

Evaluating The Performance Of Llms On SQL And Nosql Database Using Langsmith

N. Shanmugapriya¹, P. Yugeshkaran², U.Hari Charan³, M. Jayasri⁴, C. Merlyne Sandra Christina⁵, P. Anitha⁶, K. Tamilselvi⁷

¹Professor, Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan University, Trichy, India. shanmugapriyan.set@dsuniversity.ac.in

²B.Tech, Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan University, Trichy, India. yugeshkaran01@gmail.com

³B.Tech, Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan University, Trichy, India. haricharanuggirala1133@gmail.com

⁴Assistant Professor, Department of Computer Science and Engineering, Dhanalakshmi Srinivasan University, Trichy, India. jayasrim.set@dsuniversity.ac.in

⁵Assistant Professor, Department of Computer Science and Engineering, Dhanalakshmi Srinivasan University, Trichy, India. merlynesandrachristinac.set@dsuniversity.ac.in

⁶Assistant Professor, Department of Computer Science and Engineering, Dhanalakshmi Srinivasan University, Trichy, India. anithap.set@dsuniversity.ac.in

⁷Assistant Professor, Department of Computer Science and Engineering, Dhanalakshmi Srinivasan University, Trichy, India. tamilselvik.set@dsuniversity.ac.in

Abstract- The integration of Large Language Models (LLMs) with organizational databases enables powerful Retrieval Augmented Generator (RAG) systems for advanced data analysis and informed decision making. Existing solutions primarily demonstrate RAG implementations with PDFs or SQL databases, often lacking a comprehensive evaluation of LLM performance. In contrast, this research aims to identify the most effective RAG system for both SQL and NoSQL databases by evaluating the performance of twelve leading LLMs from OpenAI, Meta AI, Google AI and Anthropic. The evaluation leverages LangSmith to assess performance across key metrics such as accuracy, correctness, error rate, P50 latency and P99 latency, Ultimately proposing the best suited LLM model for RAG based database applications

Keywords: LLMs, RAG, database, SQL, LangSmith, OpenAI, Meta AI, Google AI, Anthropic, accuracy, correctness, error rate and latency.

I INTRODUCTION

Decision making is key for every organization, as it depends on data sources from the database. When an industry relies on a database for making decisions, analyzing large datasets becomes challenging due to their vast size. It requires a large team to perform preprocessing, advanced analytical skills, and complex query processing to retrieve data. To overcome these challenges, we integrate Large Language Models (LLMs) with the database system, enabling proficient communication without the need for SQL and Document query languages.

A. Purpose

The purpose of this research is not just to develop a database RAG system but to primarily focus on identifying the best LLMs for both SQL and NoSQL databases. This is achieved by evaluating their performance using various metrics such as accuracy, correctness, error rate, P50 latency and P99 latency.

B. Objective

- Advance analytics
- Reduce cost and time
- Reduce man power
- NLP driven database
- Increase productivity

II. SOFTWARE TOOLS AND TECHNOLOGIES

A. Python 3.12.4

This Retrieval Augmented Generation (RAG) system is built using the python programming language, which offers a rich set of AI and Natural Language Processing (NLP) libraries for developing LLM based applications efficiently. Notable libraries include LangChain, LlamaIndex, Google Generative AI, OpenAI and Hugging Face, which enable smooth integration of generative AI capabilities. Additionally, Python provides easy connectivity with both SQL database (MySQL, PostgreSQL, SQLite) and NoSQL databases (MongoDB, DynamoDB, Elasticsearch, Firebase). It also includes efficient data retrieval tools such as FAISS, an opensource vector search library designed for fast and scalable similarity searches in high dimensional spaces.

B. LangChain

Langchain is a python library that serves as a core package for building Retrieval Augmented Generation (RAG) application. It enables AI powered SQL query generation and document query formulation, allowing users to interact with databases using natural language. Langchain automatically converts user queries into SQL or document query languages based on the database type and schema. By leveraging LLMs, ChatPromptTemplate and database integration, langchain dynamically extracts schema information to generate accurate queries. Additionally, it integrates with langsmith to evaluate the accuracy and quality of the responses generated.

C. LangSmith

LangSmith is used to evaluate the performance of production grade LLM applications. In this RAG system, it is integrated to assess performance based on key metrics such as semantic accuracy, response correctness, error rate, P50 latency and P99 latency, comparing the ground truth values with the obtained responses. By leveraging LangSmith, we can analyze and compare the performance of multiple LLM models within the RAG system, helping to identify the best performing model for optimal results.

D. Flask

Flask is a python backend framework used in this RAG application to handle HTTP requests, process user queries, interact with the database, and manage responses between the database RAG system and the frontend.[17] It provides a lightweight and flexible REST API, allowing users to send natural language queries via POST requests to the /query-db endpoint. Flask process these queries by invoking the LangChain based SQL or document query generation pipeline, executing the generated queries on MySQL or MongoDB and returning the formatted response. Additionally, flask maintains conversation history to ensure context aware responses. It also supports Cross Origin Resource Sharing (CORS), enabling efficient communication between the frontend and backend. In this RAG system, flask plays a crucial role in integrating the evaluation module, receiving input queries, processing them through LangChain and facilitating interaction with LangSmith. Overall, flask act as a bridge between the user, LangChain, LangSmith and the database, ensuring smooth communication and efficient query processing.

E. Database

This RAG system is designed to support both SQL and NoSQL databases. The application architecture is separately built for SQL databases and NoSQL databases, utilizing different LLMs to each database type.

1. MySQL Database:

This application serves as the primary data source, where all structured information is stored and retrieved. It contains various tables with relevant business or institutional data that users can query using natural language. When a user submits a query, the system powered by the LangChain generates a corresponding SQL query, which is then executed on the MySQL database to fetch the required data. The database plays a crucial role in maintaining data integrity, consistency and accessibility, ensuring that the responses generated by LLM models are based on real time up to date information. Additionally, the SQLDatabase utility from LangChain enables dynamic schema extraction, allowing LLM models to understand the database structure and generate precise queries accordingly.

2. MongoDB Database:

MongoDB serves as the primary database for storing and retrieving JSON formatted data in this RAG system. It is integrated using the MongoDBDatabase class, which connects to the MongoDB Atlas cluster via a connection string. This application utilizes MongoDB to store JSON data and provides a key functionality. The ability to dynamically generate and execute MongoDB queries based on user questions using an AI driven generator. This process leverages LangChain and LLM models to analyze user queries, generate executable MongoDB queries and retrieve relevant data efficiently.

F. Visual Studio Code Editor

Visual Studio Code is a powerful code editor used for developing and managing this database RAG application. It provides an efficient coding experience with features such as syntax highlighting and debugging tools, making it easier to write, test and maintain python code. The integrated terminal in VS Code allows developers to run the flask application, install dependencies and execute MongoDB & SQL queries efficiently. Additionally, its built in support for virtual environment (venv) and docker helps in managing dependencies and streamlining deployment processes. VS Code also supports a provides a thunder client extension which is an alternative tool like postman, allowing developers to send HTTP requests directly from VS Code without switching to an external application. With thunder client, developers can test various API endpoints by making GET, POST, PUT and DELETE requests, providing live responses from the flask backend. It supports authentication methods, headers, query parameters and body formats like JSON and form data, making it ideal for interacting with the MongoDB and MySQL database through API calls.

III. RELATED WORKS

[1] Xiang Zhang, Khatoon Khedri and Reza Rawassizadeh conducted a study to evaluate the resource usage and accuracy of various LLM models including LLaMA 2 7B, LLaMA 2 13B, Mistral, Optimus-7B, SUS-Chat-34B, PlatypusYi-34B, NeuralHermes-2.5-Mistral-7B and Starling-LM-7B-Alpha for generating natural language queries in comparison to traditional SQL. The study advises against replacing relational databases with LLMs due to high resource demands and significant energy. [2] Xuanhe Zhou, Zhaoyan Sun and Guoliang Li has proposed the DB-GPT framework for fine tuning and designing LLMs specifically for databases to address challenges such as prompt design, lack of physical database, awareness and privacy concerns. [3] Zijin Hong and his team conducted a survey on using LLM models to convert natural language questions into SQL. While pretrained models have improved accuracy but challenges remain with complex queries and schema. [4] Johan Sandell and his team conducted research on comparing relational databases such as ArangoDB, MySQL and Neo4j, focusing on metrics like energy consumption, CPU usage and memory usage. Results show Neo4j outperforms MySQL and ArangoDB in handling connected graph data efficiently and better for big data application. [5] Mahdhindi and his team conducted a survey on the use of RAG in the legal domain, highlighting its ability to enhance LLM outputs by retrieving external knowledge. The survey reviews RAG methods, their stages, evaluation metrics, relevant datasets and legal specific applications. Identified challenges include high computational costs, complex in-house setups and retrieval inefficiencies. The accuracy of retrieval heavily affects response quality, especially due to complex legal language and lack of domain adapted models. [6] Binita Saha and his team introduced QulM-RAG, an improved RAG system for domain specific question answering utilizing a custom dataset and the LLaMA-3-8B-Instruct model. It reduces hallucination and improves accuracy compared to traditional RAG systems. [7] Rong Hu and his team introduces an Intelligence Custom Clearance Assistant (ICCA) RAG system which is used for custom clearance tasks. It integrated multimodal document parsing, hybrid storage and context aware generation to handle document complexity, query ambiguity and dynamic regulations. ICCA-RAG improves accuracy, contextual relevance and robustness to noisy queries. [8] Luis Bernardo Hernandez Salinas and his team proposed an intelligent driving assistant system (IDAS) using AI and RAG to enable multilingual voice based interaction with vehicles. It effectively answers user queries using car manual, with GPT-4o showing the best performance and smaller models

like GPT-4o mini offering faster responses. Challenges in real-time communication due to lack of database integration to track live data.

