End to End NLP Based Chatbot using Dialog Flow

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

Submitted by

Pranav Kumar Agarwal (201118)

Rohit Sahni (201162)

Under the guidance & supervision of

Dr. Pankaj Dhiman

Department of Computer Science & Engineering and

Information Technology

Jaypee University of Information Technology, Waknaghat, Solan - 173234 (India)

Candidate's Declaration

I hereby declare that the work presented in this report entitled **'End to End NLP Based Chatbot using Dialog flow'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to June 2024 under the supervision of **Dr. Pankaj Dhiman** (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)
Student Name: Pranav Kumar Agarwal, Rohit Sahni
Roll No.: 201118, 201162

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Supervisor Name: Dr. Pankaj Dhiman
Designation: Assistant Professor (SG)
Department: Computer Science & Engineering and Information Technology

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to all those who have played a crucial role in the successful completion of this major project. This endeavor has been a culmination of dedication, collaboration, and relentless effort from a diverse group of individuals, each contributing their unique skills and perspectives.

First and foremost, I express my deepest appreciation to my project supervisor **Dr. Pankaj Dhiman**, whose guidance and expertise have been invaluable throughout every stage of this project. Their mentorship provided the necessary direction and insights, shaping the project into its final form.

I extend my appreciation to Jaypee University of Information Technology for providing the necessary resources and a conducive environment for the project's execution.

Finally, my sincere thanks go to my family and friends for their unwavering support, understanding, and patience during the challenging phases of this project. Their encouragement provided the emotional sustenance needed to navigate the complexities of such a substantial undertaking.

In conclusion, the successful completion of this major project stands as a testament to the collective effort and dedication of all those mentioned above. It is with deep appreciation that I acknowledge their contributions, and I look forward to the continued success and collaboration in future endeavors.

TABLE OF CONTEXT

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This major project explores the development of an End-to-End Natural Language Processing (NLP) Based Chatbot using Google's Dialogflow platform. With the rapid advancement in conversational AI technologies, the project aims to leverage the capabilities of Dialogflow to create a sophisticated and user-friendly chatbot capable of understanding and responding to natural language input seamlessly. The project encompasses the entire chatbot development lifecycle, from designing intuitive conversational flows and defining user intents to integrating backend logic for fulfilling user requests.

The chatbot's architecture incorporates Dialogflow's powerful NLP engine, enabling it to comprehend user intents and extract relevant entities with high accuracy. The project delves into the customization of intents and entities to suit specific use cases, ensuring a tailored and contextually-aware conversational experience. Emphasis is placed on creating a multi-platform chatbot, allowing integration with diverse messaging and voice platforms, thereby enhancing accessibility for users.

Furthermore, the project explores advanced features of Dialogflow, such as context management and fulfillment, to enhance the chatbot's ability to maintain meaningful conversations and execute backend processes seamlessly. The incorporation of machine learning techniques facilitates continuous improvement in the chatbot's language understanding and response generation through iterative training.

The development process adheres to best practices in security and compliance, ensuring the protection of user data and adherence to privacy regulations. Extensive testing methodologies are employed to validate the chatbot's performance across various scenarios, ensuring robustness and reliability in real-world applications.

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Chatbot is a computer program that helps humans to interact with natural spoken language and it also includes artificial intelligence such as Natural Language Processing that makes the chatbot more interactive and humans can depend on them more.

Based on the most recent situation of the epidemic the growth of chatbots increased and at that time the electronic education faced a major difficulty to reach the universities due to the curfew imposed and they had very limited access to the information of the academic of the university.

This project aims to build a Chatbot for Food Delivery and answers to each and every inquiry of the individual.

1.2 Problem Statement

At the opening of new restaurants, the major problem faced by the employees is to handle the customer and the chatbot is the effective way for handling the customer queries and take their feedback. The workload often slows down the ability of the team to address the queries of the customer. We identify the importance of providing a flawless service and continuously enhance the chatbot capabilities to ensure a quick response, thereby keeping in mind that no user is unintentionally ignored during the major peak order time span.

1.3. OBJECTIVE

- Save effort and time for both customer as well as the staff
- Provide a detailed information about the restaurant
- Easy access to information

1.4. Significance And Motivation of the Project Work

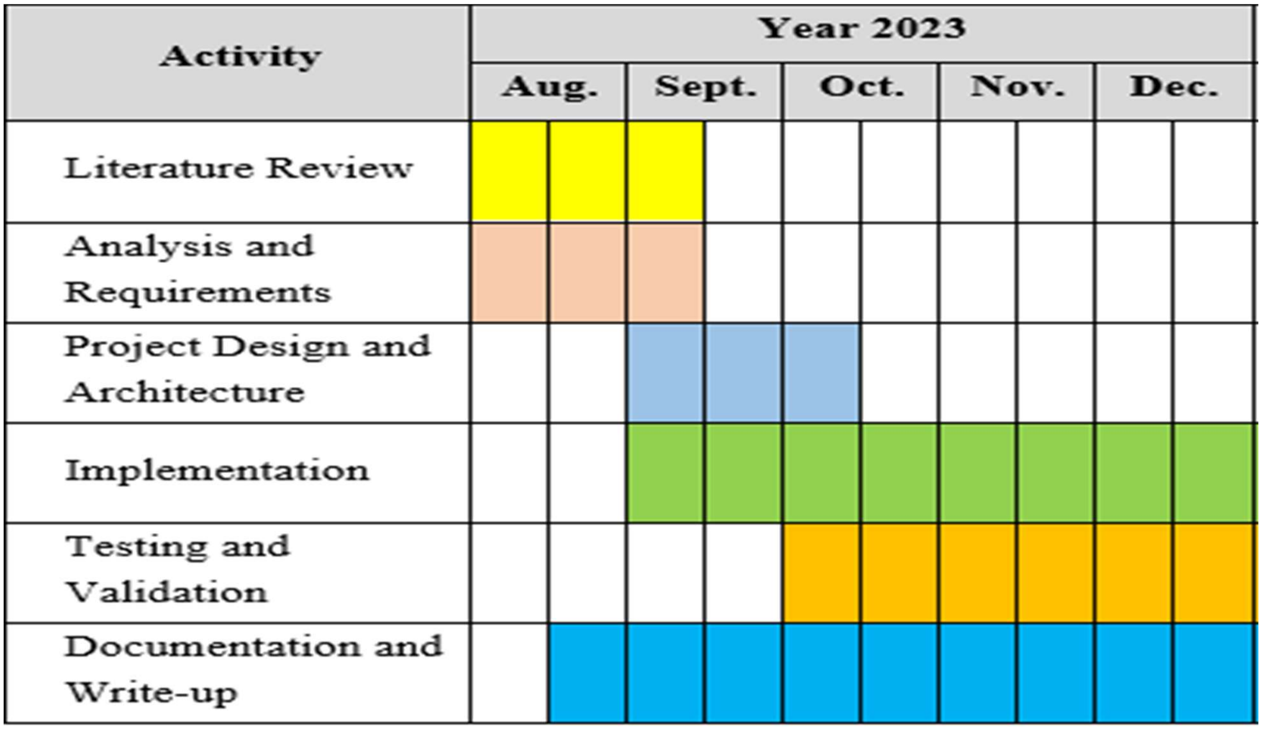
- **RELEVANCE IN CONVERSATIONAL AI:**
  1. **Customer Service**
  2. **Educational Department**
  3. **Health Care sector**
  4. **Food Department**

As there were so many diverse fields in which the AI and cloud can be used motivated us to build a project for one of them . India is a country with a diverse number of food options available that inspired more of us to carry forward this project and take it to the final step.

1.5 TIMELINE

NUMBER OF TASKS	TASK NAME	REQUIRED TIME IN WEEK
1	LITERATURE REVIEW	6
2	ANALYSIS AND REQUIREMENTS	6
3	PROJECT DESIGN AND ARCHITECTURE	6
4	IMPLEMENTATION	12
5	TESTING AND VALIDATION	12
6	DOCUMENTATION AND WRITE - UP	10

TABLE 1.1 TIMELINE



Activity	Year 2023									
	Aug.		Sept.		Oct.		Nov.		Dec.	
Literature Review	🟨	🟨	🟨							
Analysis and Requirements	■	■	■							
Project Design and Architecture			🟦	🟦	🟦					
Implementation			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Testing and Validation					🟧	🟧	🟧	🟧	🟧	🟧
Documentation and Write-up		🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦

Figure 1.1(TIMELINE DIVISION)

CHAPTER -2 LITERATURE SURVEY

S.No.	Paper	Title	Tools and Technologies	Results
1	Shang, Rui, et al.	A Literature Survey on End-to-End Neural Chatbots	-	Comprehensive overview of end-to-end neural chatbots
2	Duong, Nguyen Viet, and Dang Cong Khanh	A Survey of Chatbot Systems and Research	-	State-of-the-art chatbot systems and research
3	Serban, Iulian V., et al.	A Survey of Dialogflow Applications	Dialogflow	Applications of Dialogflow, a popular chatbot development platform
4	Shah, Ujjawal, et al.	A Survey of Natural Language Processing for Chatbots	Natural language processing (NLP) techniques	Natural language processing (NLP) for chatbots
5	Li, Zhou, et al.	A Survey of Open-Domain Chatbots	Open-domain chatbots	Open-domain chatbots, which are chatbots that can converse on a wide range of topics
6	Serban, Iulian V., et al.	Chatbot Research and Applications: A Bibliometric Analysis	Bibliometric analysis tools	Bibliometric analysis of chatbot research
7	Serban, Iulian V., et al.	Neural Conversation Modeling: A Review	Neural networks	Neural conversation modeling, a subfield of NLP that focuses on developing neural networks that can generate human-quality conversational responses
8	Henderson, Matthew, et al.	The Google Dialogflow Agent: Conversational AI for Mobile and Web Apps	Google Dialogflow Agent	Google Dialogflow Agent, a conversational AI platform that enables developers to build chatbots for mobile and web apps

9	Sun, Sicheng, et al.	A Survey of End-to-End Neural Conversational Systems	End-to-end neural conversational systems	End-to-end neural conversational systems, which are systems that learn to generate conversational responses from scratch without relying on hand-crafted rules or features
10	Li, Zhou, et al.	A Survey of Deep Reinforcement Learning for Dialogue Systems	Deep reinforcement learning (DRL) techniques	Deep reinforcement learning (DRL) for dialogue systems

2.1 Overview of Literature Survey

The literature survey on end-to-end NLP-based chatbots using Dialogflow provides a comprehensive overview of the current state of the art in chatbot development. The survey covers a wide range of topics, including the architecture of end-to-end neural chatbots, training methods, evaluation metrics, and applications. The survey also discusses the challenges and opportunities of developing end-to-end neural chatbots.

Key Gaps of Literature Survey :

One key gap of the literature survey is that it does not provide a detailed discussion of the different end-to-end neural chatbot architectures that have been proposed. The survey also does not discuss the different training methods that have been used for end-to-end neural chatbots. Additionally, the survey does not provide a comprehensive evaluation of the different end-to-end neural chatbots that have been developed.

Overall, the literature survey on end-to-end NLP-based chatbots using Dialogflow is a valuable resource for researchers and practitioners who are interested in developing chatbots.

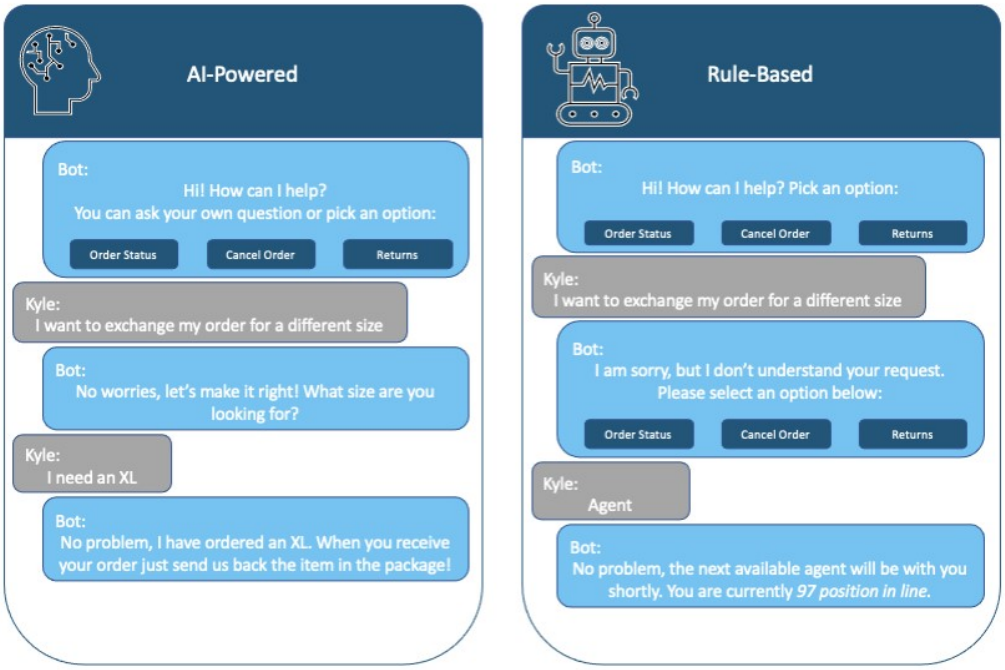
CHAPTER 3: SYSTEM DEVELOPMENT

3.1 REQUIREMENT AND ANALYSIS:

3.1.1 TYPES OF CHATBOTS

Chatbots can be of various types based on their functionality and what they are capable of ,Chatbots can be classified into following parts :

1.Rule Based Chatbot: Those chatbots that follow a certain set of rules and are best for the straight forward actions. Rule based chatbots follow a flow that is divided into certain branch-like questions . Majority of online platforms follow ruled based chatbots , they train chatbots with the answers and the pre defined rules and statements.The certain difference between AI chatbots and Rule Based Chatbots can be clearly seen in the below mentioned figures.



AI-Powered

Bot:
Hi! How can I help?
You can ask your own question or pick an option:

Kyle:
I want to exchange my order for a different size

Bot:
No worries, let's make it right! What size are you looking for?

Kyle:
I need an XL

Bot:
No problem, I have ordered an XL. When you receive your order just send us back the item in the package!

Rule-Based

Bot:
Hi! How can I help? Pick an option:

Kyle:
I want to exchange my order for a different size

Bot:
I am sorry, but I don't understand your request.
Please select an option below:

Kyle:
Agent

Bot:
No problem, the next available agent will be with you shortly. You are currently *97 position in line.*

Figure 3.1.1.1(AI vs Ruled Based Chatbots)

The basic Flow diagram of Ruled Based Chatbot is shown below :

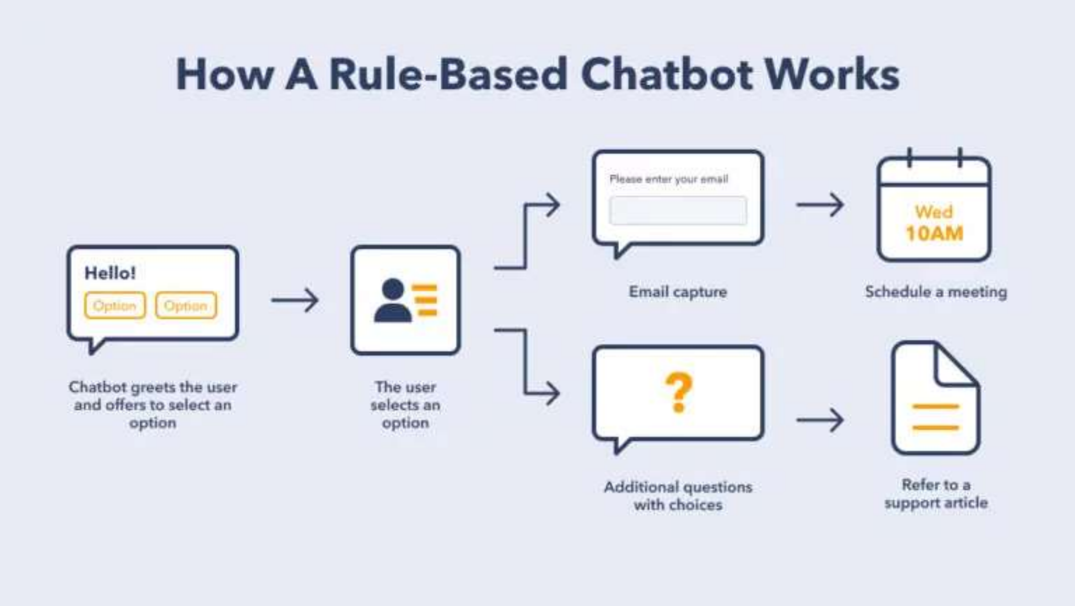

How A Rule-Based Chatbot Works
Hello!
Option
Option
Chatbot greets the user and offers to select an option
The user selects an option
Please enter your email
Email capture
Wed 10AM
Schedule a meeting
?
Additional questions with choices
Refer to a support article

Figure 3.1.1.2 (Flow diagram of Ruled Based Chatbots)

2.AI Powered Chatbots : Those chatbots that use Machine Learning as well as Artificial Intelligence to generate responses are called AI Powered Chatbots . These types of chatbots answer the queries more easily and in a time efficient manner as compared to other chatbots , refer to figure 3.1.1.1 for clear demonstration of the difference between AI powered chatbots and other chatbots .

The flow diagram of AI powered chatbots are clearly mentioned below :

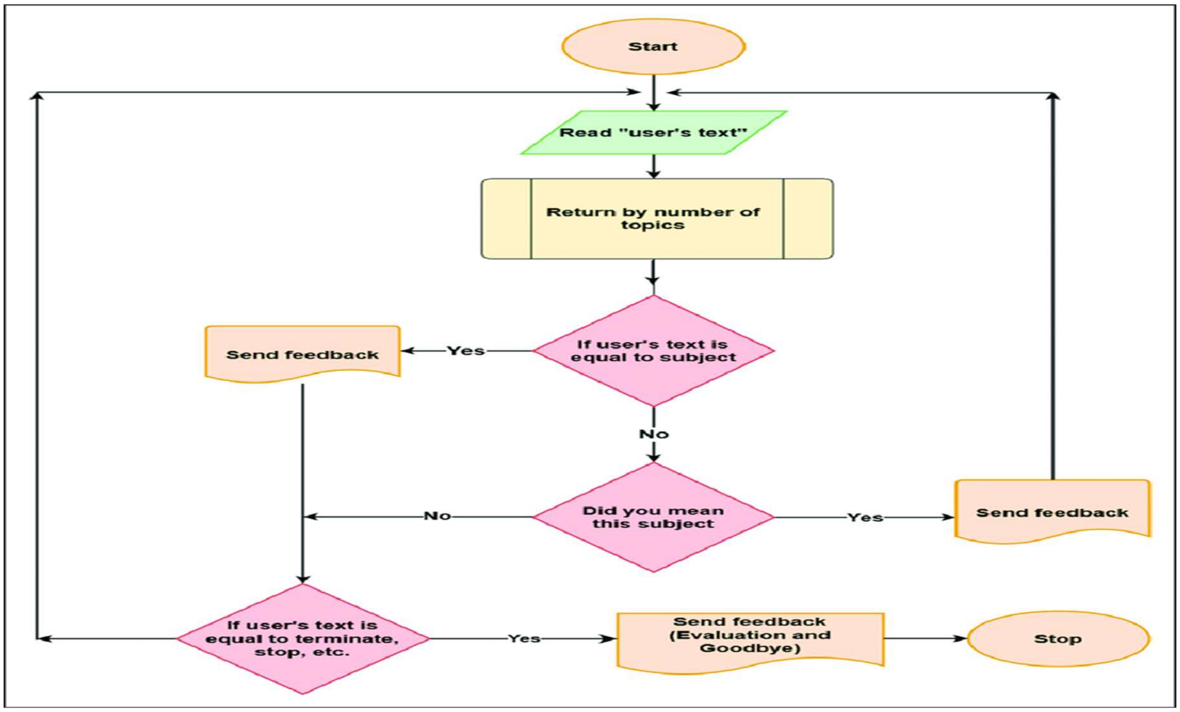

Start
Read "user's text"
Return by number of topics
If user's text is equal to subject
Yes
Send feedback
No
Did you mean this subject
No
Yes
Send feedback
If user's text is equal to terminate, stop, etc.
Yes
Send feedback (Evaluation and Goodbye)
Stop

Figure 3.1.1.3(Flow diagram of AI Powered Chatbots)

3.Scripted Chatbots: Those chatbots that work on pre-defined scripts and they help the users through a certain set of rules and path of interactions. These chat-bots are “Common-Based”, these bots understand predefined commands. The example of scripted chatbots is Square 2. Square 2 is a 24*7 web service, if the customer support is inactive the chatbot works at that moment of time.

