A Project Report on

AMIGO: A FRIENDLY CHATBOT

Submitted in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE & ENGINEERING

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
DR. MICHAEL RAJ TF
PROFESSOR

Submitted By

RUPESH KUMAR
18SCSE1010640

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project, entitled **"AMIGO : A FRIENDLY CHATBOT"** in partial fulfilment of the requirements for the award of the Bachelor of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of **JULY, 2021 to DECEMBER, 2021**, under the supervision of **DR. MICHAEL RAJ TF, PROFESSOR**, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me for the award of any other degree of this or any other places.

RUPESH KUMAR

18SCSE1010640

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

DR. MICHAEL RAJ TF

PROFESSOR

CERTIFICATE

The Final Project Viva-Voce examination of RUPESH KUMAR, 18SCSE1010640 has been held on __________________ and his/her work is recommended for the award of BACHELOR OF TECHNOLOGY.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

TABLE_OF CONTENTS

ABSTRACT

A chatbot is AI computer software that can act as a conversation through textual or auditory methods. The core of chatbots analyses a customer's data using the artificial intelligence which integrates the response with them. Different tasks can be replaced with AI-powered bots as they are much more powerful—and are capable of performing multiple tasks at once. Machine Learning techniques are basically used in the process of understanding the input that we get from the user and replying to the user. Natural language processing allows a bot to have a conversation as naturally as possible. The ideal interaction between user and chatbot is a balanced mix of Innovative technology and human Intervention.

Once me and my friend are sitting together and taking about the currencies of the different country and every time, we have to search on google to convert those currencies or data in our country's currency or desirable data, then I thought why don't we make a chatbot which convert the data for user whenever we wanted to do so.

So, I decided to create a chatbot which can convert any country's currencies into different currencies. This chatbot can be like a friend just like we are chatting with our friend on WhatsApp or Telegram. We just have to give it a message as a task and it can give us the output message which we wanted to do so.

Ex. Input:" Hey, can you convert 400 USD to INR."
Output: "400 USD is 30156.40 INR."

The tools I have used in My project are:
"PyCharm" for the python coding.
"Dialogflow" which is a Google's product for training my chatbot.
"Heroku", which is platform to make the product online.
"Telegram", to deploy our chatbot to chat.

The purpose of making this project is to provide everyone a solution of the problem mentioned above. So, the result of this project is a chatbot which can convert any type of data into our desirable data.

And finally, I came up with my friendly chatbot. The chatbot is friendly because we can chat with it as friend also. Currently this is a currency converter chatbot and in future I make sure it can convert the distances, lengths from meters to centimetres and many more.

TABLES

Table for Student Data:

S. No	Name	Enrolment Number	Admission Number	Program / Branch	Sem
1	RUPESH KUMAR	18021011865	18SCSE1010640	B. Tech/ CSE	7

Faculty data:

Guide name: Dr. Michael Raj TF

Designation: Professor

INTRODUCTION

In today world most of the technologies we are using like: ZOMATO, SWIGGY, AMAZON, FLIPCART etc. are using chatbot to deal with their customer problem, So I decided to work on the project related to that concept.

Chatbot is basically an AI which can chat with us both via text messages and voice messages as well. A chatbot can conduct smart conversation—either via text or voice. They are armed with machine learning which can interact with humans and become increasingly agile with each interaction. It recognizes using pattern matching, user input as well to access information to provide a predefined acknowledgment. In dialog systems they are used for numerous practicals intends comprising information acquisition or customer service. Keywords are scanned with the input in simple chatbot and then respond with the most similar matching keywords or patterns from a database while some chatbots use sophisticated. Their applications make the communication between people and services, intensify the experience of customer. To have better customer engagement and operational efficiency they provide companies new opportunities by lowering the cost of customer service.

Types of chatbots used:

- Support – This is used to master a single domain.
- Skills - This does not require a lot of contextual awareness.
- Assistant - This is the middle ground between a skill and support chatbot. When they know a little about a variety of topics they work great.

The main aim of this project is to create a “DATA CONVERTER CHATBOT” Which can convert the currency of any country in other currency in single command, and it can chat with us too both via text messages and voice message as well. In user side there is “CHATBOT” on any platform like: WhatsApp, Telegram or any website etc. We just have to give command to it, then chatbot understand the intent of user and extract the entity, this entity sends to the AI where conversion done and result sent back to the chatbot and chatbot reply it to the user.

We can chat normally with the chatbot as a friend that is why I named this bot as “AMIGO” which is the Spanish word of friend.

In future, demands of chatbot going to be increase because it can reduce the cost of company which they are giving to their employees who are handling the problem of users in call centres.

LITERATURE SURVEY

1. **“Recruitment Chatbots”, International Research Journal of Engineering and Technology (IRJET), vol. 5, Issue: 08, Aug 2018[1].**
   Authors: Akash Balachandar, Anusha D Kulkarni

In this paper, authors have explained how the chatbot behaving as a human conversational partner are designed to comprehend a conclusive human response. In today’s world, it is difficult to collect correct information easily while hiring the right candidate. Using simply a chatbot can be a solution to this problem. Recruiters can use this in day-to-day life to automate timeconsuming tasks.

SYSTEM DESIGN

Describing the designing process of interaction between the chatbot and the user. It uses dialogues systems, and they are of two types:

1) Goal Oriented Dialogue Systems.

2) General conversation Dialogue Systems.

We use Generative and Selective approaches in recruitment chatbot which needs a general conversational dialog system. The Machine Learning principle is a core philosophy for both these approaches: Build it, train it, and test it. By using bot characteristics, constraints, dialogue dataset, access flow, and Sequence tokens this model is built. Figure.1: Data-driven Dialogue System.

2. **Implementation of Chatbot in Online Commerce, and Open Innovation. Authors: María D. Illescas-Manzano, Noé Vicente López, Nuno Afonso González and Carmen Cristofol Rodríguez.**

In this paper the study describes the chatbot journey and focuses on its implementation within the digital marketing strategy in the first part of a company’s sales funnel. The main goal was to apply a chatbot via Facebook Messenger supported by the ManyChat platform to increase the number of leads, comparing the chatbot with the previous strategy used by the company to obtain contact information. This research work takes a step further and shows that implementing a chatbot through the ManyChat platform by a company that markets online has a positive impact on the capturing of leads, as opposed to the results obtained by authors such as Luo et al. and Leung et al. A chatbot platform used with the intention of obtaining leads seems to be an agile and powerful tool; in fact; the main conclusion of this work is that including this method can be one of the main axes of obtaining information about consumers with the aim of performing marketing actions in a two-way communication that facilitates sales by companies.

3. **Literature survey on various chatbot.**
   Authors: Harsha Pariyani, Anshika Sinha, Preeti Bhat, Roshni Rote, Asst. Prof. N. A. Mulla.

In this paper they describe the various study about chatbots like what are chatbot? how they can communicate? How their working done? What are their types?
A/c to them a chatbot is AI computer software that can act as a conversation through textual or auditory methods. In simple words a chatbot can conduct smart conversation either via text or voice. The core of the chatbots analyses a customer's data using the AI which gives responses to them. They are trained through various data using machine learning which can interact with humans and become more accurate with each interaction. They divide the chatbot in categories on the basis of what are support, what are their skills and what type of work they assist.

In support part they use to master a single domain.

A skill based chatbot does not require a lot pf contextual awareness. And an assistant chatbot is the middle of between a skill and support chatbot. They know a little about the variety of topics.

4. **"Intelligent chatbot for easy web-analytics".**
   Author: Ravi R.

In this paper, a comparison is done among chatbots based on their ease of usage, by using different analytic tools. They describe that the chatbot is build using Artificial Intelligence Markup Language contains analytics' raw data and the required data is fetched from analytics tool's raw data. They said every website note all the details user made. AIML comprises of possible queries and their responses. It consists of 3 elements such as template, categories and pattern. Each category contains pattern and a template. Patterns are the possible queries that the bot-user may type in and the template is the response to the respective pattern. They describe 3 types of queries can be considered Domain related query, General Query and None of them. The user can type to web analytics their query related to and will get an immediate reply. This chatbot tools are mastered to avoid the time taking task.
They have also mentioned some application of chatbot which are:

- *Experience and Services: Communicate in multiple language, can handle customer service request, manage account setting.*
- *Recruitment: Scheduling meeting, Candidate's interaction, Answer & Questions.*
- *Healthcare: Provide status on staffing, Communication with doctors.*
- *Government: Resolve parking tickets, Visa applications management.*
- *Smart Home Devices: control entertainment devices, control heating, manage security.*
- *Transportation: Remotely lock vehicles, retrieve information.*
- *Personal Assistant: Manage payment, search for information.*
- *Sales & Marketing: Make recommendation, and offer loyalty incentives, Deliver relevant content.*
- *Employee Productivity: Manage calenda r, manage email, plan resources, Search.*

REQUIREMENTS

Concept Used:

- Machine Learning

Coding Platforms:

- PyCharm Community Edition IDE

Training Platform:

- Dialogflow

Languages Used:

- Python

Deployment Platform Used:

- Heroku

DESCRIPTION OF REQUIREMENTS

MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, an approach based on semantic analysis mimics the human ability to understand the meaning of a text.

Some Machine Learning Methods

Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labelled and unlabelled data for training – typically a small amount of labelled data and a large amount of unlabelled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labelled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabelled data generally doesn’t require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

PyCharm Community

PyCharm is a hybrid-platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development. Some of the unicorn organizations such as Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE!

It supports two versions: v2.x and v3.x.

We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

Features of PyCharm:

1. Intelligent Code Editor:

- It helps us write high-quality codes!
- It consists of colour schemes for keywords, classes, and functions. This helps increase the readability and understanding of the code.
- It helps identify errors easily.
- It provides the autocomplete feature and instructions for the completion of the code.

2. Code Navigation:

- It helps developers in editing and enhancing the code with less effort and time.
- With code navigation, a developer can easily navigate to a function, class, or
- file.
- A programmer can locate an element, a symbol, or a variable in the source code within no time.

- Using the lens mode, further, a developer can thoroughly inspect and debug the entire source code.