[9] Nouf Alabbasi and his team introduces TeleOracle which is a telecom specific RAG system built on phi-2 model. It uses a two stage retriever with semantic chunking and hybrid search, plus expanded context window for better query handling. TeleOracle achieves 81.2% accuracy, a 30% improvement over base phi2 and performance comparably to large LLM with higher faithfulness to context. It faces limitations in handling non textual data (eg., tables, graphs) and complex reasoning tasks. [10] Gulsum Budaglu and Hakan Emekci perform evaluation on RAG system, fine-tuning LLM and hybrid of both models on the datasets like Stanford Question Answering (SQuAD) and Microsoft Machine Reading Comprehension (MS MARCO) and SQL create table statements. Fine-tuning offers strong semantic accuracy but uses heavy resource, while RAG is efficient in resource usage but less precise. The hybrid model balances both, achieving the highest semantic and contextual performance, especially on structured data. However, it is also a resource intensive.

[11] Yosua Setyawan Soekamto and his team proposed a SKYRAG (Separated Keyword Retrieval Augmentation Generation) a keyword based RAG system that integrates LLMs to generate personalized learning paths by retrieving relevant Massive Open Online Course (MOOC) content. It outperforms naïve (traditional) RAG in accuracy, relevance and learner satisfaction across multiple domain. Future plans include real-time integration, multi-language support and advanced learning analytics to enhance scalability and adaptability in personalized education. However, the study does not provide details on the integration of the database with the SKYRAG system, nor does it specify the type of database employed, such as whether it utilizes a SQL based or NoSQL based architecture.

[12] Menna Fateen and her team introduces a modular RAG based system for automatic short answer scoring with ASAS-F (Automatic Short Answer Scoring - Feedback) using ColBERT, which improves scoring and feedback generation without heavy fine-tuning or manual prompt engineering. Using RAG for few shot selection and automatic prompt generation. While it reduces dependence on expensive fine-tuning. Feedback generation quality still lags behind in BLEU and ROUGE score compared to fine-tuned approaches. However, the paper does not describe how the underlying data storage or database system is integrated with RAG system. [13] Jun Xu and his team introduces ChatTf, a knowledge graph enhanced Q&A system designed to reduce hallucination in domain like Chinese folklore. It uses custom ontology Traditional Folklore Ontology (TFOnto), a triple extraction framework Zero-Shot Traditional Folklore Triplet Extraction (ZFTE) and build a knowledge graph - Traditional Folklore Knowledge Graph (TFKG). A dual stage TFKG-RAG framework integrates this knowledge with LLMs to improve accuracy. ChatTf achieved 96.7% accuracy on a folklore Q&A test set, surpassing existing methods. Challenges includes lack of integration of folklore databases which limits intelligent connectivity and effective knowledge retrieval.

[14] Majjed AL-Qatf and his team proposes RAG data space models (RAD-DSMs), integrating RAG with data spaces to improve access to diverse, high quality external knowledge while maintaining secure data sharing. The unified lifecycle aims to enhance both LLM performance and data space services. Despite its potential, the approach faces challenges in data quality, privacy and system integration. [15] Alexander Tobias Neumann and his team proposed MoodleBot, and LLM based chatbot integrated into Moodle to support self regulated learning using the RAG approach and evaluated via the Technology Acceptance Model (TAM), it showed 88% accuracy and positive student feedback. Limitations, the study only used GPT-4, lacking comparison with other LLMs. [16] Shijun Ge and team have proposed a study showing that combining large language models (LLMs) with traditional concept generation methods (like brainstorming and TRIZ) using chain-of-thought (CoT) techniques improves conceptual design. The findings confirm that integrating LLMs with structured design methods is effective, suggesting broader future use in design fields. [18] António Mestre and his team have proposed an architecture to implement RAG effectively using relational databases as data sources, but they did not propose an architecture for NoSQL databases. [19] Sergey Vichev and team explores improving Text-to-SQL tasks using a RAG approach by dynamically augmenting LLM prompts with database context.

IV. PROPOSED SYSTEM

System Architecture

1. MySQL RAG System

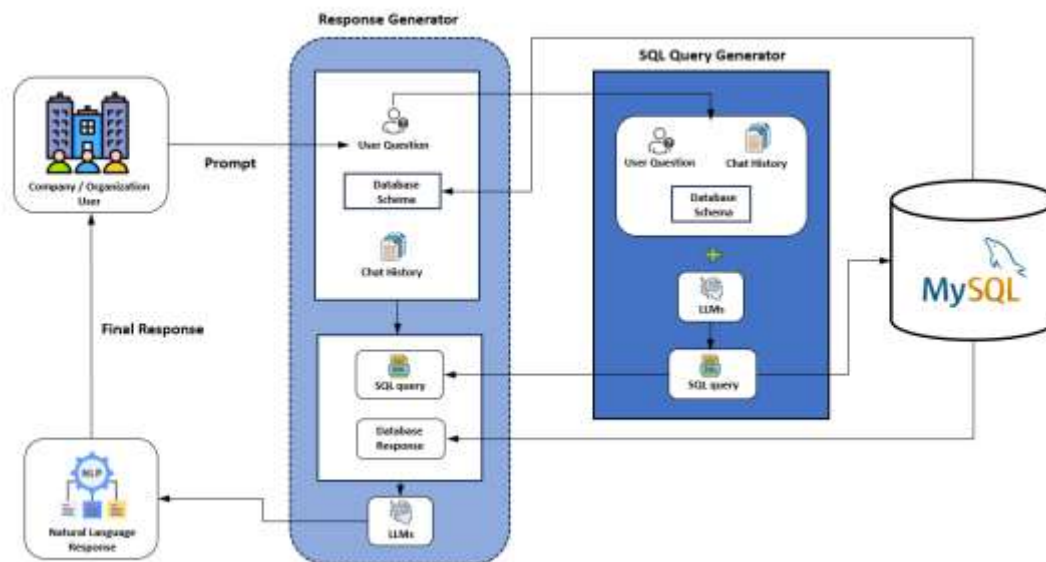


Fig. 1 MySQL RAG Architecture

The proposed solution consists of three main components namely **response_generator**, **db_query_generator** and **MySQL database**. The **response_generator** serves as the primary component of the RAG system. It is connected to the database schema using the **SQLDatabase** utility from LangChain, along with the **chat_history** from previous conversations. When a user submits a question, the **response_generator** requires a corresponding SQL query to retrieve data from the database. To generate this SQL query, the **response_generator** communicates with the **db_query_generator** component, passing parameters such as the database schema, user question and chat history. The **db_query_generator** then utilizes an LLM model to generate the appropriate SQL query and returns it to the **response_generator**. The **response_generator** executes the generated SQL query on the database and retrieves the results. Finally, the **response_generator** uses the LLM model to formulate a natural language response based on the database result, chat history and user question. This response is then returned to the user as the final answer.

Example:

User Question: Give me student name list

Db_query_generator : SELECT name FROM students;

Response_generator : The list of students are a, b,c,d,e..etc.

Example 2:

User Question: How many faculty members are working at the college?

Db_query_generator: SELECT COUNT(*) AS total_faculty FROM faculty_tbl;

Response_generator : There are totally 16 members are working in the college.

Example 3:

User Question: Student with more than 80% of mark?

Db_query_generator : select * from students where Mark >80;

Response_generator : The students who scored more than 80% marks are A, B, C, D and M.

2. MongoDB RAG System

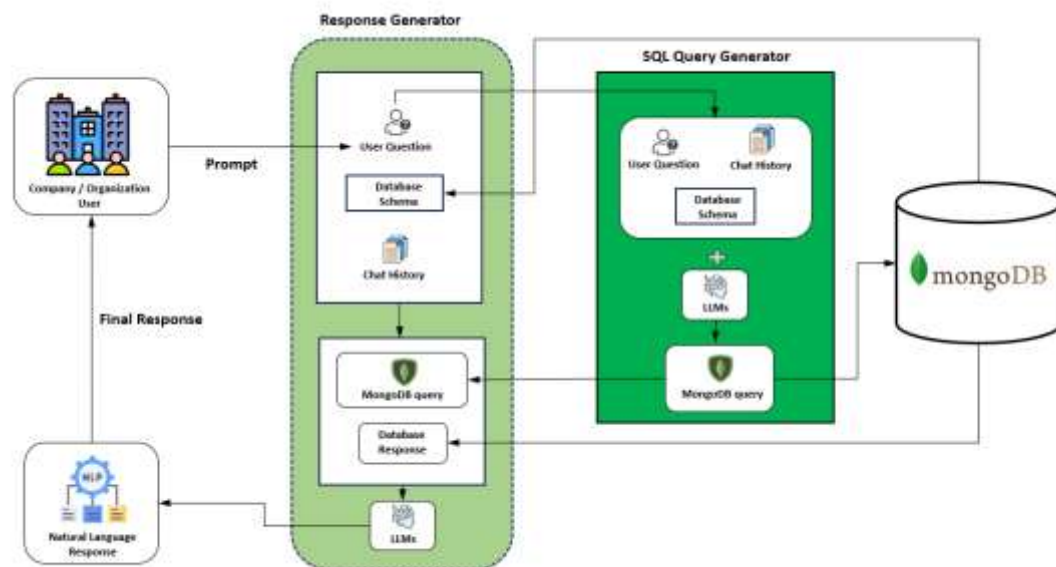


Fig. 2 MongoDB RAG Architecture

The proposed solution for the MongoDB based RAG system closely follows the architecture of the MySQL based RAG system, with the key difference being the use of MongoDB as the database and a different connectivity class. It consist of three main components namely **response_generator**, **MongoDB_query_generator** and the MongoDB database. Similar to the SQL based approach, the **response_generator** connects to the MongoDB schema using a custom **MongoDBDatabase** class, which facilitates connectivity to MongoDB Atlas. It also maintains the chat history containing previous user interaction. When a user submits a question,, the **response_generator** requires a corresponding document query to retrieve relevant data from MongoDB. To generate this query, it communicates with the **MongoDB_query_generator** component by passing parameters such as the database schema, user question and chat history. The **MongoDB_query_generator** then utilizes an LLM model to generate the appropriate MongoDB query and returns it to the **response_generator**. The generated query is then executed on the MongoDB database to fetch the required data. Finally, the **response_generator** uses the LLM model to formulate a natural language response based on the retrieved data, chat history and user question. This response is then returned to the user as the final response.

