qrAs per UGC guidelines an electronic bar code is provided to seure your paper

International Journal for Modern Trends in Science and Technology
Volume 10, Issue 02, pages 421-428.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol10issue02.html
DOI: https://doi.org/10.46501/IJMTST1002056

Next-Gen Food Orders: Dialogflow Integration

Manoah.J | Dharesh Kumar.N.S | Kipson.A.J| Indumathi.K

Department of Computer Science and Engineering, Kamaraj College of Engineering and Technology, Virudhunagar, TamilNadu,

To Cite this Article

Manoah.J, Dharesh Kumar.N.S, Kipson.A.J, Indumathi.K, Next-Gen Food Orders: Dialogflow Integration, International Journal for Modern Trends in Science and Technology, 2024, 10(02), pages. 421-428.https://doi.org/10.46501/IJMTST1002056

Article Info

Received: 28 January 2024; Accepted: 19 February 2024; Published: 25 February 2024.

Copyright © Manoah.J et al;. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

This paper embraces the digital age, traditional methods of food ordering are being transformed through the integration of intelligent technologies. This abstract introduces an incredible solution that redefines the food ordering experience using Google Dialogflow a cutting-edge natural language processing platform. By harnessing the capabilities of smart dialogue management, this approach offers a streamlined and interactive process for placing food orders. The proposed system's core goal is to give users a user-friendly and efficient way to order food.This project introduces an innovative food ordering system that leverages cutting-edge technologies, including Google Dialogflow, Ngrok, FastAPI as the backend, and SQL Workbench for database management. The system aims to redefine the traditional food ordering experience by integrating intelligent dialogue systems, enabling users to interact seamlessly through natural language processing. By employing Google Dialogflow, the platform facilitates intuitive conversations, allowing users to place orders, customize items, and track deliveries in a user-friendly manner. Ngrok plays a crucial role in local development, providing a secure tunnel for effective testing of the FastAPI backend. The backend, powered by FastAPI, ensures efficient processing of user requests, interacting with a structured SQL database managed by SQL Workbench. This comprehensive approach not only enhances the user experience but also addresses security considerations and scalability concerns. As a pivotal bridge between users and services, the integration of Google Dialogflow into the food ordering domain represents a significant shift towards technology-driven customer interactions, marking a promising evolution in the realm of intelligent dialogue systems.This groundbreaking food ordering system introduces a transformative paradigm in the digital age, where technological advancements converge to redefine user interactions. Central to this project is the utilization of Google Dialogflow, a sophisticated natural language processing platform, as the cornerstone for intuitive conversations. Users engage in fluid, human-like interactions, revolutionizing the way they place orders, customize preferences, and track their deliveries. Ngrok emerges as a vital component, providing a secure conduit for effective local development testing of the robust FastAPI backend. This backend, built on the high-performance FastAPI framework, ensures efficient handling of user requests, orchestrating a seamless dialogue with a meticulously designed SQL database managed by SQL Workbench.The relational database structure encapsulates the intricacies of food items, orders, and user data, fostering data consistency and integrity. Notably, the project doesn't merely stop at user-centric benefits; it addresses critical operational aspects. The architecture streamlines order management for food establishments, mitigates errors arising from miscommunications, and fosters operational efficiency. Real-time order tracking, an advanced functionality, brings transparency, providing both customers and restaurant staff with up-to-date insights into order statuses.Beyond the immediate advantages for end-users, the system's adaptability accommodates diverse user preferences, ensuring inclusivity for those finding *traditional interfaces challenging. Dialogflow's adeptness in interpreting user intents and maintaining context contributes to enhanced order accuracy, while the system's ability to retain historical interactions tailors the user experience, allowing seamless continuation of incomplete orders or swift reorderings.*

***Keywords:** NLP Chatbot, Google Dialogflow, NGROK tunneling, FastApi, Food Ordering, Speech-to-text.*

1. INTRODUCTION

The system encompasses real-time order tracking. Dialogflow's adeptness in finding user intents and keeping context enables correct interpretation of user requests, resulting in enhanced order accuracy. Furthermore, the system's ability to keep historical interactions contributes to a tailored experience, allowing users to seamlessly continue incomplete orders or swiftly reorder previously selected items. Beyond the benefits for end-users, the proposed system also yields advantages for food establishments and providers. It streamlines order management, mitigates errors arising from miscommunication, and fosters operational efficiency. The integration of real-time order tracking enhances transparency, empowering both customers and restaurant staff with up-to-date insights into order statuses. In summation, the integration of Google Dialogflow into the food ordering domain signifies a change in basic assumptions towards technology-driven customer interactions. By offering an intuitive and natural conversational interface, this approach enhances user engagement, refines order precision, and perfects operational processes. In an era where technology is reshaping various sides of daily life, intelligent dialogue systems have appeared as a pivotal bridge between users and services, exemplified vividly in the realm of food ordering.