Below mentioned is a common example of a scripted chatbot :

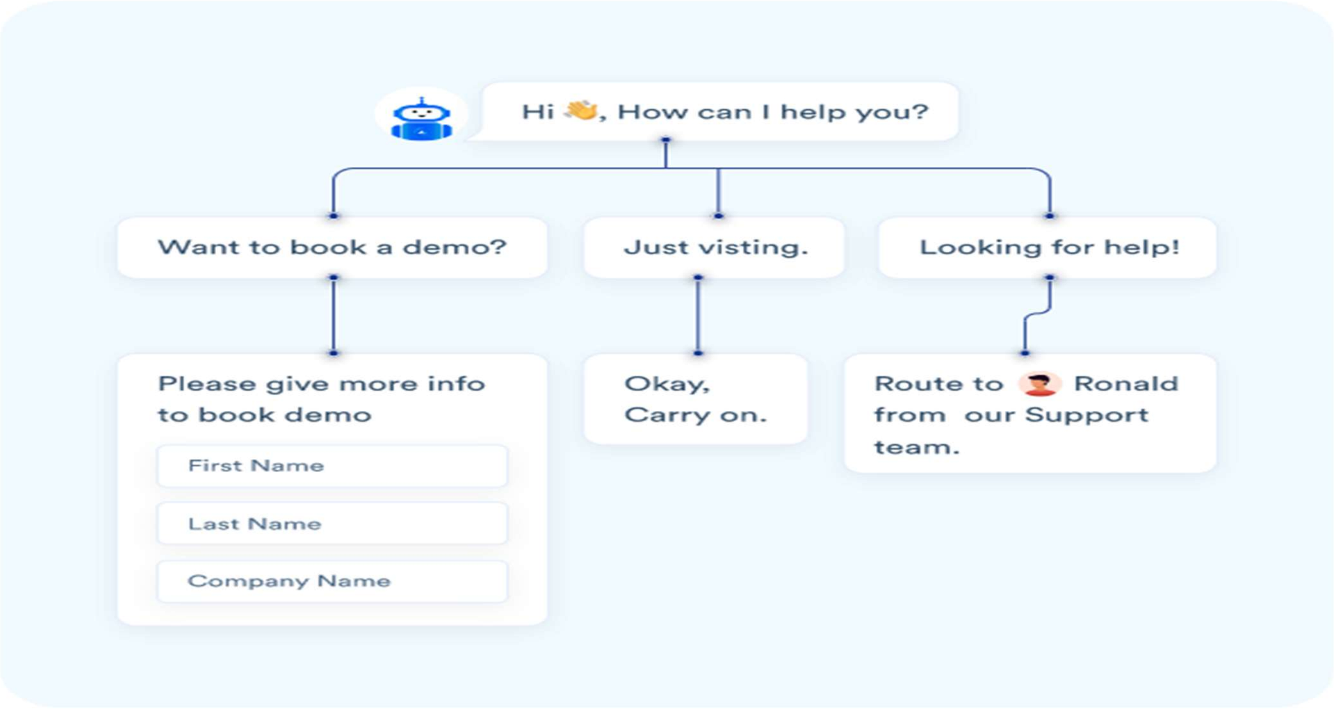

Hi 👋, How can I help you?
Want to book a demo?
Just visting.
Looking for help!
Please give more info to book demo
First Name
Last Name
Company Name
Okay, Carry on.
Route to Ronald from our Support team.

Figure 3.1.1.4(Example of Scripted Chatbot)

3.1.2 HOW DO CHATBOT WORK?

As described in the definition, chatbots connect with humans.
Basically there are 2 way by which chatbots can interact :

a.Text
b.Voice

Now describing how the chatbot uses both of the ways :

A. TEXT :

Every chatbot has intents , intents are the predefined data that are fed to the chatbot for the response . Every time whenever the data input comes the chatbot analyzes the input and then matches the input with the intents.The intent matching totally depends on user i/p as the number of i/p by the user increases the intent matching automatically increases.

B. VOICE :

Some of the chatbots are designed using API's that are able to handle the human voice and then convert the human voice into textual data . After conversion of human voice to textual data then the same process goes on of the intent matching.

The major thing about the i/p and o/p is based on the user , the i/p or the intent that has the highest matching that in machine learning is known as the “highest scoring intent” has the confidence score > than the minimum settled value then it is known as the Match of the i/p and that intent that has the highest confidence score is set as the o/p.

The flow diagram to describe how intent matching is done in a chatbot is explained below:

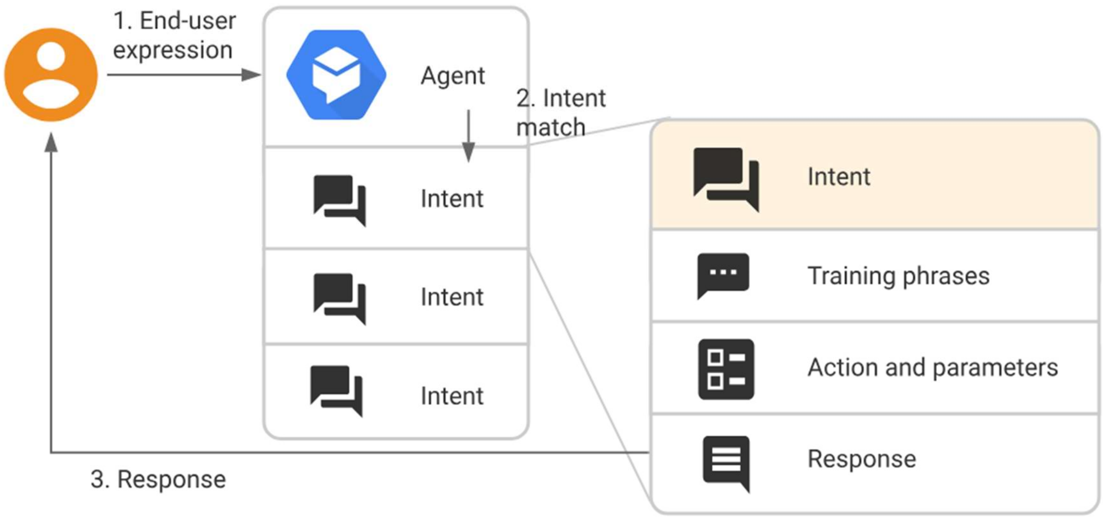


Figure 3.1.2.1(Intent Matching Process)

3.1.3 OPTIONS TO BUILD A CHATBOT

Introducing this project we need to build this chatbot for that we have two options either to build the chatbot from scratch or to choose the pre-built platforms .

A. Building Chatbot from scratch :

Now the first task is to identify the number of options available for building the chatbot and identify what is the scope to achieve the efficiency and the accuracy. For selecting the chatbot we need to examine the requirements of the customers so that it can solve the operational challenges.

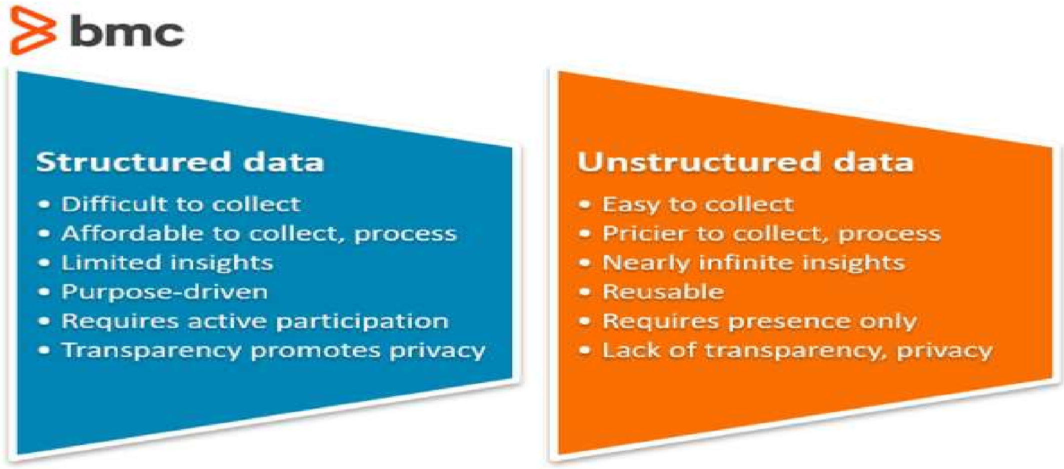
Now we can categorize the interaction of the chatbot as structured interaction and unstructured interactions .

Now understanding about what is structured interactions and unstructured interactions :

Structured Interactions :

In structured interactions we already know what the customer will be asking and then we can easily design it – this type of interaction is more like frequently asked questions from our website or the app designed . This will be linked to the order information , tracking order and else.

Unstructured Interactions :
In unstructured interactions the flow includes the plain text . In unstructured Interaction the queries are hard to guess and it looks more like a plain speech for the chatbot. AI plays a very crucial role for decoding this plain text . Natural Language processing analyzes the text for extracting useful meaning from it , voice is also provided by the same nlp.



Structured data

- Difficult to collect
- Affordable to collect, process
- Limited insights
- Purpose-driven
- Requires active participation
- Transparency promotes privacy

Unstructured data

- Easy to collect
- Pricier to collect, process
- Nearly infinite insights
- Reusable
- Requires presence only
- Lack of transparency, privacy

For the later decisions to build the chatbot we have to be well versed with the knowledge of programming languages , artificial intelligence and machine learning.

To understand much about the difference between the structured and unstructured interaction below is image :

Figure 3.1.3.1(Difference b/w unstructured And Structured Data)

B. Building chatbot using pre-defined Platforms :

Building a chatbot using a platform is much similar to building a chatbot from scratch , the difference being we need to have the knowledge of the platform to build it . The predefined platforms include Botsify, chatfuel , Rasa and much more ,it is not much have or impossible to get with . The naturally processed chatbots can deal with unstructured data.

3.1.4 CHATBOT PLATFORMS ALTERNATIVES

We have much other alternatives to build a chatbot few of them are :

a. Google Dialogflow
b. Rasa
c. IBM Watson

Now explaining each of the platforms :

a. Google Dialogflow :
   Theoretically we have a bot running up that has basically 3 core concepts on which it works : intent , dialog control and entities .These concepts are followed by a variety of chatbots that we play with in our day to day activities.
b. Rasa :
   Rasa is an open source tool that is totally equipped with Natural Language processing .This open source tool is known as the Rasa NLU. In Rasa we can generally customize the machine learning algorithm and we get the desired result. We can run it anywhere , where we want and none of our data is sent to Google , Amazon and Facebook to train the chatbot.
c. IBM Watson :
   IBM Watson is basically a question-and-answer platform that is basically used to design the general applications and the chatbot. Watson creates an application that helps us to establish a general connection between the chatbot and the user. IBM platform is very easy to use and does not require any coding prior knowledge at first sight. Also it can be integrated to a number of other applications very easily such as : Social Media Applications and much more.

3.1.5 SELECTED PLATFORM

For this project we will be using the Google Dialogflow for the chatbot , but why did we choose dialogflow?

Will , dialogflow has a lot of advantages such as :

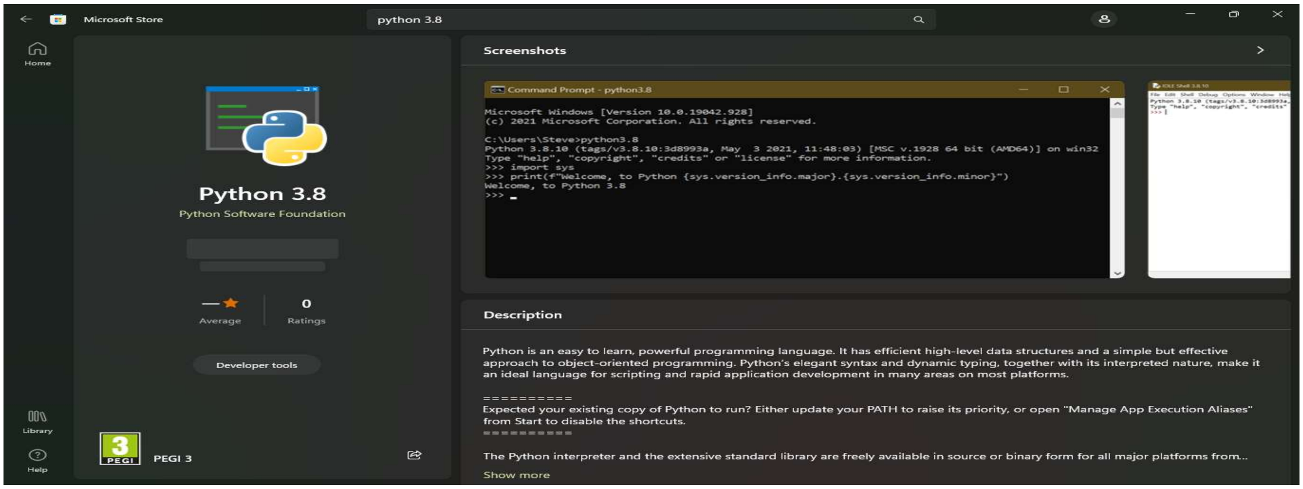
- It understands the i/p in a conversational manner and also it utilizes the NLP to understand the text
- It works without any flaw at any platform such as mobile apps, websites , messaging platforms etc.
- It provides a variety of templates that help in saving a lot of time and effort.
- It maintains a context through the whole conversation that allows the chatbot to be more accurate and relevant.
- It is a cost-efficient solution that makes this platform a best source for us to use in this project.
- It is much more flexible as compared to other cloud services and helps us to integrate anywhere.
- It allows multiple languages to be supported at a time
- The flexibility of the platform makes it accessible globally world-wide.

Apart from the above-mentioned advantages dialogflow has a series of disadvantages:

- The overall pricing structure of using this platform can be very expensive as well as complex and that makes it much challenging for the cost estimation.
- Dialogflow is a google provided service , as it is a google provided service it is the main reason of concern for the user who are looking for an independent platform for the building of the chatbot.
- Like every other cloud service the privacy problem is also a main concern in google dialogflow as well
- Dialogflow has a limited number of functionalities that are available offline to the users.
- Dialogflow has a limited number of voice functionality. By voice functionality we mean that google dialogflow can recognize a listed number of voice languages that can be translated to textual language through NLP.
- There are only a few options available for the deployment of the chatbot made using Dialogflow, it lags in the flexibility as compared to other frameworks.

3.1.6 HOW TO SETUP DIALOG FLOW AND START A DEMO CHATBOT

- Google Dialogflow is a google cloud platform used to set up chatbots as an easy way. It requires pre-knowledge of Machine Learning such as intents, entities .
- For setting up the backend we need to have prior knowledge of python and Restful Api's.
    - So first we should install python ,the required version should be 3.6 to 3.12, 3.8 should be the maximum version.
    - For setting up python first go to Microsoft store and search for the required version of python i.e. python 3.12. The next step is to check if the python environment is configured already .



```
Microsoft Windows [Version 10.0.19042.928]
(c) 2021 Microsoft Corporation. All rights reserved.

C:\Users\Steve>python3.8
Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> print(f"Welcome, to Python {sys.version_info.major}.{sys.version_info.minor}")
Welcome, to Python 3.8
>>> _
```

Description

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

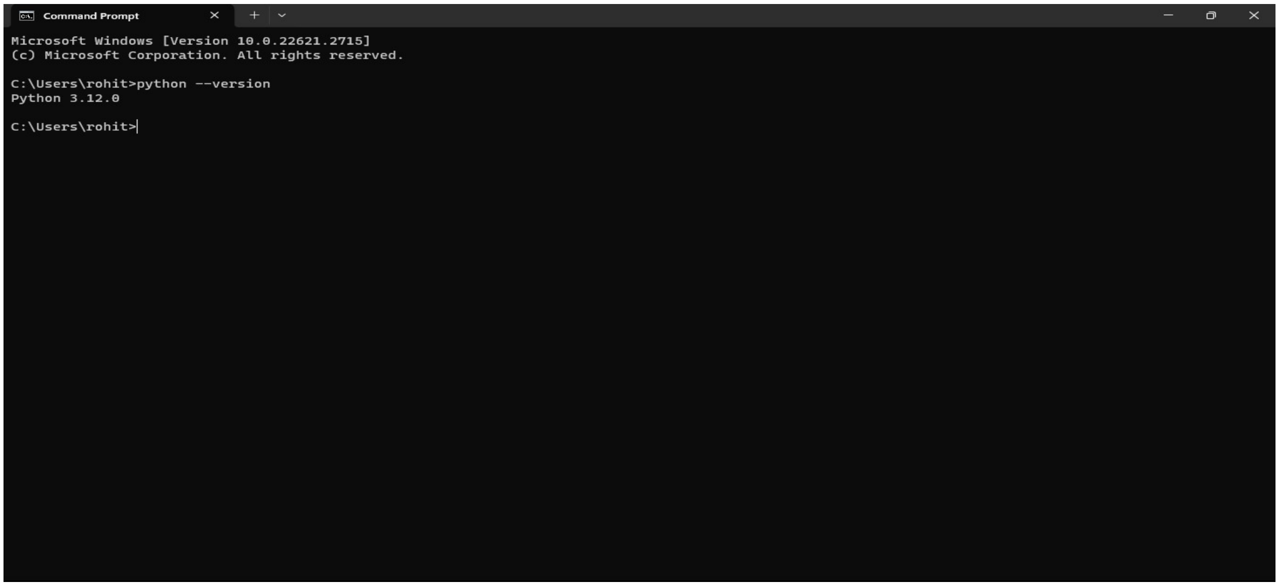
==========
Expected your existing copy of Python to run? Either update your PATH to raise its priority, or open "Manage App Execution Aliases" from Start to disable the shortcuts.
==========

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from...

Show more

Figure 3.1.6.1(Setting up of Python)

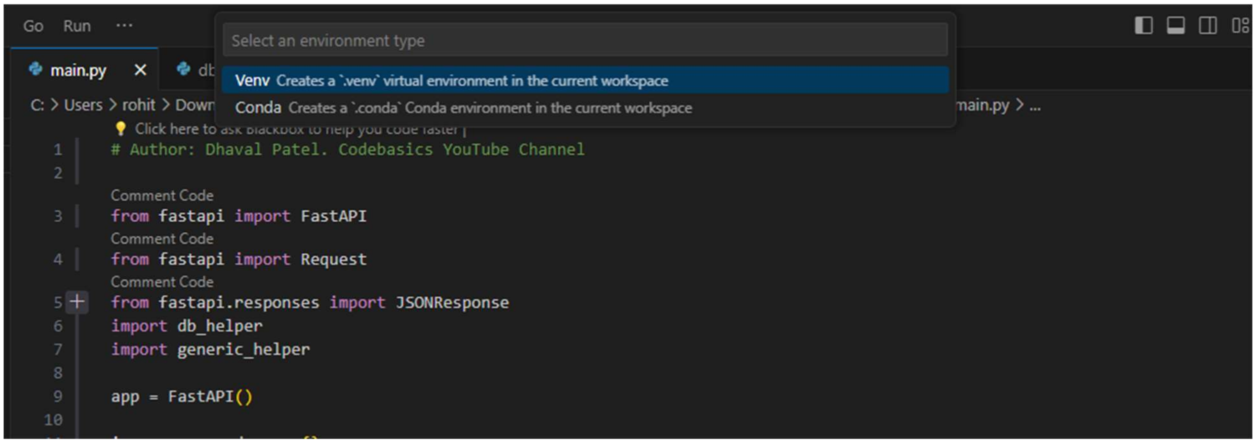
    - It can be checked by typing python3—version.



```
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rohit>python --version
Python 3.12.0

C:\Users\rohit>
```

Figure 3.1.6.2(Version Checking of Python)

- If python version is not found then we need to setup virtual environment for the starting of python
- If the required version of python is not found then we can use the following command: pip3 install -U pip.
  - Setting up of the virtual environment
    - Virtual environment can be created using the following command : “python3 -m venv ./venv” .



Go
Run
···
main.py
db
C: > Users > rohit > Down
main.py > ...
Click here to ask Blackbox to help you code faster
Comment Code

Select an environment type

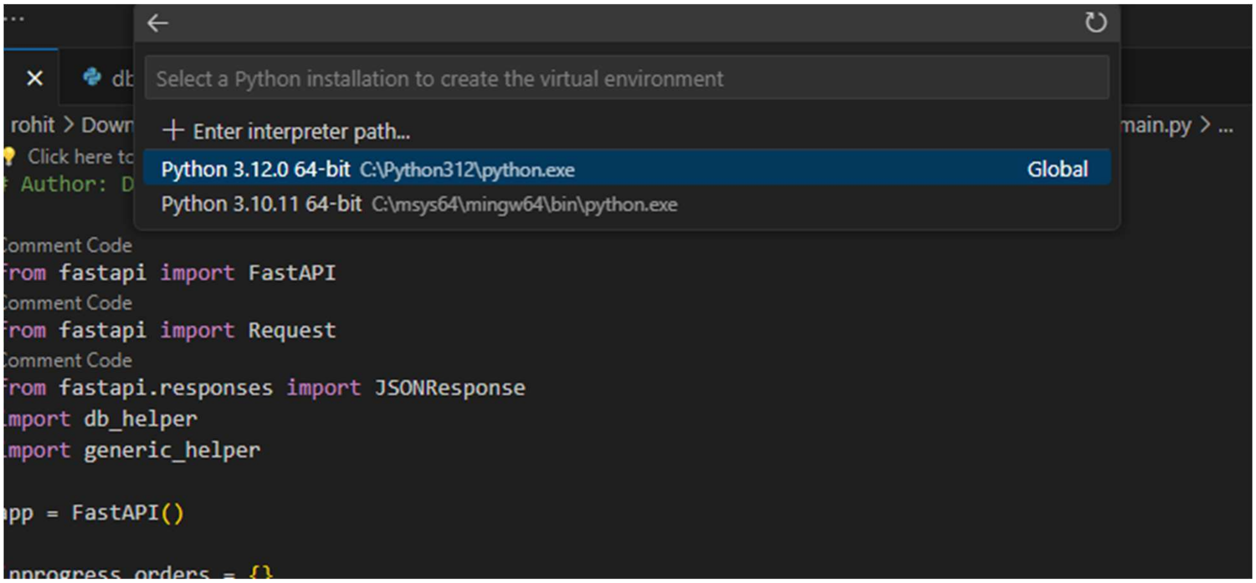
Venv Creates a `.venv` virtual environment in the current workspace

Conda Creates a `.conda` Conda environment in the current workspace

```
# Author: Dhaval Patel. Codebasics YouTube Channel

from fastapi import FastAPI
from fastapi import Request
from fastapi.responses import JSONResponse
import db_helper
import generic_helper

app = FastAPI()
```

Figure 3.1.6.3(Creating Virtual Environment)

- To activate the virtual environment we use the following command : “.\venv\Scripts\activate” .



Select a Python installation to create the virtual environment

+ Enter interpreter path...