User Question: Give me authors name list

MongoDB_query_generator

`collection.distinct("authorname")`

Response_generator : The list of authors are 'Karan', 'Pradeep', 'Rosinii', 'Sibi', 'Yugesh Karan', 'ajayvarsanr', 'haricharan_1133', 'kaali', 'tharun'.

B. LLM Chosen

The proposed system was evaluated using twelve of the most advanced Large Language Models were selected from the top five leading AI companies based free and paid models. These models were selected from the top five leading AI companies based on criteria such as innovation, market impact and funding, as highlighted in the article [“Forbes 2025 AI 50 List”](#) . The selected LLMs include: meta-llama/Llama-4-Scout-17B-16E-Instruct, meta-llama/llama-4-maverick-17b-128e-instruct, llama3-70b-8192, llama-3.3-70b-versatile, gemma-3-27b-it, gemini-1.5-flash-8b, gemini-1.5-flash, claude-3.7-sonnet-20250219, claude-3-sonnet-20240229, claude-3-haiku-20240229, gpt-4.1 and gpt-4.1-mini. The detailed descriptions are given in the table 1

TABLE 1: LLM models

S.NO	Model Name	Company	Licence	Launch Date
1	gpt-4.1	OpenAI	Proprietary	April 14, 2025
2	gpt-4.1-mini	OpenAI	Proprietary	April 14, 2025
3	claude-3-7-sonnet-20250219	Anthropic	Proprietary	February 24, 2025
4	claude-3-5-sonnet-20241022	Anthropic	Proprietary	June 20, 2024
5	claude-3-5-haiku-20241022	Anthropic	Proprietary	November 4, 2024
6	meta-llama/Llama-4-Scout-17B-16E-Instruct	Meta AI	Open (Free) Source	April 5, 2025
7	meta-llama/llama-4-maverick-17b-128e-instruct	Meta AI	Open (Free) Source	April 5, 2025
8	llama3-70b-8192	Meta AI	Open (Free) Source	April 18, 2024
9	llama-3.3-70b-versatile	Meta AI	Open (Free) Source	December 6, 2024
10	gemma-3-27b-it	Google AI	Open (Free) Source	March 10, 2025
11	gemini-1.5-flash-8b	Google AI	Proprietary	April 15, 2024
12	gemini-1.5-flash	Google AI	Proprietary	April 15th, 2024.

C. Database of RAG system

In this research, we have chosen MySQL as the database to test LLM models for the SQL based RAG system and MongoDB to test models for the NoSQL based RAG system.

1. MySQL database

We selected a simple college database for evaluation. It contains the following tables and attributes.

TABLE 2: Relations and columns of the database

Table Name	Description	Key Attributes / Column Name
Students	Stores student-related details.	Student name, admission number, roll number, degree, department, semester, section, gender, hostel or day scholar, attendance percentage
Faculty_tbl	Contains faculty member information.	Faculty name, email id, subject id, department
Subject_tbl	Stores subject-	Subject name, subject id, subject type

	related information.	
Time_tble	Stores class timetable data based on semester and department.	day, hour, semester, department, section, faculty name, subject name, faculty email
Attendance_tbl	Stores student attendance records.	Roll number, student name, semester, department, section, date, day, hour, subject name, attendance status (Present/Absent), faculty name

Schema Architecture

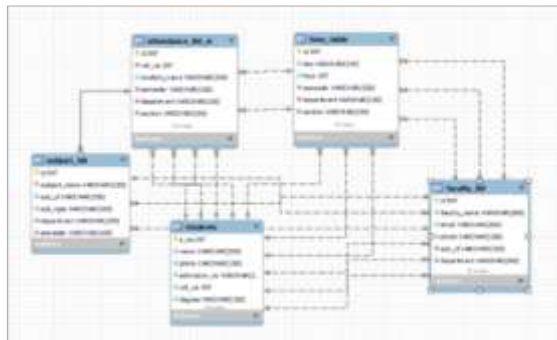


Fig. 3 MySQL Schema Architecture

2. MongoDB Database

We have chosen a E-learning blog database for the MongoDB based RAG system. It contains details of various authors and their associated tech posts based on the categories such as GenAI, Data Science, Blockchain, AI/ML, Cyber Security, IoT, Embedded System and Web Development. The schema of the MongoDB is described below:

TABLE 3: Author schema details

Field	Type	Description
authorname	String	Name of the author
email	String	Unique email (validated with regex)
password	String	Encrypted password using bcrypt

profile	String	Optional profile image URL
followers	[String]	List of followers (emails IDs)
following	[String]	List of authors this user follows
posts	[PostSchema]	Embedded list of blog posts authored
notification	[NotificationSchema]	Notifications received by this author

The Author schema act as the central model that connects to the other parts of the application such as posts, followers and notifications.

TABLE 4: Post Schema details

Field	Type	Description
title	String	Post title
image	String	Optional corresponding post image URL
links	[{ title, url }]	Optional list of resource links with unique URLs
documents	[String]	List of PDF/document file names (unique entries)
description	String	Main content of the post
category	String	Post category
views	[String]	List of users who viewed (email IDs)
likes	[String]	List of users who liked (email IDs)
messages	[MessageSchema]	Comments/messages associated with this post
timestamp	Date	Creation time (auto set to now)

The post schema defines the structure of blog content items created by authors. It encapsulates the post's content and metadata.

TABLE 5: Message schema details

Field	Type	Description
user	String	Author name of the message
message	String	The content of the message
profile	String	Optional profile image
timestamp	Date	Message time (default: now)

Message schema is embedded directly inside a post, which allows users to add comments to the corresponding posts.

TABLE 6: Notification Schema detail

Field	Type	Description
postId	ObjectId	Linked to the post (ref: 'Post')
user	String	User who triggered the notification
message	String	Description of the notification
profile	String	Optional profile image of triggering user
url	String	URL associated with the notification
authorEmail	String	The recipient of the notification
timestamp	Date	Notification time (default: now)

Notification schema models system generated messages that are sent to users to inform them about new post content published by authors they follow.

Overall Schema Architecture:

Fig. 4 MongoDB Schema Architecture

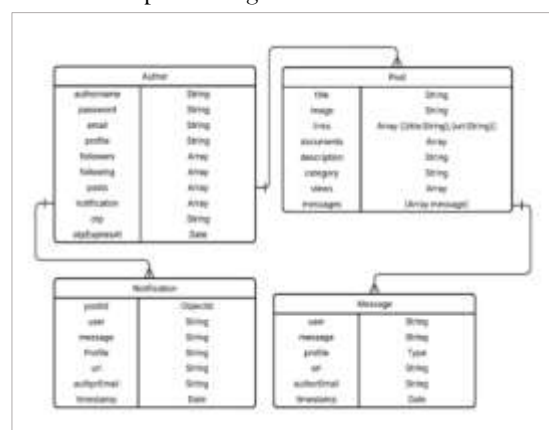
This database is structured with a main **Author** schema, which embeds both the **Post** schema and the **Notification** schema, where the Post schema further embed with the **Message** schema.

Relationship Summary:

- Author à Posts: one-to-many
- Post à Message: one-to-many
- Author à Notifications: one-to-many
- Notification à Post: linked by postId

C. Evaluation Metrics

The performance of both database RAG system was evaluated by the metrics such as Accuracy, Correctness, Error Rate, P50 Latency and P99 Latency. Each of the metric is very important to analyze the efficiency of the model. Each metrics are carefully calculated by LangSmith which is integrated to RAG system. The model which perform good on all metrics over other model will be identified as the



optimal and best model for database RAG system. Through evaluation the research will provide two different results for MySQL and MongoDB RAG system respectively.

1.) Accuracy: The **accuracy metric** is used to measure the **semantic similarity** between the model-generated response and the ground truth reference. [20] This semantic accuracy is evaluated using **Sentence-BERT (SBERT)** combined with **cosine similarity functions** in Python. SBERT helps calculate the similarity score between two responses that may use different words but convey the same meaning. By integrating SBERT with cosine similarity, the evaluation becomes more robust and semantically aware. In the evaluation pipeline, SBERT encodes both the model-generated response and the reference answer into vector embeddings. Once both sentences are represented as high-dimensional vectors, **cosine similarity** is computed between them. This metric measures the cosine of the angle between the two vectors in a multi-dimensional space. The accuracy score ranges from **0 to 1**, where **1** indicates that the vectors are identical in direction (i.e., high semantic similarity), and **0** indicates orthogonality (i.e., no semantic similarity).

