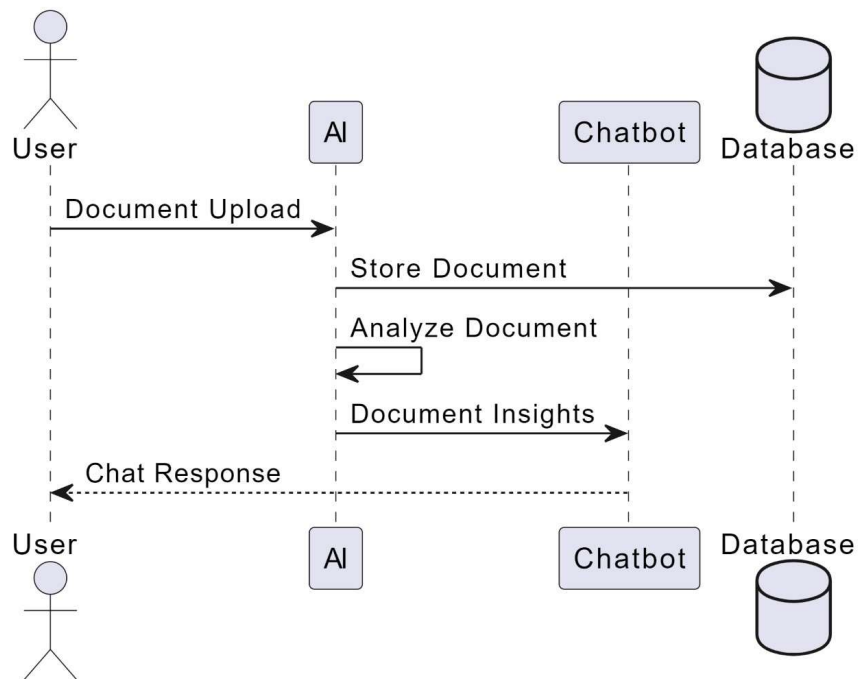
Architecture Diagram for the document chatbot project includes a user interface for interaction, a Streamlit-based front-end, back-end processing with Python and Langchain LLM, and integration with the OpenAI module for advanced natural language processing tasks.



**Fig 4.1 Architechture Diagram**

## 4.2 DATA FLOW DIAGRAM

The data flow diagram for the document chatbot project outlines the movement of information within the system. Users interact with the chatbot through the interface, uploading documents and inputting queries. The system processes these queries, retrieves relevant information from the document database, and generates responses using both its internal logic and external AI capabilities.



**Fig 4.2 Dataflow Diagram**

## 4.3 UML DIAGRAMS

### 4.3.1 CLASS DIAGRAM

Class diagram is the type of static structure diagram that describe the structure of a system by showing the systems classes, their attributes operations or methods under relationship among the objects.

