International Journal of Advanced Multidisciplinary Research and Educational Development
Volume 2, Issue 3 | May – June 2026 | www.ijamred.com
ISSN: 3107-6513

Automated Natural Language Database Query System Using NLP and Dialogflow

A Conversational Approach for Intelligent Database

Lambade Venkatesh [#1], Dr.D.William Albert [#2], G S Udaya Kiran Babu [#3]

[#1] *M.Tech Student, Dept. of CSE, Bheema Institute of Technology and Science, Adoni, India*

[#2] *Professor, Head of Dept. CSE, Bheema Institute of Technology and Science, Adoni, India*

[#3] *Professor, Dept. of CSE, Bheema Institute of Technology and Science, Adoni, India*

[#1] lmbdvnkt@gmail.com, [#2] dr.albertdwgtl@gmail.com, [#3] gsudayakiran@gmail.com

***Abstract*— Natural Language Processing (NLP) has significantly improved the development of intelligent conversational systems for database interaction. Traditional database management systems generally require structured query languages such as SQL, which may not be convenient for non-technical users. This paper presents an Automated Natural Language Database Query System using NLP and Dialogflow for simplified conversational database access. The proposed system enables users to perform database operations through natural language interaction using a chatbot interface.**

The system integrates Dialogflow for intent recognition, FastAPI for backend processing, Python for application logic, and MySQL for database management. User queries are analyzed using NLP techniques, converted into corresponding database operations, and processed automatically through webhook communication. A food ordering application is implemented to demonstrate functionalities such as order creation, item management, order tracking, and information retrieval through conversational interaction.

The developed system improves accessibility, reduces dependency on structured query knowledge, and provides an efficient human–computer interaction model for database-driven applications. The proposed approach can be extended to domains such as e-commerce, customer support, and enterprise information systems.

Keywords*— *Natural Language Processing (NLP); Dialogflow; FastAPI; Conversational AI; Chatbot System; Database Automation; MySQL; Intent Recognition; Webhook Integration; Human–Computer Interaction.

I. INTRODUCTION

Natural Language Processing (NLP) is one of the most important domains of Artificial Intelligence that focuses on enabling computers to understand, interpret, and process human language effectively. With the rapid advancement of intelligent systems and conversational technologies, modern applications increasingly aim to provide natural and user-friendly interaction between humans and computer systems. Traditional command-based interfaces are gradually being replaced by conversational systems that allow users to communicate using natural language.

Database Management Systems (DBMS) play a major role in storing, organizing, and managing information in various domains such as banking, healthcare, education, e-commerce, reservation systems, and business applications. Most database systems rely on Structured Query Language (SQL) for performing operations such as data retrieval, insertion, deletion, and modification. Although SQL is powerful and efficient, it requires users to possess technical knowledge about query syntax, database schemas, table structures, and relational operations. This creates difficulties for non-technical users who may not be familiar with database programming concepts.

In many real-world applications, users often need quick access to information without understanding complex database operations. Conventional graphical interfaces may simplify some tasks, but they still require multiple navigation steps and manual interaction. As a result, there is an increasing demand for intelligent systems capable of simplifying database communication through natural human language.

Natural Language Database Query Systems (NLDBQS) have emerged as an effective solution for overcoming these limitations. These systems enable users to communicate with databases using conversational language instead of writing structured SQL queries. By integrating Natural Language Processing with conversational AI technologies, user requests can be automatically analyzed, interpreted, and converted into corresponding database operations. Such systems improve accessibility, reduce human effort, and enhance the overall user experience.

Recent developments in chatbot technologies and machine learning have significantly improved the efficiency of conversational systems. AI-powered chatbot platforms such as Dialogflow provide advanced functionalities including intent recognition, entity extraction, conversational context handling, and webhook communication. These features allow systems to understand user intentions more accurately and support dynamic conversational interaction.

