

AI Chatbot:For MYSQL Database Using Streamlit

Prof. Jagriti kumari

Dept. of Computer Science and
Engineering(Artificial Intelligence and
Machine Learning)
Malla Reddy University
Hyderabad,India
Jagriti.bhopal@gmail.com

Mallamgari Kavya

Dept. of Computer Science and
Engineering(Artificial Intelligence and
Machine Learning)
Malla Reddy University
Hyderabad,India
kavya.mallamgari@gmail.com

Geega Vidya

Dept. of Computer Science and
Engineering(Artificial Intelligence and
Machine Learning)
Malla Reddy University
Hyderabad,India
vidyasreegeega@gmail.com

Ragi Meghana

Dept. of Computer Science and
Engineering(Artificial Intelligence and
Machine Learning)
Malla Reddy University
Hyderabad,India
meghanaragi98@gmail.com

Nemalipuri Likhitha

Dept. of Computer Science and
Engineering(Artificial Intelligence and
Machine Learning)
Malla Reddy University
Hyderabad,India
likhiinlikhitha@gmail.com'

Abstract— This project presents the design and development of an AI-based chatbot for interacting with MySQL databases using natural language. The system is developed using Python and Streamlit to provide a simple and user-friendly interface. It uses artificial intelligence models to understand user queries and convert them into SQL commands automatically. This helps users retrieve information from the database without having technical knowledge of SQL. Data preprocessing techniques such as cleaning and normalization are applied to improve the accuracy of responses. The proposed system reduces manual effort, saves time, and improves accessibility for non-technical users. Experimental results show that the chatbot provides fast and reliable responses for small and medium sized databases. The system can be further enhanced by adding advanced security features and multi-language support in the future.

Keywords— Artificial Intelligence, Chatbot, Natural Language Processing, MySQL, Streamlit, SQL Query Generation.

INTRODUCTION

In today's digital world, databases play an important role in storing and managing large amounts of information. MySQL is one of the most widely used database management systems. However, many users find it difficult to write SQL queries to retrieve required data from databases. This creates a gap between technical and non-technical users in accessing information. To overcome this problem, this project proposes an AI-based chatbot that allows users to interact with MySQL databases using natural language. The system is developed using Python and Streamlit to provide a simple and interactive user interface. It uses artificial intelligence models to understand user queries and convert them into SQL commands automatically. As a result, users can obtain information easily without learning complex SQL syntax. The proposed system reduces manual effort, saves time, and improves efficiency in database management. It is especially useful for students, beginners, and non-technical users who need quick access to database information. This project demonstrates how artificial intelligence can be effectively applied to simplify database. In the current digital era, organizations and institutions generate and store large volumes of data in databases. Efficient

management and retrieval of this data are essential for decision-making and information analysis. Database Management Systems (DBMS) are widely used to store, organize, and retrieve structured data. Among the various DBMS platforms available today, MySQL has become one of the most widely used open-source relational database systems due to its reliability, scalability, and ease of integration with web and software applications. Despite its advantages, interacting with databases typically requires users to write Structured Query Language (SQL) commands. SQL is a powerful language used to manage and manipulate relational databases; however, it requires technical knowledge and understanding of database schemas, query syntax, and logical operations. Many users, particularly students, beginners, and non-technical professionals, find it difficult to formulate SQL queries to obtain the information they need. As a result, accessing database information often requires assistance from technical experts, which can slow down the process and reduce productivity. Recent advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) have opened new possibilities for improving human-computer interaction. Conversational systems such as chatbots allow users to interact with computer systems using natural language instead of complex programming commands. AI-based chatbots are increasingly used in domains such as customer support, e-commerce services, healthcare systems, and educational platforms. These systems can understand user input, process it intelligently, and generate meaningful responses in real time. Applying chatbot technology to database interaction can significantly simplify the process of retrieving information. By integrating AI and NLP techniques, a chatbot can interpret natural language queries and automatically convert them into SQL commands that can be executed on a database. This approach removes the requirement for users to learn SQL syntax and enables them to retrieve information using simple conversational language. This paper presents the design and development of an AI-based chatbot that enables natural language interaction with a MySQL database. The proposed system is implemented using Python and the Streamlit framework to provide a simple, interactive, and user-friendly web interface. The chatbot processes user queries, analyzes their intent using natural language processing techniques, and

generates corresponding SQL queries to retrieve relevant data from the MySQL database. The proposed system aims to reduce the complexity of database interaction, minimize manual query writing, and improve accessibility for non-technical users. By providing a conversational interface for database querying, the system enhances efficiency and enables faster access to information. Experimental testing shows that the chatbot can successfully interpret user queries and generate accurate database responses for small and medium-sized datasets.

