**Leveraging Google Gemini Pro LLM For Efficient NLP Applications**

**ABSTRACT**

Large language models (LLMs) have emerged as powerful tools for various natural language processing (NLP) tasks. Google's Gemini Pro LLM has shown remarkable capabilities in understanding and generating human-like text. This research paper explores the effective utilization of Gemini Pro LLM across diverse applications, demonstrating its potential to reduce workloads and achieve great results in less time. Through the development of several Python applications, we investigate the integration of Gemini Pro LLM with Streamlit, a powerful tool for building interactive web applications. The paper presents methodologies and implementation details for tasks such as PDF question answering, resume evaluation, health view on food analysis, YouTube video transcript summarization and chat with video, and text to SQL query generation. Evaluation results and insights are provided, highlighting the challenges, best practices, and future opportunities in leveraging LLMs for NLP applications. These applications exemplify the transformative potential of LLMs in optimizing information retrieval and analysis workflows, promising enhanced productivity, and effectiveness in various domains.

**Keywords:** Large Language Models (LLMs), Natural Language Processing (NLP), Google Gemini Pro/vision, Streamlit, PDF Question Answering, Resume Evaluation, Food Nutrition Analysis, YouTube Video Content Analysis, Text to SQL Conversion, Information Retrieval.

**Author Information:**

Bharath Goud Nadimpally a, bnadimpa@depaul.edu

Vikhyath Vardhan Goud Komire a, vkomire@depaul.edu

Eddie Cojulun a , ecojulun@depaul.edu

**INTRODUCTION**

The Gemini Pro is a family of AI models from Google that can understand and process information from different sources like text, images, audio, and video. It comes in three sizes: Ultra (powerful for complex tasks), Pro (balances performance and deployment), and Nano (designed for devices). It's trained in two stages: general training and then specialized training for specific tasks (like chat or developer tools). It can understand long sequences of information (up to 32,000 tokens), handle different input formats together (text with images or videos), and even generate images itself. It allows for various applications like chatbots, developer tools, and potentially future on-device AI experiences.

The ever-growing volume of digital information presents a significant challenge in today's world. Extracting meaningful insights and efficiently retrieving relevant details from vast document collections requires innovative approaches. Large Language Models (LLMs) have emerged as a promising technology with the potential to revolutionize information processing tasks.

1.1 What are Large Language Models (LLMs)?:

LLMs are a type of artificial intelligence (AI) trained on massive datasets of text and code. Through this extensive training, they develop the ability to understand and generate human-like language. LLMs can perform various tasks, including:

* Text generation: LLMs can create different creative text formats like poems, code, scripts, musical pieces, emails, letters, etc.
* Text summarization: They can condense lengthy documents into shorter summaries capturing key points.
* Machine translation: LLMs can translate text from one language to another while preserving meaning.
* Question answering: Given a context or a set of documents, they can answer natural language questions based on the information provided.
* Information retrieval: LLMs can analyze documents and identify relevant passages based on a user's query.

1.2 Challenges in Information Retrieval:

The widespread use of digital documents like PDFs, emails, and webpages has created vast repositories of information. However, efficiently extracting meaningful insights from these collections remains a challenge. Some key obstacles include Information overload, Keyword-based search limitations, and the fact that traditional search engines often struggle to understand the context of a user's query and retrieve relevant documents that may not contain the exact keywords but still hold valuable information.

1.3 Potential Benefits of LLMs for Information Processing:

The capabilities of LLMs offer promising solutions to the challenges faced in information retrieval. LLMs offer enhanced search capabilities by understanding the semantic meaning of text, allowing them to retrieve documents relevant to the user's intent beyond mere keyword matching. They excel in context-aware processing, adeptly recognizing the nuances of a user's query and retrieving pertinent information even when phrased differently from the document content. Through natural language interaction, LLMs enable users to intuitively interact with information using everyday language, enhancing the user experience. Furthermore, their advanced question-answering abilities empower them to analyze extensive datasets and provide nuanced responses to complex inquiries, showcasing their deep understanding and reasoning capabilities.

1.4 Project Overview: LLM-based Applications for Information Retrieval and Analysis:

This project explores the potential of LLMs for practical information processing tasks through the development of five applications:

* ATS Resume Application: Leveraging the power of Google Gemini Pro, this tool evaluates user resumes in PDF format against job descriptions. Its objective is to aid individuals in refining their resumes for job applications by pinpointing crucial skills and suggesting areas for enhancement based on specific job requirements.
* Chat with PDF’s Application: This application allows users to upload multiple PDFs and ask natural language questions about their content. The application utilizes Google Gemini Pro, an LLM, to analyze the uploaded documents, retrieve relevant passages, and generate answers to the user's questions. This application showcases the potential of LLMs for facilitating information retrieval from document collections.
* Text to SQL query generation Application: The "Text to SQL query generation" application serves as a practical tool to assist users in retrieving data from an SQLite database using natural language queries. Its primary purpose is to streamline the process of querying data stored in relational databases by enabling users to input questions in plain English and receive corresponding SQL commands as responses. This application aims to bridge the gap between non-technical users and database systems, making data retrieval more accessible and efficient.
* Health View Application: The Health View Assistant application aims to provide users with nutritional information about food items captured in images. By leveraging Google's Gemini Pro Vision API, the app assists users in understanding the nutritional content of their meals, including calorie count, vitamins, minerals, and potential health implications. And it can estimate the weight of the food in the picture and gives calories, vitamins, and minerals in it. This application is helpful for individuals who want to make informed dietary choices and track their nutritional intake.
* YouTube video transcriber & chat with video: The aim of the Video Summarizer application is to help users convert YouTube video transcripts into comprehensive notes and facilitate interaction through chat-based queries. Utilizing the Google Gemini Pro API, the app condenses video transcripts, offers detailed notes within a defined word count, and enables users to inquire questions from the video content. This tool is beneficial for individuals seeking a swift understanding of lengthy videos and engaging in interactive dialogues centered around the video material.

In the subsequent sections of this report, we will provide detailed insights into the functionalities of each application, uncovering the technical intricacies, addressing encountered challenges, and outlining future development strategies. Throughout this exploration, the spotlight will be on Google Gemini Pro, the LLM powering these innovative applications. Selected for its cutting-edge natural language processing capabilities and proven performance, Google Gemini Pro has served as a cornerstone in our quest to revolutionize information processing. Our overarching aim is to showcase the profound impact of LLMs, exemplified by Google Gemini Pro, in the realm of information retrieval and analysis. By pushing the boundaries of technology and embracing the possibilities afforded by LLMs, we endeavor to pave the way for further advancements and empower users with unparalleled capabilities in navigating and understanding vast repositories of information.

**LITERATURE REVIEW**

Literature Review of Topic 1: Exploring Large Language Models and their Applications in Information Processing

Large Language Models (LLMs) have emerged as a revolutionary force in Natural Language Processing (NLP) tasks. These powerful models, trained on massive datasets of text and code, are capable of understanding and generating human-like language with remarkable fluency. This literature review delves into the recent advancements in LLM development and explores their diverse applications in information processing, with a particular focus on the capabilities of the Gemini family of LLMs developed by Google AI.

Yan et al. (2023)[1] present a comprehensive systematic scoping review focusing on the intersection of educational technology and large language models (LLMs), with a particular emphasis on the automation and support of educational tasks. Through an analysis of 118 peer-reviewed papers published since 2017, the study identifies 53 distinct use cases for LLMs across various educational functions, ranging from profiling and detection to feedback provision and content generation. However, the review also highlights practical and ethical challenges, including technological readiness, replicability, transparency, privacy considerations, and ethical principles such as beneficence. In response to these challenges, the authors propose three key recommendations for future research: updating existing innovations with state-of-the-art models like GPT-3/4, advocating for open-sourcing models and systems, and adopting a human-centered approach throughout the developmental process. By offering a comprehensive overview of the current landscape and addressing potential avenues for improvement, this study serves as a valuable resource for researchers navigating the evolving terrain of AI in education, leveraging insights from ChatGPT and other generative AI models.

Deroy and Maity (2023)[4] present a thought-provoking investigation into the potential biases inherent in case judgment summaries generated by both legal datasets and large language models (LLMs). Their study delves into the impact of biases on legal decision-making processes, examining the accuracy, fairness, and broader implications within legal contexts. Through an examination of gender-related, race-related, crime against women-related, country names, and religious keywords, the research reveals compelling evidence of biases in outputs produced by LLMs and pre-trained abstractive summarization models. This study not only sheds light on the existence of biases but also underscores the necessity for further inquiry into the underlying reasons behind these biases. By addressing this critical issue, Deroy and Maity contribute to a deeper understanding of the role of technology in shaping legal discourse and its ramifications for justice systems globally.