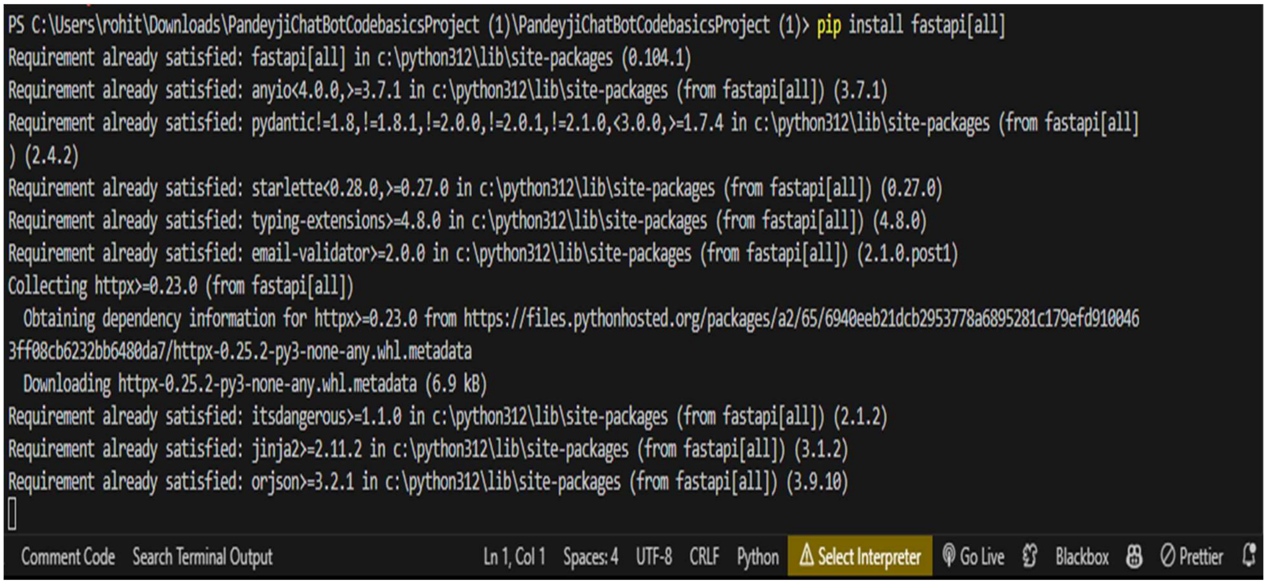
Python 3.12.0 64-bit C:\Python312\python.exe Global

Python 3.10.11 64-bit C:\msys64\mingw64\bin\python.exe

```
Author: D
omment Code
rom fastapi import FastAPI
omment Code
rom fastapi import Request
omment Code
rom fastapi.responses import JSONResponse
mport db_helper
mport generic_helper

pp = FastAPI()

nprogress_orders = {}
```

Figure 3.1.6.4(Selection of Python –version)

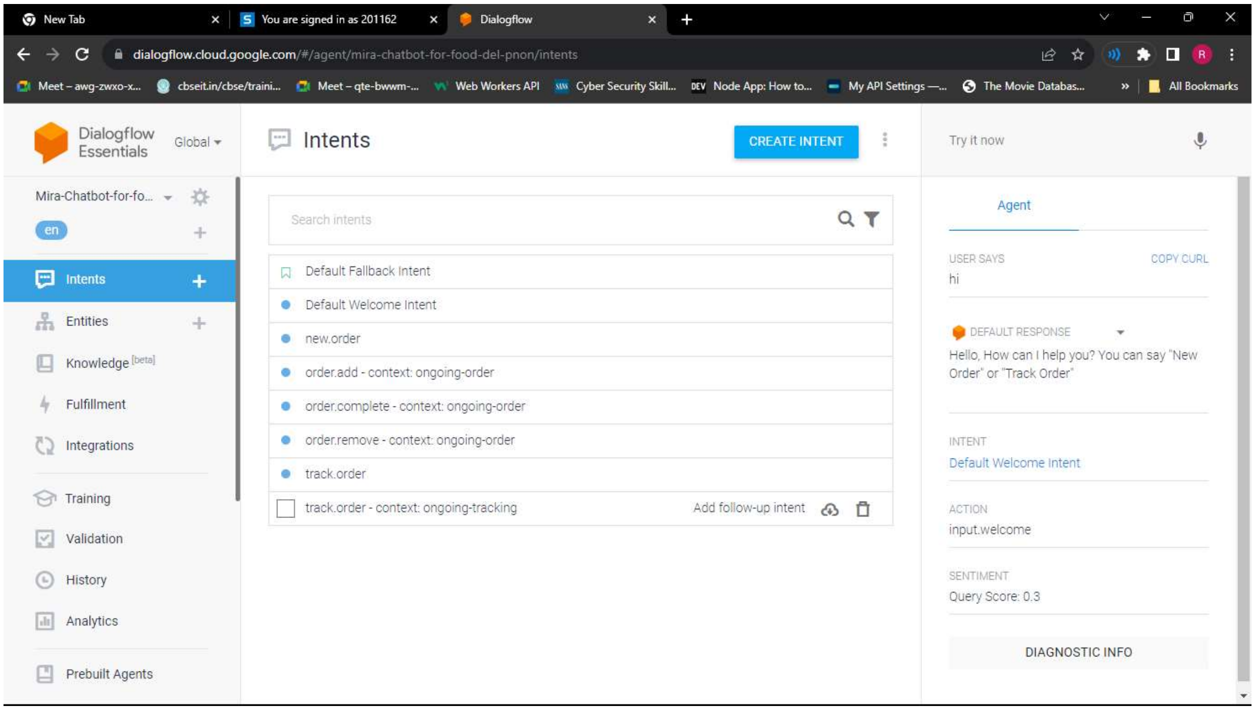
- Install the fastapi using the following command : “pip install fastapi[all]” .



```
PS C:\Users\rohit\Downloads\PandeyjiChatBotCodebasicsProject (1)\PandeyjiChatBotCodebasicsProject (1)> pip install fastapi[all]
Requirement already satisfied: fastapi[all] in c:\python312\lib\site-packages (0.104.1)
Requirement already satisfied: anyio<4.0.0,>=3.7.1 in c:\python312\lib\site-packages (from fastapi[all]) (3.7.1)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4 in c:\python312\lib\site-packages (from fastapi[all]
) (2.4.2)
Requirement already satisfied: starlette<0.28.0,>=0.27.0 in c:\python312\lib\site-packages (from fastapi[all]) (0.27.0)
Requirement already satisfied: typing-extensions>=4.8.0 in c:\python312\lib\site-packages (from fastapi[all]) (4.8.0)
Requirement already satisfied: email-validator>=2.0.0 in c:\python312\lib\site-packages (from fastapi[all]) (2.1.0.post1)
Collecting httpx>=0.23.0 (from fastapi[all])
  Obtaining dependency information for httpx>=0.23.0 from https://files.pythonhosted.org/packages/a2/65/6940eeb21dcb2953778a6895281c179efd910046
3ff08cb6232bb6480da7/httpx-0.25.2-py3-none-any.whl.metadata
  Downloading httpx-0.25.2-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: itsdangerous>=1.1.0 in c:\python312\lib\site-packages (from fastapi[all]) (2.1.2)
Requirement already satisfied: jinja2>=2.11.2 in c:\python312\lib\site-packages (from fastapi[all]) (3.1.2)
Requirement already satisfied: orjson>=3.2.1 in c:\python312\lib\site-packages (from fastapi[all]) (3.9.10)
```

Figure 3.1.6.5(Installation of FAST API)

- Create a new project with the name as Mira Chatbot that would be the name of our chatbot ,while setting up the chatbot dialogflow has some predefined welcome intents that would not be changed in our case . The welcome intents would be the same for the chatbot Mira as well , below mentioned is an example of welcome intent and its response .



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

USER SAYS
hi

DEFAULT RESPONSE
Hello, How can I help you? You can say "New Order" or "Track Order"

INTENT
Default Welcome Intent

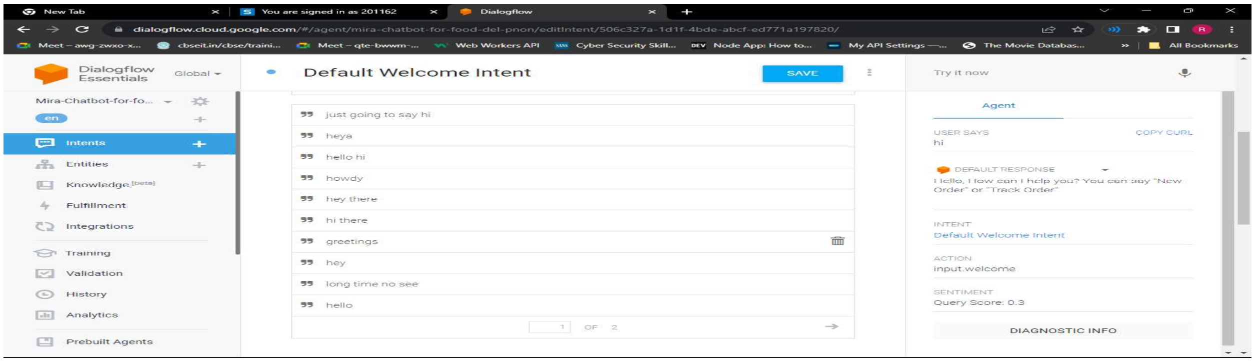
ACTION
input.welcome

SENTIMENT
Query Score: 0.3

Figure 3.1.6.6 (Number of Intents)

The intents are divided into mainly 2 phases :

a.Training Phase : That part of the intent in which we feed what user can give the chatbot as input parameter. A simple example of the training phase is shown below.



Default Welcome Intent

- just going to say hi
- heya
- hello hi
- howdy
- hey there
- hi there
- greetings
- hey
- long time no see
- hello

USER SAYS
hi

DEFAULT RESPONSE
Hello, How can I help you? You can say "New Order" or "Track Order"

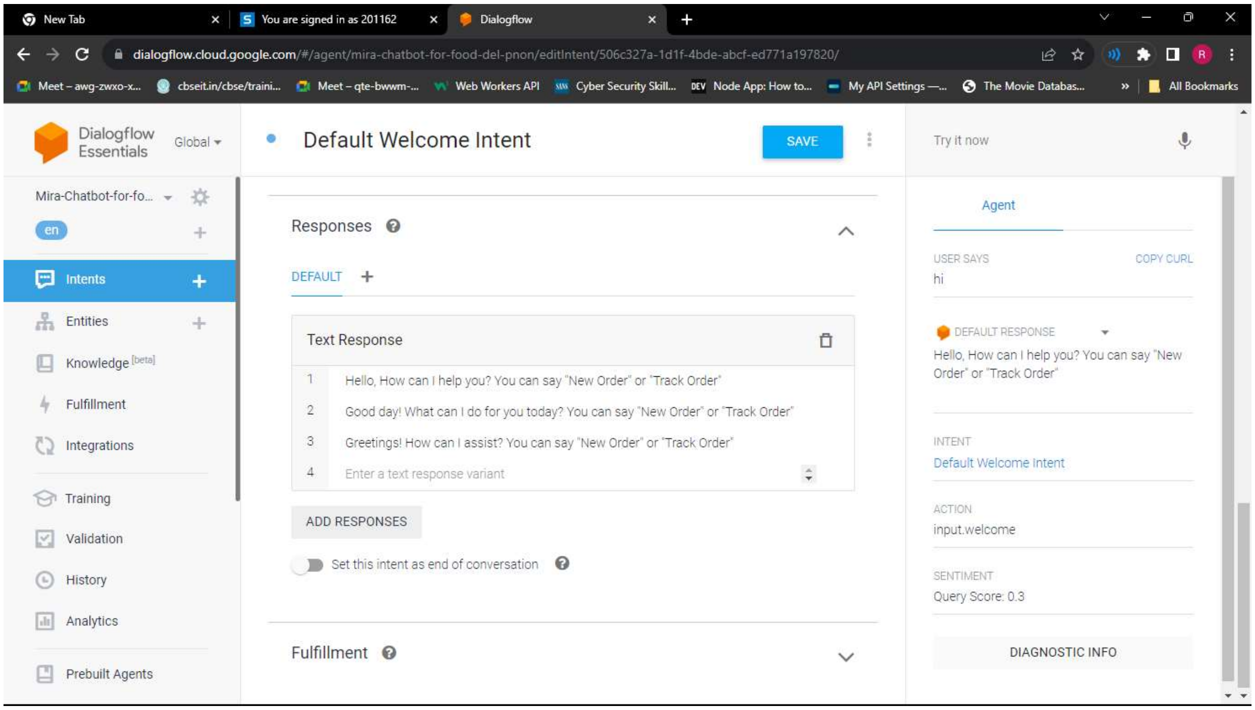
INTENT
Default Welcome Intent

ACTION
input.welcome

SENTIMENT
Query Score: 0.3

Figure 3.1.6.7(Welcome Intent)

b . Text Response : That part of the chatbot which acts accordingly and gives the user a desired output . A simple of text response is shown below :



Default Welcome Intent

Responses

Text Response

1 Hello, How can I help you? You can say "New Order" or "Track Order"

2 Good day! What can I do for you today? You can say "New Order" or "Track Order"

3 Greetings! How can I assist? You can say "New Order" or "Track Order"

4 Enter a text response variant

Set this intent as end of conversation

Fulfillment

USER SAYS: hi

COPY CURL

DEFAULT RESPONSE: Hello, How can I help you? You can say "New Order" or "Track Order"

INTENT: Default Welcome Intent

ACTION: input.welcome

SENTIMENT: Query Score: 0.3

Figure 3.1.6.8(Text Response : Welcome Intent)

- Main Question: What are the prerequisites required for the setting up of the chatbot using the Dialog flow chatbot?

Dialogflow has been divided into a number of subparts and all together when integrated successfully makes a complete working chatbot.

The part of the dialogflow that make it a complete chatbot are :

- Enabling of Dialogflow API
- Creating a Dialogflow Agent
- Intent and Training Phase
- Entities
- Fulfillment setup
- Testing
- Deployment

Now the main part is to discuss all of them in brief

- Enabling of Dialogflow API : The enabling of Dialogflow API can be done under the API's and services through the GCP console > Dashboard.

- Creating a Dialogflow Agent : It is a virtual agent that helps in processing the user input and it provides the response. For this we need to generate a new agent in the dialogflow console and after the creation of the console we need to console its setting.
- Intent and Training Phase : It includes the defining of the intents that should be used in the chatbot that it needs to handle . Intents include the different types of user input and we need to provide a separate training phase to each intent.

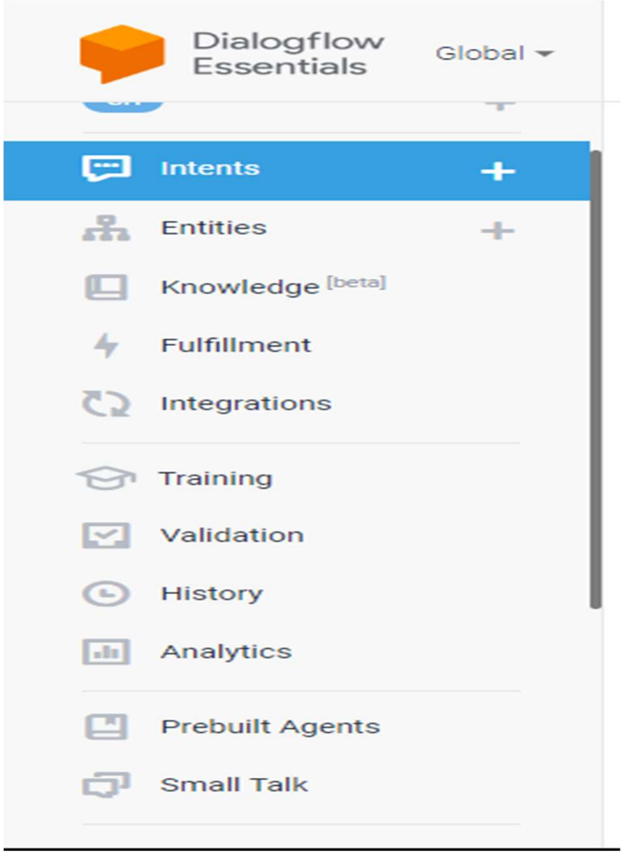
- Entities : When we need to extract something useful from the chatbot we need to have entities for that . Entities are like supporting variables that capture data from the messages of different users.

- Fulfillment Setup: If we need to extract data from some external source, we can use fulfillment for that. Fulfillment requires the connection of webhook, webhook is a kind of server that handles all the requests whatever the Dialogflow provides.

- Testing : We can test our chatbot using the pre-built functions of the google dialogflow to test our chatbot simulator . Testing is used to ensure that all our entities , intents and the responses are working properly.

- Deployment : Once the user is satisfied with the performance of the chatbot , then the user can deploy the chatbot to the desired platform in our case it is a web based platform .

For all the above information here is a supportive image that correctly defines all the parameters of the google dialogflow .



3.1.6.9(Structure of Dialogflow)

- The next Main Question : If I have to build a chatbot from scratch, what are the required algorithms and what are the required information and the skills that a user needs to have about NLP ? What are the main components that are required to achieve the work target ?

  "It all depends on how powerful we need to make our chatbot . There are various simple algorithms as well as many complex algorithms to design the chatbot . It also depends on the purpose of the chatbot , some of the simple classification algorithms are there that map the given intents with the phases and similarly when a new intent comes in then it again maps the intent to it and after whole mapping it responds back to the user.

- Our Question : What are the resources available that can help me to complete my project and overcome all the challenges that come during the completion of the project ?

  “Google dialogflow don’t have much information on youtube but we can get help from the dialogflow tutorials itself on how to setup dialogflow , create intents and entities” . There are a few videos on the youtube too that may help us achieving the completion of the project.

3.1.7 SUPPORTIVE INFORMATION

During the starting of the project the main problem was to identify the problem statement that should be unique and as well as easy to justify . For that we choose the project to be a delivery application , using this we can resolve a lot of issues of the customers as well the working staff . Keeping the following points in mind we decided to move forward with this project.

- Time Management : In today's world, time constraint is the major problem for all the parts of the society and keeping this thing in mind we decided to form an order page so that users can easily place orders within a few seconds without wasting much time.
- Data Collection : It helps in the collection of data of various users that may be a beneficial thing for the company , they keep track of the previously ordered things by a user so that they can match up to user expectation.
- User Satisfaction : User Satisfaction is a key to success for many firms , users may be satisfied by the fast ordering facility , on screen order , all those things that saves the user energy as well time.

CHAPTER – 4 : TESTING

Testing Level	Objective	Tools	Approach
Unit Testing	Test individual components of the chatbot.	unittest, pytest	Verify that components work as expected.
Fulfillment Testing	Validate fulfillment logic and accurate responses.	Dialogflow Fulfillment Simulator	Test end-to-end flow, including fulfillment actions.
End-to-End Testing	Test entire conversation flow for a seamless experience.	Dialogflow Simulator, External tools	Simulate complete conversations covering scenarios.
User Experience Testing	Evaluate overall user experience and interface design.	Manual testing, User feedback	Ensure the chatbot is intuitive and effective.
Regression Testing	Ensure new updates do not introduce regressions.	Automated testing frameworks	Regularly run tests on new and existing features.
Performance and Scalability Testing	Assess performance under different loads.	Load testing tools, Cloud services	Simulate high volume to identify performance issues.
Security Testing	Identify and address potential security vulnerabilities.	Security testing tools	Conduct security assessments to mitigate threats.
CI/CD Integration	Automate testing in the CI/CD pipeline.	CI/CD platforms	Ensure continuous integration and deployment.

CHAPTER – 5 : RESULT AND EVALUATION

5.1 Basics of Google Dialogflow

The diagram below correctly shows the working of the google dialogflow with machine learning techniques .

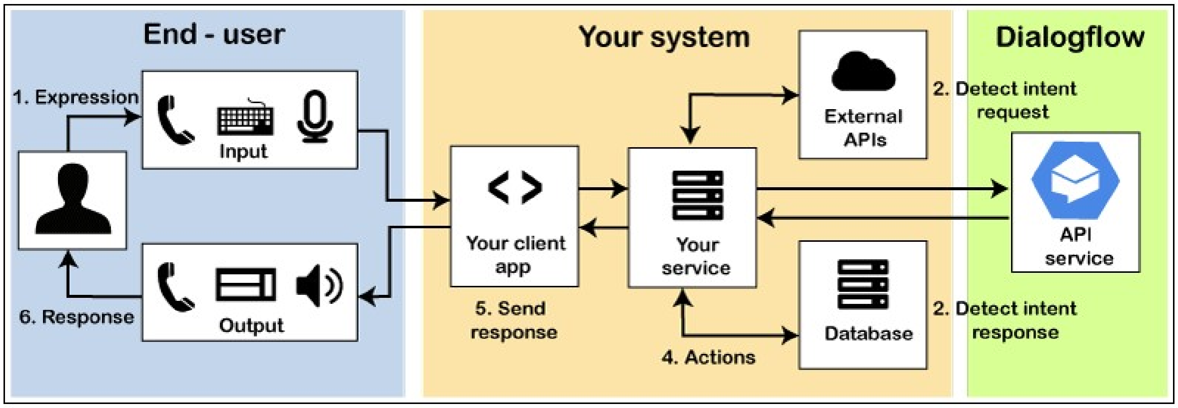


Figure 5.1.1(Basic of Google Dialogflow)

5.2 Architecture Model Of Google Dialogflow

The diagram provided below will give a basic overview of google dialogflow. The two primary components for this are NLU and the NLG .

All the intent classification part, entity extraction and response retrieval are handled by the NLU . The NLU pipeline has been shown below to clearly justify the process. The dialogue management component decides the next action that needs to be taken in the conversation. This is displayed as follows :

The figure includes the following parts :

- Action Server
- Tracker Order
- New Order
- Track Order : Ongoing

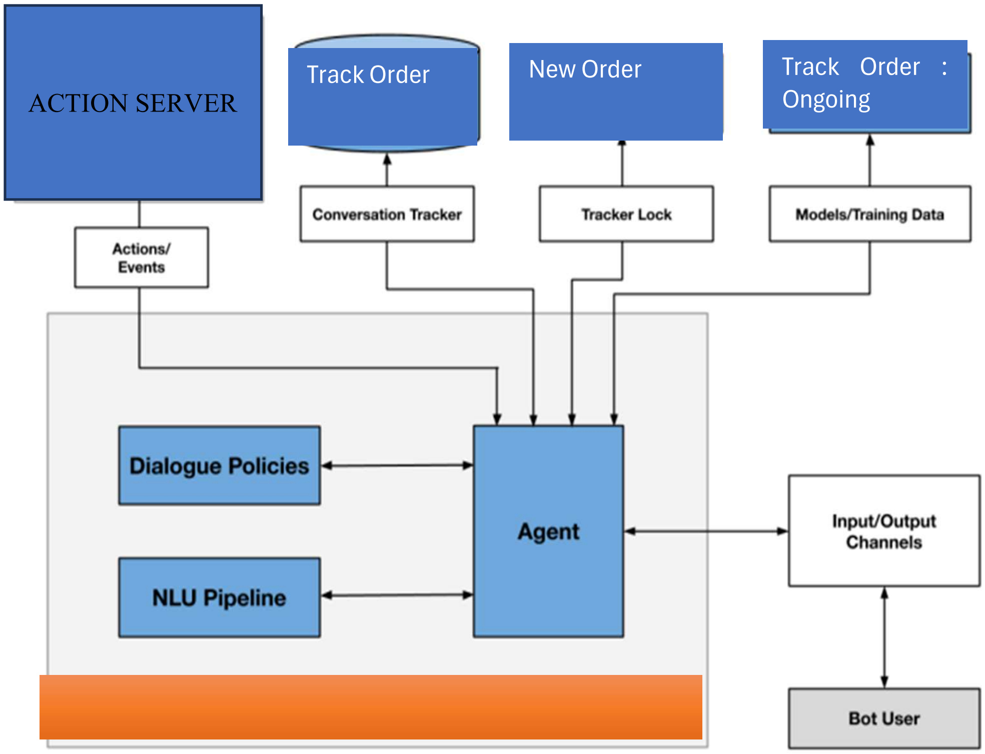


Figure 5.2.1(Architecture Model of Google Dialogflow)

5.3 Google DialogFlow Framework

At this point of time while writing the report we have been using the latest version of the google dialogflow , dialogflow is a google platform that's the main reason we need not to update the platform it keeps on updating itself .

