A Project Report

on

Assistant Food Chatbot

Submitted By

Deepti Agarwal, 2215800006

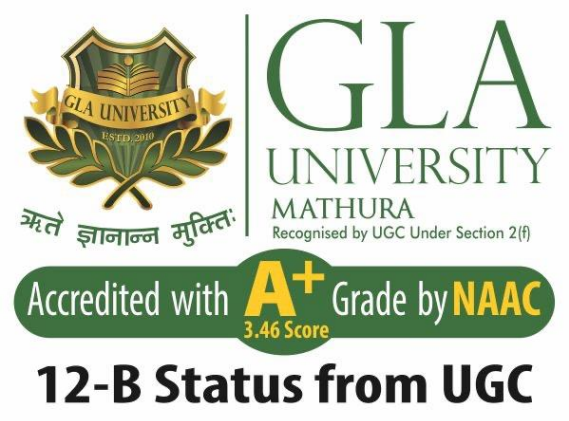
Srishti Agarwal, 2215800028

Group Number : 5

Supervisor

Dr. Sayantan Sinha

Technical Trainer



Department of Computer Engineering & Applications

Institute of Engineering & Technology

GLA University, Mathura – 281406, INDIA

Date of Submission – May 7, 2024

CONTENTS

DECLARATION

I hereby declare that the work which is being presented in the B.Tech. Project "Assistant Food Chatbot", in partial fulfillment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my own work carried under the supervision of Dr. Sayantan Sinha, Technical Trainer, Department of Computer Engineering & Applications.

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign _____

Name of Student: Deepti Agarwal

University Roll No.: 2215800006

Sign _____

Name of Student: Srishti Agarwal

University Roll No.: 2215800028

CERTIFICATE

This is to certify that the project report entitled “Assistant Food Chatbot”, submitted to the Department of Computer Science and Engineering, GLA University, in partial fulfilment for the award of the degree of Bachelor of Technology Honors in Computer Science and Engineering, is a record of bona fide work carried out by **Deepti Agarwal**, Roll No. **2215800006**, under my supervision and guidance.

_____

Supervisor

Dr. Sayantan Sinha

Technical Trainer

Dept. of Computer Engineering & Applications

Date: May 7, 2024

CERTIFICATE

This is to certify that the project report entitled “Assistant Food Chatbot”, submitted to the Department of Computer Science and Engineering, GLA University, in partial fulfilment for the award of the degree of Bachelor of Technology Honors in Computer Science and Engineering, is a record of bona fide work carried out by **Srishti Agarwal**, Roll No. **2215800028**, under my supervision and guidance.

_____

Supervisor

Dr. Sayantan Sinha

Technical Trainer

Dept. of Computer Engineering & Applications

Date: May 7, 2024

ACKNOWLEDGEMENT

We would like to extend our sincere and heartfelt thanks towards all those who have helped us in making this project. Without their active guidance, help, cooperation and encouragement, we would not have been able to present the project on time.

We extend our sincere gratitude to our trainer Dr. Sayantan Sinha for his moral support and guidance during the tenure of our project.

We also acknowledge with a deep sense of reverence, our gratitude towards our parents and otherfaculty members of the college for their valuable suggestions given to us in completing the project.

Sign _____

Name of Student: Deepti Agarwal

University Roll No.: 2215800006

Sign _____

Name of Student: Srishti Agarwal

University Roll No.: 2215800028

ABSTRACT

A ChatBot is a sophisticated software application crafted to mimic human conversation, employing text or voice interactions to engage users effectively while efficiently handling routine tasks. Developed using cutting-edge technologies, this project leverages modern programming languages such as Python and web frameworks like FastAPI for robust API construction. The development process is facilitated by Visual Studio Code, offering a seamless coding experience coupled with intuitive debugging capabilities.

The project's frontend is meticulously crafted using HTML and CSS to ensure visually captivating user interfaces. Data storage and management are entrusted to MySQL, a powerful relational database management system, ensuring the swift retrieval and storage of user interaction data.

Integrating Dialogflow, a natural language understanding platform, enriches the conversational experience by comprehending user queries and delivering pertinent information, thereby enhancing user satisfaction and interaction.

By synergizing these technologies and frameworks, the project aims to elevate user engagement and interaction. This comprehensive report delves into the project's developmental journey, elucidating its key features, functionalities, and encountered challenges. Furthermore, it explores avenues for future enhancements and expansions, paving the way for sustained growth and innovation.