Backend frameworks also play a critical role in implementing intelligent conversational systems. FastAPI has become a popular Python-based framework for developing high-performance APIs due to its lightweight architecture, asynchronous request handling, and efficient backend processing capabilities. Database systems such as MySQL provide reliable storage and retrieval mechanisms for application data and support efficient integration with backend services.

ISSN: 3107-6513

This paper presents an Automated Natural Language Database Query System using NLP and Dialogflow. The proposed system is designed to simplify database interaction by allowing users to communicate through a chatbot interface using natural language queries. The system integrates Dialogflow for intent detection and parameter extraction, FastAPI for backend webhook processing, Python for implementing application logic, and MySQL for database management.

The proposed system is implemented using a food ordering database application as a case study. Users can perform various operations such as placing food orders, adding or removing items, checking order status, retrieving menu details, and managing order information through conversational interaction. Instead of manually writing SQL queries, users can interact naturally with the chatbot, and the system automatically converts the requests into corresponding database operations.

The overall workflow of the system begins when the user submits a natural language query through the chatbot interface. Dialogflow analyzes the query and identifies the corresponding intent and entities. The extracted information is forwarded to the FastAPI backend server through webhook communication. The backend server processes the request, interacts with the MySQL database, performs the required operations, and returns the generated response to the chatbot interface.

The proposed system offers several advantages including simplified database interaction, reduced dependency on technical query knowledge, faster information retrieval, improved accessibility, and enhanced human–computer interaction. The conversational approach also improves usability for users who may not possess technical expertise in database management systems.

The developed system can be extended to multiple real-world domains including customer support systems, e-commerce applications, healthcare information systems, educational portals, reservation management systems, and enterprise database applications. Future improvements may include multilingual query support, voice-based interaction, advanced contextual understanding, and integration with large-scale AI-based conversational models.

II. LITERATURE REVIEW

The rapid growth of Artificial Intelligence and Natural Language Processing technologies has significantly influenced the development of intelligent conversational systems. In recent years, researchers have focused on designing systems that allow users to interact with computer applications using natural language instead of traditional command-based methods. Conversational AI systems are now widely used in domains such as customer support, online shopping, healthcare, banking, and information retrieval applications.

Traditional database management systems mainly depend on Structured Query Language (SQL) for performing database operations. Although SQL provides efficient mechanisms for data retrieval and management, it requires users to understand database syntax, relational structures, and query formulation techniques. This creates difficulties for non-technical users and limits accessibility in many practical applications.

To address these limitations, several Natural Language Interface to Database (NLIDB) systems have been proposed. Early NLIDB systems were primarily based on rule-based methods and keyword matching techniques. These systems converted predefined natural language patterns into corresponding SQL queries. However, such approaches were limited in handling ambiguous queries, contextual understanding, and dynamic conversational interaction.

Recent advancements in machine learning and NLP techniques have improved the performance of conversational systems significantly. Modern chatbot frameworks are capable of understanding user intent, extracting entities, managing conversational context, and generating intelligent responses. Platforms such as Dialogflow provide advanced NLP capabilities that simplify the development of conversational applications through intent classification, entity recognition, and webhook integration.

Several studies have demonstrated the effectiveness of integrating chatbot systems with backend database services. Conversational database systems can automate repetitive tasks, simplify information retrieval, and improve user interaction efficiency. Researchers have implemented chatbot-based systems for applications including hotel booking, food ordering, healthcare assistance, educational support, and online customer service.

Backend technologies also contribute significantly to the performance of intelligent conversational systems. FastAPI has emerged as a lightweight and high-performance backend framework for API development due to its asynchronous processing capability and simplified architecture. Its efficient request handling mechanism makes it suitable for real-time chatbot applications and webhook communication systems.

Database management systems such as MySQL are commonly integrated with conversational applications for storing and managing application data. MySQL provides reliable relational database functionality, fast query execution, and efficient backend connectivity. Combined with Python-based backend services, it supports the development of scalable and efficient database-driven applications.

