Galava : A Scalable Search Engine for Organizational Database Pools using Large Language Models

Abstract

The widespread adoption of big data and varied database technologies has significantly enhanced organizational capabilities for managing large volumes of information. However, this progress also introduces challenges in data retrieval due to data fragmentation across multiple systems, the need for specialized query languages for different databases, and scalability issues. These barriers impede a unified view of data, restrict access to users without technical expertise, and slow down the decision-making process due to prolonged data querying times.

In response, we propose Galava, a search engine specifically designed to address these challenges. It integrates seamlessly with both SQL and NoSQL systems, offering a unified, scalable interface that enhances data accessibility and retrieval efficiency. Galava enables swift, intuitive access to a comprehensive data spectrum, improving organizational agility and empowering a broader range of stakeholders to engage in data-driven decision-making without requiring in-depth technical knowledge.

Acknowledgment

We would like to express our profound gratitude to our project mentor, Dr. Hanumat Sastry G, for his invaluable guidance and unwavering support throughout our project. Dr. Sastry's deep expertise in data science and large language models has been instrumental in developing this innovative project. His keen insights, detailed feedback, and encouragement played a pivotal role in navigating the complexities of integrating diverse database systems and applying advanced natural language processing techniques.

We are immensely thankful for his mentorship, which has enriched our understanding of the technical and strategic aspects of building scalable search engines tailored to complex organizational data environments. The skills in analytical thinking, problem-solving, and interdisciplinary collaboration that we have honed under his guidance are invaluable and will undoubtedly influence our future endeavors in the field of computer science and artificial intelligence.

Contents

1 Introduction **3**

2 Problem Statement **4**

3 Literature Review **4**

4 Objectives **5**

5 Methodology **6**

6 Results **11**

7 Conclusion **12**

8 Future Work **13**

9 References **13**

List of Figures

1 Configuration file design 6
2 Entity Metadata design 7
3 High-level representation of Knowledge Graph 8
4 Hierarchical Navigable Small World (HNSW) 9
5 UML Class diagram 10
6 Authentication page 11
7 Query Page 12
8 Result 12

1 Introduction

In the era of information, data has become the lifeblood of organizations, driving decision-making, innovation, and strategic planning across various sectors. The ability to efficiently store, manage, and retrieve data is paramount to leveraging its full potential. Over the years, advancements in database technologies have led to the development of diverse systems tailored to specific types of data and use cases, ranging from traditional relational databases to more flexible NoSQL databases designed to handle unstructured data. While these technological advancements have significantly enhanced data storage and management capabilities, they have also introduced a new set of challenges in data retrieval, particularly in organizational environments characterized by a multitude of disparate database systems.

The proliferation of diverse databases within organizations has led to data silos, where information is stored in separate, often incompatible systems. This fragmentation hinders the ability to perform comprehensive searches across the entire data landscape, limiting the visibility and accessibility of valuable information. Moreover, the complexity of query languages specific to each database system necessitates a high degree of technical expertise, restricting data access to a select group of individuals within the organization.

Furthermore, as organizations continue to grow and their data ecosystems expand, scalability becomes a critical concern. Traditional data retrieval methods struggle to keep pace with the increasing volume of data and the demand for real-time access, leading to delays and inefficiencies that can hamper operational agility and responsiveness. Recognizing these challenges, the need for a unified, efficient, and user-friendly solution for navigating the complex web of organizational databases has become more apparent than ever. Such a solution would not only break down data silos and provide a holistic view of the data assets but also simplify the data retrieval process, making it accessible to a broader range of users regardless of their technical expertise.

In response to this need, we introduce Galava, a novel search engine designed to bridge the gap between diverse database systems and the users seeking to access their data. Galava stands out by offering a low-latency, scalable, and intuitive interface that allows for seamless navigation across a plethora of databases, effectively democratizing data access within organizations. By leveraging advanced natural language processing (NLP) techniques and knowledge graphs, Galava aims to transform the data retrieval landscape, making it more efficient, inclusive, and agile.

This project outlines the development of Galava, detailing its innovative approach to overcoming the challenges of data fragmentation, complexity, and scalability in organizational database environments. Through a comprehensive examination of its architecture, functionality, and potential impact, we aim to contribute to the ongoing discourse on data management and retrieval solutions, offering insights and perspectives that could shape the future of organizational data strategies.