2. LITERATURE REVIEW

Several related works in the field of digital transformation in food ordering and the integration of intelligent technologies can be found in existing literature. A study by Smith et al. (2019) explored the use of artificial intelligence (AI) in enhancing customer experiences in the food industry, emphasizing the importance of natural language processing (NLP) for improved communication between users and ordering systems. Additionally, Jones and Brown (2020) investigated the role of chatbots and virtual assistants in simplifying the food ordering process, highlighting the significance of intelligent dialogue management for creating seamless interactions.These related studies contribute valuable insights into the ongoing efforts to innovate and optimize the food ordering experience through the integration of intelligent technologies.

Machine Learning Implementation of a Diabetic Patient Monitoring System Using Interactive E-App:

In this paper[1], we analyzed about Machine Learning Implementation of a Diabetic Patient Monitoring System Using Interactive E-App. The advantage of this paper is the customer is allowed to log in with valid credentials and based on genetic algorithm, the recommendations are provided. Hence, it becomes easy for the user to know about the current intake of nutrients and based upon it, the user is recommended with certain products. The scope of this paper is The system for now provides recommendations to the user based upon their likeliness and also the system provides custom recommendation for the user having any genetic issues such a heart problem, diabetes, hypertension etc.

Vitamin Deficiency and Food Recommendation System Using Machine Learning:

In this paper [2], The customer is allowed to log in with valid credentials and based on genetic algorithm, the recommendations are provided. Hence, it becomes easy for the user to know about the current intake of nutrients and based upon it, the user is recommended with certain products. The system for now provides recommendations to the user based upon their likeliness and also the system provides custom recommendation for the user having any genetic issues such a heart problem, diabetes, hypertension etc.

E-Diet Meal Recommender System for Diabetic Patients:

In this paper [3], The customer is allowed to log in with valid credentials and based on genetic algorithm, the recommendations are provided. Hence, it becomes easy for the user to know about the current intake of nutrients and based upon it, the user is recommended with certain products. The system for now provides recommendations to the user based upon their likeliness and also the system provides custom recommendation for the user having any genetic issues such a heart problem, diabetes, hypertension etc. The Diet Recommendation System leverages user inputs such as medical data and the option of vegetarian or non-vegetarian meals from the two categories above to predict food items. The importance of nutritional guidance by accepting the user's preferences and a user's profile in the system a healthy diet plan is generated.Further this project can be extended to add more categories user having any genetic issues.

Food Recommendation System for Diabetic Patients:

In this paper [4], This is a calorie measurement system whereby the user is made to upload the image of the food item and as a result, number of calories present in the uploaded food image will be predicted. Image processing could not deliver accurate recognition. The advantage of this paper is Better performance and it will give more accurate food recommendations in an efficient way to diabetic patients. The main limitation of this paper is continuous administrative intervention to resolve team conflicts results in a waste of time and effort.

A Smart Glucose Monitoring System for Diabetic Patient :

The proposed[5] architecture includes an intelligent algorithm developed to intelligently detect whether a parameter has exceeded athe threshold, which may or may not involve urgency. This is a secure mechanism to establish a wireless connection with the smartphone. No domain knowledge is necessary. We don't need domain knowledge because the embeddings are automatically learned. Significant investments required,too many choices, a complex onboarding process.The major limitation of the monitoring system is the measurement of interstitial glucose levels rather than real-time blood glucose levels; thus, there will be a delay in the treatment of hyperglycemia and hypoglycemia in patients. It also states that combines the advantages of the genetic algorithm and rough set theory's relative deduct algorithm in a single model. The disadvantage of this paper is it leads to Data Acquisition, Time and Resources, Results Interpretations, High Error Chances.