Pseudocode for accuracy function:

Algorithm 1: Accuracy function to find semantic similarity

```

1: FUNCTION accuracy(outputs, reference_outputs):
2: output_text ← outputs["response"] OR empty string if not found
3: reference_text ← reference_outputs["output"] OR empty
  string if not found
4: IF output_text is empty OR reference_text is empty
  THEN
5: PRINT "Warning: Missing output text, returning 0
  similarity."
6: RETURN 0.0
7: model ← load model from "sbert_model.pkl"
8: output_embedding ← encode output_text using model (as tensor)
9 reference_embedding ← encode reference_text using
  model (as tensor)

10 similarity ← compute cosine similarity between
  output_embedding and reference_embedding

11 RETURN similarity

```

The pseudocode describes a function designed to measure the similarity between two pieces of text: one generated by a model (output_text) and one considered as the correct or reference output (reference_text). First, it safely extracts these texts from two input dictionaries. If either text is missing, the function returns a similarity score of 0.0 and prints a warning to avoid runtime errors. Otherwise, it proceeds to load a pre-trained Sentence-BERT (SBERT) model, which converts the texts into numerical embeddings suitable for comparison. The function then computes the cosine similarity between the two embeddings, which quantifies how similar the two texts are in meaning. Finally, it returns this similarity score as a float, providing a measure of accuracy for the generated output in terms of semantic similarity to the reference.

2.) Correctness: Correctness metric is used to find the truthiness of the response generated. It is represented as a binary score 1 (correct) or 0 (incorrect) by checking whether the model output is more than the threshold value. For this system the threshold value is set as 0.69 of semantic similarity calculated by SBERT and cosine function.

Pseudocode for Correctness function:

Algorithm 2: Correctness function find the truthiness of the RAG response.

```

1:      Function      Correctness(outputs,
reference_outputs):
2: Call the accuracy function with outputs
and
   reference_outputs
3: Store the returned similarity score in a
variable named
   similarity_score
4: If similarity_score is greater than 0.69:
5:   Return 1 // Indicates output is correct
6: Else:
7:   Return 0 // Indicates output is incorrect

```

The Correctness function evaluates whether the similarity between a generated output and a reference output is above a specified threshold (0.69). It does this by first calling the accuracy function, which returns a similarity score between the two texts based on sentence embeddings. This score is stored in a variable. The function then checks if the score exceeds the threshold of 0.69. If it does, the function returns 1, indicating that the output is considered correct or sufficiently similar. Otherwise, it returns 0, signifying that the output does not meet the similarity standard. This function is useful for binary classification of correctness in evaluation tasks.

Example:

Reference: "The capital of France is Paris." *Model 1 response: "Paris is the capital of France."

Cosine similarity : 0.96 à Above threshold value (>0.69) à **Correctness = 1**

* Model 2 response: "Berlin is the capital of France."

Cosine similarity : 0.50 à Above threshold value (<0.69) à **Correctness = 0**

3.) Error Rate: The error rate will be calculated in binary 0 or 1. Whereas, if the model does not provided an output, then it is marked as 1 else 0.

Pseudocode for Error Rate function:

Algorithm 3: Error rate function to calculate the total errors

```

1:      FUNCTION      error_rate(outputs,
reference_outputs):

2: SET output_text TO value of "response"
key from
   outputs, default to empty string
3: IF output_text is empty OR missing:
4:   RETURN 1.0 // Indicates 100% erro
5: ELSE
6:   RETURN 0.0 // Indicates no error

```

The pseudocode for the error_rate function is de+signed to calculate the likelihood of a failure in generating a valid response. It takes in two dictionaries outputs and reference_outputs but primarily uses the outputs dictionary. It attempts to retrieve the value associated with the "response" key, defaulting to an empty string if the key is missing. If the output_text is empty or not present, the function concludes that an error has occurred and returns an error rate of 1.0, which corresponds to a 100% error. Otherwise, it returns 0.0, indicating no error. This simple binary approach effectively flags instances where the model failed to produce any meaningful output.

4.) Latency: **P50** and **P99 latencies** are used to evaluate the performance of the database RAG system in terms of the time delay taken by the LLM to respond to user queries. These metrics help assess how quickly the model is capable of generating a response. P50 latency is used to find the time it takes for 50% of the requests to be served, half of the requests take less than this time and half take longer. P99 latency

is the time it takes for 99% of the requests to be completed i.e., only 1% of the requests take longer than this. It is used to understand the worst case and outlier behaviour

D. Model Performance Evaluation

The performance evaluation is conducted using different LLM models with both MySQL and MongoDB RAG systems separately. The evaluation involves testing each model on both RAG systems with 10 questions from their respective databases, along with corresponding ground truth references. These tests are connected to LangSmith dataset experiments, which compare the LLM responses to the ground truth values based on metrics such as Accuracy, Correctness, Error rate, P50 and P99 latency

A. MySQL RAG Evaluation

1. Gpt-4.1: The experiment was conducted on gpt-4.1 using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation gpt 4.1 rag](#).

Query-1: List of subjects

TABLE 7: Ground Truth Reference VS Model gpt-4.1 output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	Here is the list of subjects:\n\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS\n\nIf you need details about any specific subject or their schedule, please let me know!

Accuracy: 0.81/ 1.0,Correctness: 1/1,Error rate:0, Latency: 5.58s

2. Gpt-4.1-mini: The experiment was conducted on gpt-4.1-mini using 10 test questions. Detailed information for 1 of the queries, along with their corresponding metric measurements, is provided below. To view the full validation, click on [evaluation 4.1 mini rag](#).

Query-1: List of subjects

TABLE 8: Ground Truth Reference VS Model gpt-4.1-mini output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application	The list of subjects available in the database are:\n\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n-

Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	Web Mining Lab \n- Data Visualization Lab \n- Mini Project \n- HRDC \n- MENTOR \n- NSS \n\nLet me know if you need more details about any specific subject!
--	---

Accuracy: 0.96/1.0, Correctness:1/1, Error rate: 0, Latency: 5.36s

3. Claude-3-7-sonnet-20250219: The experiment was conducted on claude-3-7-sonnet-20250219 using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation claude 3.7 sonnet rag](#).

Query-1: list of subjects

TABLE 9: Ground Truth Reference VS Model claude-3-7-sonnet-20250219 output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	Here is the complete list of subjects available in our curriculum:\n\n**Theory Subjects:**\n1. Mobile Computing (21CSE12)\n2. Information Retrieval System (21INT01)\n3. Web Mining (21CYS04)\n4. Internet Of Things (21IOT02)\n5. Sensors and Transducers (21OEE13)\n6. Predictive Modeling (21AID07)\n7. Data Visualization (21AID05)\n8. NSS (123)\n\n**Practical Subjects:**\n1. Predictive Modeling Lab (21AIDP5)\n2. Mobile Application Development Lab (21INTP2)\n3. Web Mining Lab (21CYSP2)\n4. Data Visualization Lab (21AIDP4)\n5. Mini Project (21CSEMP)\n\n**Student Development Courses:**\n1. HRDC (HRDC)\n2. MENTOR (MENTOR)\n\nAll these subjects are currently

	offered for AI&DS department in Semester VI.
--	--

Accuracy: 0.54/1.0, Correctness: 0/1, Error rate: 0, Latency: 5.153s

4. claude-3-5-sonnet-20241022: The experiment was conducted on claude-3-7-sonnet-20250219 using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation claude 3.5 sonnet rag](#).

Query-1: list of subjects

TABLE 10: Ground Truth Reference VS Model claude-3-5-sonnet-20250219 output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	Here are all the subjects offered for AI&DS department in the VI semester:\n\nTheory Subjects:\n1. Data Visualization (21AID05)\n2. Information Retrieval System (21INT01)\n3. Internet Of Things (21IOT02)\n4. Mobile Computing (21CSE12)\n5. Predictive Modeling (21AID07)\n6. Sensors and Transducers (21OEE13)\n7. Web Mining (21CYS04)\n\nPractical Subjects:\n1. Data Visualization Lab (21AIDP4)\n2. Mobile Application Development Lab (21INTP2)\n3. Predictive Modeling Lab (21AIDP5)\n4. Web Mining Lab (21CYSP2)\n5. Mini Project (21CSEMP)\n\nStudent Development Courses:\n1. HRDC\n2. MENTOR\n\nAdditional Course:\n1. NSS\n\nThe curriculum offers a good mix of theoretical and practical subjects, focusing on various aspects of Artificial Intelligence and Data Science

Accuracy: 0.53/1.0, Correctness: 0/1, Error rate: 0 Latency: 7.44s

5. Claude-3-5-haiku-20241022: The experiment was conducted on claude-3-5-haiku-20241022 using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation claude 3.5 haiku rag](#).

Query-1: List of subjects

TABLE 11: Ground Truth Reference VS Model Claude-3-5-haiku-20241022 output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	Based on the SQL query results, here's a comprehensive list of subjects for the AI&DS (Artificial Intelligence and Data Science) department in the VI semester:\n\nTheory Subjects:\n1. Data Visualization\n2. Information Retrieval System\n3. Internet Of Things\n4. Mobile Computing\n5. Predictive Modeling\n6. Sensors and Transducers\n7. Web Mining\n\nPractical/Lab Subjects:\n1. Data Visualization Lab\n2. Mobile Application Development Lab\n3. Mini Project\n4. Predictive Modeling Lab\n5. Web Mining Lab\n\nStudent Development Subjects:\n1. HRDC\n2. MENTOR\n3. NSS\n\nKey Observations:\n- All subjects are from the AI&DS department\n- All subjects are for the VI semester\n- There's a mix of theory, practical, and student development courses\n- Some subjects have corresponding lab courses\n\nWould you

	like me to provide any additional details about these subjects, such as their subject codes or types?
--	---

Accuracy: 0.72/1.0, Correctness: 1/1, Error rate: 0, Latency: 8.94s

6.Meta-llama/llama-4-maverick-17b-128e-instruct: The experiment was conducted on meta-llama/llama-4-maverick-17b-128e-instruct using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation llama 4 maverick rag](#).

