DEPARTMENT OF INFORMATION TECHNOLOGY

SHRI SHANKARACHARYA TECHNICAL CAMPUS

JUNWANI BHILAI (C.G.) 490020

Session: 2023 - 2024

A

Project Report

On

COGNITIVE DOCUMENT LLM AGGREGTOR

Submitted to

CHATTISGARH SWAMI VIVEKANAND TECHNICAL UNIVERSITY

BHILAI (INDIA)

In partial fulfillment of requirement for the award of degree of

Bachelor of Technology

In

INFORMATION TECHNOLOGY

by

AMAN UMRAO

UNDER THE GUIDANCE OF

Mr. GOVIND SINGH

DECLARATION

We the undersigned solemnly declare that the report of the project work entitled "**Cognitive Document LLM Aggregator**" is based on our own work carried out during the course of our study under the supervision of **Mr. GOVIND SINGH**.

We assert that the statements made and the conclusions drawn are an outcome of the project work. We further declare that to the best of our knowledge and belief that the report does not contain any part of any work which has been submitted for the award of any other degree/diploma/certificate in this university or any other university.

(Signature of candidate)

Aman Umrao

Roll No:-301403321021

Enrollment No:-CA8604

CERTIFICATE

This is to certify that the report of the project submitted is an outcome of the project work entitled ” Cognitive Document LLM Aggregator ” carried out by

AMAN UMRAO Roll No: 301403321021 Enrollment No:CA8604

under my guidance and supervision for the award of Degree in Bachelor of Technology in **INFORMATION TECHNOLOGY** of Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G.), India.

To the best of my knowledge the report

i) Embodies the work of the candidate him/herself,

ii) Has duly been completed,

iii) Fulfills the requirement of the Ordinance relating to the BE degree of the University and

iv) Is up to the desired standard for the purpose of which is submitted.

(Signature of the Guide)
Mr. Govind Singh
Professor
Department of Information Technology

(Head of Department)
Dr. Avinash Dhole
Department of Information Technology

Principal

Shri Shankaracharya Technical Campus, Junwani, Bhilai.

CERTIFICATE BY THE EXAMINERS

This is to certify that the project work entitled

“*COGNITIVE DOCUMENT LLM AGGREGATOR*”

Submitted by

AMAN UMRAO Roll No: 301403321021 Enrollment No:CA8604

has been examined by the undersigned as a part of the examination for the award of Bachelor of Technology degree in Information Technology of Chhattisgarh Swami Vivekanand Technical University, Bhilai.

Internal Examiner
Date:

External Examiner
Date:

ACKNOWLEDGEMENT

I have great pleasure in the submission of this project report entitled **COGNITIVE DOCUMENT LLM AGGREGATOR** in partial fulfilment the degree of B.Tech. **Information Technology**. While Submitting this Project report, I take this opportunity to thank those directly or indirectly related to project work. I would like to thank my guide **Mr. Govind Singh** of the guide in Company who has provided the opportunity and organizing project for me. Without his active co-operation and guidance, it would have become very difficult to complete task in time.

I would like to express sincere thanks and gratitude **to Director of the College (P.B. Deshmukh**), **Principal Jaspal Bagga** , **Head of Department Dr. Avinash Dhole , (Information Technology).**

While Submission of the project, I also like to thanks to Project Coordinator Govind Sir, and the staff of Name of the College for their continuous help and guidance throughout the course of project. Acknowledgement is due to our parents, family members, friends and all those persons who have helped us directly or indirectly in the successful completion of the project work.

AMAN UMRAO

Table of Contents

ABSTRACT

The Cognitive Document LLM Aggregator is a web application designed to enhance document management and information retrieval through advanced natural language processing. By leveraging large language models (LLMs), the application allows users to upload various document types and process them to answer context-specific queries. The system provides capabilities for document summarization, advanced search, and multi-document aggregation, ensuring comprehensive and accurate responses. This innovative tool streamlines the extraction of valuable insights from documents, making it a powerful resource for individuals and organizations seeking efficient document analysis and contextual information retrieval.

Chapter 1

INTRODUCTION

1.1. Background of study

In today's information-driven world, efficiently managing and extracting insights from vast amounts of documents is crucial for individuals and organizations alike. Traditional methods of document analysis and information retrieval are often time-consuming and require significant manual effort. To address these challenges, we present the Cognitive Document LLM Aggregator, a cutting-edge web application designed to revolutionize the way users interact with their documents. Developed using Python and Streamlit, the Cognitive Document LLM Aggregator harnesses the power of the latest Gemini Pro model to deliver advanced natural language processing capabilities. Users can effortlessly upload various document types, including PDFs, Word files, and text files, into the application. Once uploaded, the system processes these documents to answer user queries, providing context-specific information with unparalleled accuracy. Key features of the application include document summarization, advanced search functionalities, and multi-document aggregation. These features enable users to quickly obtain comprehensive insights, streamline their workflow, and make informed decisions based on the analyzed data. By leveraging the state-of-the-art Gemini Pro model, the application ensures that the responses are not only relevant but also contextually accurate. The Cognitive Document LLM Aggregator is designed to be an invaluable resource for anyone needing efficient document analysis and information retrieval. Whether for academic research, business intelligence, or personal use, this application offers a robust solution to manage and derive meaningful insights from large collections of documents.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

We present the Cognitive Document LLM Aggregator, a cutting-edge web application designed to revolutionize the way users interact with their documents. Developed using Python and Streamlit, the Cognitive Document LLM Aggregator harnesses the power of the latest Gemini Pro model to deliver advanced natural language processing capabilities. Users can effortlessly upload various document types, including PDFs, Word files, and text files, into the application. Once uploaded, the system processes these documents to answer user queries, providing context-specific information with unparalleled accuracy. Key features of the application include document summarization, advanced search functionalities, and multi-document aggregation. These features enable users to quickly obtain comprehensive insights, streamline their workflow, and make informed decisions based on the analyzed data. By leveraging the state-of-the-art Gemini Pro model, the application ensures that the responses are not only relevant but also contextually accurate. The Cognitive Document LLM Aggregator is designed to be an invaluable resource for anyone needing efficient document analysis and information retrieval. Whether for academic research, business intelligence, or personal use, this application offers a robust solution to manage and derive meaningful insights from large collections of documents.