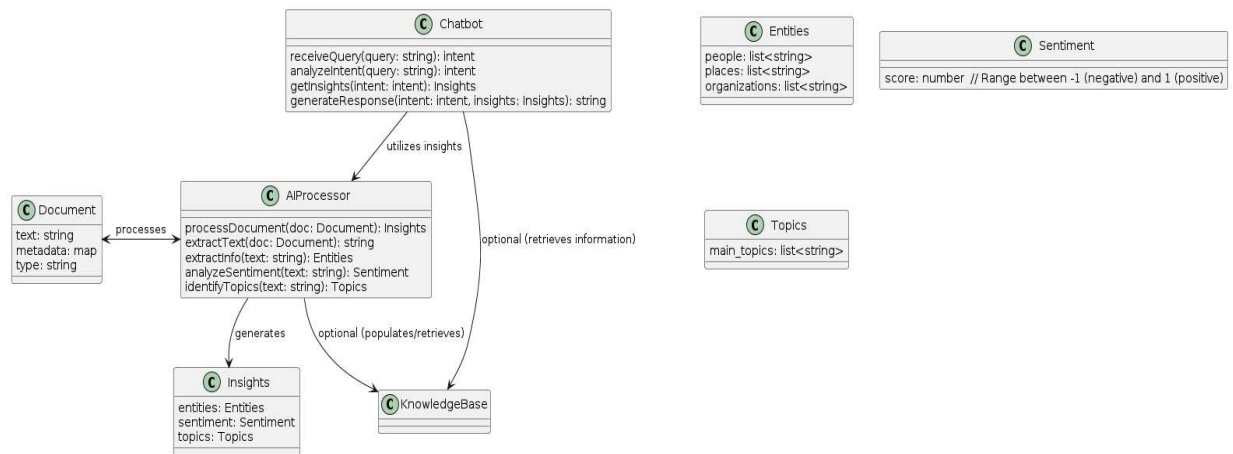
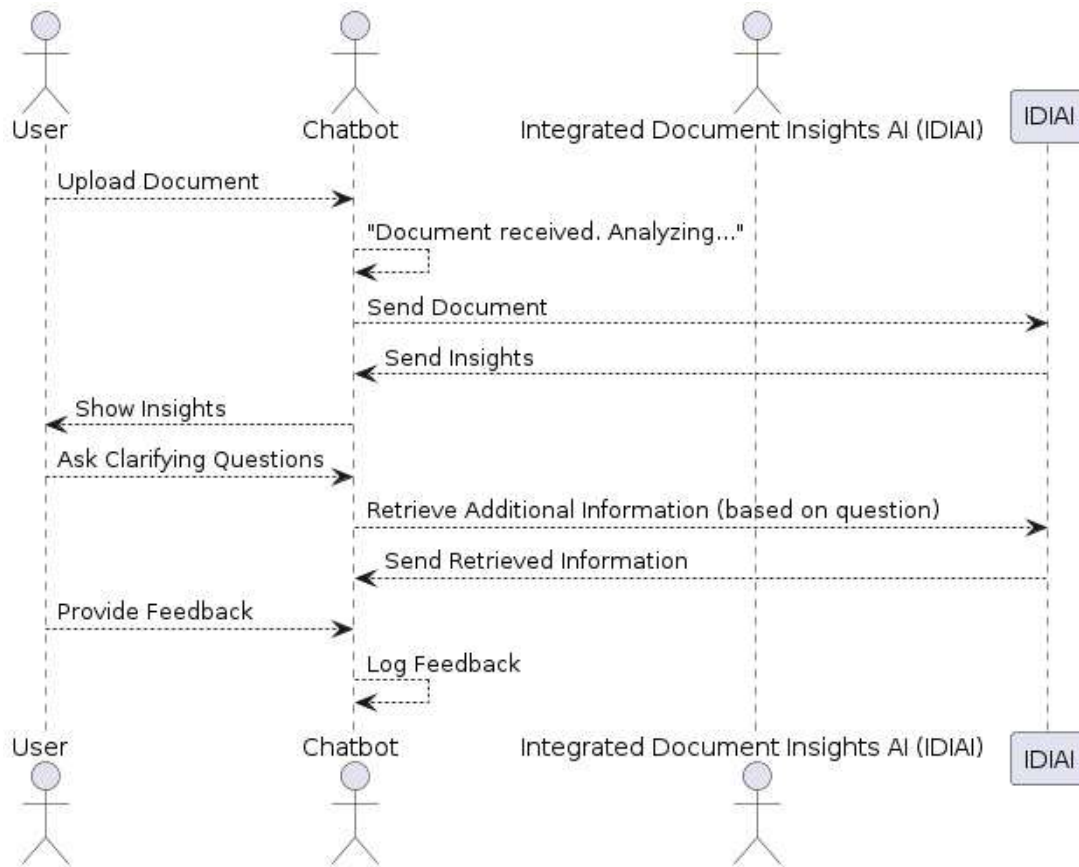