Query-1: List of subjects

TABLE 12: Ground Truth Reference VS Model llama-4-maverick output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	The list of subjects available in the database is as follows:\n\n1. Mobile Computing\n2. Information Retrieval System\n3. Web Mining\n4. Internet Of Things\n5. Sensors and Transducers\n6. Predictive Modeling\n7. Data Visualization\n8. Predictive Modeling Lab\n9. Mobile Application Development Lab\n10. Web Mining Lab\n11. Data Visualization Lab\n12. Mini Project\n13. HRDC\n14. MENTOR\n15. NSS\n\nThese subjects are part of the curriculum for various departments and semesters.

Accuracy: 0.86/ 1.0, Correctness: 1/1, Error rate: 0, Latency: 68.45s

7.Meta-llama/Llama-4-Scout-17B-16E-Instruct: The experiment was conducted on meta-llama/Llama-4-Scout-17B-16E-Instruct using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation llama 4 scout rag](#).

Query-1: List of subjects

TABLE 13: Ground Truth Reference VS Model llama-4-scout output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and	Based on the data, I have retrieved the list of subjects for you.\n\nHere is the list of subjects:\n\n* Mobile Computing\n* Information Retrieval System\n* Web Mining\n* Internet Of Things\n* Sensors and Transducers\n* Predictive Modeling\n* Data Visualization\n*

Transducers\n-	Predictive Modeling
Predictive Lab\n*	Mobile
Modeling\n-	Application
Data Development	Lab\n*
Visualization\n-	Web Mining Lab\n*
Predictive Data	Visualization
Modeling Lab\n-	Lab\n* Mini Project\n*
Mobile HRDC\n*	
Application MENTOR\n*	
Development NSS\n\nHere's	a
Lab\n- Web summary	of the
Mining Lab\n-	results.\n\nThe SQL
Data query	used to generate
Visualization this	information
Lab\n- Mini is:\n\n```\nsql\nSELECT	
Project\n-	subject_name FROM
HRDC\n-	subject_tbl;\n.
MENTOR\n-	
NSS	

Accuracy: **0.79/ 1.0**, Correctness: **1/1**, Error rate: **0**, Latency: **66.58s**

8.llama-3.1-8b-instant: The experiment was conducted on llama-3.1-8b-instant using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation llama3.1 8b rag](#).

Query-1: List of subjects

TABLE 14: Ground Truth Reference VS Model llama-3-instant-8b output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	Based on the data we have in the database, here is the list of subjects:\n\n1. Mobile Computing\n2. Information Retrieval System\n3. Web Mining\n4. Internet Of Things\n5. Sensors and Transducers\n6. Predictive Modeling\n7. Data Visualization\n8. Predictive Modeling Lab\n9. Mobile Application Development Lab\n10. Web

	Mining Lab\n11. Data Visualization Lab\n12. Mini Project\n13. HRDC\n14. MENTOR\n15. NSS
--	---

Accuracy: **0.84/ 1.0**, Correctness: **1/1**, Error rate: **0**, Latency: **66.91s**

9.llama3-70b-8192: The experiment was conducted on llama3-70b-8192 using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation llama3 70b rag](#).

Query-1: list of subjects

TABLE 15: Ground Truth Reference VS Model llama3-70b-8192 output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	Here is the list of subjects:\n\n1. Mobile Computing\n2. Information Retrieval System\n3. Web Mining\n4. Internet Of Things\n5. Sensors and Transducers\n6. Predictive Modeling\n7. Data Visualization\n8. Predictive Modeling Lab\n9. Mobile Application Development Lab\n10. Web Mining Lab\n11. Data Visualization Lab\n12. Mini Project\n13. HRDC\n14. MENTOR\n15. NSS

Accuracy: **0.72/ 1.0**, Correctness: **1/1**, Error rate: **0**, Latency: **62.50s**

10.Gemini-1.5-flash: The experiment was conducted on gemini-1.5-flash using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation gemini 1.5 flash rag](#).

Query-1: List of subjects

TABLE 16: Ground Truth Reference VS Model Gemini-1.5-flash output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	The subjects offered are: Mobile Computing, Information Retrieval System, Web Mining, Internet of Things, Sensors and Transducers, Predictive Modeling, Data Visualization, Predictive Modeling Lab, Mobile Application Development Lab, Web Mining Lab, Data Visualization Lab, Mini Project, HRDC, MENTOR, and NSS.\n

Accuracy: 0.79/ 1.0, Correctness: 1/1, Error rate: 0, Latency: 2.12s

11.Gemini-1.5-flash-8 b: The experiment was conducted on Gemini-1.5-flash-8 b using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation gemini 1.5 flash 8b rag](#).

Query-1: List of subjects

TABLE 17: Ground Truth Reference VS Model Gemini-1.5-flash-8 b output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	The subjects are: Mobile Computing, Information Retrieval System, Web Mining, Internet Of Things, Sensors and Transducers, Predictive Modeling, Data Visualization, Predictive Modeling Lab, Mobile Application Development Lab, Web Mining Lab, Data Visualization Lab, Mini Project, HRDC, MENTOR, and NSS.\n

Accuracy: 0.78/ 1.0, Correctness: 1/1, Error rate: 0, Latency: 1.92s

12.Gemma-3-27b-it: The experiment was conducted on gemma-3-27b-it using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation gemma 3 27b it rag](#).

Query-1: List of subjects

TABLE 18: Ground Truth Reference VS Model Gemma-3-27 b-it output for query 1

Ground Truth Reference	Model Output
The list of subjects in the database includes:\n- Mobile Computing\n- Information Retrieval System\n- Web Mining\n- Internet Of Things\n- Sensors and Transducers\n- Predictive Modeling\n- Data Visualization\n- Predictive Modeling Lab\n- Mobile Application Development Lab\n- Web Mining Lab\n- Data Visualization Lab\n- Mini Project\n- HRDC\n- MENTOR\n- NSS	null

Accuracy: 0.0/ 1.0, Correctness: 0/1, Error rate: 1, Latency: 1.09s

Models Benchmark for MySQL RAG system :

The performance of each model was evaluated, and the metric details for the MySQL RAG system are presented in the figures below:

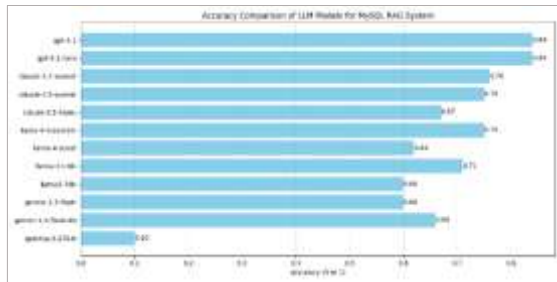


Fig. 5 Accuracy Comparison of LLM models of MySQL RAG system.

From the MySQL RAG system experiment shown in figure 5, the overall accuracy was measured, and it was observed that gpt-4.1 and gpt-4.1-mini achieved the highest accuracy of 0.84/1.0 (i.e., 85%) compared to other models, while gemma-3-27-it recorded the lowest accuracy of 0.10/1.0 (i.e., 10%).

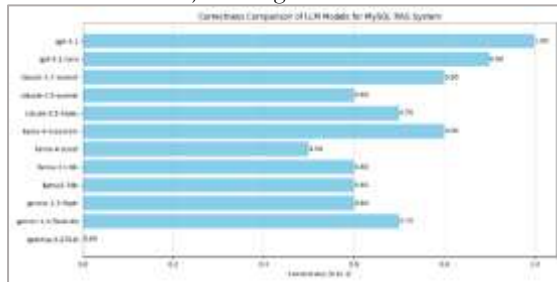


Fig. 6 Correctness Comparison of LLM models of MySQL RAG system.

The correctness metric was evaluated for all models shown in figure 6, and it shows that the GPT-4.1 model achieved the highest correctness score of 1/1 (i.e., 100%), indicating that it provides highly reliable responses. On the other hand, the Gemma-3-27-it model recorded the lowest correctness score of 0%, making it the least accurate among all the models evaluated.

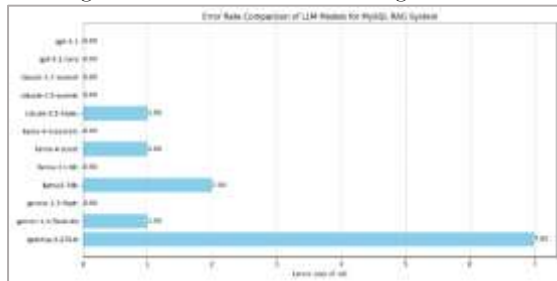


Fig. 7 Error Rate of LLM models of MySQL RAG system.

It is observed that on experiment out of twelve models seven models are executed with 0% of error rate that are gpt-4.1, gpt-4.1-mini, claude-3.7-sonnet, claude-3.5-sonnet, llama-4-maverick, llama-3.1-8b and gemini-1.5-flash and the maximum error rate was observed in gemma-3-27b-it with the error rate of 70% which is shown in figure 7.

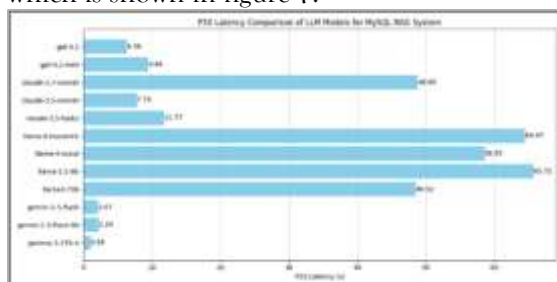


Fig. 8 P50 latency of LLM models of MySQL RAG system.

Based on the experimental data from Figure 8, the P50 latency was calculated using LangSmith. It was observed that gemini-1.5, gemini-1.5-flash-8b, gemma-3-27b-it, gpt-4.1, gpt-4.1-mini, and claude-3.5-sonnet

took less than 10 seconds to execute half of the query requests, with durations of 2.07s, 2.24s, 0.99s, 6.37s, 9.46s, and 7.79s respectively, compared to the other models.