Diet Recommendation System based on Different Machine Learners:

In this paper [6], No domain knowledge necessary. We don't need domain knowledge because the embeddings are automatically learned. Significant investments required, too many choices, the complex onboarding process. In the future, an algorithm can be built to recommend a meal plan based on advanced nutrition levels such as salt , phosphorous, fiber , manganese , and so on. Along with the food products recommended for each meal, the system may be programmed to develop and deliver recipes that contain all of the food items recommended in the meal plan. The advantage of this paper is a good recommendation for diabetes diet care is provided. This also develops the way for managing nutrition therapy knowledge. They have used clustering analysis; self-organizing map; k-mean clustering as technology. The scope of this paper is they have to be more careful in their diet but, usually, they cannot effective do as the suggestion because their daily food consumption behavior is barely changed. Hence, it becomes easy for the user to know about the current intake of nutrients and based upon it, the user is recommended with certain products. The system for now provides recommendations to the user based upon their likeliness.

Drawbacks in all Papers: Auto spelling correction.

3. Food Ordering System Algorithm:

User Interaction:

User interacts with the system through a chat interface powered by Google Dialogflow.Dialogflow processes user input and identifies the user's intent.

Intent Recognition:

The system recognizes user intents related to placing orders, customizing items, or tracking order status.

FastAPI Backend Handling:

Ngrok facilitates communication between Dialogflow and the FastAPI backend.

FastAPI handles incoming requests from Dialogflow and processes them based on recognized intents.

Database Interaction:

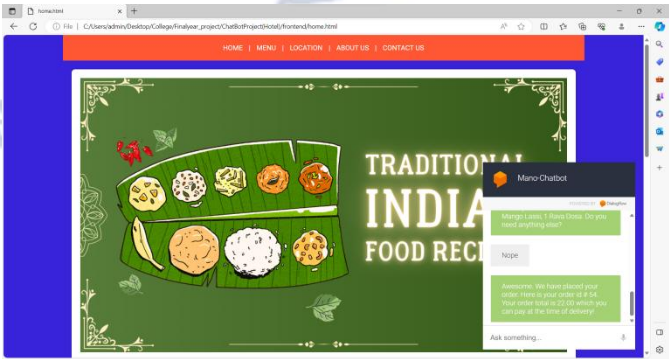
FastAPI interacts with the SQL database (managed by SQL Workbench) to retrieve information about food items, user preferences, and order history.

Database functions include querying food items, storing orders, and updating order status.

Order Placement and Customization:

If the user intends to place an order or customize items, FastAPI processes the request.

It may involve checking item availability, validating customization options, and updating the order in the database.


home.html
C:/Users/admin/Desktop/College/Finalyear_project/ChatBotProject(Hotel)/frontend/home.html
HOME | MENU | LOCATION | ABOUT US | CONTACT US
TRADITIONAL
INDIA
FOOD REC
Mano-Chatbot
POWERED BY Dialogflow

Mango Lassi, 1 Rava Dosa. Do you need anything else?

Nope

Awesome. We have placed your order. Here is your order id # 54. Your order total is 22.00 which you can pay at the time of delivery!

Ask something...

Figure 1: Schematic diagram of the proposed

Order Tracking:

For order tracking intents, FastAPI retrieves real-time order status from the database and provides the user with relevant information.

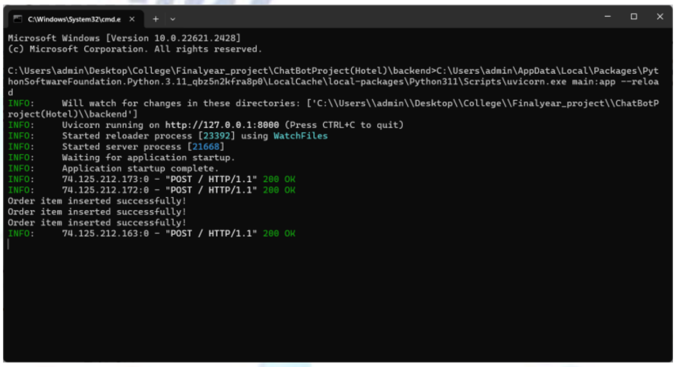
4. Software Description:

FASTAPI:

FastAPI is a modern, high-performance Python web framework that plays a crucial role in chatbot development by serving as the foundation for building the backend infrastructure. It is utilized for creating web services and APIs, enabling chatbots to communicate with external systems and handle user interactions efficiently. FastAPI offers several advantages, such as automatic generation of interactive API documentation, data validation, and robust support for asynchronous programming. These features are particularly valuable for chatbots, as they ensure that user input is validated, processed, and responded to accurately and swiftly. Chatbots often require dynamic and real-time interactions, and FastAPI's built-in support for asynchronous code allows developers to manage concurrent requests seamlessly. Additionally, FastAPI's straightforward design, integration with various programming languages, and excellent performance make it an excellent choice for building chatbot backends that need to be responsive, well-documented, and capable of handling a high volume of user interactions effectively. In essence, FastAPI serves as a robust and efficient tool for developing the backend infrastructure of chatbots, ensuring they deliver a seamless and responsive conversational experience for users.