Literature Review of Topic 2 : Exploring LAW’s for LLM

In the landscape of legal scholarship, Charlotin (2023)[3] at offers a compelling exploration of the transformative impact of Large Language Models (LLMs) on the legal profession and the future trajectory of legal practice. His work, "Large Language Models and the Future of Law," published in SSRN, delves into the emergence of LLMs, epitomized by ChatGPT, and their implications for legal practitioners. Charlotin not only elucidates the core functionalities of LLMs and their integration into legal workflows but also examines the profound implications for legal education and the evolving role of jurists in light of this technological advancement.

Literature Review of Topic 3 : Information Retrieval and Question Answering:

The ability of LLMs to analyze vast amounts of text has propelled significant progress in information retrieval (IR) and question answering (QA) systems. Studies like "Benchmarking Large Language Models in Retrieval-Augmented Generation" by Lewis et al. (2020)[8] highlight the effectiveness of the Retrieval-Augmented Generation (RAG) technique. RAG leverages pre-trained LLMs to retrieve relevant passages from a document corpus, allowing for the generation of high-quality answers to complex questions. Similarly, Nie et al. (2022)[1] in "Information Retrieval Meets Large Language Models: A Strategic Report from the Chinese IR Community" emphasize the strategic role of LLMs in modern IR systems.

Kwiatkowski et al. (2019)[15] introduced the Natural Questions dataset and explored the use of LLMs for answering questions based on real-world web data. Their work highlights the potential of LLMs to understand and retrieve information from unstructured text sources, forming the foundation for applications like our "Chat With PDF" system for question answering from PDF documents. Our project builds upon these advancements by incorporating the RAG technique into the "Chat with PDF" application. This application leverages Gemini's LLM capabilities to analyze and understand the content of PDF documents. By integrating RAG, users can engage in a conversation-like interaction with the PDF, asking specific questions and receiving answers retrieved and summarized by the LLM. This application exemplifies the potential of LLMs to enhance user experience by facilitating efficient retrieval of knowledge from readily available documents.

Literature Review of Topic 4 : Text Summarization and Video Analysis:

LLMs have demonstrably improved text summarization techniques. Liu (2020)[24] in "Large Language Models for Information Retrieval: A Survey" provides a comprehensive overview of LLM applications in NLP tasks, including text summarization. The study emphasizes the importance of metrics like ROUGE scores to evaluate the quality and coherence of summaries generated by LLMs.

Atluri et al. (2022)[18] propose a video summarization system that leverages Natural Language Processing (NLP) and machine learning to automatically generate concise summaries of YouTube video transcripts. Their system retrieves transcripts from user-provided video links and employs techniques like Hugging Face Transformers and pipelining to condense the information while retaining key points. This approach addresses the challenge of efficiently extracting core content from lengthy video lectures or educational materials. The study suggests that their method offers faster summarization compared to other techniques, while still maintaining fidelity to the video's central concepts. This research aligns with our project's application, "YouTube video transcriber & chat with video," which explores similar functionalities for video transcript summarization and user interaction with video content through natural language queries.

Addressing the challenge of information overload, Lyu et al. (2020)[10] investigated the use of LLMs for text summarization, focusing on generating concise summaries from lengthy documents or transcripts. Their research is particularly relevant to our "YouTube video transcriber & chat with video" application, where LLMs provide summaries of video content.

While the current iteration of Gemini focuses primarily on text-based information processing, the "Gemini: A Family of Highly Capable Multimodal Models" paper by Hernandez et al. (2023)[14] highlights its potential for future expansion into areas like video analysis. The "YouTube video transcriber & chat with video" application leverages these future possibilities. By enabling users to obtain summaries of key information from YouTube videos, this application demonstrates the potential of LLMs to evolve into comprehensive information processing tools.

Literature Review of Topic 5 : LLM Applications in Specific Domains:

The versatility of LLMs extends to specific domains as well. Liu et al (2020)[24] in "Large Language Models for Information Retrieval: A Survey" explores the use of LLMs in tasks like resume evaluation. However, the study cautions about potential biases present in the training data, which can lead to unfair evaluations.

The "ATS Resume Model" application within our project takes this challenge into consideration. While leveraging Gemini's capabilities for analyzing resume content, the application prioritizes factual accuracy and avoids perpetuating biases often present in traditional resume evaluation methods. chengguanf et al(2024) [6]

Focusing on the recruitment process, Zhang et al. (2024)[23] proposed an LLM-based resume evaluation system that analyzes resumes and job descriptions. Similar to our "ATS\_Resume" application, their approach leverages LLMs' language understanding to assess candidate suitability and streamline recruitment.

Skondras et al. (2023)[13] explore the use of large language models (LLMs) like ChatGPT for generating synthetic resume data. Their findings highlight the potential of LLMs to address data sparsity challenges in resume classification tasks. They demonstrate that augmenting real-world datasets with synthetic resumes, particularly when using advanced models like BERT, can significantly improve classification accuracy, especially for underrepresented job categories

Further emphasizing the impact of AI in recruitment, Yadav et al. (2023)[12] investigate the adoption of AI tools within the IT industry's talent acquisition process. Their study, conducted through surveys with HR professionals, reveals significant improvements in efficiency and speed after AI implementation. The research highlights AI's ability to streamline tasks like resume screening and candidate selection, ultimately contributing to faster time-to-hire and improved quality of hires. These studies underscore the growing role of AI and machine learning in resume processing and talent acquisition. Our project aligns with this trend by leveraging the capabilities of the Gemini LLM to develop applications that can contribute to a more efficient and effective recruitment process.

Similarly, the "Health View Application" application exemplifies the limitations of LLMs discussed by Nie et al. (2022)[1] in "Information Retrieval Meets Large Language Models: A Strategic Report from the Chinese IR Community." LLMs, due to security and privacy concerns, may not be suitable for accessing confidential health information. The " Health View Application" acknowledges this limitation and focuses on readily available health data sources that comply with data privacy regulations.

In the field of healthcare, Wang et al. (2022)[20] explored the use of LLMs for medical image analysis and diagnosis. Their work aligns with our "Health View Application" application, demonstrating the potential of LLMs in interpreting multimodal data, combining text and image inputs to inform accurate diagnosis and treatment recommendations. Artificial intelligence (AI) presents a transformative potential for healthcare delivery, particularly in resource-limited settings.

The recent advancements in large language models (LLMs) have sparked a growing interest in their potential applications beyond natural language processing tasks. Van et al. (2024)[17] delve into the integration of LLMs with computer vision, specifically focusing on visual language models (VLMs) like LLaVA, Flamingo, and CLIP. Their work highlights the promising capabilities of VLMs in handling multimodal data, suggesting their potential applicability in the domain of medical imaging analysis. While the application of LLMs for medical diagnosis remains an under-explored area, Van et al. demonstrate the effectiveness of VLMs in analyzing various biomedical images, including brain MRIs, blood cell microscopy images, and chest X-rays.

Wahl et al. (2018)[19] discuss the applicability of AI subfields like machine learning and image processing for tasks such as disease outbreak prediction and clinical decision support, highlighting the potential to improve healthcare efficiency in low-income regions. However, challenges like data scarcity and infrastructure limitations require careful consideration for successful implementation.

In the domain of nutrition, deep learning approaches are making significant strides in automated dietary assessment. Mezgec and Koroušić Seljak (2017)[11] introduce NutriNet, a deep learning system for food and drink image recognition. Their work demonstrates promising results with high classification accuracy, paving the way for applications like mobile-based dietary monitoring tools. This aligns with our project's focus on leveraging AI for healthcare applications, and the exploration of dietary assessment through food image recognition holds promise for improving dietary monitoring and personalized health management.

Literature Review of Topic 6 : Data Analytics and Natural Language Interfaces:

LLMs are revolutionizing data analysis by enabling users to interact with databases using natural language. Feng et al. (2023)[5] in "Improving LLM-based Machine Translation with Systematic Self-Correction" explore the potential of LLMs for generating SQL queries from natural language instructions. However, the study emphasizes the importance of factual accuracy in generated queries. The ability to generate natural language instructions that translate into syntactically correct SQL queries presents a significant opportunity for data analysis.

Troy et al. (2023)[16] propose a framework that leverages generative AI and Extended Backus-Naur Form (EBNF) context-free grammars to achieve this goal. Their framework utilizes Augmented Transition Networks (ATNs) within the ANTLR4 platform to process EBNF grammars and generate SQL statements based on predefined objectives. The study delves into critical considerations for this process, including Grammar Scope Reduction, Recursion Handling, Generative Repetition Control, and ATN-based Generator Design. While the prototype demonstrates promising results in generating syntactically valid SQL queries, the authors acknowledge limitations, including challenges in achieving semantic validity and controlling randomized transition selection. They propose future work to address these limitations and explore the framework's applicability in cybersecurity for AI-driven detection of cyberattacks through SQL analysis. This research aligns with our project's goals by exploring the potential of AI for natural language interaction with databases, paving the way for more user-friendly data exploration tools.