3. Refactoring:

- It has the advantage of making efficient and quick changes to both local and global variables.
- Refactoring in PyCharm enables developers to improve the internal structure without changing the external performance of the code.
- It also helps split up more extended classes and functions with the help of the extract method.

4. Assistance for Many Other Web Technologies:

- It helps developers create web applications in Python.
- It supports popular web technologies such as HTML, CSS, and JavaScript.
- Developers have the choice of live editing with this IDE. At the same time, they can preview the created/updated web page.
- The developers can follow the changes directly on a web browser.
- PyCharm also supports AnglularJS and NodeJS for developing web applications.

5. Support for Popular Python Web Frameworks:

- PyCharm supports web frameworks such as Django.
- It provides the autocomplete feature and suggestions for the parameters of Django.
- It helps in debugging the codes of Django.
- It also assists web2py and Pyramid, the other popular web frameworks.

6. Assistance for Python Scientific Libraries:

- PyCharm supports Python’s scientific libraries such as Matplotlib, NumPy, and Anaconda.
- These scientific libraries help in building projects of Data Science and
- Machine Learning.

- It consists of interactive graphs that help developers understand data.
- It is capable of integrating with various tools such as IPython, Django, and Pytest. This integration helps innovate unique solutions.

Python

Nowadays, Python is in great demand. It is widely used in the software development industry. There are 'n' number of reasons for this.

High-level object-oriented programming language: Python includes effective symbolism.

Rapid application development: Because of its concise code and literal syntax, the development of applications gets accelerated. The reason for its wide usability is its simple and easy-to-master syntax. The simplicity of the code helps reduce the time and cost of development.

Dynamic typescript: Python has high-level incorporated data structures blended with dynamic typescript and powerful binding.

Features of Python:

- Python supports code reusability and modularity.
- It has a quick edit-inspect-debug cycle.
- Debugging is straightforward in Python programs.
- It has its own debugger written in Python itself, declaring to Python's reflective power.
- Python includes a plethora of third-party components present in the Python Package Index (PyPI).

Heroku

Heroku is a cloud service platform whose popularity has grown in recent years. Heroku is so easy to use that it's a top choice for many development projects.

With a special focus on supporting customer-focused apps, it enables simple application development and deployment. Since the Heroku platform manages

hardware and servers, businesses that use Heroku are able to focus on perfecting their apps. And not the infrastructure that supports them.

Heroku, a Platform-as-as-Service solution, is generally easy-to-use. But it's most beneficial to businesses in specific situations. Heroku has a free service model for small projects. Also, tiered service packages exist for cases where more complex business needs must be addressed.

The Heroku cloud service platform is based on a managed container (called dynos within the Heroku paradigm) system. It has integrated data services and a powerful ecosystem for deploying and running modern applications.

Features of Heroku:

1. Heroku Accommodates Many Development Languages:

Heroku supports several programming languages that are used as a web application deployment model. As one of the first cloud platforms, Heroku has been in development since June 2007. Back then, it supported only the Ruby programming language.

But now it also supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. This

means a variety of developers can look to Heroku for an inexpensive way to scale their application, no matter their preferred development language.

2. Heroku Supports Diverse Solutions:

Heroku also provides custom buildpacks, where developers can deploy apps in any other programming language. For this reason, Heroku is a polyglot platform. It lets the developer build, run, and scale applications in a similar manner across all programming languages.

Polymorphism and scalability are reasons why Heroku is often seen as a preferred platform amongst developers.

3. Heroku Dynos Enable Easy Development and Better Usability:

Applications that are run on Heroku typically have unique domain names, which are used to route HTTP requests to the correct container. Applications as services use application containers. Containers are designed to package and run services. Each of the application containers is a smart container on a reliable, fully-managed runtime environment.

4. Heroku Lets Developers Scale Applications Instantly:

This is accomplished either by increasing the number of dynos or by changing the type of dyno in which the app runs. When the application can scale so easily, the user can always expect more speed when using that application.

FEASIBILITY ANALYSIS

QUITE
FEASIBLE

FREE
OF
COST

PYTHON
BASED
PROJECT

EASY
TO
USE

WORK
VERY
SMOOTHLY

DEPLOYED
WITH
HEROKU

OBJECTIVE

- The main aim of this project is to create a “DATA CONVERTER CHATBOT” Which can convert the currency of any country in other currency in single command, and it can chat with us too both via text messages and voice message as well.

- Purpose of creating this chatbot is to provide easy way and basic facilities which the user’s need.

- With this system, the tasks would be performed in appropriate amount of time and more efficiently.

- Additional benefit is that we can chat normally with the chatbot as a friend that is why I named this bot as “AMIGO” which is the Spanish word of friend.

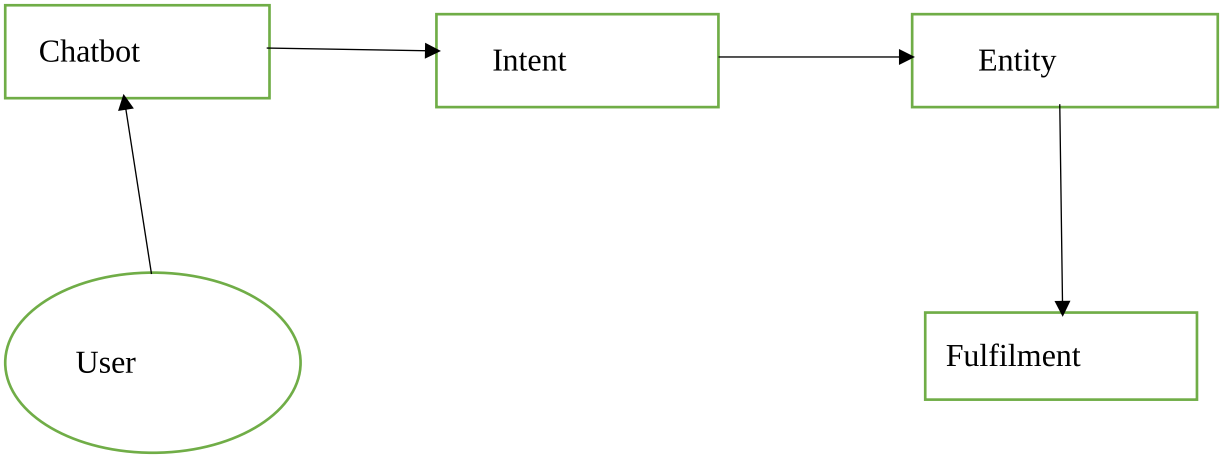
- When user enters the message or intent what they wanted to be done, the output will be their desirable result.

- The main purpose is to provide the basic facilities which the users need.

- With this system, the tasks would be performed in appropriate amount of time and more efficiently.

Functionality & Design

In this project Dialogflow is used. It is a chatbot building framework. The query done by user and according to that the chatbot extract the intent of user.



- User chat with the chatbot.
- In intent part query done by the user. (Intent means user telling to chatbot or want some work done by the chatbot.
- Behind the scene chatbot extract the entity from intent.
- Depending on the chatbot entity can be one or more.
- Most important stage is fulfilment, in this stage we are making an app using python or flask API. API which work is to do conversion of currency.
- Flask API do conversion and result send to chatbot as a reply.
- This how the entire things will work.
- So, in user side there is a chatbot, which can be on any platform like: WhatsApp, telegram or any website etc.
- So, the user chat with chatbot, then chatbot understand the intent and extract the entity and send to our flask web app where conversion done and result sent back to user as a reply.

IMPLEMENTATION

For making a chatbot, we have to know the deep knowledge of AI & ML concepts, which is difficult to learn in less time. If someone have knowledge of it then that's very good if not then there is a chatbot framework "Dialogflow" which is a Google's product. By using it we can make and train the chatbot easily.

DIALOGFLOW

In beginning dialogflow was a company API-AI later it acquires by Google and named as Dialogflow. Its work is to provide NLU (Natural Language Understanding) by using this chatbot can be create easily, we don't have to train that from scratch. That is the best part of Dialogflow.

It is not the only service for creating chatbot, there are many more services of others companies are also available.

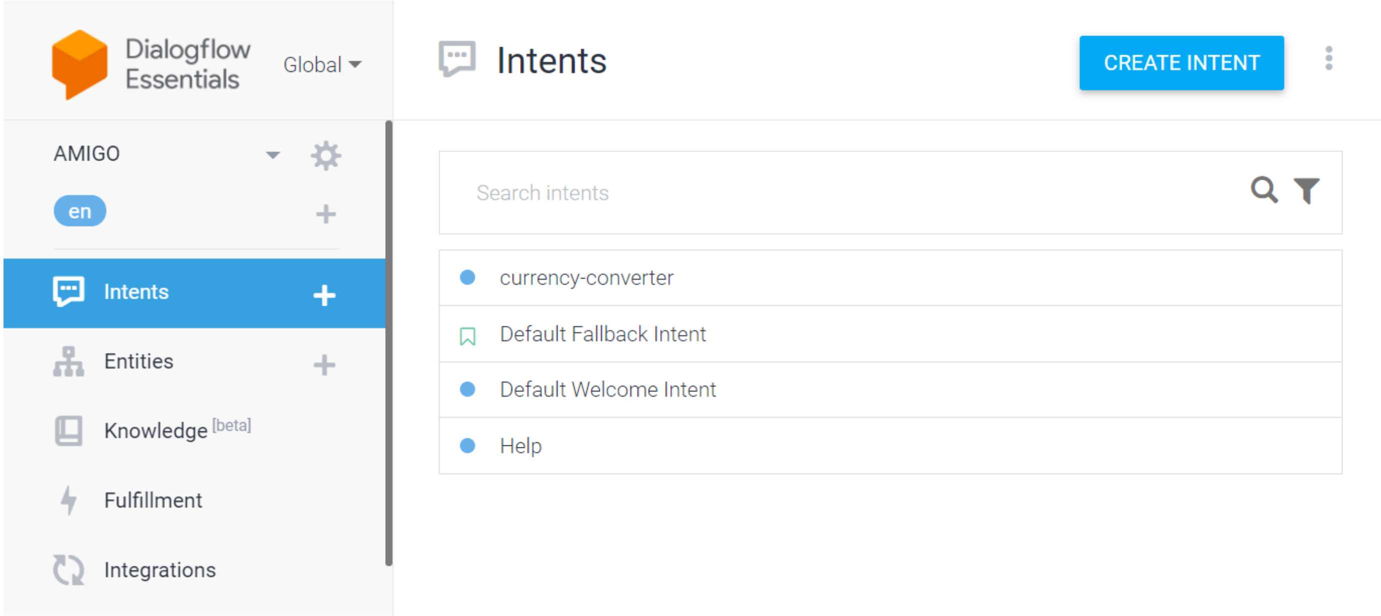
Like: AWS Lex by Amazon.
    Wit.AI by Facebook.
    Whatson by IBM.
    Azure by Microsoft.

