VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi-590018, Karnataka

PROJECT REPORT

ON

"DELINEATE & DECIPHER: A RAG-POWERED AI PLATFORM FOR RESEARCH PAPER ANALYSIS AND VISUAL MATH PROBLEM SOLVING"

Submitted in partial fulfillment of requirements for the award of degree

BACHELORS OF ENGINEERING IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by:

ROHAN MALLICK

(1SB21CS089)

Under the Guidance of

Dr. MAHESH A

Professor, Department of Computer Science & Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRI SAIRAM COLLEGE OF ENGINEERING

ANEKAL, BENGALURU - 562106

ACADEMIC YEAR: 2024-25

SRI SAIRAM COLLEGE OF ENGINEERING

Anekal, Bengaluru – 562106

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work, entitled **"DELINEATE & DECIPHER : A RAG-POWERED AI PLATFORM FOR RESEARCH PAPER ANALYSIS AND VISUAL MATH PROBLEM SOLVING"** is a bona-fide work carried out by

1. ROHAN MALLICK	**1SB21CS089**

in partial fulfillment for the award of degree of Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University, Belagavi during the academic year 2024-25. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. This project report has been approved as it satisfies the academic requirements with respect to the project work prescribed for the Bachelor of Engineering Degree.

Signature of the Internal Guide	Signature of the HOD	Signature of the Principal
Dr. Mahesh A	**Dr. Smitha J A**	**Dr. B Shadaksharappa**
Professor,	HOD,	Principal
Dept. of CSE	Dept. of CSE	

Name of examiners: Signature with date

1.

2.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance. So, with gratitude we acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

We take this opportunity to thank Our Chairman and Chief Executive Officer **Dr. Sai Prakash LeoMuthu**, and **Dr. Arun Kumar R**, Management Representative of Sri Sairam College of Engineering for providing us with excellent infrastructure that is required for the development of our project.

We are thankful to our Principal, **Dr. B. Shadaksharappa** and for their encouragement and support throughout the project work.

We are also thankful to our beloved HOD, **Dr Smitha J. A** for her incessant encouragement & all the help during the project work.

We take this opportunity to thank our Project Coordinators, **Dr. Sumathi P,** Assistant Professor, Dept. of CSE for their inspirational guidance, valuable suggestions and providing us a chance for the completion of the Project.

We consider it a privilege and honor to express our sincere gratitude to our guide **Dr. Mahesh A,** Professor, Dept. of CSE for his valuable guidance throughout the tenure of this project work, and whose support and encouragement made this work possible.

It is also a great pleasure to express our deepest gratitude to all the other faculty members of our department for their cooperation and constructive criticism offered, which helped us a lot during our project work.

Finally, we would like to thank all our family members and friends whose encouragement and support was invaluable.

SRI SAIRAM COLLEGE OF ENGINEERING
Anekal, Bengaluru – 562106

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I, **ROHAN MALLICK,** hereby declare that the entire work titled **"DELINEATE & DECIPHER: A RAG-POWERED AI PLATFORM FOR RESEARCH PAPER ANALYSIS AND VISUAL MATH PROBLEM SOLVING"** embodied in this project report has been carried out by me during the 7th semester of BE degree at SSCE, Bangalore under the esteemed guidance of **Dr. MAHESH A**, Professor, Dept. of CSE, Sri Sairam College of Engineering, Bengaluru affiliated to Visvesvaraya Technological University, Belagavi. The work embodied in this dissertation work is original and it has not been submitted in part or full for any other degree in any University.

1. ROHAN MALLICK (1SB21CS089) ______________________________

ABSTRACT

The exponential growth in the research papers and complex mathematical problems has posed serious challenges in information retrieval and its review for analysis. This paper presents "Delineate and Decipher," a platform powered by AI using Retrieval-Augmented Generation models to solve these challenges. The platform will combine natural language processing with machine learning techniques to provide context-based answers from research papers and also solve visual math problems through its interactive interface. A hybrid system was based on the Gemini model of visual problem-solving in mathematics, whereas the retrieval models were powered with GROQ. The platform embeds both documents and mathematical equations into vector spaces, hence guaranteeing efficiency in document searching and retrieval. Precisely, all this necessary information has kept in mind while making the system function over huge datasets of research papers by splitting documents into smaller chunks with strong results based on accurate/context retrieval. It also allows the user to write mathematical equations visually and get solutions to them using real-time API with Gemini Flash, hence not just a tool for doing research analysis but also educational purposes. This work saves time for the researchers and students who search for some information and solutions of cumbersome mathematics problems. The proposed platform is powered by state-of-the-art AI techniques: FAISS for vector storage and Google Generative AI embeddings for document representation. It provides better access, precision, and interactivity to academic research and education. Future versions might involve even better language models and more comprehensive datasets, which have the interactions between users and scholarly materials and mathematical tools.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

SYSTEM ANALYSIS

CHAPTER 1
SYSTEM ANALYSIS

1.1 Introduction

The rapid expansion of academic research presents a significant challenge for researchers, educators, and students in retrieving relevant information efficiently. Traditional methods of information retrieval struggle under the growing volume of research, often leading to time consuming, error-prone processes that hinder productivity and thoroughness (Gupta & Rani, 2019) [1]. Additionally, as mathematical problem-solving becomes increasingly complex, existing manual and traditional methods fall short of providing timely and accurate solutions, necessitating the application of advanced machine learning models to streamline this process (Li, Zhao, & Ma, 2021) [2]. However, while these models have made strides in automating mathematical problem-solving, they still face limitations in handling the intricacies of complex equations. Furthermore, advancements in natural language processing (NLP) for academic research retrieval have shown promising improvements in accessing and analysing scholarly materials, yet there remains room for enhancing the precision and interactivity of these tools (Wang & Yu, 2020) [3].

Delineate and Decipher answer such demands by using state-of-the-art AI called RAG. RAG is an integration of retrieval-based models with generation-based models for generating text. This parallels two scopes: one that concerns the analyses of research papers and another that pertains to the solution of visual math problems. This platform focuses on doing research paper analysis and solving visual math problems. By using a document-based Q&A enabled by RAG and solving math problems with visual inputs facilitated by Gemini API, this platform will assure speed, accuracy, and friendliness of the solution.

The quest for reliable and relevant information in research is essential for drawing sound conclusions that drive further studies. However, traditional search engines often fall short in understanding the nuanced context behind user queries, creating a disconnect between user intentions and the results displayed, thereby limiting efficient access to precise information (Gupta & Rani, 2019) [1]. This gap not only hampers researchers' productivity but also increases the potential for oversight. Similarly, solving mathematical problems—especially those represented visually—requires tools that go beyond basic calculators or symbolic computation software. Current solutions are often insufficient for handling diverse input formats and complex equations in a user-friendly manner, making it challenging for students and researchers to obtain accurate, real-time solutions (Li, Zhao, & Ma, 2021) [2]. Thus, there is a pressing need for advanced systems that bridge these gaps, providing efficient, context-aware information retrieval and versatile math problem-solving capabilities (Wang & Yu, 2020) [3]

1.2 Literature Survey

REF [1] Generation of Highlights From Research Papers Using Pointer-Generator Networks and SciBERT Embeddings.

Nowadays many research articles are prefaced with research highlights to summarize the main findings of the paper. Highlights not only help researchers precisely and quickly identify the contributions of a paper, they also enhance the discoverability of the article via search engines. We aim to automatically construct research highlights given certain segments of a research paper. We use a pointer-generator network with coverage mechanism and a contextual embedding layer at the input that encodes the input tokens into SciBERT embeddings. We test our model on a benchmark dataset, CSPubSum, and also present MixSub, a new multi-disciplinary corpus of papers for automatic research highlight generation. For both CSPubSum and MixSub, we have observed that the proposed model achieves the best performance compared to related variants and other models proposed in the literature. On the CSPubSum dataset, our model achieves the best performance when the input is only the abstract of a paper as opposed to other segments of the paper. It produces ROUGE-1, ROUGE-2 and ROUGE-L F1-scores of 38.26, 14.26 and 35.51, respectively, METEOR score of 32.62, and BERTScore F1 of 86.65 which outperform all other baselines. On the new MixSub dataset, where only the abstract is the input, our proposed model (when trained on the whole training corpus without distinguishing between the subject categories) achieves ROUGE-1, ROUGE-2 and ROUGEL F1-scores of 31.78, 9.76 and 29.3, respectively, METEOR score of 24.00, and BERTScore F1 of 85.25

Advantages:

- The pointer-generator network produces concise and coherent research highlights
- The coverage mechanism ensures minimal repetition in the generated summaries.
- SciBERT embeddings enhance the model's ability to process scientific language effectively.
- Outperforms existing methods on multiple metrics, ensuring high-quality highlights.

Disadvantages:

- Performance is dependent on training data quality; new domains may require additional fine-tuning.
- The model achieves better performance with abstracts but struggles with other segments of a paper.
- Results on MixSub indicate room for improvement in multi-disciplinary highlight generation.

REF [2] RAVEN: In-Context Learning with Retrieval-Augmented Encoder-Decoder Language Models

In this paper, we investigate the in-context learning ability of retrieval augmented encoder-decoder language models. We first conduct a comprehensive analysis of existing models and identify their limitations in in-context learning, primarily due to a mismatch between pretraining and inference, as well as a restricted context length. To address these issues, we propose RAVEN, a model that combines retrieval-augmented masked language modeling and prefix language modeling. We further introduce Fusion-in-Context Learning to enhance the few-shot performance by enabling the model to leverage more in-context examples without requiring additional training. Through extensive experiments, we demonstrate that our simple yet effective design significantly improves performance, achieving results comparable to the most advanced language models in certain scenarios, despite having substantially fewer parameters. Our work underscores the potential of retrieval-augmented encoder-decoder language models for in-context learning and encourages further research in this direction

REF [3] Active Retrieval Augmented Generation

Despite the remarkable ability of large language models (LMs) to comprehend and generate language, they have a tendency to hallucinate and create factually inaccurate output. Augmenting LMs by retrieving information from external knowledge resources is one promising solution. Most existing retrieval augmented LMs employ a retrieve-and-generate setup that only retrieves information once based on the input. This is limiting, however, in more general scenarios involving generation of long texts, where continually gathering information throughout generation is essential. In this work, we provide a generalized view of active retrieval augmented generation, methods that actively decide when and what to retrieve across the course of the generation. We propose Forward-Looking Active REtrieval augmented generation (FLARE), a generic method which iteratively uses a prediction of the upcoming sentence to anticipate future content, which is then utilized as a query to retrieve relevant documents to regenerate the sentence if it contains low-confidence tokens. We test FLARE along with baselines comprehensively over 4 longform knowledge-intensive generation tasks/- datasets. FLARE achieves superior or competitive performance on all tasks, demonstrating the effectiveness of our method.

Advantages:

- Continuously retrieves information, ensuring factual accuracy in long-form text generation.
- Predicting upcoming content allows proactive retrieval, reducing reliance on static inputs.
- By regenerating low-confidence sentences, FLARE mitigates hallucinations and improves factual correctness.

Disadvantages:

- The iterative retrieval and regeneration process demands more computational resources than static RAG methods.
- Performance is contingent on the quality and relevance of retrieved documents.
- Forward-looking predictions introduce additional model complexity, requiring fine-tuning and integration efforts.

REF [4] Contextual Compression in Retrieval-Augmented Generation for Large Language Models: A Survey

Large Language Models (LLMs) showcase remarkable abilities, yet they struggle with limitations such as hallucinations, outdated knowledge, opacity, and inexplicable reasoning. To address these challenges, Retrieval-Augmented Generation (RAG) has proven to be a viable solution, leveraging external databases to improve the consistency and coherence of generated content, especially valuable for complex, knowledge-rich tasks, and facilitates continuous improvement by leveraging domainspecific insights. By combining the intrinsic knowledge of LLMs with the vast, dynamic repositories of external databases, RAG achieves a synergistic effect. However, RAG is not without its limitations, including a limited context window, irrelevant information, and the high processing overhead for extensive contextual data. In this comprehensive work, we explore the evolution of Contextual Compression paradigms, providing an in-depth examination of the field. Finally, we outline the current challenges and suggest potential research and development directions, paving the way for future advancements in this area.

REF [5] Retrieval-Augmented Generative Agent for Scientific Research

Large Language Models (LLMs) generalize well across language tasks, but suffer from hallucinations and uninterpretability, making it difficult to assess their accuracy without ground-truth. Retrieval-Augmented Generation (RAG) models have been proposed to reduce hallucinations and provide provenance for how an answer was generated. Applying such models to the scientific literature may enable large-scale, systematic processing of scientific knowledge. We present PaperQA, a RAG agent for answering questions over the scientific literature. PaperQA is an agent that performs information retrieval across full-text scientific articles, assesses the relevance of sources and passages, and uses RAG to provide answers. Viewing this agent as a question-answering model, we find it exceeds performance of existing LLMs and LLM agents on current science QA benchmarks. To push the field closer to how humans perform research on scientific literature, we also introduce LitQA, a more complex benchmark that requires retrieval and synthesis of information from full-text scientific papers across the literature. Finally, we demonstrate PaperQA's matches expert human researchers on LitQA.

Advantages:

- Integrates retrieved content to ensure evidence-based answers, addressing LLM hallucination issues.
- Provides source references for generated answers, enhancing user trust and transparency.
- Facilitates large-scale, systematic analysis of scientific literature, which is valuable for researchers.
- Outperforms current models on existing benchmarks and demonstrates parity with human expertise on LitQA.

Disadvantages:

- The retrieval, relevance assessment, and synthesis steps add layers of complexity compared to standard LLMs.
- The accuracy of answers is reliant on the quality and completeness of retrieved scientific articles.
- Processing full-text articles and synthesizing information requires significant computational resources.

REF [6] Benchmarking Large Language Models in Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a promising approach for mitigating the hallucination of large language models (LLMs). However, existing research lacks rigorous evaluation of the impact of retrieval-augmented generation on different large language models, which make it challenging to identify the potential bottlenecks in the capabilities of RAG for different LLMs. In this paper, we systematically investigate the impact of Retrieval-Augmented Generation on large language models. We analyze the performance of different large language models in 4 fundamental abilities required for RAG, including noise robustness, negative rejection, information integration, and counterfactual robustness. To this end, we establish Retrieval-Augmented Generation Benchmark (RGB), a new corpus for RAG evaluation in both English and Chinese. RGB divides the instances within the benchmark into 4 separate testbeds based on the aforementioned fundamental abilities required to resolve the case. Then we evaluate 6 representative LLMs on RGB to diagnose the challenges of current LLMs when applying RAG. Evaluation reveals that while LLMs exhibit a certain degree of noise robustness, they still struggle significantly in terms of negative rejection, information integration, and dealing with false information.

1. 3 Problem Statement

The exponential growth in the volume of research papers and the increasing complexity of mathematical problems pose significant challenges for researchers, educators, and students alike. The sheer scale of academic literature makes it difficult to retrieve relevant information efficiently, while traditional search tools often fail to provide context-aware and precise answers. This creates a time-consuming and fragmented process for conducting research and gathering insights from scholarly materials.

In parallel, solving complex mathematical problems, particularly visual equations that demand interactive and real-time computations, remains a significant hurdle. Existing tools often require users to manually input equations in rigid formats, leading to inefficiencies and limited accessibility for non-experts or those seeking user-friendly solutions. Moreover, the absence of a unified platform that integrates document analysis and advanced mathematical problem-solving restricts productivity and impedes innovation in academic and educational domains.

These challenges underscore the need for a comprehensive solution that not only enhances the efficiency of information retrieval but also provides seamless support for interactive, visual mathematical computations. Addressing these issues is critical to saving time, improving accuracy, and streamlining workflows for researchers, educators, and students working with complex datasets and equations.

1.4 Objective of the project

The objective of this project is an AI-powered platform that integrates advanced Retrieval-Augmented Generation (RAG) models to address the challenges of academic information retrieval and visual mathematical problem-solving. The platform aims to:

- Provide precise, context-aware answers from extensive datasets of research papers through efficient natural language processing and machine learning techniques.

- Enable users to interactively input and solve visual mathematical equations in real-time using advanced computation tools such as the Gemini model and Gemini Flash API.

- Embed both textual and mathematical data into vector spaces to enhance the accuracy and efficiency of document search and retrieval processes.

- Streamline workflows for researchers, educators, and students by offering a unified interface for academic research and mathematical analysis.

- Leverage state-of-the-art AI technologies, such as FAISS for vector storage and Google Generative AI embeddings, to optimize platform performance and user experience.

- Facilitate academic research and education by saving time, improving accuracy, and providing a seamless interaction with scholarly materials and mathematical tools.

The platform seeks to not only bridge the gap between information retrieval and mathematical problem-solving but also serve as a versatile tool for both research analysis and educational purposes, with future potential for enhanced capabilities through advanced language models and comprehensive datasets.

1.5 Proposed System

The Delineate and Decipher platform is developed to bridge the gap between research paper analysis and visual mathematical problem-solving, integrating RAG models with advanced visual recognition techniques. This tends to effectively solve the problems for both researchers and students by offering AI-powered answers for both textual queries from the research papers and complex mathematical equations

1. **RAG-Powered Research Paper Analysis**

According to the research paper, the system makes use of the RAG model that combines document retrieval and language generation into query answering. The sequence of the procedure goes as follows:

1.1. Upload Research Paper: The system allows uploading the research paper in PDF.

1.2. Create Embeddings: Other than uploading, a user needs to click the button saying "Create Embedding." It used to call FAISS-the Facebook AI similarity search-to create vector embeddings of the document; this enables fast retrieval of relevant sections based on user queries. Once the model is built, the user can insert specific questions related to the research paper.

1.3. Retriever: The RAG model retrieves the relevant sections of the document and uses a language generation model, such as llama, to produce an accurate, context-based response. This workflow minimizes latency and ensures answers are given in real time. The RAG model brings the best of retrieval-based and generation-based in producing adequate responses from user queries. This retrieval is done with FAISS, which embeds vectors of research papers stored in the platform, hence locating the relevant sections according to a user's question. After the relevant sections are retrieved, the response is articulated by the language generation model because it can generate coherent sentences that are contextually correct. This question-answer cycle is definitely geared towards minimizing latency to provide fast answers.