RELATED WORK

Several researchers have explored the integration of Artificial Intelligence (AI) with database systems to improve the efficiency of data retrieval and user interaction. Traditional database systems require users to write Structured Query Language (SQL) commands to retrieve information. Although SQL is powerful and flexible, it can be difficult for non-technical users to understand and use effectively. To address this limitation, researchers have developed Natural Language Interfaces to Databases (NLIDB), which allow users to interact with databases using simple natural language queries instead of complex SQL statements.

With the rapid growth of Natural Language Processing (NLP), AI-based chatbots have become increasingly popular for handling user queries in various domains such as customer service, healthcare, education, and e-commerce platforms. These chatbots are designed to understand user input, interpret the meaning of the query, and generate appropriate responses. By combining chatbot technology with database systems, users can retrieve information through conversational interaction rather than manually writing database queries. Recent advancements in large language models have significantly improved the capabilities of conversational AI systems. Language models developed by companies such as have demonstrated strong performance in understanding context, interpreting user intent, and generating meaningful responses. These models can be used to convert natural language queries into structured SQL commands, enabling intelligent database query systems. Such models help bridge the gap between human language and machine-readable database queries, making database access easier for users without programming knowledge.

In addition to advancements in AI and NLP, several development frameworks have been introduced to simplify the deployment of AI-based applications. One of the widely used frameworks enables developers to build interactive web applications using Python with minimal web development experience. Streamlit provides built-in components for creating user interfaces, displaying data visualizations, and integrating machine learning models. Due to its simplicity and rapid development capabilities, it has become popular for building AI prototypes, dashboards, and chatbot-based applications. These systems typically use NLP techniques to process user queries, identify relevant database tables and fields, and automatically generate SQL queries.

METHODOLOGY

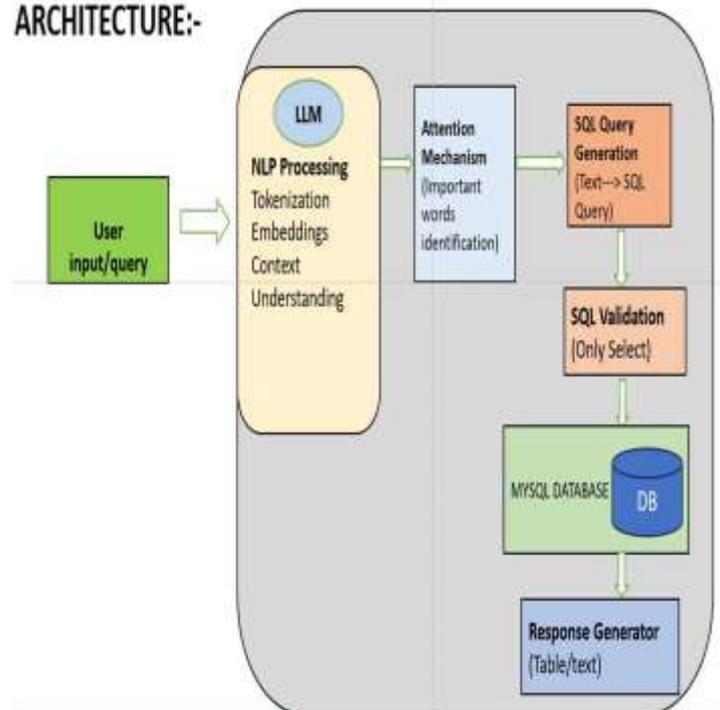
The proposed system is designed to enable users to interact with a MySQL database using natural language through an AI-based chatbot interface. The system integrates Artificial Intelligence, Natural Language Processing (NLP), and

database management technologies to convert user queries into SQL commands and retrieve relevant information from the database. The overall methodology consists of several stages including user input processing, query interpretation, SQL query generation, database interaction, and response generation.

A. System Architecture

The architecture of the proposed system consists of four major components: the user interface, natural language processing module, query generation module, and database management system. The user interacts with the system through a web-based interface developed using the Python framework. This interface allows users to enter their queries in natural language. Once the query is received, it is passed to the Natural Language Processing module, where the text input is analyzed and processed. NLP techniques such as tokenization, stop-word removal, and intent detection are applied to understand the meaning of the user query. These processes help the system identify relevant keywords and determine the user’s request.