The basic framework and the flow that would be required majorly in our project is shown below

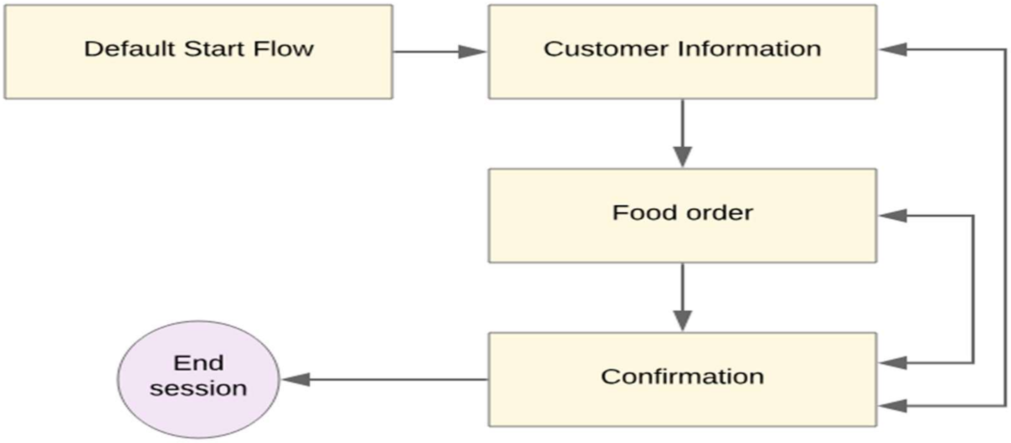
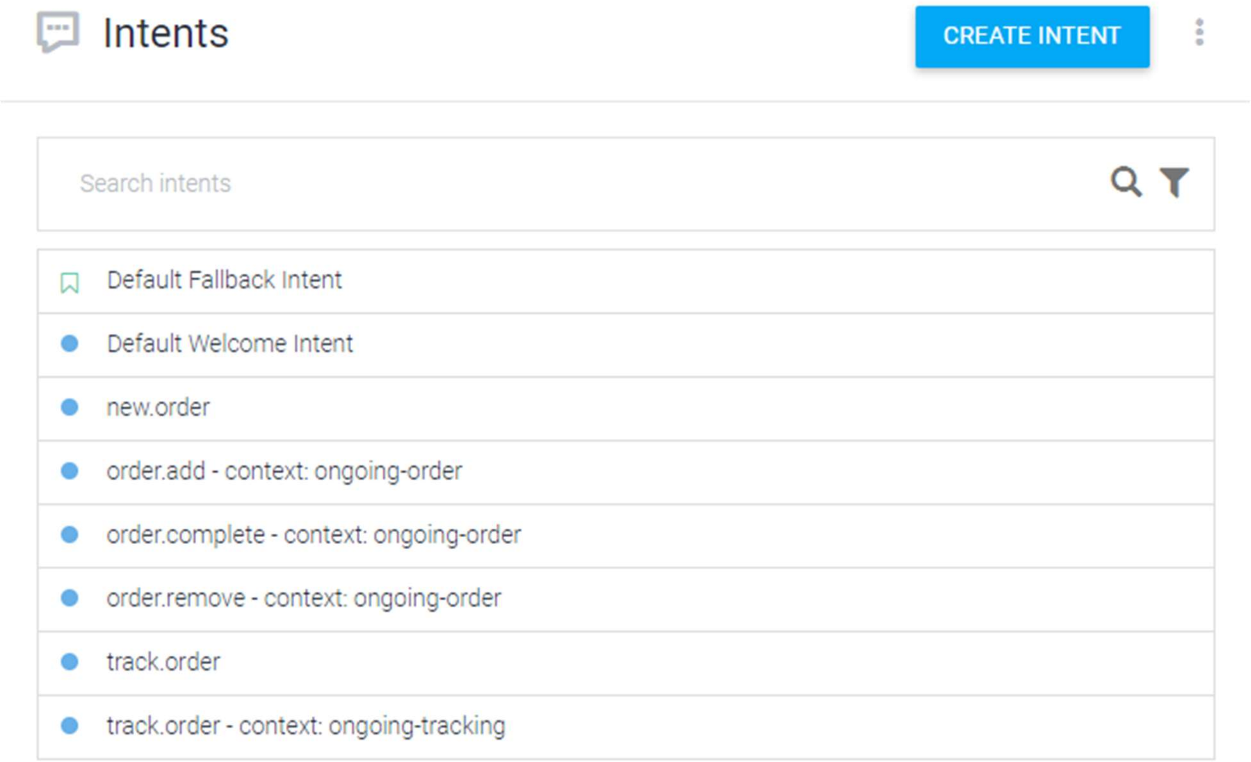


Figure 5.3.1(Framework of Chatbot)

Unlike other chatbots we don't need to have a separate file for the handling of the intent and the entities , that is what makes the google dialogflow a bit faster and quicker .

- Intent : It contains all the parameters that can be a user goal , and the basic examples of utterances contains those goals . As already mentioned above NLP is a combination of NLU and NLG , therefore the NLU contains the entities that are labeled.
- Entities : The most important keyword that needs to be taken care of by the assistant . Let’s understand this with the help of an example : considering a sentence “ My Name is Pranav” has the name “Pranav” in it , now the assistant should keep in mind where the exact word “pranav” occurred to keep the conversation more natural.
- Utterances : Whatsoever a user says is counted in an Utterance .Whenever a whole sentence is passed at a single time it is known as the single utterance.

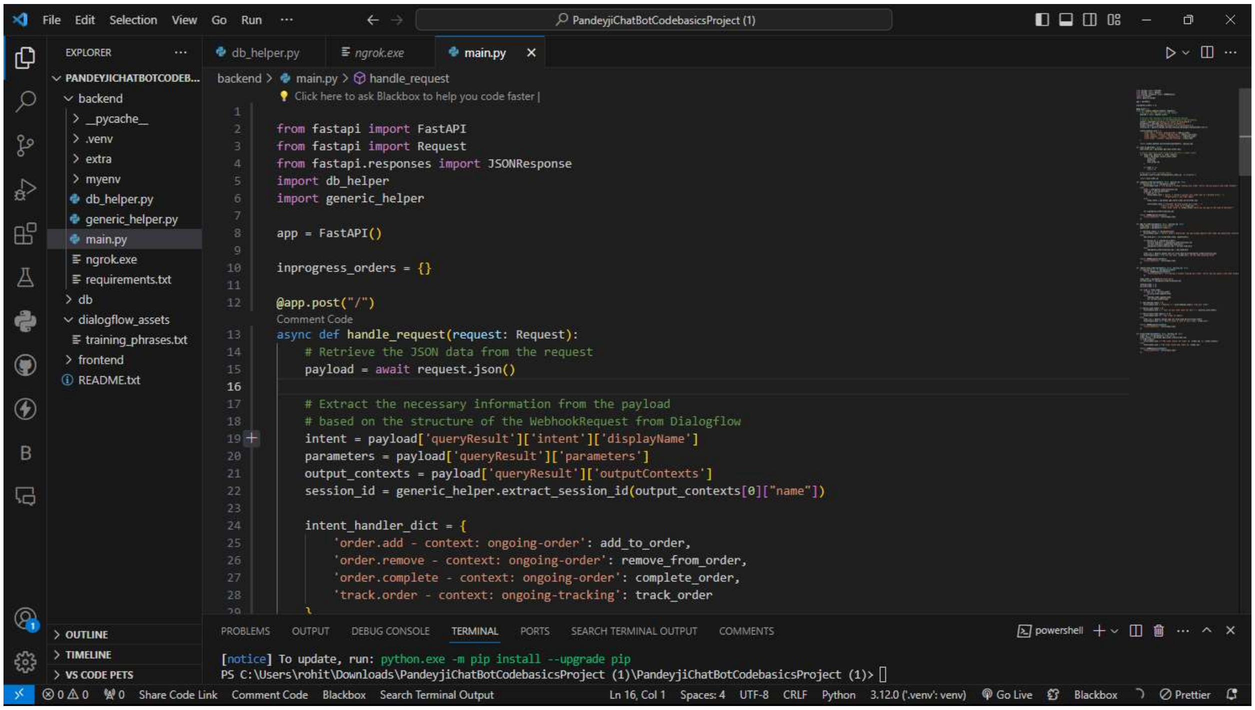


Default Fallback Intent
Default Welcome Intent
new.order
order.add - context: ongoing-order
order.complete - context: ongoing-order
order.remove - context: ongoing-order
track.order
track.order - context: ongoing-tracking

Figure 5.3.2(LIST OF INTENTS)

- **Main.py :** This is main file for the connection of the backend part so that the user can get what it wants

- Includes the coding of the backend
- Connection of the backend to the chatbot
- Connection of the backend to the frontend (i.e. a general frontend web page for the food ordering).



```python

from fastapi import FastAPI
from fastapi import Request
from fastapi.responses import JSONResponse
import db_helper
import generic_helper

app = FastAPI()

inprogress_orders = {}

@app.post("/")
async def handle_request(request: Request):
    # Retrieve the JSON data from the request
    payload = await request.json()

    # Extract the necessary information from the payload
    # based on the structure of the WebhookRequest from Dialogflow
    intent = payload['queryResult']['intent']['displayName']
    parameters = payload['queryResult']['parameters']
    output_contexts = payload['queryResult']['outputContexts']
    session_id = generic_helper.extract_session_id(output_contexts[0]["name"])

    intent_handler_dict = {
        'order.add - context: ongoing-order': add_to_order,
        'order.remove - context: ongoing-order': remove_from_order,
        'order.complete - context: ongoing-order': complete_order,
        'track.order - context: ongoing-tracking': track_order
    }
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\rohit\Downloads\PandeyjiChatBotCodebasicsProject (1)\PandeyjiChatBotCodebasicsProject (1)> 
```

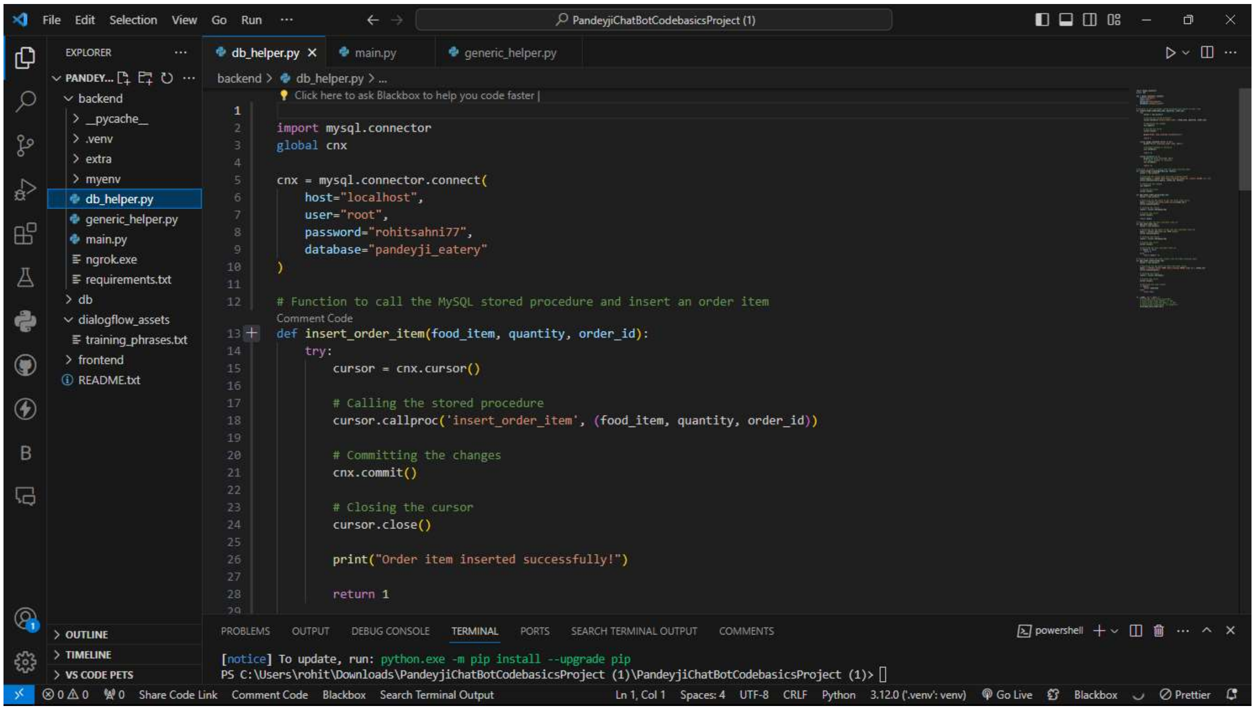
Figure 5.3.3(MAIN.PY – BACKEND FILE I)

This part of the project completely includes the handling of the backend , it includes

- ❖ Addition of the food items
- ❖ Getting of the order_id
- ❖ Retrieval of the next order id

For the retrieval of all this db_helper has played a major role in figuring all the things out.

DB_HELPER : Whenever db_helper comes it is a bit obvious that we need to fetch something out from the documents based on the filter condition and the query that the user provided according to the condition.



File Edit Selection View Go Run ···
PandeyjiChatBotCodebasicsProject (1)
EXPLORER
PANDEY...
backend
pycache
.venv
extra
myenv
db_helper.py
generic_helper.py
main.py
ngrok.exe
requirements.txt
db
dialogflow_assets
training_phrases.txt
frontend
README.txt
OUTLINE
TIMELINE
VS CODE PETS
db_helper.py
main.py
generic_helper.py
backend > db_helper.py > ...
Click here to ask Blackbox to help you code faster |
Comment Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH TERMINAL OUTPUT COMMENTS
powershell
0 0 0 Share Code Link Comment Code Blackbox Search Terminal Output Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.0 ('.venv': venv) Go Live Blackbox Prettier

```python

import mysql.connector
global cnx

cnx = mysql.connector.connect(
    host="localhost",
    user="root",
    password="rohitsahni77",
    database="pandeyji_eatery"
)

# Function to call the MySQL stored procedure and insert an order item
def insert_order_item(food_item, quantity, order_id):
    try:
        cursor = cnx.cursor()

        # Calling the stored procedure
        cursor.callproc('insert_order_item', (food_item, quantity, order_id))

        # Committing the changes
        cnx.commit()

        # Closing the cursor
        cursor.close()

        print("Order item inserted successfully!")

        return 1
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\rohit\Downloads\PandeyjiChatBotCodebasicsProject (1)\PandeyjiChatBotCodebasicsProject (1)>
```

FIGURE 5.3.4(DB_HELPER – BACKEND PART II)

- Generic Helper : It is mainly divided into 2 parts :
  - Get_str_from_food_dict
  - Extract_session_id

The “get_str_from_food_dict” treats the “food_dict” as the function input. There are various numbers of keys in the dictionaries , each key is expected to return a value of the food item, and the quantity of the food item.

-This function uses the format “ {quantity}{food item}” to create the list of the food items in the string format.

- The concatenation of the strings are done using the “join” methodology and these are separated using “,” (commas).

As ““get_str_from_food_dict” treats the “food_dict” as the function input , the “extract_session_id” treats “session_str” as the functional input.

-It uses the regular expression function to extract specific patterns from the input string.

-The pattern is searched for in the session i.e. “/sessions/(.*?)/contexts”. When the substring starts with the “/session” then it is used.

- If a match is found with this format then it returns the substring else an empty substring is returned.

Extract session is broken down in the following way as shown below :

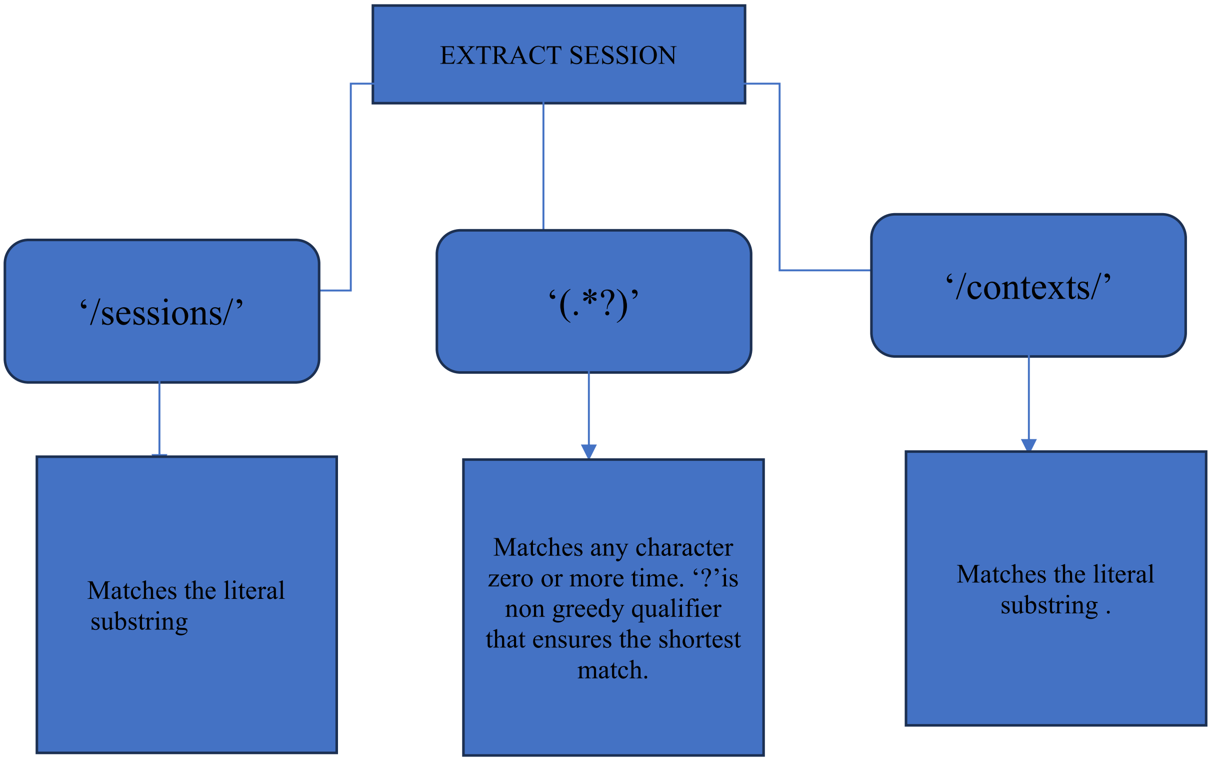
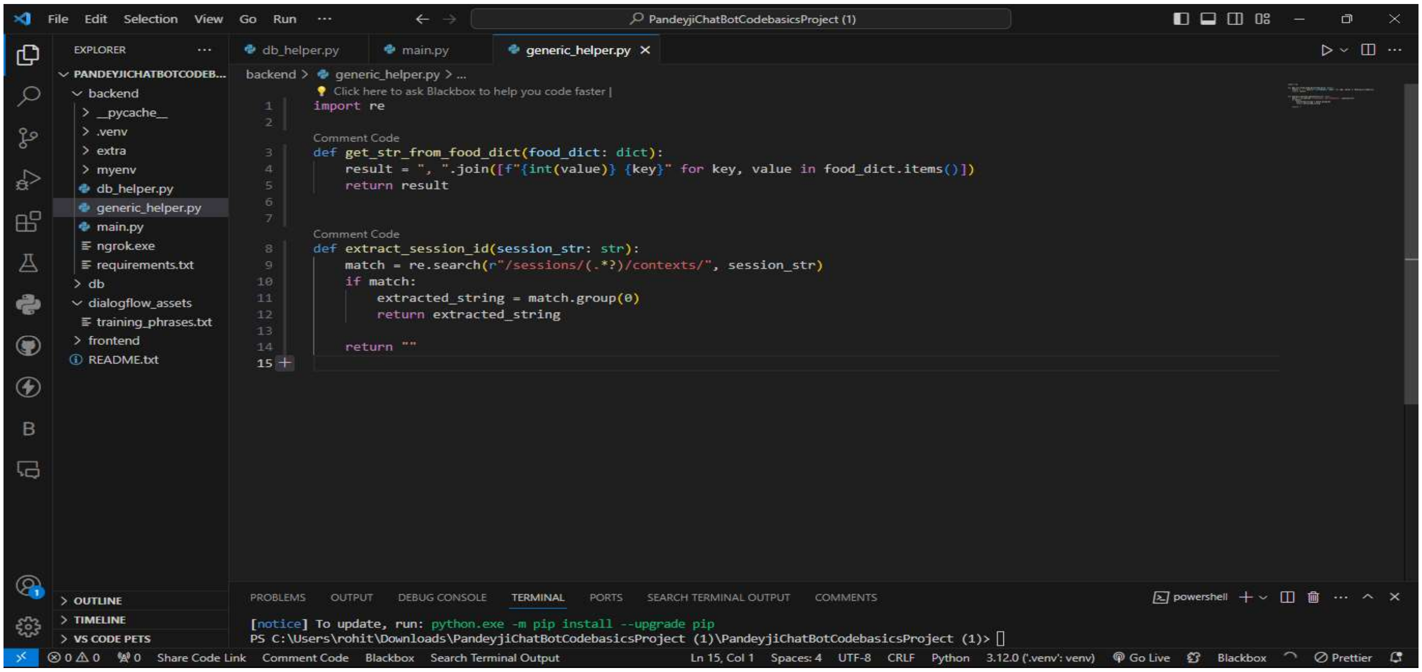

EXTRACT SESSION
'/sessions/'
'(.*?)'
'/contexts/'
Matches the literal substring
Matches any character zero or more time. '?'is non greedy qualifier that ensures the shortest match.
Matches the literal substring .