LIST OF FIGURES

Figure **Page Number**

LIST OF TABLES

WORKFLOW

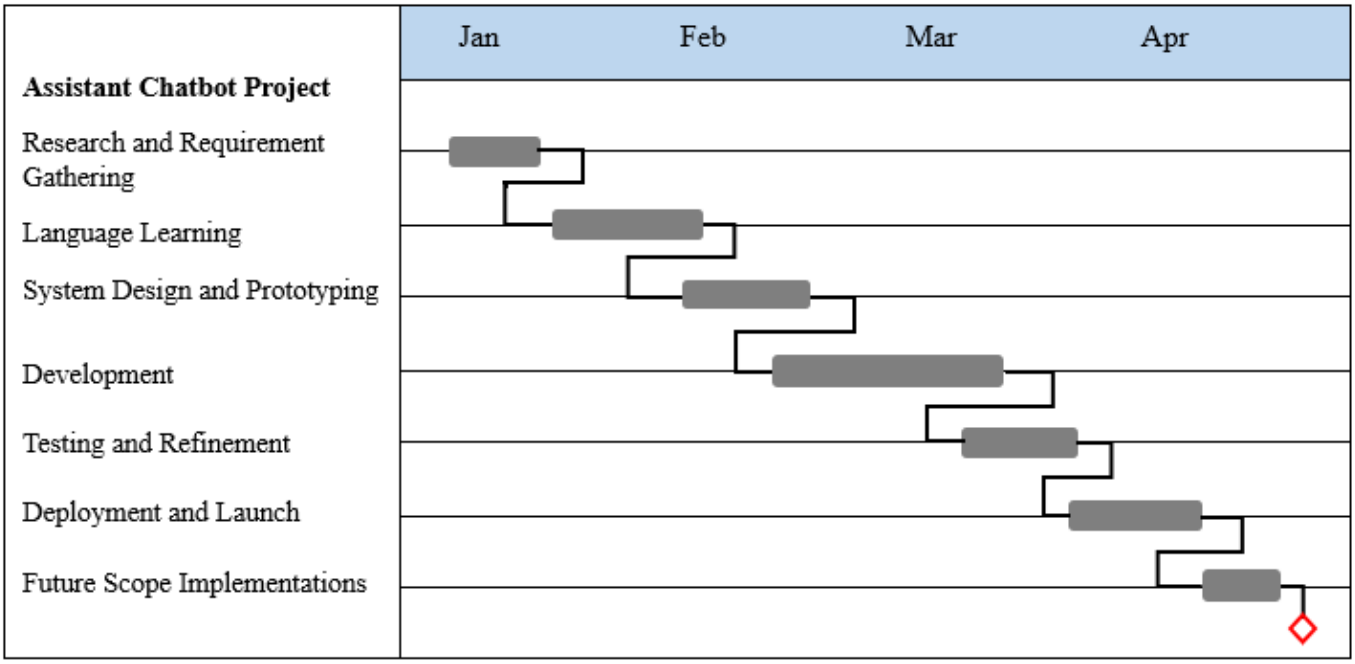


Fig 1 : Workflow Chart

CHAPTER - 1 : INTRODUCTION

1.1 Overview and Motivation

FoodieBot is a sophisticated food ordering chatbot integrated with a website, designed to streamline the process of ordering food online. Leveraging modern programming languages like Python and web frameworks such as FastAPI, FoodieBot offers users a seamless experience by allowing them to browse menus, place orders, track their order status, and calculate bills—all within the chatbot interface. With an intuitive frontend developed using HTML and CSS, and robust data storage managed through MySQL, FoodieBot ensures efficiency and reliability at every step of the ordering process. Powered by Dialogflow, a natural language understanding platform, the chatbot intelligently comprehends user queries, providing relevant information and enhancing user interaction and satisfaction[7].

The motivation behind the development of FoodieBot stems from the growing demand for convenient and efficient food ordering solutions in today's digital age. With the rise of online food delivery services, there is a need for platforms that simplify the ordering process and provide a seamless user experience. FoodieBot aims to address this need by offering a user-friendly interface that eliminates the need for users to navigate through complex websites to place their orders. By integrating modern technologies and frameworks, FoodieBot seeks to enhance user interaction and engagement, ultimately improving the overall customer experience in the realm of online food ordering[8].

1.2 Objective

The objective of FoodieBot is to redefine the food ordering experience by offering users a seamless and convenient platform to order meals online. Through the integration of advanced technologies such as natural language processing and web frameworks, FoodieBot aims to simplify the ordering process, enhance user engagement, improve efficiency, and expand accessibility. By providing a chat-based interface, users can easily browse menus, place orders, and track their order status without the need to

navigate through multiple web pages. FoodieBot also strives to create an intuitive and visually appealing user experience, encouraging frequent interaction and ensuring swift and accurate processing of orders. Accessible across various devices, including desktops, laptops, and mobile phones, FoodieBot aims to deliver a superior customer experience that sets new standards in online food ordering.

1.3 Summary of similar application

In the competitive landscape of online food ordering and delivery services, FoodieBot finds itself among several similar applications that aim to simplify and enhance the user experience. Platforms like Uber Eats, Grubhub, and ChatGourmet offer comparable functionalities, providing users with convenient ways to order food online. Uber Eats boasts an extensive restaurant network and real-time delivery tracking, while Grubhub focuses on connecting users with local restaurants through its comprehensive online marketplace. Similarly, ChatGourmet utilizes natural language processing technology to facilitate conversational food ordering, akin to FoodieBot's chat-based interface[8]. Additionally, Facebook Messenger Chatbots enable users to place orders directly within the messaging platform, leveraging AI algorithms to understand preferences and provide personalized recommendations. These similar applications share the common goal of improving user convenience and satisfaction in the realm of online food ordering, each with its unique features and strengths. Analyzing these platforms offers valuable insights for further refining FoodieBot's user experience and functionality.

1.4 Organization of the Project

The organization of the project have several phases aimed at ensuring its successful execution and fulfillment of objectives.

Starting with the research and requirement gathering[5], to understand the user needs, research on chatbot development frameworks and technologies, collecting requirements for menu integration, order tracking and billing calculations.
Following this, the process moves to language learning, learning the necessary frameworks for chatbot development.

Then comes the system design and prototyping in which we designed the website interface and data flow, prototype basic interface of the website and planning the integration of chatbot with the food ordering website[5].

Then comes the development, implementing the chatbot functionalities according to the design, integrating the chatbot with the website's backend systems for the real-time data exchange and implementing the frontend design of the website[5].
Testing is then conducted to identify and fix any bugs or issues then gathering feedback from potential users for improvements. Refining of interface and functionalities based on feedback.

Model deployment on the website, ensuring seamless integration and compatibility of the chatbot with the website, launching the chatbot and monitoring its performance.
Future scope implementations consider additional features such as providing food recommendations and continuously monitoring user feedback and make necessary improvements[5].

CHAPTER – 2 : SOFTWARE REQUIREMENT ANALYSIS

2.1 Technical Feasibility

2.1.1 Programming Languages and Frameworks: FoodieBot utilizes modern programming languages like Python and web frameworks such as FastAPI for building APIs[2]. These technologies provide a solid foundation for developing the chatbot and integrating it with the website, ensuring compatibility and scalability.

2.1.2 Frontend Development: The frontend development of FoodieBot is implemented using HTML and CSS, which are widely supported and well-established technologies for creating visually appealing user interfaces. This ensures that the chatbot interface is accessible and user-friendly across different devices and browsers.