Although many conversational systems have been developed, several challenges still remain in areas such as contextual understanding, ambiguity handling, multilingual support, and dynamic query generation. Some systems also face limitations in handling complex user interactions and maintaining conversational continuity.

The proposed

system addresses these challenges by integrating Dialogflow, FastAPI, Python, and MySQL to create an efficient Automated Natural Language Database Query System. The system focuses on improving accessibility, simplifying database interaction, and providing a user-friendly conversational environment for performing database operations through natural language communication.

ISSN: 3107-6513

III. PROPOSED METHOD

A. System Overview

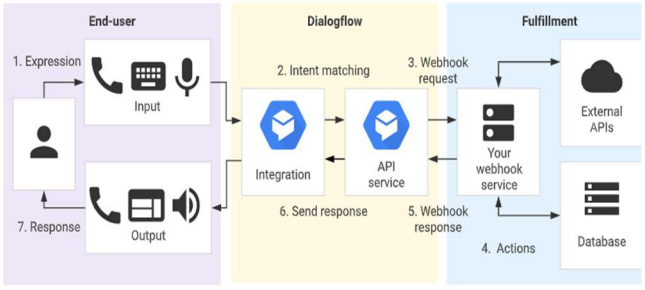
The proposed Automated Natural Language Database Query System is designed to simplify the interaction between users and database systems through conversational communication. Instead of using traditional SQL queries or complex graphical interfaces, users can interact with the system using natural language through a chatbot interface. The system automatically interprets user requests and performs corresponding database operations.

The proposed model integrates Natural Language Processing, conversational AI, backend API services, and relational database management technologies to provide intelligent database interaction. The system mainly uses Dialogflow for intent recognition and entity extraction, FastAPI for backend webhook processing, Python for application logic implementation, and MySQL for database management.

The system allows users to perform various operations such as creating orders, modifying order details, retrieving information, tracking records, and checking application-related details using conversational commands. When a user submits a message through the chatbot interface, Dialogflow analyzes the query and identifies the corresponding user intent and required parameters.

The extracted information is forwarded to the FastAPI webhook server, where backend processing is performed. The server interacts with the MySQL database to execute required operations such as insertion, deletion, updating, or retrieval of records. After processing the request, the generated response is returned to Dialogflow and displayed to the user through the chatbot interface.

The proposed system reduces the complexity involved in database interaction and improves accessibility for users without technical knowledge of SQL or database structures. The conversational approach also improves user experience by providing faster and more intuitive communication with the database system.



The overall architecture of the proposed system is shown in Fig. 1.

B. System Overview

The architecture of the proposed system is designed to establish efficient communication between the user, chatbot platform, backend server, and database system. The architecture mainly consists of five important modules: User Interface, Dialogflow NLP Engine, FastAPI Backend Server, Application Processing Module, and MySQL Database.

The User Interface acts as the communication medium between the user and the system. Users interact with the chatbot through a web-based interface integrated using Dialogflow Messenger. The interface allows users to submit natural language queries related to database operations.

Dialogflow functions as the Natural Language Processing engine of the system. It analyzes the user query and identifies the corresponding intent using trained conversational models. It also extracts important entities and parameters such as food item names, quantities, and order details from the user message.

After intent detection, Dialogflow forwards the request to the FastAPI webhook server through webhook communication. The FastAPI backend server processes the request and executes the required application logic. The backend module handles tasks such as processing user requests, generating SQL operations, managing order details, and retrieving information from the database.

The MySQL database stores all application-related information including food items, customer orders, and order tracking details. The backend server communicates with the database to perform insertion, deletion, updating, and retrieval operations based on user queries.

The overall workflow of the system begins when the user submits a natural language query through the chatbot interface. Dialogflow processes the request and forwards the extracted information to the backend server. The FastAPI server performs the required database operation and returns the generated response to Dialogflow, which is then displayed to the user through the conversational interface.

The architecture provides a scalable, user-friendly, and efficient solution for conversational database interaction.