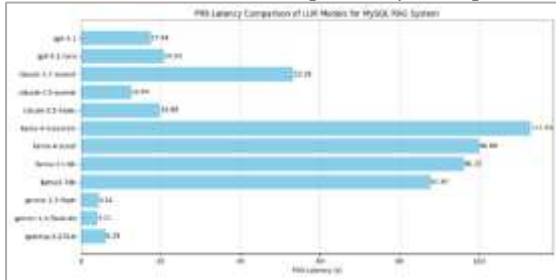


Fig. 9: P99 latency of LLM models of MySQL RAG system.

P99 latency of the MySQL RAG system is shown in figure 9 which shows that gemini-1.5, gemini-1.5-flash-8b, gemma-3-27b-it, gpt-4.1, gpt-4.1-mini, and claude-3.5-sonnet are taken less than 15 seconds to complete all 10 queries response, in that still gemini-1.5-flash and gemma-3-27b-it are generate responses with errors which is show in figure 7.

Models Benchmark for MySQL RAG system :

TABLE 19: Model comparison based on overall metrics of each model

S.NO	LLM Model	Accuracy	Correctness	Error Rate	P50 Latency	P99 Latency
1	gpt-4.1	0.84 / 1.0	1 / 1.0	0 / 10	6.36s	17.54s
2	gpt-4.1-mini	0.84 / 1.0	0.90 / 1.0	0 / 10	9.46s	20.81s
3	claude-3.7-sonnet-20250219	0.76 / 1.0	0.80 / 1.0	0 / 10	48.80s	53.39s
4	claude-3.5-sonnet-20241022	0.75 / 1.0	0.60 / 1.0	0 / 10	7.79s	12.64s
5	claude-3.5-haiku-20241022	0.67 / 1.0	0.70 / 1.0	1 / 10	11.77s	19.88s
6	meta-llama/llama-4-maverick-17b-128e-instruct	0.75 / 1.0	0.80 / 1.0	0 / 10	64.47s	112.95s
7	meta-llama/llama-4-scout-17b-16e-instruct	0.62 / 1.0	0.50 / 1.0	1 / 10	58.55s	99.99s
8	llama-3.1-8b-instant	0.71 / 1.0	0.60 / 1.0	0 / 10	65.72s	96.15s
9	llama3-70b-8192	0.60 / 1.0	0.60 / 1.0	2 / 10	48.52s	87.87s
10	gemini-1.5-flash	0.60 / 1.0	0.60 / 1.0	0 / 10	2.07s	4.52s
11	gemini-1.5-flash-8b	0.66 / 1.0	0.70 / 1.0	1 / 10	2.24s	4.21s
12	gemma-3-27b-it	0.10 / 1.0	0.0 / 1.0	7 / 10	0.99s	6.29s

B. MongoDB RAG Evaluation

1. gpt-4.1: The experiment was conducted on gpt-4.1 using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation gpt 4.1 rag](#).

Query-1: Which author has the highest number of followers?

TABLE 20: Ground Truth Reference VS Model gt-4.1 output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	Based on the information retrieved from the blog's database, the author with the highest number of followers is haricharan_1133. This author has three followers: yugeshkaran01@gmail.com, test@dsuniversity.ac.in, and yugeshkaran001@gmail.com. Having the largest follower base among all authors highlights haricharan_1133's influence and presence on the platform. This level of engagement suggests that their content resonates well with readers, attracting a broader audience and possibly fostering active discussions. If you are interested in exploring the specific posts or contributions made by haricharan_1133, feel free to let me know for more details.

Accuracy: 0.84/1.0, Correctness: 1/1, Error rate: 0, Latency: 22.66s

2. gpt-4.1-mini: The experiment was conducted on gpt-4.1-mini using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation gpt 4.1 mini rag](#).

Query-1: Which author has the highest number of followers?

TABLE 21: Ground Truth Reference VS Model gt-4.1-mini output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	The author with the highest number of followers is haricharan_1133, who has three followers including yugeshkaran01@gmail.com, test@dsuniversity.ac.in, and yugeshkaran001@gmail.com. This author leads in terms of follower count among the registered contributors on our blog platform. If you would like, I can provide more details about their posts or other statistics related to their contributions.

Accuracy: 0.87/1.0, Correctness: 1/1, Error rate: 0, Latency: 6.43s

3. claude-3-7-sonnet-20250219: The experiment was conducted on claude-3-7-sonnet-20250219 using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation 3.7 sonnet rag](#).

Query-1: Which author has the highest number of followers?

TABLE 22: Ground Truth Reference VS Model claude-3-7-sonnet-20250219 output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	Null.

Accuracy: 0.0/1.0, Correctness: 0/1, Error rate: 1, Latency: 24.96s

4. claude-3-5-sonnet-20241022: The experiment was conducted on claude-3-5-sonnet-20241022 using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation claude 3.5 sonnet rag](#).

Query-1: Which author has the highest number of followers?

TABLE 23: Ground Truth Reference VS Model claude-3-5-sonnet-20241022 output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	Based on the MongoDB response, I can tell you that the author with the highest number of followers is \"haricharan_1133\" who has three followers: yugeshkaran01@gmail.com, test@dsuniversity.ac.in, and yugeshkaran001@gmail.com. This indicates that haricharan_1133 has built a small but engaged following on the platform, with three distinct users actively following their content.

Accuracy: 0.85/1.0, Correctness: 1/1, Error rate: 0, Latency: 7.59s

5. claude-3-5-haiku-20241022: The experiment was conducted on claude-3-5-haiku-20241022 using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation claude 3.5 haiku rag](#).

Query-1: Which author has the highest number of followers?

TABLE 24: Ground Truth Reference VS Model claude-3-5-haiku-20241022 output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	Based on the MongoDB query and response, I can provide a detailed answer about the author with the highest number of followers.\n\nThe author with the

	highest number of followers is Haricharan, with the username 'haricharan_1133'. This author currently has three followers:\n1. yugeshkaran01@gmail.com\n2. test@dsuniversity.ac.in\n3. yugeshkaran001@gmail.com\n\nIn the context of the current schema, Yugesh Karan (the primary author in this dataset) only has one follower (test@dsuniversity.ac.in), which confirms that Haricharan has more followers.\n\nIt's worth noting that while Haricharan has the most followers in this query result, the specific posts or content by this author are not visible in the current schema. The response simply highlights the follower count and the author's name.\n\nFor users interested in exploring more about this author or their content, I would recommend conducting additional queries to retrieve more detailed information about Haricharan's posts or profile.
--	---

Accuracy: 0.83/1.0, Correctness: 1/1, Error rate: 0, Latency: 13.42s

6.Meta-llama/llama-4-maverick-17b-128e-instruct: The experiment was conducted on meta-llama/llama-4-maverick-17b-128e-instruct using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation-llama-4-maverick-rag](#).

Query 1: Which author has the highest number of followers?

TABLE 25: Ground Truth Reference VS Model llama-4-maverick output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	The author with the highest number of followers is Haricharan_1133, with a total of 3 followers, including yugeshkaran01@gmail.com, test@dsuniversity.ac.in, and yugeshkaran001@gmail.com. To confirm the existence of the author's profile and their associated content, you can visit the blog website and navigate to the author's profile. There, you can find their blog posts, profile information, and other relevant details.

Accuracy: 0.87/1.0, Correctness: 1/1, Error rate: 0, Latency: 1264.31s

7.meta-llama/Llama-4-Scout-17B-16E-Instruct: The experiment was conducted on meta-llama/Llama-4-Scout-17B-16E-Instruct using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation llama 4 scout rag](#).

Query 1: Which author has the highest number of followers?

TABLE 26: Ground Truth Reference VS Model llama-4-scout output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	Based on the search query, we found that the author with the highest number of followers is \"haricharan_1133\". This author has followers including 'yugeshkaran01@gmail.com', 'test@dsuniversity.ac.in', and 'yugeshkaran001@gmail.com'.\n\nHowever, I couldn't find any posts directly associated with this

	author in the provided schema. But I can confirm that there are posts from other authors, including Yugesh Karan, who has posted content in various categories such as Data Science, GenAI, and Education. If you're interested in finding posts by "haricharan_1133" or learning more about their content, I suggest trying an alternative search or providing more specific details. We can also try searching for posts by this author or within a particular category. Please let me know how I can assist you further. In general, our database does contain posts from various authors, and some authors have received engagement on their content. For instance, Yugesh Karan has posted content in multiple categories and has received views and likes on some of his posts. If you're interested in exploring more, I can provide information on his posts or help with another query.
--	---

Accuracy: 0.72/1.0, Correctness: 1/1, Error rate: 0, Latency: 25.24s

8. llama-3.1-8b-instant: The experiment was conducted on llama-3.1-8b-instant using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation llama 3.1 8b rag](#).

Query 1: Which author has the highest number of followers?

TABLE 27: Ground Truth Reference VS Model llama-3.1-8b output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	Based on the conversation history and MongoDB response, here's a natural language response with pre-sized content:\n\n\"Hello, I'd be happy to help you with your query. You asked which author has the highest number of followers. After searching through our database, I found that the author with the highest number of followers is Haricharan_1133. Haricharan_1133 has 3 followers including Yugesh Karan, and to confirm that the author exists, I found a post made by Haricharan_1133 on December 25, 2024, at 13:00:24, titled 'Striking a Balance: Smart Work vs. Hard Work'. Here's a brief description of the post: \n\n\"In the world of work, two philosophies often come to the forefront: smart work and hard work. Hard work involves putting in long hours, grinding away, and persevering through challenges. Smart work, on the other hand, is about leveraging one's skills, resources, and efficiency to achieve the same or better results with less

	effort\".\n\nHowever, I couldn't find any posts from Haricharan_1133 in the Data Science or GenAI categories. If you'd like to see more posts from this author, you can check out their other work on our blog.\n
--	--

Accuracy: 0.80/1.0, Correctness: 1/1, Error rate: 0, Latency: 118.16s

9. llama3-70b-8192: The experiment was conducted on llama3-70b-8192 using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation llama3 70b rag](#).