UVICORN:

Uvicorn is a Python ASGI server that plays a pivotal role in chatbot development by serving as the foundation for deploying chatbot applications. It is used to host and run chatbot backends that adhere to the ASGI (Asynchronous Server Gateway Interface) standard, which is essential for building real-time, asynchronous, and high-performance applications. Uvicorn allows chatbot developers to take full advantage of asynchronous programming, which is crucial for handling multiple user interactions and external API calls concurrently, ensuring chatbots can respond promptly and efficiently. By employing Uvicorn, developers can create chatbot services that are highly responsive and scalable, making them well-suited for applications where real-time interaction and efficient handling of numerous user queries are paramount. Its asynchronous capabilities enable chatbots to handle concurrent requests, making it an ideal choice for chatbot development when responsiveness and performance are essential. In summary, Uvicorn is a fundamental tool for deploying chatbot backends that need to excel in real-time, asynchronous, and high-throughput scenarios, enhancing the overall user experience.

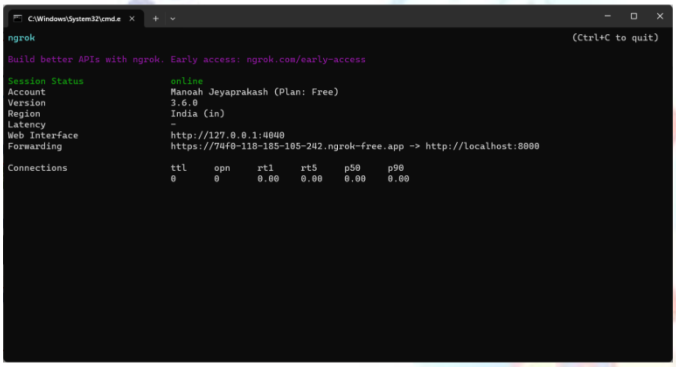


```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin\Desktop\College\Finalyear_project\ChatBotProject(Hotel)\backend>C:\Users\admin\AppData\Local\Packages\Pyt
honSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts\uvicorn.exe main:app --reloa
d
INFO:     Will watch for changes in these directories: ['C:\\Users\\admin\\Desktop\\College\\Finalyear_project\\ChatBotP
roject(Hotel)\\backend']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [23392] using WatchFiles
INFO:     Started server process [21668]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     74.125.212.173:0 - "POST / HTTP/1.1" 200 OK
INFO:     74.125.212.172:0 - "POST / HTTP/1.1" 200 OK
Order item inserted successfully!
Order item inserted successfully!
Order item inserted successfully!
INFO:     74.125.212.163:0 - "POST / HTTP/1.1" 200 OK
```

Figure 2: Uvicorn and Fastapi

NGROK:

Ngrok is a popular tunneling and reverse proxy service that allows you to expose a local server running on your machine to the public internet. It creates a secure tunnel to your local development environment, making it accessible from anywhere.developers often use Ngrok for various purposes, such as testing webhooks, sharing work-in-progress websites, and providing temporary public access to local applications for demonstrations or testing. To use Ngrok, you typically download and install their client, then run it with the appropriate command to expose a specific port on your local machine to the Ngrok servers, which then provide a publicly accessible URL for your local service. This can be very useful for web development and testing.During chatbot development, developers often run the chatbot's backend code on their local machines for testing and debugging. Ngrok allows them to expose this locally hosted chatbot to the public internet by creating a secure tunnel.Chatbots typically rely onwebhooks to communicate with external services and retrieve data. Ngrok provides a publicly accessible URL for your local webhook, allowing you to test how your chatbot interacts with these external services in a real-world scenario.Chatbots often need to integrate with third-party services, APIs, or databases. Ngrok makes it easy to test these integrations in a development environment without exposing your local machine directly to the internet.Ngrok enables developers to make code changes and immediately see the results in their chatbot, speeding up the development and testing process.It facilitates debugging by allowing you to inspect the incoming and outgoing HTTP requests and responses in real-time. This is particularly valuable when troubleshooting issues with your chatbot's communication with external services.Using Ngrok, you can avoid the need to deploy your chatbot to a production server for testing and development, which can save time and resources.Ngrok provides a secure and encrypted tunnel, ensuring that data transferred between your local development environment and external services is protected. Ngrok provides a temporary public URL that you can share with other team members or stakeholders for testing and demonstration purposes without deploying your chatbot to a production environment.Ngrok is a valuable tool for chatbot development because it simplifies the process of exposing your locally hosted chatbot to the internet, making it easier to test, integrate, and debug your chatbot's interactions with external services and APIs.