The "Text to SQL query generation Application" application addresses this need. By leveraging Gemini's ability to understand and respond to complex instructions, the application allows users to formulate SQL queries using natural language. Behind the scenes, Gemini translates these instructions into accurate and efficient SQL code, empowering users with an intuitive interface for data exploration. In the realm of data analytics, Li et al. (2024)[9] proposed a system that generates SQL queries from natural language input using LLMs. Their approach, like our "Test to SQL query generation" application, aims to democratize data access and analysis by enabling users to interact with databases through natural language queries. [4

Literature Review of Topic 7: Ethical Considerations in LLM Development:

The rapid advancements in LLM development necessitate a focus on responsible creation and deployment. Liu et al(2020)[24] in "Large Language Models for Information Retrieval: A Survey" highlights the importance of mitigating bias and ensuring fairness in LLMs.

Similarly, Hernandez et al. (2023)[7] in "Gemini: A Family of Highly Capable Multimodal Models" discuss the safety measures employed during Gemini's training to promote responsible LLM development. While Large Language Models (LLMs) offer exciting possibilities, it is crucial to acknowledge and mitigate potential risks associated with their development and deployment. Weidinger et al. (2021)[21] provide a comprehensive analysis of these risks across six key areas in their work "Ethical and Social Risks of Harm from Language Models". These areas encompass discrimination, information privacy, misinformation, malicious uses, human-computer interaction, and automation-related concerns. The study highlights how biased training data can lead LLMs to perpetuate discrimination and exclusion, while information leakage and the potential for generating misleading content pose further challenges.

Recognizing the importance of ethical considerations, Bender et al. (2021)[2] explored the potential risks and challenges associated with LLMs, including issues related to bias, privacy, and transparency. Their work highlights the need for responsible development and deployment of LLM-based applications, a critical aspect considered throughout the design and implementation of our project applications.

Literature Review of Topic 8 : Leveraging Python Libraries and AI Models for Natural Language Processing:

Streamlit emerges as a prominent tool for simplifying web app development using Python libraries. By abstracting away, the complexities of web development languages like HTML, CSS, and JavaScript, Streamlit empowers developers to create interactive web apps effortlessly. Its intuitive API reference, organized by activity type, facilitates easy navigation and implementation of methods for displaying data and optimizing performance (Streamlit Documentation). While primarily known for image processing, the Python Imaging Library (PIL) finds relevance in NLP applications through its capabilities to handle image data. PIL enables developers to preprocess and analyze textual information embedded within images, a task increasingly important in NLP tasks involving multimedia content (Pillow Documentation). Integrating PIL with NLP workflows opens avenues for extracting textual insights from images, enriching the understanding of textual data in diverse contexts.

In the realm of information retrieval, FAISS offers efficient implementations for similarity search tasks. By indexing large datasets of text embeddings, FAISS facilitates fast and accurate retrieval of documents like a given query embedding (FAISS Documentation). This capability is invaluable in NLP applications requiring retrieval of relevant information, such as question answering systems and document summarization. FAISS enables developers to build robust retrieval systems that enhance the effectiveness of NLP applications. PyPDF2, a pure-Python library for working with PDF files, provides essential functionalities for NLP tasks involving document analysis and text extraction (PyPDF2 Documentation). Its ability to read, extract text, manipulate, and analyze PDF documents makes it a valuable tool for developers dealing with textual data stored in PDF format. Integrating PyPDF2 into NLP workflows enables efficient processing of PDF documents, facilitating tasks such as information retrieval, document summarization, and data extraction. By harnessing the power of Streamlit, PIL, LangChain, FAISS, and PyPDF2, developers can create sophisticated NLP solutions that address diverse challenges and empower users with actionable insights derived from textual data.

**DATA**

Our project involved working with a variety of data types to train and evaluate the capabilities of the Gemini LLM across different information processing applications. This section details the data sources leveraged for each application and the preprocessing techniques employed to ensure data quality and compatibility with the LLM.

ATS Resume Application Data:

* ATS Resume: The Automatic Text Summarization (ATS) Resume application analyzes both resumes and job descriptions. Both data types were collected in PDF format. Similar to the Chat With PDF application, text extraction and cleaning techniques were employed to prepare the data for analysis by Gemini Pro LLM.

PDF Documents and Text Extraction:

* Chat With PDF’s: This application focuses on questions answering from PDF documents. A dataset of PDFs was collected, and text extraction techniques were applied to convert the document content into plain text. This text was further cleaned to remove noise and inconsistencies, such as headers, footers, and page numbers. To facilitate efficient processing by the LLM, the extracted text was divided into smaller chunks.

Data for SQL Query Generation:

* Text to SQL query generation: This application focuses on generating SQL queries from natural language instructions. SQL databases were used to store the data relevant for this application. The data was originally stored in CSV format, a common comma-separated values format, and then imported into SQLite databases, a lightweight relational database management system.

Image Preprocessing for Health Analysis:

* Health View: This application focuses on health analysis from food images. A collection of food images was obtained, and preprocessing techniques were applied to prepare them for input into Gemini Pro LLM's vision capabilities. The specific preprocessing steps might have involved resizing images, converting them to a specific color format, or removing irrelevant background information.

YouTube Video Transcripts & chat:

* YouTube transcriber & chat with video: This application incorporates functionalities for both video transcript summarization and chat interactions with video content. YouTube Transcript API was utilized to obtain transcripts for the videos used in the dataset. The transcripts underwent text cleaning and preprocessing steps similar to those used for PDF documents. Additionally, the transcripts were segmented into smaller chunks to enable efficient processing by the LLM.

By employing these data preprocessing techniques, we ensured that the information fed into the Gemini LLM was clean, consistent, and formatted appropriately for each application's specific requirements. This data preparation step plays a crucial role in achieving optimal performance and accurate results from the LLM models.

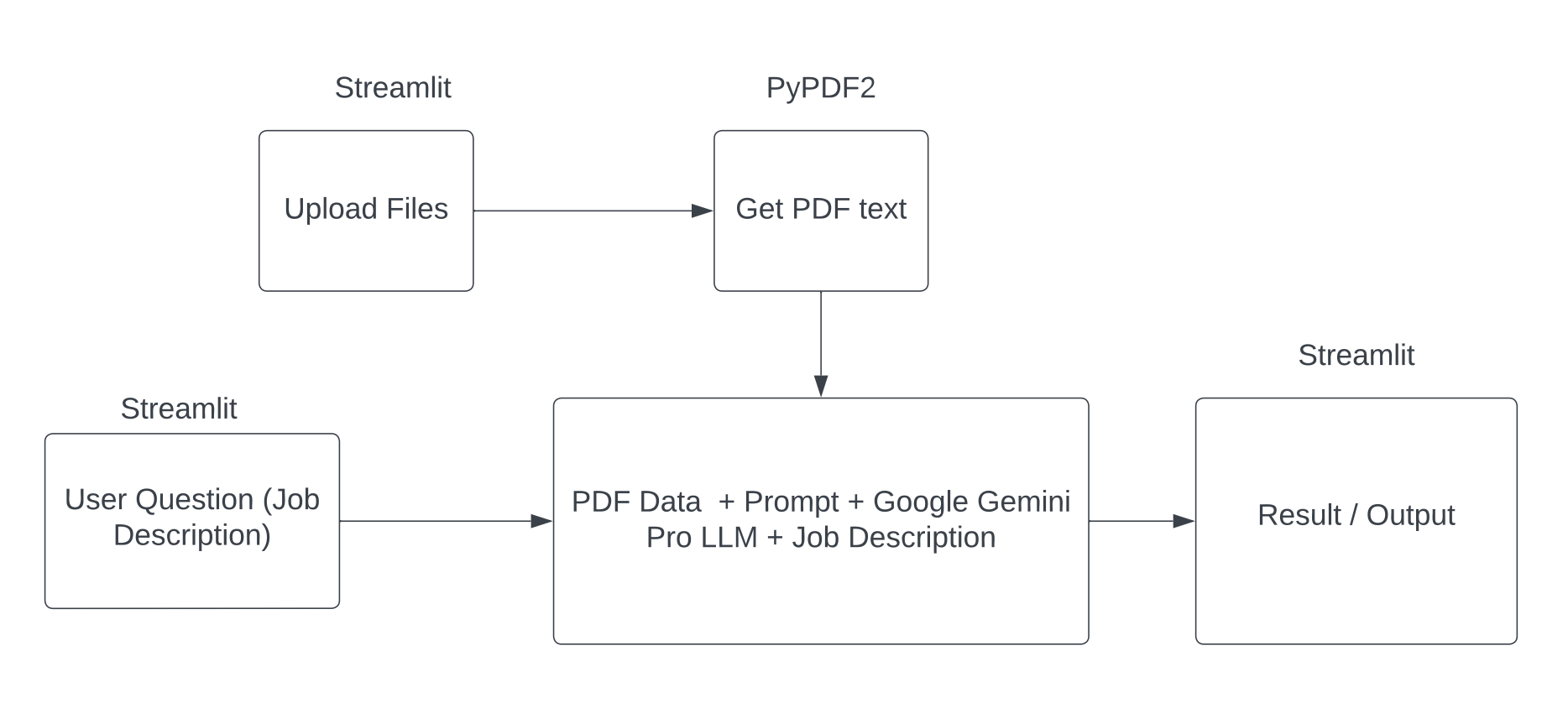
**METHODOLOGIES**

1.1 ATS Resume Application

This application serves as a resume evaluator, leveraging Gemini Pro AI to analyze resumes and provide insights based on job descriptions. It empowers users to upload their resume (PDF format) and paste a relevant job description. Upon submission, the application generates an evaluation report highlighting percentage match between the resume and job description, missing keywords that could be addressed, and a profile summary.