2.2 Literature Review

The field of document management and information retrieval has undergone significant advancements with the advent of artificial intelligence (AI) and natural language processing (NLP). Traditional methods often relied on keyword-based searches and manual document organization, which were both time-consuming and prone to human error. The integration of NLP techniques has revolutionized these processes, enabling more sophisticated and efficient document analysis.

Natural Language Processing in Document Analysis

Natural language processing has been extensively studied and applied to various aspects of document management. Early works, such as those by Manning and Schütze (1999), laid the foundation for understanding and processing human language using computational methods. More recent advancements, particularly in transformer-based models like BERT (Devlin et al., 2018) and GPT (Radford et al., 2019), have significantly improved the capabilities of NLP systems in understanding and generating human-like text.

Large Language Models (LLMs) and Their Applications

The introduction of large language models (LLMs) has marked a turning point in NLP. Models like GPT-3 (Brown et al., 2020) and its successors have demonstrated unprecedented abilities in text generation, summarization, and contextual understanding. These models are trained on diverse and extensive datasets, allowing them to capture nuanced meanings and provide more accurate and contextually relevant responses.

Document Summarization and Query Answering

Document summarization is a critical feature for efficient information retrieval. Techniques such as extractive summarization (Nallapati et al., 2016) and abstractive summarization (See et al., 2017) have been developed to generate concise summaries of lengthy documents. Query answering systems, leveraging the capabilities of LLMs, can provide precise answers to user queries by understanding the context within the documents (Rajpurkar et al., 2016; Yang et al., 2019).

Streamlit and User-Friendly NLP Applications

Streamlit, an open-source Python library, has emerged as a popular tool for developing interactive and user-friendly web applications. Its simplicity and integration capabilities make it an ideal choice for deploying NLP models and creating accessible interfaces for end-users (Mayer et al., 2020).

Gemini Pro Model

The latest Gemini Pro model represents a significant advancement in the field of LLMs, offering enhanced performance in various NLP tasks. Its architecture and training methodology have been optimized to deliver superior accuracy and contextual understanding, making it a powerful tool for document analysis and query answering.

2.3 Conclusion

The Cognitive Document LLM Aggregator builds on these advancements by integrating the Gemini Pro model with Streamlit to create a powerful and user-friendly application for document management and information retrieval. By leveraging state-of-the-art NLP techniques, this project addresses the limitations of traditional methods and provides an efficient solution for extracting valuable insights from documents.

Chapter 3

PROBLEM IDENTIFICATION

In the digital age, the sheer volume of documents generated and stored by individuals and organizations poses significant challenges in terms of management, retrieval, and analysis. Traditional methods of handling documents often fall short in several critical areas, leading to inefficiencies and missed opportunities for leveraging valuable information. The primary problems can be identified as follows**:**

1. **Inefficient Document Management**:
   - Manual Sorting and Organization: Traditional document management systems require extensive manual effort to categorize, sort, and organize documents. This process is not only time-consuming but also prone to errors and inconsistencies.
   - Difficulty in Accessing Relevant Information: With large volumes of documents, finding specific information quickly becomes a daunting task. Users often have to sift through numerous files, leading to significant time wastage.
2. **Limited Information Retrieval Capabilities**:
   - Keyword-Based Searches: Most conventional systems rely on simple keyword searches, which can be inadequate for understanding the context and nuances of user queries. This often results in irrelevant or incomplete search results.
   - Lack of Contextual Understanding: Traditional search engines and document management systems struggle to interpret the context of a query relative to the content of the documents, leading to less accurate and meaningful responses.
3. **Inadequate Summarization and Insight Extraction**:
   - Manual Summarization: Extracting key points and summarizing documents manually is labor-intensive and not scalable. This limits the ability to quickly grasp the essential information from lengthy documents.
   - Missed Insights: Without advanced summarization tools, important insights and trends within the documents may be overlooked, reducing the overall value extracted from the data.
4. **Scalability Issues**:
   - **Handling Large Volumes of Data**: As the volume of documents grows, traditional systems face scalability challenges, resulting in slower performance and decreased efficiency.

    - **Aggregation of Information**: Combining insights from multiple documents to answer complex queries is often beyond the capabilities of conventional document management systems.

5. **User Experience and Accessibility**:
    - **Complex Interfaces**: Many document management systems are not user-friendly, requiring extensive training and technical knowledge to operate effectively.
    - **Limited Interactivity**: The lack of interactive features and real-time processing capabilities hampers the user experience, making it difficult to quickly obtain and utilize information.

Addressing the Problems

The Cognitive Document LLM Aggregator aims to address these challenges by leveraging the latest advancements in natural language processing and machine learning. By integrating the Gemini Pro model with a user-friendly Streamlit interface, the application provides:

- **Automated and accurate document categorization and organization.**
- **Context-aware query processing to deliver precise and relevant answers.**
- **Advanced summarization techniques to quickly distill key points from documents.**
- **Scalable solutions for managing and retrieving information from large datasets.**
- **An intuitive interface that enhances user experience and accessibility**.

Through these capabilities, the Cognitive Document LLM Aggregator offers a comprehensive solution to the inefficiencies and limitations of traditional document management systems, enabling users to efficiently manage, retrieve, and analyze their documents with ease and precision.

CHAPTER 4
METHODOLOGY

4.1 Introduction

The Cognitive Document LLM Aggregator is designed to enhance document management and information retrieval by leveraging state-of-the-art natural language processing (NLP) techniques and large language models (LLMs). The following methodology outlines the systematic approach taken to develop, implement, and evaluate this application.

4.2 Modules

There are three modules used in this project:

- **Single Page App**

Text Extraction and Tokenization

- Extracted text is tokenized to prepare it for further analysis. Tokenization splits the text into meaningful units (tokens) such as words or sentences.