2 Problem Statement

Organizations today face significant challenges in data retrieval due to the fragmentation of information across various database systems, the complexity of query languages that require specialized technical skills, and the scalability issues associated with handling large volumes of data across multiple databases. This situation creates barriers to efficient and timely access to data, limiting the ability of organizations to make informed decisions and leverage their full data potential. There is a critical need for a solution that simplifies and unifies data access, enabling users to retrieve information quickly and efficiently from diverse databases without requiring extensive technical expertise, and that can scale effectively to meet growing data demands.

Thus, the core problem this project aims to address is the development and introduction of a scalable search interface facilitated by language modeling to overcome the limitations in current approaches to data retrieval practices in distributed pools of databases. Leveraging natural language processing and intelligent query optimization, it promises to transform data retrieval practices, ensuring comprehensive, efficient, and user-friendly access to data assets, thereby enhancing decision-making processes and operational efficiency across multiple sectors.

3 Literature Review

References	Findings	Applications
1. Bazaga2021	Demonstrated the capability to predict ranked answers to natural language questions over SQL database tables by embedding both the question and the table into distributed representations.	Develop more intuitive and efficient natural language interfaces for SQL databases, particularly in domains where accuracy and the ability to interpret complex queries are critical, such as in business intelligence tools, customer support databases, and academic research databases.

References	Findings	Applications
2. Dar2019	Revealed a strong inclination towards English language support in these frameworks, and categorization of SQLbased frameworks into statistical, symbolic, and connectionist approaches, providing a comprehensive overview of the landscape of natural language database querying frameworks and their methodological diversities.	Guide the development of next-generation database querying systems that are more inclusive of various database types and languages. It can also inform the expansion of natural language processing capabilities to accommodate linguistic variations, making database querying more accessible to non-technical users across different domains.
3. Papenmeier2021	The study found that chatbotlike interfaces could elicit significantly longer and more detailed natural language queries from users, with a greater number of key facts mentioned. This indicates the potential of conversational interfaces to enhance the richness of user queries, which could lead to more accurate and tailored search results.	Designing user interfaces for search engines and databases that encourage more natural and detailed user queries. This could be particularly beneficial in e-commerce search engines, online libraries, and other digital platforms where users seek specific information but may not know the exact terms to use in their search queries. .

4 Objectives

- Design a scalable search engine that seamlessly interfaces with distributed pools of databases, ensuring consistent performance and adaptability across varying data volumes and user demands

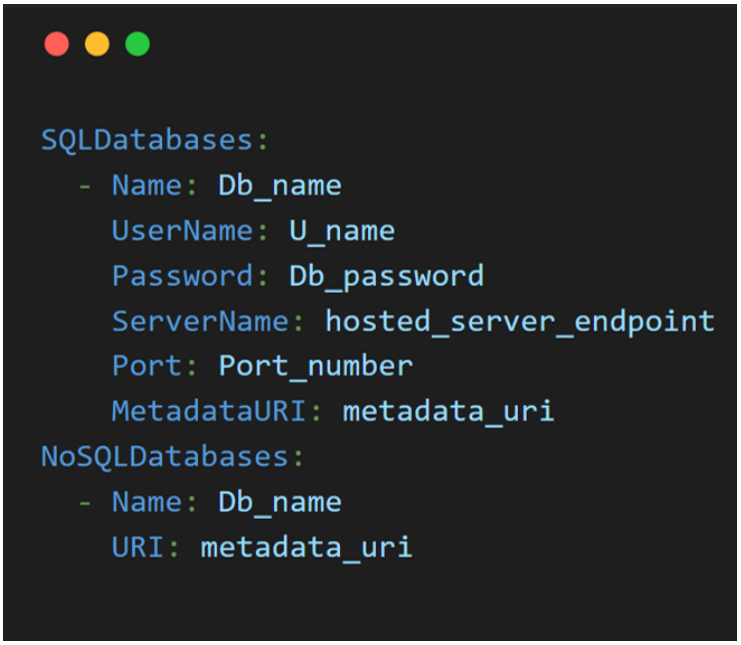
- Leverage fast and efficient HNSW (Hierarchical Navigable Small World) indexing algorithms to maintain an optimal balance between accuracy and speed, enabling rapid and precise data retrieval in large-scale database environments

