Volume: 12 Issue: VI Month of publication: June 2024
DOI: https://doi.org/10.22214/ijraset.2024.63199

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

MySQL based Chit Chat Application

Ritik Gupta[1], Dr. Rika Sharma[2]
Amity University

Abstract: This project provides a modern chat application designed to work well with MySQL databases. The app uses Streamlit on the front end, providing a user interface through which users can query and manage data in natural language. The combination of OpenAI's GPT framework and LangChain provides powerful natural language processing and conversational capabilities, making data interaction more intuitive and efficient. This project demonstrates the integration of modern AI capabilities with traditional information management systems. The software bridges the gap between simple searches and user experience by providing a natural language interface to the database. This demonstrates the potential of AI to simplify data analysis and management tasks and make these powerful tools accessible to more people.
Key Features: 1.Background Notes: The program enhances user experience with custom CSS to set background images both in the main app and in sidebars. This feature provides a visual interface that can be customized according to the user's preferences.
2. Database Connection: This simple connection is made possible by the init_database function, which creates the appropriate directory URI and initiates the connection.
3. Natural Language Processing and SQL Generation: The program uses OpenAI's GPT model to interpret user queries expressed in natural language. The system efficiently converts these queries into SQL statements using a predefined template. The LangChain library manages the content of the conversation and ensures that each SQL query is presented according to the schema and history of the conversation.
4. Call output: Internet chat simulates a conversation with a data analyst. Users enter their queries and the program responds with generated SQL queries and corresponding results in natural language. This interaction is designed to be natural and engaging; It encourages users to search and query the database effortlessly.
5. Features include: The application displays messages from the user and the AI in a dynamic, ongoing dialogue. Custom fonts differentiate user messages with different colors and backgrounds for AI solutions to improve readability and user experience

I. INTRODUCTION

Welcome to our latest interactive software designed to transform the way users interact with SQL databases. This project delivers a seamless user experience by leveraging the power of Streamlit for deep web analytics, OpenTI's GPT model for natural language processing, and LangChain for speech language management.
The main goal of this project is to create an easy-to-use, interactive application that allows users to interact with SQL databases using natural language queries. Using the latest AI technology and a powerful platform, the app aims to make database problems accessible to users with a wide range of technical skills.

A. *Important Facts*

1) *Custom Interface:* The app allows users to create custom background images, enhance visuals and personalization. The use of Base64 encoding ensures compatibility of images in different formats. . This functionality is made possible by the init_directory function, which creates the appropriate URI directory and initiates the connection.
2) *Natural Language Query:* Using OpenAI's GPT framework, the program interprets user queries expressed in natural language and converts them into SQL statements. The LangChain library ensures that each SQL query is executed based on the existing schema and conversation history, making data interaction more intuitive and efficient.
3) *Call Output:* The interface simulates a conversation with a data analyst. Users enter their queries and the program responds with generated SQL queries and corresponding results rendered in natural language. This interaction is designed to be natural and engaging; It encourages users to search and query the database effortlessly.
4) *Visualization:* Many CSS extensions allow you to design a variety of user interface elements, including background colors, text colors, and input fields. This flexibility allows the program to be tailored to your specific marketing needs or personal preferences.

B. Technology Stack

1) *Streamlit:* Used to create a user-friendly interface. Streamlit's simplicity and flexibility make it an excellent choice for rapid development and deployment of Internet networks.
2) *OpenAI GPT Model:* Provides natural language and rendering capabilities. These models identify user queries and effectively create SQL queries.
3) *LangChain:* Manages the content of conversations and ensures that each SQL query is relevant and accurate based on ongoing interactions. LangChain creates templates quickly and integrates with the template language.
4) *MySQL:* The database management system of choice for storing and retrieving data. MySQL's reliability and performance make it a good choice for a wide range of applications.
5) *dotenv:* Used to correctly load environment variables; ensures understanding of data when using a database document correctly.
6) *Custom CSS:* Enhances the appearance and user experience of applications with custom styles.