But dialogflow is easy and provide good result.

FIG- 01

- Figure 01 show the Dilogflow website.
- First, we have to login on Dialogflow window using our google account.
- Then make an Agent and named it AMIGO (because the chatbot named as AMIGO).
- Default language is English, Default time zone.
- As soon as we make our agent, a project will create on google cloud.
- After naming and clicking on create our agent is ready.
- Now we have some option of stages Intents, Entities, Knowledge, Fulfilment, Integration, Training, Validation, History, etc.

INTENT

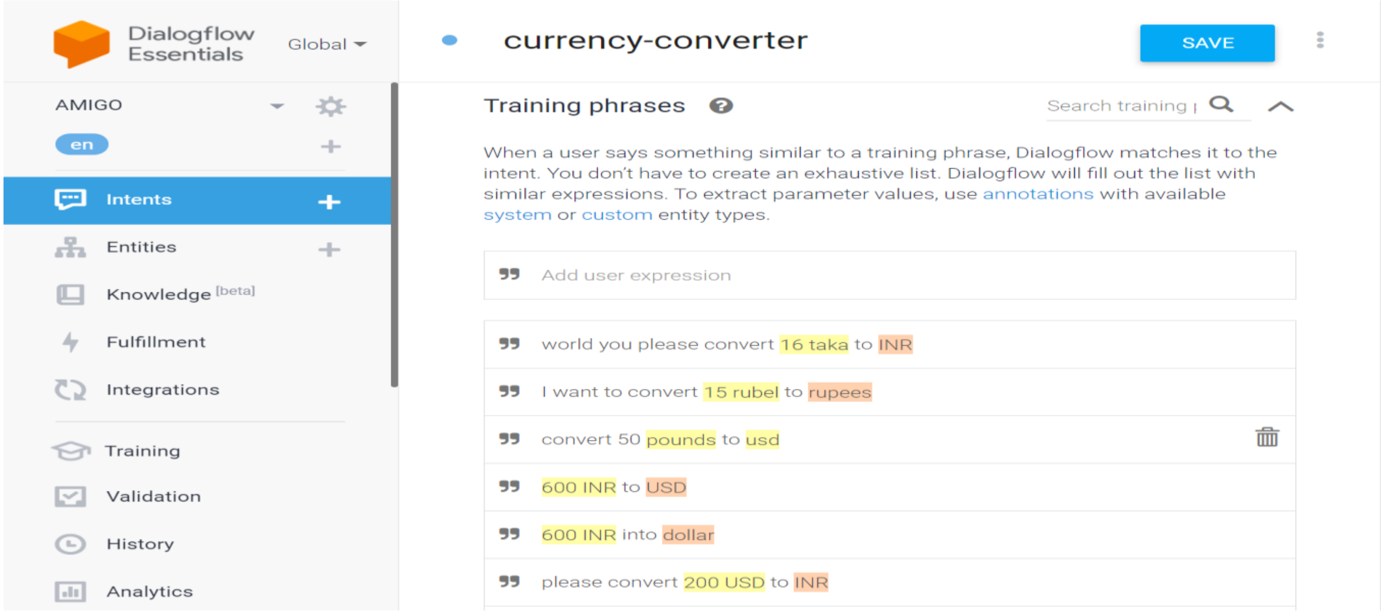


- currency-converter
- Default Fallback Intent
- Default Welcome Intent
- Help

Fig- 02

- Figure 02 show us the intent part of our project.

- According to dialogflow intents are mapping between a user’s queries and actions fulfilled by the software.
- In fig-02 we can see there are two inbuild intents in our project one is Default fallback intent and second is Default welcome intent.
- Fallback intent is trigger when chatbot does not understand what to reply.
- Then we have to create out intent like help intent.

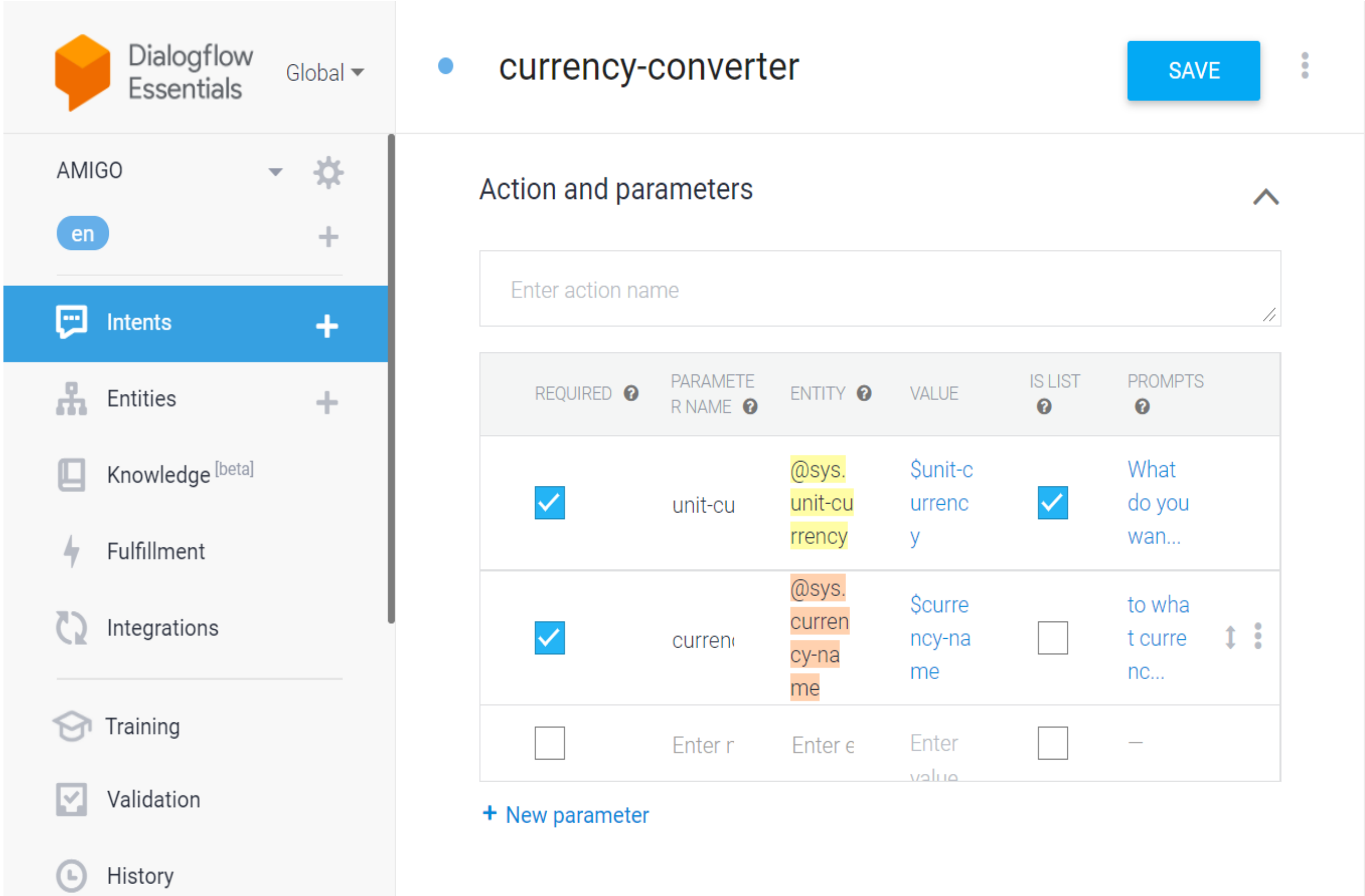


When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don’t have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use annotations with available system or custom entity types.

- world you please convert 16 taka to INR
- I want to convert 15 rubel to rupees
- convert 50 pounds to usd
- 600 INR to USD
- 600 INR into dollar
- please convert 200 USD to INR

Fig- 03

- In figure 03, we can see there is training phrases where we have to train our chatbot by providing phrases.
- There is a section *“add user expression”* ,where we can type our training phrases.
- Then train the phrases, phrases we can expect from user, that will trigger the intent.
- We have to provide how users will express their intent in natural language, adding numerous with different variations and parameters will improve the accuracy of intent matching.



currency-converter

Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
☑	unit-cu	@sys.unit-currency	$unit-currency	☑	What do you wan...
☑	curren	@sys.currency-name	$currency-name	□	to what currenc...
□	Enter r	Enter e	Enter value	□	–