Embedding Generation

- The tokenized text is converted into embeddings using the Gemini Pro model. These embeddings represent the semantic meaning of the text, enabling the model to understand context and relationships between different parts of the document.

Contextual Query Understanding

- User queries are processed using the Gemini Pro model to understand their intent and context.
- The model leverages its extensive training on diverse datasets to interpret the nuances of the query accurately.

Document Matching and Relevance Scoring

- The system compares the query embeddings with the document embeddings to identify relevant documents.
- A relevance score is calculated for each document based on its alignment with the query, ensuring the most pertinent information is retrieved.

Extractive Summarization

- The application employs extractive summarization techniques to identify and extract key sentences or paragraphs from the documents that are most relevant to the query.

Abstractive Summarization

- For more comprehensive insights, the system uses abstractive summarization to generate concise summaries that capture the essence of the documents in a more natural language form.

Streamlit Integration

- The user interface is built using Streamlit, a Python library that facilitates the creation of interactive and user-friendly web applications.
- Streamlit allows for real-time interaction and immediate feedback, improving the overall user experience.

4.3 Proposed Solution

The Cognitive Document LLM Aggregator aims to address the challenges of traditional document management and information retrieval systems by leveraging advanced natural language processing (NLP) and large language models (LLMs). The proposed solution involves a comprehensive approach that combines cutting-edge technology with user-friendly design to deliver an efficient, accurate, and scalable document analysis tool.

1. **Document Upload and Management**
    - **User-Friendly Interface**: A web-based interface developed using Streamlit allows users to easily upload various types of documents (PDFs, Word files, text files) into the system.
    - **Secure Storage**: Uploaded documents are securely stored in a database, ensuring data integrity and privacy.
2. **Advanced Natural Language Processing**
    - **Text Extraction and Preprocessing**: The system extracts text from uploaded documents and preprocesses it to remove noise and irrelevant formatting.

- **Tokenization and Embedding**: Using the Gemini Pro model, the text is tokenized and converted into semantic embeddings, facilitating accurate understanding and analysis of the document content.

3. **Contextual Query Processing**
   - **Context-Aware Interpretation**: The system leverages the Gemini Pro model to interpret user queries within the context of the documents, ensuring accurate and relevant responses.
   - **Relevance Scoring**: Queries are matched against document content using embedding comparison, and relevance scores are calculated to prioritize the most pertinent information.
4. **Multi-Document Aggregation**
   - **Cross-Document Analysis**: The application can aggregate information from multiple documents, providing holistic responses to complex queries.
   - **Insight Synthesis**: By analyzing data across multiple sources, the system synthesizes insights to ensure completeness and accuracy.
5. **Interactive User Experience**
   - **Streamlit-Based Interface**: The use of Streamlit enables the creation of an interactive, responsive web application that enhances user experience.
   - **Advanced Search and Navigation**: Users can perform advanced searches, navigate through documents with ease, and access visual aids like charts and graphs to better understand the information.
6. **Scalability and Performance**
   - **Optimized Processing**: The system is designed to handle large volumes of documents and user queries efficiently, with optimized processing techniques to maintain performance.
   - **Load Balancing**: Techniques such as load balancing are employed to ensure the application remains responsive under high demand.

4.4 Platform Requirements

Hardware/Software	Hardware / Software element	Specification /version
Hardware	**Processor**	**I5**
	RAM	**8GB**
	Hard Disk	**250GB**
Software	**OS**	**Windows 7/8/10/11**
	IDE **Language Used** **Database Used**	**Visual Studio Code** **HTML,CSS, JavaScript, Python(Django)** **FIASS**

4.4.1Other Requirements

- **Check Database Connection**: Ensure that the database connection is stable and secure.
- **Ensure Internet connection is there**: Connect to wifi or hotspot data.
- **Check API KEYS**: Verify that API Keys are correctly working.

4.5 Use Case Diagram

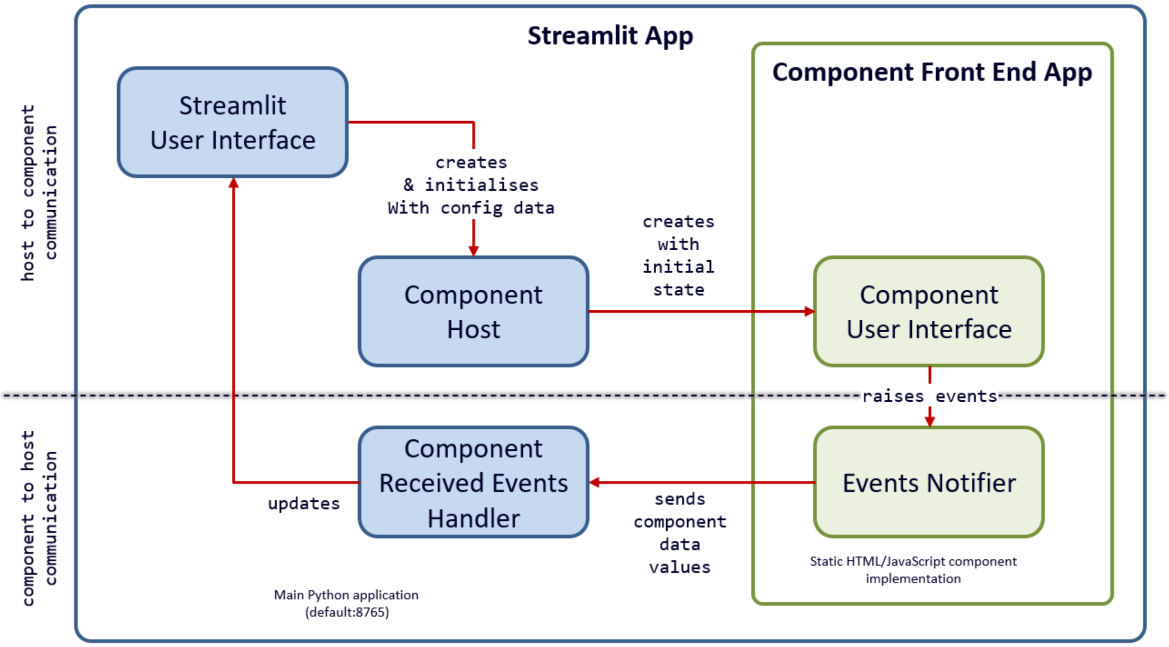

Streamlit App
Component Front End App
Streamlit User Interface
creates & initialises With config data
Component Host
creates with initial state
Component User Interface
raises events
Events Notifier
sends component data values
Component Received Events Handler
updates
host to component communication
component to host communication
Main Python application (default:8765)
Static HTML/JavaScript component implementation