C. Details

1) *Initialization and Installation:* Once the program is installed, it sets a session state to maintain chat history and data connections. Custom CSS is injected into the background layout and various UI elements.
2) *Database Connection:* On the side, users enter their database information. When you click the "Connect" button, the program will try to establish a connection using the information provided. If successful, a success message will be displayed.
3) *Issue Handling:* When a user submits an issue, it is added to the discussion history. The program then processes the query using the get_response function; This process involves several steps:
   - ♣ Creating the SQL query using the language model based on the nature of the current dialog.
   - ♣ Run a SQL query against a linked database.
   - ♣ Prepare answers in natural language and express them in conversation.
4) **Show results:** The software separates user input from AI results using a special method. AI solutions provide transparency and clarity by containing the generated SQL queries and their results.
5) *Usage and User Experience:* The use of custom balls and messages enhances the overall user experience, making the app more interesting and fun to use.

II. LITERATURE SURVEY

Creating a Streamlit application for SQL queries involves many things like designing user interfaces, merging SQL data, and using natural language to describe user queries. The literature provides a comprehensive foundation for understanding the various aspects involved. There are studies in all of the books covering these areas: -

A. User Interface Design in Streamlit

Streamlit is a popular framework for quickly building data applications. It provides an easy way to create a web-based user interface using Python. Key resources on this topic are:
• Streamlit Documentation: Official documentation provides guidance and examples on how to use Streamlit's features.
• Community forums and tutorials: Streamlit has many tutorials and community discussions about UI design best practices on sites like Stack Overflow, Medium, and Towards Data Science.

B. SQL Databases SQL

Connecting to SQL databases in Python, SQLAlchemy, mysql-connector-python, etc. It can be achieved using various libraries such as.
• SQLAlchemy: This is a powerful ORM library that provides a complete package known for Python. persistence at the feature level. It includes a database connection and audio system to support multiple databases.

C. Natural Language Processing (NLP) for SQL Queries Question

The purpose of NLP in this section is to convert native language queries into SQL queries. Recent advances in machine learning and artificial intelligence have made this possible.
• Transformer models: Models such as GPT-3/4 (and OpenAI) have demonstrated impressive abilities to understand and render human-like text. These models can be easily adapted to specific tasks, such as converting natural expressions to SQL.

D. Data Extraction and Data Extraction Diagram

Understanding database architecture is crucial to executing SQL queries effectively.

• Downloading databases: The method of downloading database information may vary depending on the database used. For example, the INFORMATION_SCHEMA table can be used to retrieve metadata about a database in MySQL.

Literature: Description of the database system, Abraham Silberschatz, Henry Korth and S. Sudarshan.

E. Responsive Design

Making an application interactive and responsive improves the user experience. This includes implementing features such as updated, interactive features and user feedback.

• Web Development Principles: General principles of web development and design apply here. Techniques like CSS for rendering, JavaScript for interactivity, and AJAX for updates are rarely used.

III. IMPLEMENTATION STUDY

This project aims to develop an unstructured and interactive chat application that connects to a MySQL database. The software uses artificial intelligence to create SQL queries based on input and present the results in natural language. This app is built using Streamlit, a powerful and easy-to-use web application development tool in Python. Key features include the use of LangChain for fast management and SQL queries, and OpenAI's GPT () for natural language processing.

This app demonstrates the ability to combine modern AI technologies with user experience to create powerful and intuitive applications. Future developments could expand the database and increase natural language processing possibilities.

A. Objectives

1) Create an interactive interface that allows users to interact with SQL databases in natural language.
2) Create SQL queries based on user input and present the results in an easy-to-understand format.
3) Check whether store operations are safe.

B. Architecture

The architecture of the Chat program consists of the following main parts:

1) *Streamlit Interface:* The front-end of the application is built using Streamlit for user interaction.
2) *LangChain Commands:* Used to execute SQL queries based on user input.
3) *OpenAI GPT:* Responsible for processing and displaying the native language.
4) *MySQL Database:* Backend database where user queries are processed.