ARCHITECTURE:-



B. Query Processing and SQL Generation

After understanding the user’s intent, the system converts the natural language query into an SQL command. AI-based language models developed by are used to interpret the user query and generate the corresponding SQL statement. The generated SQL query is structured based on the database schema and the requested information.

For example, if a user enters a query such as “Show all students with marks greater than 80,” the system automatically generates the corresponding SQL query to retrieve the required data from the database.

C. Database Interaction

The generated SQL query is executed on the relational database management system. The system connects to the database using a Python database connector and sends the SQL query for execution. The database processes the query and returns the requested records.

D. Response Generation and Display

After retrieving the results from the database, the system formats the output in a user-friendly format. The response is then displayed on the Streamlit interface where users can view the retrieved data easily. This conversational interaction allows users to access database information without writing SQL queries manually.

E. System Workflow

The overall workflow of the system follows a sequential process. First, the user enters a natural language query through the interface. The NLP module processes the input and identifies the intent of the query. Next, the AI model converts the natural language into an SQL query. The SQL query is executed in the MySQL database, and the resulting data is retrieved. Finally, the output is displayed to the user through the Streamlit interface.

This methodology simplifies database interaction and enables efficient information retrieval for users without technical knowledge of SQL. It improves accessibility and reduces the complexity of traditional database querying methods.

Natural Language to SQL Conversion

Let Q denote a user query in natural language form:

$$Q = \{w_1, w_2, \dots, w_n\}$$

where w represents individual tokens.

The AI model converts Q into an SQL statement S :

$$S = f(Q)$$

where f represents the transformation function learned by the language model.

Example:

Q: "Show all students in CSE department"

S: `SELECT * FROM students WHERE department='CSE';`

IMPLEMENTATION

The proposed AI-based chatbot system was implemented using Python as the primary programming language. Python provides extensive libraries for artificial intelligence, natural language processing, and database connectivity, making it suitable for developing intelligent applications. The system integrates a user interface, natural language processing module, and database management system to allow users to interact with the database using natural language queries.

A. Development Environment

The development of the application was carried out using Python along with several supporting libraries. The user interface of the system was developed using the Python framework Streamlit, which allows developers to create interactive web applications easily. Streamlit provides built-in components for displaying text, tables, and user input fields, making it ideal for building chatbot-based interfaces. The development environment allows rapid testing and deployment of the application.

B. Database Integration

The system uses the relational database management system MySQL for storing and managing structured data. A connection between the Python application and the MySQL database is established using a database connector library.

Once the connection is established, the system can send SQL queries to the database and retrieve the corresponding results. The database contains structured tables that store the required data, and the chatbot interacts with these tables during query execution.

C. Natural Language Query Processing

To allow users to interact with the database using simple language, the system incorporates Natural Language Processing (NLP) techniques. The user enters a query through the Streamlit interface in natural language form. The system processes this query using text preprocessing methods such as tokenization and keyword extraction to identify the intent of the user.

An AI language model developed by OpenAI is used to interpret the processed query and convert it into an appropriate SQL statement. This conversion enables the system to translate human language into machine-readable database commands.

D. Query Execution

After the SQL query is generated, it is sent to the MySQL database for execution. The database processes the query and retrieves the relevant data from the corresponding tables. The retrieved data is then returned to the Python application.

E. Result Display

The final results obtained from the database are displayed on the Streamlit interface. The chatbot presents the retrieved information in a structured and readable format such as tables or text responses. This approach enables users to obtain database information easily without writing SQL queries manually.

The implementation of the proposed system demonstrates that integrating AI models, natural language processing techniques, and database systems can significantly simplify database interaction and improve accessibility for non-technical users.