C. Working Procedure

The proposed system follows a conversational workflow for processing user queries and performing database operations automatically. The complete process involves communication between the chatbot interface, Dialogflow NLP engine, backend server, and MySQL database.

Initially, the user submits a query through the chatbot interface using natural language. The query may contain requests such as placing a food order, adding items, removing products, checking order status, or retrieving information from the database.

After receiving the message, Dialogflow processes the input using Natural Language Processing techniques. The system identifies the user intent and extracts required entities and parameters from the query. These parameters may include item names, quantities, order IDs, or other related information.

The processed request is then forwarded to the FastAPI backend server through webhook integration. The backend server interprets the received intent and executes the

ISSN: 3107-6513

corresponding application logic. Depending on the request type, the backend module performs operations such as data insertion, record updating, deletion, or information retrieval.

The FastAPI server communicates with the MySQL database to perform the required database operations. The database processes the query and returns the requested information or operation status to the backend server.

After successful processing, the backend server generates a response message and sends it back to Dialogflow. Finally, the chatbot interface displays the generated response to the user in conversational form.

The complete workflow improves the efficiency of database interaction and allows users to communicate with the system in a simple and user-friendly manner without requiring knowledge of SQL queries or database structures.

The proposed system follows a conversational workflow for processing user queries and performing database operations automatically. The complete process involves communication between the chatbot interface, Dialogflow NLP engine, backend server, and MySQL database.

Initially, the user submits a query through the chatbot interface using natural language. The query may contain requests such as placing a food order, adding items, removing products, checking order status, or retrieving information from the database.

After receiving the message, Dialogflow processes the input using Natural Language Processing techniques. The system identifies the user intent and extracts required entities and parameters from the query. These parameters may include item names, quantities, order IDs, or other related information.

The processed request is then forwarded to the FastAPI backend server through webhook integration. The backend server interprets the received intent and executes the corresponding application logic. Depending on the request type, the backend module performs operations such as data insertion, record updating, deletion, or information retrieval.

The FastAPI server communicates with the MySQL database to perform the required database operations. The database processes the query and returns the requested information or operation status to the backend server.

After successful processing, the backend server generates a response message and sends it back to Dialogflow. Finally, the chatbot interface displays the generated response to the user in conversational form.

The complete workflow improves the efficiency of database interaction and allows users to communicate with the system in a simple and user-friendly manner without requiring knowledge of SQL queries or database structures.

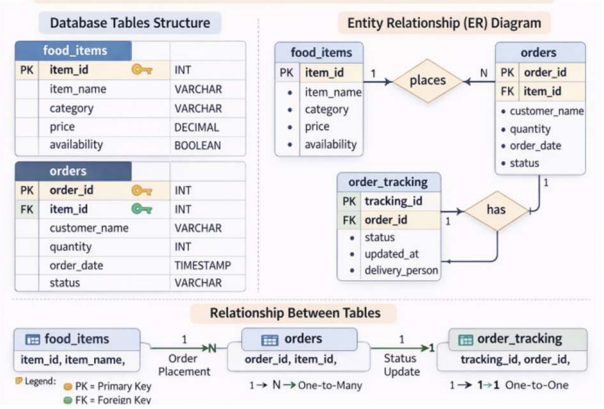
D. Database Design

The proposed system uses a MySQL relational database for storing and managing application data. The database is designed to support efficient storage, retrieval, updating, and tracking of records generated through conversational interactions. The database structure is organized to maintain proper relationships between food items, customer orders, and order tracking details.

The database mainly consists of three important tables: food_items, orders, and order_tracking. These tables are interconnected to support automated order management and conversational database operations.

The food_items table stores information related to available food products and their prices. The orders table maintains customer order details including ordered items and quantity information. The order_tracking table is used to maintain the current status of each order such as order received, preparing, ready, or delivered.

The Entity Relationship (ER) diagram representing the relationship between different database entities is shown in Fig. 2.