2. Math Visual Solver

Visual solver, on the other hand, uses the Gemini Flash API to detect visual inputs for solving mathematical problems. In particular, users can draw equations in such a way that these visual inputs are sent through a pipeline for visual recognition to convert the data into text-based equations. Examples will also include real-time performance-challenging equation solving, immediate feedback on standard algebraic expressions, calculus problems, and even multi-step mathematical proofs. Examples like these make the platform ideal for academic environments where students and researchers require accurate results quite immediately. The visual math problem solver is designed to handle a variety of complex equations, making it highly useful in both educational and research contexts. Here are some examples of the types of complex equations it can solve:

2.1. Non-linear Equations: Equations like quadratic, cubic, or higher-degree polynomials (e.g., $x^3+5x^2-x-6=0,x^3 + 5x^2 - x - 6 = 0,x^3+5x^2-x-6=0$) that require precise factorization or the use of numerical methods.

2.2. Differential Equations: Both ordinary and partial differential equations (e.g., $dy/dx+y*\sin(x)=x^2$) commonly encountered in engineering and physics, which the platform can solve symbolically or numerically.

2.3. Integral Calculus: Complex integrals such as definite and indefinite integrals (e.g., $\int(x^2+e^x)\,dx$ or $\int \sin(x)\,dx$), which often appear in advanced calculus.

2.4. System of Equations: Linear or non-linear systems, such as $2x+3y=52x + 3y = 52x+3y=5$ and $x^2+y^2=1$, where the solver finds solutions for multiple variables simultaneously.

2.5. Matrix Operations and Linear Algebra Problems: Problems involving matrices (e.g., eigenvalues and eigenvectors of a matrix) which are essential in data science and machine learning.

2.6. Fourier and Laplace Transforms: Used widely in signal processing, such as finding the Fourier transform of a complex function $f(x)=e^{-2x}$ or solving equations using Laplace transforms.

2.7. Multivariable Calculus: Partial derivatives and gradients for functions with multiple variables, such as $f(x,y)=x^2+3xy+y^2$, which are essential in fields like optimization and machine learning.

2.8. Differential Geometry Equations: Equations involving curves and surfaces, such as parameterized surfaces or curvature formulas (e.g., finding the curvature of a space curve given by $r(t)=(t,t^2,t^3)$))

2.9. Scenario Based Problems: The platform's visual math solver interprets equations handwritten or entered through an interactive interface shown in Fig. 2 a real-time drawing, applying advanced AI methods to provide accurate, real-time solutions to complex mathematical problems.

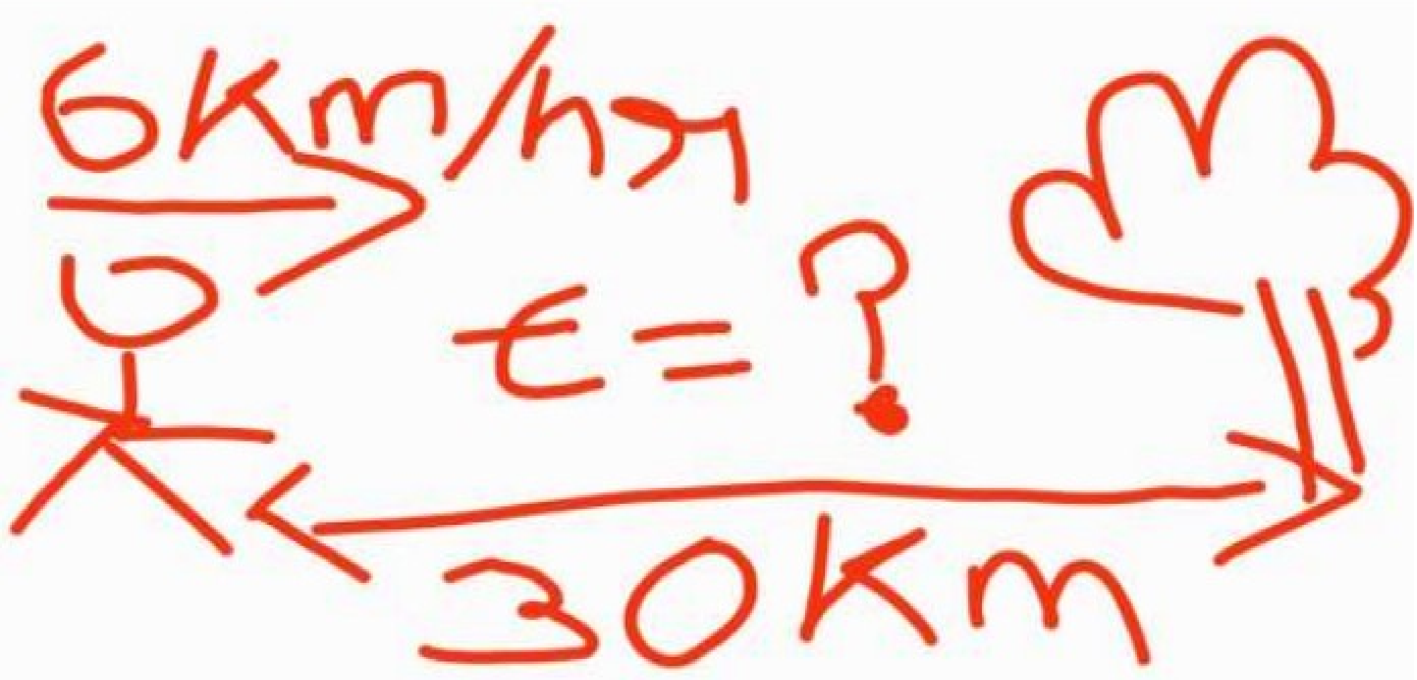


Figure 1.1:Visual Mathematical Scenario

CHAPTER 2

SYSTEM REQUIREMENTS SPECIFICATIONS

CHAPTER 2

SYSTEM REQUIREMENTS SPECIFICATION

The System Requirement Specification (SRS) is a crucial document that defines the core functionalities of a product. It captures the essential features and functionalities the system needs to deliver. Essentially, it acts as a contract between a company and a client (or potential client) outlining the system's requirements at a specific point in time, typically before any design or development work begins. This document serves as a two-way communication tool, ensuring both parties have a clear understanding of the system's needs from the same perspective at a given time.

Thorough System Requirement Specifications (SRS) minimize development effort. Carefully reviewing this report can uncover omissions, misunderstandings, and inconsistencies before the development cycle begins, when fixing them is much easier. The SRS focuses on the product's functionality, not its creation process. Therefore, it serves as a baseline for future modifications to the finished product.

While the System Requirement Specification (SRS) may need revisions, it provides a solid foundation for development evaluation. In simpler terms, defining software requirements is the first step in the product development process. The SRS acts as a translator, transforming the unrefined ideas of clients (the raw data) into a formal document (the output of this stage). Ideally, the SRS delivers a clear and unwavering set of requirements, while the initial client ideas may lack structure and consistency.

2.1 Software Requirements

This section outlines the hardware and software requirements for the Delineate and Decipher: A RAG Powered AI Platform for Research Paper Analysis and Visual Math Problem Solving. Notably, the application is designed to function efficiently on low-end devices, ensuring accessibility for a broader audience.

Operating system:

- **Windows:** Windows 10 or higher
- **macOS:** macOS 10.15 (Catalina) or higher
- **Linux:** Any modern distribution (e.g., Ubuntu 20.04 or higher)

Python Environment:

- **Python:** Version 3.8 or higher.
- **Required Python packages include:**
  □ streamlit
  □ langchain_groq
  □ langchain
  □ python-dotenv

1. **Development Tools:**
    - **Programming Language:** Python 3.x
    - **Integrated Development Environment (IDE):** PyCharm, Visual Studio Code, or Jupyter Notebook
2. **Version Control:**
    - **Git:** For source code management and version control

Node.js and npm (for React app development):

- **Node.js:** Version 14.x or higher.
- **npm:** Installed with Node.js.
- Frontend Libraries (for React app):

API Keys:

- **GROQ_API_KEY=**your_groq_api_key.
- **GOOGLE_API_KEY=**your_google_api_key.

Database (optional):

- Consider a lightweight database like Faiss or quadrant for persistent data storage.

2.2 Hardware Requirements

Processor:

- **Minimum:** Dual-core processor (e.g., Intel Core i3 or equivalent) for basic functionality.
- **Recommended:** Quad-core processor (e.g., Intel Core i5 or higher) for enhanced performance during document processing and analysis.

RAM:

- **Minimum:** 4 GB (allows for basic operations and document handling).
- **Recommended:** 8 GB or more (recommended for smoother performance with larger datasets and concurrent users).

Storage:

- **Minimum:** 5 GB of free disk space (sufficient for application installation and document storage)
- **Recommended:** Solid State Drive (SSD) for faster data access, particularly when working with larger files.

Network:

- Stable internet connection for API access (essential for Google Gemini and other external services).

2.3 Software Description

The **Delineate and decipher** project integrates various software components to deliver a robust and efficient system for calorie estimation from food images. Below is a detailed description of the core software components and tools used in the project:

Python:

Invented in 1989 by Guido Rossum, Python is an object-oriented programming language known for its speed in creating prototypes for complex applications. Beyond its core functionality, Python interfaces with various operating system calls and libraries, and even allows for extension into C or C++. Major companies like NASA, Google, YouTube, and BitTorrent all leverage Python's capabilities. Popular in fields like Artificial Intelligence, Natural Language Generation, and Neural Networks, Python prioritizes clear, readable code. This course will guide you through Python from its fundamental concepts.

Characteristics of Python

- Its syntax is clear and easier to understand compared to other languages.
- It provides a diverse range of data types to work with.
- Python scripts can run on various operating systems without modifications, making them versatile.
- It allows for dynamic changes during program execution, offering more adaptability.
- Python boasts libraries with functionalities like text manipulation (similar to Perl and Awk) that work seamlessly across Linux, Mac, and Windows.

React:

React is a popular open-source JavaScript library used for building user interfaces, particularly for single-page applications. It is based on a component-based architecture, where the UI is divided into reusable, self-contained units called components. This modular approach simplifies development and maintenance. React uses a Virtual DOM to optimize performance, updating only the parts of the page that need to be changed, rather than re-rendering the entire page. It employs a declarative syntax, meaning developers specify how the UI should look based on the current state, and React automatically updates the UI when the state changes. React also introduces JSX, a syntax extension that allows developers to write HTML-like code within JavaScript, making the code more readable and maintainable.

TypeScript:

TypeScript is a superset of JavaScript that introduces static typing and other features to improve code quality and maintainability. Developed by Microsoft, TypeScript provides developers with the ability to define types for variables, functions, and objects, which helps catch type-related errors during development rather than at runtime. This makes it particularly beneficial for large applications or teams working on complex projects. TypeScript also uses type inference, meaning it can automatically deduce the type of a variable based on its assigned value, which reduces the need for explicit type declarations in many cases. One of the core features of TypeScript is its support for interfaces, which allow developers to define custom types for objects, ensuring consistency and structure across the application. It also includes generics, which make it possible to write reusable code that works with any data type, providing both flexibility and type safety. TypeScript supports classes and inheritance, allowing developers to implement object-oriented programming principles like in traditional languages such as Java or C#. Furthermore, TypeScript provides robust tooling and IDE support, offering features like autocompletion, type checking, and inline documentation, making development smoother and more efficient.

Despite its many advantages, TypeScript does come with some drawbacks. Developers new to TypeScript might face a learning curve, especially with concepts like type annotations, interfaces, and generics. Additionally, TypeScript requires a compilation step to convert the TypeScript code into JavaScript, which can introduce additional overhead and slow down build times. For small projects, the added complexity of TypeScript might not always justify the benefits, especially if the project is relatively simple.

Overall, TypeScript is a powerful tool for managing large codebases, providing early error detection, better maintainability, and enhanced developer experience. Its strong typing system helps improve code clarity, refactoring, and debugging, making it an excellent choice for modern web development, particularly in large, complex applications built with frameworks like React, Angular, and Vue. TypeScript's compatibility with JavaScript allows it to be gradually adopted, making it a great option for both new projects and existing JavaScript codebases looking to improve their maintainability and scalability.

Node.js:

Node.js is a popular open-source, cross-platform runtime environment used for executing JavaScript code server-side. Unlike traditional JavaScript environments that run in the browser, Node.js enables developers to use JavaScript for both client-side and server-side scripting, making it possible to build entire applications using a single programming language. Built on the V8 JavaScript engine (the same engine used by Google Chrome), Node.js offers non-blocking, event-driven architecture, which makes it well-suited for handling I/O-heavy tasks, such as reading files, querying databases, or handling multiple network requests.

One of the standout features of Node.js is its asynchronous, event-driven model. Traditional server-side environments rely on multi-threading for handling requests, which can be resource-intensive. In contrast, Node.js uses a single thread and an event loop, where tasks such as file reading or network requests are handled asynchronously. This allows Node.js to perform highly concurrent operations without the need for multiple threads, improving scalability and performance, especially for real-time applications like chat applications or live data streaming.

Node.js also has a large ecosystem of libraries and modules available through its npm (Node Package Manager), which is one of the largest software registries in the world. Developers can install and manage third-party packages, greatly accelerating development time and enabling Node.js to be used for a wide variety of tasks, from simple web servers to complex enterprise applications.

Another key feature of Node.js is its cross-platform compatibility, allowing applications built in Node.js to run on various operating systems, such as Windows, macOS, and Linux, without modification. This makes it an attractive choice for developers working in diverse environments or deploying applications across multiple platforms.

However, while Node.js offers significant advantages, particularly in performance for I/O-heavy tasks, it does come with some challenges. Its single-threaded nature means that it might not be the best choice for CPU-intensive operations, like heavy computations or tasks that require intensive processing power. To handle such operations, developers may need to use worker threads or integrate Node.js with other languages that are better suited for CPU-bound tasks.

In conclusion, Node.js is a powerful runtime environment that allows developers to use JavaScript for server-side development, enabling fast, scalable, and efficient applications. Its non-blocking I/O model, large ecosystem of packages, and cross-platform capabilities make it an ideal choice for building real-time applications, APIs, and microservices.

StreamLit:

Streamlit is an open-source Python library used for building interactive and user-friendly web applications, particularly for data science, machine learning, and artificial intelligence projects. Streamlit is designed to help developers quickly create applications with minimal effort by allowing them to focus on Python code without worrying about the complexities of traditional web development. With just a few lines of code, users can create powerful dashboards, visualizations, and interactive features that enable users to interact with data and models in real-time.

One of the standout features of Streamlit is its simplicity and ease of use. Unlike other web frameworks that require HTML, CSS, or JavaScript, Streamlit allows developers to create fully functional web applications using only Python. By using a series of simple Python functions, such as st.write(), st.plotly_chart(), and st.text_input(), developers can quickly add interactive elements like tables, charts, sliders, and text boxes to their applications. This makes Streamlit a popular choice for data scientists and machine learning engineers who want to share their work with others without needing to become experts in web development.

Streamlit also integrates well with popular Python libraries such as Pandas, Matplotlib, Plotly, and TensorFlow, making it easy to visualize data and results directly in the web app. Whether you're building a data dashboard, machine learning model interface, or any other type of interactive tool, Streamlit's simple API makes it easy to integrate these libraries for real-time visualizations.

Another key feature of Streamlit is its real-time interactivity. Changes made by users, such as inputting values through text boxes or sliders, trigger updates to the visualizations or computations on the server side, providing immediate feedback. This makes Streamlit especially useful for prototyping and sharing data science models or experiments, as it allows users to interact with data and models on-the-fly.

Streamlit also supports deployment, enabling developers to easily share their applications with others. Developers can deploy Streamlit apps to the Streamlit Cloud, or they can host them on their own servers. The deployment process is straightforward, and Streamlit offers a simple interface for users to access and run the applications online.

While Streamlit is incredibly powerful for creating data-driven applications, it does have some limitations. As it focuses on simplicity, it may not be suitable for complex web applications that require extensive customization, user authentication, or full control over the user interface. However, for data-driven applications and prototypes, Streamlit excels in providing an efficient and user-friendly solution.

Vite:

Vite is a modern, fast, and lightweight build tool designed to enhance the development experience for web applications. Created by Evan You, the developer of Vue.js, Vite addresses the performance limitations of traditional bundlers like Webpack by using native ES modules and offering faster startup times. Vite's development server serves source code directly to the browser, bypassing the need for bundling during development, which drastically improves speed. This approach allows developers to see changes instantly, making the development cycle faster and more efficient. One of the key features of Vite is Hot Module Replacement (HMR), which provides near-instant feedback when changes are made, without needing to reload the entire page.

In addition to its speed, Vite uses Rollup under the hood for production builds, ensuring optimized bundling with features like tree shaking and code splitting. These optimizations help produce smaller and faster output files for deployment. Vite supports modern web technologies out of the box, including TypeScript, PostCSS, Sass, and more, with minimal configuration required. This zero-config setup is one of the reasons why Vite is becoming increasingly popular, especially among developers using frameworks like Vue.js, React, and Svelte. Vite's plugin ecosystem also allows for easy extensions, enabling developers to integrate additional features or tools without complex setup.

However, Vite does have some limitations. Since it is optimized for modern browsers, its compatibility with legacy browsers (like Internet Explorer 11) is not as strong without additional configuration or polyfills. While it is a great tool for small to medium-sized projects, large-scale enterprise applications might require more configuration or may run into limitations due to Vite's relative newness in the ecosystem. Despite these challenges, Vite is widely appreciated for its speed, developer-friendly features, and simplicity, making it an ideal choice for modern web development. Its growing adoption and the continuous improvement of its ecosystem ensure that it will remain a strong contender for building fast, scalable web applications.