- Employ a write-ahead caching mechanism to enhance retrieval speeds, allowing for the immediate availability of frequently accessed data and thereby improving the overall efficiency of the search process

- Integrate advanced Natural Language Processing (NLP) techniques to interpret and process user queries, facilitating an intuitive search experience that allows users to engage with the system using everyday language, thereby lowering the technical barrier to data access

5 Methodology

1. **Vault Design:**
   - **Properties Definition:**
     - **Secrets:**
       - The vault securely stores a secret key containing the URI to an object storage location. This object storage holds a crucial YAML configuration file that Galava utilizes to establish connections. Galava is granted read-only access to this object storage, ensuring it can securely retrieve the necessary configuration without compromising the integrity or security of the underlying databases.
     - **Configuration YAML File:**
       - Galava relies on a configuration file, retrieved from the designated object storage bucket as specified by the vault's URI. This YAML file is meticulously structured to include all necessary credentials, connection parameters, and configurations required for the application to interface with the various databases within the organizational ecosystem. See in 1.

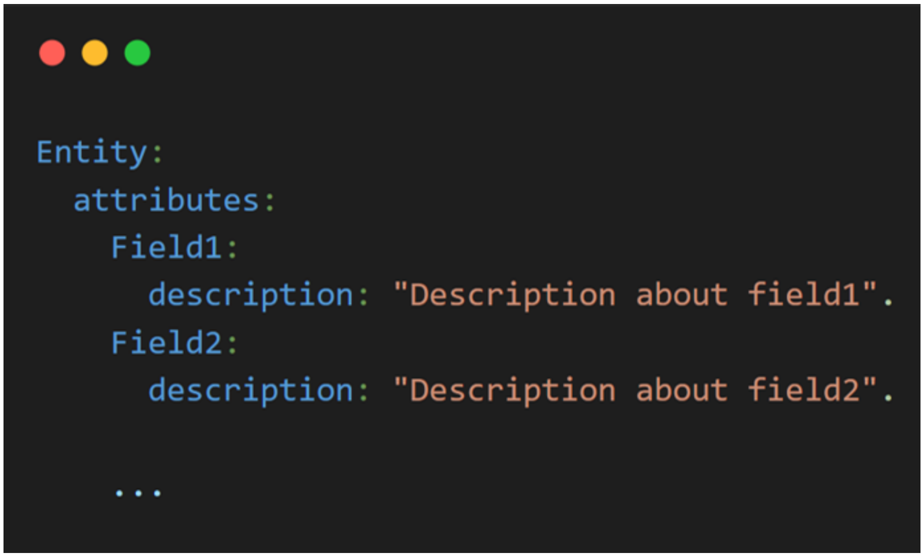


```yaml
SQLDatabases:
  - Name: Db_name
    UserName: U_name
    Password: Db_password
    ServerName: hosted_server_endpoint
    Port: Port_number
    MetadataURI: metadata_uri
NoSQLDatabases:
  - Name: Db_name
    URI: metadata_uri
```

Figure 1: Configuration file design

2. **Behavior Definition:**

- **Metadata:**
  - Each entity, whether a table in an SQL database or a document in a NoSQL database, is associated with metadata that encapsulates its characteristics, semantics, and additional relevant information. This metadata, crucial for understanding the structure and context of the data, is stored in an object storage location specified by the ”Metadata URI” in the configuration file. See Figure 2.



```
Entity:
  attributes:
    Field1:
      description: "Description about field1".
    Field2:
      description: "Description about field2".

    ...
```

Figure 2: Entity Metadata design

- **Knowledge Graph Structure:**
  - Each entity node in the knowledge graph will have an entity name.
  - Entity nodes are grouped by a defined relationship with corresponding database nodes.
  - These nodes will contain an attribute that stores its corresponding vector. See Figure 3.