Query-1: Which author has the highest number of followers?

TABLE 28: Ground Truth Reference VS Model llama-3.1-8b output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	**Author with the Highest Number of Followers Found**\n\nWe have found the author with the highest number of followers on our blog. The author's name is Yugesh Karan, and they have a total of 3 followers.

Accuracy: 0.76/1.0, Correctness: 1/1, Error rate: 0, Latency: 122.79s

10. gemini-1.5-flash: The experiment was conducted on gemini-1.5-flash using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation gemini 1.5 flash rag](#).

Query-1: Which author has the highest number of followers?

TABLE 29: Ground Truth Reference VS Model gemini-1.5-flash output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	Yugesh Karan is the author with the most posts, having contributed 17 posts to the blog. However, the author with the highest number of followers is haricharan_1133, with a total of 3 followers. Unfortunately, the provided database schema does not contain any posts authored by haricharan_1133, so I cannot provide the content of their posts. To view the posts by haricharan_1133, additional information or a different query might be needed.\n

Accuracy: 0.81/1.0, Correctness: 1/1, Error rate: 0, Latency: 3.27s

11. gemini-1.5-flash-8b: The experiment was conducted on gemini-1.5-flash-8b using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation gemini 1.5 flash 8b rag](#).

query-1: Which author has the highest number of followers?

TABLE 30: Ground Truth Reference VS Model gemini-1.5-flash-8b output for query 1

Ground Truth Reference	Model Output
------------------------	--------------

The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	Yugesh Karan is not the author with the highest number of followers. Haricharan_1133 has the most followers, with a list including yugeshkaran01@gmail.com, test@dsuniversity.ac.in, and yugeshkaran001@gmail.com.\n
--	--

Accuracy: 0.77/1.0, Correctness: 1/1, Error rate: 0, Latency: 2.29s

12. gemma-3-27b-it: The experiment was conducted on gemma-3-27b-it using 10 test questions. Detailed information for 1 of the queries is provided below, along with their corresponding metric measurements. To view the full validation, click on [evaluation gemma 3 27b it rag](#).

Query-1: Which author has the highest number of followers?

TABLE 31: Ground Truth Reference VS Model gemma-3-27b-it output for query 1

Ground Truth Reference	Model Output
The author with the highest number of followers is haricharan_1133, with a total of 3 followers.	null

Accuracy: 0.0/1.0, Correctness: 0/1, Error rate: 1, Latency: 0.90s

Models Benchmark for MongoDB RAG system:

The performance of each model was evaluated, and the metric details for the MongoDB RAG system are presented in the figures below:

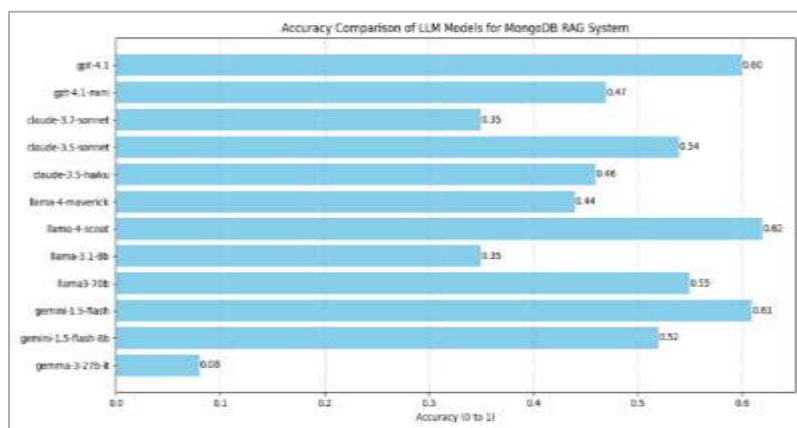


Fig. 10 Accuracy Comparison of LLM models of MongoDB RAG system.

From the MongoDB RAG system experiment shown in Figure 10, the overall accuracy was measured, and it was observed that meta-llama/llama-4-scout, gemini-1.5-flash, and gpt-4o-mini-2024-07-18 performed with higher accuracy, recording values of 0.62, 0.61, and 0.60 respectively, compared to the other models. In contrast, gemma-3-27-it recorded the lowest accuracy of 0.08/1.0 (i.e., 8%)

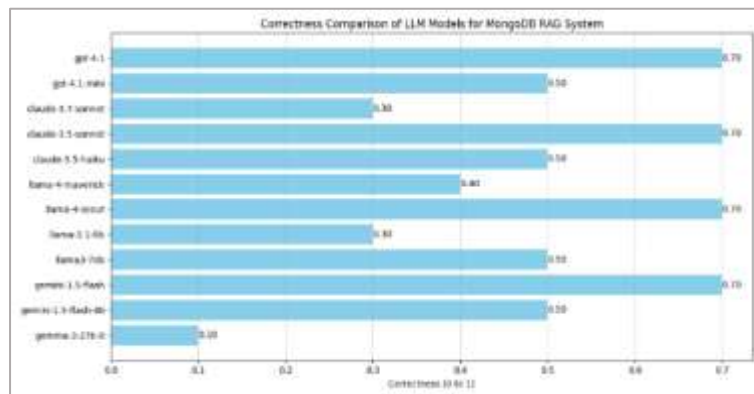


Fig. 11 Correctness Comparison of LLM models of MongoDB RAG system.

The correctness metric was evaluated for all models shown in figure 6, and it shows that the GPT-4.1 model, claude-3.5-sonnet, llama4-scout, gemini-1.5-flash achieved the highest correctness score of 0.70/1.0 (i.e., 70%), indicating that it provides highly reliable responses. On the other hand, the Gemma-3-27-it model recorded the lowest correctness score of 10%, making it the least accurate among all the models evaluated.

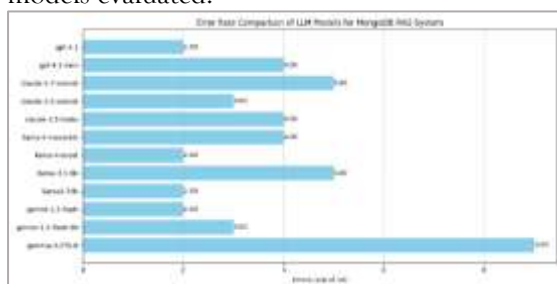


Fig. 12 Error Rate of LLM models of MongoDB RAG system.

It is observed from the experiment that out of the twelve models, four models gpt-4o-mini-2024-07-18, meta-llama/llama4-scout, llama3-70b-8192, and gemini-1.5-flash recorded an error rate of 20%, which is the lowest among all the models. The highest error rate was observed for gemma-3-27b-it, with a rate of 90%, as shown in Figure 12.

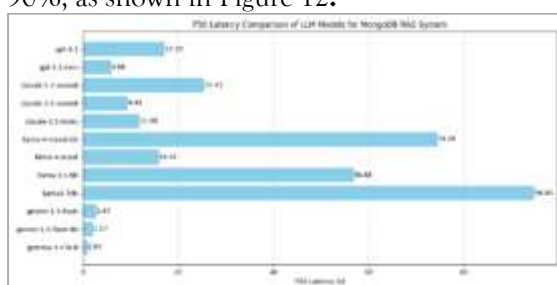


Fig. 13: P50 latency of LLM models of MySQL RAG system.

Based on the experimental data from Figure 13, the P50 latency was calculated for MongoDB RAG system using LangSmith. It was observed that gemini-1.5-flash, gemini-1.5-flash-8b, gemma-3-27b-it, gpt-4.1-mini, claude-3.5-sonnet took less than 10 seconds to execute half of the query requests, with durations of 2.87s, 2.17s, 0.94s, 5.98s and 9.42s respectively, compared to the other models.

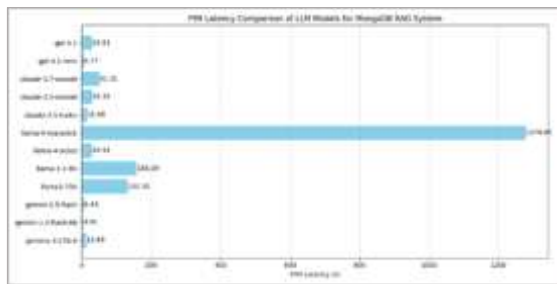


Fig. 14: P99 latency of LLM models of MongoDB RAG system

P99 latency of the MongoDB RAG system is shown in figure 14 which shows that gemini-1.5-flash, gemini-1.5-flash-8b, gemma-3-27b-it and gpt-4.1-mini are taken less than 15 seconds to complete all 10 queries response, in that still gemini-1.5-flash and gemma-3-27b-it are generate responses with errors which is show in figure 12.