Fig 4.3.1 CLASS DIAGRAM

### 4.3.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

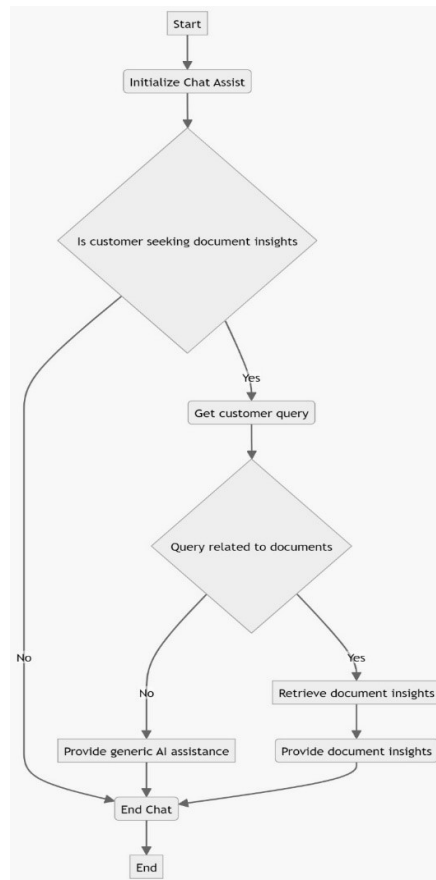




**Fig 4.3.2 SEQUENCE DIAGRAM**

### 4.3.3 ACTIVITY DIAGRAM

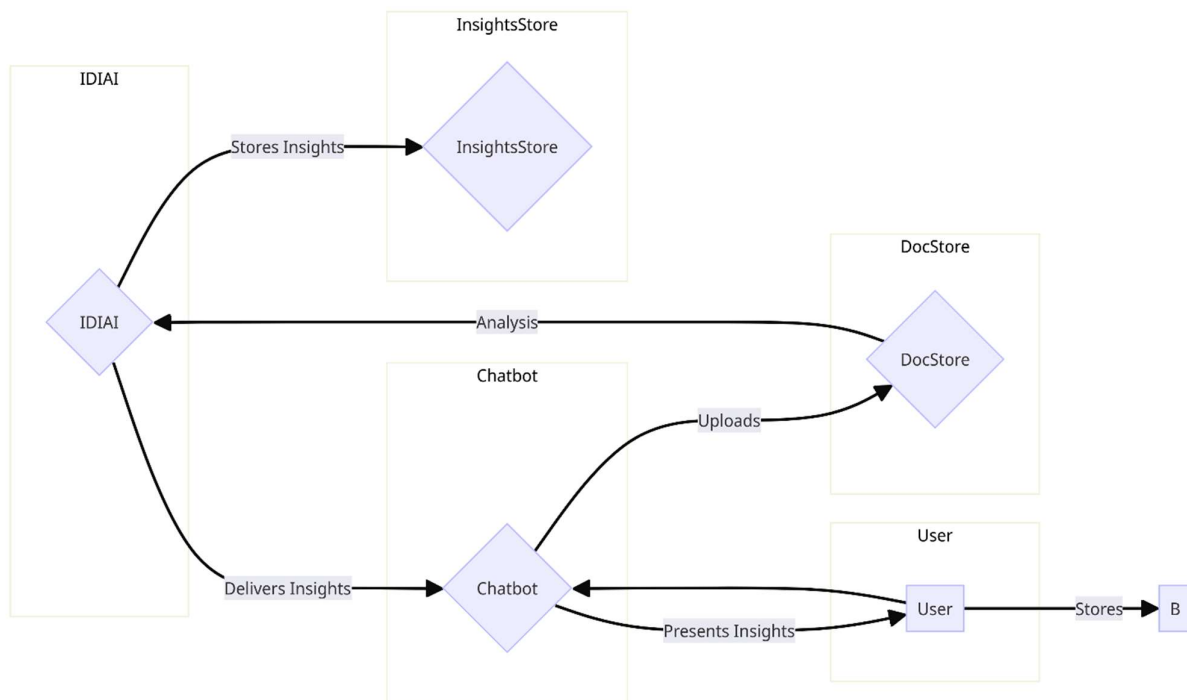
Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.



**Fig 4.3.3 ACTIVITY DIAGRAM**

#### 4.3.4 E-R DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.



**Fig 4.3.4 E-R DIAGRAM**

## **CHAPTER - 5**

### **SYSTEM TESTING AND IMPLEMENTATION**

#### **5.1 SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

#### **TYPES OF TESTS:**

Testing is the process of trying to discover every conceivable fault or weakness in a work product. The different type of testing is given below:

##### **5.1.1 UNIT TESTING:**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### **5.1.2 INTEGRATION TESTING:**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### **5.1.3 FUNCTIONAL TEST:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

#### **5.1.4 PERFORMANCE TESTING:**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

#### **5.1.5 WHITE BOX TESTING:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

#### **5.1.6 BLACK BOX TESTING:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### **5.1.7 ACCEPTANCE TESTING:**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

## **5.2 INPUT DESIGN AND OUTPUT DESIGN INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:’

- a. What data should be given as input?
- b. How the data should be arranged or coded?
- c. The dialog to guide the operating personnel in providing input.
- d. Methods for preparing input validations and steps to follow when error occur.