2.1.3 Data Storage and Management: FoodieBot relies on MySQL, a relational database management system, for efficiently storing and retrieving information related to user interactions, menu items, orders, and more. MySQL is a robust and reliable solution for managing data, ensuring the smooth functioning of the chatbot.

2.1.4 Natural Language Understanding: Dialogflow, the natural language understanding platform used by FoodieBot, enables the chatbot to comprehend user queries and provide relevant responses. Dialogflow's advanced NLP capabilities enhance the user experience by allowing for more natural and intuitive interactions[1].

2.1.5 Integration and Compatibility: FoodieBot integrates seamlessly with the website, allowing users to access its features without the need for additional installations or plugins. Compatibility with popular web browsers and devices ensures that users can interact with the chatbot across various platforms.

Overall, FoodieBot demonstrates strong technical feasibility by leveraging modern technologies and frameworks to deliver a robust and user-friendly food ordering solution. The integration of these technologies ensures efficient communication, data management, and user interaction, contributing to the success of the project.

2.2 System Requirement Analysis

2.2.1 Functional Requirements

- **Menu Display:** The chatbot should be able to display the menu items available for ordering, including descriptions, prices, and images.
- **Order Placement:** Users should be able to place orders through the chatbot by selecting items from the menu and specifying quantities.
- **Order Tracking:** FoodieBot should provide users with the ability to track the status of their orders, including confirmation, preparation, and delivery stages.
- **Bill Calculation:** The chatbot should calculate the total bill based on the items selected by the user and any applicable taxes or fees.

2.2.2 Non-Functional Requirements

- **Performance:** FoodieBot should respond to user queries promptly and efficiently, with minimal latency.
- **Reliability:** The system should be reliable and available to users without frequent downtime or disruptions.
- **Scalability:** FoodieBot should be able to handle a growing number of users and orders without significant degradation in performance.
- **Security:** Measures should be implemented to ensure the security and integrity of user data, including encryption and secure authentication mechanisms.
- **Usability:** The chatbot interface should be intuitive and user-friendly, catering to users with varying levels of technical expertise.

CHAPTER – 3 : SOFTWARE DESIGN

3.1 Architecture Design

In addition to conceptualizing the architecture and components of FoodieBot, a flowchart diagram serves as a valuable tool for visually depicting the system's logic and flow of operations.

The flowchart illustrates the sequence of steps involved in various processes, such as user interaction, order processing, and data management. It offers a high-level overview of how different components of the system interact with each other and how user inputs are processed to fulfill their requests.

By presenting the system's functionality in a graphical format, the flowchart facilitates communication among stakeholders, aids in identifying potential bottlenecks or areas for optimization, and serves as a reference for developers during implementation.

Incorporating the flowchart diagram into the software design documentation enhances its comprehensibility and clarity, contributing to the successful development and deployment of FoodieBot.