1.2 ATS Resume Methodology

* UI: Streamlit is employed to create a user-friendly web interface for file uploads, text input, and displaying results.
* PDF Processing: PyPDF2 library extracts text from uploaded PDF resumes.
* Gemini Pro Integration: The genai library interacts with Gemini Pro to generate content based on prompts.
  + The get\_gemini\_response function handles model configuration and prompt generation.
* Prompt Design: A comprehensive prompt guides Gemini Pro, defining its role as an experienced ATS, emphasizing expertise in tech domains, and specifying the desired output format.
* Custom Function: The input\_pdf\_text function accurately extracts text from PDF.

 figure 1: Flow chart of ATS Resume Application

2.1 Chat with PDF Application

This application facilitates a conversational style of information retrieval from uploaded PDF documents using Gemini Pro AI. Users can upload multiple PDF files. The application processes the uploaded PDFs, extracting text content and creating text chunks. Users can then ask questions about the content of the PDFs in a conversational manner. Gemini Pro, guided by a carefully crafted prompt, analyzes the uploaded PDFs (context) and the user's question to generate a relevant answer.

2.2 Chat with PDF Methodology

* UI: Streamlit is used to build the user interface, enabling file upload, text input, and displaying results.
* PDF Processing: PyPDF2 extracts text from uploaded PDFs.
* Text Chunking: RecursiveCharacterTextSplitter from Langchain breaks down the extracted text into manageable chunks.
* Embeddings & Vector Store:
  + GoogleGenerativeAIEmbeddings create numerical representations (embeddings) of text chunks using a pre-trained model.
  + FAISS vector stores efficiently stores and retrieves these embeddings based on similarity.
* Conversational Chain:
  + A custom PromptTemplate defines the format for prompts sent to Gemini Pro.
  + ChatGoogleGenerativeAI specifies the model (Gemini Pro) and temperature parameter.
  + The load\_qa\_chain function builds a question-answering chain using the prompt and model.
* User Input Processing:
  + New embeddings are created for the user's question.
  + The vector store searches for similar documents (text chunks) based on the user's question embedding.
  + The question-answering chain is called with the relevant documents (context) and the user's question.
* Output: The application displays the answer generated by Gemini Pro to the user.

2.3 Explanation of get\_pdf\_text function:

This function iterates through each uploaded PDF document using a loop. Within the loop:

1. A PdfReader object is created for the current PDF.
2. It iterates through each page of the PDF.
3. The extract\_text method extracts text content from each page.
4. The extracted text from each page is appended to a single text variable.
5. Finally, the function returns the combined text content from all pages of all uploaded PDFs.

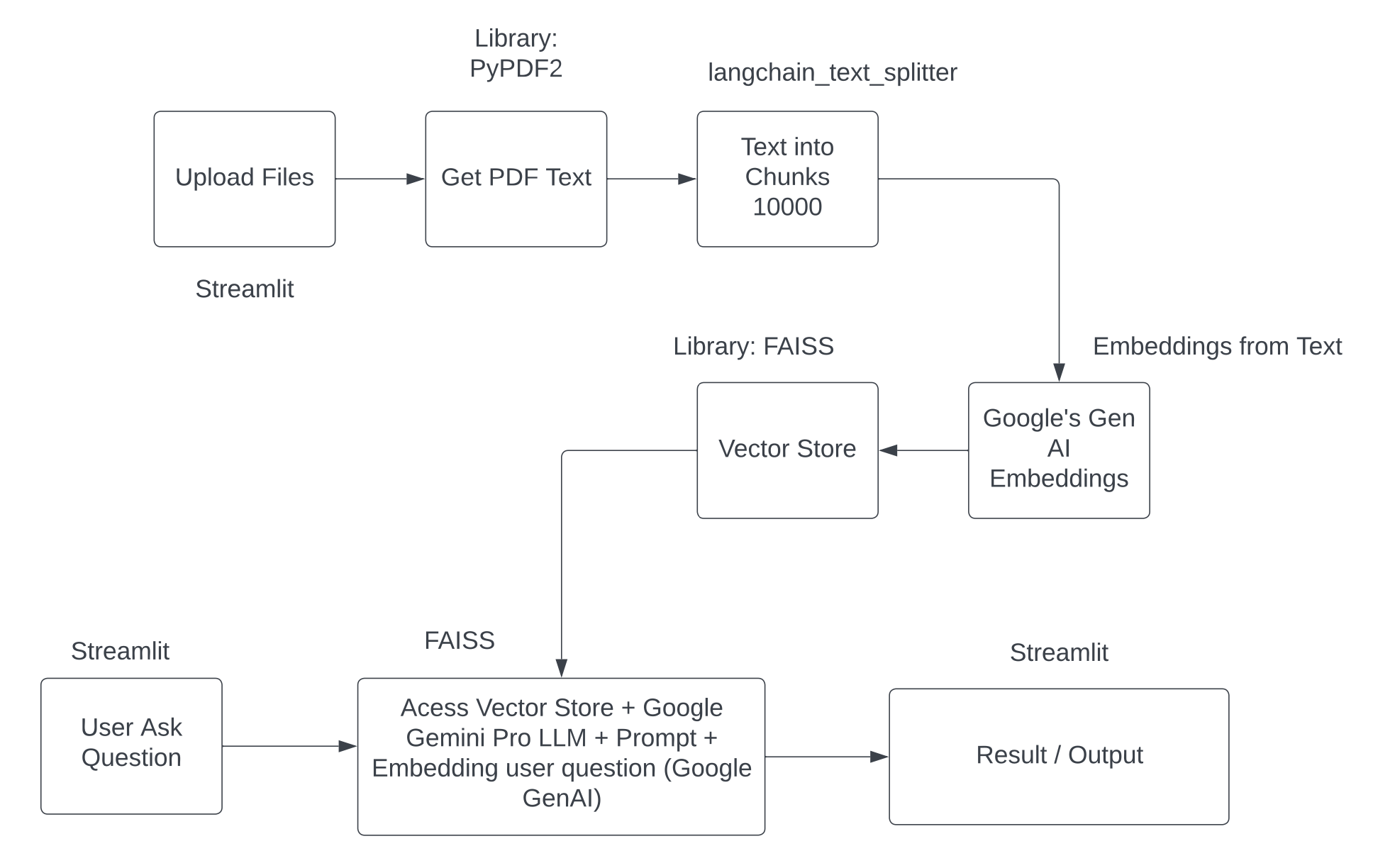


Figure-2: Flow chart of Chat with PDF’s

3.1 Text to SQL query generation Application

This application enables upload CSV file to create tables in a SQLite database. Users ask questions in natural language, and Gemini Pro generates SQL queries to retrieve data from the database. The application executes the generated SQL queries, displaying the results for users.

3.2 Text to SQL query generation Methodology

* User Interface (UI):
  + Streamlit Framework: Streamlit provides a fast and efficient framework for building the web application, enabling quick deployment and user-friendly interactions.
  + Interface Elements:
    - File Upload: Handles CSV file uploads with clear guidance for users.
    - Text Input: Facilitates natural language question entry.
    - Table Name Input: Enables users to specify the target table within the database.
    - "Ask the question" Button: Triggers the query generation and execution process.
    - Information Display: Presents table details, query results, and feedback messages.
* Data Storage and Handling:
  + SQLite Database:
    - Adoption of SQLite for its lightweight and embedded nature, suitable for smaller datasets and local application usage.
    - Data Persistence: Stores uploaded CSV data for later queries.
* CSV Data Handling:
  + insert\_csv\_to\_db Function:
    - Validates file format and table name for validity.
    - Creates a new table if it doesn't exist, dynamically constructing column definitions.
    - Inserts CSV data into the specified table efficiently.
    - Provides clear feedback on completion or errors.
* Database Interactions:
  + read\_sql\_query Function:
    - Retrieves table details such as column names, row count, and available tables.
    - Executes SQL queries using SQLite's Python API.
    - Handles potential errors with informative messages.
    - Returns query results and updated prompt for further interactions.
* Natural Language Processing (NLP) and Query Generation:
  + Gemini Pro Integration:
  + genai Library: Simplifies interaction with the Gemini Pro language model.
  + API Key Management: Stores the API key securely using environment variables.
  + get\_gemini\_response Function:
    - Formats a comprehensive prompt for Gemini Pro, incorporating table details and column names.
    - Communicates with Gemini Pro through API calls to generate SQL queries based on user questions.
    - Receives and returns the generated SQL query as a plain text string.
* Query Execution and Result Display:
  + SQLite Interaction:
    - Establishes a connection to the SQLite database.
    - Executes the generated SQL query using a cursor object.
    - Retrieves query results as a list of rows.
    - Commits any changes to the database for persistence.
    - Closes the database connection properly.
  + Streamlit Result Presentation:
    - Displays the retrieved query results in a structured format using subheaders and text elements.