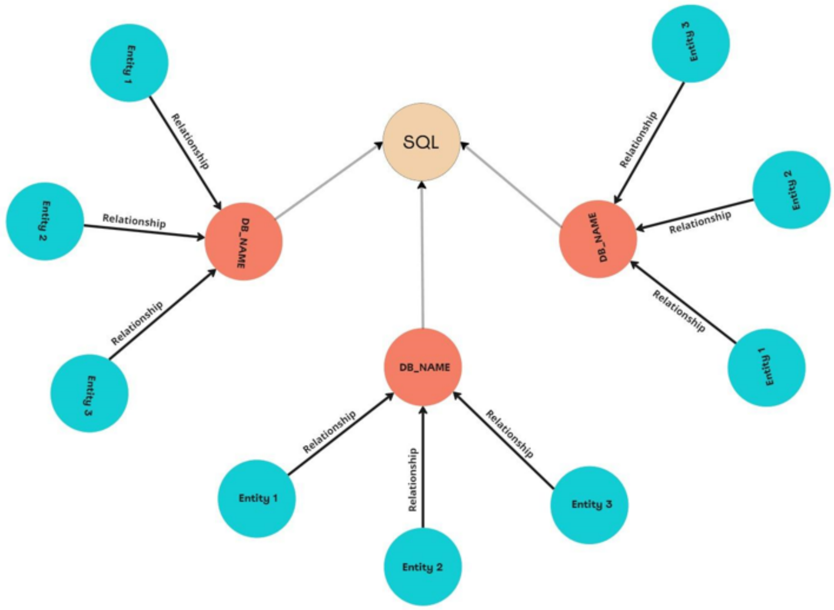


Figure 3: High-level representation of Knowledge Graph

3. **Vectorization:**
   - **Embeddings:**
     - Each entity is pre-computed and stored as a vector embedding, which represents its semantic structure having a fixed dimension.
4. **Approximate Nearest Neighbour Search:**
   - **Indexing Algorithm:**
     - The Hierarchical Navigable Small Worlds (HNSW) provides an efficient ANN search algorithm to identify relevant nodes for the querying vector. See Figure 4.

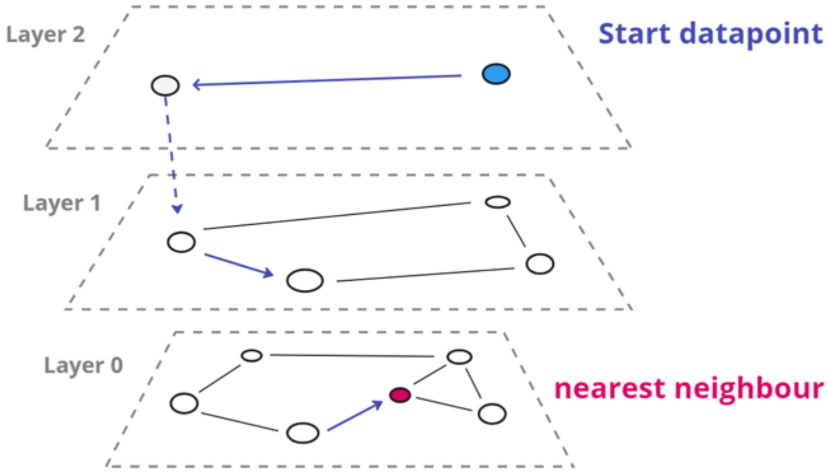


Figure 4: Hierarchical Navigable Small World (HNSW)

ALGORITHM

Input: User query **q**, Vault URI **v_uri**, Configuration file **f1**, Metadata URI **m_uri**

Output: Query results for **q**

1. Retrieve the Vault URI **v_uri** to access the secure storage location.
2. Load the configuration file **f1** from object storage specified by **v_uri**.
3. For each database configuration in **f1**:
    - (a) Establish a connection to the database using its specific connection parameters.
    - (b) Fetch the metadata from **m_uri** associated with the database.
    - (c) Vectorize the fetched metadata to create a numerical representation for indexing.
    - (d) Add a node for each vectorized metadata entity to the Knowledge Graph.
4. Vectorize the user query **q** to transform it into a numerical vector **q_vec**.
5. Utilize the HNSW (Hierarchical Navigable Small World) algorithm to identify the nearest nodes in the Knowledge Graph to the vectorized query **q_vec**.