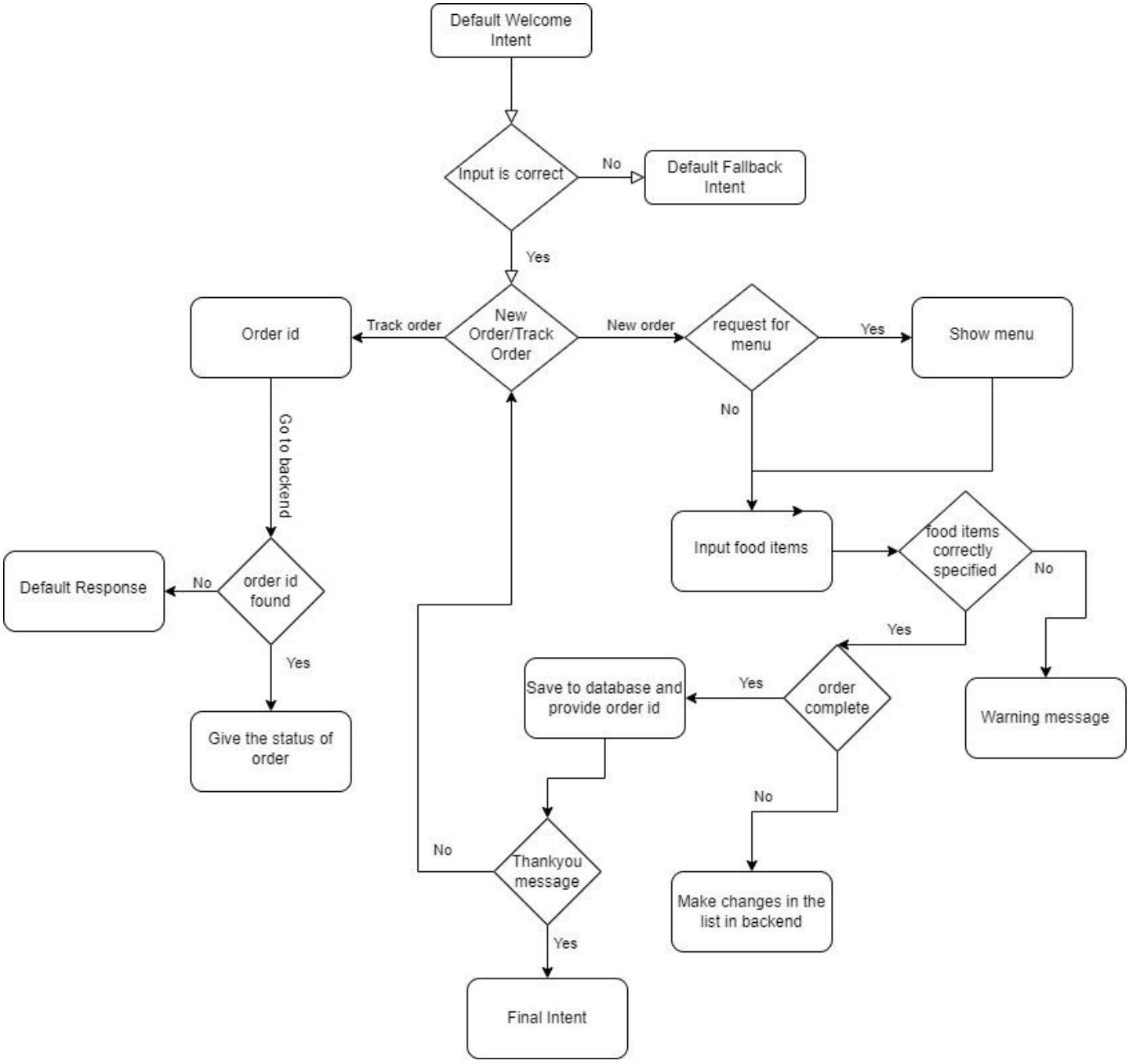


Fig-2 : Flowchart of Assistant Food Chatbot (FoodieBot)

3.2 Database Design

The database design for FoodieBot is a critical aspect of the software architecture, as it dictates how data is stored, organized, and accessed within the system. The database schema is carefully crafted to accommodate the various entities and relationships involved in the food ordering process, ensuring efficient data management and retrieval.

The database design encompasses several key considerations.

3.2.1 Entity-Relationship Model

The first step in database design involves identifying the entities (such as orders, menu items, etc.) and their relationships with each other. This is represented using entity-relationship diagrams (ERDs), which illustrate the structure of the database and the connections between different entities.

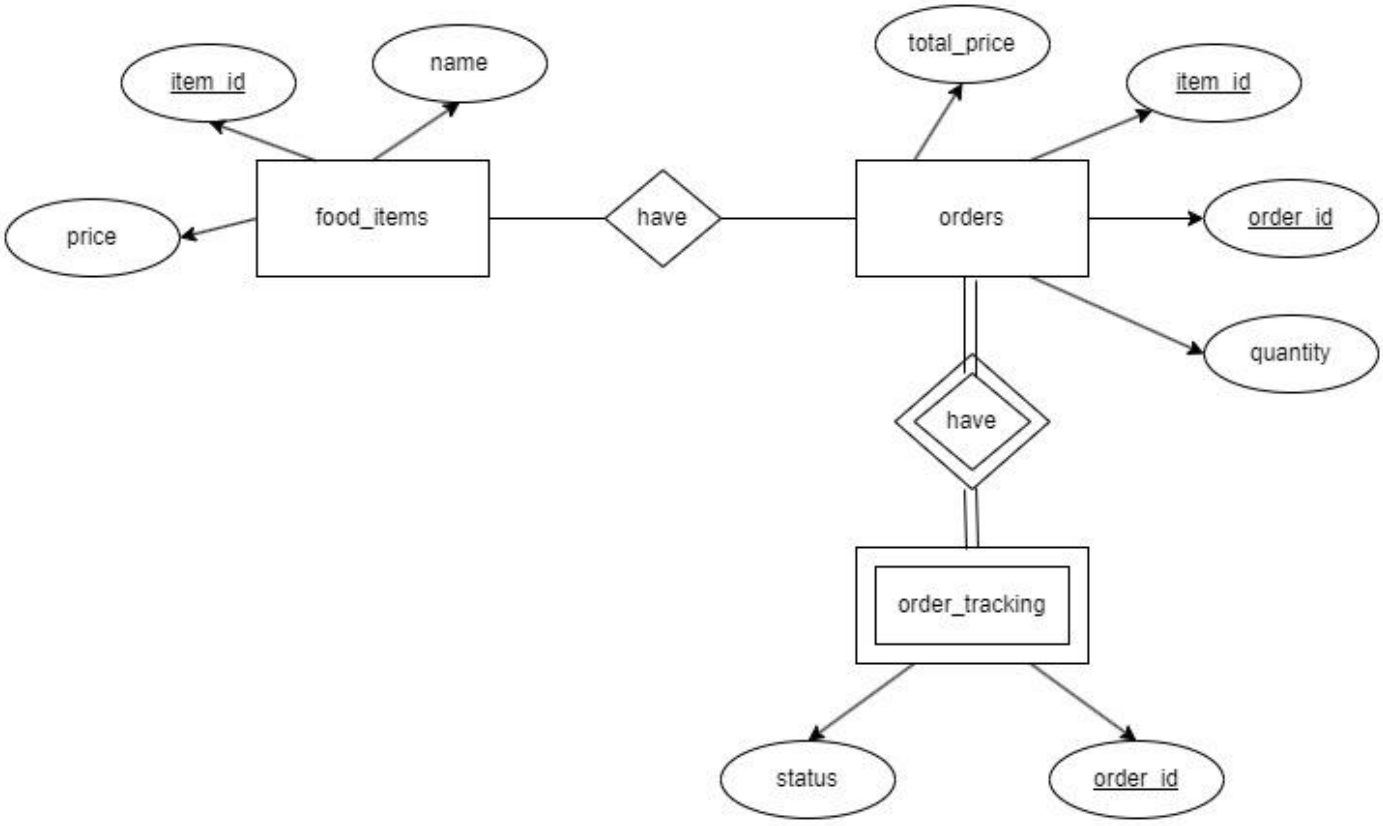


Fig-3 : ER Diagram of Database

3.2.2 Relational Model

The relational model diagram for FoodieBot illustrates the structure of the database schema, depicting the various entities and their relationships within the system. The diagram consists of tables representing the main entities involved in the food ordering process, along with their attributes and relationships.