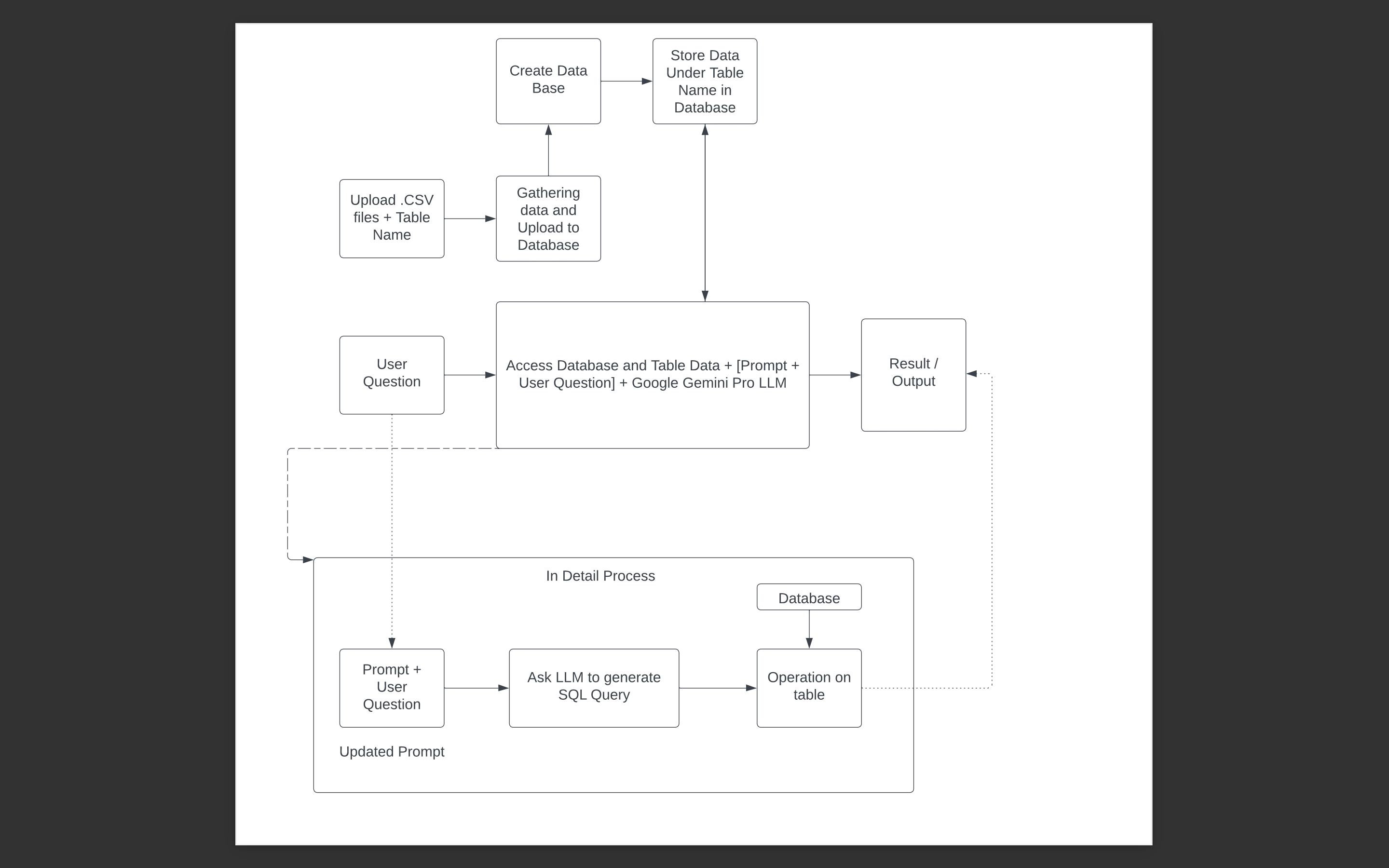


Figure-3: flow chart of Text to SQL query generation Application

4.1 Health View Application

This application assists users with health and nutrition analysis using an image and a text prompt. Users upload an image of food items. Users can optionally provide a text prompt to guide the analysis. Gemini Pro Vision analyzes the uploaded image and generates a comprehensive nutritional breakdown based on the prompt.

4.2 Health View Methodology

* UI: Streamlit creates the user interface for file upload, text input, and displaying results.
* Image Processing: Image library from Pillow is potentially used to open and display the uploaded image.
* Gemini Pro Vision Integration:
  + get\_gemini\_repsonse function prepares image data and sends it along with the prompt to Gemini Pro Vision.
  + A prompt template likely guides Gemini Pro Vision to:
    - Identify food items in the image.
    - Estimate the quantity of each item.
    - Calculate total calorie count.
    - Analyze vitamin and mineral content.
    - Assess the overall healthiness of the food.
    - Provide a breakdown of macronutrients (carbs, fats, fibers, sugar).

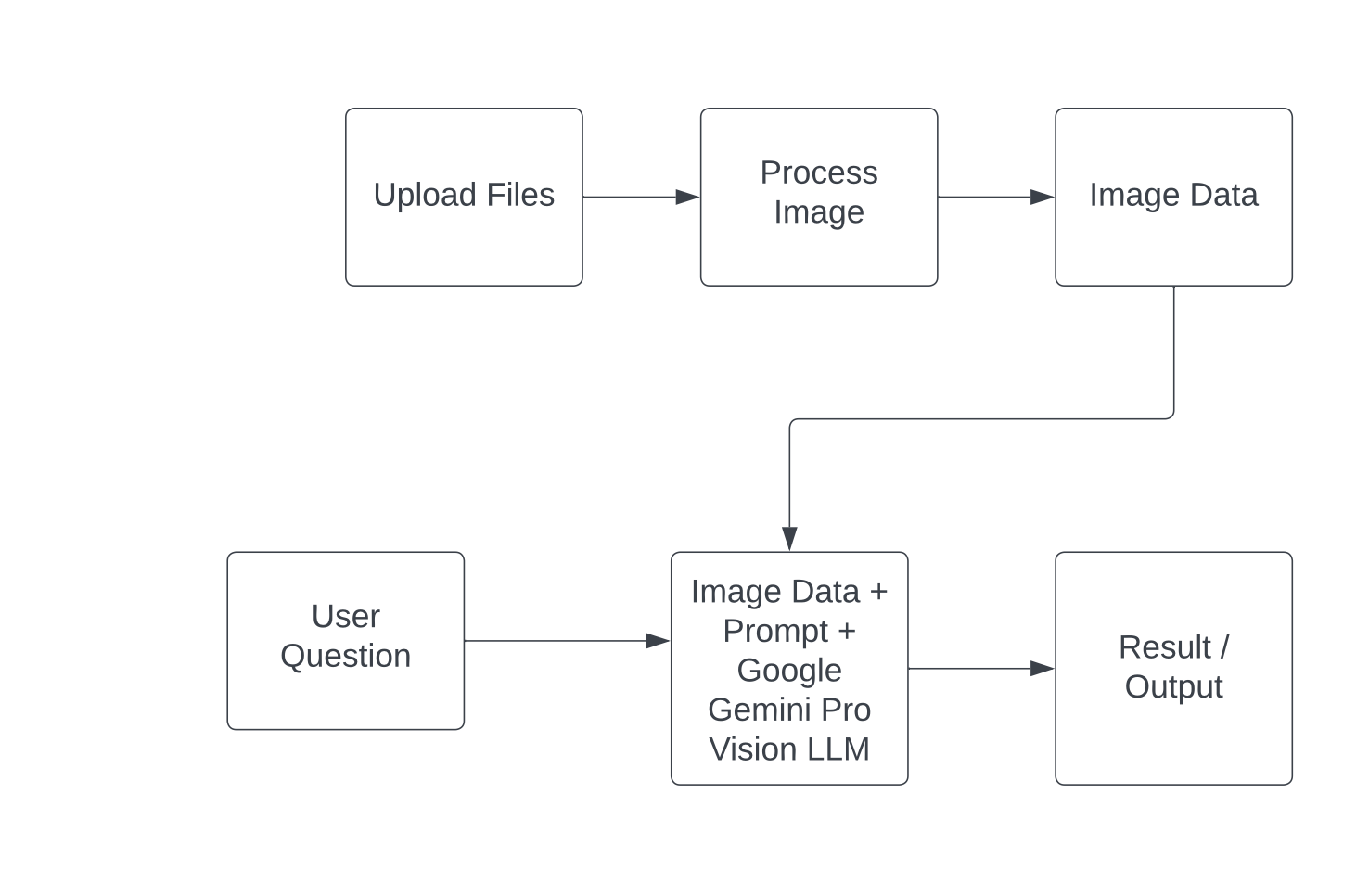
gl

Figure-4: Flow chart of Health view application

4.3 Explanation of input\_image\_setup function:

1. It checks if a file has been uploaded.
2. If a file exists, it reads the file content into bytes.
3. It creates a list of image parts with the following structure:
   * mime\_type: The MIME type of the uploaded file (e.g., "image/jpeg").
   * data: The image data in bytes.
4. If no file is uploaded, it raises a FileNotFoundError.