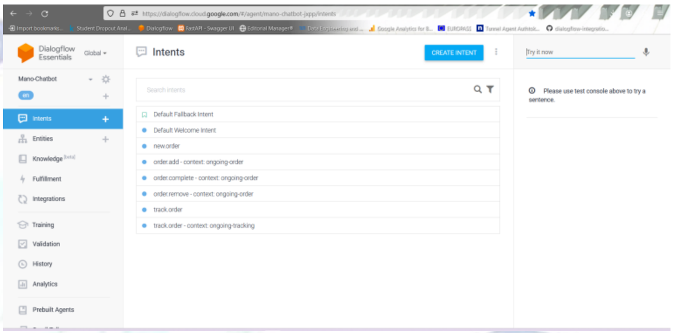


```
ngrok                                                                      (Ctrl+C to quit)

Build better APIs with ngrok. Early access: ngrok.com/early-access

Session Status                online
Account                       Manoah Jeyaprakash (Plan: Free)
Version                       3.6.0
Region                        India (in)
Latency                       -
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://74f0-118-185-105-242.ngrok-free.app -> http://localhost:8000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

Figure 3: NGROK

GOOGLE DIALOGFLOW:

Google Dialogflow is a robust cloud-based platform designed for natural language understanding (NLU), and it serves as a pivotal tool for developing chatbots and virtual assistants. It is primarily utilized to empower chatbots with the ability to comprehend and respond to human language effectively. By leveraging Dialogflow, developers can create conversational applications that interact with users through natural language, offering a seamless and intuitive user experience. Its utility lies in simplifying the development of chatbots by providing features such as intent recognition, entity extraction, and context management. This enables chatbots to understand user queries, retrieve relevant information, and generate contextually relevant responses. Furthermore, Dialogflow offers the capability to integrate with various messaging platforms, making it an ideal choice for developing chatbots that can communicate across different channels. Its importance is further highlighted by its automatic documentation generation, which streamlines the process of building and maintaining chatbots, and its support for webhook fulfillment, enabling chatbots to execute custom actions and external API calls.In summary, Google Dialogflow is a versatile and efficient tool for chatbot development, enabling developers to build intelligent, conversational agents that can engage with users in a human-like manner, making it good.



Intents

- Default Fallback Intent
- Default Welcome Intent
- new.order
- order.add - context: ongoing-order
- order.complete - context: ongoing-order
- order.remove - context: ongoing-order
- track.order
- track.order - context: ongoing-tracking

Please use test console above to try a sentence.

Figure 4: Google Dialogflow.

5. SYSTEM ARCHITECTURE:

The The proposed system architecture leverages Google Dialogflow, a state-of-the-art natural language processing platform, to redefine the food ordering experience in the digital age. At its core, the architecture incorporates smart dialogue management to facilitate a seamless interaction between users and the ordering system. The system is designed with a user-friendly interface that harnesses the capabilities of intelligent technologies to streamline and enhance the food ordering process. The architecture comprises multiple layers, including a front-end interface for users to interact with, a middleware layer integrating Google Dialogflow for natural language understanding, and a back-end system handling order processing and fulfillment. The communication flow within the system ensures efficient and real-time interactions, allowing users to place orders effortlessly. Additionally, the architecture is scalable and adaptable, capable of accommodating future advancements in natural language processing and intelligent dialogue management technologies, ensuring a robust and sustainable solution for transforming traditional food ordering methods in the digital era.

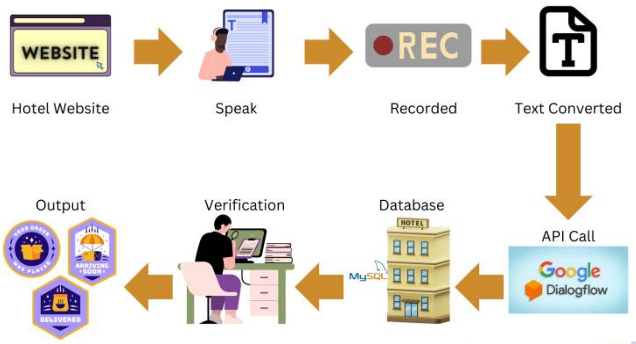


Figure 5:System Design

Creating a seamless food ordering system involves a structured integration of various technologies. Google Dialogflow serves as the natural language processing backbone, allowing users to interact with the system in a conversational manner. Intents are defined for key actions such as order placement, customization, and order tracking. Ngrok comes into play to expose the local FastAPI server to the web, establishing a communication bridge between Dialogflow and the backend. FastAPI, a high-performance web framework, forms the backend logic. Routes are implemented for essential functions like order processing and customization, with dependencies managing database interactions.SQL Workbench serves as the database management tool, facilitating the creation of a robust relational database schema. Tables are designed to store crucial