C. Details

1) *Streamlit Interface:* Streamlit provides an interactive interface where users can enter their questions. The interface is designed with custom CSS to enhance user experience, including background image and text styling.
2) *LangChain Prompts:* LangChain is used to create and manage prompts for creating SQL queries. ChatPromptTemplate is configured to interpret user input and provide the appropriate SQL expression.
3) *OpenAI GPT Integration:* OpenAI GPT is used to process the user's natural language input and deliver human-like results. It connects to LangChain and requests to retrieve the necessary information from the database.
4) *MySQL Database Connection:* The program connects to the MySQL database using user-supplied credentials. This allows users to interact with their store via chat.

IV. PROPOSED METHODOLOGY

A. Introduction

1) *Background:* Discuss the motivation behind creating Streamlit software that simplifies SQL interactions. Demonstrate the importance of using the database.
2) *Goals:* State the main goals of the project, such as creating an interactive user interface, integrating SQL databases, and providing natural language-based solutions to database problems.

B. Storage System

1) Summary

Provide a schematic diagram of the system.
Identify relevant components, including the frontend (Streamlit interface), backend (database interaction), and any middleware or APIs used.

2) Components

Streamlit Frontend
♣ Define Streamlit Frontend options.
♣ Details about app layouts, including sidebars, sidebars, and custom CSS.

Database Connections
♣ Describe the methods used to connect to SQL databases.
♣ Specify the security precautions taken when using directory references (for example, the use of environment variables and dotenv).

C. Implementation Details

1) User Interface

Describe the design and functionality of the side, including database documentation entry and parameters of the structure.
Describe the conversational interface that includes messaging as a form of interaction.

2) Database Interaction

Explain how to create a database using SQLDatabase.
Explain how database statistics are retrieved and used in query construction.

3) Troubleshooting

Discuss a template-based approach to creating SQL queries from user queries.
Explain the role of ChatOpenAI or ChatGroq in interpreting user queries and generating SQL queries.
Describe the process of creating SQL queries and processing the results.

4) Natural Language Results

Explain how to convert SQL results to natural language results.
Discuss the use of ChatPromptTemplate and StrOutputParser in generating relevant results.

D. Editing and Development

Writing: Describe the use of custom CSS in designing various elements of the application, such as basic images, text messages, and messages.

E. Testing and Evaluation

Testing Procedures
Describe the testing process, including component testing and overall system integration testing.
Specify the instrument or equipment used in the measurement.
User feedback
If possible, include feedback from users who tested the app.
Discuss changes or improvements based on user feedback.

V. ALOGRITHMS

A. Base64 encoding for images

• Function: get_base64_of_bin_file
• Purpose: This function reads an image file and encodes it on line 64. This is used to embed images in the HTML/CSS of the Streamlit application.

B. Add Custom CS

• Purpose: Adds custom CSS to Streamlit applications for styling elements such as background images, text, and containers.

C. Database Initialization

• Function: init_database
• Purpose: Initializes a SQL database connection using the specified credentials and returns a SQL database object.

D. SQL Chaining

• Function: get_sql_chain
• Purpose: To create a chain of operations to create SQL queries from user queries using a modeling language (OpenAI/Groq) and a predefined model.

E. Returns the Response

• Function: get_response
• Purpose: Returns the original response based on the user query, SQL query, and SQL response. Use a variety of actions, including verbal interactions and quick visuals.