6. Add the identified nearest nodes to a processing queue **Q**.
7. For each node in **Q**:
   (a) Extract the table details and relevant query information.
   (b) Formulate an equivalent database query using an LLM.
   (c) Execute the formulated query against the respective database.
   (d) Collect the query results.
8. Aggregate the collected query results from all processed nodes.

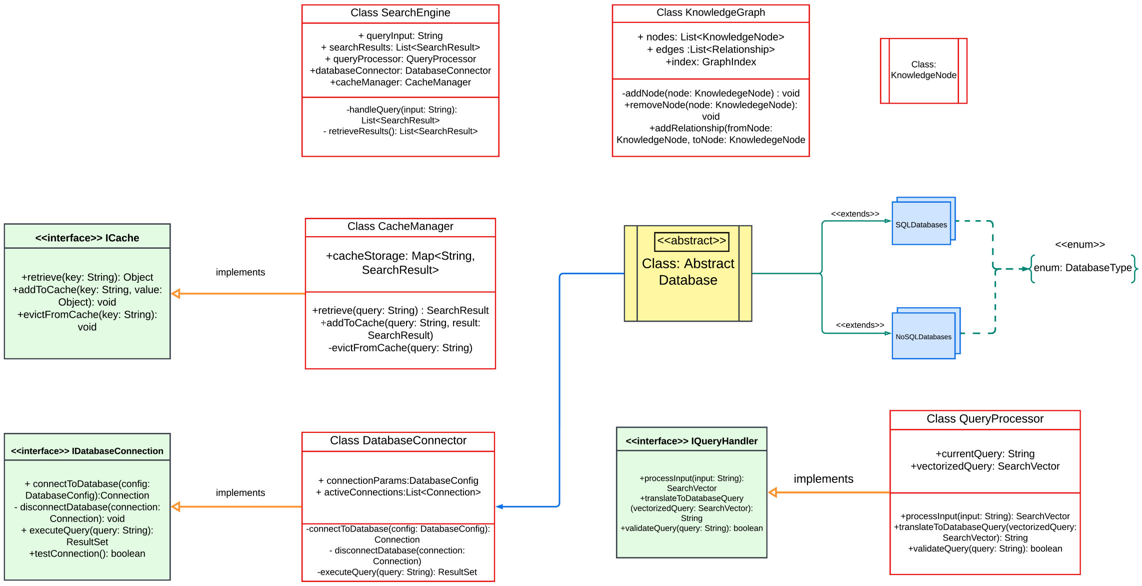

Class SearchEngine
+ queryInput: String
+ searchResults: List<SearchResult>
+ queryProcessor: QueryProcessor
+databaseConnector: DatabaseConnector
+cacheManager: CacheManager
-handleQuery(input: String): List<SearchResult>
- retrieveResults(): List<SearchResult>
Class KnowledgeGraph
+ nodes: List<KnowledgeNode>
+ edges :List<Relationship>
+index: GraphIndex
-addNode(node: KnowledegeNode) : void
+removeNode(node: KnowledegeNode): void
+addRelationship(fromNode: KnowledgeNode, toNode: KnowledegeNode
Class: KnowledgeNode
<<interface>> ICache
+retrieve(key: String): Object
+addToCache(key: String, value: Object): void
+evictFromCache(key: String): void
implements
Class CacheManager
+cacheStorage: Map<String, SearchResult>
+retrieve(query: String) : SearchResult
+addToCache(query: String, result: SearchResult)
-evictFromCache(query: String)
<<abstract>>
Class: Abstract Database
<<extends>>
SQLDatabases
<<extends>>
NoSQLDatabases
<<enum>>
enum: DatabaseType
<<interface>> IDatabaseConnection
+ connectToDatabase(config: DatabaseConfig):Connection
- disconnectDatabase(connection: Connection): void
+ executeQuery(query: String): ResultSet
+testConnection(): boolean
implements
Class DatabaseConnector
+ connectionParams:DatabaseConfig
+ activeConnections:List<Connection>
-connectToDatabase(config: DatabaseConfig): Connection
- disconnectDatabase(connection: Connection)
-executeQuery(query: String): ResultSet
<<interface>> IQueryHandler
+processInput(input: String): SearchVector
+translateToDatabaseQuery (vectorizedQuery: SearchVector): String
+validateQuery(query: String): boolean
implements
Class QueryProcessor
+currentQuery: String
+vectorizedQuery: SearchVector
+processInput(input: String): SearchVector
+translateToDatabaseQuery(vectorizedQuery: SearchVector): String
+validateQuery(query: String): boolean