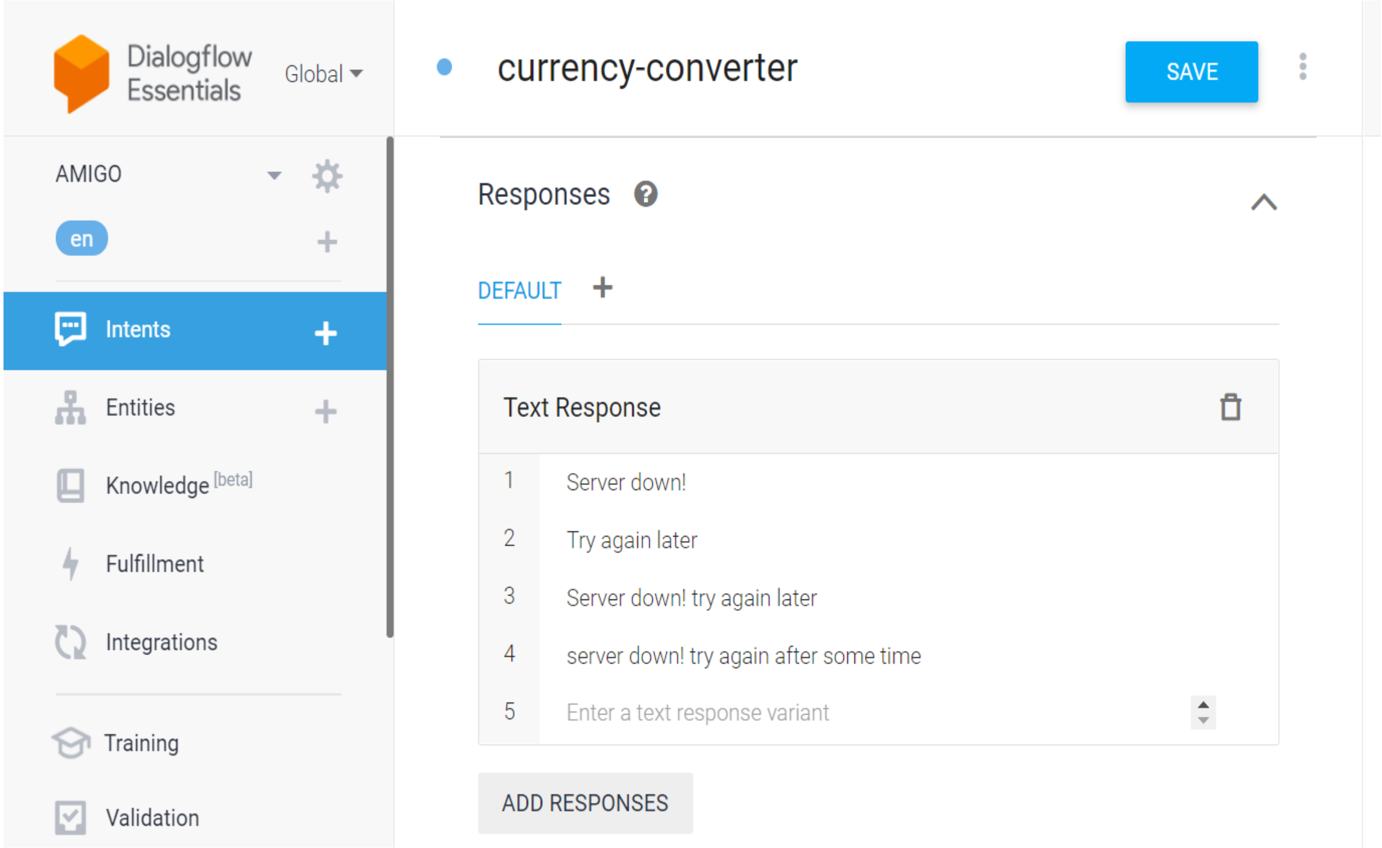
Fig-04

- In figure 04, there is a section of Action and parameters, it is the entity extracted by the chatbot from the intent of the user.

- This entity is very useful, it gives us dictionary of unit currency and name currency. By using this we can find the conversion factor.

- This conversion factor gets multiplied to the amount which we wanted to convert which gives us final converted amount.

- Also, there is section in Action and parameters called prompts, if user does not provide the unit currency, then it asks for the input, "*What do you want to convert?*".
- And if user does not provide the currency name, then it asks "*to what currency do want to convert?*".



currency-converter

Text Response

1 Server down!
2 Try again later
3 Server down! try again later
4 server down! try again after some time
5 Enter a text response variant

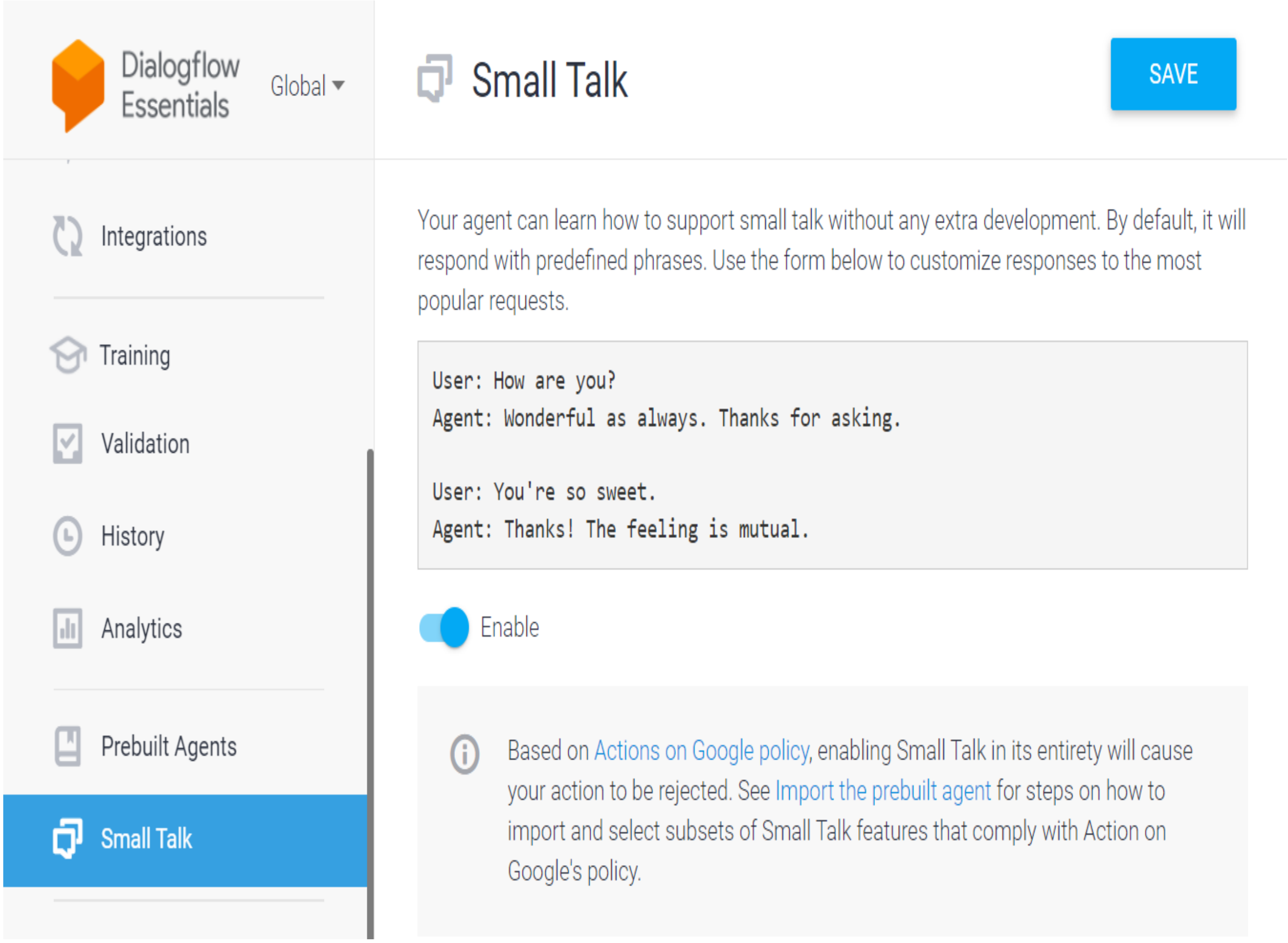
Fig- 05

- In this figure there is a section of responses where we have to add responses according to the intent, if user input something which is not recognised by the chatbot, then instead of empty responses it can responses as "*try again later.*", "*Server down.*"
- After adding phrases, we have to add responses.

#Giving personality to our chatbot:

- Like SIRI is witty, means it gives answer smartly.
- By using small talk, we can also personalize our chatbot.
- There is multiple reason to give personality to chatbot, that because company wants to use chatbot as an employee in the call centre, due to sometimes user get know that the person to whom they are chatting is chatbot then the flow of chat was messed up, so it's important to give personality to the chatbot.

SMALL TALK



Small Talk

Your agent can learn how to support small talk without any extra development. By default, it will respond with predefined phrases. Use the form below to customize responses to the most popular requests.

```
User: How are you?
Agent: Wonderful as always. Thanks for asking.

User: You're so sweet.
Agent: Thanks! The feeling is mutual.
```