F. Streamlit Course Management

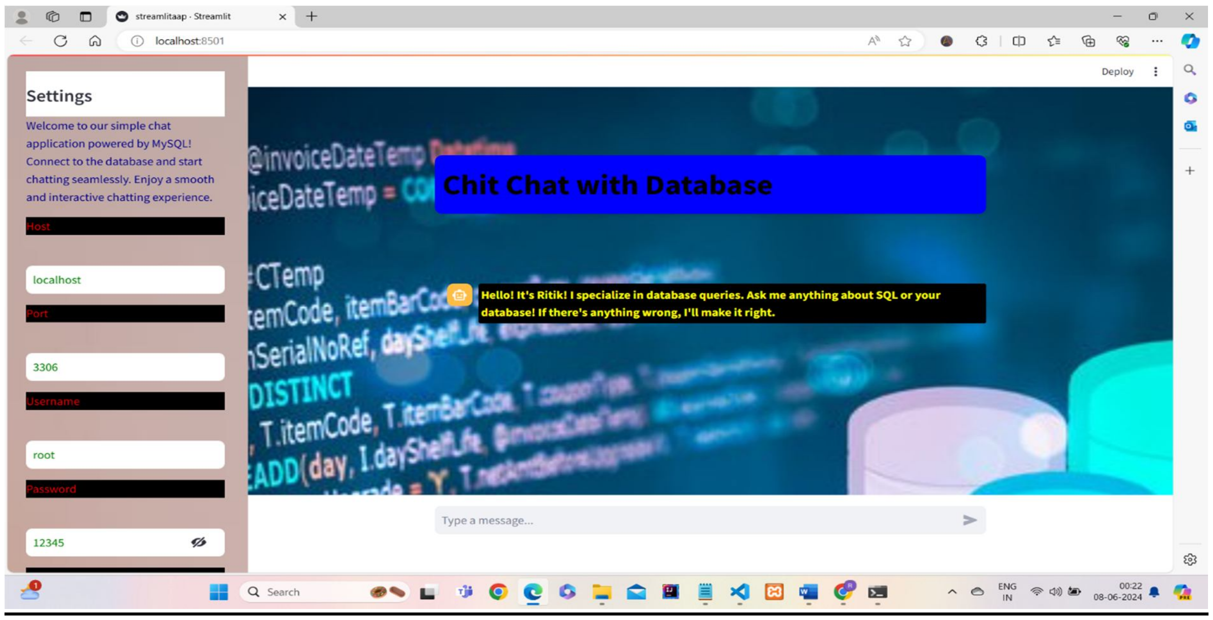
• Purpose: To manage the history of conversations and communications in the Streamlit system.
• User interface customization:
• Purpose: Use custom design styles for various user interface elements such as text, headings, and sidebar content.

G. Use of user Query

• Purpose: Use user feedback to discuss and update chat history with user messages and AI messages.

VI. RESULTS AND DISCUSSION SCREENSHOTS

The integration of LangChain and OpenAI GPT ensures that SQL queries are formulated and implemented correctly, ensuring users are presented with the desired results in a logical manner. Using Streamlit improves the user experience by providing a better and more intuitive interface.