Models Benchmark for MONGO DB RAG system :

TABLE 32: Model comparison based on overall metrics of each model

S.NO	LLM Model	Accuracy	Correctness	Error Rate	P50 Latency	P99 Latency
1	gpt-4.1	0.60 / 1.0	0.70 / 1.0	2 / 10	17.22s	29.53s
2	gpt-4.1-mini	0.47 / 1.0	0.50 / 1.0	4 / 10	5.98s	6.77s
3	claude-3-7-sonnet-20250219	0.35 / 1.0	0.30 / 1.0	5 / 10	25.43s	51.31s
4	claude-3-5-sonnet-20241022	0.54 / 1.0	0.70 / 1.0	3 / 10	9.42s	29.10s
5	claude-3-5-haiku-20241022	0.46 / 1.0	0.50 / 1.0	4 / 10	11.88s	15.48s
6	meta-llama/llama-4-maverick-17b-128e-instruct	0.44 / 1.0	0.40 / 1.0	4 / 10	74.48	1278.85S
7	meta-llama/Llama-4-Scout-17B-16E-Instruct	0.62 / 1.0	0.70 / 1.0	2 / 10	16.01s	29.54s
8	llama-3.1-8b-instant	0.35 / 1.0	0.30 / 1.0	5 / 10	56.88s	158.09s
9	llama3-70b-8192	0.55 / 1.0	0.50 / 1.0	2 / 10	94.80s	132.30s
10	gemini-1.5-flash	0.61 / 1.0	0.70 / 1.0	2 / 10	2.87s	6.43s
11	gemini-1.5-flash-8b	0.52 / 1.0	0.50 / 1.0	3 / 10	2.17s	4.41s
12	gemma-3-27b-it	0.08 / 1.0	0.10 / 1.0	9 / 10	0.94s	13.69s

CONCLUSION

The evaluation was conducted on both the MySQL and MongoDB RAG systems using the twelve selected LLM models listed in Table 1. The assessment was based on key performance metrics including accuracy, correctness, error rate, P50 latency, and P99 latency. In the MySQL RAG system, it was observed that **gpt-4.1** and **gpt-4.1-mini** demonstrated the best overall performance compared to the other models, as shown in Table 40. Specifically, **gpt-4.1** achieved an accuracy of 85%, correctness of 100%, an error rate of 0%, and a latency of 17.54 seconds to complete all queries. Similarly, **gpt-4.1-mini** recorded an accuracy of 85%, correctness of 90%, an error rate of 0%, and a latency of 20.81 seconds. Although the Gemini models exhibited better latency performance than other models including GPT-4 variants they performed poorly in terms of accuracy, correctness, and error rate, as detailed in **Table 40**.

In the MongoDB RAG system, the performance of the LLM models varied significantly compared to the MySQL RAG system. It was observed that **meta-llama/llama-4-scout** and **gemini-1.5-flash** performed better than the other models, as shown in **Table 77**. **meta-llama/llama-4-scout** achieved an accuracy of 62%, correctness of 70%, an error rate of 20%, and a latency of 29.54 seconds. Meanwhile, **gemini-1.5-flash** recorded an accuracy of 61%, correctness of 70%, an error rate of 20%, and a latency of just 6.43 seconds. Notably, in terms of latency, **gemini-1.5-flash** is approximately four times faster than **meta-llama/llama-4-scout**.

In this research, it was also observed that LLM models performed better in the MySQL RAG system compared to the MongoDB RAG system. This is primarily because LLMs face challenges in generating document-based queries due to the unstructured nature of NoSQL data. In the future, integrating both SQL and NoSQL databases with a real-time **vector database** could help overcome the limitations in the query generation layer of the proposed system, leading to improved performance and faster query retrieval.

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to Dhanalakshmi Srinivasan University for their support throughout this research. We are especially thankful to our guide, Dr. N. Shanmugapriya, Professor and Head of the Department, School of Engineering and Technology, for her unwavering guidance and valuable insights that greatly contributed to the success of this study.

REFERENCES

- [1] Xuanhe Zhou, Zhaoyan Sun, Guoliang Li, DBGPT: Large Language Model Meets Database. Published online: 19 January 2024 Data Science and Engineering (2024) 9:102–111 DOI:<https://doi.org/10.1007/s41019-023-00235-6>
- [2] Xiang Zhang, Khatoon Khedri, and Reza Rawassizadeh Can LLMs substitute SQL? Comparing Resource Utilization of Querying LLMs versus Traditional Relational Databases Published online: 12 April 2024 DOI: <https://doi.org/10.48550/arXiv.2404.08727>
- [3] Zijin Hong, Zheng Yuan, Qinggang Zhang, Hao Chen, Junnan Dong, Feiran Huang, and Xiao Huang Next-Generation Database Interfaces: A Survey of LLM-based Text-to-SQL Published Online:12 January 2024 DOI: <https://doi.org/10.48550/arXiv.2406.08426>
- [4] Johan Sandell, Einar Asplund, Workneh Yilma Ayele, Martin Duneld. Performance Comparison Analysis of ArangoDB, MySQL, and Neo4j: An Experimental Study of Querying Connected Data Published online: 2024-01-03 DOI:<https://hdl.handle.net/10125/107319>
- [5] M. Hindi, L. Mohammed, O. Maaz and A. Alwarafy, "Enhancing the Precision and Interpretability of Retrieval-Augmented Generation (RAG) in Legal Technology: A Survey," in IEEE Access, vol. 13, pp. 46171-46189, 2025, doi: 10.1109/ACCESS.2025.3550145
- [6] B. Saha, U. Saha and M. Zubair Malik, "QuIM-RAG: Advancing Retrieval-Augmented Generation With Inverted Question Matching for Enhanced QA Performance," in IEEE Access, vol. 12, pp. 185401-185410, 2024, DOI:10.1109/ACCESS.2024.3513155.
- [7] R. Hu, S. Liu, P. Qi, J. Liu and F. Li, "ICCA-RAG: Intelligent Customs Clearance Assistant Using Retrieval-Augmented Generation (RAG)," in IEEE Access, vol. 13, pp. 39711-39726, 2025, DOI: 10.1109/ACCESS.2025.3544408.
- [8] L.-B. Hernandez-Salinas et al., "IDAS: Intelligent Driving Assistance System Using RAG," in IEEE Open Journal of Vehicular Technology, vol. 5, pp. 1139-1165, 2024, doi: 10.1109/OJVT.2024.3447449.
- [9] N. Alabbasi, O. Erak, O. Alhussein, I. Lotfi, S. Muhaidat and M. Debbah, "TeleOracle: Fine-Tuned Retrieval-Augmented Generation With Long-Context Support for Networks," in IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2025.3553161.

- [10] G. Budakoglu and H. Emekci, "Unveiling the Power of Large Language Models: A Comparative Study of Retrieval-Augmented Generation, Fine-Tuning, and Their Synergistic Fusion for Enhanced Performance," in *IEEE Access*, vol. 13, pp. 30936-30951, 2025, DOI:10.1109/ACCESS.2025.3542334.
- [11] Y. Setyawan Soekanto, L. Christopher Limanjaya, Y. Kaleb Purwanto and D. -K. Kang, "From Queries to Courses: SKYRAG's Revolution in Learning Path Generation via Keyword-Based Document Retrieval," in *IEEE Access*, vol. 13, pp. 21434-21455, 2025, DOI:10.1109/ACCESS.2025.3535618.
- [12] M. Fateen, B. Wang and T. Mine, "Beyond Scores: A Modular RAG-Based System for Automatic Short Answer Scoring With Feedback," in *IEEE Access*, vol. 12, pp. 185371-185385, 2024, doi: 10.1109/ACCESS.2024.3508747
- [13] J. Xu, H. Zhang, H. Zhang, J. Lu and G. Xiao, "ChatTf: A Knowledge Graph-Enhanced Intelligent Q&A System for Mitigating Factuality Hallucinations in Traditional Folklore," in *IEEE Access*, vol. 12, pp. 162638-162650, 2024, DOI:10.1109/ACCESS.2024.3485877.
- [14] M. Al-Qatf et al., "RAG4DS: Retrieval-Augmented Generation for Data Spaces—A Unified Lifecycle, Challenges, and Opportunities," in *IEEE Access*, vol. 13, pp. 39510-39522, 2025, DOI: 10.1109/ACCESS.2025.3545387.
- [15] A. T. Neumann, Y. Yin, S. Sowe, S. Decker and M. Jarke, "An LLM-Driven Chatbot in Higher Education for Databases and Information Systems," in *IEEE Transactions on Education*, vol. 68, no. 1, pp. 103-116, Feb. 2025, DOI:10.1109/TE.2024.3467912.
- [16] S. Ge, Y. Sun, Y. Cui and D. Wei, "An Innovative Solution to Design Problems: Applying the Chain-of-Thought Technique to Integrate LLM-Based Agents With Concept Generation Methods," in *IEEE Access*, vol. 13, pp. 10499-10512, 2025, doi: 10.1109/ACCESS.2024.3494054.
- [17] Z. Keskin et al., "LLM-Enhanced Human-Machine Interaction for Adaptive Decision-Making in Dynamic Manufacturing Process Environments," in *IEEE Access*, vol. 13, pp. 44650-44661, 2025, doi: 10.1109/ACCESS.2025.3549529.
- [18] A. Mestre, R. Marques, A. Fernandes and B. Silva, "RAGNAR: Retrieval-Augmented Generation using Networked and Advanced Relational Data," 2024 8th International Symposium on Innovative Approaches in Smart Technologies (ISAS), İstanbul, Türkiye, 2024, pp. 1-6, doi: 10.1109/ISAS64331.2024.10845598.
- [19] S. Vichev and A. Marchev, "RAGSQL: Context Retrieval Evaluation on Augmenting Text-to-SQL Prompts," 2024 IEEE 12th International Conference on Intelligent Systems (IS), Varna, Bulgaria, 2024, pp. 1-6, doi: 10.1109/IS61756.2024.10705186.
- [20] M. M. Sanjeev, B. Ramalingam and S. Kumar T.K., "Realtime Semantic Similarity Analysis of Bulk Outlook Emails Using BERT," 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), Dehradun, India, 2020, pp. 89-94, doi: 10.1109/ICACCM50413.2020.9212979.