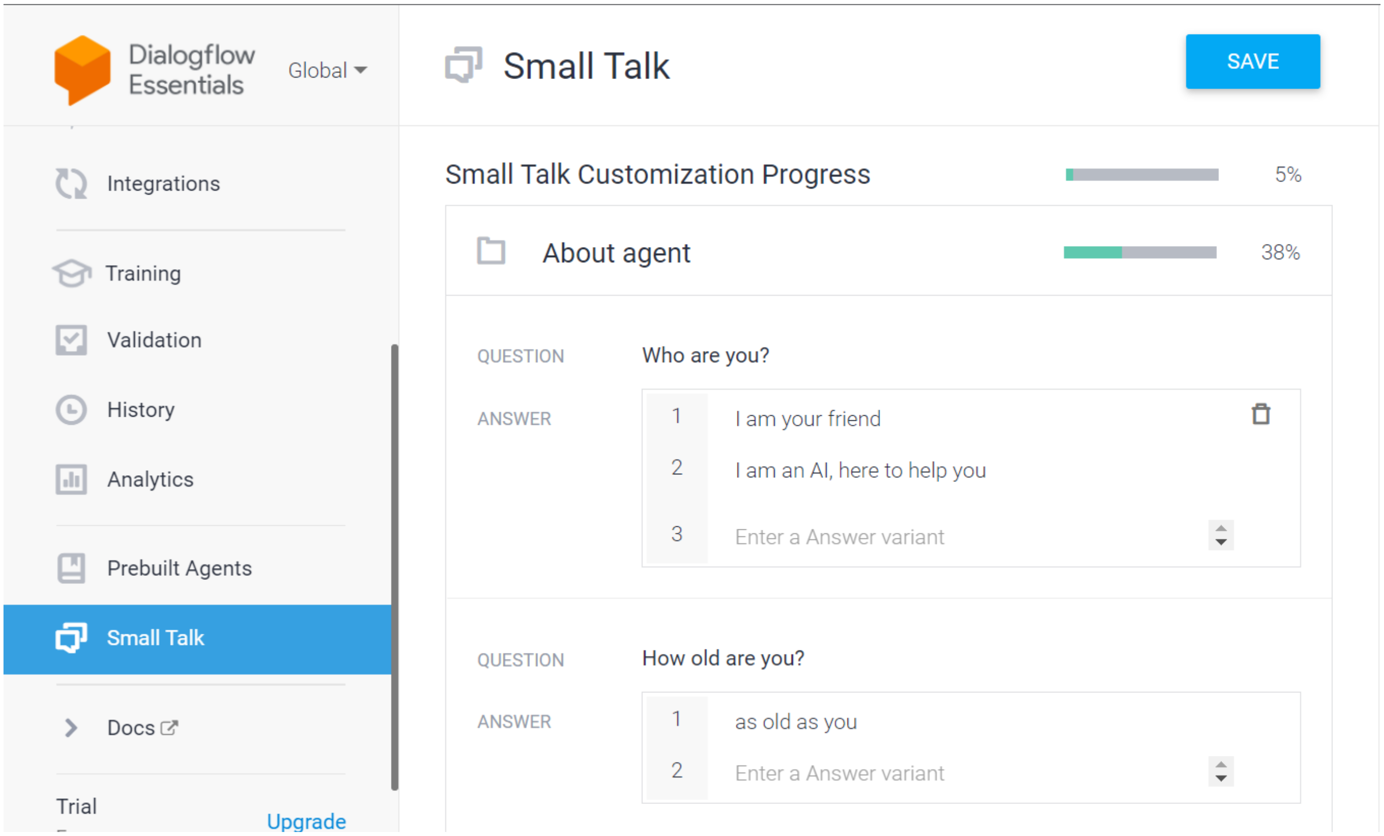
Enable

Based on Actions on Google policy, enabling Small Talk in its entirety will cause your action to be rejected. See Import the prebuilt agent for steps on how to import and select subsets of Small Talk features that comply with Action on Google's policy.

Fig- 06

- In figure 06 we can see there is a small talk section where we can give personality to our chatbot.
- Our agent can learn how to support small talk without any extra development. By default, it will respond with predefined phrases.
  - Ex: User- “*How are you?* “

    Agent- *“Wonderful as always, thanks for asking.* “



Small Talk

Small Talk Customization Progress 5%

About agent 38%

QUESTION Who are you?

ANSWER
1 I am your friend
2 I am an AI, here to help you
3 Enter a Answer variant

QUESTION How old are you?

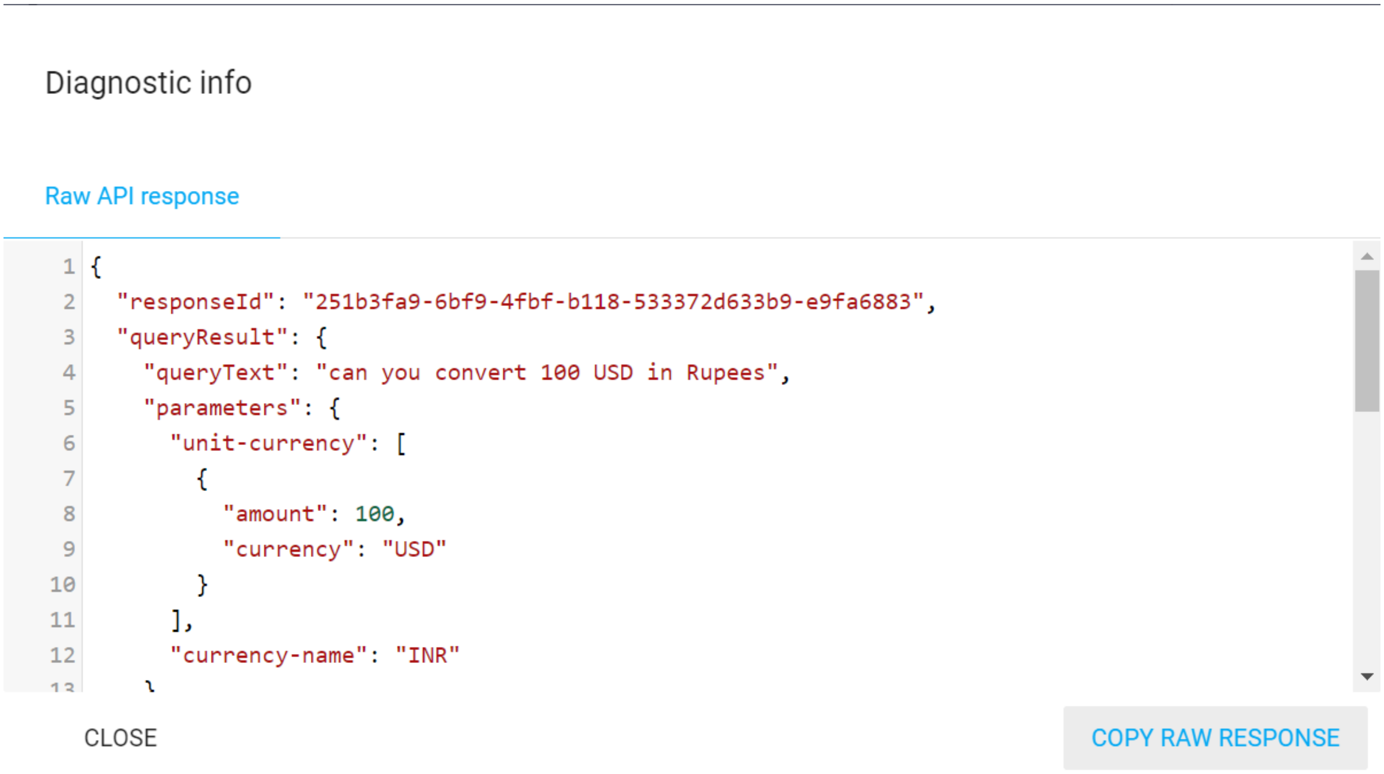
ANSWER
1 as old as you
2 Enter a Answer variant

Fig- 07

- Small talk customization process as seen in figure 07 like: Who are you? How old are you? Answer my question? And then SAVE.
- Now we will make an intent which convert currency as sawn in figure 03. The name of this intent is “Currency-converter” and add some training phrases.
  - Ex: *Can you convert 500 rupees in USD.*
    Please convert 200 pounds to rupees.
- Dialogflow extract the intent from message and store it in a dictionary. Like if message is “would you please convert 100 USD to INR”.

Parameter	value
Currency name	INR
Unit currency	{‘currency’:’USD’; ‘amount’: 100}

- And this dictionary value sends to flask API which convert the currency.
- Unfortunately, if the flask API server is down, then instead of not getting any response, we can add a default response in our intent.
  - Ex: *“Server down! Try again after some time.”*

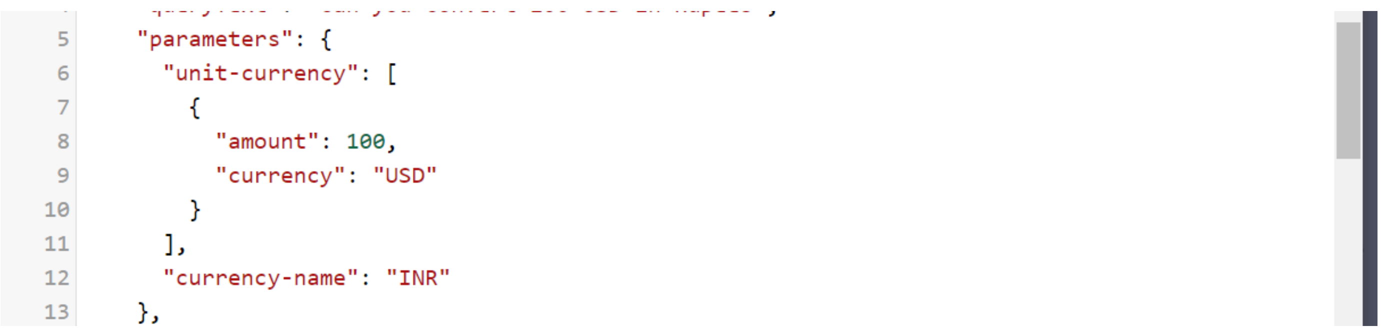


Diagnostic info

Raw API response

```
{
  "responseId": "251b3fa9-6bf9-4fbf-b118-533372d633b9-e9fa6883",
  "queryResult": {
    "queryText": "can you convert 100 USD in Rupees",
    "parameters": {
      "unit-currency": [
        {
          "amount": 100,
          "currency": "USD"
        }
      ],
      "currency-name": "INR"
    }
```