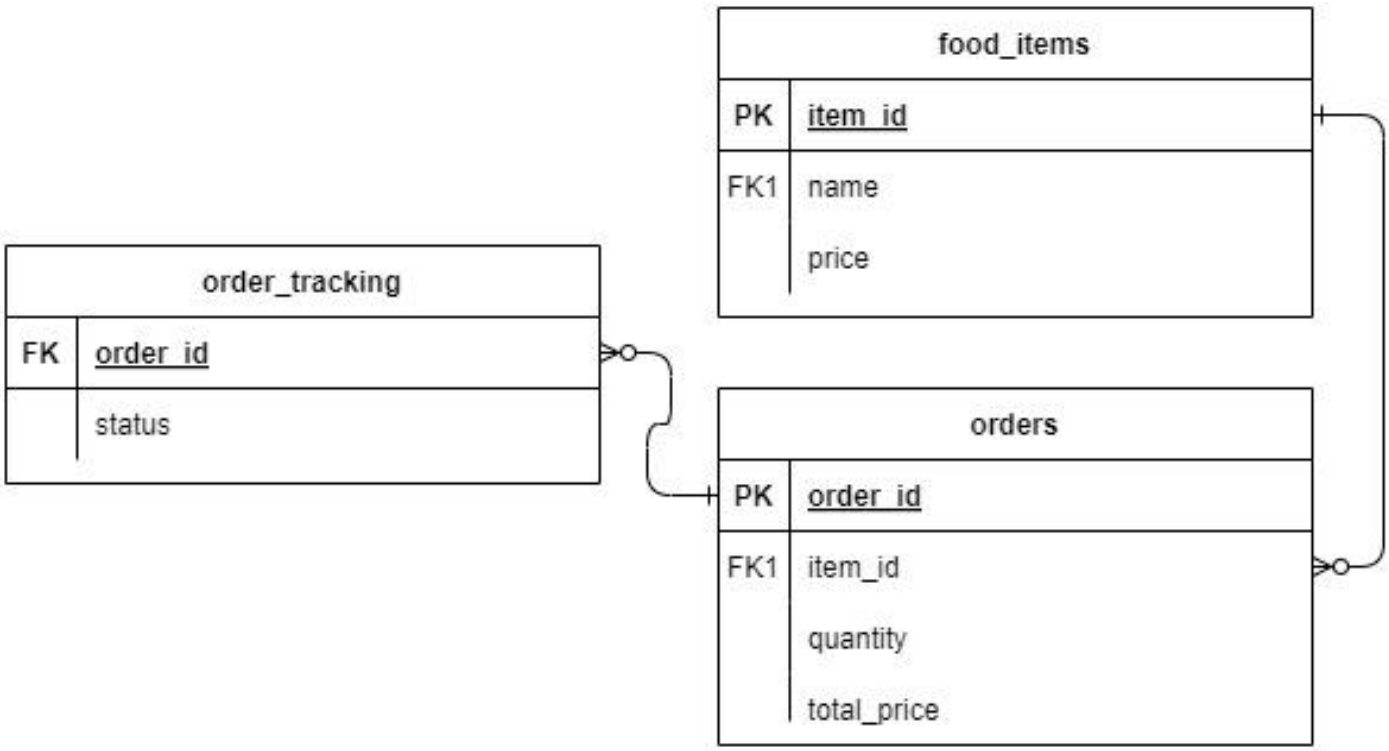


Fig-4 : Relational Model of Database

3.2.3 Data Types and Constraints

Each attribute in the database schema is assigned an appropriate data type and constraints to enforce data integrity and validity. For example, fields such as user IDs, order numbers, and menu item IDs may be defined as primary keys with unique constraints, while fields like names and descriptions may have length constraints.

item_id	name	price
1	Pasta	250.00
2	Noodles	280.00
3	Pav Bhaji	180.00
4	Chhole Bhature	200.00
5	Pizza	350.00
6	Burger	180.00
7	Coca-Cola	60.00
8	Lemon Soda	60.00
9	Chocolate Shake	220.00
10	Strawberry Shake	180.00
11	Sandwich	200.00

Table 1 : food_items

order_id	status
40	delivered
41	in transit
42	in progress
43	in progress
44	in transit
45	delivered
46	in progress
47	in progress
48	delivered
49	in progress
50	in progress

Table 2 : order_tracking

order_id	item_id	quantity	total_price
40	1	2	500.00
40	3	1	180.00
41	4	3	600.00
41	6	2	360.00
41	9	4	880.00
42	5	1	350.00
43	7	1	60.00
43	11	1	200.00
44	3	1	180.00
44	7	1	60.00
48	5	1	350.00
48	7	1	60.00

Table 3 : orders

CHAPTER – 4 : IMPLEMENTATION & USER INTERFACE

4.1 System Implementation

It involves developing the chatbot, integrating it with the website, and implementing the necessary functionalities to enable users to browse menus, place orders, track order status, and calculate bills seamlessly. Here's an overview of the system implementation process:

4.1.1 Chatbot Development

The first step in system implementation is the development of the FoodieBot chatbot. This involves coding the conversational logic using technologies such as Python and integrating it with Dialogflow for natural language understanding[1]. The chatbot is designed to respond to user queries, display menu options, process orders, and provide relevant information throughout the ordering process.

4.1.2 Website Integration

Once the chatbot is developed, it needs to be integrated with the FoodieBite website. This involves embedding the chatbot interface into the website's frontend using HTML, CSS, and JavaScript. Users should be able to access the chatbot seamlessly from the website and initiate food orders without any friction.

4.1.3 Backend Development

Concurrently with the frontend integration, the backend development work takes place. This includes setting up the server-side components using frameworks like FastAPI for handling HTTP requests[6], managing user sessions, and communicating with the database.

4.1.4 Database Implementation

The database schema designed during the software design phase is implemented using a relational database management system such as MySQL. Tables are created, and data is populated to support user registration, menu management, order processing, and other functionalities.

4.1.5 Functionality Implementation

With the chatbot, website integration, backend, and database in place, the next step is to implement the core functionalities of FoodieBot. This includes implementing features such as menu display, order placement, order tracking and bill calculation.

4.1.6 Testing and Quality Assurance

Throughout the implementation process, rigorous testing is conducted to ensure that the system functions as intended and meets the specified requirements. This includes unit testing, integration testing, and end-to-end testing to identify and address any bugs or issues.