5.1 YouTube video transcriber & chat with video Application

This application creates detailed notes from YouTube videos using Gemini Pro. It also allows users to ask questions about the video content and get answers directly from the transcript, powered by Gemini Pro and LangChain.

5.2 YouTube video transcriber & chat with video Methodology

* YouTube Transcript API: Extracts video transcripts.
* Streamlit: Builds the user interface.
* Gemini Pro: Generates summaries and answers questions.
* LangChain: Facilitates conversational interactions with text-based content.
* FAISS: Enables efficient retrieval of relevant text sections for question-answering.

5.3 Workflow:

1. User enters a YouTube video link.
2. Application fetches the transcript using YouTubeTranscriptAPI.
3. User clicks "Get Detailed Notes" to initiate summarization.
4. Gemini Pro generates a summary based on the transcript and a prompt.
5. User can ask questions about the video content.
6. LangChain processes the question, leverages FAISS to find relevant transcript sections, and generates an answer using a Gemini Pro-based conversational chain.

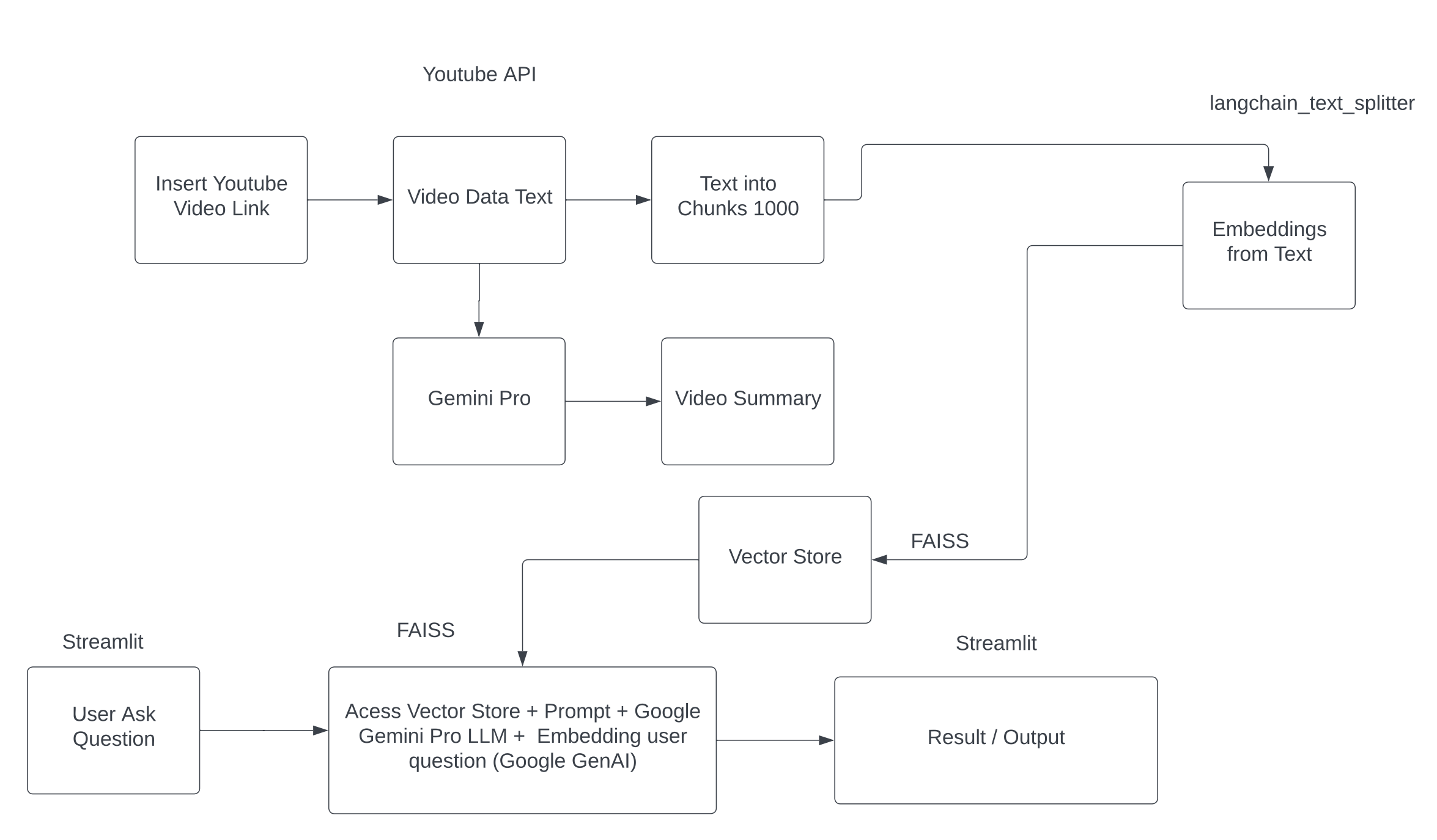


Figure-5: Flow chart of YouTube transcriber & chat with video

**RESULTS**

1. ATS Resume:

Input Job Description: [Job description](https://www.indeed.com/jobs?q=data+analyst&l=Chicago%2C+IL&from=searchOnHP&vjk=6e0c1288e02da36b) from here. + User Resume as input.

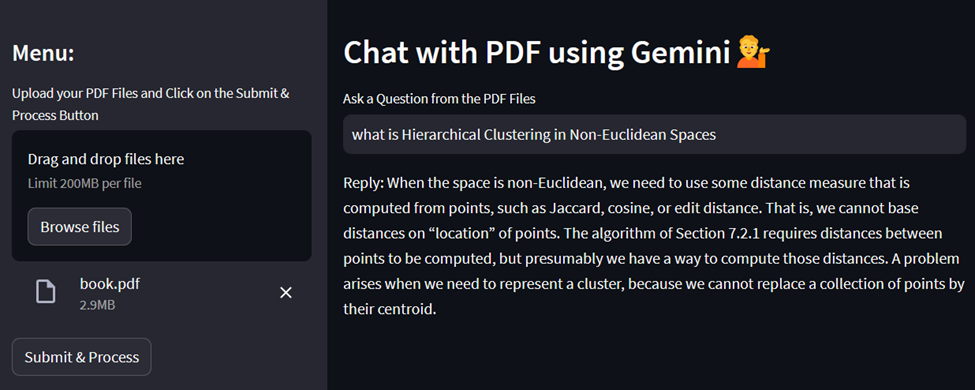
Output: {{"JD Match":"55%", "MissingKeywords":["Python", "Spark", "Tableau", "Agile", "Hadoop", "Hive", "Flume", "Elasticsearch", "Hadoop Distributed File System (HDFS)", "NoSQL"], "Profile Summary of the candidate":"The candidate has 8 years of experience working as a Data Engineer. They have experience in designing, developing, and maintaining data pipelines. They have a strong understanding of data engineering principles and technologies. The candidate is also proficient in SQL, Hive, HBase, and Pig. They have experience working with big data technologies such as Hadoop and Spark. The candidate is a skilled communicator and has a strong work ethic."}}

1. Chat with PDF’s:

Name of the input PDF: [Mining Of Massive Datasets - Stanford](http://infolab.stanford.edu/~ullman/mmds/book.pdf)

Input from the PDF: What is Hierarchical Clustering in Non-Euclidean Spaces

Output: When the space is non-Euclidean, we need to use some distance measure that is computed from points, such as Jaccard, cosine, or edit distance. That is, we cannot base distances on “location” of points. The algorithm of Section 7.2.1 requires distances between points to be computed, but presumably we have a way to compute those distances. A problem arises when we need to represent a cluster, because we cannot replace a collection of points by their centroid.