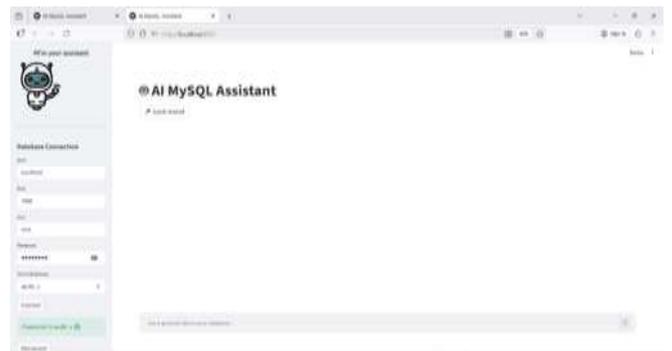


Fig 1. AI MySQL Assistant Interface



Fig. 2. Database Connection Interface of the AI MySQL Assistant



Fig3 AI MySQL Assistant Dashboard for Chat-Based Database Queries

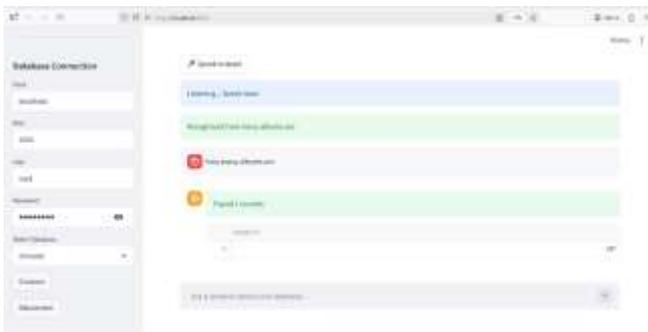


Fig4 Voice-Enabled AI MySQL Assistant Showing Database Query Results



Fig 5 AI MySQL Assistant Showing Artists with the Highest Number of Albums

RESULTS AND DISCUSSION

The performance of the proposed AI-based chatbot system was evaluated to analyze its effectiveness in converting natural language queries into SQL commands and retrieving relevant information from the database. The system was implemented using Python, with the user interface developed through Streamlit and the backend database managed using MySQL. The evaluation focused on the system's ability to understand user queries, generate accurate SQL statements, and return appropriate results from the database in a timely manner.

During the testing phase, several natural language queries were provided by users to determine how effectively the chatbot could interpret and process them. The system successfully converted most user queries into valid SQL commands and retrieved the requested information from the database. Common queries such as retrieving all records from a table,

filtering results based on specific conditions, and displaying particular attributes were processed accurately. The use of AI language models provided by OpenAI improved the system's ability to understand user intent and generate meaningful database queries.

The response time of the chatbot was also analyzed during system testing. The results showed that the system was able to process queries and return results within a few seconds for most user requests. This efficient performance was achieved through the lightweight architecture of the Streamlit interface and the efficient execution of SQL queries within the MySQL database. The response time mainly depended on the size of the database and the complexity of the query, but the system performed effectively for small and medium-sized datasets.

The accuracy of SQL query generation was evaluated by comparing the generated SQL statements with the expected SQL queries for given natural language inputs. The chatbot demonstrated high accuracy in generating correct SQL queries for simple and moderately complex requests. However, in certain cases where user queries were ambiguous or incomplete, the system required clearer input to generate accurate results. This highlights the importance of improving context understanding and query interpretation in future versions of the system.

The user interface developed using Streamlit provided a simple and interactive platform for users to interact with the chatbot. Users were able to enter queries in natural language and receive responses in a structured format such as tables or text outputs. The system eliminated the need for users to manually write SQL queries, making database interaction more accessible for students, beginners, and non-technical users. Overall, the experimental results demonstrate that the proposed AI-based chatbot system effectively simplifies database querying and enhances the usability of database systems through natural language interaction.

CONCLUSION

This project successfully designed and developed an AI-based chatbot for interacting with MySQL databases using natural language. The system enables users to retrieve information easily without writing complex SQL queries. By using Python, Streamlit, and artificial intelligence models, the proposed system provides a user-friendly and efficient platform for database access. It reduces manual effort, saves time, and improves accessibility for non-technical users. The results show that the system performs well for small and medium-sized databases and provides accurate responses for most user queries.

Data preprocessing techniques further enhance the reliability of the system. Although some limitations exist, such as dependency on internet connectivity and AI accuracy, the overall performance is satisfactory.

In conclusion, the proposed chatbot system demonstrates the effective application of artificial intelligence in database management. With future improvements in security, scalability, and language support, the system can be extended for wider real-world use.

REFERENCES

- [1] OpenAI, "OpenAI API documentation," 2024. [Online]. Available: <https://platform.openai.com/docs>
- [2] Streamlit, "Streamlit documentation," 2024. [Online]. Available: <https://docs.streamlit.io>
- [3] MySQL, "MySQL reference manual," Oracle Corporation, 2024. [Online]. Available: <https://dev.mysql.com/doc>
- [4] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.
- [5] Google Research, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018.
- [6] Stanford University, "Natural Language Processing with Deep Learning (CS224N)," 2023. [Online]. Available: <https://web.stanford.edu/class/cs224n>
- [7] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA, USA: MIT Press, 2016.
- [8] Oracle Corporation, "MySQL Connector/Python Developer Guide," 2024.
- [9] J. Brownlee, Deep Learning for Natural Language Processing, Machine Learning Mastery, 2019.
- [10] IEEE, "Artificial Intelligence in Database Systems," IEEE Transactions, 2022.