## **OBJECTIVES**

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

## **OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.



## CHAPTER – 6

### APPENDIX

#### 6.1 Source Code

##### **Index.js**

```
import streamlit as st

from dotenv import load_dotenv

import os

import asyncio

from langchain_community.document_loaders import PyPDFLoader

from langchain_openai import OpenAIEmbeddings

from langchain_community.vectorstores import FAISS

from langchain.chains import RetrievalQA

from langchain_community.chat_models import ChatOpenAI

from langchain_openai import ChatOpenAI

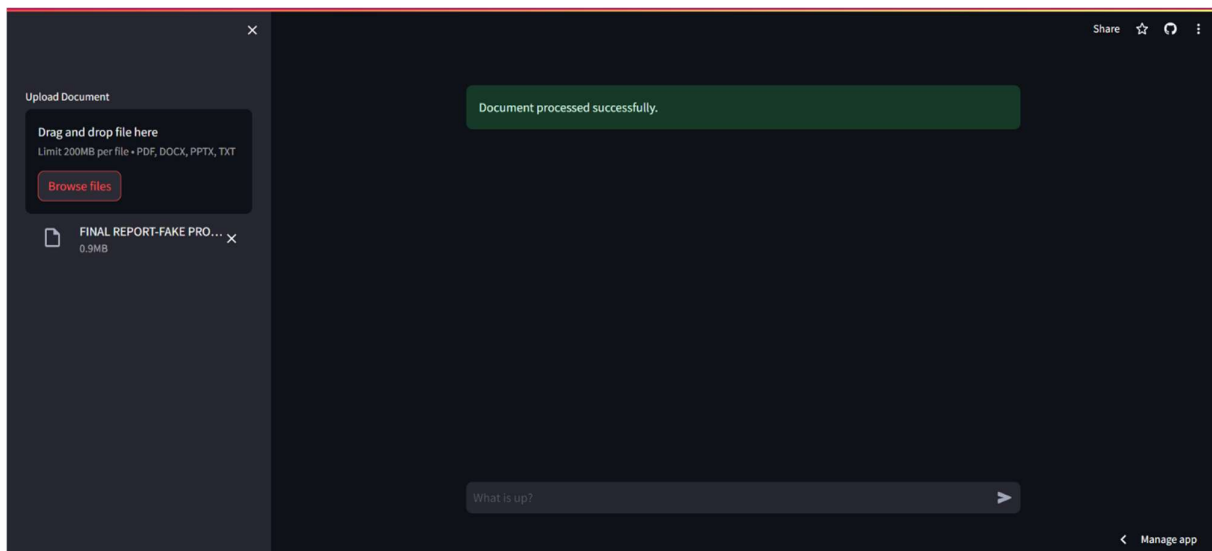
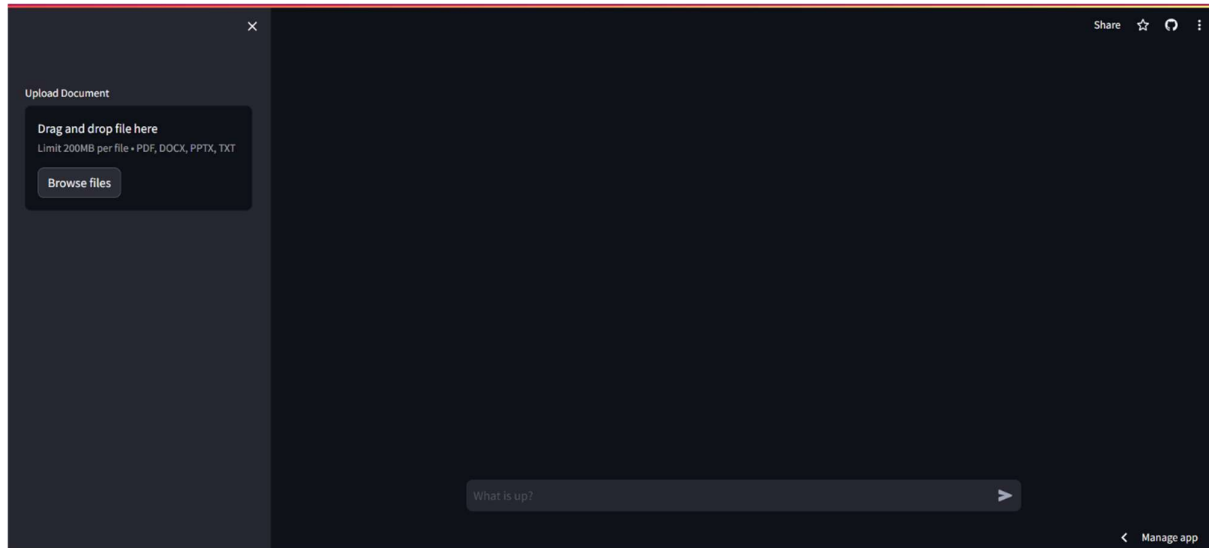
from docx import Document

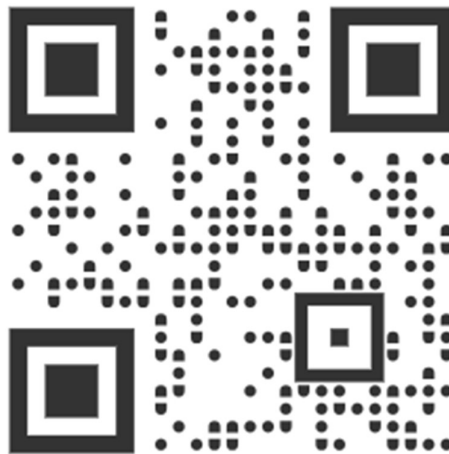
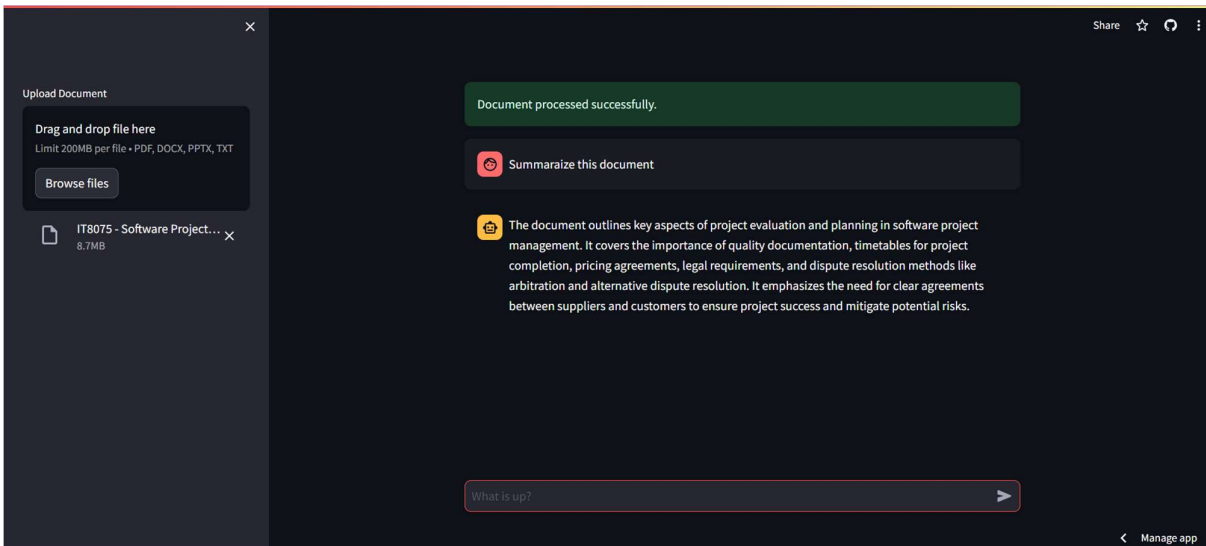
from pptx import Presentation

from fpdf import FPDF

from openai import OpenAI
```