FIGURE 5. 3.5(DIVISION OF THE EXTRACT SESSION)

The main purpose of the is to extract the session id from the string itself that contains a list of special patterns.



```
import re

def get_str_from_food_dict(food_dict: dict):
    result = ", ".join([f"{int(value)} {key}" for key, value in food_dict.items()])
    return result


def extract_session_id(session_str: str):
    match = re.search(r"/sessions/(.*?)/contexts/", session_str)
    if match:
        extracted_string = match.group(0)
        return extracted_string

    return ""
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\rohit\Downloads\PandeyjiChatBotCodebasicsProject (1)\PandeyjiChatBotCodebasicsProject (1)>
```

File Edit Selection View Go Run ···
PandeyjiChatBotCodebasicsProject (1)
EXPLORER
PANDEYJICHATBOTCODEB...
backend
__pycache__
.venv
extra
myenv
db_helper.py
generic_helper.py
main.py
ngrok.exe
requirements.txt
db
dialogflow_assets
training_phrases.txt
frontend
README.txt
OUTLINE
TIMELINE
VS CODE PETS
db_helper.py
main.py
generic_helper.py
backend > generic_helper.py > ...
Click here to ask Blackbox to help you code faster |
Comment Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH TERMINAL OUTPUT COMMENTS
powershell
Share Code Link Comment Code Blackbox Search Terminal Output
Ln 15, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.0 ('.venv': venv) Go Live Blackbox Prettier

FIGURE 5. 3.6(GENERIC_HELPER – BACKEND FILE III)

The next plan was to set up the database from the whole pricing and the list of the items would be matched . The user will see the price of each item from the my-sql database,named as the pandey_ji_eatery.

This includes the following steps :

- Setting up of the account of my-sql
- Generating a my-sql table
- Setting up of different tables
  - Food-items
  - Order_tracking
  - Orders

- Contains two functions namely
  - Get_price_for_item
  - Get_total_order_price

The basic schema looks like this:

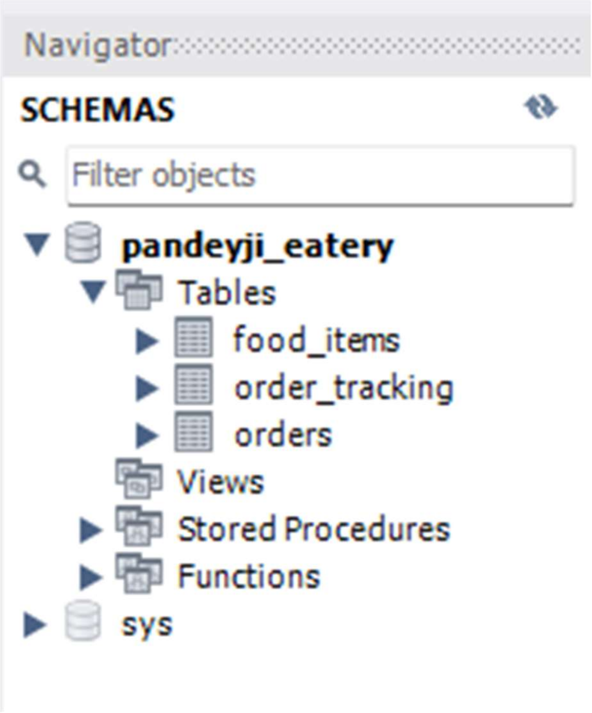
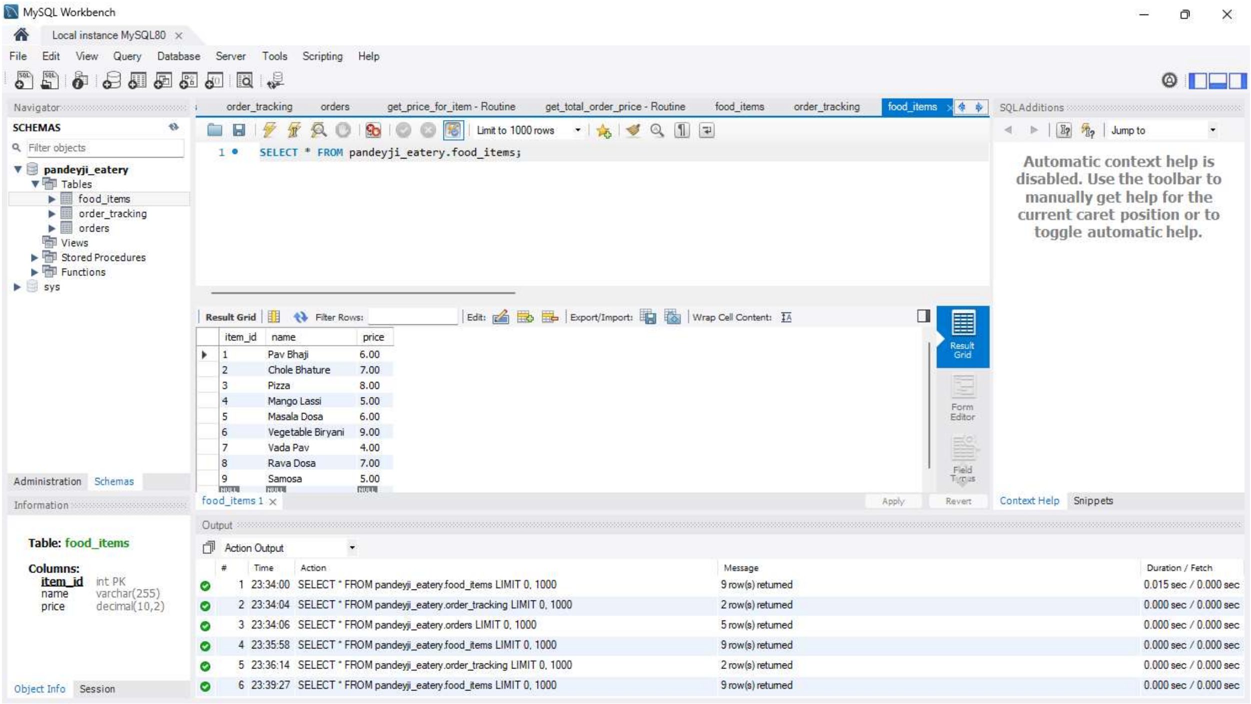


FIGURE 5. 3.7(SCHEMA STRUCTURE)

FOOD-ITEM :

For the "food_items" we considered only a few food items that we also considered during the time of the framing of intents , this is done so that the intent matching could be done easily and the "food_items" gets matched easily. In the future we are planning to add more of the food items to the list . Here's the schema of the food items.



```
1 ● SELECT * FROM pandeyji_eatery.food_items;
```

item_id	name	price
1	Pav Bhaji	6.00
2	Chole Bhature	7.00
3	Pizza	8.00
4	Mango Lassi	5.00
5	Masala Dosa	6.00
6	Vegetable Biryani	9.00
7	Vada Pav	4.00
8	Rava Dosa	7.00
9	Samosa	5.00
NULL	NULL	NULL

Table: food_items

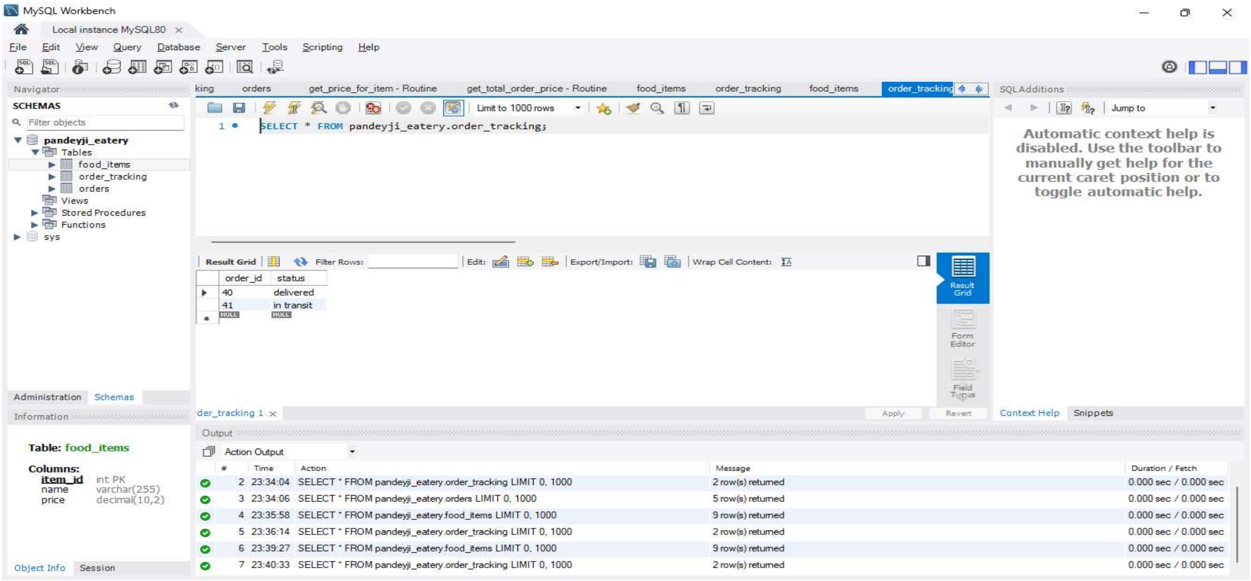
Columns:
item_id int PK
name varchar(255)
price decimal(10,2)

```
#  Time      Action                                                      Message            Duration / Fetch
1  23:34:00  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000      9 row(s) returned  0.015 sec / 0.000 sec
2  23:34:04  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000  2 row(s) returned  0.000 sec / 0.000 sec
3  23:34:06  SELECT * FROM pandeyji_eatery.orders LIMIT 0, 1000          5 row(s) returned  0.000 sec / 0.000 sec
4  23:35:58  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000      9 row(s) returned  0.000 sec / 0.000 sec
5  23:36:14  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000  2 row(s) returned  0.000 sec / 0.000 sec
6  23:39:27  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000      9 row(s) returned  0.000 sec / 0.000 sec
```

FIGURE 5. 3.8(SCHEMA STRUCTURE : FOOD – ITEMS)

ORDER-TRACKING :

Order tracking includes the current status of the order , the current status of the order is either delivered or in-transit. The schema is shown below.



```
SELECT * FROM pandeyji_eatery.order_tracking;
```

order_id	status
40	delivered
41	in transit
NULL	NULL

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Table: food_items

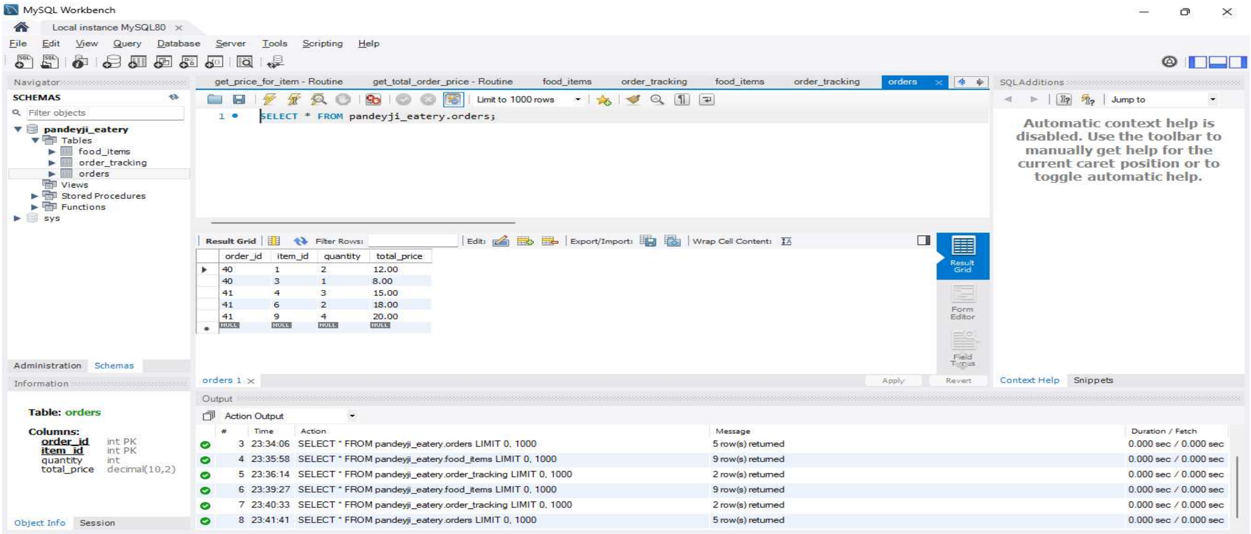
Columns:
item_id int PK
name varchar(255)
price decimal(10,2)

```
#  Time      Action                                                     Message              Duration / Fetch
2  23:34:04  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000  2 row(s) returned    0.000 sec / 0.000 sec
3  23:34:06  SELECT * FROM pandeyji_eatery.orders LIMIT 0, 1000          5 row(s) returned    0.000 sec / 0.000 sec
4  23:35:58  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000      9 row(s) returned    0.000 sec / 0.000 sec
5  23:36:14  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000  2 row(s) returned    0.000 sec / 0.000 sec
6  23:39:27  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000      9 row(s) returned    0.000 sec / 0.000 sec
7  23:40:33  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000  2 row(s) returned    0.000 sec / 0.000 sec
```

FIGURE 5. 3.9(SCHEMA STRUCTURE : ORDER TRACKING)

ORDER:

This consist of the following key field order_id , item_id , quantity , total_price .The schema is shown below :



```
SELECT * FROM pandeyji_eatery.orders;
```

order_id	item_id	quantity	total_price
40	1	2	12.00
40	3	1	8.00
41	4	3	15.00
41	6	2	18.00
41	9	4	20.00
NULL	NULL	NULL	NULL

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Table: orders

Columns:
order_id int PK
item_id int PK
quantity int
total_price decimal(10,2)

```
#  Time      Action                                                     Message              Duration / Fetch
3  23:34:06  SELECT * FROM pandeyji_eatery.orders LIMIT 0, 1000          5 row(s) returned    0.000 sec / 0.000 sec
4  23:35:58  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000      9 row(s) returned    0.000 sec / 0.000 sec
5  23:36:14  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000  2 row(s) returned    0.000 sec / 0.000 sec
6  23:39:27  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000      9 row(s) returned    0.000 sec / 0.000 sec
7  23:40:33  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000  2 row(s) returned    0.000 sec / 0.000 sec
8  23:41:41  SELECT * FROM pandeyji_eatery.orders LIMIT 0, 1000          5 row(s) returned    0.000 sec / 0.000 sec
```

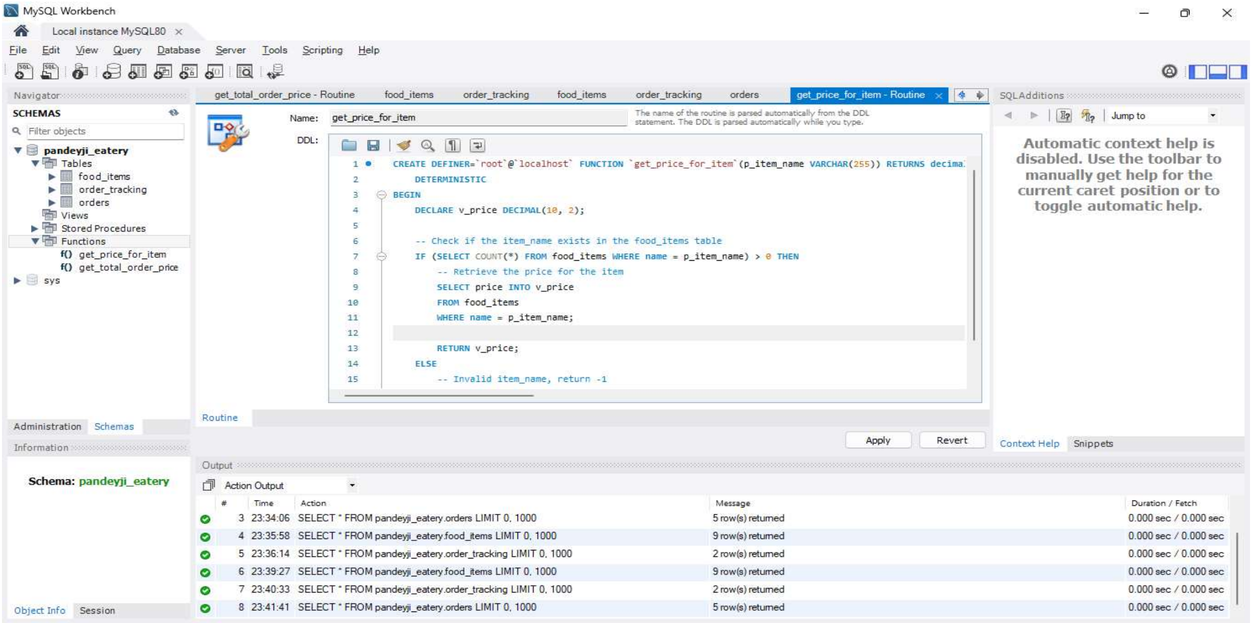
FIGURE 5. 3.10(SCHEMA STRUCTURE : ORDER)

There were 2 kinds of function designed :

1.TO GET THE PRICE OF THE ITEM

2.TO GET THE TOTAL PRICE OF THE ORDER

The following is the DDL query for the both :

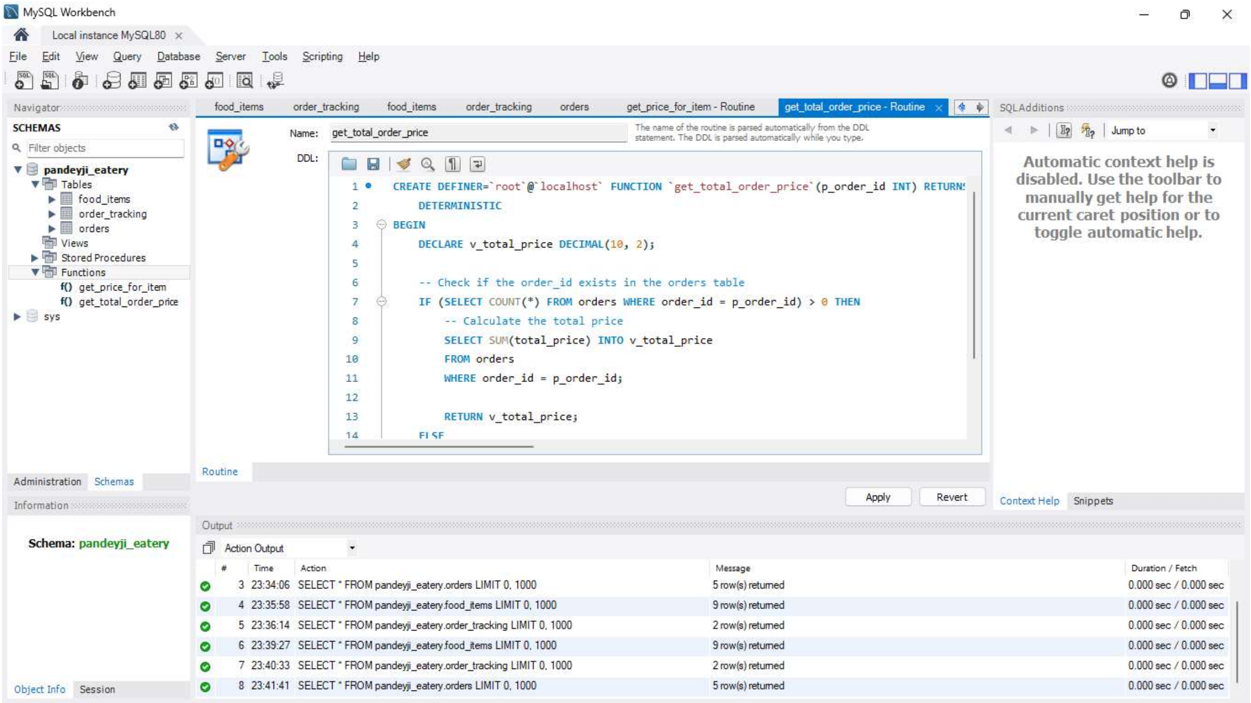


Name: get_price_for_item

```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_price_for_item`(p_item_name VARCHAR(255)) RETURNS decima
    DETERMINISTIC
BEGIN
    DECLARE v_price DECIMAL(10, 2);

    -- Check if the item_name exists in the food_items table
    IF (SELECT COUNT(*) FROM food_items WHERE name = p_item_name) > 0 THEN
        -- Retrieve the price for the item
        SELECT price INTO v_price
        FROM food_items
        WHERE name = p_item_name;

        RETURN v_price;
    ELSE
        -- Invalid item_name, return -1
```

```
#  Time      Action                                                   Message           Duration / Fetch
3  23:34:06  SELECT * FROM pandeyji_eatery.orders LIMIT 0, 1000         5 row(s) returned  0.000 sec / 0.000 sec
4  23:35:58  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000     9 row(s) returned  0.000 sec / 0.000 sec
5  23:36:14  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000 2 row(s) returned  0.000 sec / 0.000 sec
6  23:39:27  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000     9 row(s) returned  0.000 sec / 0.000 sec
7  23:40:33  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000 2 row(s) returned  0.000 sec / 0.000 sec
8  23:41:41  SELECT * FROM pandeyji_eatery.orders LIMIT 0, 1000         5 row(s) returned  0.000 sec / 0.000 sec
```

FIGURE 5. 3.11(DDL QUERY: get_price_for_item)