Fig- 08

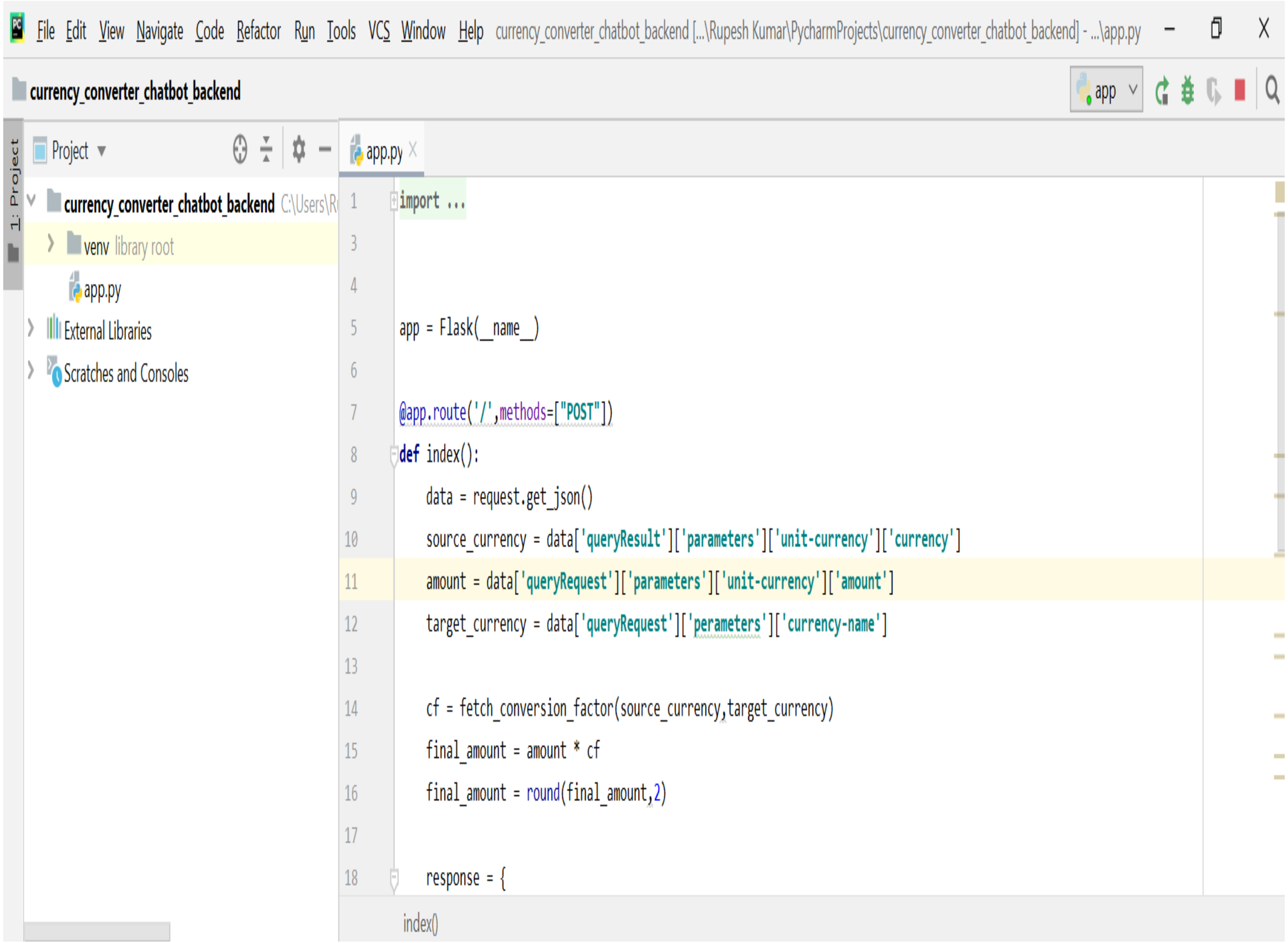


```
    "queryText": "can you convert 100 USD in Rupees",
    "parameters": {
      "unit-currency": [
        {
          "amount": 100,
          "currency": "USD"
        }
      ],
      "currency-name": "INR"
    },
```

Fig - 09

- To attached our API to our Python App, we need a JSON object.
- The JSON object is a list of name-value pairs surrounded in curly braces, in simple word JSON object are written in key-value pairs.
- Keys must be strings, and values must be a valid JSON data types like: String, number, object, array, Boolean or NULL.
- We can see JSON response code in figure 08, which will be use in our backend flask API.
- To add this, we only need the parameter not the whole JSON code.
- We copy the useful response than paste it in the API code.

#Now In PyCharm, we will start our project.

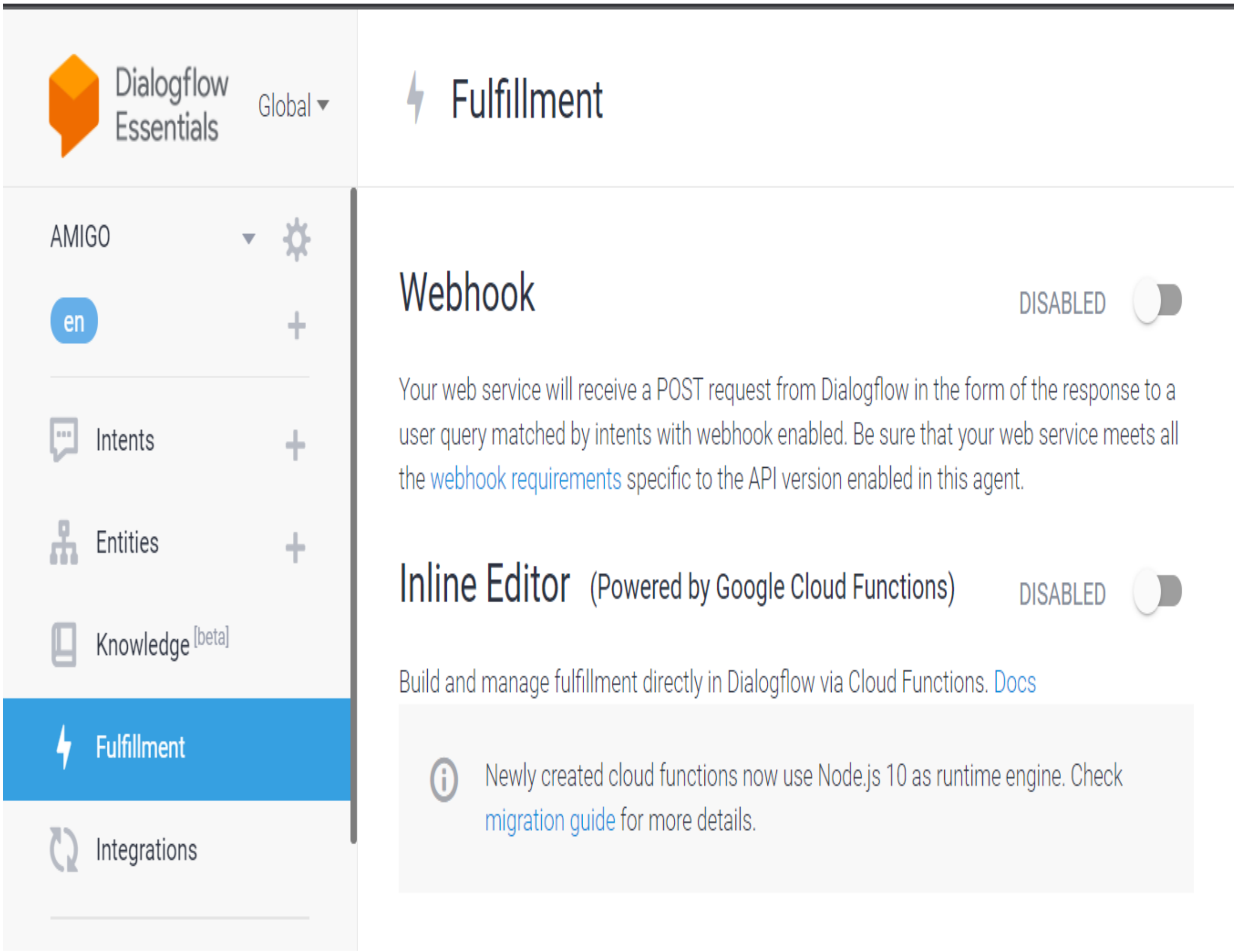

File Edit View Navigate Code Refactor Run Tools VCS Window Help currency_converter_chatbot_backend [...\Rupesh Kumar\PycharmProjects\currency_converter_chatbot_backend] - ...\app.py
currency_converter_chatbot_backend
app
1: Project
Project
currency_converter_chatbot_backend C:\Users\R
venv library root
app.py
External Libraries
Scratches and Consoles
app.py
index()

```
import ...


app = Flask(__name__)

@app.route('/',methods=["POST"])
def index():
    data = request.get_json()
    source_currency = data['queryResult']['parameters']['unit-currency']['currency']
    amount = data['queryRequest']['parameters']['unit-currency']['amount']
    target_currency = data['queryRequest']['perameters']['currency-name']

    cf = fetch_conversion_factor(source_currency,target_currency)
    final_amount = amount * cf
    final_amount = round(final_amount,2)

    response = {
```

Fig – 10

- The main problem is the communication between our offline app and the online google framework dialogflow, when the is underdevelopment period.
- After development it can deploy online on Heroku or AWS.
- So, for underdevelopment period “ngrok” software will be use to establish the communication.
- “ngrok” is a software which make the app online for 8 hours, while the app is underdevelopment.
- Then download “ngrok”. After downloading extract “ngrok” from zip file and install it then open it.
- Now type command as given example on the app.
    - Ex: ngrok http 5000, then enter.
    - (5000 is given because the app is running on 5000 port).
- It gives us two URL, we have to copy to secure URL and paste it into the browser then the app will get online.

- Hence, we have understood the working of ngrok and the app is online. Now we have to connect the app to the chatbot.
- For connecting chatbot with the python app.



Fulfillment

Webhook DISABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the webhook requirements specific to the API version enabled in this agent.