Figure 1

4.6 Activity Diagram

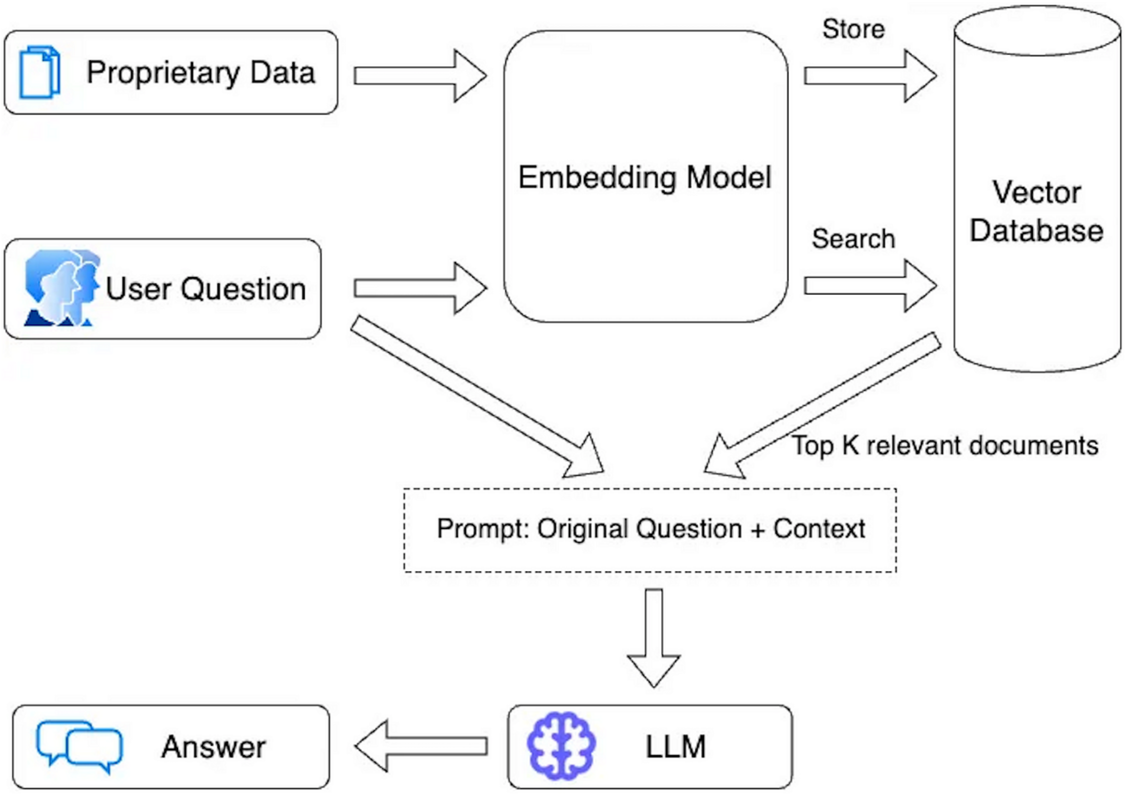


Figure 2 (a)

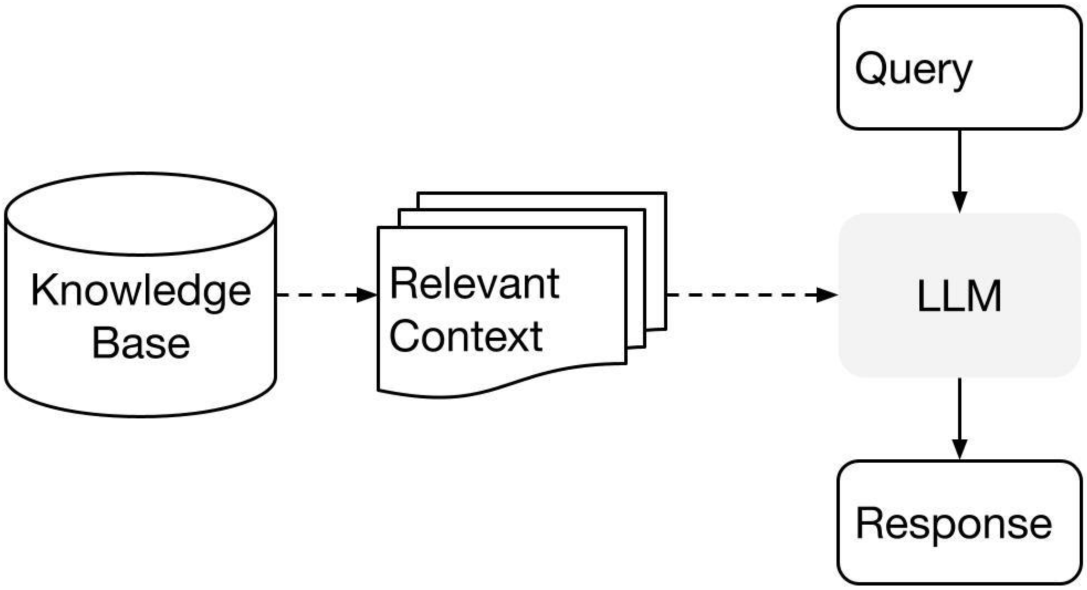


Figure 2(b)