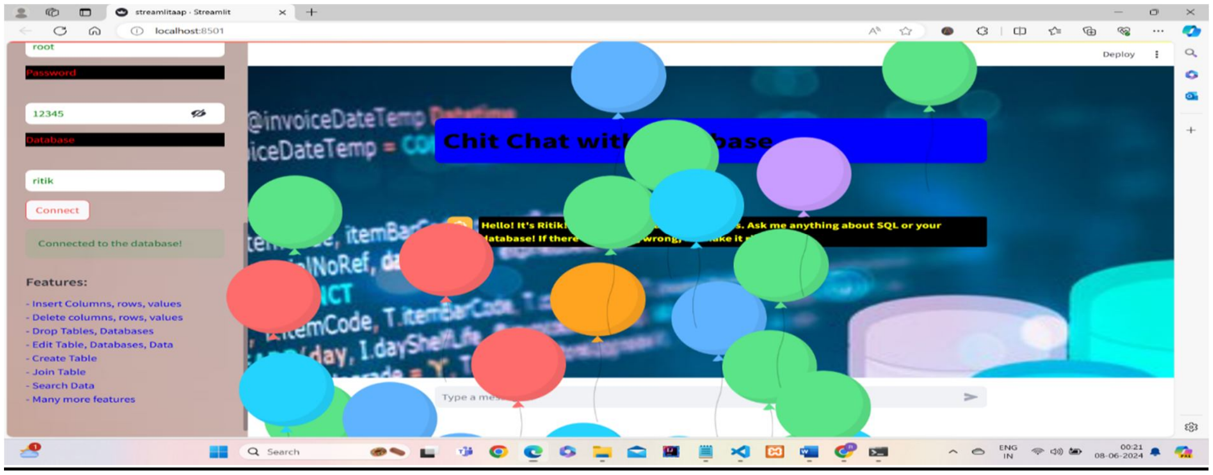


Welcome to our simple chat application powered by MySQL! Connect to the database and start chatting seamlessly. Enjoy a smooth and interactive chatting experience.

Host: localhost
Port: 3306
Username: root
Password: 12345

Hello! It's Ritik! I specialize in database queries. Ask me anything about SQL or your database! If there's anything wrong, I'll make it right.

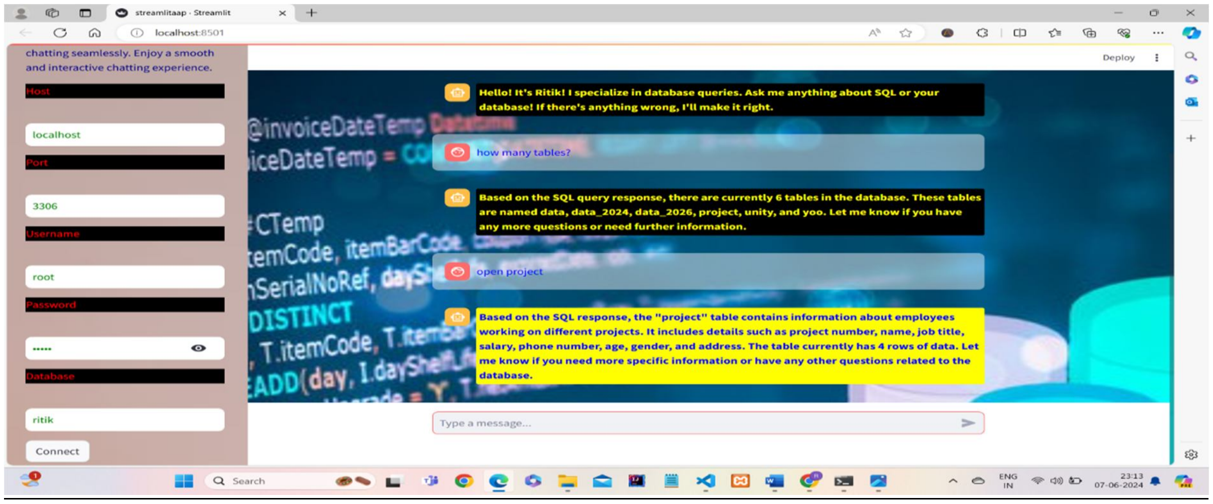
Fig 1: This is the main output page where it will ask you to fill the values in the left side to connect the database , after it will start working.



Features:

- Insert Columns, rows, values
- Delete columns, rows, values
- Drop Tables, Databases
- Edit Table, Databases, Data
- Create Table
- Join Table
- Search Data
- Many more features

Fig 2: This page will pop up the ballons after you have connected the database and it will show the message of "connected to the database" in left side.



Hello! It's Ritik! I specialize in database queries. Ask me anything about SQL or your database! If there's anything wrong, I'll make it right.