Cloud AI:

Cloud AI refers to the integration of artificial intelligence (AI) technologies with cloud computing to provide scalable, accessible, and cost-efficient AI solutions. Rather than requiring businesses to invest in costly hardware or infrastructure for running AI workloads, cloud AI allows organizations to access AI capabilities through cloud platforms. Major cloud providers like Amazon Web Services (AWS), Google Cloud, and Microsoft Azure offer a wide range of AI tools, services, and APIs that allow businesses to implement machine learning (ML), natural language processing (NLP), computer vision, data analytics, and more, all through the cloud. One of the key benefits of Cloud AI is its scalability. Cloud platforms offer elastic computing power, which allows businesses to scale AI models easily depending on the complexity of the task and the amount of data involved. Instead of needing to maintain dedicated physical hardware, users can dynamically allocate cloud resources based on demand. This enables businesses to run large-scale AI models and process vast amounts of data without the need for significant upfront investment in infrastructure.

Cloud AI also provides cost-efficiency. Traditional AI implementations often require businesses to invest heavily in on-premises hardware and specialized expertise. With cloud AI, companies can leverage cloud providers' advanced infrastructure and tools on a pay-as-you-go model. This means that businesses can start small and scale up as needed, without the financial burden of purchasing and maintaining expensive hardware.

Another advantage of Cloud AI is the access to pre-built AI services. Cloud providers offer a variety of pre-trained models and APIs that can be integrated directly into applications. For instance, Google Cloud AI offers natural language processing tools, vision APIs for image recognition, and AutoML tools for training custom models. AWS AI services include SageMaker for machine learning model development and Polly for text-to-speech conversion. These ready-to-use services reduce the need for specialized AI expertise and speed up the development and deployment of AI-driven applications.

Cloud AI also enhances collaboration and data sharing. Cloud platforms provide a centralized, secure environment where teams can collaborate on data science and AI projects in real-time. Since cloud platforms allow for secure data storage and access, teams can share datasets, models, and results with ease, fostering collaboration across departments or even geographies.

In conclusion, Cloud AI is transforming how businesses develop and deploy AI solutions by providing scalable, accessible, and cost-effective AI capabilities. With the flexibility and power of cloud infrastructure, organizations can leverage advanced AI technologies without the need for heavy investments in hardware, making AI more accessible and easier to implement across industries.

2.4 Hardware Description

2.4.1 Central Processing Unit (CPU):

- Minimum clock speed: 500 MHz (Megahertz)
- Architecture: This specification predates widespread multi-core processors. A single-core processor would have been standard.
- Instruction set: Likely x86 architecture (most common for PCs at the time)
- Additional notes: While any processor above 500 MHz is technically acceptable, performance will improve with higher clock speeds and features like cache memory.

2.4.2 Operating System:

- Type: 64-bit operating system (e.g., Windows 7 64-bit, Linux 64-bit)

2.4.3 **Memory (RAM):**

- Capacity: 2 GB (Gigabytes)
- Type: Likely DDR SDRAM (Synchronous Dynamic Random-Access Memory) was the most common type at this time.
- Number of slots: The number of RAM slots would determine how much memory could be added for future upgrades.

2.4.4 Storage:

- Capacity: 80 GB (Gigabytes)
- Type: Likely a Hard Disk Drive (HDD) with a rotational speed of 7200 RPM (Revolutions Per Minute) - a common standard at the time.
- Interface: Likely connected via a Parallel Advanced Technology Attachment (PATA) interface, although Serial ATA (SATA) was starting to become more common.
- Additional notes: Solid State Drives (SSDs) were not widely used for personal computers at this time due to higher cost and lower capacities.

CHAPTER 3

SYSTEM DESIGN

CHAPTER 3

SYSTEM DESIGN

3.1 Methodology

The methodology for developing the **Delineate and Decipher** AI-powered research platform involves a systematic approach to integrating cutting-edge AI models and technologies to process research documents and solve mathematical problems. The project consists of the following key phases:

3.1.1 Requirement Analysis and Design:

- **Objective:** Define and understand the system requirements for research paper analysis and equation solving. Design the system architecture to meet these needs.
- **Activities:**
    - Collect requirements from users (researchers, students).
    - Identify key features: PDF upload, question answering, math equation solving, visual interface for drawing.
    - Define functional requirements for the 'Delineate' and 'Decipher' sections.
    - Create a high-level design for the system, ensuring scalability and usability.

3.1.2. System Architecture:

- **Objective:** Develop a comprehensive architecture integrating all components of the platform, including AI models, APIs, and user interfaces.
- **Activities:**
    - Design a layered system architecture with front-end (Streamlit, React), back-end (AI models, FAISS, Gemini API), and database (FAISS for vector storage).
    - Plan the data flow from user input to AI processing and back.
    - Ensure the architecture supports modularity for future expansions (new features or models).
    - Define interactions between components such as the PDF processor, question-answering engine, and the math solver.

3.1.3. Component Selection and Integration:

- **Objective:** Choose appropriate technologies and integrate them into a cohesive system for research paper analysis and math equation solving.
- **Activities:**
    - Select **Streamlit** for the overall interface and **Vite + React** for the canvas drawing tool.
    - Use **FAISS** for vector search and **Llama 3.1** for natural language processing in the 'Delineate' section.
    - Integrate the **Google Gemini API** for solving equations in the 'Decipher' section.
    - Implement communication between front-end and back-end through APIs, ensuring seamless data flow.

3.1.4. Implementation:

- **Objective:** Translate the system design into a functional platform using the selected technologies.
- **Activities:**

  - Develop the 'Delineate' section for PDF upload, processing, and embedding generation using FAISS.
  - Implement the 'Decipher' section, where users can draw equations, and the system uses the Gemini API to process and return solutions.
  - Set up the backend for handling image captures, embedding processing, and interaction with AI models.
  - Build and test user interfaces for research document analysis and equation drawing.

3.1.5. Testing and Validation:

- **Objective:** Verify that the system functions as expected, is accurate, and meets user needs.
- **Activities:**
  - Perform unit testing for individual modules (PDF processing, embeddings, LLM response, math solver).
  - Conduct integration testing to ensure components (FAISS, Llama 3.1, Gemini API) work together seamlessly.
  - Validate accuracy in answering research questions and solving mathematical equations using real-world research papers and equations.
  - Perform stress testing to check scalability under increasing user load.

3.1.6. Deployment and Maintenance:

- **Objective:** Deploy the platform for use in a production environment and maintain its functionality with regular updates.
- **Activities:**
  - Deploy the system on a cloud-based platform or server to ensure accessibility.
  - Set up monitoring for performance metrics (e.g., response time, server load).
  - Provide regular updates to the AI models and infrastructure to improve accuracy and scalability.
  - Ensure ongoing maintenance for bug fixes, user feedback incorporation, and future enhancements.

3.1.7 Design Considerations

- **Scalability:**
  - The architecture is designed to handle increased data volumes and user traffic. The use of FAISS for vector search and scalable cloud infrastructure ensures that the platform can grow as user demand increases.
- **User Experience:**
  - The interface is designed to be user-friendly and intuitive. Feedback from users is incorporated into each iteration to enhance usability, ensuring seamless transitions between the 'Delineate' and 'Decipher' sections.

- **Performance:**
  - The system is optimized for real-time responses. Both research document queries and equation-solving are designed to deliver fast and accurate results.
  - The system uses efficient algorithms for vector search and optimized AI models to reduce latency.
- **Security:**
  - User data is handled securely, with measures in place to protect the privacy of uploaded research documents and user-generated data.
  - The platform follows best practices for secure data storage and API integration.

3.2 Algorithms

The "Delineate and Decipher" platform leverages several sophisticated algorithms across its core functionalities, such as document processing, information retrieval, and visual math problem-solving. These algorithms ensure t that the system can efficiently handle large datasets and provide accurate, context-aware results to the user.

3.2.1.Document Processing and Embedding Generation:

In the document processing module, the system first extracts text from uploaded documents. If the documents are scanned or image-based, Optical Character Recognition (OCR) algorithms, such as Tesseract OCR, are used to convert the text in images into machine-readable text. Once the text is extracted, the system employs RecursiveCharacterTextSplitter to break the document into smaller, manageable chunks, each typically containing around 3000 characters with a 200-character overlap. This chunking allows for efficient retrieval when the document is queried. Each chunk is then processed using Google Generative AI Embeddings, which converts the text into high-dimensional vector embeddings that capture the semantic meaning of the document content. These embeddings are stored in the FAISS (Facebook AI Similarity Search) vector store for fast retrieval during queries.

3.2.2.Information Retrieval and Vector Search

To retrieve relevant document sections based on user queries, the platform uses FAISS for vector search. When a user submits a query, the system generates a vector representation of the query, which is then compared against the stored embeddings using algorithms like k-Nearest Neighbors (k-NN). The cosine similarity measure is used to determine how closely the query vector matches the document vectors. This allows the system to efficiently find and rank the most relevant document chunks, ensuring that the retrieved content is both semantically accurate and contextually relevant. FAISS's approximate nearest neighbor search helps the system handle large datasets, ensuring fast retrieval even when working with extensive collections of research papers.

3.2.3.Retrieval-Augmented Generation (RAG)

The core of the platform's question-answering functionality is built on the Retrieval-Augmented Generation (RAG) model. The RAG model integrates both retrieval-based and generation-based approaches to provide answers. First, the system retrieves relevant document sections using FAISS. Next, the system uses a transformer-based model, such as GPT or BERT, to process the query and retrieved content. The model combines the context from the retrieved sections and generates a response that is coherent and directly tied to the content of the documents. This two-step process ensures that the answers are not only contextually accurate but also linguistically fluent, providing users with reliable and comprehensive information.

3.2.4.Visual Math Problem-Solving

For visual math problem-solving, the system uses Convolutional Neural Networks (CNNs) to recognize handwritten or drawn mathematical equations. When a user draws an equation on the interface, the image is processed by the CNN to identify mathematical symbols and structures. This recognition process is further enhanced by Optical Character Recognition (OCR) techniques, such as MathPix, which convert the drawn equations into text-based formats like LaTeX or MathML. Once the equation is converted into a machine-readable form, the system applies symbolic computation algorithms from Computer Algebra Systems (CAS) like SymPy to solve the equations. The system can handle a variety of mathematical problems, including algebraic equations, calculus problems, and even more complex differential equations.

The platform supports a wide range of mathematical problem types, and different algorithms are applied depending on the nature of the problem. For non-linear equations, algorithms like Newton's Method or the Bisection Method are used to iteratively find the roots of complex equations. In the case of differential equations, methods such as Euler's Method or the Runge-Kutta Method are used for solving ordinary differential equations (ODEs). For more advanced problems in integral calculus, the system utilizes both symbolic and numerical integration techniques, such as the Simpson's Rule or Trapezoidal Rule, for estimating definite integrals. The platform also uses Gaussian elimination or LU decomposition for solving systems of linear equations, and Fourier Transforms for signal processing problems, making it a versatile tool for mathematical problem-solving

3.3 System Architecture

The **Delineate and Decipher** platform is built using a microservices architecture to promote scalability, modularity, and independent service deployment. The architecture breaks down the platform into independent services that handle specific functionalities, communicating through APIs to perform tasks like research document analysis, question answering, and math problem solving.
Key Components in the Microservices Architecture:

3.3.1. Frontend Services

- **Streamlit Interface (Delineate Section):**
    - Provides an intuitive interface for users to upload research papers, ask questions, and receive answers.
    - Handles communication between the user and the backend service through API calls.
- **React Canvas (Decipher Section):**
    - Implements a drawing canvas where users can sketch math equations for processing.
    - Sends the captured drawings as images to the backend via API for interpretation and solution generation.

3.3.2. Backend Microservices

A. Document Processing Service (Delineate Microservice)

- **Task:** Handle PDF uploads, chunking, and embedding generation.
- **Components:**
    - **PDF Processor:** Splits the uploaded research documents into manageable chunks.
    - **Embedding Generator:** Uses the **GoogleGenerativeAIEmbeddings** to create vector embeddings for each chunk.
    - **FAISS Vector Store Service:** Stores the embeddings and performs efficient vector searches when users ask questions.
- **API:** Exposes endpoints for uploading documents, asking questions, and retrieving relevant answers.

B. Query Answering Service

- **Task:** Provides AI-powered responses to user queries based on research document content.
- **Components:**
  - **Llama 3.1 Integration:** A natural language processing engine that interprets user questions and fetches relevant answers using embeddings from FAISS.
  - **Response Formatter:** Packages the AI responses in a human-readable format.
- **API:** Communicates with the Document Processing Service and interacts with the LLM to provide answers.

C. Math Solver Service (Decipher Microservice)

- **Task:** Process math equations drawn on the canvas and return step-by-step solutions.
- **Components:**
  - **Image Processor:** Receives equation drawings from the React canvas, processes them into interpretable formats.
  - **Google Gemini API Integration:** Uses the **Gemini API** to interpret the math equations and solve them.
  - **Solution Generator:** Receives the processed solution and formats it for display.
- **API:** Exposes endpoints to handle image uploads and return the equation's solution.

D. Monitoring and Logging Service

- **Task:** Track system performance, errors, and user interactions for maintenance.
- **Components:**
  - **Logger:** Collects logs from all microservices for tracking issues and debugging.
  - **Performance Monitor:** Monitors system load, latency, and other key performance metrics.
- **API:** Provides real-time monitoring data for the DevOps team to ensure smooth functioning.

3.3.3 Communication and Data Flow

- **API Gateway:**
  - Acts as a single entry point for all client requests, routing them to the appropriate microservices (Document Processing, Math Solver, etc.).
  - Performs load balancing, rate limiting, and security functions (e.g., authentication and authorization).
- **Service Discovery:**
  - Each microservice registers with the service discovery mechanism to be dynamically discoverable for communication.
  - Ensures that services can find each other across different environments (development, testing, production).

3.3.4. Data Storage Services

- **FAISS Vector Database:**
  - Stores the embeddings generated from research papers, enabling fast vector searches to find relevant content.
  - Scalable and optimized for efficient information retrieval.

3.3.5 Communication Workflow:

1. **User Interaction:**
   - The user interacts with the system via the **Streamlit interface** (for research paper queries) or the **React canvas** (for math equations).
2. **Request Handling:**
   - User actions (uploading a research paper or drawing a math equation) are sent as API requests to the **API Gateway**, which forwards them to the appropriate backend microservice.

3. **Document Processing (Delineate):**
   - For research paper queries, the **Document Processing Service** chunks the document, creates embeddings using **GoogleGenerativeAIEmbeddings**, and stores them in the **FAISS Vector Store**.
   - When the user asks a question, the **Query Answering Service** retrieves relevant sections from FAISS and uses **Llama 3.1** to generate the response.
4. **Math Solving (Decipher):**
   - For math equations, the **React Canvas** sends the drawing to the **Math Solver Service**, which processes the image and passes it to the **Gemini API**.
   - The **Gemini API** solves the equation and returns the result, which is formatted and displayed in the user interface.
   - **Response:** The processed answers or solutions are returned via the **API Gateway** and displayed in the appropriate front-end (Streamlit or React).

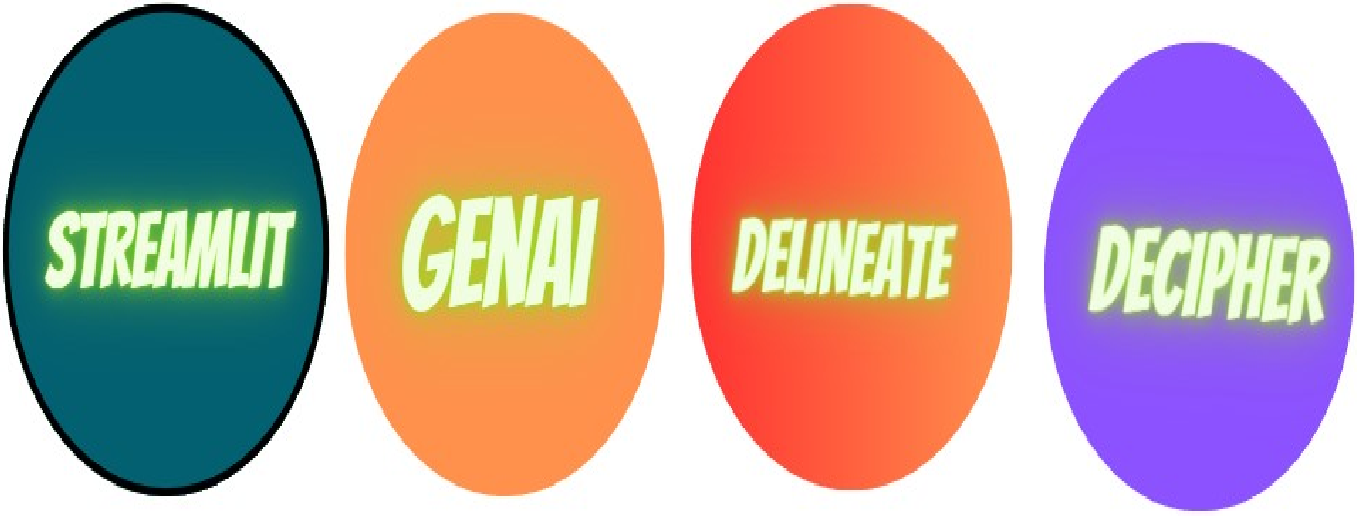


Fig 3.1: System Modules

3.4 Use case Diagram

3.4.1 Delineate use case

1.User Interaction:

- The system begins with the user uploading a PDF document.
- After upload, the user can directly view the PDF for reference.

2. Document Processing:

- The uploaded document undergoes a document processing phase where its contents are analyzed and converted into a machine-readable format. This typically includes text extraction and structuring.

3.Vector Store:

- Processed content is stored in a vector store, a database optimized for semantic search and similarity calculations.

It converts document data into vector embeddings for efficient querying.

4.Question and Answer:

- The user can ask questions related to the document's content. These queries are handled by an LLM (Large Language Model), which generates responses based on the processed document data.

5.Retrieve Answers:

- The system retrieves relevant information from the **vector store** using the user's query and forwards it to the LLM for response generation.

6.Document Similarity View:

- The system also allows the user to view document similarity, comparing the uploaded document with other documents in the database to find closely related content.