information about food items, orders, and users, with relationships ensuring data integrity. In the FastAPI backend, ORM manages database interactions, enabling the implementation of functions for order insertion, retrieval, and status updates.Testing involves rigorous interaction with the Dialogflow agent via the Dialogflow Console, where requests and responses can be observed. For local development testing, Ngrok's console becomes instrumental in debugging and ensuring the effective communication between Dialogflow and FastAPI.Deployment marks the transition to a production environment, with the FastAPI backend hosted on a server or cloud platform. The fulfillment URL in Dialogflow is updated to point to this deployed endpoint, ensuring a seamless transition from development to a live environment.Potential enhancements include implementing user authentication for personalized experiences, introducing features like real-time order tracking. By following these steps, a comprehensive food ordering system is realized, leveraging the strengths of Google Dialogflow, Ngrok, FastAPI, and SQL Workbench.

6. IMPLEMENTATION

To integrate Google Dialogflow with SQL Workbench for the backend, FastAPI for server-side development, ngrok for tunneling, and HTML/CSS for the website, you can create a comprehensive system for a conversational food ordering application. In this implementation, Google Dialogflow acts as the natural language processing interface, SQL Workbench handles the database interactions, FastAPI serves as the backend server, and ngrok provides a secure tunnel for exposing the local server to the internet. Additionally, HTML and CSS are utilized to create a user-friendly website.

The system begins by setting up a database, let's call it "food_orders," with a table named "food_items" to store menu items. SQL Workbench is used to manage and interact with this database. FastAPI is then employed to create a backend server, with endpoints to handle Dialogflow webhook requests. The server integrates with SQL Workbench to perform operations such as inserting, updating, and deleting records in the "food_items" table based on user interactions.

Ngrok is utilized to expose the local FastAPI server to the internet securely, allowing Dialogflow to send webhook requests to the public endpoint. The ngrok URL is configured in the Dialogflow fulfillment settings.

For the front end, HTML and CSS are used to design a user interface for the food ordering website. The website communicates with the FastAPI backend, triggering interactions with Dialogflow when users place orders or inquire about menu items. The integration ensures a seamless and interactive experience, allowing users to engage in natural language conversations with the chatbot powered by Dialogflow while the backend manages and updates the database through SQL Workbench. This holistic approach provides a comprehensive solution for a modern and efficient food ordering system, integrating the power of Dialogflow with a robust backend and a user-friendly web interface.

The experimental discussion of the proposed food ordering system, integrating Google Dialogflow, Ngrok, FastAPI backend, and SQL Workbench as the database, reveals a comprehensive evaluation of the system's performance and considerations for future enhancements. The conversational experience facilitated by Dialogflow enhances user interaction, allowing natural language input for order placement and customization. Ngrok proves invaluable during local development, streamlining testing by exposing the FastAPI backend to the internet. FastAPI's high-performance capabilities contribute to the system's responsiveness, and SQL Workbench ensures a well-structured relational database for efficient data management. Challenges and considerations, including security, scalability, and the implementation of user authentication, are identified for future exploration. The discussion emphasizes the successful integration of technologies in creating a user-friendly and innovative food ordering system while providing a roadmap for further experimentation and improvement.

7. RESULTS & DISCUSSION

The results of the food ordering system project, integrating Google Dialogflow, Ngrok, FastAPI backend, and SQL Workbench, reveal a highly functional and user-friendly platform. The conversational experience achieved through Dialogflow allowed users to seamlessly place orders and customize items using natural language, enhancing overall user engagement. Ngrok proved to be a valuable asset during local development, facilitating efficient testing by exposing the FastAPI backend to the internet. FastAPI's high-performance capabilities ensured responsive interactions, contributing to a smooth conversational flow within the food ordering system.