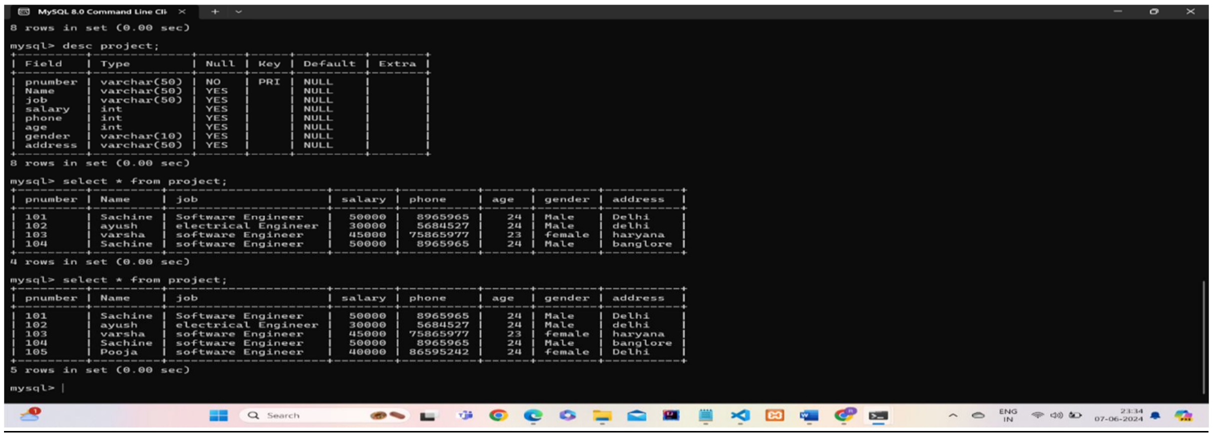
how many tables?

Based on the SQL query response, there are currently 6 tables in the database. These tables are named data, data_2024, data_2026, project, unity, and yoo. Let me know if you have any more questions or need further information.

open project

Based on the SQL response, the "project" table contains information about employees working on different projects. It includes details such as project number, name, job title, salary, phone number, age, gender, and address. The table currently has 4 rows of data. Let me know if you need more specific information or have any other questions related to the database.

Fig 3: After connection you can ask your queries ,features are mentioned in left side shown in fig 2. Here you can see one with black and yellow background color in middle of page , so the one with black background is your answer of the previous question and for current question answer there is yellow background.



```
8 rows in set (0.00 sec)

mysql> desc project;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| pnumber | varchar(50) | NO   | PRI | NULL    |       |
| Name    | varchar(50) | YES  |     | NULL    |       |
| job     | varchar(50) | YES  |     | NULL    |       |
| salary  | int         | YES  |     | NULL    |       |
| phone   | int         | YES  |     | NULL    |       |
| age     | int         | YES  |     | NULL    |       |
| gender  | varchar(10) | YES  |     | NULL    |       |
| address | varchar(50) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

mysql> select * from project;
+---------+---------+----------------------+--------+----------+------+--------+----------+
| pnumber | Name    | job                  | salary | phone    | age  | gender | address  |
+---------+---------+----------------------+--------+----------+------+--------+----------+
| 101     | Sachine | Software Engineer    |  50000 |  8965965 |   24 | Male   | Delhi    |
| 102     | ayush   | electrical Engineer  |  30000 |  5684527 |   24 | Male   | delhi    |
| 103     | varsha  | software Engineer    |  45000 | 75865977 |   23 | female | haryana  |
| 104     | Sachine | software Engineer    |  50000 |  8965965 |   24 | Male   | banglore |
+---------+---------+----------------------+--------+----------+------+--------+----------+
4 rows in set (0.00 sec)

mysql> select * from project;
+---------+---------+----------------------+--------+----------+------+--------+----------+
| pnumber | Name    | job                  | salary | phone    | age  | gender | address  |
+---------+---------+----------------------+--------+----------+------+--------+----------+
| 101     | Sachine | Software Engineer    |  50000 |  8965965 |   24 | Male   | Delhi    |
| 102     | ayush   | electrical Engineer  |  30000 |  5684527 |   24 | Male   | delhi    |
| 103     | varsha  | software Engineer    |  45000 | 75865977 |   23 | female | haryana  |
| 104     | Sachine | software Engineer    |  50000 |  8965965 |   24 | Male   | banglore |
| 105     | Pooja   | software Engineer    |  40000 | 86595242 |   24 | female | Delhi    |
+---------+---------+----------------------+--------+----------+------+--------+----------+
5 rows in set (0.00 sec)

mysql> |
```

Fig 4: It is a backend process, so the one you will execute your code in the frontend will save here in the database and give you output.