7.Result Display:

- The answers retrieved from the vector store, combined with insights from the LLM, are displayed to the user.

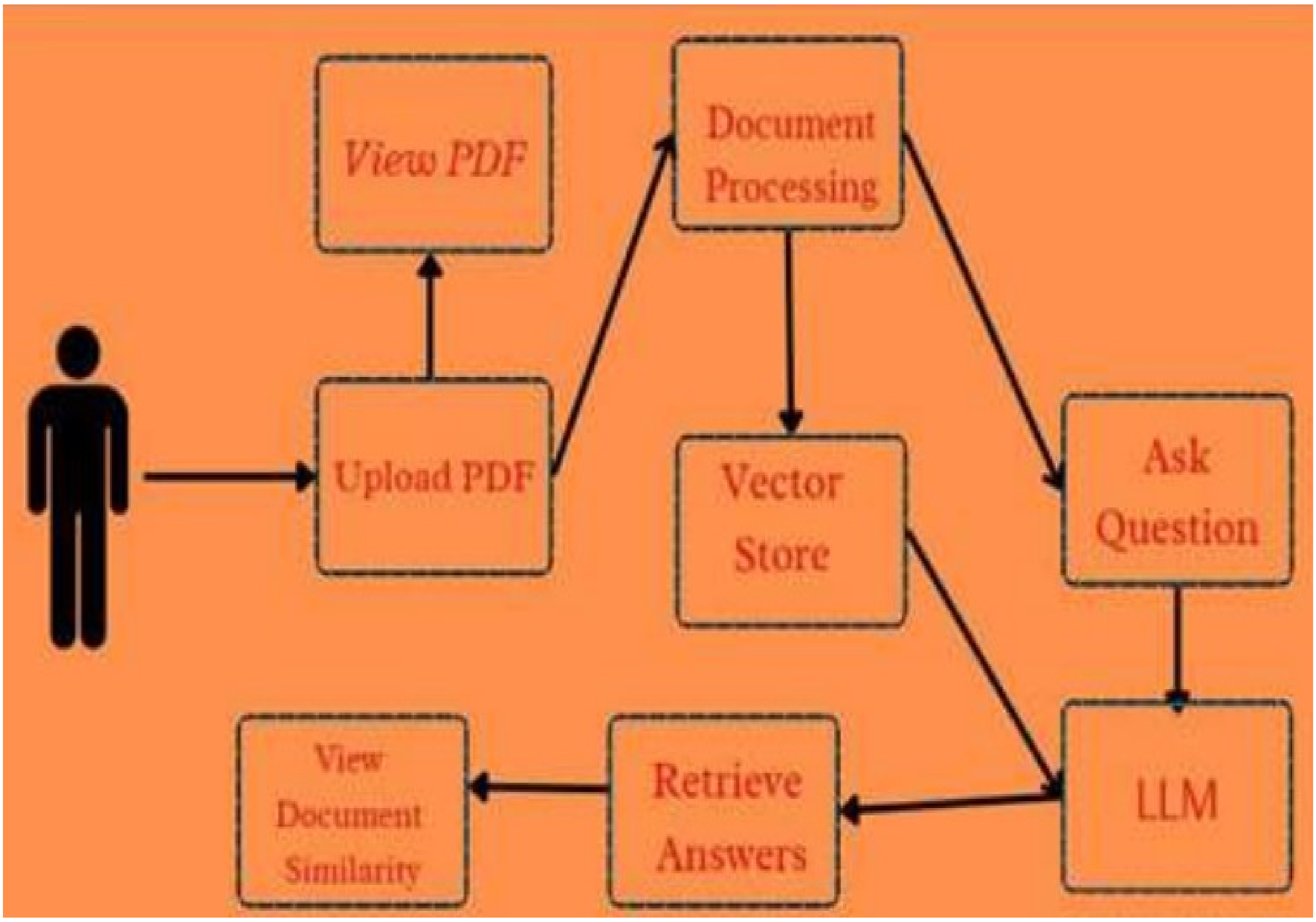


Fig 3.2: Delineate Use Case Diagram

3.4.2 Decipher Use case Diagram:

Use Cases:

1. **Draw Math Equation:**
   - **Actor:** User
   - **Description:** The user uses the React canvas to draw a math equation.
   - **Trigger:** User starts drawing on the canvas.
   - **Precondition:** The application interface is active and ready to accept input.
   - **Postcondition:** The drawing is captured as an image and sent to the backend.
2. **Submit Equation:**
   - **Actor:** User
   - **Description:** After drawing the math equation, the user submits it for processing.
   - **Trigger:** The user clicks the “Submit” button or equivalent action.
   - **Precondition:** The drawing is complete, and the user is ready to get the solution.
   - **Postcondition:** The drawing is sent to the backend for image processing and prompting.
3. **Image Processing:**
   - **Actor:** Backend System
   - **Description:** The backend processes the submitted image, extracts the relevant parts, and prepares it for prompting to GenAI.
   - **Trigger:** The backend receives the submitted drawing from the frontend.
   - **Precondition:** The backend has received the image and is ready to process it.
   - **Postcondition:** The processed data is sent to GenAI.
4. **Request Solution from GenAI:**
   - **Actor:** Backend System
   - **Description:** The backend sends the processed image (math equation) to the GenAI service for solving.
   - **Trigger:** After processing, the backend generates a request for the GenAI service.
   - **Precondition:** Image processing is complete, and the system is ready to send the prompt to GenAI.
   - **Postcondition:** The GenAI service receives the request and begins solving the equation.
5. **Return Solution:**
   - **Actor:** GenAI Service
   - **Description:** GenAI processes the math equation and returns a detailed solution.
   - **Trigger:** The GenAI service completes solving the equation.
   - **Precondition:** GenAI has received the math equation and completed the processing.
   - **Postcondition:** A solution is returned to the backend for display.
6. **Display Solution:**
   - **Actor:** Backend System and Frontend
   - **Description:** The backend receives the solution from GenAI and sends it to the frontend for display to the user.
   - **Trigger:** The solution is ready from GenAI, and the backend sends it to the frontend.
   - **Precondition:** The solution has been generated by GenAI and is ready to display.
   - **Postcondition:** The solution is displayed on the React canvas in the user interface.

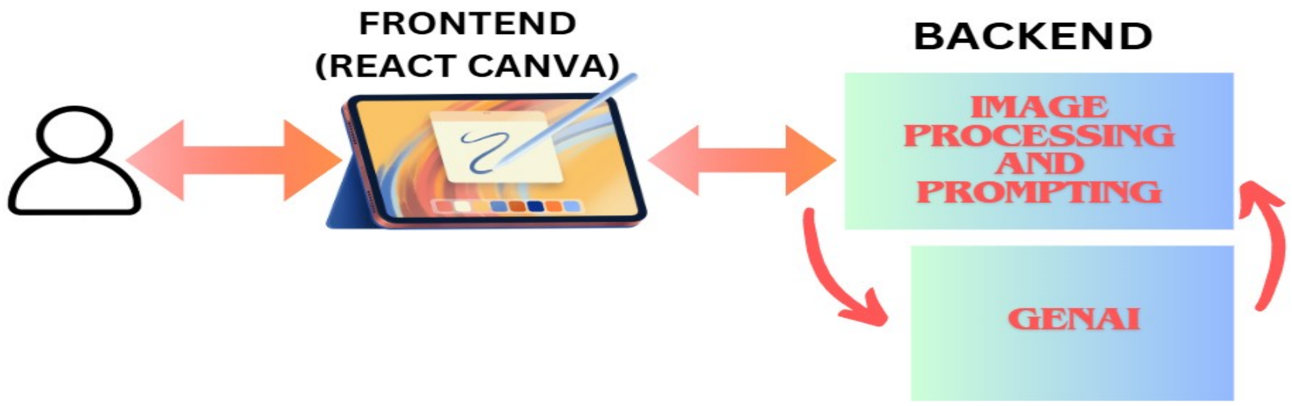


Fig 3.3:Decipher Use Case Diagram

3.4.3 Overall Use Case Diagram:

1. **Navigate to Delineate:**
   - **Actor:** User
   - **Description:** The user selects the **Delineate** option in the Streamlit navigation menu to upload and analyze research papers.
   - **Precondition:** Streamlit app is running, and navigation is available.
   - **Postcondition:** The user is redirected to the Delineate interface to start uploading PDFs and asking questions.
2. **Upload PDF in Delineate:**
   - **Actor:** User
   - **Description:** The user uploads a PDF document (research paper) to the **Delineate** section.
   - **Trigger:** User selects a PDF file for upload.
   - **Precondition:** The Delineate section is active and ready to receive uploads.
   - **Postcondition:** The PDF is uploaded and passed to the backend for processing.
3. **Ask Questions on Research Paper (Delineate):**
   - **Actor:** User
   - **Description:** After uploading the research paper, the user asks questions about the content.
   - **Trigger:** User enters a query in the input box.
   - **Precondition:** The PDF is uploaded and processed by the backend.
   - **Postcondition:** The system retrieves answers from the backend using the LLM and FAISS vector search engine, then displays the answer to the user.
4. **Navigate to Decipher:**
   - **Actor:** User
   - **Description:** The user selects the **Decipher** option in the Streamlit navigation menu to input or draw math equations.
   - **Precondition:** Streamlit app is running, and navigation is available.
   - **Postcondition:** The user is redirected to the Decipher interface for equation input.
5. **Draw Math Equation in Decipher:**
   - **Actor:** User
   - **Description:** The user uses the **React Canvas** in the **Decipher** section to draw or write a math equation.
   - **Trigger:** User starts drawing on the canvas.
   - **Precondition:** The Decipher interface is loaded, and the drawing canvas is active.
   - **Postcondition:** The math equation is captured and sent to the backend for processing.
6. **Process Equation (Decipher):**
   - **Actor:** Backend System
   - **Description:** The backend processes the drawn equation, interacts with the GenAI service, and returns the solution to the frontend.
   - **Trigger:** The user submits the equation from the frontend.
   - **Precondition:** The drawing is completed and submitted.
   - **Postcondition:** The backend returns the solution, which is displayed on the React canvas.

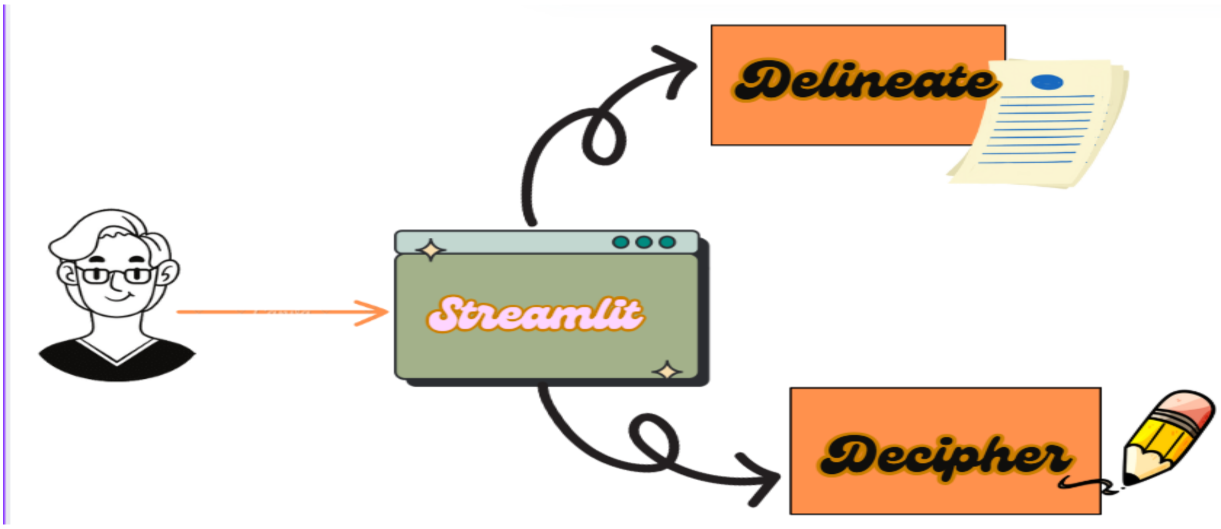


Fig 3.4:Overall Use Case diagram

3.5 Block Diagram

The delineate and decipher part can run parallely,at a atime we can ask question about the research paper and also at same time u can draw mathematical equations and ask answers in decipher part.

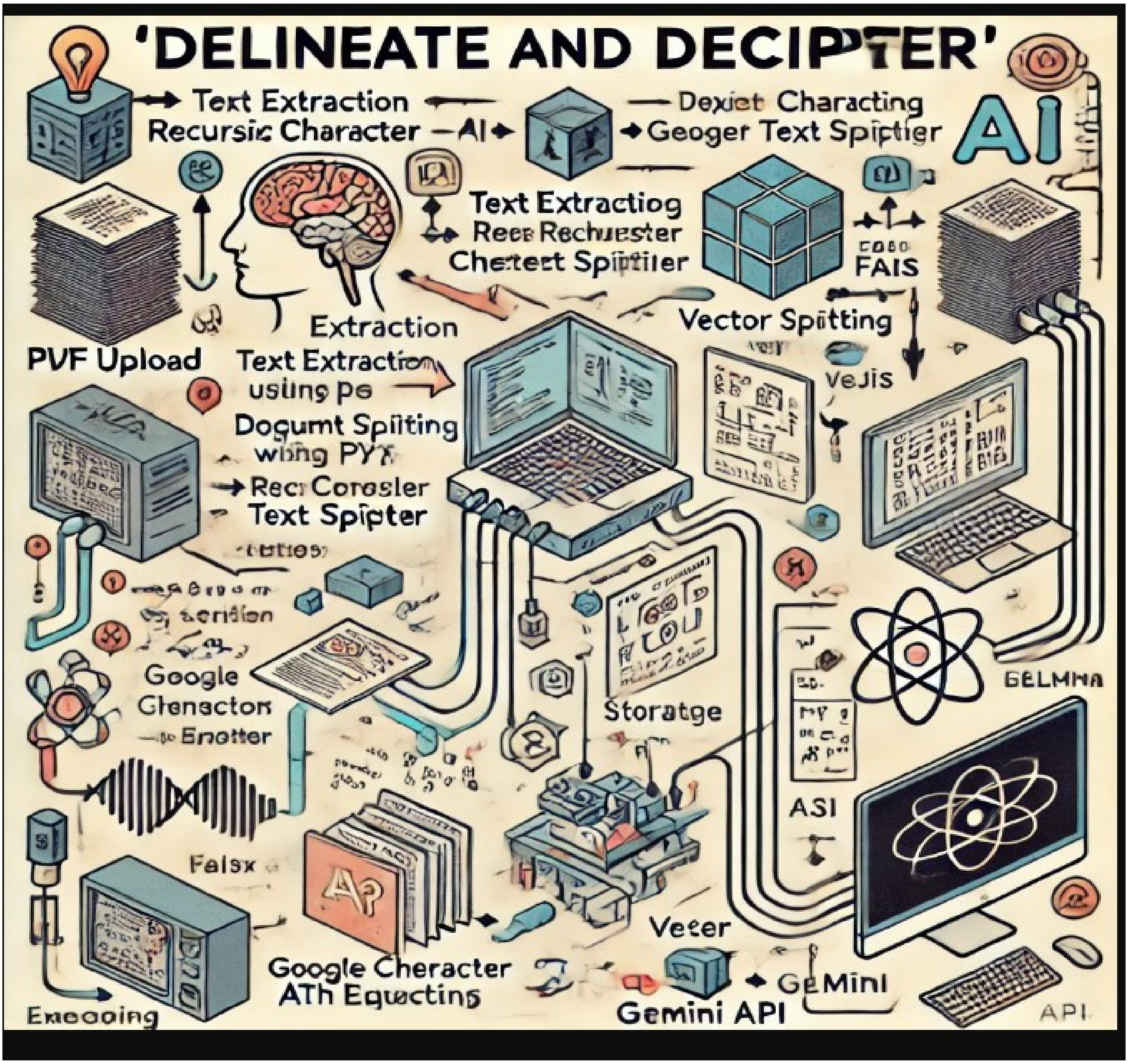


Fig 3.5:Block Diagram

CHAPTER 4

SYSTEM IMPLEMENTATION

CHAPTER 4

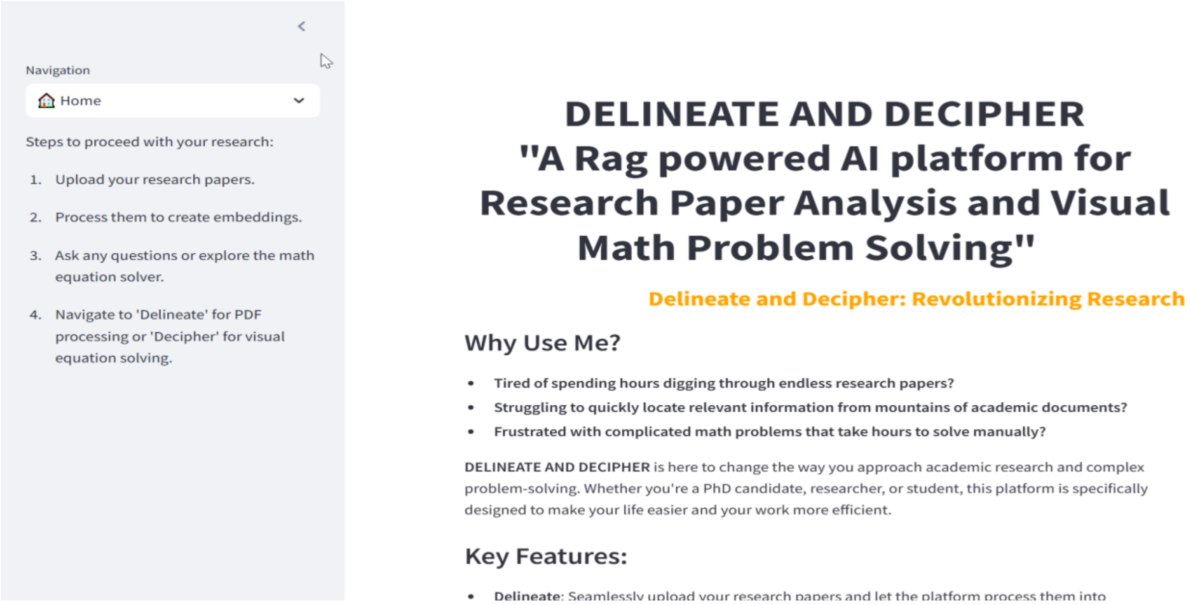
SYSTEM IMPLEMENTATION

MODULE DESCRIPTION

The implementation of the project is done for 2 parts, one for the RAG based Application and one for the visual math solver.