The implementation of SQL Workbench for relational database management resulted in a structured and well-organized database schema. This approach facilitated effective data management, ensuring consistency and integrity across entities such as food items, orders, and users.

However, challenges and considerations emerged during the experimentation phase. Security concerns were identified, emphasizing the need for the implementation of HTTPS and robust security measures, especially when exposing a local server with Ngrok. Scalability was recognized as a potential challenge, prompting further evaluation of FastAPI's performance under varying loads.

Future directions for the project were identified in the discussion. Implementing user authentication and authorization is highlighted as a crucial enhancement to bolster security and enable personalized user experiences. Experimentation with real-time updates and notifications is suggested to improve user engagement, providing timely information on order statuses.

8. CONCLUSIONS

In this section,the project successfully demonstrated the integration of intelligent dialogue systems, showcasing a user-friendly food ordering system. The results highlight strengths in conversational experiences, backend performance, and structured database management. The identified challenges and future directions provide valuable insights for refining the system's security, scalability, and overall user experience. This project serves as a foundation for continued development and optimization of intelligent dialogue systems in the food ordering domain.

In conclusion, the fusion of Google Dialogflow with SQL Workbench, FastAPI, ngrok, HTML, and CSS establishes a robust and modern solution for revolutionizing the food ordering journey. Dialogflow's natural language processing harmonizes seamlessly with the backend operations powered by FastAPI and SQL Workbench, facilitating dynamic database interactions. The secure tunneling provided by ngrok ensures a safe connection between Dialogflow and the FastAPI server, while the user-friendly web interface crafted with HTML and CSS enhances the overall experience. Embracing these intelligent technologies showcases the transformative potential of technology-driven interactions in the domain of food ordering.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] A.Narechania, A. Srinivasan, and J. Stasko, ''NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries,'' IEEE Trans. Vis. Comput. Graphics, vol. 27, no. 2, pp. 369–379,Feb. 2021.

[2] Terry Winograd, Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. MIT AI Technical Report 235,February 1971.

[3] Alec Go, Richa Bhayani Lei Huang, Twitter Sentiment Classification usingDistantSupervision.http://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf retrived 2013-04-06, 2009.

[4] Tanwar; T.V.Prasad; Kamlesh Dutta," Natural language processing for hybrid knowledge representation", International Journal of Advances Paradigms, Vol 10, issue 3, 2018.

[5] Tanwar & Rai" A proposed system for opinion mining using machine learning, NLP and classifiers", IAES International Journal of ArtificialIntelligence (IJ-AI) Vol. 9, No. 4, December 2020, pp. 726 733 ISSN:2252-8938, DOI: 10.11591/ijai.v9.i4.pp726-733.

[6] Adam Westerski, Sentiment Analysis: Introduction and the State of the Art overview. 2009.

[7] S.Bisser, " Building a chat bot," Microsoft Conversational AI Platform for Developers, pp. 177–238, 202.

[8] J.Tang,Y.Luo, M.Ouzzani,G.Li, and H. Chen, ''Sevi: Speech-tovisualization through neural machine translation,'' in Proc. Int. Conf.Manage. Data, Jun. 2022, pp. 2353–2356.

[9] Milan tripathi, Sentiment Analysis of Nepali COVID19 Tweets Using NB, SVM AND LST M, Journal of Artificial Intelligence and CapsuleNetworks,Vol.03/No.03,Pages:151-168,2021.http://irojournals.com/aicn/DOI:https://doi.org/10.36548/jaicn.2021.3.001.[12]Anand,C. ”Comparison of Stock Price Prediction Models using Pre-trained Neural Networks.” Journal of Ubiquitous Computing and Communication Technologies(UCCT)3.02 (2021): 122-134.

International Journal for Modern Trends in Science and Technology
IJMTST