4.7 Data Flow Diagram

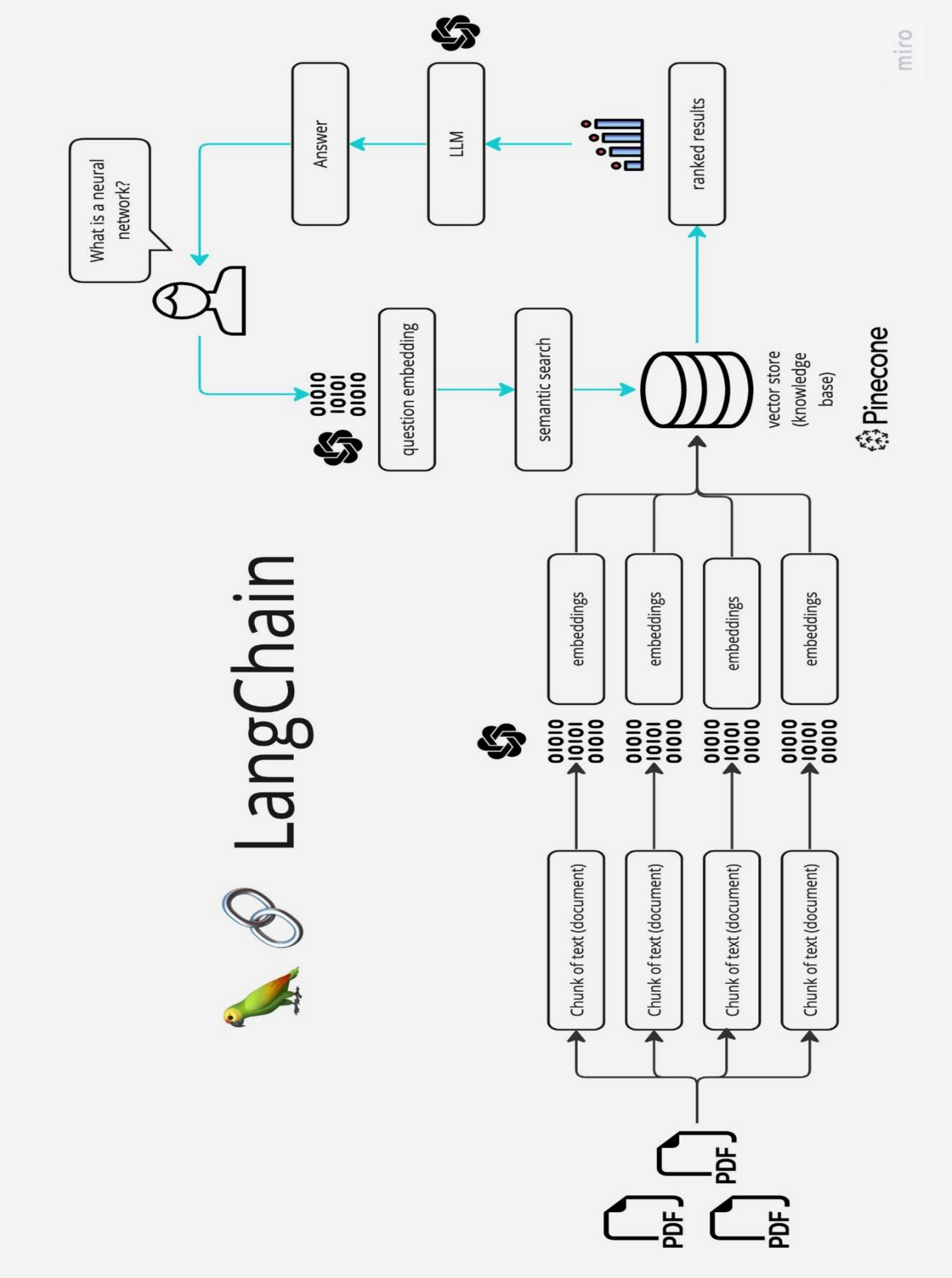


Figure 3

4.8 Database Design

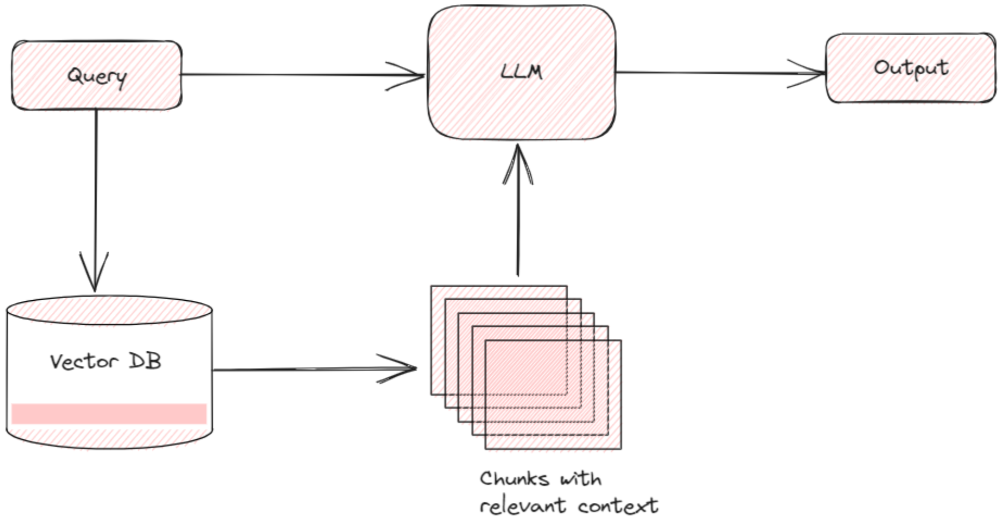