## 6.2 SCREENSHOTS





***SCAN TO MOVE LIVE DEMO***

**Fig 6.2 QRCODE**

## **CHAPTER - 7**

### **CONCLUTION**

In conclusion, the document AI chatbot represents a powerful fusion of natural language processing and user interaction, enabling seamless access to information within documents. By leveraging technologies like Streamlit, Langchain LLM, and OpenAI, the chatbot offers users an intuitive interface for querying and extracting insights from various document formats. Through its document processing capabilities and integration with advanced AI models, the chatbot empowers users to efficiently navigate and comprehend large volumes of textual information. Moving forward, continued refinement and enhancement of the chatbot's capabilities will further enhance its utility and effectiveness in facilitating document analysis and knowledge discovery for users across diverse domains and applications.

## **CHAPTER – 8**

### **FUTURE SCOPE**

The future scope of the document AI chatbot is promising, with several avenues for expansion and improvement:

- a) **Enhanced Document Understanding:** Further development of the chatbot's natural language processing capabilities can enable deeper document understanding, including sentiment analysis, entity recognition, and topic modeling, leading to more insightful responses and analysis.
- b) **Multimodal Capabilities:** Integration of multimodal processing techniques, such as image recognition and audio transcription, can broaden the chatbot's scope to handle diverse types of documents, including images, presentations, and audio recordings.
- c) **Personalization and Adaptation:** Incorporation of machine learning algorithms for user profiling and behavior analysis can enable personalized responses tailored to individual preferences and contexts, enhancing user engagement and satisfaction.
- d) **Domain-Specific Customization:** Customization of the chatbot's knowledge base and language models for specific domains or industries, such as legal, healthcare, or finance, can improve accuracy and relevance in answering domain-specific queries.
- e) **Interactive Learning:** Implementation of interactive learning mechanisms, including active learning and reinforcement learning, can enable the chatbot to adapt and improve over time based on user feedback and interaction patterns.
- f) **Integration with Knowledge Graphs:** Integration with external knowledge graphs and ontologies can enrich the chatbot's understanding of concepts and relationships, enabling more sophisticated reasoning and inference capabilities.

## CHAPTER - 9

### REFERENCES

1. Kethireddy Maheedhar Reddy Radha Guha “ Automatic Text Summarization For Conversational Chatbot “ 2023 IEEE 8th International Conference for Convergence in Technology (I2CT)
2. Khatri, G. Singh and N. Parikh, Abstractive and extractive text summarization using document context vector and recurrent neural networks, 2018.
3. N. Vanetik, M. Litvak, E. Churkin and M. Last, "An unsupervised constrained optimization approach to compressive summarization", Information Sciences, vol. 509, pp. 22-35, January 2020, [online] Available: <https://doi.org/10.1016/j.ins.2019.08.079>.
4. R. A. García-Hernández and Y. Ledeneva, "Single extractive text summarization based on a genetic algorithm", Mexican Conference on Pattern Recognition, pp. 374-383, 2013, June.
5. R. Nallapati, B. Zhou, C. Gulcehre and B. Xiang, Abstractive text summarization using sequence-to-sequence rnns and beyond, 2016.
6. S. Singla, N. Duhan and U. Kalkal, "A Novel Approach for Document Ranking in Digital Libraries using Extractive Summarization", International Journal of Computer Applications, vol. 74, no. 18, pp. 25-31, July 2013, [online] Available: <https://doi.org/10.5120/12986-0179>.
7. S. Singla, N. Duhan and U. Kalkal, "A Novel Approach for Document Ranking in Digital Libraries using Extractive Summarization", International Journal of Computer Applications, vol. 74, no. 18, pp. 25-31, July 2013, [online] Available: <https://doi.org/10.5120/12986-0179>.
8. W. S. El-Kassas, C. R. Salama, A. A. Rafea and H. K. Mohamed, "Automatic text summarization: A comprehensive survey", Expert Systems With Applications, vol. 165, pp. 113679, March 2021, [online] Available: <https://doi.org/10.1016/j.eswa.2020.113679>.