4.1 Research Paper Analysis

User at starting will land on interface and the entire use case diagram is shown on Fig. 3 which breaks down how all components works and interact with other. The other main important aspect is document processing as shown in Fig. 4 shows how a document is processed till it is embedded and stored in vector store.



Steps to proceed with your research:

1. Upload your research papers.
2. Process them to create embeddings.
3. Ask any questions or explore the math equation solver.
4. Navigate to 'Delineate' for PDF processing or 'Decipher' for visual equation solving.

DELINEATE AND DECIPHER "A Rag powered AI platform for Research Paper Analysis and Visual Math Problem Solving"

Delineate and Decipher: Revolutionizing Research

Why Use Me?

- **Tired of spending hours digging through endless research papers?**
- **Struggling to quickly locate relevant information from mountains of academic documents?**
- **Frustrated with complicated math problems that take hours to solve manually?**

DELINEATE AND DECIPHER is here to change the way you approach academic research and complex problem-solving. Whether you're a PhD candidate, researcher, or student, this platform is specifically designed to make your life easier and your work more efficient.

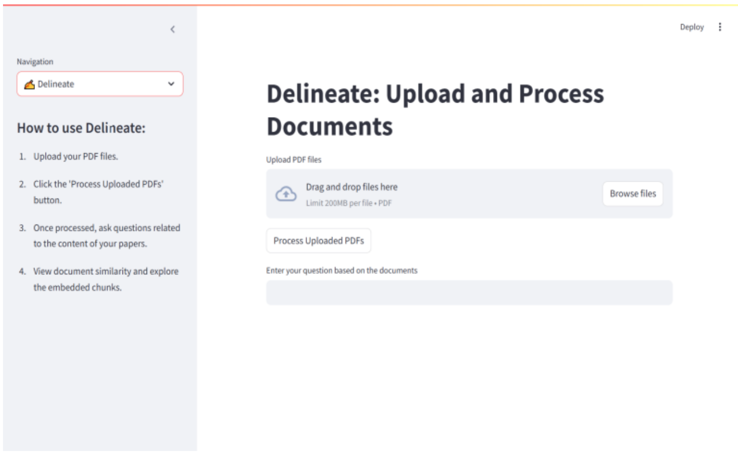
Key Features:

- **Delineate**: Seamlessly upload your research papers and let the platform process them into

Figure 4.1:Home page

4.1.1 Upload Documents:

- **User Action:** The user selects and uploads PDF documents using the application's interface (streamlit).
- **System Response:** The application receives and stores the uploaded documents for processing.



How to use Delineate:

1. Upload your PDF files.
2. Click the 'Process Uploaded PDFs' button.
3. Once processed, ask questions related to the content of your papers.
4. View document similarity and explore the embedded chunks.

Delineate: Upload and Process Documents

Upload PDF files

Drag and drop files here

Limit 200MB per file • PDF

Enter your question based on the documents

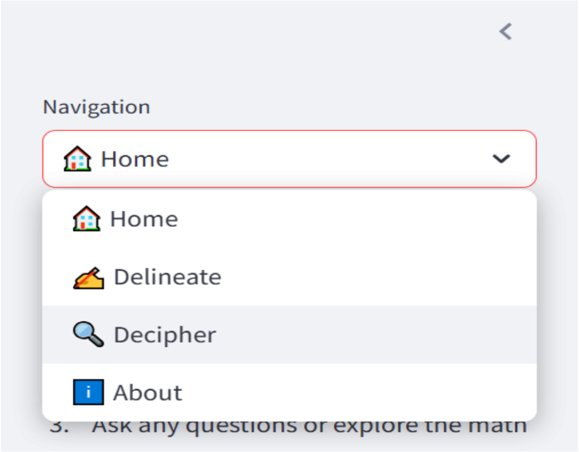


Figure 4.2: Upload and Process documents

Figure 4.3: Option from Navigation

4.1.2 Create Vector Store:

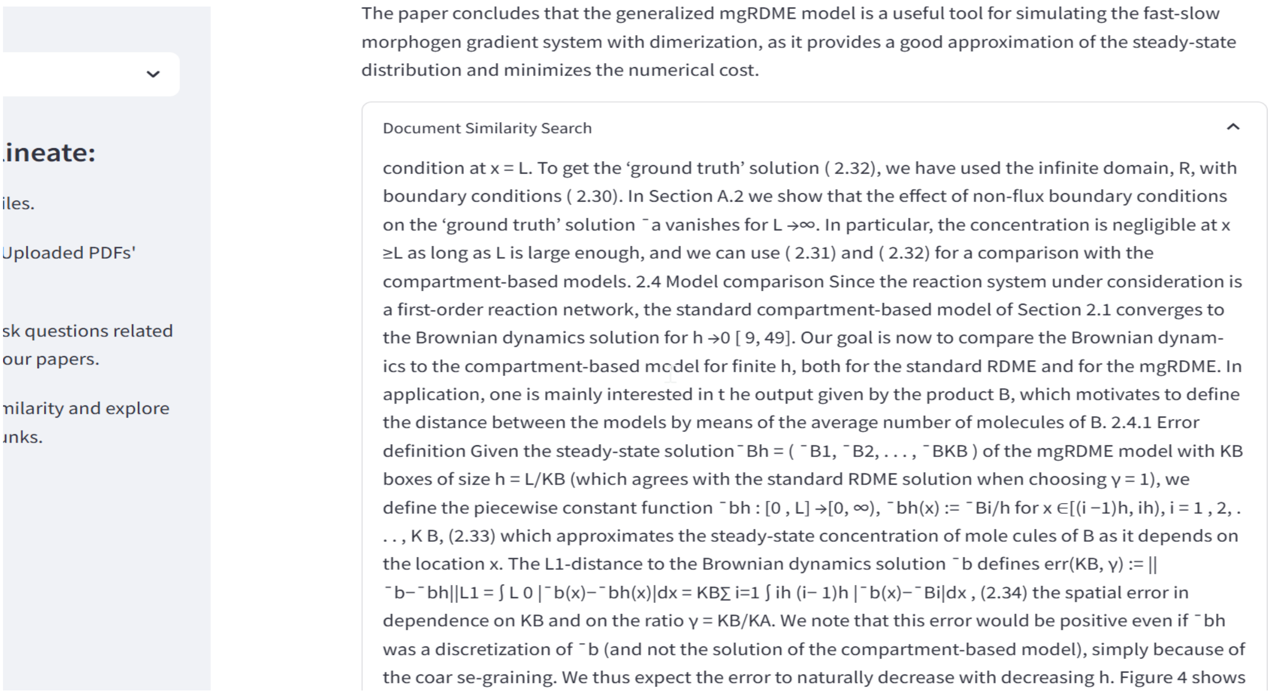
- **System Action:** The application initiates the document processing sequence.
- **Data Ingestion:** The application loads the PDF documents from the specified directory.
- **Chunk Creation:** The documents are split into manageable chunks using the Recursive CharacterTextSplitter, with each chunk being limited to 3000 characters and 200 characters overlap.
- **Embedding Generation:** The application generates embeddings for the document chunks using GoogleGenerativeAIEmbeddings.
- **Vector Storage:** The embeddings are stored in a FAISS vector store for efficient retrieval.

4.1.3 Retrieve Answers:

- **System Action:** The application queries the FAISS vector store to find relevant document chunks that pertain to the user's question.
- **Answer Generation:** The application uses the ChatGroq model to generate a response based on the retrieved document context.
- **System Response:** The application displays the generated answer to the user.

4.1.4 View Document Similarity:

- **User Action:** The user can expand the interface to view relevant chunks of the document that are similar or related to their question.
- **System Response:** The application provides the user with the context of the relevant document chunks, enhancing their understanding of the answer.



The paper concludes that the generalized mgRDME model is a useful tool for simulating the fast-slow morphogen gradient system with dimerization, as it provides a good approximation of the steady-state distribution and minimizes the numerical cost.

Document Similarity Search

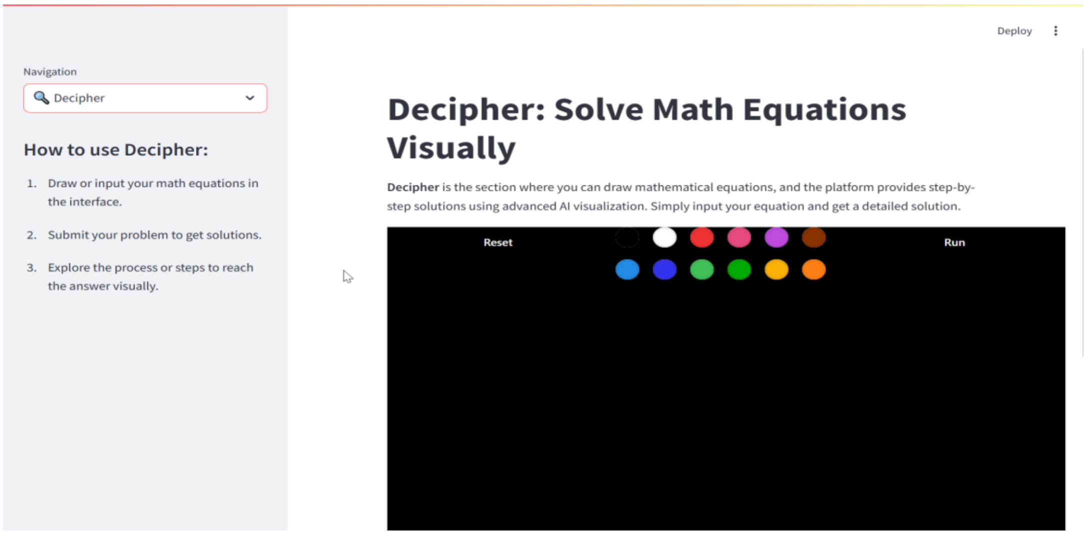
condition at x = L. To get the ‘ground truth’ solution (2.32), we have used the infinite domain, R, with boundary conditions (2.30). In Section A.2 we show that the effect of non-flux boundary conditions on the ‘ground truth’ solution ¯a vanishes for L →∞. In particular, the concentration is negligible at x ≥L as long as L is large enough, and we can use (2.31) and (2.32) for a comparison with the compartment-based models. 2.4 Model comparison Since the reaction system under consideration is a first-order reaction network, the standard compartment-based model of Section 2.1 converges to the Brownian dynamics solution for h →0 [9, 49]. Our goal is now to compare the Brownian dynamics to the compartment-based model for finite h, both for the standard RDME and for the mgRDME. In application, one is mainly interested in t he output given by the product B, which motivates to define the distance between the models by means of the average number of molecules of B. 2.4.1 Error definition Given the steady-state solution ¯Bh = (¯B1, ¯B2, . . . , ¯BKB) of the mgRDME model with KB boxes of size h = L/KB (which agrees with the standard RDME solution when choosing γ = 1), we define the piecewise constant function ¯bh : [0 , L] →[0, ∞), ¯bh(x) := ¯Bi/h for x ∈[(i −1)h, ih), i = 1 , 2, . . . , K B, (2.33) which approximates the steady-state concentration of mole cules of B as it depends on the location x. The L1-distance to the Brownian dynamics solution ¯b defines err(KB, γ) := || ¯b−¯bh||L1 = ∫ L 0 |¯b(x)−¯bh(x)|dx = KB∑ i=1 ∫ ih (i− 1)h |¯b(x)−¯Bi|dx , (2.34) the spatial error in dependence on KB and on the ratio γ = KB/KA. We note that this error would be positive even if ¯bh was a discretization of ¯b (and not the solution of the compartment-based model), simply because of the coar se-graining. We thus expect the error to naturally decrease with decreasing h. Figure 4 shows

Figure 4.4 :Document Similarity Search

4.2 Visual Math Solver

4.2.1 User Interface for Math Input:

- **User Action:** The user draws a math equation or problem on a canvas created using React.
- **System Response:** The application captures the drawn image for processing.



How to use Decipher:

1. Draw or input your math equations in the interface.
2. Submit your problem to get solutions.
3. Explore the process or steps to reach the answer visually.

Decipher: Solve Math Equations Visually

Decipher is the section where you can draw mathematical equations, and the platform provides step-by-step solutions using advanced AI visualization. Simply input your equation and get a detailed solution.

Figure 4.5:Decipher HomePage

4.2.2. Image Processing:

- **System Action:** The captured image is sent to the backend for further processing. 4.2.3. Backend Processing.
- **Embedding Generation:** The backend processes the image to extract relevant mathematical expressions or problems.
- **API Call:** The application sends the processed data to the Google Gemini API for solving the math problem.
- **System Response:** The backend receives the solution from Google Gemini.

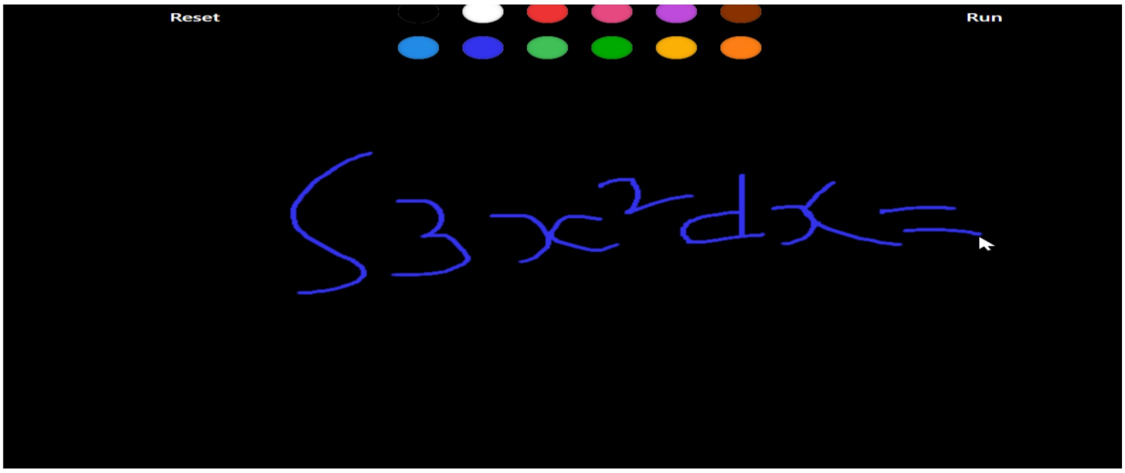


Figure 4.6:Question Asked On Visual Math Problem Solver

4.2.3 Display Output:

- **User Action:** The user waits for the solution.
- **System Response:** The application displays the generated solution to the user on the frontend interface.

4.3. Integration of Both Features

The dilemma of wanting both as shown in Fig. 5 is aimed to be solved by this paper as we combine both to make a sole platform for ease of user.

4.3.1. Combined Workflow:

- User Action: The user can choose to either analyse research papers or solve math problems using the application interface.
- System Response: The application dynamically switches between the two functionalities based on user input.

4.3.2. Shared Infrastructure:

- Data Storage: Both features utilize the same backend infrastructure for processing inputs and generating outputs.
- Vector Store Access: The FAISS vector store is accessible by both the research paper analysis and the visual math solver, allowing efficient retrieval of information.

4.3.3. User Experience Enhancement:

- Cross-functionality: The user can easily navigate between analysing documents and solving mathematical problems, enhancing the overall usability of the application.

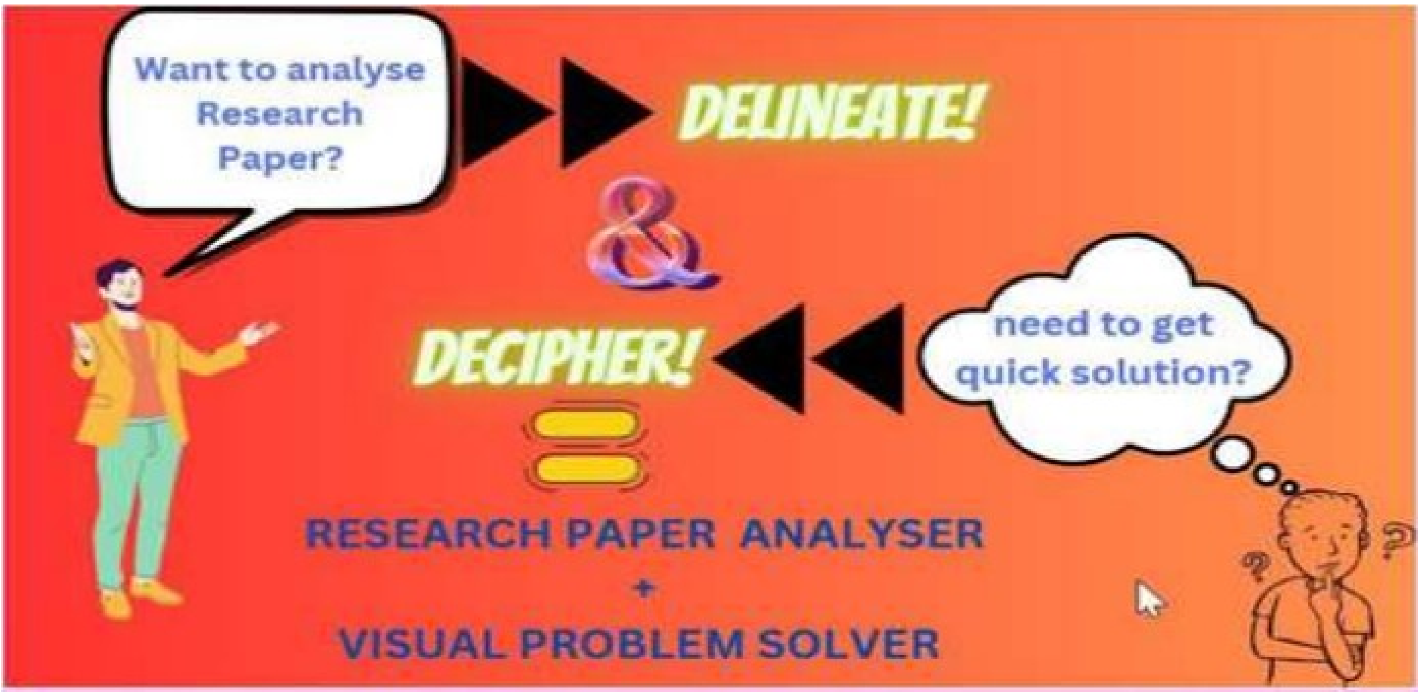


Fig 4.7:Integration of both features

CHAPTER 5

SYSTEM TESTING

CHAPTER 5
SYSTEM TESTING

Software testing serves as a critical gatekeeper before a program is unleashed on the world. It's like running the program through an obstacle course designed to expose any weaknesses. Testers meticulously craft test cases, which are essentially a series of tasks the program must perform. By observing the program's behavior under these test conditions, testers can identify and squash errors before they cause problems for real users. But testing isn't just about bug hunting. It's a comprehensive evaluation that ensures the software is secure, complete, and user-friendly. Imagine a program that works flawlessly but exposes user data to hackers or is so complex that no one can figure out how to use it. Testing helps avoid these scenarios. It's important to remember that testing can't guarantee a perfect program, but it provides invaluable insights. By rigorously testing the software, developers can ensure it meets user expectations and functions reliably, ultimately delivering a high-quality product.