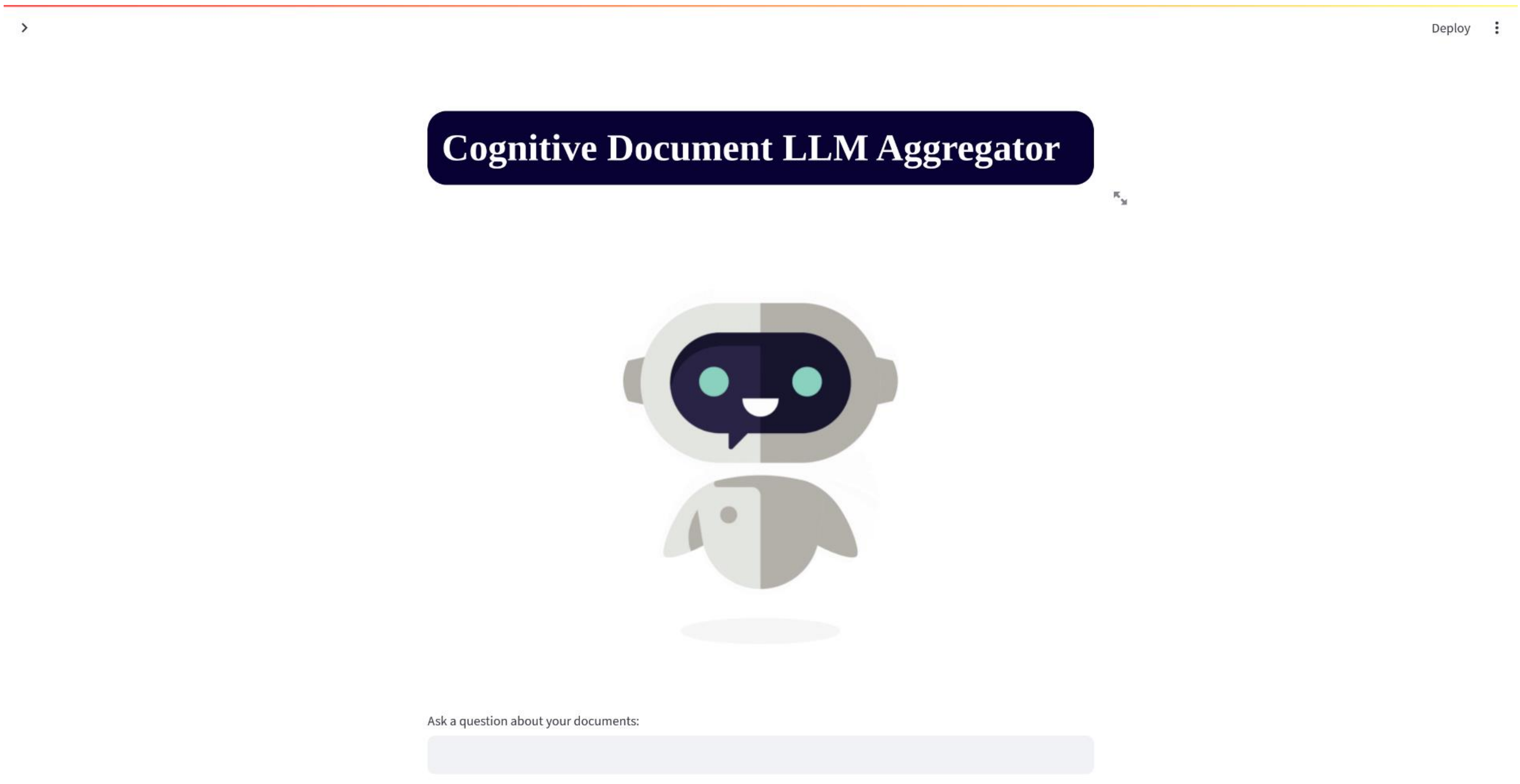


Figure 4

Chapter 5
RESULT

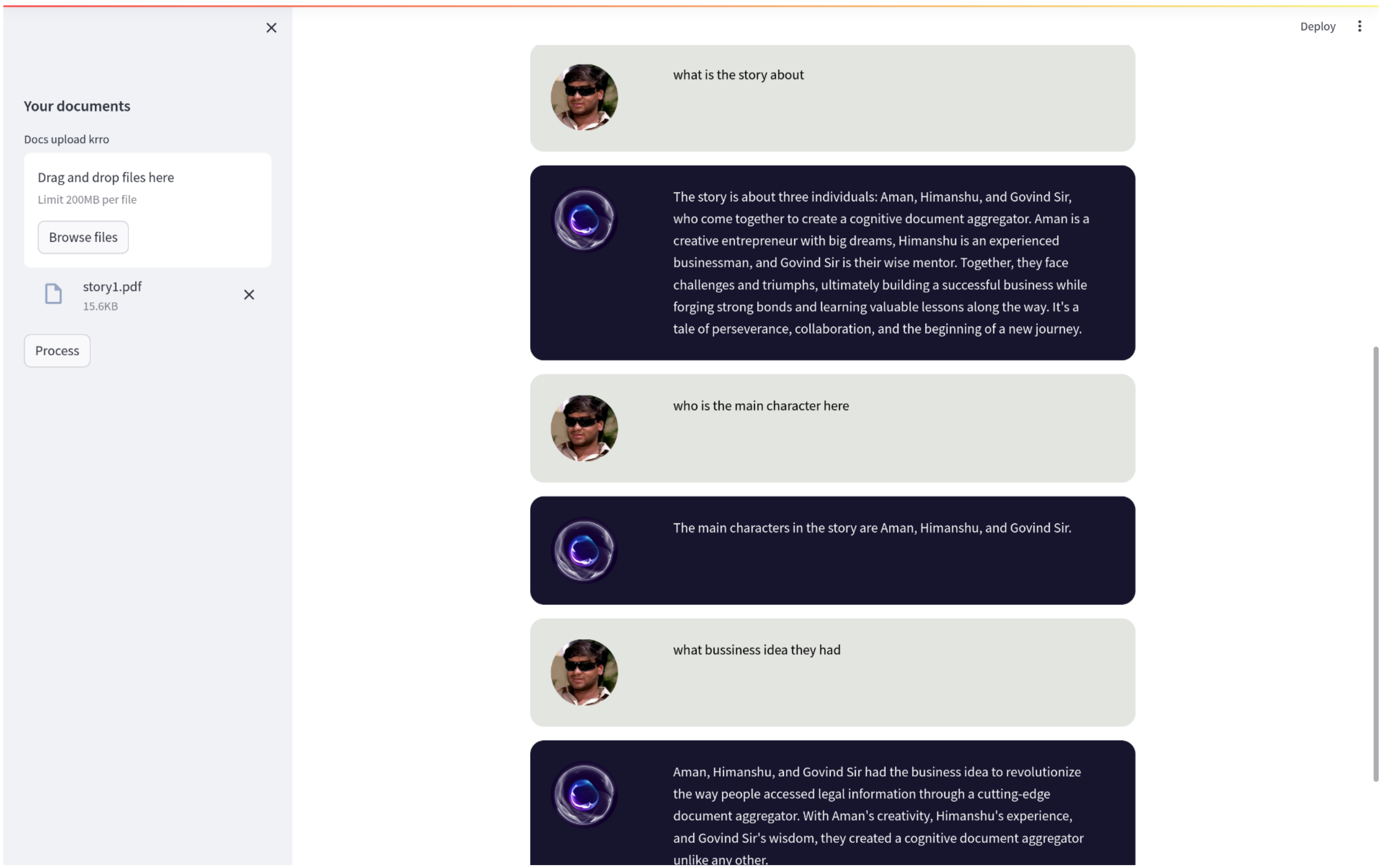
Home Page:



Cognitive Document LLM Aggregator

Ask a question about your documents:

CHAT Page:



what is the story about

The story is about three individuals: Aman, Himanshu, and Govind Sir, who come together to create a cognitive document aggregator. Aman is a creative entrepreneur with big dreams, Himanshu is an experienced businessman, and Govind Sir is their wise mentor. Together, they face challenges and triumphs, ultimately building a successful business while forging strong bonds and learning valuable lessons along the way. It's a tale of perseverance, collaboration, and the beginning of a new journey.

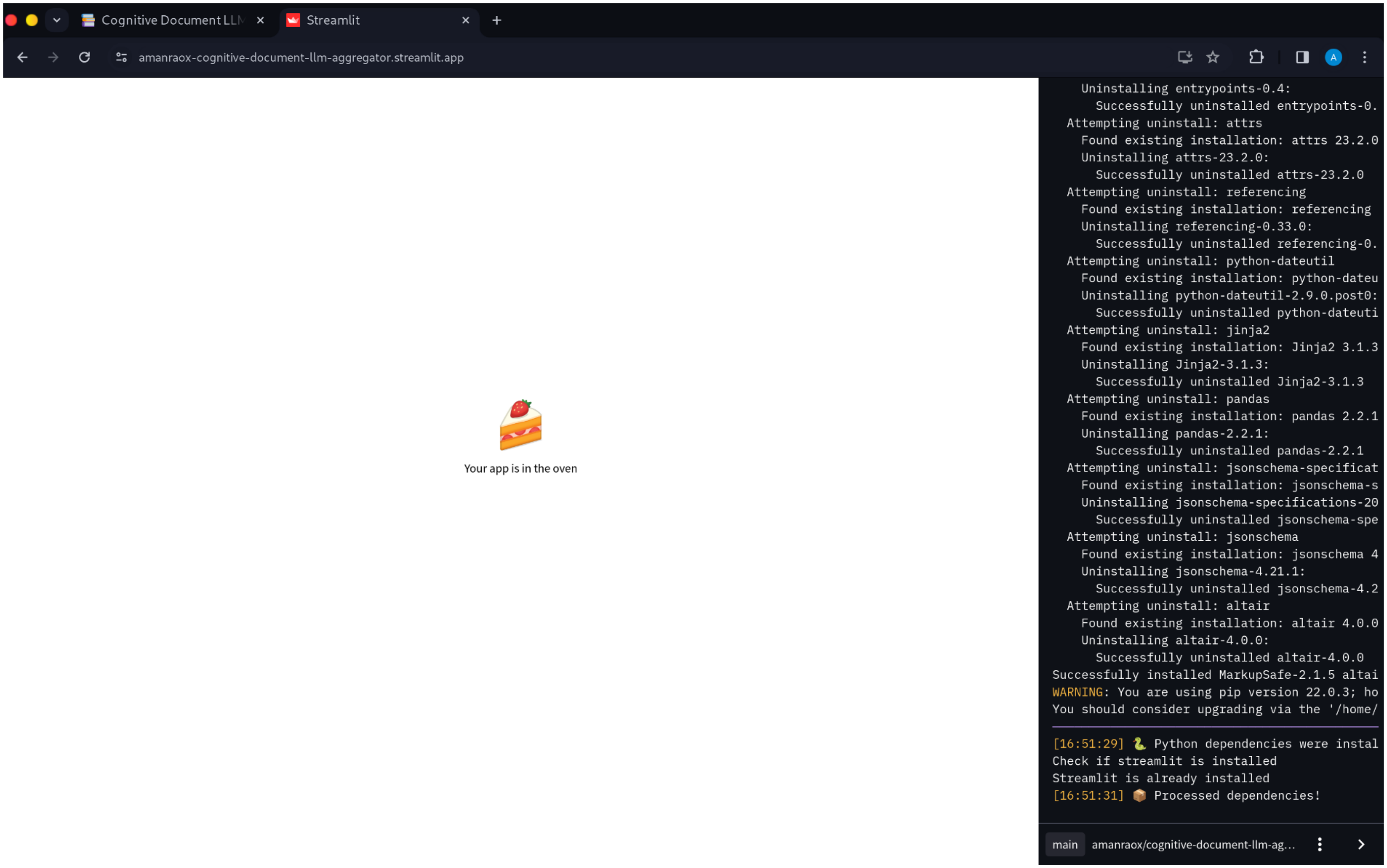
who is the main character here

The main characters in the story are Aman, Himanshu, and Govind Sir.

what bussiness idea they had

Aman, Himanshu, and Govind Sir had the business idea to revolutionize the way people accessed legal information through a cutting-edge document aggregator. With Aman's creativity, Himanshu's experience, and Govind Sir's wisdom, they created a cognitive document aggregator unlike any other.