VII. CONCLUSION & FUTURE WORK

In this project, I developed a MySQL-integrated Streamlit-based conversational application designed to connect users to natural language database queries. The application provides a seamless experience for library operations using Streamlit for internet users. Key features include real-time conferencing, SQL query-based user interface, and fast database management.

Key features of the application are:

1) *Real-time Chat Interaction:* Users can interact with the application in real-time via chat, allowing them to discuss database issues.
2) *Dynamic SQL Query Generation:* The program generates SQL queries based on user queries and conversation history. This provides personalized answers to user questions and compatibility with previous interactions.
3) *Full Database Connection:* Users can connect to the MySQL database directly from the application. This feature allows data to be captured and processed in real time, providing users with up-to-date information.
4) *Natural Language Processing (NLP) Integration:* OpenAI's GPT-3.5 integration allows applications to understand and render natural language results. This improves the user experience by making interactions more intuitive and human.
5) *User-friendly Interface:* The Streamlit platform provides an intuitive and intuitive interface that increases usability and flexibility for users using a variety of technologies.

A. Future

This project demonstrates the best way to combine different technologies to create a functional and easy-to-use device. It bridges the gap between SQL database and technical users, making it easier to access data and perform database operations through natural language queries. 977 Future work mentioned above aims to further improve the program's capabilities, usability, and performance, making it a powerful tool for data analysts and users:

1) *Advanced Security Features:* Implement strong security measures such as encryption of storage data and data transmission to ensure secure communications.
2) *Advanced Natural Language Processing (NLP):* Implement substantive discussions to maintain consistency and coordination.
3) *Improve user usage and Experience:* Implement additional features to enhance user experience, such as information visualization tools, design tools, and information search capabilities. Improve the usability of the interface to provide a smooth and intuitive experience.
4) *Distribution of Functions:* Change the operating system, including making the questioning time faster and reducing the delay in information search and processing. Implement storage systems to store frequently accessed data and improve overall performance.
5) *Size and Application:* Design the app store structure to be large to support the increasing number of users and amount of data.

REFERNCES

[1] Streamlit Documentation: Streamlit. (n.d.). Streamlit Documentation. from https://docs.streamlit.io/
[2] OpenAI GPT-3 Model: OpenAI. (2024). GPT-3. from https://www.openai.com/gpt-3
[3] OpenAI API Documentation: https://beta.openai.com/docs/
[4] Python Documentation: Python Software Foundation. (n.d.). Python Documentation. from https://docs.python.org/
[5] MySQL Database Integration: MySQL. (n.d.). MySQL Documentation. from https://dev.mysql.com/doc/
[6] MySQL Connector Documentation: https://dev.mysql.com/doc/connector-python/en/
[7] dotenv Python Library: Motley. (2024). python-dotenv. from https://pypi.org/project/python-dotenv/
[8] LangChain Core and Utilities: LangChain. (n.d.). LangChain Documentation, from [LangChain GitHub repository or official documentation]
[9] LangChain Documentation: https://langchain.com/docs/
[10] Base64 Encoding in Python: Python Software Foundation. (n.d.). base64 — RFC 3548: Base16, Base32, Base64 Data Encodings. from https://docs.python.org/3/library/base64.html
[11] CSS Styling for Streamlit: Streamlit Community. (n.d.). Customizing Style in Streamlit Apps. from [appropriate Streamlit documentation or community post]
[12] Background Image in Streamlit: Streamlit Community. (n.d.). Adding Background Images to Streamlit Apps. from [appropriate Streamlit documentation or community post]
[13] APA Citation for Code: ritik. (2024). Chit Chat with Database [Python script]. from OpenAI GPT-3 model.