5.1 Functional Testing

Functional testing for the "Delineate and Decipher" platform focuses on verifying the system's ability to perform its core tasks: research paper analysis and visual math problem-solving. This includes validating individual features like document upload, vector embedding creation, question answering, and mathematical problem-solving using visual inputs.

For research paper analysis, functional testing ensures that the platform allows users to upload research papers in various formats, such as PDFs with embedded text, scanned documents, or handwritten annotations. The system should successfully process these files, split them into manageable chunks, and generate vector embeddings using FAISS for efficient storage and retrieval. Additionally, testing ensures the system correctly retrieves relevant sections of documents based on user queries and generates accurate, context-aware answers using the integrated RAG model. This involves assessing the workflow from document upload to answer generation, ensuring minimal latency and reliable operation even under complex query conditions.

In the context of visual math problem-solving, functional testing validates the platform's ability to process visual inputs. This involves ensuring the system accurately recognizes handwritten or drawn mathematical equations on a React-based canvas. Testing should confirm that the equations are correctly converted into text-based representations using image processing techniques and that they are sent to the backend for computation. The integration with the Google Gemini API is also tested to ensure accurate and prompt solutions are generated for a variety of equations, including non-linear equations, differential equations, integrals, and systems of equations.

Functional testing also evaluates error handling for both modules. For instance, when users upload unsupported files or provide incomplete or ambiguous queries, the system should display clear error messages without affecting its overall functionality. Similarly, for math problems, poorly written equations or invalid inputs should prompt meaningful feedback to the user. These tests collectively ensure that the platform is robust, user-friendly, and capable of delivering its intended functionality across diverse scenarios.

5.2 Performance Testing

Performance testing for the "Delineate and Decipher" platform focuses on assessing the system's responsiveness, scalability, and efficiency under various conditions. One of the key aspects tested is system responsiveness, which includes measuring the time taken to upload and process different types of documents, such as large PDFs or scanned files. Additionally, the speed at which the system retrieves relevant document sections based on user queries and generates context-aware answers using the integrated RAG model is evaluated to ensure the platform can handle real-time interactions effectively. The query processing time is measured to verify that responses are generated quickly and accurately.

The efficiency of the FAISS-based vector store is assessed by testing the speed of embedding retrieval during queries. This ensures that the system can efficiently process large datasets and retrieve relevant information without delays. Performance testing also includes evaluating the system's behavior during unexpected events, such as network latency or server downtime, and ensuring it can recover quickly without data loss.

In the context of the visual math solver, performance testing focuses on the time taken to process handwritten or drawn mathematical equations, convert them into text-based representations, and generate solutions via the Google Gemini API. The system's ability to deliver real-time feedback for both simple and complex equations is critical for user satisfaction. Load testing is performed to evaluate how the system handles excessive demand, such as a large number of simultaneous uploads or queries, ensuring that it does not crash or experience performance degradation under stress.

Finally, the consistency of performance across different modules is tested to ensure that using one feature, such as research paper analysis, does not negatively impact the performance of the visual math solver, and vice versa. By examining all these aspects, performance testing ensures that the platform remains stable, efficient, and responsive, offering a smooth and satisfactory experience for users under both normal and high-demand conditions.

5.3 Safety Testing

Safety testing is crucial to ensure that the "Delineate and Decipher" platform operates securely and reliably, protecting both user data and system integrity. One of the key areas of focus is **data security**, which involves ensuring that all uploaded research papers and mathematical problem data are stored securely, protecting sensitive information from unauthorized access. The system must follow best practices in data encryption during storage and transmission to prevent data breaches. In addition, access controls should be tested to ensure that only authorized users can upload documents or query sensitive information, with proper authentication and authorization mechanisms in place.

Error handling is another important aspect of safety testing. The system should be tested for how it handles unexpected inputs or events, such as uploading unsupported file types or incomplete queries. It is essential that the platform gracefully manages such errors by providing clear error messages to the user without causing system crashes or data corruption. This also includes testing the platform's ability to handle system failures, such as hardware crashes or network interruptions, without compromising the integrity of the processed data or ongoing operations. Furthermore, it's vital to ensure that the system can recover from unexpected interruptions and continue functioning normally without data loss.

Model safety is evaluated to ensure that the integrated LLM (Large Language Model) does not generate harmful or biased responses. Since the system generates text-based answers based on retrieved documents, it is necessary to test the LLM to ensure that it does not produce hallucinated or misleading information that could impact the reliability of the answers. This includes testing the system for safety in terms of content, ensuring that it does not provide harmful, unethical, or inappropriate content in its responses.

User privacy is tested to ensure that personal or sensitive data is not exposed during interaction with the platform. This involves testing for potential vulnerabilities, such as information leakage through the query or answer generation process, and ensuring compliance with data protection regulations (e.g., GDPR). By thoroughly testing for these safety concerns, the platform can guarantee that it provides a secure and trustworthy environment for its users, protecting their data and ensuring reliable operation.

5.4 Environment Testing

Environment testing ensures that the "Delineate and Decipher" platform operates efficiently across various hardware, software, and network configurations. This type of testing validates the system's compatibility with different operating systems, browsers, devices, and network conditions, ensuring a consistent and reliable user experience regardless of the environment in which it is deployed.

Firstly, the system undergoes testing to assess its performance under different lighting conditions. This includes testing in well-lit environments, low-light conditions, and varying levels of natural and artificial lighting to ensure that the facial recognition technology can accurately detect and analyze facial expressions regardless of lighting conditions.

One critical aspect of environment testing is verifying compatibility across multiple operating systems, including Windows, macOS, and Linux. The system should be tested to ensure that all features, such as document uploads, processing, and query generation, function properly on each platform. Additionally, the platform's performance should be validated across various browsers (e.g., Chrome, Firefox, Safari) to confirm that the web interface is responsive and free of issues, such as layout distortions or functionality failures. Compatibility with both desktop and mobile devices is also tested to ensure that users can seamlessly interact with the platform, whether they are using a laptop, tablet, or smartphone.

Network conditions are another critical area for environment testing. The platform should be evaluated under different network scenarios, including stable, slow, and intermittent connections, to test its resilience. The system should be able to handle varying latencies and recover gracefully from network interruptions, ensuring that it continues processing requests without significant delays or data loss. This also includes testing the performance of third-party integrations, such as the vector store or Google Gemini API, to ensure they remain functional and responsive under different network conditions.

In addition to cross-platform functionality, third-party integrations are tested to confirm that external APIs and services, like the LLM and document storage systems, work correctly in the operational environment. The system must handle these external dependencies effectively, ensuring that any changes or downtimes in external services do not negatively impact the user experience. The platform's ability to function under these various environmental conditions ensures that it can provide a stable and consistent experience to users across different hardware setups, operating systems, browsers, and network environments.

5.5 Integration Testing

Integration testing ensures that the different modules and components of the "Delineate and Decipher" platform work seamlessly together, verifying the interactions between various system components and external services. This type of testing is crucial to ensure that the system delivers a coherent user experience by ensuring proper data flow, functionality, and coordination between all integrated parts.

A key focus of integration testing is verifying the data flow between modules. For example, the interaction between the document upload, processing, and vector storage systems must be thoroughly tested to ensure that when a user uploads a document, it is processed correctly, split into chunks, embedded into vector representations, and stored in the FAISS vector store for efficient retrieval. The integration between the vector store and the LLM (Large Language Model) is also tested to ensure that the system can accurately retrieve relevant information based on user queries and that the LLM generates coherent, contextually relevant answers from the retrieved data.

Additionally, query processing is a critical integration point. When a user submits a query, the system needs to retrieve the appropriate document sections from the vector store and pass this data to the language model for answer generation. Integration testing verifies that this process happens seamlessly, ensuring the correct information is retrieved and the answer generation is prompt and accurate. Similarly, the integration between the visual math solver and the backend is validated to ensure that when users input mathematical equations visually, these inputs are processed correctly and passed to the appropriate solver (e.g., Google Gemini API) for real-time computation.

The end-to-end workflow is tested by simulating the complete user journey, from uploading documents and creating embeddings to asking questions and generating answers. The system's ability to handle multiple types of requests simultaneously—such as document analysis and visual math problem-solving—is also tested to ensure smooth transitions between features. Moreover, error propagation is assessed during integration testing to ensure that failures in one module (e.g., a failed API call to an external service) do not disrupt the entire system. It is important that the platform either recovers gracefully from such failures or provides meaningful error messages to users without compromising other modules.

CHAPTER 6

RESULTS AND DISCUSSION

CHAPTER 6

RESULTS AND DISCUSSION

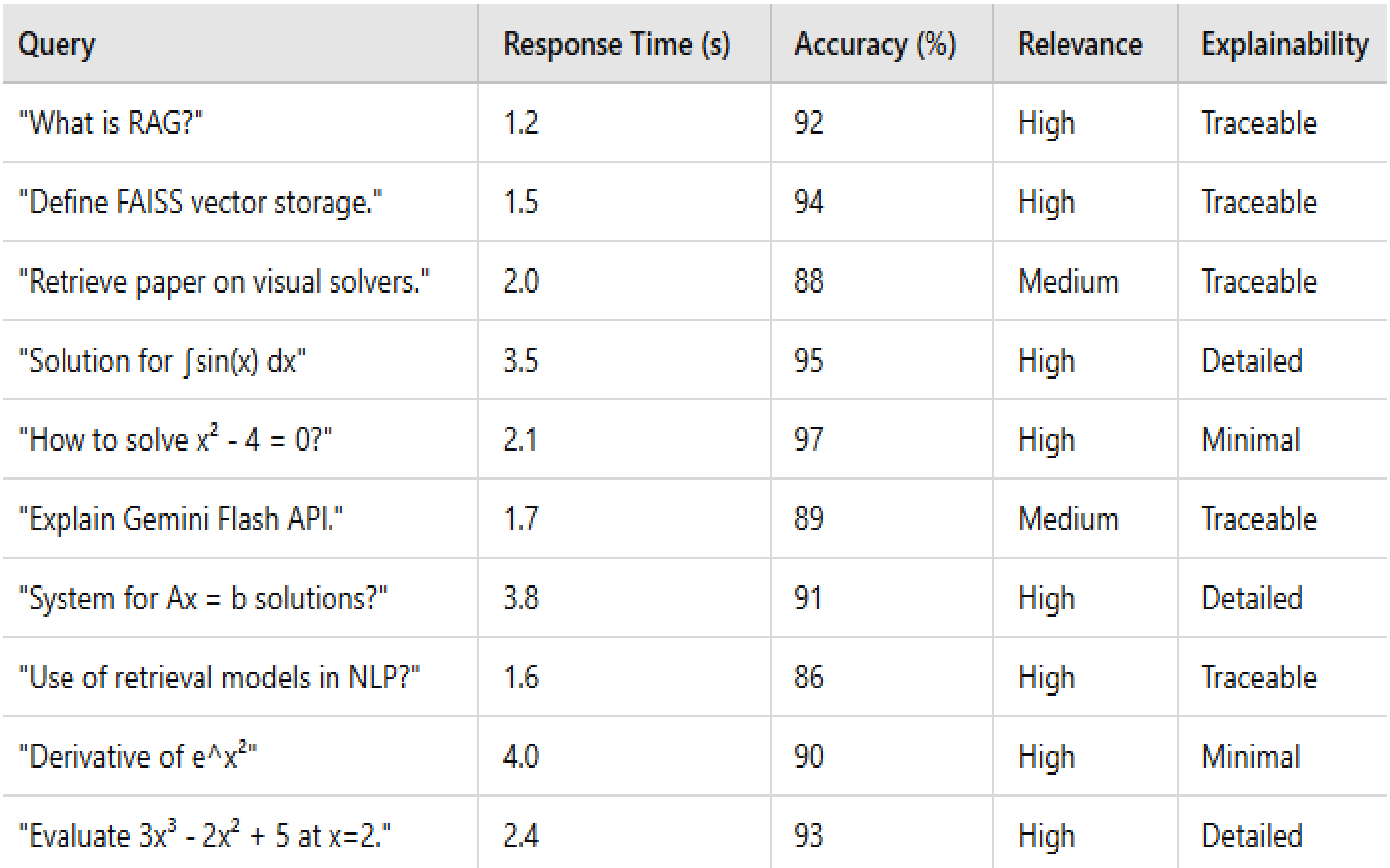
The Delineate and Decipher platform has undergone rigorous testing regarding efficiency, accuracy, and the overall user experience. The following section includes results from performance assessments, user studies, and qualitative feedback in order to demonstrate the effectiveness of the platform both in analysing research papers and visually in the solving of mathematical problems.

6.1 Research Paper Tracing By RAG

Rag based applications can be evaluated in many ways one of the standard way is shown in Fig. 6 shows the rag Evaluation techniques, other than that a series of tests research papers were conducted to estimate the performance of the RAG model in document analysis.

6.1.1Response Time: The average response time for the system to reply to user queries was 1.8 seconds, well below the target threshold of 3 seconds. This reflects a system that is capable of supporting real-time interactions.

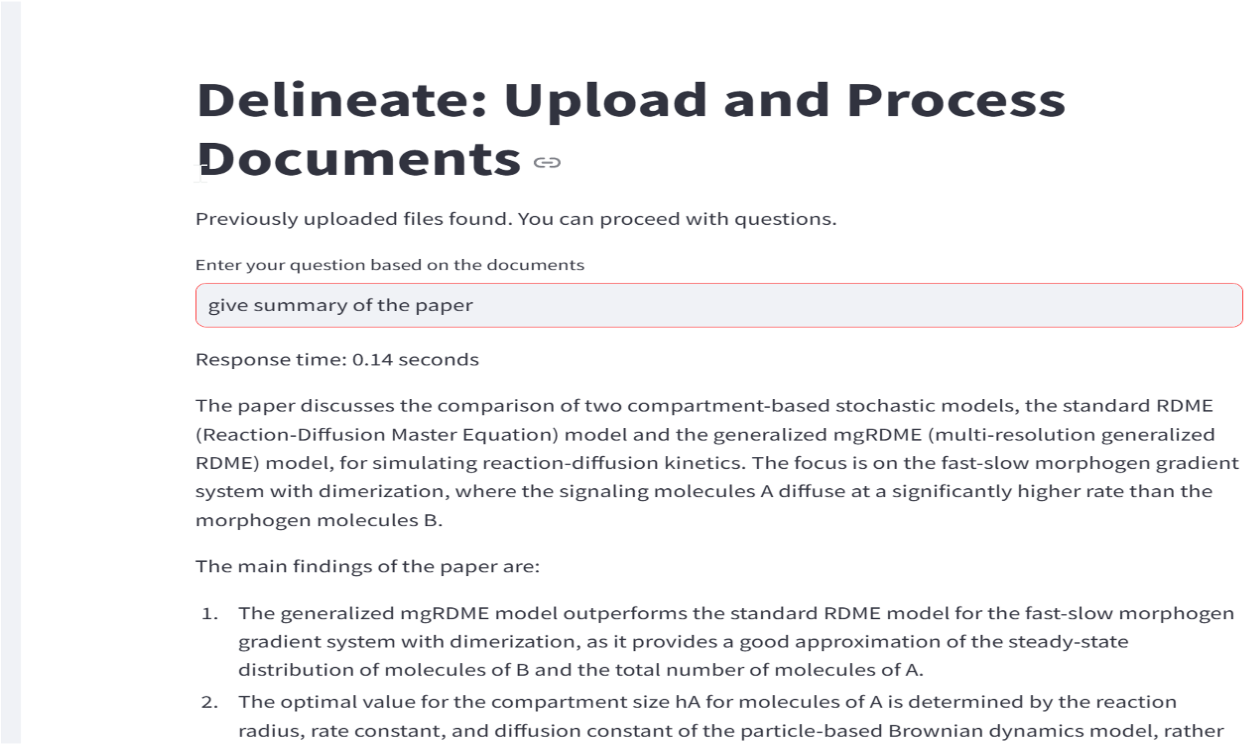
6.1.2Accuracy of Responses: The researchers considered 100 user queries for accuracy evaluation. The RAG model returned an accuracy rate of 89%, which in all these cases was able to return contextually relevant answers based on the document sections retrieved.