4.1.7 Deployment

Once testing is complete and the system is deemed ready for production, it is deployed to the production environment. This involves setting up servers, configuring network settings, and deploying the application code and database to the hosting environment.

By following a systematic approach to system implementation, FoodieBot can be successfully developed and deployed, providing users with a seamless and satisfying food ordering experience.

4.2 User Interface

The user interface (UI) of FoodieBot plays a crucial role in facilitating user interactions and enhancing the overall user experience. As FoodieBot is a chatbot integrated into

the FoodieBite website, the UI primarily consists of the conversational interface through which users interact with the chatbot to browse menus, place orders, track order status, and perform other actions related to food ordering.

4.2.1 Conversational Interface

The main UI component of FoodieBot is the conversational interface, where users engage in natural language conversations with the chatbot. This interface is designed to be intuitive and user-friendly, providing users with prompts and suggestions to guide them through the ordering process. Messages are displayed in a chat-like format, allowing users to type or select options using buttons or quick replies.

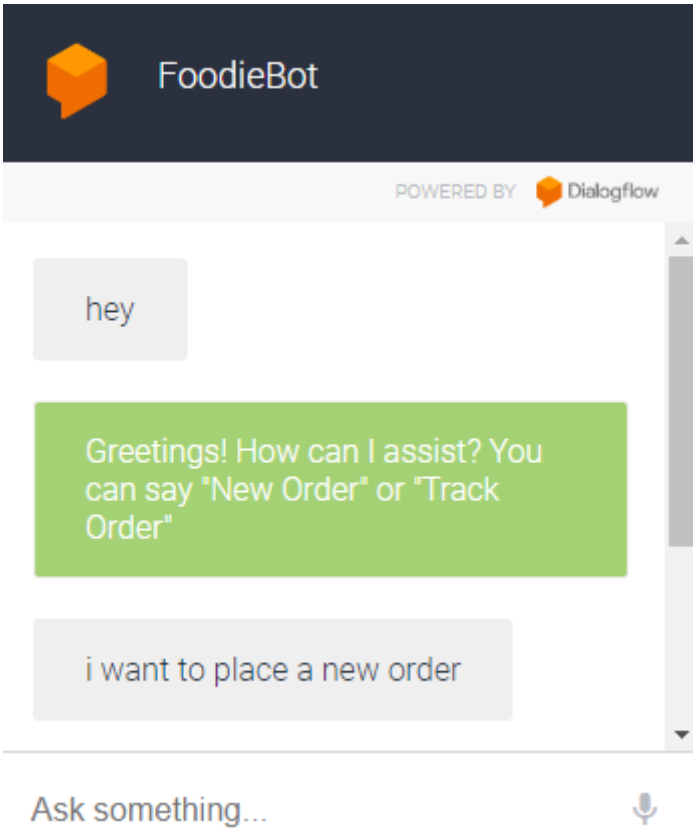


Fig 5 : FoodieBot Chatbot

4.2.2 Ordering Actions

FoodieBot's UI includes interactive elements for users to perform ordering actions such as adding items to their cart, adjusting quantities, and removing items. Users can also specify additional preferences or customizations for their orders, such as special instructions or dietary restrictions.

4.2.3 Order Tracking

Once an order is placed, FoodieBot provides users with the ability to track the status of their order directly within the chat interface. Users can receive real-time updates on order confirmation, preparation, and delivery, helping them stay informed and engaged throughout the process.

4.2.4 Responsive Design

The UI of FoodieBot is designed to be responsive and accessible across various devices and screen sizes, including desktops, laptops, tablets, and smartphones. This ensures a consistent and seamless user experience regardless of the device being used.

Overall, the user interface of FoodieBot is carefully crafted to prioritize simplicity, intuitiveness, and functionality, enabling users to effortlessly navigate the ordering process and enjoy a delightful food ordering experience through natural language conversations.