The structure of the main database tables is shown in Table I.

TABLE I

DATABASE TABLE STRUCTURE

Table Name	Purpose
food_items	Stores food item details and prices
orders	Stores customer order information
order_tracking	Stores current order status

The food_items table contains attributes such as item identification number, item name, and price details. The orders table stores order identification, ordered item names, and quantity information. The order_tracking table maintains the delivery and preparation status of customer orders for tracking purposes.

This database structure supports efficient conversational database interaction and enables smooth integration between the backend application and the chatbot system.

E. Advantages of the Proposed System

Paste this:

The proposed conversational database query system provides several advantages compared to traditional database interaction methods. The system allows users to communicate

ISSN: 3107-6513

with the database using natural language instead of structured SQL queries. This reduces the requirement for technical knowledge and improves accessibility for non-technical users.

The integration of Dialogflow and FastAPI improves conversational interaction and backend processing efficiency. Automated intent recognition and parameter extraction simplify database operations and reduce manual effort. The chatbot-based interface also provides faster information retrieval and improves the overall user experience.

The proposed system can be easily extended to multiple real-world applications such as customer support systems, e-commerce platforms, healthcare information systems, reservation systems, and educational applications.

F. Algorithm for Query Processing

The proposed system follows a conversational query processing algorithm for converting natural language queries into database operations. The algorithm processes user requests through intent detection, parameter extraction, backend processing, and database execution.

Algorithm Steps

Step 1: Start the system.

Step 2: Receive user query from chatbot interface.

Step 3: Send the query to Dialogflow NLP engine.

Step 4: Identify user intent and extract entities.

Step 5: Forward webhook request to FastAPI server.

Step 6: Process the request using backend application logic.

Step 7: Execute corresponding database operation in MySQL.

Step 8: Retrieve generated response from database.

Step 9: Send response back to Dialogflow.

Step 10: Display conversational response to the user.

Step 11: Stop.

TABLE III

COMPARISON BETWEEN TRADITIONAL AND PROPOSED SYSTEM

Feature	Traditional Database system	Proposed system
Query Method	SQL Queries	Natural Language
Technical Knowledge	Required	Not Required
User Interaction	Form-based	Conversational
Accessibility	Limited	Improved
Database Handling	Manual	Automated
User Experience	Complex	User-Friendly

IV. IMPLEMENTATION

The implementation of the proposed Automated Natural Language Database Query System is carried out using Dialogflow, FastAPI, Python, MySQL, HTML, and CSS technologies. The system integrates conversational AI with backend database operations to provide intelligent natural language interaction.

Dialogflow is used as the Natural Language Processing platform for intent recognition and entity extraction. Different intents are created to represent user operations such as creating orders, adding items, removing products, tracking orders, and retrieving information. Training phrases are provided for each intent to improve conversational understanding and response generation.

The FastAPI framework is used for implementing the backend webhook server. The backend server receives requests from Dialogflow, processes user intents, and executes corresponding database operations. Python helper modules are used to simplify request handling, application logic implementation, and database communication.

MySQL is used as the relational database management system for storing application data. Database tables maintain information related to food items, customer orders, and order tracking details. SQL queries and helper functions are used to perform insertion, updating, deletion, and retrieval operations efficiently.

A simple web interface is developed using HTML and CSS to integrate the chatbot into the application environment. Dialogflow Messenger is embedded within the website to provide conversational interaction between the user and the system.

The implementation workflow of the proposed system is shown in Fig. 3.