Name: get_total_order_price

```
CREATE DEFINER=`root`@`localhost` FUNCTION `get_total_order_price`(p_order_id INT) RETURNS
    DETERMINISTIC
BEGIN
    DECLARE v_total_price DECIMAL(10, 2);

    -- Check if the order_id exists in the orders table
    IF (SELECT COUNT(*) FROM orders WHERE order_id = p_order_id) > 0 THEN
        -- Calculate the total price
        SELECT SUM(total_price) INTO v_total_price
        FROM orders
        WHERE order_id = p_order_id;

        RETURN v_total_price;
    ELSE
```

```
#  Time      Action                                                   Message           Duration / Fetch
3  23:34:06  SELECT * FROM pandeyji_eatery.orders LIMIT 0, 1000         5 row(s) returned  0.000 sec / 0.000 sec
4  23:35:58  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000     9 row(s) returned  0.000 sec / 0.000 sec
5  23:36:14  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000 2 row(s) returned  0.000 sec / 0.000 sec
6  23:39:27  SELECT * FROM pandeyji_eatery.food_items LIMIT 0, 1000     9 row(s) returned  0.000 sec / 0.000 sec
7  23:40:33  SELECT * FROM pandeyji_eatery.order_tracking LIMIT 0, 1000 2 row(s) returned  0.000 sec / 0.000 sec
8  23:41:41  SELECT * FROM pandeyji_eatery.orders LIMIT 0, 1000         5 row(s) returned  0.000 sec / 0.000 sec
```

FIGURE 5. 3.12(DDL QUERY: get_total_order_price)

KEY CHALLENGES :

While making of this project we encountered a lot of key challenges that dragged us away from our project a lot many time :

- DATA SET AVAILABILITY :

  The one and only problem that arose is for designing of chatbot using google dialogflow a google platform we don't have any data set available for this purpose . This is was one of the major challenge for our project . 2 weeks were completely dedicated towards the preparation of the data set and the what the intents would be and how will they link to each other . A simple demonstration of how the intents are linked to each other is shown below in the form of a pictorial diagram :

THIS IS A BASIC FLOW HOW OUR CHATBOT WORKS:

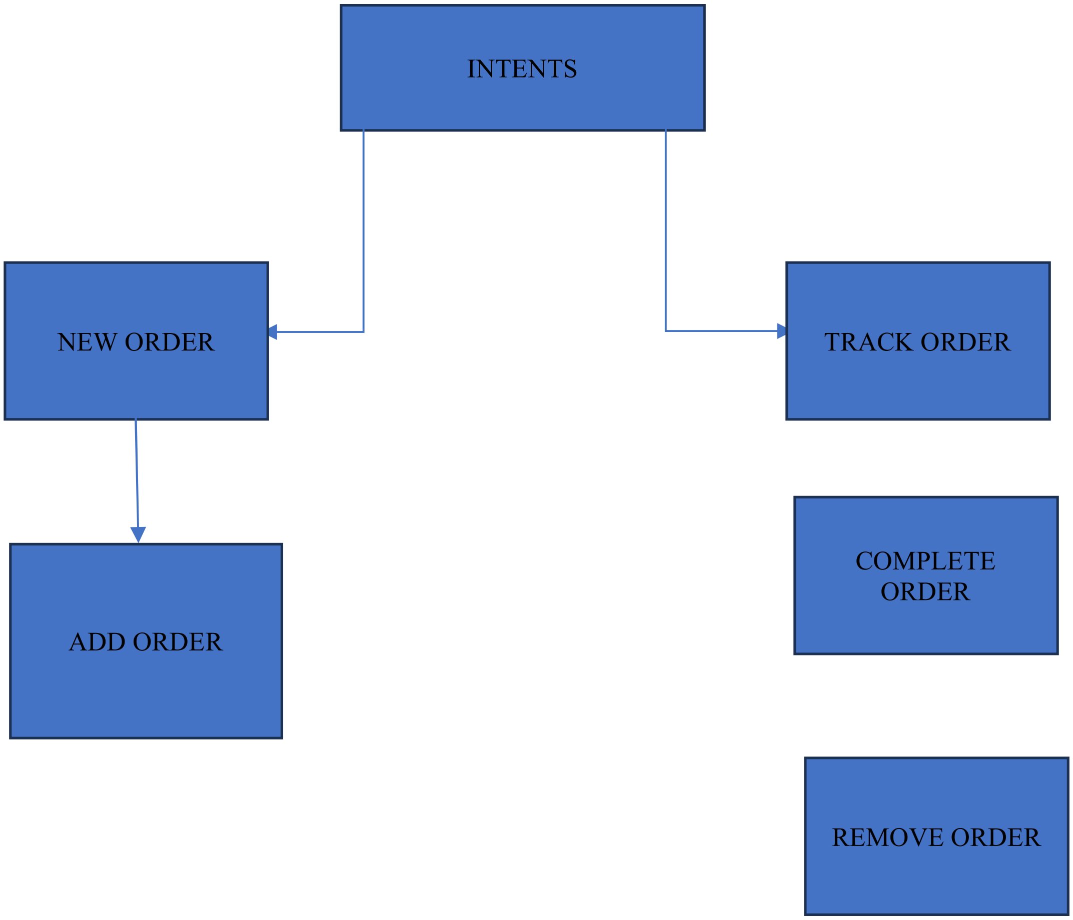
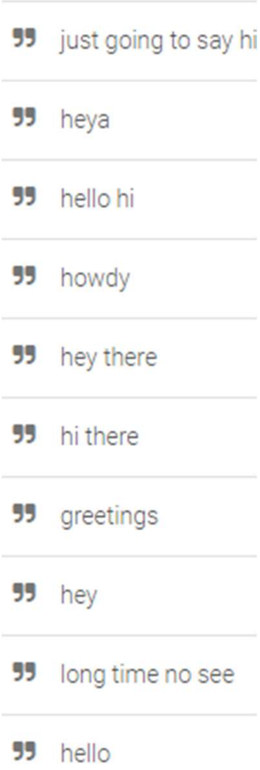


FIGURE 5. 3.13(BASIC INTENT FLOW)

IMPLEMENTATION :

The general implementation of the working chatbot is shown below :

1. THE WELCOME INTENT :
   This is managed by the welcome intent of the the dialogue flow this includes the following intents :



just going to say hi

heya

hello hi

howdy

hey there

hi there

greetings

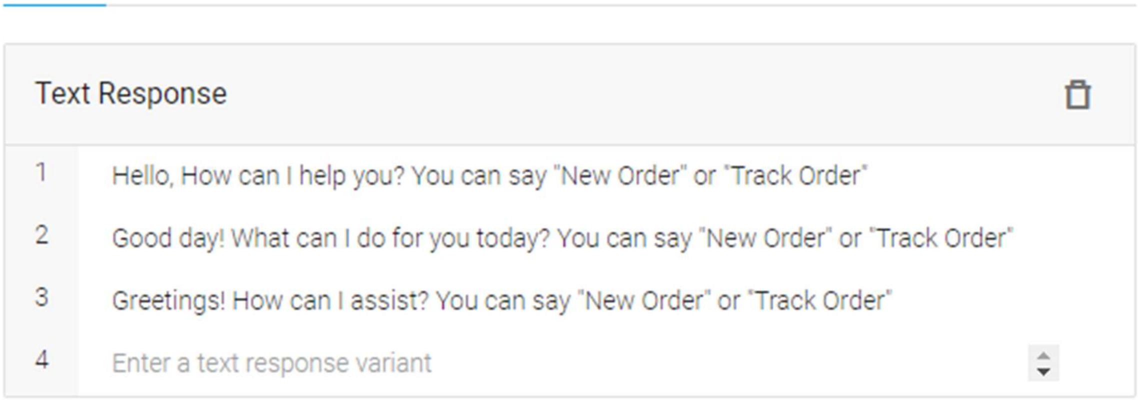
hey

long time no see

hello

FIGURE 5.3.13(Training Phase : Welcome Intent)

Our chatbot would answer accordingly



Text Response

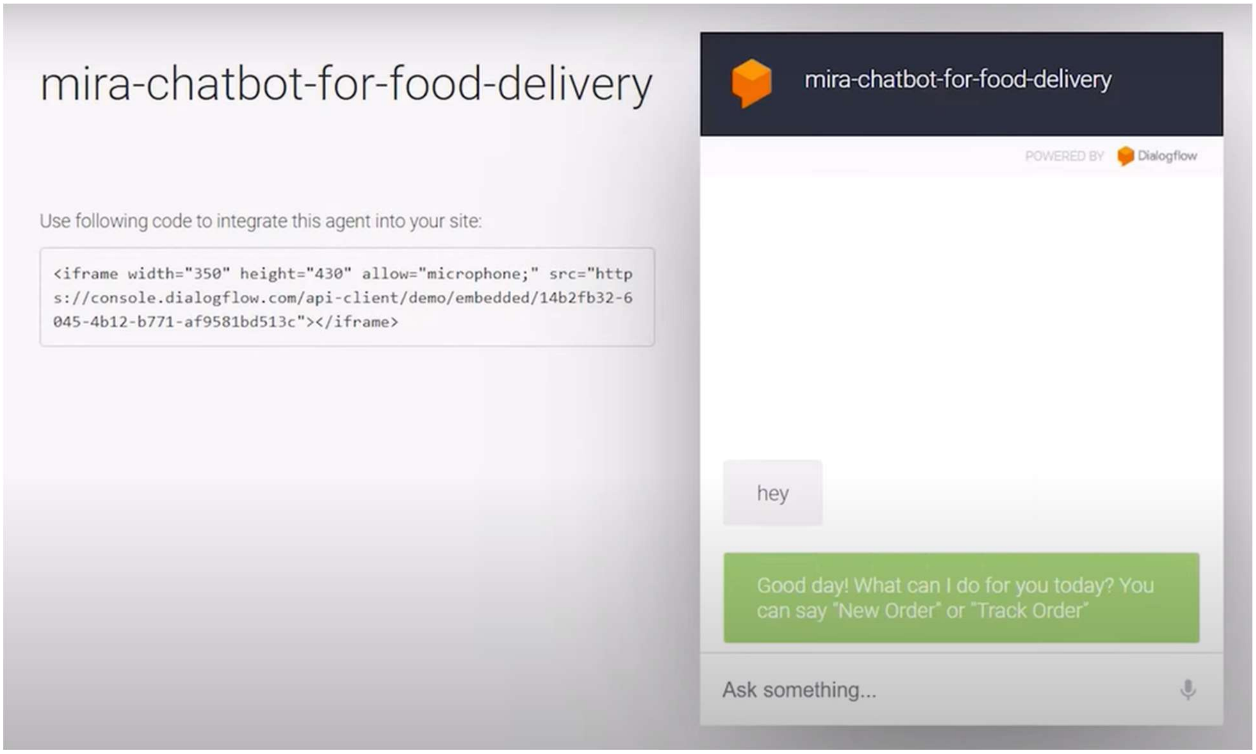
1 Hello, How can I help you? You can say "New Order" or "Track Order"

2 Good day! What can I do for you today? You can say "New Order" or "Track Order"

3 Greetings! How can I assist? You can say "New Order" or "Track Order"

4 Enter a text response variant

Figure 5. 3.14(Text Response : Welcome Intent)
This clearly demonstrates the responses mapped to the welcome intent and the example of the welcome intent is mentioned below -



mira-chatbot-for-food-delivery

Use following code to integrate this agent into your site:

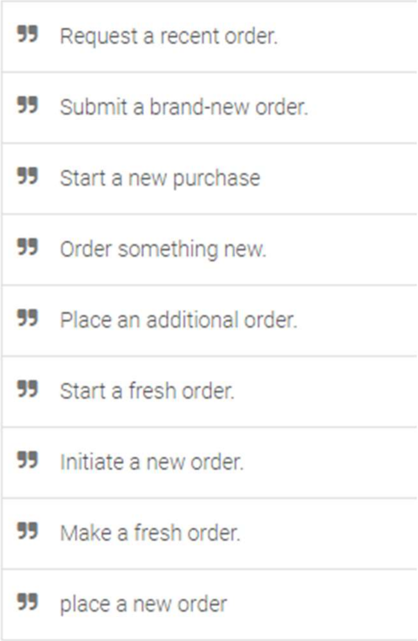
```
<iframe width="350" height="430" allow="microphone;" src="http
s://console.dialogflow.com/api-client/demo/embedded/14b2fb32-6
045-4b12-b771-af9581bd513c"></iframe>
```

hey

Good day! What can I do for you today? You can say "New Order" or "Track Order"

Figure 5. 3.15(Implementation : Welcome Intent)

2.PLACING OF THE NEW ORDER: Whenever placing of order is done the dialogflow goes to new-order intent . This intent has the following training phase –

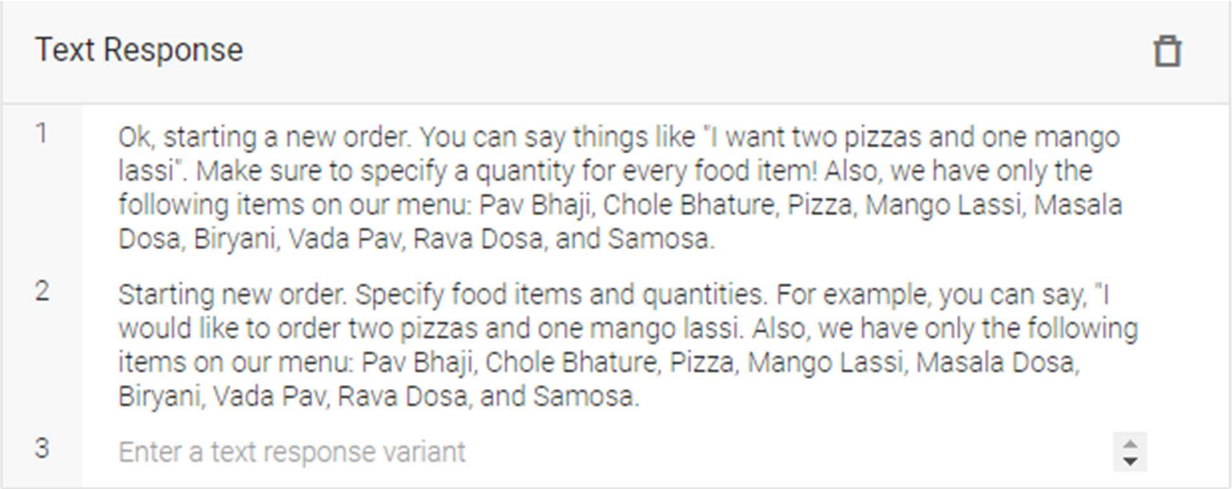


- Request a recent order.
- Submit a brand-new order.
- Start a new purchase
- Order something new.
- Place an additional order.
- Start a fresh order.
- Initiate a new order.
- Make a fresh order.
- place a new order

Figure 5. 3.16(Training Phase : New Order)

Whenever the user says anything matching to this then the dialogflow uses the NLP to break the words into the tokens and then matches the intent , and then responds back to the user with the text-response mapped with the training phase .

The following is the text-response that is mapped with the new.order :-



Text Response

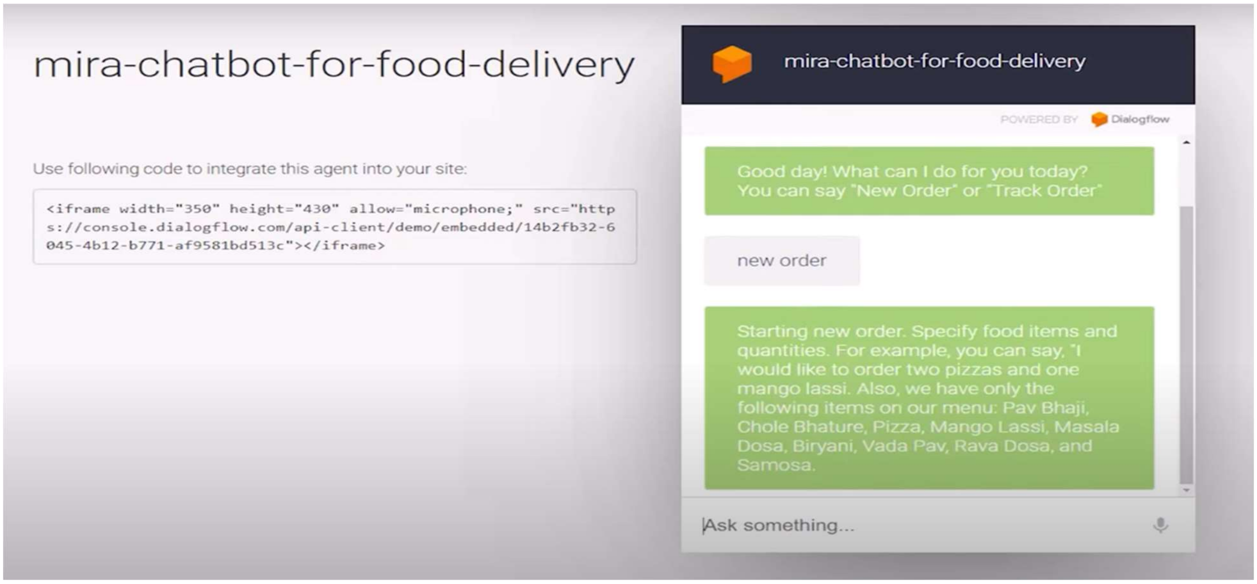
1 Ok, starting a new order. You can say things like "I want two pizzas and one mango lassi". Make sure to specify a quantity for every food item! Also, we have only the following items on our menu: Pav Bhaji, Chole Bhature, Pizza, Mango Lassi, Masala Dosa, Biryani, Vada Pav, Rava Dosa, and Samosa.

2 Starting new order. Specify food items and quantities. For example, you can say, "I would like to order two pizzas and one mango lassi. Also, we have only the following items on our menu: Pav Bhaji, Chole Bhature, Pizza, Mango Lassi, Masala Dosa, Biryani, Vada Pav, Rava Dosa, and Samosa.

3 Enter a text response variant

Figure 5. 3.17(Text Response : New Order)

According to this the user need to clearly mention the food quantity , then only the chatbot would accept the new order else it would respond back with “PLEASE SPECIFY THE FOOD QUANTITY”.



mira-chatbot-for-food-delivery

Use following code to integrate this agent into your site:

```
<iframe width="350" height="430" allow="microphone;" src="http
s://console.dialogflow.com/api-client/demo/embedded/14b2fb32-6
045-4b12-b771-af9581bd513c"></iframe>
```

mira-chatbot-for-food-delivery

Good day! What can I do for you today? You can say "New Order" or "Track Order"

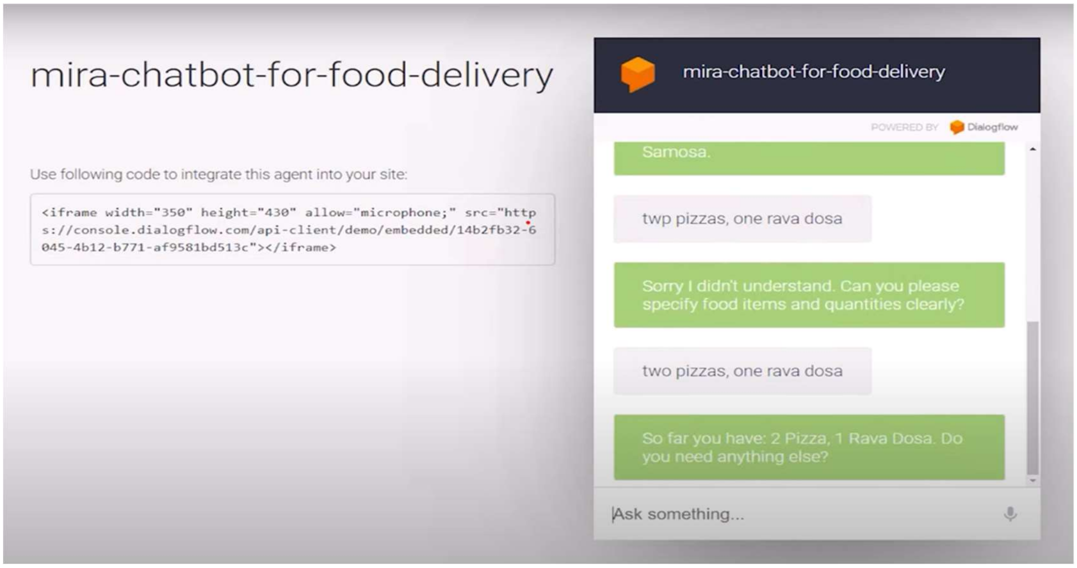
new order

Starting new order. Specify food items and quantities. For example, you can say, "I would like to order two pizzas and one mango lassi. Also, we have only the following items on our menu: Pav Bhaji, Chole Bhature, Pizza, Mango Lassi, Masala Dosa, Biryani, Vada Pav, Rava Dosa, and Samosa.

Figure 5. 3.18(Implementation : New Order)

3.ADDITION OF ITEMS :

This text response from the chatbot clearly indicates If there is a typo even in the food quantity then also the chatbot would not accept the order and return with “Sorry I didn’t understand . Can you please specify the food item and quantity clearly?”.


mira-chatbot-for-food-delivery
mira-chatbot-for-food-delivery
POWERED BY Dialogflow
Ask something...

Use following code to integrate this agent into your site:

```
<iframe width="350" height="430" allow="microphone;" src="http
s://console.dialogflow.com/api-client/demo/embedded/14b2fb32-6
045-4b12-b771-af9581bd513c"></iframe>
```

Samosa.

twp pizzas, one rava dosa

Sorry I didn't understand. Can you please specify food items and quantities clearly?

two pizzas, one rava dosa

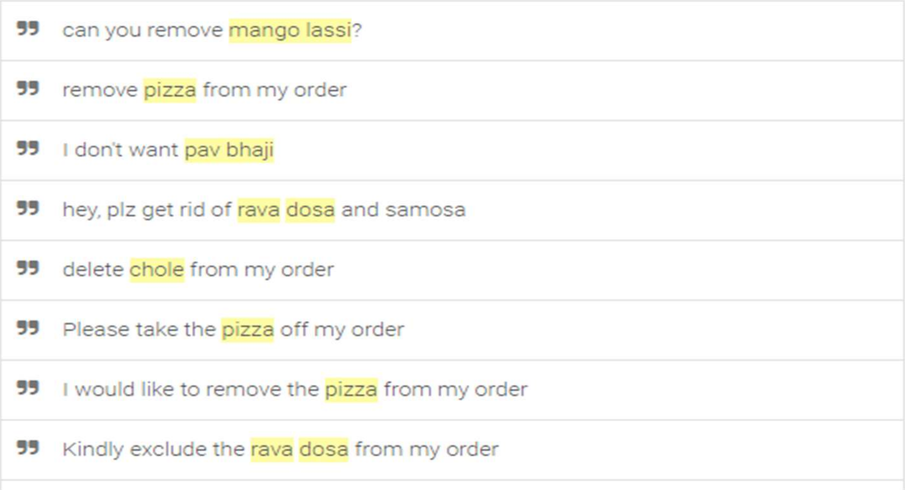
So far you have: 2 Pizza, 1 Rava Dosa. Do you need anything else?