1. Text to SQL query generation:

Input: [Tesla stock data](https://www.kaggle.com/datasets/timoboz/tesla-stock-data-from-2010-to-2020) CSV file

And provide table name to store & input Text for operation: how many unique values in all columns

SQL generated by LLM:

SELECT

COUNT(DISTINCT Date) AS UniqueDate, COUNT(DISTINCT Open) AS UniqueOpen, COUNT(DISTINCT High) AS UniqueHigh, COUNT(DISTINCT Low) AS UniqueLow, COUNT(DISTINCT Close) AS UniqueClose, COUNT(DISTINCT Adj) AS UniqueAdj, COUNT(DISTINCT Volume) AS UniqueVolume

FROM tesla;

Output: (2416, 2132, 2128, 2136, 2225, 2225, 2391)

Input: Name the columns are in the table

Output: ('Date',) ('Open',) ('High',) ('Low',) ('Close',) ('Adj',) ('Volume',)

SQL query generated by LLM: SELECT name FROM pragma\_table\_info('tesla');

1. Health View App:

|  |  |  |
| --- | --- | --- |
| Input | LLM output | Original Count |
|  | Total Calories: 590  Carbohydrates: 45g  Fats: 35g  Protein: 20g  Fibers: 10g  Sugar: 5 g  Other: 5g | Total Calories: 664  Carbs: 7.8g  Fats: 51g  Protein: 43g  Fiber: 4.7g  Sugar: 1.4g |
|  | Total Calories: 540  Carbohydrates: 55g  Fats: 25g  Proteins: 25g  Fibers: 10g  Sugar: 10g  Other: 5g | Total Calories: 715  Carbs: 67g  Fats: 27g  Protein: 55g  Fiber: 12g  Sugar: 17g |
|  | Total Calories: 600  Carbohydrates: 75g  Fats: 25g  Proteins: 30g  Fibers: 12g  Sugar: 13g  Other: 5g | Total Calories: 803  Carbs: 105g  Fats: 23g  Protein: 47g  Fiber: 4.1g  Sugar: 26g |

Table 1: Health View application, comparison between LLM generated output to the Original Counts.

This Health View LLM model able to generate results of Nutrition count with 75% accuracy.

1. YouTube transcriber & chat with video:

Input: [YouTube Video link](https://www.youtube.com/watch?v=tYWQK0hR-Yk&t=6s)

Output: Video Summary:

This project aimed to assess ESG risk profiles across industries using machine learning. The analysis was based on a dataset with ESG metrics and financial data. Random Forest and XGBoost models were employed.

Key Findings:

* The Random Forest model identified ESG risk percentile as the most important predictor of overall ESG score.
* XGBoost model provided the best performance with lower Mean Absolute Error and Root Mean Squared Error.
* Key insights were gained on the relationship between ESG factors and risk profiles.
* Analysis revealed that the healthcare and energy sectors had higher ESG risk levels, while the technology sector had lower risk levels.
* Visualization techniques helped identify sectors with higher or lower ESG risk levels.
* Feature importance identified the most influential factors driving ESG risk scores.

Implications:

* Provides a comprehensive understanding of ESG risk profiles across industries.
* Aids investors and stakeholders in making informed decisions about ESG-related investments.
* Empowers companies to manage ESG risks more effectively.
* Highlights the potential of machine learning in enhancing ESG risk forecasting.

Follow up question from the video summary:

Input: what kind of visualization done

Output: The following visualizations were done:

Histogram, KD plot, Box plot, Scatter plot, Density plot, Score distribution

**CONCLUSION**

This research project successfully demonstrated the versatility and effectiveness of Google's Gemini Pro LLM across various natural language processing (NLP) applications. Through the development of five distinct applications, the study showcased the potential of Gemini Pro LLM in streamlining processes, providing valuable insights, and enhancing efficiency in diverse domains.

The resume evaluation application leveraged Gemini Pro LLM's language understanding capabilities to analyze resumes and job descriptions, providing insights into candidate suitability and job fit. The application's output, including the percentage match, missing keywords, and profile summaries, showcased the model's potential in streamlining the recruitment process and reducing workloads for human recruiters.

The PDF question answering application demonstrated Gemini Pro LLM's ability to comprehend and retrieve relevant information from unstructured text sources, such as PDF documents. By analyzing the PDF content and answering user queries accurately, this application highlighted the model's potential in facilitating efficient information retrieval and knowledge extraction.

The SQL query generation application addressed the challenge of democratizing data access and analysis by enabling users without extensive SQL knowledge to interact with databases through natural language queries. By generating SQL queries based on user input, Gemini Pro LLM demonstrated its ability to bridge the gap between human language and structured data, empowering users to perform data retrieval and analysis tasks efficiently.

The YouTube video transcriber & chat with video application highlighted Gemini Pro LLM's ability to process and summarize lengthy transcripts, presenting concise summaries of video content. This application's output, including key findings, implications, and visualizations, demonstrated the model's potential in facilitating efficient comprehension and knowledge extraction from multimedia sources.

The health analysis application showcased Gemini Pro LLM's vision capabilities by analyzing food images and providing detailed nutritional information, such as calorie counts, vitamin and mineral content, and overall healthiness assessments. This application demonstrated the model's potential in promoting informed dietary choices and supporting a healthier lifestyle.

While the applications exhibited promising results, it is essential to acknowledge the challenges and limitations associated with LLMs, such as prompt design, output quality, computational resource requirements, and ethical considerations. Continuous research efforts are crucial to address these challenges and further enhance the performance and reliability of LLM-based applications.

Overall, this research project has successfully demonstrated the potential of Gemini Pro LLM in reducing workloads and achieving great results in less time across various NLP tasks. The developed applications provide a foundation for further exploration and implementation of LLMs in various domains, paving the way for more efficient and insightful solutions to complex problems.

**Limitation:**

ATS Resume Application: it takes only pdf type files, and it supports only English.

Chat with PDF’s: it takes only pdf type files and it supports only English language. It can’t process mathematics and no mathematics related questions allowed. Multiples files size should be less than 200MB.

Text to SQL query generation: Only one operation at a time. No rollback after executing the command. Operation not allowed on 2 tables in the database at a time. No primary and foreign key concept in the application.

Health View Application: Here we are not training the model, so if the input image is not been trained by the google Gemini developer, then we can’t ask that. Accuracy of weight the food in the image is 75% accurate. The image should be clear.

YouTube transcriber & chat with video: it supports only English language videos and YouTube video should have caption setting enabled by that YouTube API can get text from the videos. User questions should be from the context only.

**Future Work:**

1. ATS Resume:

* Expand support for multiple file formats beyond PDFs, such as Word documents or plain text files.
* Incorporate natural language processing (NLP) techniques to handle resumes in languages other than English, enabling a more diverse and inclusive resume evaluation process.

2. Chat with PDF’s:

* Enhance the system's capabilities to process and understand mathematical equations, formulas, and symbols, enabling question-answering for technical and scientific documents.
* Implement techniques to handle larger file sizes, allowing the system to process and index multiple PDFs or documents simultaneously without size restrictions.
* Explore the integration of optical character recognition (OCR) technologies to support PDFs with scanned text or images, further expanding the system's applicability.

3. Text to SQL query generation:

* Extend the system to support multiple operations in a single query, enabling more complex data retrieval and manipulation tasks.
* Implement rollback functionality to ensure data integrity and provide a mechanism for undoing changes in case of errors or unintended operations.
* Incorporate support for primary and foreign key relationships, enabling more robust and efficient database operations while maintaining data consistency.
* Explore the possibility of handling multiple tables simultaneously, allowing users to query and join data from different tables through natural language input.

4. Health View:

* Investigate transfer learning techniques to fine-tune the language model on domain-specific data, such as food images and nutritional information, improving the accuracy and relevance of the analysis.
* Enhance the image processing capabilities to improve the accuracy of weight estimation for food items, potentially leveraging computer vision techniques or additional training data.
* Explore multimodal input scenarios, combining text and image inputs seamlessly, to provide more comprehensive analysis and recommendations.

5. YouTube transcriber & chat with video:

* Extend support for multiple languages, enabling the system to process and summarize videos in various languages, promoting accessibility and inclusivity.
* Investigate techniques to handle videos without captions or transcripts, potentially leveraging automatic speech recognition (ASR) technologies to generate transcripts from audio.
* Enhance the system's ability to understand and answer questions beyond the provided context, potentially by integrating external knowledge sources or information retrieval mechanisms.

Additionally, cross-cutting improvements could include:

* Exploring techniques to improve the interpretability and explainability of the language model's outputs, enhancing transparency and trust in the system's decisions and recommendations.
* Implementing robust error handling and user feedback mechanisms to improve the overall user experience and assist in identifying potential issues or areas for improvement.
* Continuously monitoring and addressing potential biases in the language model's outputs, ensuring fairness and ethical considerations are upheld in the applications.
* Investigating deployment strategies and optimizations for efficient and scalable hosting of the applications, enabling wider accessibility and adoption.