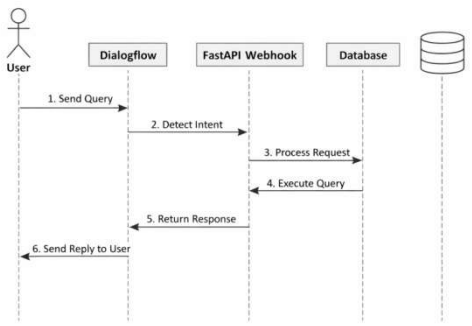
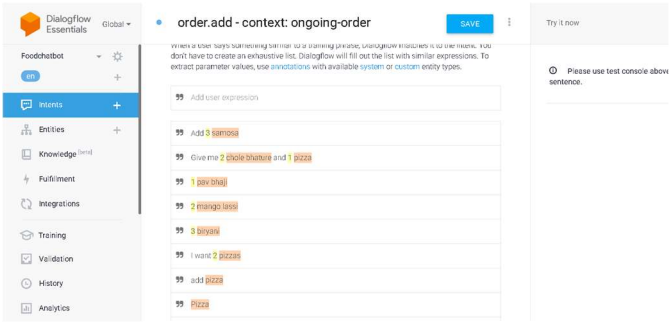


Figure 7.3: Sequence Diagram of the Chatbot Interaction

ISSN: 3107-6513

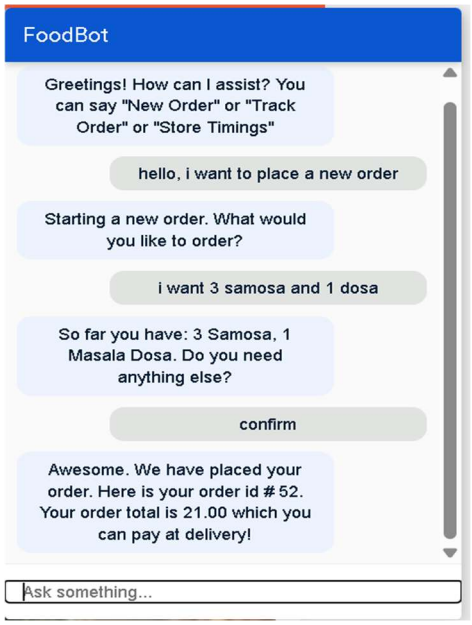
a. Dialogflow Intent Configuration

The chatbot intents are configured in Dialogflow to identify different user operations and extract required entities from natural language queries. Intents such as new.order, order.add, order.remove, and track.order are used for processing conversational requests.



b. Website Interface with Chatbot

The frontend interface provides a conversational environment where users can interact with the chatbot system through natural language messages. The chatbot is integrated using Dialogflow Messenger.

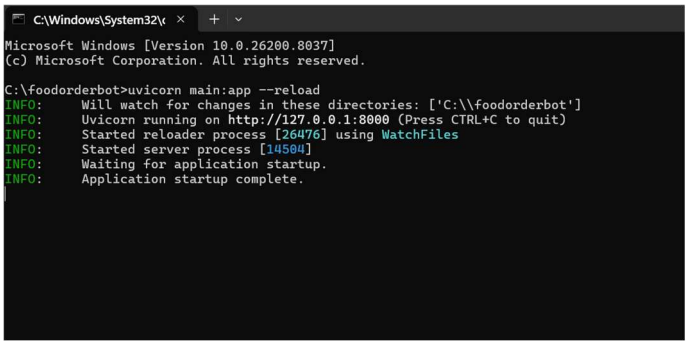


c. Webhook Request Processing

When a user submits a conversational query through the chatbot interface, Dialogflow generates a structured webhook request containing the detected intent and extracted entities. The FastAPI backend server processes the request and performs the required database operation.

The webhook request generally contains details such as user intent, item names, quantities, and order information. The backend server interprets these parameters and executes corresponding SQL operations through the MySQL database.

The response generated by the backend server is returned to Dialogflow and displayed to the user through the chatbot interface. This communication mechanism enables real-time conversational database interaction.



d. Technologies Used

The proposed system uses multiple technologies for implementing conversational database interaction. Dialogflow is used for Natural Language Processing and conversational management. FastAPI is used for backend webhook processing and API handling. Python is used for implementing application logic and database communication. MySQL is used as the relational database management system for storing and managing application data. HTML and CSS are used for developing the frontend web interface integrated with the chatbot system.