Inline Editor (Powered by Google Cloud Functions) DISABLED

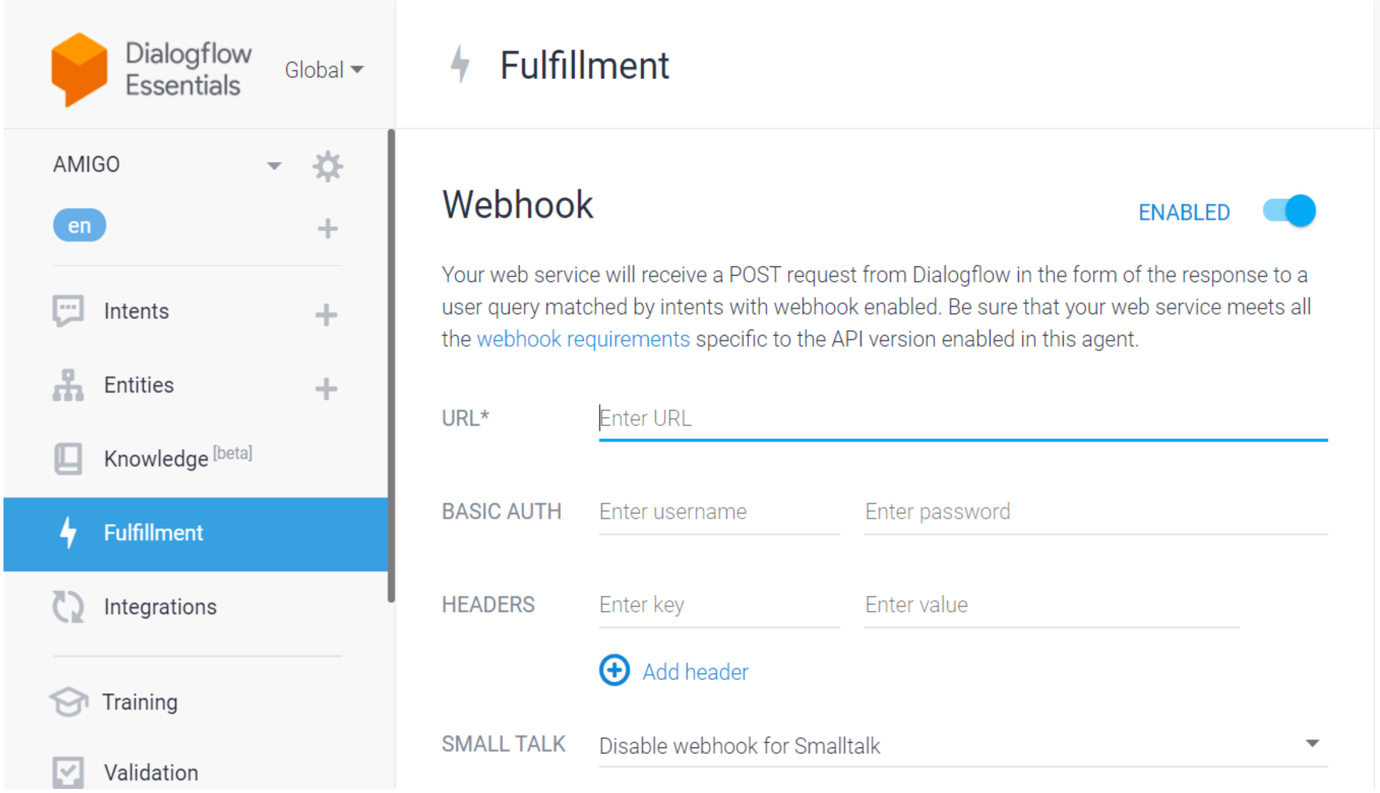
Build and manage fulfillment directly in Dialogflow via Cloud Functions. Docs

Newly created cloud functions now use Node.js 10 as runtime engine. Check migration guide for more details.

Fig - 11

- Open dialogflow window.
- Go to fulfilment section as sawn in figure 10. {fulfilment is done by the flask application.}
- In this section, enable ‘Webhook’ {The code which is written in the backend flask app this is called ‘webhook’ in dialogflow.}
- After enabling, paste the URL of ngrok in the URL section.
- And after that we have to enable fulfilment section in every intent which we have created.

WEBHOOK



Webhook

ENABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the webhook requirements specific to the API version enabled in this agent.

URL* Enter URL

BASIC AUTH Enter username Enter password

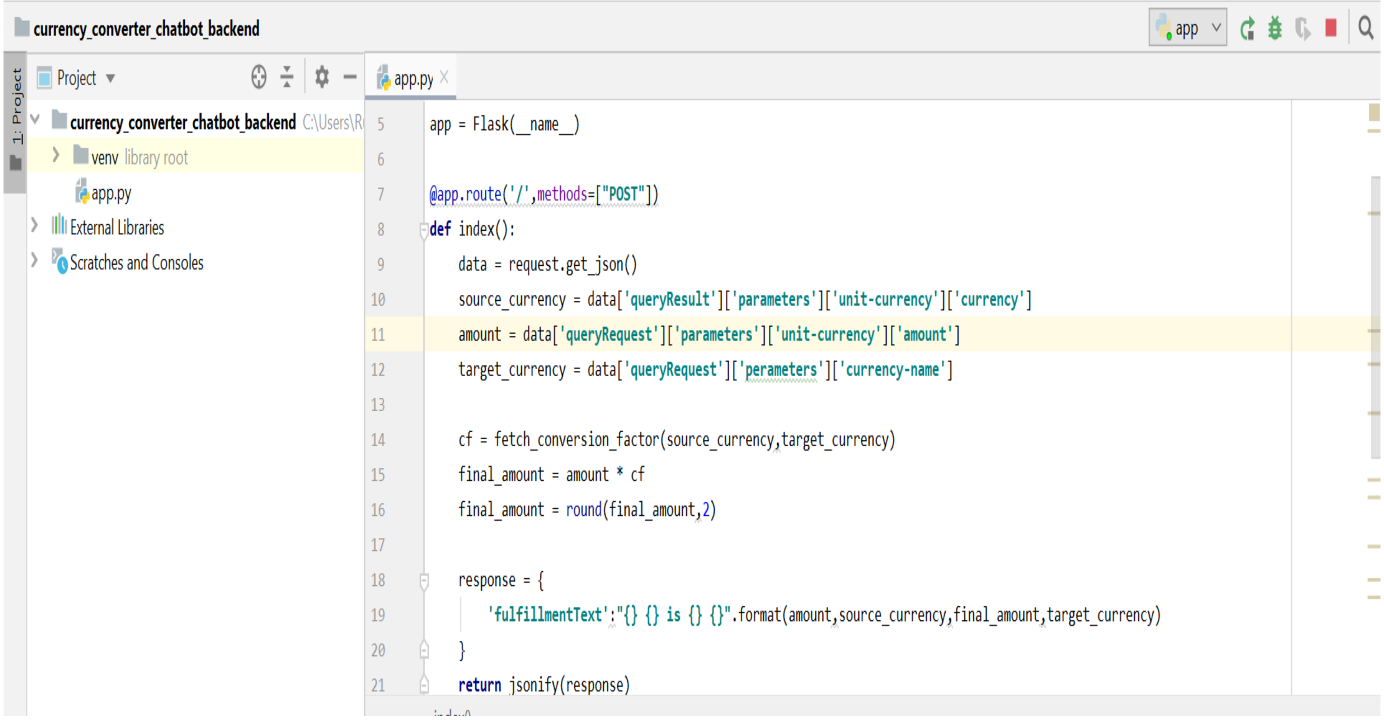
HEADERS Enter key Enter value

Add header

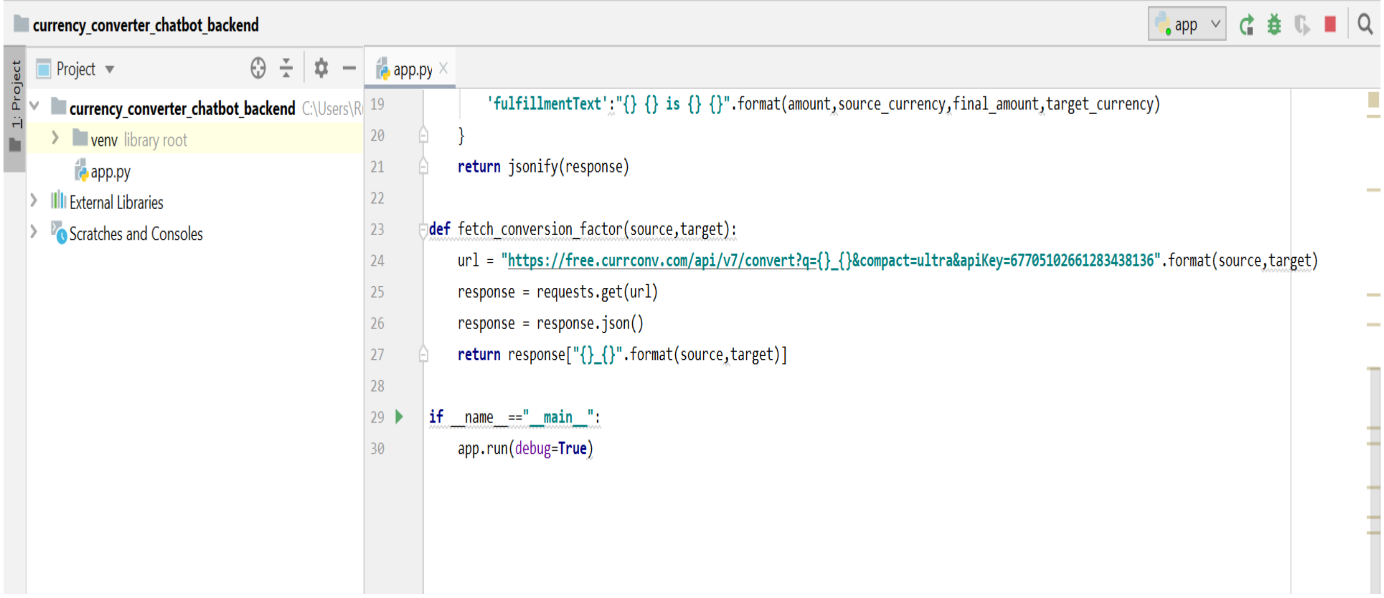
SMALL TALK Disable webhook for Smalltalk

Fig – 12

- When service will receive a POST request from dialogflow in the form of response to a user query matched by intents with webhook enabled.
- We have to be sure that our web services meet all the webhook requirements specific to the API version enabled in this agent.
- In simple word the JSON object coding will reflect or connect on our flask application.
- After doing that just save that.
- Then go to intent section.
- Go down, on the fulfilment enable the webhook call for this intent in the currency converter intent.
- By doing that our chatbot and flask app are connected to each other.



```
app = Flask(__name__)


@app.route('/',methods=["POST"])
def index():
    data = request.get_json()
    source_currency = data['queryResult']['parameters']['unit-currency']['currency']
    amount = data['queryRequest']['parameters']['unit-currency']['amount']
    target_currency = data['queryRequest']['perameters']['currency-name']

    cf = fetch_conversion_factor(source_currency,target_currency)
    final_amount = amount * cf
    final_amount = round(final_amount,2)

    response = {
        'fulfillmentText':"{} {} is {} {}".format(amount,source_currency,final_amount,target_currency)
    }
    return jsonify(response)
```