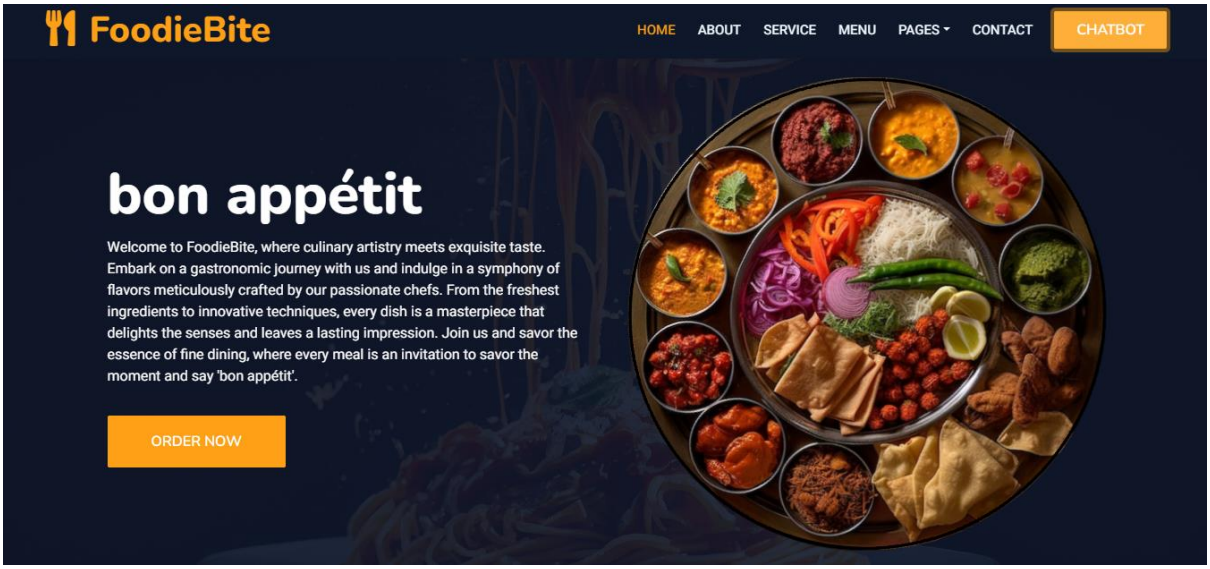


Fig 6 : Homepage of FoodieBite Website

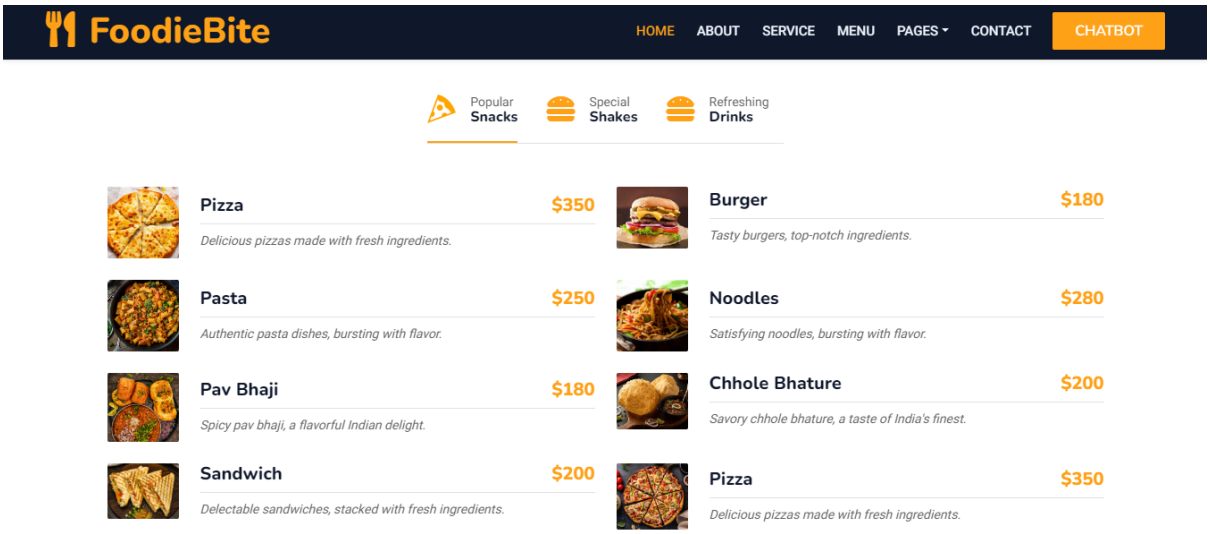


Fig 7 : Menu of FoodieBite Website

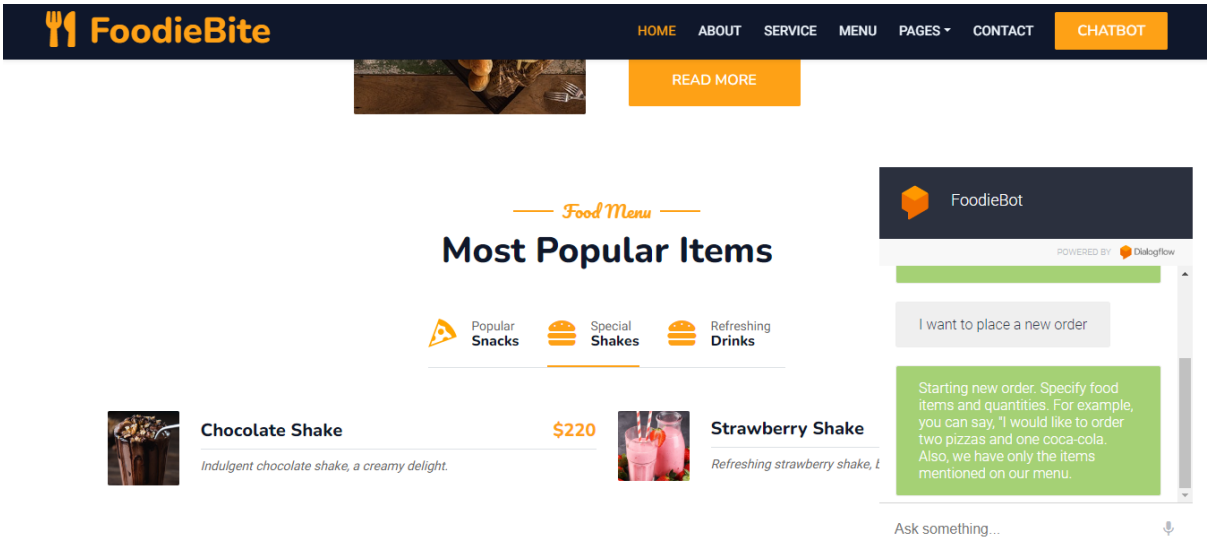


Fig 8 : FoodieBot integrated to FoodieBite Website

4.3 Technology Stack

In the development of FoodieBot, we have utilized a range of technologies, frameworks, and tools to create a seamless and efficient food ordering experience for users. Here's an overview of what we have used:

4.3.1 Programming Languages

Python was chosen as the primary programming language for developing the backend logic of FoodieBot. Its simplicity, versatility, and extensive library support make it well-suited for building chatbots and web applications.

4.3.2 Web Frameworks

We have leveraged FastAPI, a modern web framework for building APIs with Python. FastAPI provides high performance, asynchronous capabilities, and automatic validation of request parameters, making it ideal for developing the backend API endpoints of FoodieBot[2][6].

4.3.3 Frontend Development

HTML and CSS were used for frontend development to create visually appealing and responsive user interfaces for the FoodieBot chatbot embedded within the FoodieBite website. These standard web technologies ensure cross-browser compatibility and accessibility across different devices.

4.3.4 Natural Language Understanding

Dialogflow, a natural language understanding platform developed by Google, was integrated into FoodieBot to process and understand user queries. Dialogflow's advanced NLP capabilities enable FoodieBot to interpret user inputs, extract relevant information, and generate appropriate responses[1].