Figure 5: UML Class diagram

6 Results

The implementation of Galava has yielded a robust and intuitive platform that significantly enhances the data retrieval capabilities across organizational database pools. Key results from the deployment of Galava are illustrated through its web interface, which encompasses an authentication page and a dynamic query page.

Authentication Page: The authentication mechanism is designed to ensure secure access to the Galava platform. It uses state-of-the-art security protocols to protect sensitive data and user credentials, thus upholding our commitment to data privacy and security. The authentication page serves as the gateway to accessing Galava's powerful search capabilities, reinforcing the secure framework within which organizational data interactions occur. See Figure 6

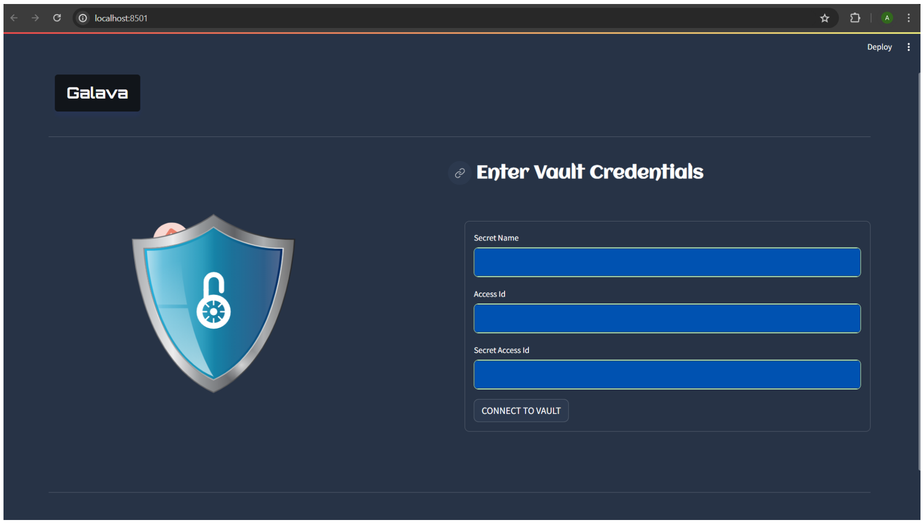
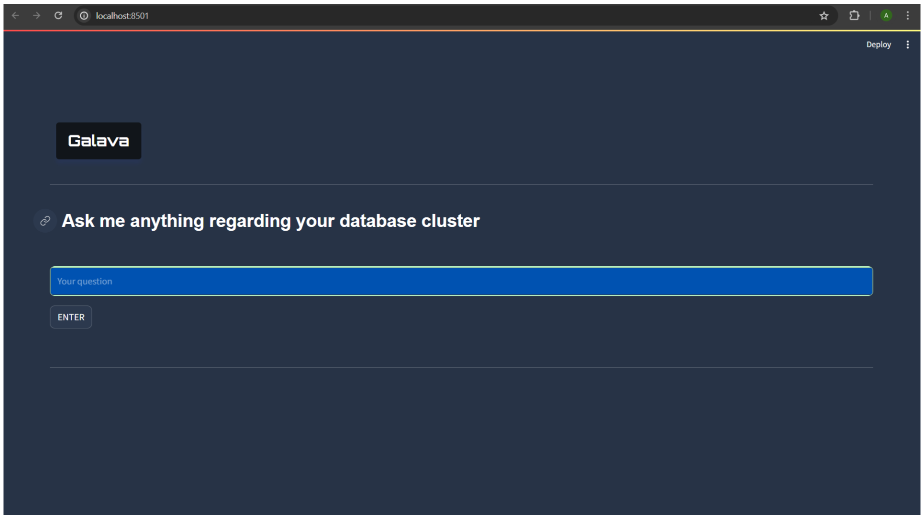