Figure 5. 3.19(Implementation : Addition of Items)

As soon as the user gives the chatbot the input as “two pizza , one rava dosa” then the chatbot accepts the order and return with “any thing else”.

4. REMOVAL OF ITEM FROM THE ORDER :

The removal of the order is done by the “order.remove-context:ongoing order” intent . It has the all the intents which a user can give to remove or delete his current order .



can you remove mango lassi?

remove pizza from my order

I don't want pav bhaji

hey, plz get rid of rava dosa and samosa

delete chole from my order

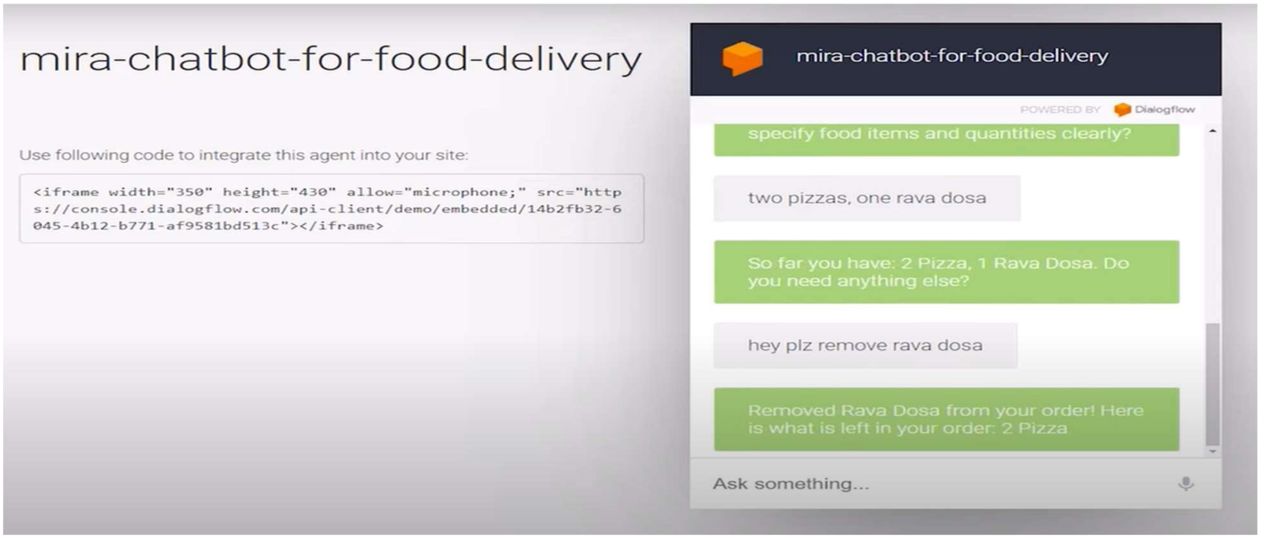
Please take the pizza off my order

I would like to remove the pizza from my order

Kindly exclude the rava dosa from my order

Figure 5. 3.20(Training Phase : Order.Remove)

Whenever the chatbot is encountered with “remove” it redirects itself to “order.remove-context:ongoing order” intent and provides the response mapped with it .


mira-chatbot-for-food-delivery

Use following code to integrate this agent into your site:

```
<iframe width="350" height="430" allow="microphone;" src="https://console.dialogflow.com/api-client/demo/embedded/14b2fb32-6045-4b12-b771-af9581bd513c"></iframe>
```

specify food items and quantities clearly?

two pizzas, one rava dosa

So far you have: 2 Pizza, 1 Rava Dosa. Do you need anything else?

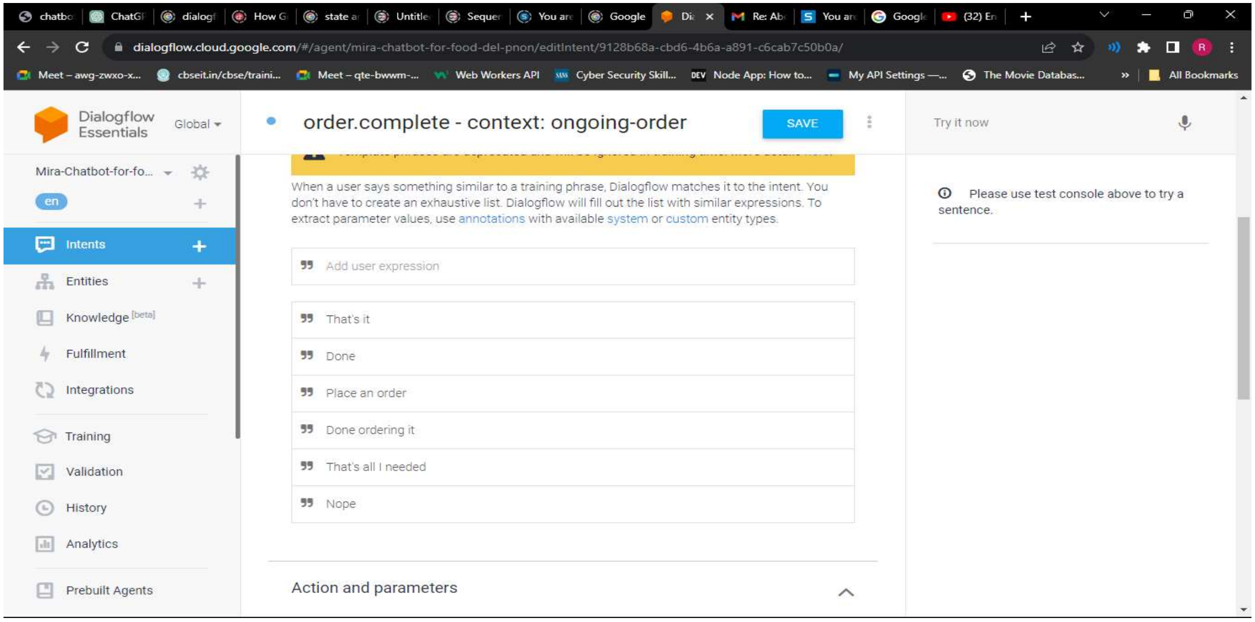
hey plz remove rava dosa

Removed Rava Dosa from your order! Here is what is left in your order: 2 Pizza

Figure 5. 3.21(Implementation : Order.Remove)

5. FINISHING OF THE ORDER :

For the finishing of the order the dialogflow goes to the “order.complete – context:ongoing-order” . This intent has the following training phase .


dialogflow.cloud.google.com/#/agent/mira-chatbot-for-food-del-pnon/editIntent/9128b68a-cbd6-4b6a-a891-c6cab7c50b0a/

order.complete - context: ongoing-order

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use annotations with available system or custom entity types.

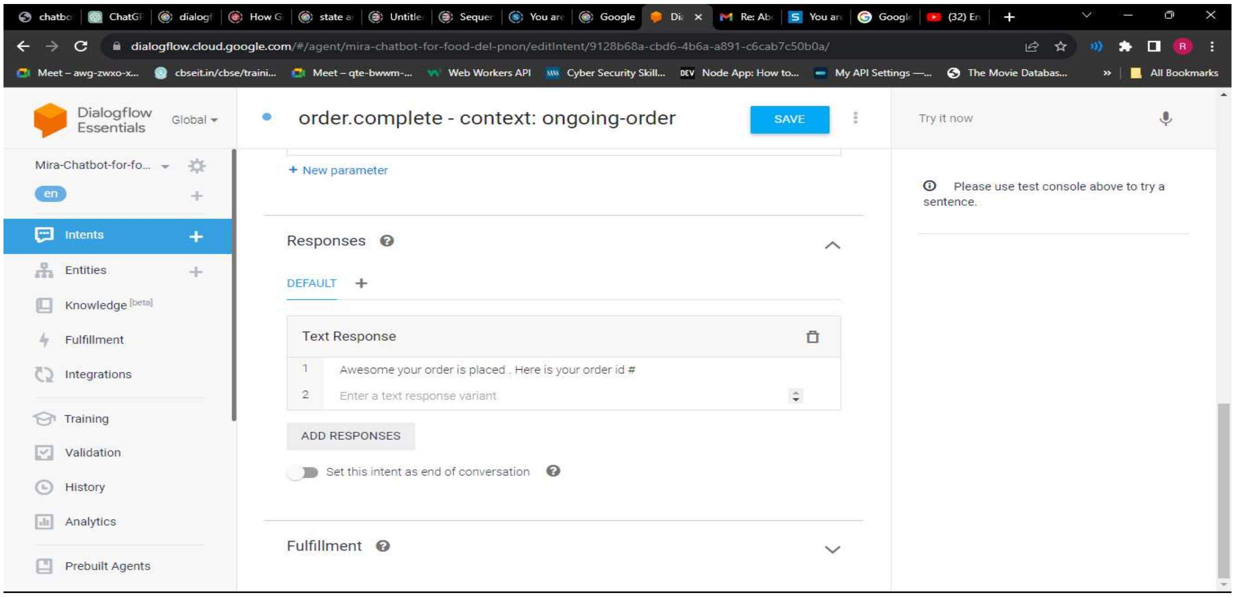
- That's it
- Done
- Place an order
- Done ordering it
- That's all I needed
- Nope

Please use test console above to try a sentence.

Figure 5. 3.22(Training Phase : Order.Complete)

Whenever the user places the order then it maps the intent to the respective text response associated with this intent

The Text response for the intent “order.complete – context:ongoing-order” is shown below :



order.complete - context: ongoing-order

Text Response

1 Awesome your order is placed . Here is your order id #

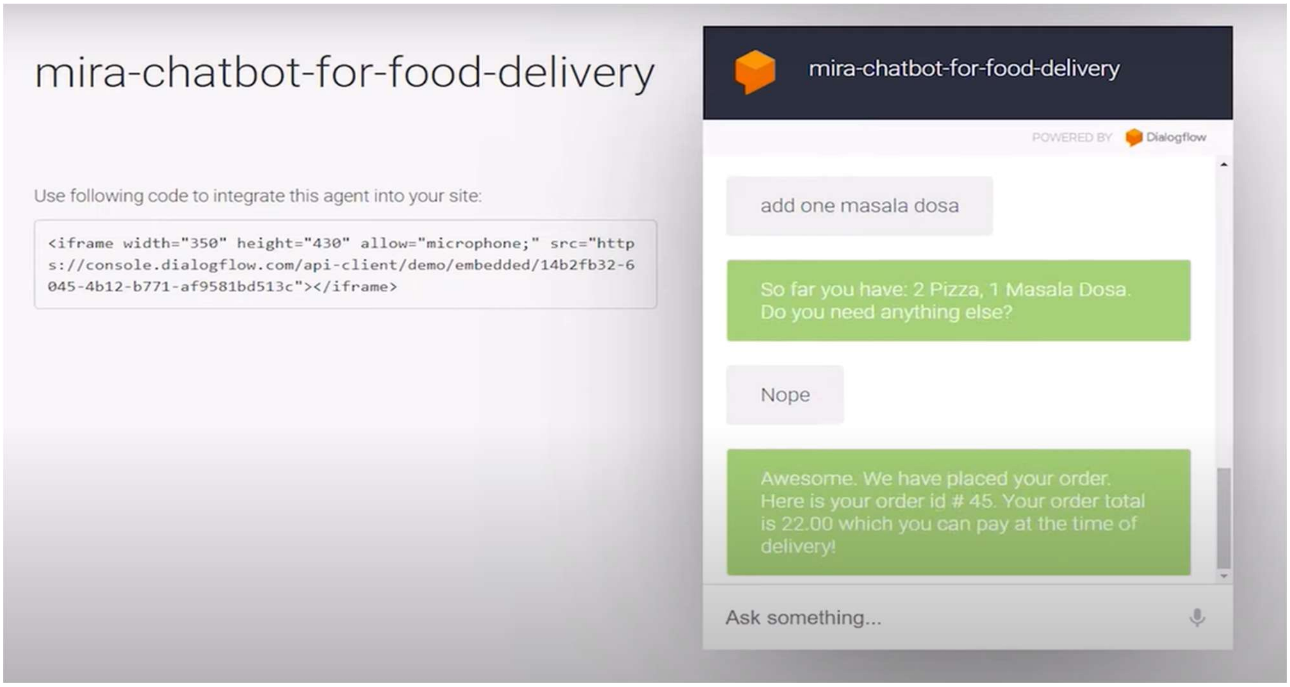
2 Enter a text response variant

Set this intent as end of conversation

Please use test console above to try a sentence.

Figure 5. 3.23(Text Response : Order.Complete)

The real demonstration how our chatbot works is given below :



mira-chatbot-for-food-delivery

Use following code to integrate this agent into your site:

```
<iframe width="350" height="430" allow="microphone;" src="http
s://console.dialogflow.com/api-client/demo/embedded/14b2fb32-6
045-4b12-b771-af9581bd513c"></iframe>
```

mira-chatbot-for-food-delivery

add one masala dosa

So far you have: 2 Pizza, 1 Masala Dosa. Do you need anything else?

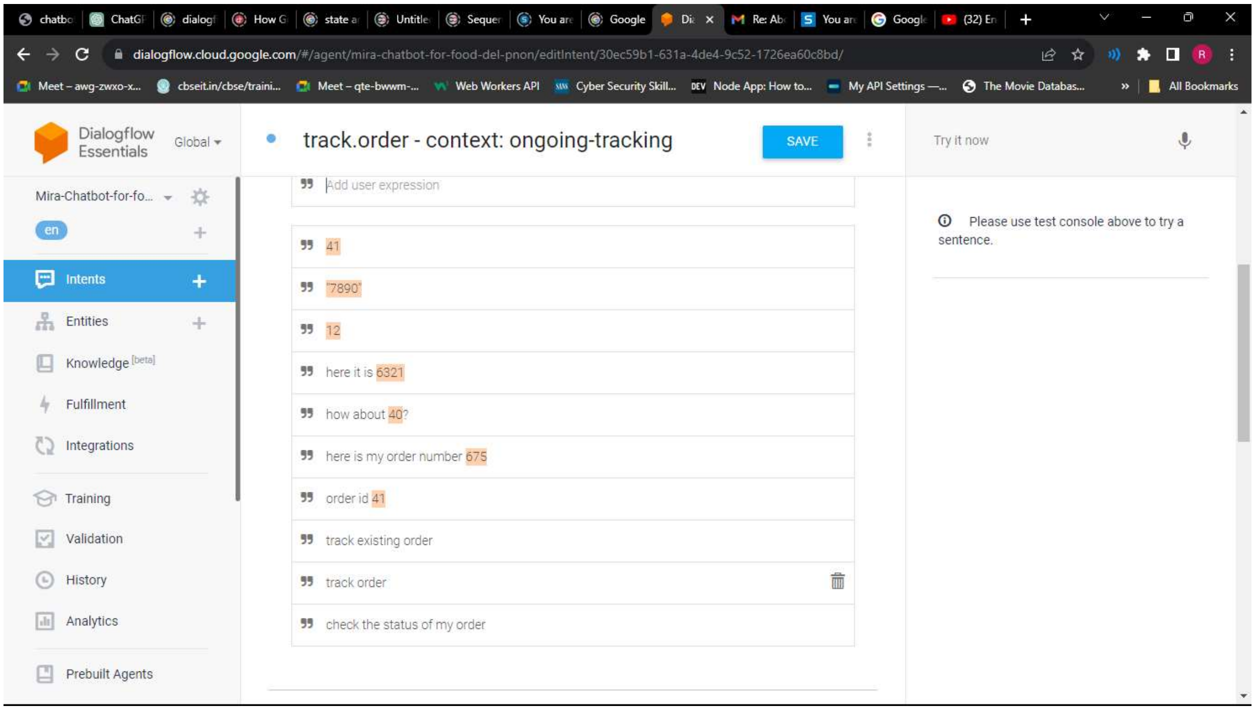
Nope

Awesome. We have placed your order. Here is your order id # 45. Your order total is 22.00 which you can pay at the time of delivery!

Figure 5. 3.24(Implementation : Completion of order)

6. TRACKING OF THE ORDER

Whenever the user gives the input “TRACK ORDER” to the chatbot . The chatbot follows up the “track.order -context:ongoing-tracking” , this intent is linked to the “track.order” intent . This has the following training phase :



track.order - context: ongoing-tracking

Add user expression

41

'7890'

12

here it is 6321

how about 40?

here is my order number 675

order id 41

track existing order

track order

check the status of my order

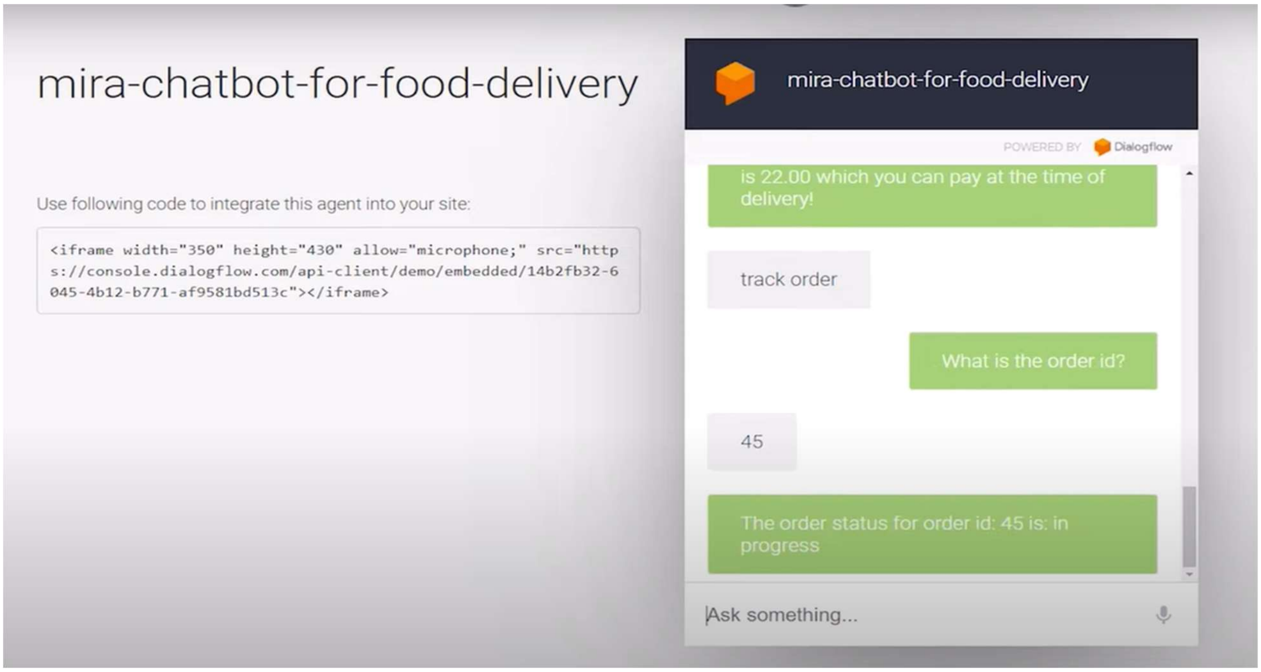
Figure 5. 3.25(Training Phase : Track.order: ongoing-order)

Whenever the user provides the chatbot with the order-id in whatever manner it first goes to track.order then follows up to the ongoing-tracking and responds back with the mapped text response.

The text response for this is mentioned in the backend part .In the main.py it is clearly mentioned that if there are order items then what should be the response linked to it .

```
def track_order(parameters: dict, session_id: str):
    order_id = int(parameters['order_id'])
    order_status = db_helper.get_order_status(order_id)
    if order_status:
            fulfillment_text = f"The order status for order id: {order_id} is:
{order_status}"
    else:
        fulfillment_text = f"No order found with order id: {order_id}"

    return JSONResponse(content={
        "fulfillmentText": fulfillment_text
    })
```

This clearly shows how our backend works for tracking the order.



mira-chatbot-for-food-delivery

Use following code to integrate this agent into your site:

```
<iframe width="350" height="430" allow="microphone;" src="https://console.dialogflow.com/api-client/demo/embedded/14b2fb32-6045-4b12-b771-af9581bd513c"></iframe>
```

is 22.00 which you can pay at the time of delivery!

track order

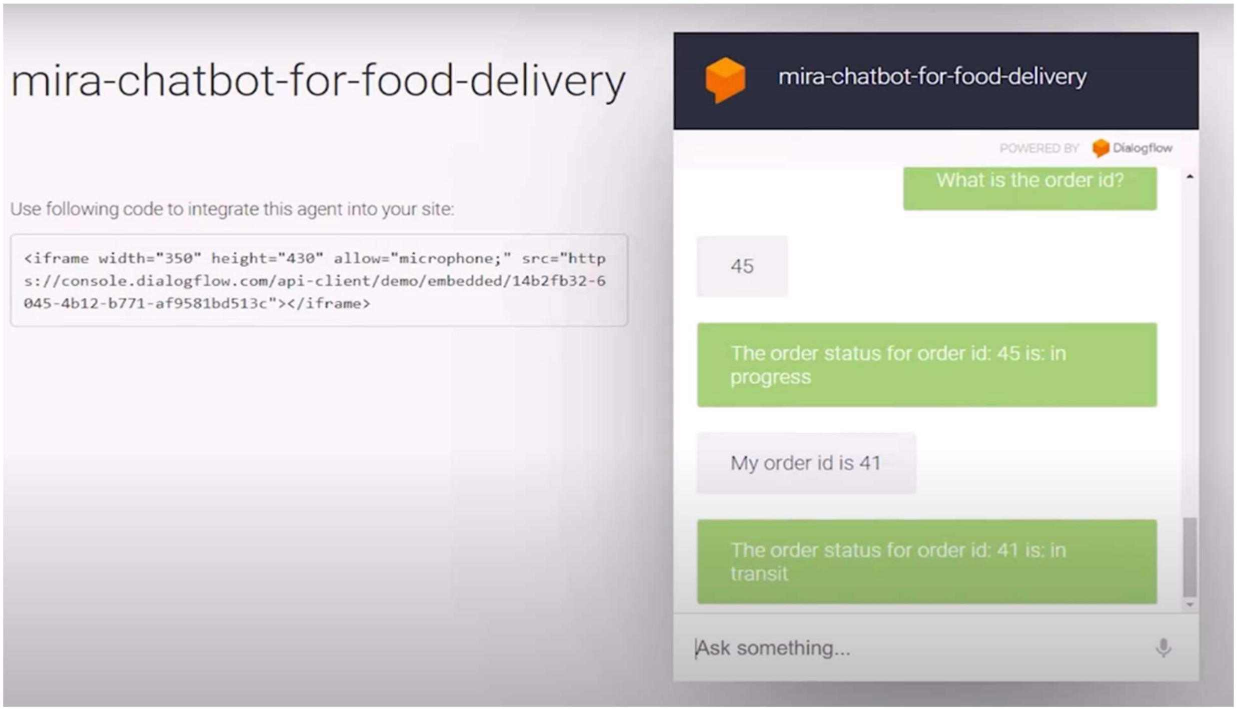
What is the order id?