4.3.5 Database Management System

MySQL, a popular relational database management system, was chosen for storing and managing data related to user interactions, menu items, orders, and other aspects of the FoodieBot system. MySQL provides scalability, reliability, and robust data management capabilities essential for a food ordering application.

4.3.6 Integrated Development Environment (IDE)

Visual Studio Code (VS Code) was utilized as the primary integrated development environment for coding and debugging the backend logic of FoodieBot. Its intuitive interface, extensive plugin ecosystem, and built-in Git integration streamline the development process.

By harnessing the power of these technologies and tools, we have been able to develop a sophisticated and user-friendly food ordering chatbot that seamlessly integrates with the FoodieBite website, offering users a convenient and enjoyable experience from browsing menus to tracking their orders.

CHAPTER – 5 : SOFTWARE TESTING

Software testing is a crucial aspect of the development process for FoodieBot, ensuring that the chatbot functions as intended, meets the specified requirements, and delivers a satisfying user experience. Here's how software testing is conducted for FoodieBot:

5.1 Unit Testing

Unit testing involves testing individual components or modules of the chatbot in isolation to verify their functionality. This includes testing functions, methods, and classes to ensure they produce the expected outputs for given inputs. Python's built-in testing framework, such as unittest or pytest, can be used for unit testing.

5.2 Integration Testing

Integration testing focuses on testing the interactions between different components or modules of FoodieBot to ensure they work together seamlessly. This includes testing API endpoints, database interactions, and external integrations to verify data flow and communication between various parts of the system.

5.3 Functional Testing

Functional testing involves testing the overall functionality of FoodieBot to ensure it performs the intended tasks correctly. This includes testing user interactions such as browsing menus, placing orders, tracking order status, and calculating bills to verify that they meet the specified requirements.

5.4 User Interface (UI) Testing

UI testing involves testing the user interface of FoodieBot to ensure it is visually appealing, responsive, and easy to use. This includes testing the chatbot interface, menu displays, interactive elements, and navigation to ensure a seamless and intuitive user experience across different devices and screen sizes.

By rigorously testing FoodieBot at each stage of development, we can ensure its reliability, functionality, and user satisfaction, ultimately delivering a high-quality food ordering chatbot that meets the needs and expectations of our users.

CHAPTER – 6 : CONCLUSION

In conclusion, the development of FoodieBot represents a significant milestone in revolutionizing the food ordering experience for users. By leveraging modern technologies, intuitive user interfaces, and robust software engineering practices, we have created a sophisticated chatbot integrated into the FoodieBite website that streamlines the process of ordering food online. Throughout the development process, we have prioritized user experience, functionality, and reliability, ensuring that FoodieBot meets the needs and expectations of our users.

With its intuitive conversational interface, users can easily browse menus, place orders, track order status, and calculate bills without the need to navigate through complex websites. The integration of natural language understanding technology enables FoodieBot to comprehend user queries and provide relevant information, enhancing user interaction and satisfaction. Additionally, robust backend infrastructure, including FastAPI for building APIs and MySQL for data storage, ensures efficiency, scalability, and reliability in handling user interactions and processing orders[6].

6.1 Future Aspects

Looking ahead, there are opportunities for further enhancement and expansion of FoodieBot. This includes integrating additional features such as personalized recommendations, loyalty programs, and social sharing functionalities to further enrich the user experience. Additionally, continuous monitoring, testing, and refinement will be essential to address any issues, optimize performance, and adapt to evolving user needs and technological advancements.

Overall, FoodieBot represents a transformative solution in the realm of online food ordering, offering users a convenient, efficient, and enjoyable way to order their favorite meals with just a few simple interactions. As we continue to iterate and innovate, we remain committed to delivering a superior food ordering experience that delights our users and sets new standards in the industry.

CHAPTER – 7 : SUMMARY

In summary, FoodieBot is a cutting-edge food ordering chatbot integrated with the FoodieBite website, offering users a seamless and intuitive way to order their favorite meals online. Developed using modern technologies such as Python, FastAPI, and Dialogflow, FoodieBot boasts a user-friendly interface that simplifies the ordering process and enhances user engagement.

Through its conversational interface, users can easily browse menus, place orders, track order status, and calculate bills without the need to navigate through complex websites. With robust backend infrastructure powered by MySQL, FoodieBot ensures efficient data storage and retrieval, while its natural language understanding capabilities enable it to comprehend user queries and provide relevant information. Moving forward, there are opportunities for further enhancement and expansion of FoodieBot, including the integration of personalized recommendations and loyalty programs.

Overall, FoodieBot represents a transformative solution in the online food ordering landscape, delivering a superior user experience that sets new standards in convenience and satisfaction.

REFERENCES

1. Google. "Dialogflow Documentation." *Google Cloud*. Accessed May 6, 2024. https://cloud.google.com/dialogflow/docs.

2. Tiangolo. "FastAPI Documentation." *FastAPI*. Accessed May 6, 2024. https://fastapi.tiangolo.com/.

3. ngrok. Accessed May 6, 2024. https://ngrok.com/.

4. Simplilearn. "Understanding Natural Language Processing." *Simplilearn.* Accessed May 6, 2024. https://www.simplilearn.com/natural-language-understanding-article.

5. GeeksforGeeks. "Software Development Life Cycle (SDLC)." *GeeksforGeeks*. Accessed May 6, 2024. https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/.

6. "FastAPI Documentation." *DevDocs.* Accessed May 6, 2024. https://devdocs.io/fastapi/.

7. JavaTpoint. "Dialogflow Tutorial." *JavaTpoint*. Accessed May 6, 2024. https://www.javatpoint.com/dialogflow.

8. Saha, Dipan. "Chatbot Development Made Easy: Creating a Simple Bot with Dialogflow." *Medium.* Accessed May 6, 2024. https://medium.com/@dipan.saha/chatbot-development-made-easy-creating-a-simple-bot-with-dialogflow-ade69caac37d.