**CONTRIBUTIONS**

Bharath’s Contribution:

I leveraged powerful libraries and frameworks such as Google Generative AI (google.generativeai) for text generation, SQLite3 (sqlite3) for database interactions, YouTube Transcript API (youtube\_transcript\_api) for fetching video transcripts, and Langchain (langchain) for building question-answering systems with Retrieval Augmented Generation (RAG). My architectural design skills were instrumental in determining the appropriate Large Language Model (LLM) to solve the core problems and devising the overall approach to address the problem statements. I effectively managed the team, ensuring alignment with project goals and timely submission of updates to the professor. Additionally, I developed and verified robust applications, leveraging functions like RecursiveCharacterTextSplitter from Langchain for efficient text processing, and resolved any errors encountered during development. My contributions enabled the creation of applications that harness the power of natural language processing and large language models to solve complex problems, such as generating summaries, answering context-based questions, and extracting insights from various data sources.

Furthermore, I took on the responsibility of handling critical sections of the final report, including:

1. Limitations: I thoroughly analyzed and documented the limitations of each application, providing insights into areas such as file format support, language compatibility, mathematical processing capabilities, file size restrictions, and model training limitations.
2. Future Work: Based on the identified limitations, I proposed comprehensive future work suggestions, outlining potential improvements, enhancements, and research directions to address the current limitations and expand the capabilities of the applications.
3. Results and Conclusions: I evaluated the performance and outcomes of the developed applications, presenting the results and drawing meaningful conclusions regarding their effectiveness, strengths, and areas for improvement.
4. Abstract and Keywords: I crafted a concise and informative abstract that effectively summarizes the research objectives, methodologies, and key findings. Furthermore, I identified and compiled a comprehensive list of relevant keywords to facilitate indexing and discoverability of the research work.

Through these contributions, I played a pivotal role in ensuring the quality, completeness, and academic rigor of the final report, while also showcasing my technical expertise in developing innovative applications leveraging the power of large language models and natural language processing techniques.

Eddie’s Contribution:

As a key member of the team, I took charge of several critical components, including the implementation of essential Python libraries such as os and PIL, which facilitated seamless interaction with the operating system and enabled sophisticated image processing functionalities. Additionally, my expertise in leveraging the (langchain\_google\_genai) library was instrumental in developing the ChatGoogleGenerativeAI function, which served as the backbone for creating interactive chat interfaces powered by Google's generative AI models. This function played a central role in generating text responses based on user prompts or questions, enhancing the overall user experience. Furthermore, I spearheaded the integration of the (langchain.chains.question\_answering) library, particularly the load\_qa\_chain function, which streamlined the question-answering process by combining various functionalities such as text splitting, embedding generation, retrieval, and answer generation into a cohesive workflow. Moreover, my involvement in developing prompt templates using the (langchain.prompts) library ensured that the generative AI models received clear instructions and context, resulting in more accurate and informative responses. Beyond these technical contributions, I played a pivotal role in crafting the Introduction, Methodology, and Literature Review sections of the Final Report, providing valuable insights and context to the project's objectives and methodologies. Additionally, I meticulously filled out the slides for the Final Presentation, ensuring that they effectively communicated our project's goals, progress, and outcomes.

Vikhyath’s Contribution:

I designed interactive web apps specifically for data analysis using Streamlit, a strong and adaptable framework. Developers can easily transform data scripts into shareable web apps with Streamlit. I also designed PyPDF2, a package that offers Python users the ability to read from and write PDF files. When conducting batch actions on PDF files, such text extraction, merging, splitting, and other processes, this can be quite helpful. I worked on integrating the FAISS library, a tool for efficiently searching for similarities and grouping dense vectors. When retrieving similar things quickly is needed in large-scale machine learning applications, it is extremely helpful, and I created a feature that uses Google's generative AI models to generate text embeddings. Various tasks related to natural language processing are made possible by these embeddings, which convert textual input into a high-dimensional space where semantic similarity corresponds to proximity. I meticulously reviewed and updated the literature review section, ensuring that it accurately reflects the current state of research in the field and provides a comprehensive overview of relevant studies and publications. Creating detailed flowcharts that describe each application's data preprocessing pipeline was my main contribution to the project. These flowcharts ensured compliance with the Gemini LLM by outlining each stage of the process, from data collecting to final preprocessing. I was a key player in making sure that the preprocessing was clear and effective, which helped us successfully integrate many data sources and get the best performance out of the LLM models. Additionally, I curated and organized the references, adhering to the appropriate citation style and academic standards.

References:

1. Ai, Qingyao, et al. "Information Retrieval Meets Large Language Models: A Strategic Report from Chinese IR Community." 2023. arXiv:2307.09751 [cs.IR].
2. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big?. In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (pp. 610-623).
3. Charlotin, Damien, Large Language Models and the Future of Law (August 22, 2023). Available at SSRN: https://ssrn.com/abstract=4548258 or http://dx.doi.org/10.2139/ssrn.4548258
4. Deroy, Aniket, and Subhankar Maity. "Questioning Biases in Case Judgment Summaries: Legal Datasets or Large Language Models?." arXiv preprint arXiv:2312.00554 (2023).
5. Feng, Zhaopeng, et al. "Improving LLM-based Machine Translation with Systematic Self-Correction." arXiv:2402.16379v2 [cs.CL], March 4, 2024. [CC BY 4.0].
6. Gan, Chengguang, Qinghao Zhang, and Tatsunori Mori. "Application of llm agents in recruitment: A novel framework for resume screening." arXiv preprint arXiv:2401.08315 (2024).
7. Gemini Team et al., "Gemini: A Family of Highly Capable Multimodal Models" (2023), arXiv:2312.11805 [cs.CL].
8. Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in Neural Information Processing Systems 33 (2020): 9459-9474.
9. Li, Haoyang, Jing Zhang, Hanbing Liu, Ju Fan, Xiaokang Zhang, Jun Zhu, Renjie Wei, Hongyan Pan, Cuiping Li, and Hong Chen. "CodeS: Towards Building Open-source Language Models for Text-to-SQL." arXiv preprint arXiv:2402.16347 (2024).
10. Lyu, C., Tan, L., & Liu, Y. (2020). Abstractive text summarization with large language models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (pp. 5678-5692).
11. Mezgec, Simon, and Barbara Koroušić Seljak. "NutriNet: a deep learning food and drink image recognition system for dietary assessment." Nutrients 9, no. 7 (2017): 657.
12. P. V. Yadav, U. S. Kollimath, T. V. Chavan, D. T. Pisal, S. A. Giramkar and S. M. Swamy, "Impact of Artificial Intelligence (AI) in Talent Acquisition Process: A study with reference to IT Industry," 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bengaluru, India, 2023, pp. 885-889, doi: 10.1109/IITCEE57236.2023.10090973.
13. Skondras, Panagiotis, Panagiotis Zervas, and Giannis Tzimas. 2023. "Generating Synthetic Resume Data with Large Language Models for Enhanced Job Description Classification" Future Internet 15, no. 11: 363. https://doi.org/10.3390/fi15110363
14. Team, Gemini, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut et al. "Gemini: a family of highly capable multimodal models." arXiv preprint arXiv:2312.11805 (2023).
15. Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, Slav Petrov; Natural Questions: A Benchmark for Question Answering Research. Transactions of the Association for Computational Linguistics 2019; 7 453–466
16. Troy, Christopher, Sean Sturley, Jose Maria Alcaraz-Calero and Qi Wang. “Enabling Generative AI to Produce SQL Statements: A Framework for the Auto- Generation of Knowledge Based on EBNF Context-Free Grammars.” IEEE Access 11 (2023): 123543-123564.
17. Van, Minh-Hao, Prateek Verma, and Xintao Wu. "On Large Visual Language Models for Medical Imaging Analysis: An Empirical Study." arXiv preprint arXiv:2402.14162 (2024).
18. Vybhavi, Atluri Naga Sai Sri, Laggisetti Valli Saroja, Jahnavi Duvvuru, and Jayanag Bayana. "Video transcript summarizer." In 2022 International mobile and embedded technology conference (MECON), pp. 461-465. IEEE, 2022.
19. Wahl, Brian, Aline Cossy-Gantner, Stefan Germann, and Nina R. Schwalbe. "Artificial intelligence (AI) and global health: how can AI contribute to health in resource-poor settings?." BMJ global health 3, no. 4 (2018): e000798.
20. Wang, X., Ren, X., & Lin, J. (2022). Medical image analysis with multimodal large language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1234-1243).
21. Weidinger, Laura, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng et al. "Ethical and social risks of harm from language models." arXiv preprint arXiv:2112.04359 (2021).
22. Yan, Lixiang, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez‐Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. "Practical and ethical challenges of large language models in education: A systematic scoping review." British Journal of Educational Technology 55, no. 1 (2024): 90-112
23. Zhang, X., Li, J., & Wang, H. (2021). Resume evaluation using large language models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 1234-1248).
24. Zhu, Yutao, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. "Large language models for information retrieval: A survey." arXiv preprint arXiv:2308.07107 (2023).