TABLE III

TECHNOLOGIES USED IN THE SYSTEM

Technology	Purpose
Dialogflow	Intent Recognition
FastAPI	Backend API Processing
Python	Application Logic
MySQL	Database Management
HTML/CSS	Frontend Interface

IV. RESULTS AND DISCUSSION

The proposed Automated Natural Language Database Query System was tested using different conversational queries to evaluate the performance of the chatbot and database interaction process. The testing process focused on intent recognition accuracy, conversational response generation, database operation execution, and overall system functionality.

The system successfully interpreted user queries related to food ordering, item management, order tracking, and information retrieval. Dialogflow accurately detected user intents and extracted required parameters from natural language queries. The FastAPI backend server efficiently processed webhook requests and communicated with the MySQL database for performing corresponding operations.

Several test cases were executed to verify the working performance of the system. The chatbot successfully created new

ISSN: 3107-6513

orders, updated order details, removed items, and retrieved tracking information based on conversational commands. The generated responses were displayed correctly through the chatbot interface.

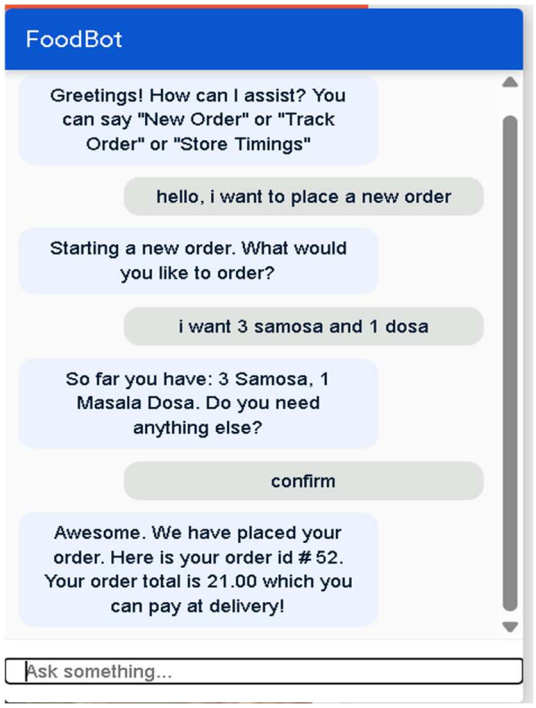
The sample test cases used for system evaluation are shown in Table II.

TABLE III

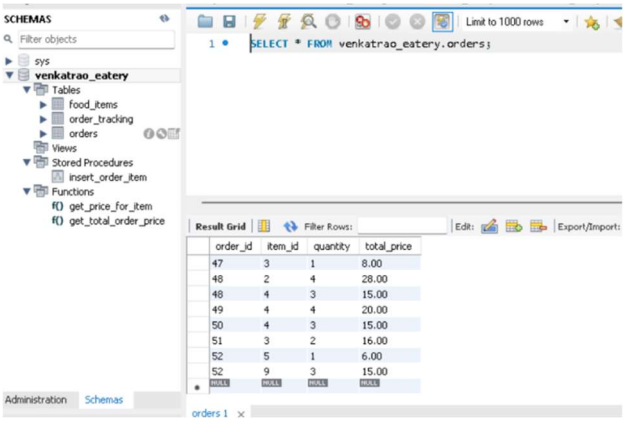
SAMPLE TEST CASES AND RESULTS

Test Case	User Query	Expected Result	Status
TC01	I want to order pizza	New order created	Success
TC02	Add two burgers to my order	Item added successfully	Success
TC03	Remove pizza from my order	Item removed successfully	Success
TC04	Track my order	Order status displayed	Success

The chatbot interaction during order placement is shown in Fig. 6.



The database records generated after chatbot interaction are shown in Fig. 7.