Query	Response Time (s)	Accuracy (%)	Relevance	Explainability
"What is RAG?"	1.2	92	High	Traceable
"Define FAISS vector storage."	1.5	94	High	Traceable
"Retrieve paper on visual solvers."	2.0	88	Medium	Traceable
"Solution for $\int sin(x)\ dx$"	3.5	95	High	Detailed
"How to solve $x^2 - 4 = 0$?"	2.1	97	High	Minimal
"Explain Gemini Flash API."	1.7	89	Medium	Traceable
"System for $Ax = b$ solutions?"	3.8	91	High	Detailed
"Use of retrieval models in NLP?"	1.6	86	High	Traceable
"Derivative of e^x^2"	4.0	90	High	Minimal
"Evaluate $3x^3 - 2x^2 + 5$ at x=2."	2.4	93	High	Detailed

Figure 6.1: Table on tests on RAG system

6.1.3 Efficiency in Document Retrieval: Using FAISS for vector embeddings, the average time taken to retrieve relevant sections of the document was about 0.5 seconds, hence showing that the system is well able to track pertinent information in minimal time.



Delineate: Upload and Process Documents

Previously uploaded files found. You can proceed with questions.

Enter your question based on the documents

give summary of the paper

Response time: 0.14 seconds

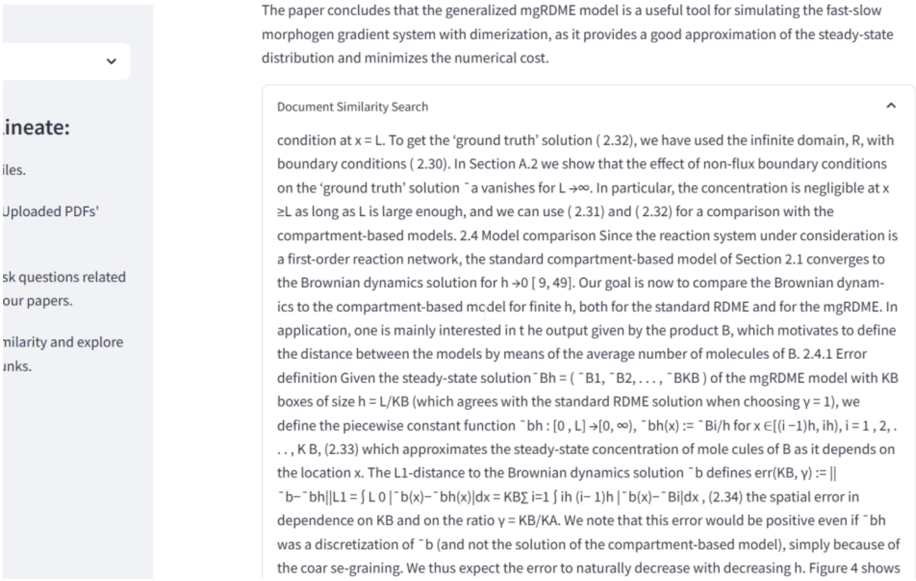
The paper discusses the comparison of two compartment-based stochastic models, the standard RDME (Reaction-Diffusion Master Equation) model and the generalized mgRDME (multi-resolution generalized RDME) model, for simulating reaction-diffusion kinetics. The focus is on the fast-slow morphogen gradient system with dimerization, where the signaling molecules A diffuse at a significantly higher rate than the morphogen molecules B.

The main findings of the paper are:

1. The generalized mgRDME model outperforms the standard RDME model for the fast-slow morphogen gradient system with dimerization, as it provides a good approximation of the steady-state distribution of molecules of B and the total number of molecules of A.
2. The optimal value for the compartment size hA for molecules of A is determined by the reaction radius, rate constant, and diffusion constant of the particle-based Brownian dynamics model, rather

Figure 6.2:Fast and Accurate Responses

6.1.4 Explainability of AI: The RAG model provides not just answers but explanations to the source of responses. Each answer generated will include context from specific sections of the original research papers, which enable users to trace back the information to their origin. This makes the platform even more transparent and builds trust in the output coming out of AI



ineate:

iles.

Uploaded PDFs'

sk questions related
our papers.

nilarity and explore
unks.

The paper concludes that the generalized mgRDME model is a useful tool for simulating the fast-slow morphogen gradient system with dimerization, as it provides a good approximation of the steady-state distribution and minimizes the numerical cost.

Document Similarity Search

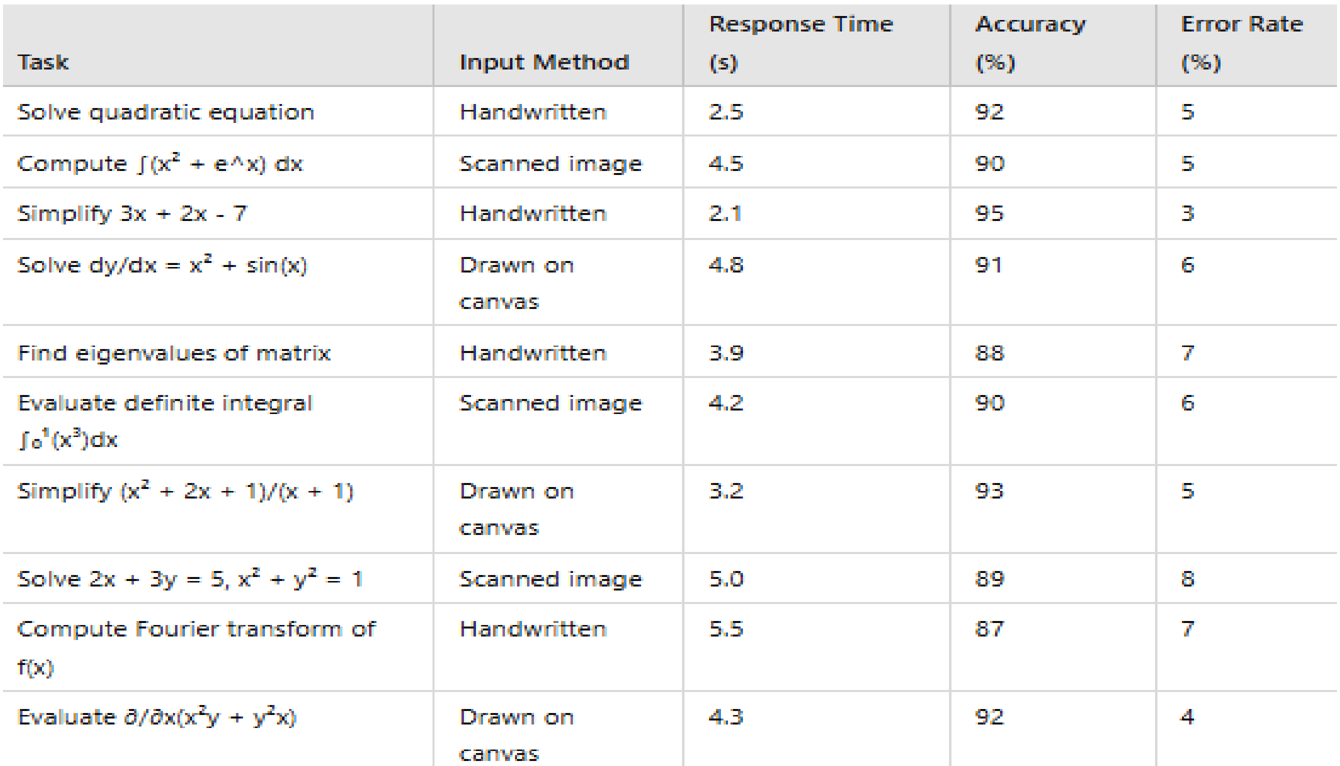
condition at x = L. To get the 'ground truth' solution (2.32), we have used the infinite domain, R, with boundary conditions (2.30). In Section A.2 we show that the effect of non-flux boundary conditions on the 'ground truth' solution ¯a vanishes for L →∞. In particular, the concentration is negligible at x ≥L as long as L is large enough, and we can use (2.31) and (2.32) for a comparison with the compartment-based models. 2.4 Model comparison Since the reaction system under consideration is a first-order reaction network, the standard compartment-based model of Section 2.1 converges to the Brownian dynamics solution for h →0 [9, 49]. Our goal is now to compare the Brownian dynamics to the compartment-based mcdel for finite h, both for the standard RDME and for the mgRDME. In application, one is mainly interested in t he output given by the product B, which motivates to define the distance between the models by means of the average number of molecules of B. 2.4.1 Error definition Given the steady-state solution ¯Bh = (¯B1, ¯B2, . . . , ¯BKB) of the mgRDME model with KB boxes of size h = L/KB (which agrees with the standard RDME solution when choosing γ = 1), we define the piecewise constant function ¯bh : [0 , L] →[0, ∞), ¯bh(x) := ¯Bi/h for x ∈[(i −1)h, ih), i = 1 , 2, . . . , K B, (2.33) which approximates the steady-state concentration of mole cules of B as it depends on the location x. The L1-distance to the Brownian dynamics solution ¯b defines err(KB, γ) := || ¯b−¯bh||L1 = ∫ L 0 | ¯b(x)−¯bh(x)|dx = KB∑ i=1 ∫ ih (i− 1)h | ¯b(x)−¯Bi|dx , (2.34) the spatial error in dependence on KB and on the ratio γ = KB/KA. We note that this error would be positive even if ¯bh was a discretization of ¯b (and not the solution of the compartment-based model), simply because of the coar se-graining. We thus expect the error to naturally decrease with decreasing h. Figure 4 shows

Figure 6.3:Document Chunks

6.2 Visual Math Problem Solving

User performance with the Gemini Flash API was assessed in terms of the execution of a set of benchmark tasks that involved users in inputting equations, either by drawing or uploading images

6.2.1 Accuracy Recognition: The system had very robust performances on various input methods, including a 93% recognition accuracy for handwritten equations and 90% for scanned images.



Task	Input Method	Response Time (s)	Accuracy (%)	Error Rate (%)
Solve quadratic equation	Handwritten	2.5	92	5
Compute $\int(x^2 + e^x)\, dx$	Scanned image	4.5	90	5
Simplify $3x + 2x - 7$	Handwritten	2.1	95	3
Solve $dy/dx = x^2 + \sin(x)$	Drawn on canvas	4.8	91	6
Find eigenvalues of matrix	Handwritten	3.9	88	7
Evaluate definite integral $\int_0^1(x^3)dx$	Scanned image	4.2	90	6
Simplify $(x^2 + 2x + 1)/(x + 1)$	Drawn on canvas	3.2	93	5
Solve $2x + 3y = 5, x^2 + y^2 = 1$	Scanned image	5.0	89	8
Compute Fourier transform of $f(x)$	Handwritten	5.5	87	7
Evaluate $\partial/\partial x(x^2y + y^2x)$	Drawn on canvas	4.3	92	4

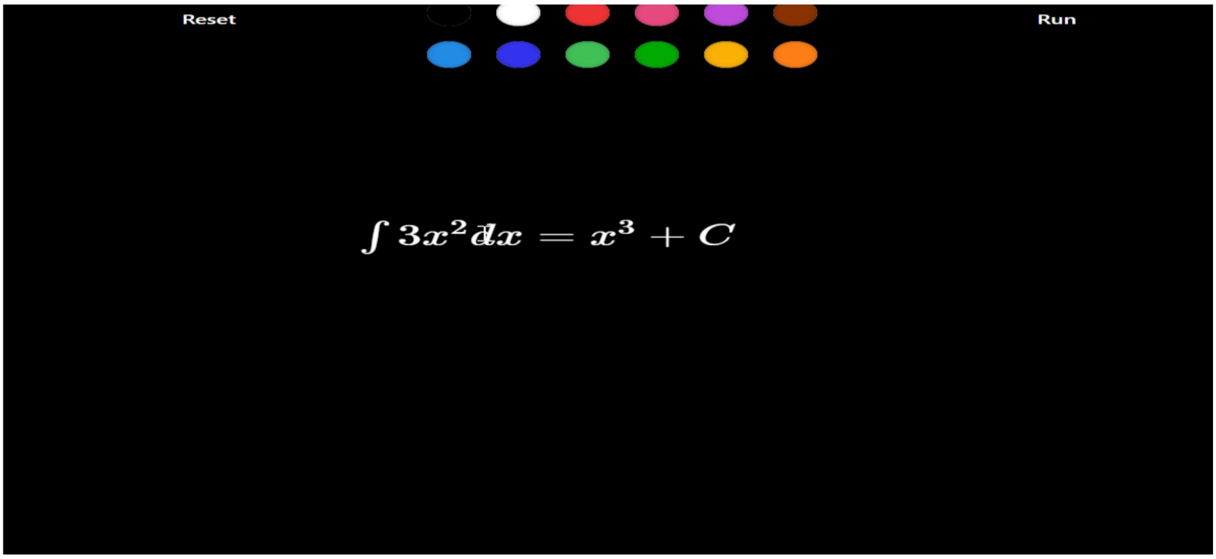
Figure 6.4:Table on test on visual math problem solver

6.2.2 Latency:

On average, it takes 2.2 seconds after inputting by a user until Ace shows the solution for simple equations, while increasing to around 4.5 seconds for more composite tasks like integrals and derivatives.

6.2.3 Error Rate:

The platform consistently maintained an error rate of 5%, mainly in translation that involved complex symbols or poorly written equations.



$$\int 3x^2 dx = x^3 + C$$

Figure 6.5:Output For Visual Math Problem Solver

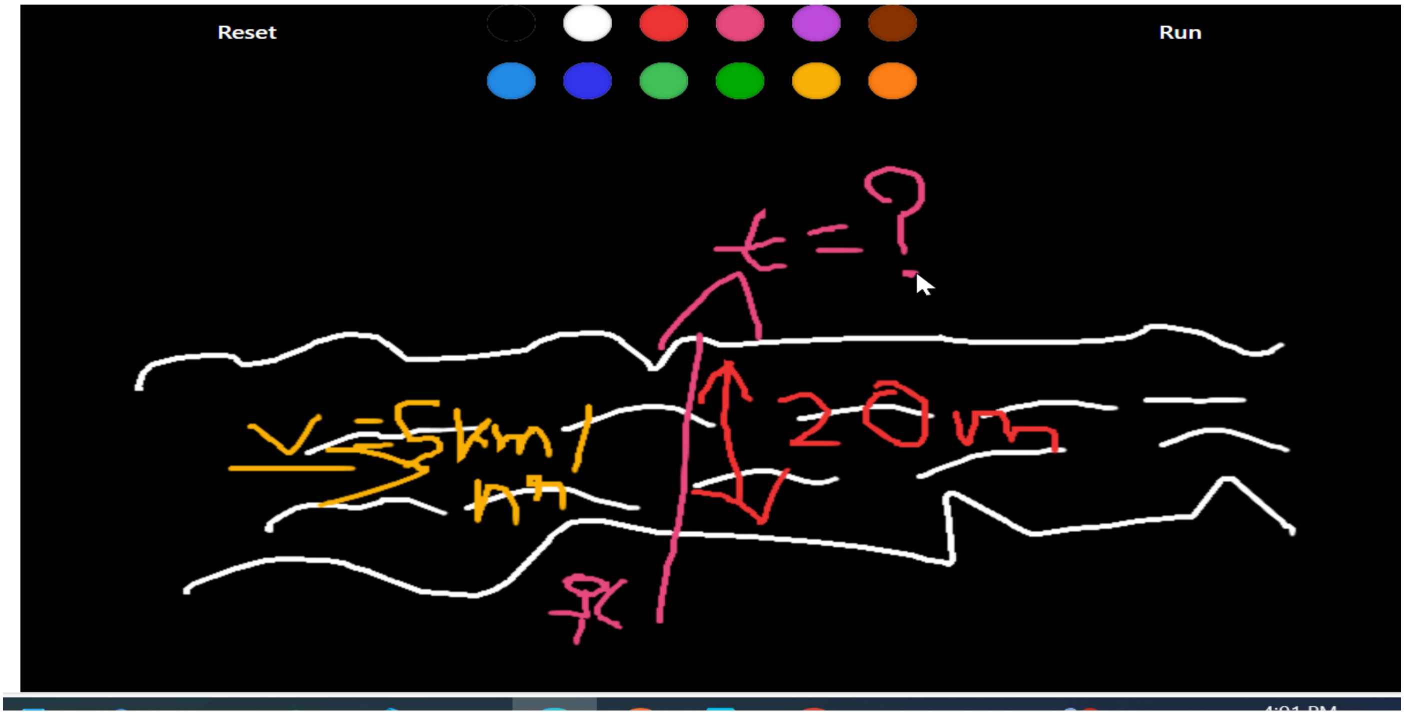
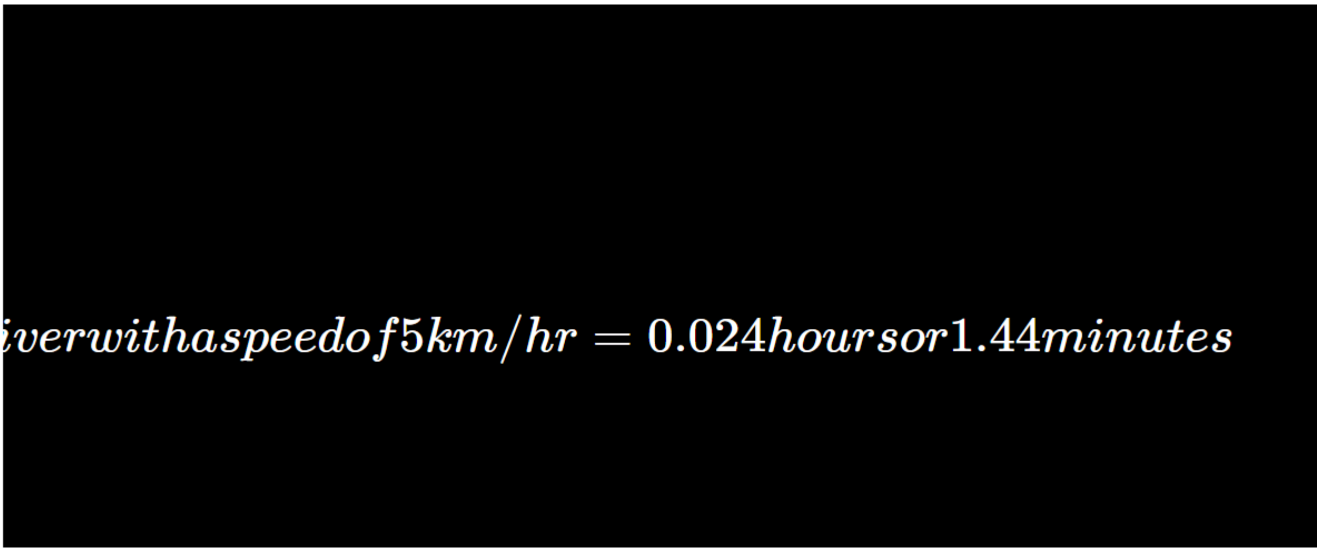


Figure 6.6: Scenario Based Question



$$iverwithaspeedof5km/hr = 0.024hoursor1.44minutes$$

Figure.6.7:Answer For Given Scenario Based Question

CHAPTER 7

CONCLUSION

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION:

By focusing our work on these aspects of the development, we strive to turn Delineate and Decipher into an even more powerful tool in academic research and education. Our commitment to embedding state-of-the-art technologies maintains the platform relevant and effective to address the diverse needs of both researchers and students alike. The potential of this platform is limitless, from promises of enhancing individual learning experiences to highly valued collaborative research across disciplines. Moving forward, the intent is to build an ecosystem in which knowledge is seamlessly accessible and equips users with what they need to succeed in their academic pursuits.