45

The order status for order id: 45 is: in progress

Figure 5. 3.26(Implementation : Placement of Order)

7. TRACKING OF THE OTHER ORDER :



mira-chatbot-for-food-delivery

Use following code to integrate this agent into your site:

```
<iframe width="350" height="430" allow="microphone;" src="https://console.dialogflow.com/api-client/demo/embedded/14b2fb32-6045-4b12-b771-af9581bd513c"></iframe>
```

What is the order id?

45

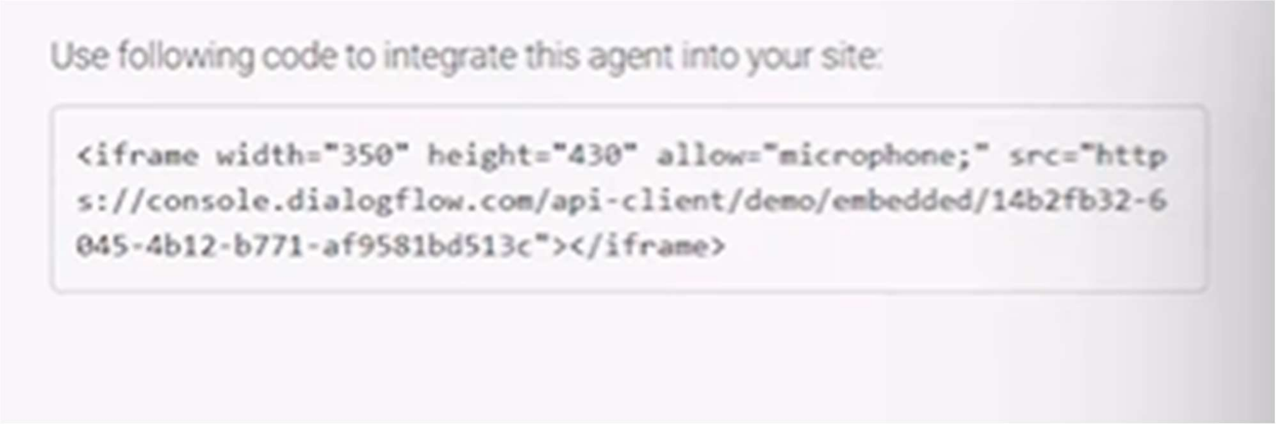
The order status for order id: 45 is: in progress

My order id is 41

The order status for order id: 41 is: in transit

Figure 5. 3.27(Implementation : Placement of Order)

The next is the final part of the project i.e to frame a frontend for the chatbot that has all the valid options . To integrate the chatbot to the frontend we just need to copy paste the code provided by the google dialogflow itself i.e.

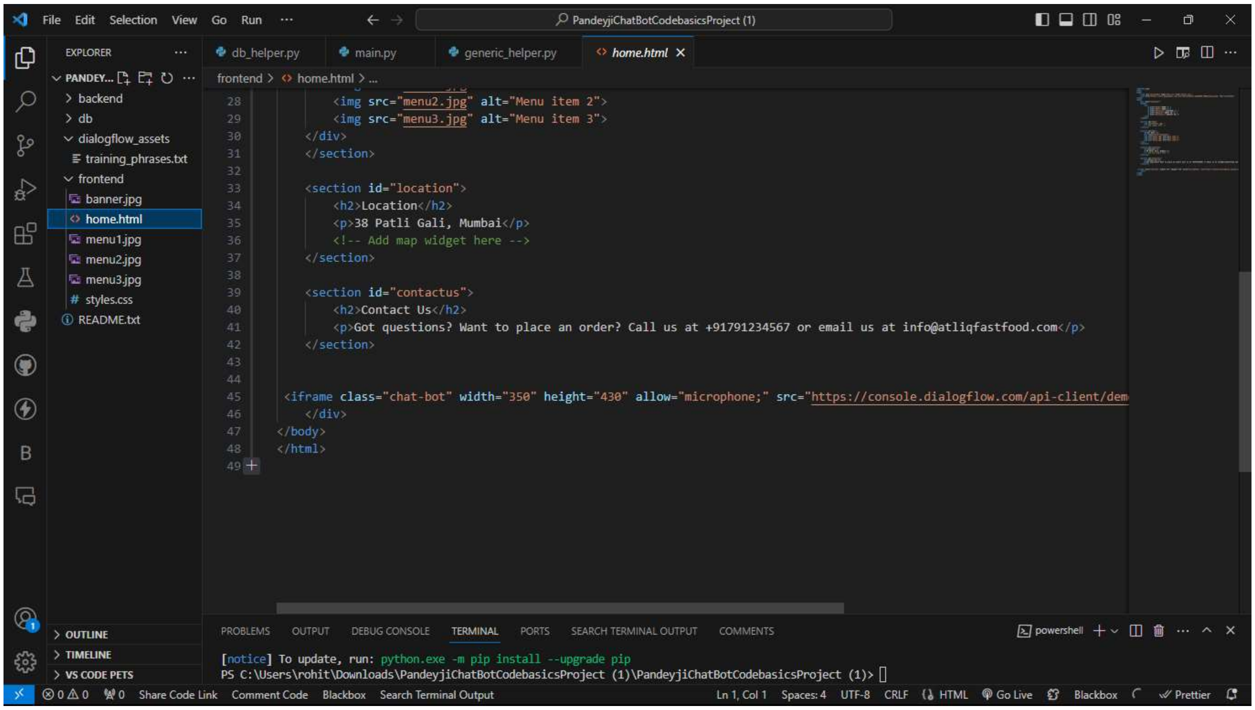


Use following code to integrate this agent into your site:

```
<iframe width="350" height="430" allow="microphone;" src="https://console.dialogflow.com/api-client/demo/embedded/14b2fb32-6045-4b12-b771-af9581bd513c"></iframe>
```

Figure 5.3.28(Integration of Chatbot - code)

After the integration part here what's our code looks like for the frontend part :



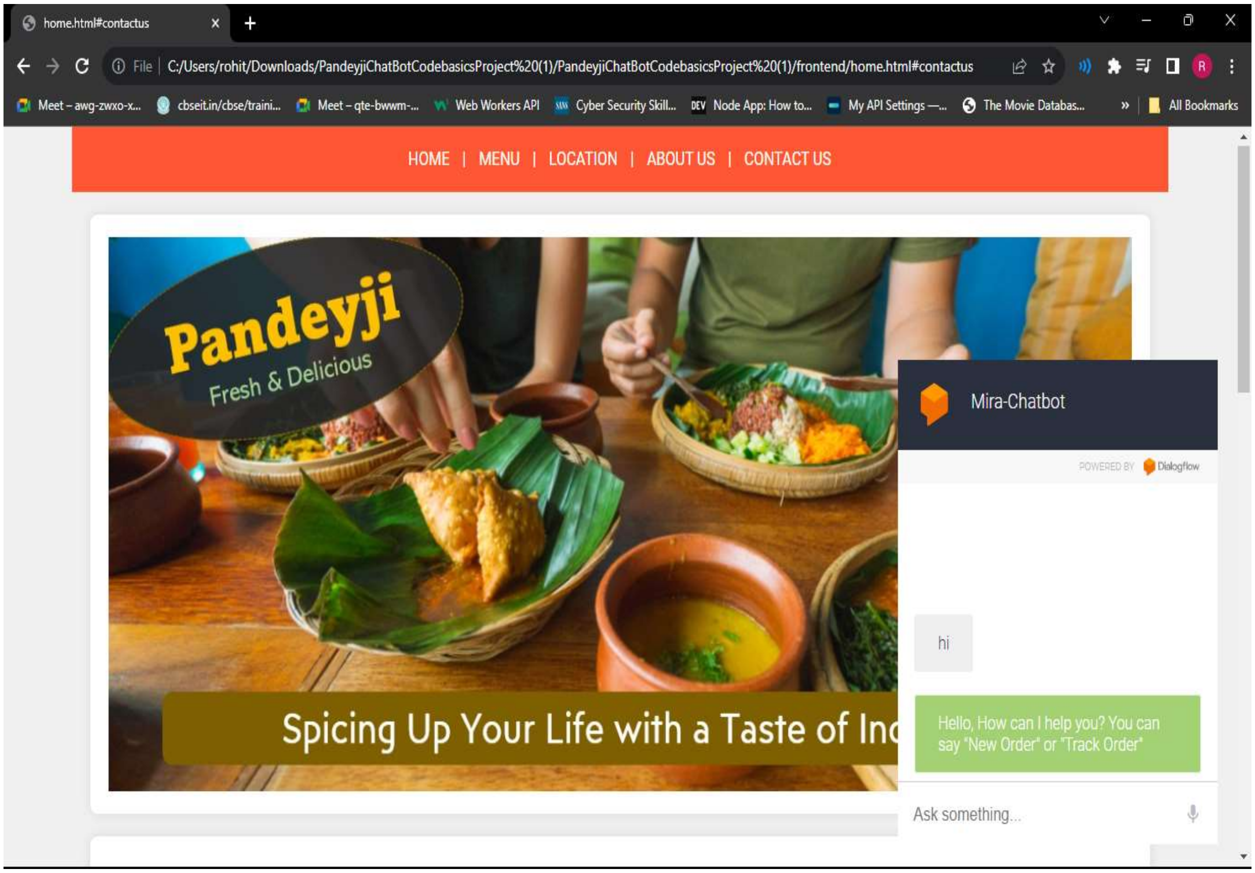
File Edit Selection View Go Run ···
PandeyjiChatBotCodebasicsProject (1)
EXPLORER
PANDEY...
backend
db
dialogflow_assets
training_phrases.txt
frontend
banner.jpg
home.html
menu1.jpg
menu2.jpg
menu3.jpg
styles.css
README.txt
db_helper.py
main.py
generic_helper.py
home.html
frontend > home.html > ...
OUTLINE
TIMELINE
VS CODE PETS
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH TERMINAL OUTPUT COMMENTS
powershell
0 0 0 Share Code Link Comment Code Blackbox Search Terminal Output
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML Go Live Blackbox Prettier

```
            <img src="menu2.jpg" alt="Menu item 2">
            <img src="menu3.jpg" alt="Menu item 3">
        </div>
        </section>

        <section id="location">
            <h2>Location</h2>
            <p>38 Patli Gali, Mumbai</p>
            <!-- Add map widget here -->
        </section>

        <section id="contactus">
            <h2>Contact Us</h2>
            <p>Got questions? Want to place an order? Call us at +91791234567 or email us at info@atliqfastfood.com</p>
        </section>


      <iframe class="chat-bot" width="350" height="430" allow="microphone;" src="https://console.dialogflow.com/api-client/dem
        </div>
    </body>
    </html>
```

```
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\rohit\Downloads\PandeyjiChatBotCodebasicsProject (1)\PandeyjiChatBotCodebasicsProject (1)>
```

Figure 5.3.29(Frontend Code)

THE FRONTEND WEBSITE IS SHOWN BELOW :



hi

Hello, How can I help you? You can say "New Order" or "Track Order"

Ask something...

.

Figure 5.3.30 (Implementation: Chatbot (mira) integrated with Frontend)

In Continuation to the older project several new changes have been implemented to the ongoing project making it more efficient and several learning algos has been taken place.We have learnt new features of MySQL , several features about the Google-Dialogflow platform , expert guidance of our mentor helped us to achieve our heights of the project that would not be possible without him.

CHAPTER – 6: CONCLUSION AND FUTURE SCOPE

The project concluded with the successful completion of the project “End to End NLP Based Chatbot using Dialogflow” which was integrated with a mysql database schema named “Pandeyji Eatery”.This mysql database enhances the user experience . The integration is carried out through Natural Language Processing. The natural language processing helps in extracting the meaningful information and let it have a context-awareness with the chatbot.

The Mysql database acts as a robust backend system , stores all the information about the product , all the information about the order tracking and delivery. This also includes the user history , what price did the user paid for the product/ food item.

Furthermost not the least the integration of the basic HTML webpage , the designing of the poster required a basic knowledge of photoshop.This page not only enhances the accessibility for the users to decide what to order but also increases the visual appearance of the chatbot.

Through this approach NLP tends to play a major role for the project “The PandeyJi Eatery ''. This could only be possible due to the seamless guidance of the Dialogflow , HTML and the Mysql database.

For the future we have decided to move forward with this project that will include the following points:

-Support a payment method for the chatbot
-Track order by the customer name
-Track order by phone number
-Addition of the store hours in the chatbot that would allow user to know when are the visiting hours for the store

A basic flow diagram is shown below which shows the future scope for this project.

FUTURE SCOPE :

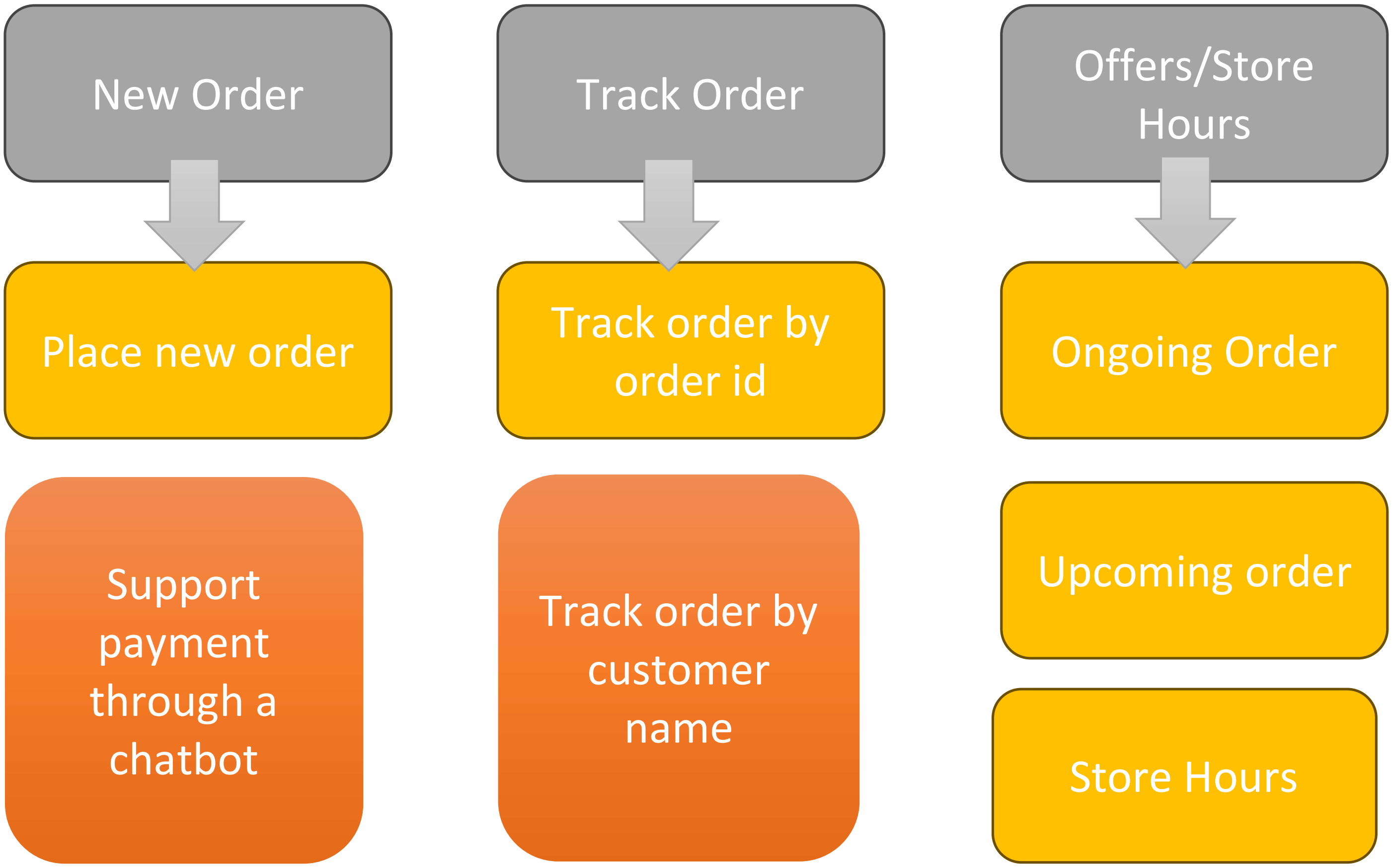


Figure 6.1 (Future Scope)

Support Payment through a chatbot requires a paid payment gateway that was a challenging step for the development of this project at this stage.

REFERENCES

[1] Jain, A., et al. (2019). Chatbots and Conversational User Interfaces: A Survey. arXiv preprint arXiv:1902.07655.

[2] Li, Z., et al. (2020). A Survey of Deep Reinforcement Learning for Dialogue Systems. arXiv preprint arXiv:2006.1[3127.

[3]. Liu, B., et al. (2019). A Survey of Dialogue Systems: Approaches and Trends. arXiv preprint arXiv:1901.03459.

[4]. Luong, M., et al. (2015). Effective Approaches to Attention Mechanism for Neural Machine Translation. arXiv preprint arXiv:1508.04559.

[5]. Parikh, A., et al. (2019). A Survey of Neural Conversation Models. arXiv preprint arXiv:1909.05885.

[6]. Sutskever, I., et al. (2014). Sequence to Sequence Learning with Neural Networks. arXiv preprint arXiv:1409.3215.

[7]. Young, T., et al. (2017). The Stanford Dialogue Dataset: An Annotated Corpus for Interpersonal Dialogue. arXiv preprint arXiv:1706.03626.

[8]. Serban, I. V., et al. (2016). Neural Conversation Modeling: A Review. arXiv preprint arXiv:1506.05869.

[9]. Henderson, M., et al. (2020). The Google Dialogflow Agent: Conversational AI for Mobile and Web Apps. arXiv preprint arXiv:1909.06667.

[10]. Shang, R., et al. (2021). A Literature Survey on End-to-End Neural Chatbots. [Online]. Available: https://arxiv.org/abs/2201.06657: https://arxiv.org/abs/2201.06657: https://arxiv.org/abs/2201.06657: https://arxiv.org/abs/2201.06657

[11]. Banschs, M., et al. (2020). Towards a Robust Evaluation of End-to-End Dialogue Systems. arXiv preprint arXiv:2010.03210.

[12]. Serban, I. V., et al. (2020). Chatbot Research and Applications: A Bibliometric Analysis. arXiv preprint arXiv:2002.12253.

[13]. Ghazvininejad, M., et al. (2020). A Survey on User Satisfaction with Dialogue Systems. arXiv preprint arXiv:2004.00792.

[14]. Duong, N. V., and D. C. Khanh (2020). A Survey of Chatbot Systems and Research. IEEE Access, 8, 42843-42950.

[15]. Serban, I. V., et al. (2019). A Survey of Dialogflow Applications. In A Survey of Generative AI Applications, pp. 1-24.

[16]. Shah, U., et al. (2020). A Survey of Natural Language Processing for Chatbots. arXiv preprint arXiv:2003.01200.

[17]. Li, Z., et al. (2022). A Survey of Open-Domain Chatbots. IEEE Transactions on Artificial Intelligence, 6, 1-20.

[18]. Li, Z., et al. (2021). Deep Learning for Chatbots. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43, 992-1007.

[19]. Shah, U., et al. (2021). End-to-End Dialogue Systems: A Review. ACM Computing Surveys, 54, 1-39.

[20]. Li, J., et al. (2020). A Survey on End-to-End Neural Conversational Systems. arXiv preprint arXiv:2007.16222.

[21]. Sun, S., et al. (2022). A Survey on End-to-End Neural Conversational Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44, 81-107.

[22]. Wang, D., et al. (2021). A Survey on End-to-End Neural Conversational Systems. arXiv preprint arXiv:2109.02136.

[23]. Wen, T., et al. (2022). A Survey on End-to-End Neural Conversational Systems. IEEE Transactions on Artificial Intelligence, 7, 1-20.

[24]. Wu, Z., et al. (2023). A Survey on End-to-End Neural Conversational Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45, 1-29.

[25]. Huang, H., et al. (2023). A Survey on Explainable NLG for Dialogue Systems. IEEE Transactions on Artificial Intelligence, 8, 1-32.

Pranav_Report

ORIGINALITY REPORT

7%	7%	1%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	scholar.ppu.edu Internet Source	2%
2	www.ir.juit.ac.in:8080 Internet Source	2%
3	ir.juit.ac.in:8080 Internet Source	1%
4	pdfslide.tips Internet Source	<1%
5	digitalcollection.utem.edu.my Internet Source	<1%
6	etheses.dur.ac.uk Internet Source	<1%
7	www.geeksforgeeks.org Internet Source	<1%
8	vocal.media Internet Source	<1%
9	Andrés Felipe Solis Pino, Pablo H. Ruiz, Alicia Mon, Cesar Alberto Collazos. "Mechanisms for measuring technology maturity on the	<1%

	Internet of Things in enterprises: A systematic literature mapping", Internet of Things, 2024 Publication	
10	"Computational Intelligence in Communications and Business Analytics", Springer Science and Business Media LLC, 2022 Publication	<1 %
11	Gayathri Rajagopalan. "Chapter 6 Prepping Your Data with Pandas", Springer Science and Business Media LLC, 2021 Publication	<1 %
12	www.coursehero.com Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ....

Type of Document (Tick): **PhD Thesis** | **M.Tech Dissertation/ Report** | **B.Tech Project Report** | **Paper**

Name: ____________________________ __Department: _________________ Enrolment No _________

Contact No. _______________________________E-mail. _______________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

– Total No. of Pages =
– Total No. of Preliminary pages =
– Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

....

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com