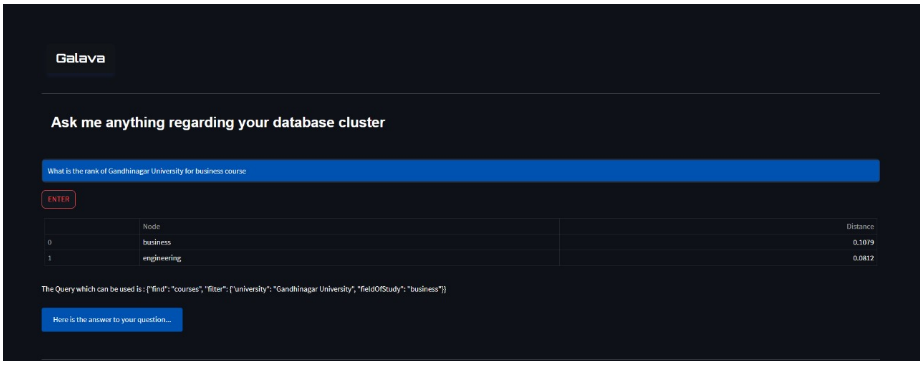
Figure 6: Authentication page

Query Page: The core functionality of Galava is showcased on the query page, where users can input natural language queries to retrieve data seamlessly from multiple databases. This page highlights the application of Large Language Models to interpret and process user queries, converting them into effective database-specific queries. For instance, users can ask, **"What is the rank of Gandhinagar University for business course?"** and Galava will fetch relevant data across connected databases, showcasing its capability to handle complex queries with ease. Figure 8 illustrates the query processing.



Ask me anything regarding your database cluster

Figure 7: Query Page



Ask me anything regarding your database cluster

What is the rank of Gandhinagar University for business course

	Node	Distance
0	business	0.1079
1	engineering	0.0812

The Query which can be used is : {"find": "courses", "filter": {"university": "Gandhinagar University", "fieldOfStudy": "business"}}

Here is the answer to your question...

Figure 8: Result

7 Conclusion

In conclusion, the Galava project represents a significant stride forward in addressing the prevailing challenges of data retrieval in organizational database environments. Through the innovative integration of Large Language Models (LLMs), advanced indexing algorithms, and natural language processing techniques, Galava offers a novel solution that transcends traditional data retrieval boundaries. It simplifies the process, removing technical barriers, and enables efficient and intuitive access to a plethora of distributed data sources. The system's ability to interface with both SQL and NoSQL databases demonstrates its versatility and adaptability, proving to be an indispensable tool in an era where data is a pivotal asset for decision-making and strategic planning. By successfully overcoming data fragmentation and scalability issues, Galava empowers a wider range of organizational stakeholders, fostering a data-driven culture that is both inclusive and agile.

8 Future Work

Expansion of Language Support: Although Galava has made strides in querying databases through English natural language processing, the addition of multilingual support would make the system accessible to a broader global user base. This will involve training LLMs on diverse linguistic datasets to ensure accuracy and cultural relevance in query understanding and generation.
Improved Customization for Domain-Specific Queries: Customization of the system for different industry-specific databases could enhance the relevancy and precision of the search results. Future work could focus on tailoring LLMs to understand and process domain-specific terminology and contexts, further refining the search experience.
Development of a More User-Friendly Interface: While the current interface is functional, there is always room for improvement in user experience design. Future versions could include a more intuitive graphical interface, interactive data visualization tools, and personalized search options to cater to non-technical users' preferences.
Performance Optimization in Large-Scale Deployments: As deployment scales, performance optimization becomes crucial. Future iterations of Galava could explore distributed computing frameworks to manage load balancing, fault tolerance, and query optimization in large-scale environments.

9 References

1. Bazaga2021 Translating synthetic natural language to database queries with a polyglot deep learning framework.

2. Dar2019 Lali, M. I., Din, M. U., Malik, K. M., Bukhari, S. A. . Frameworks for Querying Databases Using Natural Language: A Literature Review.

3. Papenmeier2021 Starting Conversations with Search Engines – Interfaces that Elicit Natural Language Queries.