The obtained results demonstrate that the proposed system effectively converts natural language queries into database operations using conversational AI techniques. The integration of Dialogflow, FastAPI, and MySQL provides efficient communication between the chatbot interface and backend database system. The proposed approach improves user accessibility, reduces dependency on SQL knowledge, and provides a user-friendly database interaction environment.

Performance Analysis

The performance of the proposed system was evaluated based on conversational response accuracy, intent detection efficiency, database operation execution, and user interaction capability. The chatbot successfully handled multiple conversational queries related to food ordering and information retrieval.

The integration of Dialogflow with FastAPI provided efficient communication between the conversational interface and backend processing modules. The MySQL database successfully maintained order records and tracking information generated through chatbot interaction.

The system demonstrated reliable conversational performance and reduced the complexity involved in traditional SQL-based database interaction methods.

VI. CONCLUSION

This paper presented an Automated Natural Language Database Query System using NLP and Dialogflow for simplifying conversational interaction with database systems. The proposed system enables users to perform database operations through natural language queries instead of traditional SQL-based interaction methods.

The system integrates Dialogflow for intent recognition and entity extraction, FastAPI for backend webhook processing, Python for application logic implementation, and MySQL for database management. A food ordering application was implemented as a case study to demonstrate conversational operations such as order creation, item management, order tracking, and information retrieval.

ISSN: **3107-6513**

The obtained results show that the proposed system successfully interprets user queries, performs corresponding database operations, and generates conversational responses efficiently. The chatbot-based interaction model improves accessibility for non-technical users and reduces the complexity involved in traditional database communication.

The experimental results demonstrate that conversational AI techniques can significantly improve database accessibility and simplify user interaction with backend systems. The proposed chatbot-based model successfully combines NLP, webhook processing, and relational database management into a unified conversational framework.

The proposed approach can be extended to several real-world applications including e-commerce platforms, healthcare systems, reservation management systems, customer support services, and enterprise information systems. Future enhancements may include multilingual support, voice-based interaction, advanced contextual understanding, and integration with AI-based intelligent conversational models for improving system performance and user experience.

Future Scope

The proposed system can be further enhanced by integrating advanced Artificial Intelligence and machine learning techniques for improving conversational understanding and contextual response generation. Future improvements may include multilingual query support, voice-based interaction, cloud database integration, and AI-powered recommendation systems.

The system can also be extended for applications such as healthcare management systems, online reservation systems, banking services, educational platforms, and enterprise information systems. Integration with large language models and intelligent analytics can further improve conversational accuracy and user experience.

VII. REFERENCES

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Pearson Education, 2021.

[2] T. Winograd, “Procedures as a Representation for Data in a Computer Program for Understanding Natural Language,” *MIT Artificial Intelligence Laboratory*, 1971.

[3] A. Kumar and S. Roy, “Natural Language Interface for Database Query Processing Using NLP Techniques,” *International Journal of Computer Applications*, vol. 179, no. 12, pp. 20–25, 2020.

[4] Google, “Dialogflow Documentation,” Available: https://cloud.google.com/dialogflow/docs

[5] S. Ramakrishnan and P. Geetha, “Conversational AI Chatbot for Database Query Processing,” *International Journal of Advanced Research in Computer Science*, vol. 11, no. 4, pp. 45–50, 2020.

[6] S. Kumar, “FastAPI for Modern Web Application Development,” *Journal of Web Engineering and Technology*, vol. 5, no. 2, pp. 15–22, 2021.

[7] Oracle Corporation, “MySQL Reference Manual,” Available: https://dev.mysql.com/doc/

[8] M. Naik and R. K. Shyamasundar, “Natural Language Interfaces to Databases: An Introduction,” *Computer Science Review*, vol. 8, no. 3, pp. 125–136, 2019.

[9] T. B. Brown et al., “Language Models are Few-Shot Learners,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[10] A. Vaswani et al., “Attention Is All You Need,” *Neural Information Processing Systems (NIPS)*, pp. 5998–6008, 2017.