Embeddings Module:



Cognitive Document LLM
Streamlit
amanraox-cognitive-document-llm-aggregator.streamlit.app
Your app is in the oven
main
amanraox/cognitive-document-llm-ag…

```
    Uninstalling entrypoints-0.4:
      Successfully uninstalled entrypoints-0.
  Attempting uninstall: attrs
    Found existing installation: attrs 23.2.0
    Uninstalling attrs-23.2.0:
      Successfully uninstalled attrs-23.2.0
  Attempting uninstall: referencing
    Found existing installation: referencing
    Uninstalling referencing-0.33.0:
      Successfully uninstalled referencing-0.
  Attempting uninstall: python-dateutil
    Found existing installation: python-dateu
    Uninstalling python-dateutil-2.9.0.post0:
      Successfully uninstalled python-dateuti
  Attempting uninstall: jinja2
    Found existing installation: Jinja2 3.1.3
    Uninstalling Jinja2-3.1.3:
      Successfully uninstalled Jinja2-3.1.3
  Attempting uninstall: pandas
    Found existing installation: pandas 2.2.1
    Uninstalling pandas-2.2.1:
      Successfully uninstalled pandas-2.2.1
  Attempting uninstall: jsonschema-specificat
    Found existing installation: jsonschema-s
    Uninstalling jsonschema-specifications-20
      Successfully uninstalled jsonschema-spe
  Attempting uninstall: jsonschema
    Found existing installation: jsonschema 4
    Uninstalling jsonschema-4.21.1:
      Successfully uninstalled jsonschema-4.2
  Attempting uninstall: altair
    Found existing installation: altair 4.0.0
    Uninstalling altair-4.0.0:
      Successfully uninstalled altair-4.0.0
Successfully installed MarkupSafe-2.1.5 altai
WARNING: You are using pip version 22.0.3; ho
You should consider upgrading via the '/home/

[16:51:29] 🐍 Python dependencies were instal
Check if streamlit is installed
Streamlit is already installed
[16:51:31] 📦 Processed dependencies!
```

Chapter 6

CONCLUSION & FUTURE SCOPE

Conclusion:

The Cognitive Document LLM Aggregator addresses critical challenges in document management and information retrieval by leveraging advanced natural language processing (NLP) and the latest large language models (LLMs). By integrating the state-of-the-art Gemini Pro model with a user-friendly Streamlit interface, the application provides a robust, efficient, and accurate solution for handling diverse document types.

Future Scope:

1. Integration with Emerging Models and Technologies

- **Advancements in LLMs**: As newer and more powerful large language models (LLMs) are developed, integrating these models into the application can further enhance its capabilities in understanding and generating natural language text.
- **Multi-modal Capabilities**: Exploring the integration of multi-modal models that can process both textual and visual information to provide richer insights from documents.

2. Enhanced User Interaction and Personalization

- **Natural Language Generation**: Expanding the application’s capabilities to generate natural language responses that go beyond simple query answering to provide insightful explanations and detailed reports.
- **Personalized Recommendations**: Implementing machine learning algorithms to provide personalized document recommendations based on user preferences and past interactions.

3. Advanced Analytics and Visualization

- **Interactive Dashboards**: Developing interactive dashboards with advanced visualization techniques to present complex document insights in a clear and intuitive manner.
- **Temporal Analysis**: Enabling temporal analysis capabilities to track changes and trends in document content over time, offering deeper historical insights.

4. Enhanced Security and Collaboration Features

- **Secure Document Handling**: Implementing advanced security measures to ensure the confidentiality and integrity of uploaded documents, especially for sensitive data.
- **Collaborative Features**: Introducing real-time collaborative editing and annotation tools to facilitate teamwork and knowledge sharing among users.

5. Integration with External Data Sources

- **API Integrations**: Integrating with external data sources and APIs to enrich document analysis with real-time data feeds and supplementary information.
- **Database Connectivity**: Offering seamless integration with databases and cloud storage platforms to facilitate unified data management and retrieval.

6. Continuous Learning and Improvement

- **Feedback Mechanisms**: Implementing robust feedback mechanisms to collect user input and usage data for continuous model refinement and improvement.
- **Automated Model Updates**: Developing mechanisms for automatic model updates based on new data and advancements in NLP research.

.

BIBLIOGRAPHY

- **Django Documentation.** (n.d.). "Official Django Documentation." Retrieved from https://docs.djangoproject.com/en/stable/.
The official documentation for Django, providing comprehensive guides on how to use the framework for web development.

- **Real Python.** (2021). "Build a Quiz App with Django and Vue." Retrieved from https://realpython.com/.
A tutorial on building a quiz application using Django for the backend and Vue.js for the frontend, relevant to creating an online examination system.

- **Django Girls Tutorial.** (n.d.). "Introduction to Django." Retrieved from https://tutorial.djangogirls.org/en/.
A beginner-friendly tutorial on Django, covering the basics of setting up and developing a web application.

- **Simple is Better Than Complex.** (2016). "How to Create a Simple Quiz in Django." Retrieved from https://simpleisbetterthancomplex.com/.
A practical guide on creating a simple quiz application with Django, which can be extended to develop a full-fledged online examination system.

- **DigitalOcean Community.** (2020). "How to Build a Web Application Using Django and React." Retrieved from https://www.digitalocean.com/community/tutorials/.
An in-depth tutorial on building web applications with Django and React, useful for creating a responsive and dynamic online examination system.

- **Stack Overflow.** (n.d.). "Best Practices for Django Projects." Retrieved from https://stackoverflow.com/.
A collection of community-contributed best practices for developing and managing Django projects, including tips relevant to online examination systems.

- **GitHub.** (n.d.). "Django Online Exam System Repository." Retrieved from https://github.com/.
A repository of an open-source Django project for an online exam system, providing code examples and implementation details.

- **GeeksforGeeks.** (2021). "How to Create a Quiz App Using Django." Retrieved from https://www.geeksforgeeks.org/.
A step-by-step guide on creating a quiz application with Django, covering essential features needed for an online examination system.