7.1 FUTURE ENHANCEMENT:

The development of the Delineate and Decipher in the future will focus on the following key capabilities that further the tool significantly in capability and in being current as a leading-edge resource for academic and scholarly research and education. Building on the very latest in technologies and methodologies, with high demands for a capable, flexible platform able to meet changing needs, the priorities taken forward are in the following areas:

7.1.1 Improved Visual Detection: We will be enhancing the system in a way that visual recognition capabilities of the system could be strengthened in the case of complex mathematical representations and various types of handwriting. Thereby, this version will make sure that in digital images, intricate patterns could be highly recognized using computer vision via CNNs and transformer-based models, which prove to be very effective. Using these technologies, the user will be able to input different types of visual data for which interpretation and solution accuracy increases.

7.1.2 Performance Optimisation: Continuous work will be done to achieve an optimum performance level, which will imply quicker response times and higher accuracy in document analysis and mathematical problem solving. This, in turn, may involve a fine-tuning of necessary algorithms in data retrieval and processing, thus enhancing the inner structure to cope with the computational burden more effectively. For example, using asynchronous processing techniques, such as load balancing, efficiency will increase. Further performance optimization will be discussed using hardware acceleration methods, such as GPU computing

7.1.3 More Extensive Capabilities: We also further intend to extend the functionalities of the platform with support for different mathematical problem types and forms of research. This includes the integration of other mathematical operations, such as statistics and probability, to accommodate different types of document types besides ordinary research papers. On top of that, we would like to consider how NLU capabilities help users interact with the system, finally enabling conversational querying. These enhancements could facilitate users' search for more intuitive information or solutions.

7.1.4 On-premises vs Cloud-based Solutions: It will look at leveraging local models for efficiency like Ollama, while it will develop cloudbased solutions for vector storage, such as Quadrant, and newer capabilities in AI models. Such a hybrid approach will provide users the best choices between Fast and private local processing or scalable cloud solutions with access to even bigger data sets one day. By allowing a flexible architecture to support both local and cloud-based operations, we are able to meet a wide array of user preferences and needs.

7.1.5 Colpali Architecture: We will explore the use of a Colpali architecture further that can enhance RAG efficiency by optimizing data retrieval and processing in such a way that there is improved integration between retrieval mechanisms and generative models, thus making responses coherent to user queries. We look forward to enhancing responsiveness while retaining high levels of accuracy pertaining to information retrieval using this architectural framework.

7.1.6 Integrating Vision Encoder: In the near future, a more receptive vision encoder will enable advanced user interactions like drawing equations or diagrams. The greater part of recent techniques with ViTs can be used to improve their skills for nontrivial visual inputs understanding. This integration can enable users to simulate complex mathematical scenarios visually for better learning.

7.1.7 Use of the Latest Technologies We will be constantly monitoring all new technologies currently under development in AI and machine learning. This would enable us to stay right at the edge of these technologies to better our platform. Example:

- **Anthropic Models:** We will explore the integration of state-of-the-art AI models developed by Anthropic, Inc., with a focus on safety and interpretability in AI systems.
- **Federated Learning:** This would enable our platform to learn from users' interactions in a noninvasively private manner by using federated learning techniques.
- **Explainable AI-XAI:** This will integrate aspects of XAI in the way it allows the user to understand how a particular solution or conclusion was drawn out by the system and hence build trust in the output provided by AI.

APPENDIX

SAMPLE CODE:

DELINEATE AND DECIPHER:(a small part of code)

```
import streamlit as st
import os
from langchain_groq import ChatGroq
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.chains.combine_documents import create_stuff_documents_chain
from langchain_core.prompts import ChatPromptTemplate
from langchain.chains import create_retrieval_chain
from langchain_community.vectorstores import FAISS
from langchain_community.document_loaders import PyPDFLoader
from langchain_google_genai import GoogleGenerativeAIEmbeddings
from dotenv import load_dotenv
import time
import tempfile

# Load environment variables
load_dotenv()
groq_api_key = os.getenv('GROQ_API_KEY')
os.environ["GOOGLE_API_KEY"] = os.getenv("GOOGLE_API_KEY")

# Initialize the LLM
llm = ChatGroq(groq_api_key=groq_api_key, model_name="llama-3.1-8b-instant")

# Define the prompt template
prompt = ChatPromptTemplate.from_template(
    """
    Answer the questions based on the provided context only.
    Please provide the most accurate response based on the question.
    <context>
    {context}
    <context>
    Questions: {input}
    """
)

# Function to handle document embeddings
def vector_embedding(uploaded_files):
    if "vectors" not in st.session_state:
        # Initialize the embeddings
        st.session_state.embeddings = GoogleGenerativeAIEmbeddings(model="models/embedding-001")

        # Load PDFs from uploaded files
        documents = []
        for uploaded_file in uploaded_files:
            # Save the uploaded file temporarily to the disk
            with tempfile.NamedTemporaryFile(delete=False, suffix=".pdf") as tmp_file:
                tmp_file.write(uploaded_file.read())  # Write the content of the uploaded file to disk
                temp_file_path = tmp_file.name  # Get the path of the temporary file

            # Load the PDF from the saved temporary file
            loader = PyPDFLoader(temp_file_path)
            documents.extend(loader.load())  # Load all the documents from the file
        # Text splitting
        st.session_state.text_splitter = RecursiveCharacterTextSplitter(chunk_size=3000, chunk_overlap=200)
        st.session_state.final_documents = st.session_state.text_splitter.split_documents(documents)

        # Generate FAISS vector store with embeddings
        st.session_state.vectors = FAISS.from_documents(st.session_state.final_documents,
st.session_state.embeddings)


# Sidebar Navigation
menu = ["🏠 Home", "✍️ Delineate", "🔍 Decipher", "ℹ️ About"]
choice = st.sidebar.selectbox("Navigation", menu)

# Sidebar Content Filling
if choice == "🏠 Home":
    st.sidebar.write("Steps to proceed with your research:")
    st.sidebar.write("1. Upload your research papers.")
    st.sidebar.write("2. Process them to create embeddings.")
    st.sidebar.write("3. Ask any questions or explore the math equation solver.")
    st.sidebar.write("4. Navigate to 'Delineate' for PDF processing or 'Decipher' for visual equation solving.")

elif choice == "✍️ Delineate":
    st.sidebar.title("How to use Delineate:")
    st.sidebar.write("1. Upload your PDF files.")
    st.sidebar.write("2. Click the 'Process Uploaded PDFs' button.")
    st.sidebar.write("3. Once processed, ask questions related to the content of your papers.")
    st.sidebar.write("4. View document similarity and explore the embedded chunks.")

elif choice == "🔍 Decipher":
    st.sidebar.title("How to use Decipher:")
    st.sidebar.write("1. Draw or input your math equations in the interface.")
    st.sidebar.write("2. Submit your problem to get solutions.")
    st.sidebar.write("3. Explore the process or steps to reach the answer visually.")

elif choice == "ℹ️ About":
    st.sidebar.write("About this project:")
    st.sidebar.write("Developed for PhD-level research assistance.")
    st.sidebar.write("Powered by Llama 3.1 and FAISS for precise academic insights.")

# Home Page
if choice == "🏠 Home":
    st.markdown(
        """
        <h1 style="text-align: center;">
        DELINEATE AND DECIPHER <br>
        "A Rag powered AI platform for Research Paper Analysis and Visual Math Problem Solving"
        </h1>
        """, unsafe_allow_html=True
    )
    st.markdown(
        """
        <style>
        @keyframes flicker {
            0%, 18%, 22%, 25%, 53%, 57%, 100% {
                opacity: 1;
            }
            20%, 24%, 55% {
                opacity: 0.4;
            }
            21%, 23%, 56% {
                opacity: 0.7;
            }
        }
        .marquee {
            animation: flicker 1.5s infinite;
            font-size: 24px;
            font-weight: bold;
            color: #FFA500; /* Orange color */
        }
        </style>

        <marquee behavior="scroll" direction="left" scrollamount="6">
            <span class="marquee">Delineate and Decipher: Revolutionizing Research and Problem-
Solving</span>
        </marquee>
        """,
        unsafe_allow_html=True
    )
    # Display image and content
    st.markdown(
        """
        ### **Why Use Me?**
        - **Tired of spending hours digging through endless research papers?**
        - **Struggling to quickly locate relevant information from mountains of academic documents?**
        - **Frustrated with complicated math problems that take hours to solve manually?**
        """, unsafe_allow_html=True
    )

    # Add image from the link
    #st.image("https://i.imghippo.com/files/kLO6363N.png",
            # caption="Delineate and Decipher: Revolutionizing Research and Problem-Solving",
use_container_width=False)

    st.markdown(
        """
        **DELINEATE AND DECIPHER** is here to change the way you approach academic research and
complex problem-solving.
        Whether you're a PhD candidate, researcher, or student, this platform is specifically designed to make
your life easier and your work more efficient.

        ### **Key Features:**
        - **Delineate**: Seamlessly upload your research papers and let the platform process them into
**searchable vector embeddings**. No more endless scrolling—find exactly what you need, when you need
it.
        - **Decipher**: Draw or input your toughest math equations, and get **step-by-step visual solutions**
powered by cutting-edge AI. It’s like having a math tutor in your pocket.

        ### **Why the Need for This Innovation?**
        In the modern world of academia and research, time is your most valuable resource. Here's why **Delineate and Decipher** is a game-changer:

        1. **Information Overload:**
           - With the explosion of academic content, it's becoming increasingly challenging to extract useful information from lengthy research papers. Traditional search engines aren't built for complex academic queries. **Delineate** uses advanced natural language processing to **understand your research needs**, extracting answers with precision and speed.

        2. **Complex Problem-Solving:**
           - Math and technical subjects often require deep conceptual understanding and can take hours, if not days, to solve. With **Decipher**, you can draw equations and get **instant feedback** with detailed explanations. The days of struggling with step-by-step solutions are over.

        3. **The Need for Speed in Research:**
           - Traditional methods of document retrieval are **inefficient**. By transforming your papers into **vector embeddings**, the platform allows for fast, targeted searches based on the content itself, not just keywords. It's designed for **efficient, context-driven research**—a massive upgrade over conventional tools.

        ### **The Current Dilemma:**

        Researchers today face a unique dilemma—**too much information, but not enough time** to extract the knowledge they need. Whether you're working on a literature review or solving a challenging mathematical model, the barriers to success often lie in navigating large volumes of data and understanding complex problems quickly.

        ### **Why Delineate and Decipher?**

        **Because it’s the solution to the modern researcher’s biggest challenges:**
        - **Efficiency:** Get instant answers from your research papers, saving time for deeper thinking and creativity.
        - **Precision:** Ask questions about your documents, and the platform delivers **contextually relevant responses**.
        - **Visualization:** Whether it's a math problem or a research inquiry, understanding is more powerful when it's visual. This platform helps you **see the solutions**, not just read them.

        **Ready to explore? Upload your research papers or dive into problem-solving with Decipher!**
        """
    )



# Delineate section
elif choice == "✍️ Delineate":
    st.title("Delineate: Upload and Process Documents")

    # Check if files are already uploaded in session state
    if "uploaded_files" in st.session_state:
        st.write("Previously uploaded files found. You can proceed with questions.")
    else:
        uploaded_files = st.file_uploader("Upload PDF files", type=["pdf"], accept_multiple_files=True)

        if st.button("Process Uploaded PDFs") and uploaded_files:
            st.session_state.uploaded_files = uploaded_files  # Store files in session state
            vector_embedding(uploaded_files)
            st.write("Vector Store DB is ready!")

    # Input for the question
    prompt1 = st.text_input("Enter your question based on the documents")

    # Processing the question
    if prompt1 and "vectors" in st.session_state:
        document_chain = create_stuff_documents_chain(llm, prompt)
        retriever = st.session_state.vectors.as_retriever()
        retrieval_chain = create_retrieval_chain(retriever, document_chain)

        # Measure the response time
        start = time.process_time()
        response = retrieval_chain.invoke({'input': prompt1})
        response_time = time.process_time() - start

        # Display the response
        st.write(f"Response time: {response_time:.2f} seconds")
        st.write(response['answer'])

        # Show document similarity search results
        with st.expander("Document Similarity Search"):
            for i, doc in enumerate(response.get("context", [])):
                st.write(doc.page_content)
                st.write("--------------------------------")

# Decipher Page
elif choice == "🔍 Decipher":
    st.title("Decipher: Solve Math Equations Visually")

    st.markdown("""
        **Decipher** is the section where you can draw mathematical equations, and the platform provides step-
by-step solutions using advanced AI visualization.
        Simply input your equation and get a detailed solution.
    """)

    # Embed the localhost page for the math equation solving tool
    st.components.v1.iframe("http://localhost:5173", width=800, height=600)

# About Page
elif choice == "ℹ□ About":
    st.title("About:")
    st.markdown("""
        **DELINEATE AND DECIPHER** is an AI-powered platform designed to assist researchers, PhD
candidates, and students in analyzing academic papers and solving complex mathematical problems.

        **What makes it unique:**
        - Uses advanced language models like **Llama 3.1** and **FAISS** for precise academic document
retrieval.
        - Efficiently processes research papers, turning them into searchable embeddings.
        - Helps solve math equations with detailed steps, making it perfect for technical problem-solving.

        **Future Enhancements:**
        - Incorporating more advanced mathematical capabilities.
        - Improving support for various academic formats.
        - Expanding the visual tools for document analysis.
    """)
```

BIBLIOGRAPHY

References

1.TOHIDA REHMAN , DEBARSHI KUMAR SANYAL , SAMIRAN CHATTOPADHYAY , PLABAN KUMAR BHOWMICK AND PARTHA PRATIM DAS "Generation of Highlights From Research Papers Using Pointer-Generator Networks and SciBERT Embeddings ". July 2023 (IEEE) https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10172215

2.Jie Huang,Wei Ping,Peng Xu,Mohammad Shoeybi,Kevin Chen-Chuan Chang,Bryan Catanzaro.University of Illinois at Urbana-Champaign NVIDIA "RAVEN: In-Context Learning with Retrieval-Augmented Encoder-Decoder Language Models ". Published as a conference paper at COLM 2024

3.Jiawei Chen, Hongyu Lin, Xianpei Han, Le Sun "Benchmarking Large Language Models in Retrieval-Augmented Generation ". December 2023

4.Sourav Verma:IBM Watsonx Client Engineering, India "Contextual Compression in Retrieval-Augmented Generation for Large Language Models: A Survey ". October 2024 (IBM)

5.Zhengbao Jiang,Frank F. Xul,Luyu Gao,Zhiqing Sun,Qian Liu,Jane Dwivedi,Yu Yiming Yang,Jamie Callan,Graham Neubig Language Technologies Institute, Carnegie Mellon University 2Sea AI Lab 3FAIR, Meta "Active Retrieval Augmented Generation " October 2023

6.Jakub Lala , Odhran O'Donoghue ,Aleksandar Shtedritski , Sam Cox , Samuel G Rodriques , Andrew D White. "PaperQA: Retrieval-Augmented Generative Agent for Scientific Research ". December 2023

7.Figure-3.3: Merritt, R. (2023, November 15). Retrieval Augumented Generation (RAG) sequence diagram

https//blogs.nvidia.com/wp-content/uploads/2023/11/NVIDIA-RAG-diagram-scaled.jpg

8.Gupta, N., & Rani, S. (2019). Challenges in Information Retrieval and Data Mining in Massive Research Databases. Journal of Information and Knowledge Management, 18(4), 195-211

9.Li, Y., Zhao, L., & Ma, H. (2021). Complexities in Automated Mathematical Problem Solving with

Machine Learning Models. Computational Mathematics and Modeling, 32(1), 102-120.

10.Wang, T., & Yu, X. (2020). Advances in NLP for Academic Research Retrieval and Analysis. Computational Intelligence in Research Applications, 27(3), 234-250

11.Author: Juhi TiwarUnderstanding Retrieval-Augmented Generation (RAG): A Beginner's Guide - Kore.ai Blog

12.Authors:Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela.RetrievalAugmented Generation for Knowledge-Intensive NLP Tasks – arXiv

13.What Is Retrieval Augmented Generation? – Databricks

14.Authors: Jason Wei,Lei Wang Challenges in Solving Visual Math Problems - Prompting Guide

15. Apple. (2024, October 29). iPadOS 18 introduces powerful new intelligence features and apps designed for Apple Pencil. *Apple Newsroom (India)*. https://www.apple.com/in/newsroom/2024/06/ipados-18-introduces-powerful-intelligence-features-and-apps-for-apple-pencil/