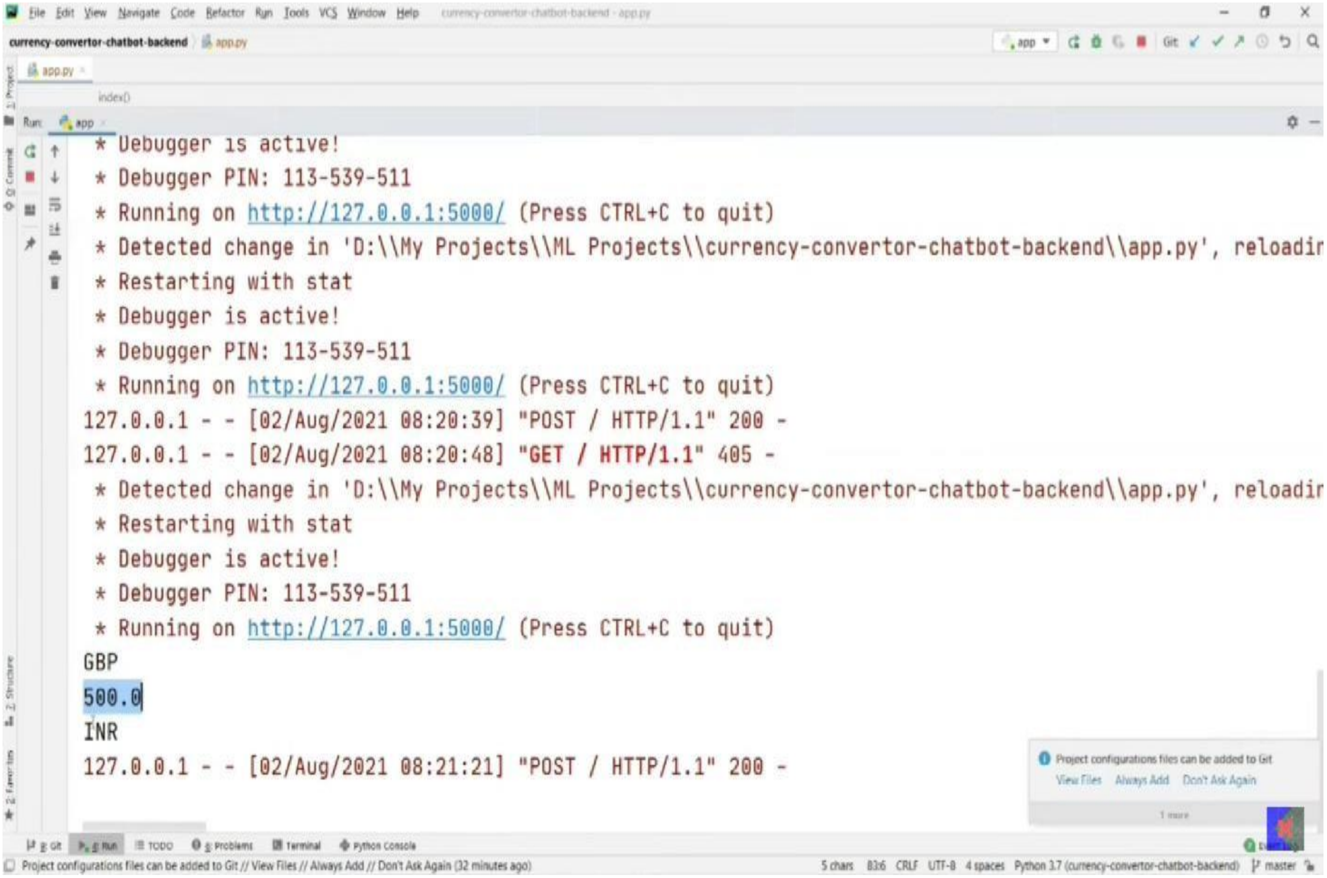


```
        'fulfillmentText':"{} {} is {} {}".format(amount,source_currency,final_amount,target_currency)
    }
    return jsonify(response)

def fetch_conversion_factor(source,target):
    url = "https://free.currconv.com/api/v7/convert?q={}_{}&compact=ultra&apiKey=67705102661283438136".format(source,target)
    response = requests.get(url)
    response = response.json()
    return response["{}_{}".format(source,target)]

if __name__=="__main__":
    app.run(debug=True)
```

Fig - 13

#Now, we have to code in the flask app, which we wanted to do to the parameter provided by the chatbot to receive the data as sawn in figure 13.

API OUTPUT IMAGE



File Edit View Navigate Code Refactor Run Tools VCS Window Help
currency-convertor-chatbot-backend - app.py
currency-convertor-chatbot-backend
app.py
app
Git
index()
Run: app
Git
Run
TODO
Problems
Terminal
Python Console

```
 * Debugger is active!
 * Debugger PIN: 113-539-511
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Detected change in 'D:\\My Projects\\ML Projects\\currency-convertor-chatbot-backend\\app.py', reloadir
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 113-539-511
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [02/Aug/2021 08:20:39] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [02/Aug/2021 08:20:48] "GET / HTTP/1.1" 405 -
 * Detected change in 'D:\\My Projects\\ML Projects\\currency-convertor-chatbot-backend\\app.py', reloadir
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 113-539-511
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
GBP
500.0
INR
127.0.0.1 - - [02/Aug/2021 08:21:21] "POST / HTTP/1.1" 200 -
```

Project configurations files can be added to Git
View Files Always Add Don't Ask Again

Project configurations files can be added to Git // View Files // Always Add // Don't Ask Again (32 minutes ago)

5 chars 83:6 CRLF UTF-8 4 spaces Python 3.7 (currency-convertor-chatbot-backend) master

Fig – 14

- Figure 14 shows the output of the Flask API.
- API fetches the intent of user and extract the entity.
- This entity got print on the terminal in this figure.
- Like if user says *"hey please convert 500 GBP to INR."*
- Here unit currency is GBP, amount is 500 and currency name is INR which has printed on the terminal.
- We can also see a portal named *http://127.0.0.1:5000*
- On this portal our output got printed, if we click on that portal, it redirects us to that window.

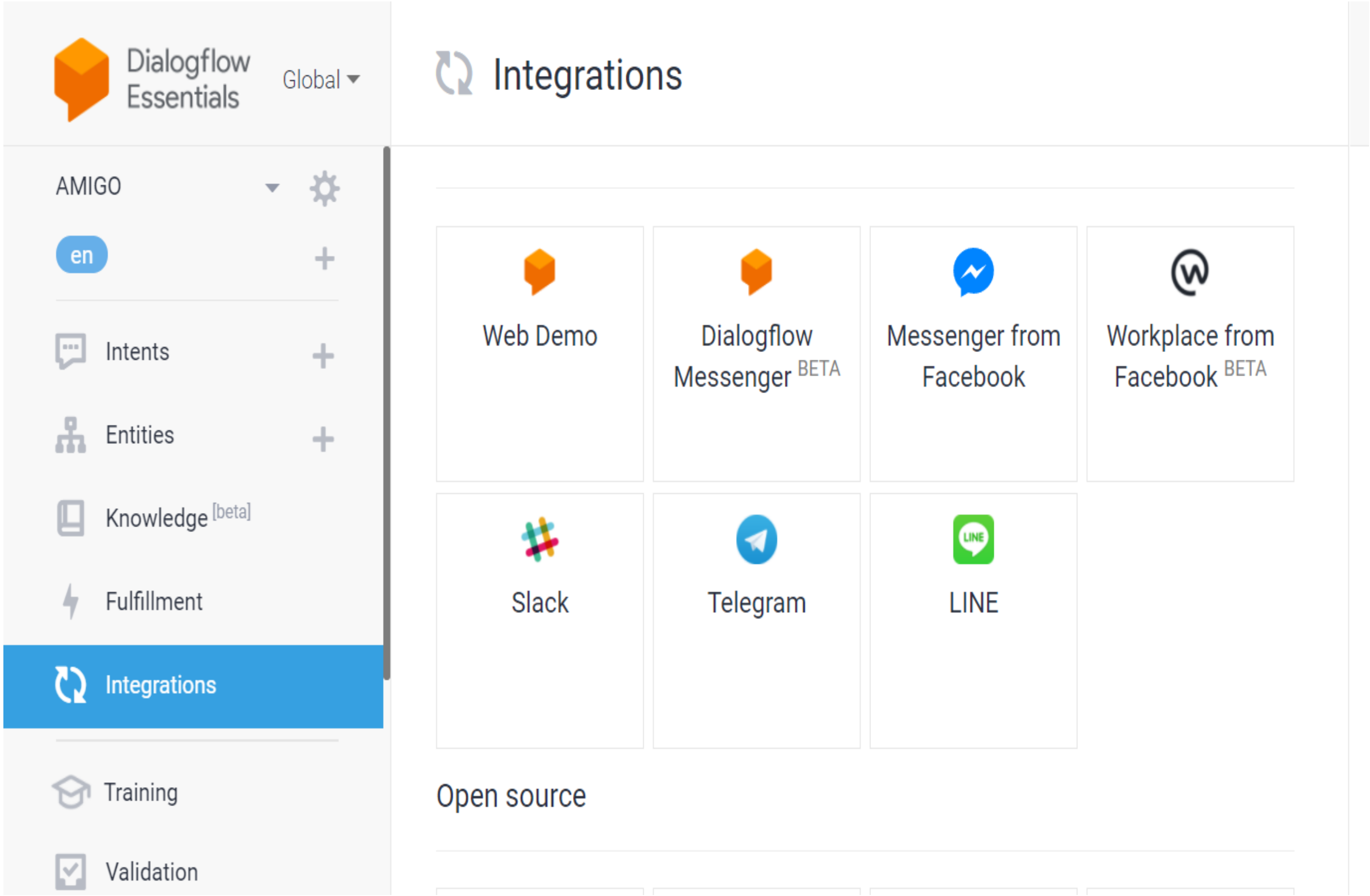
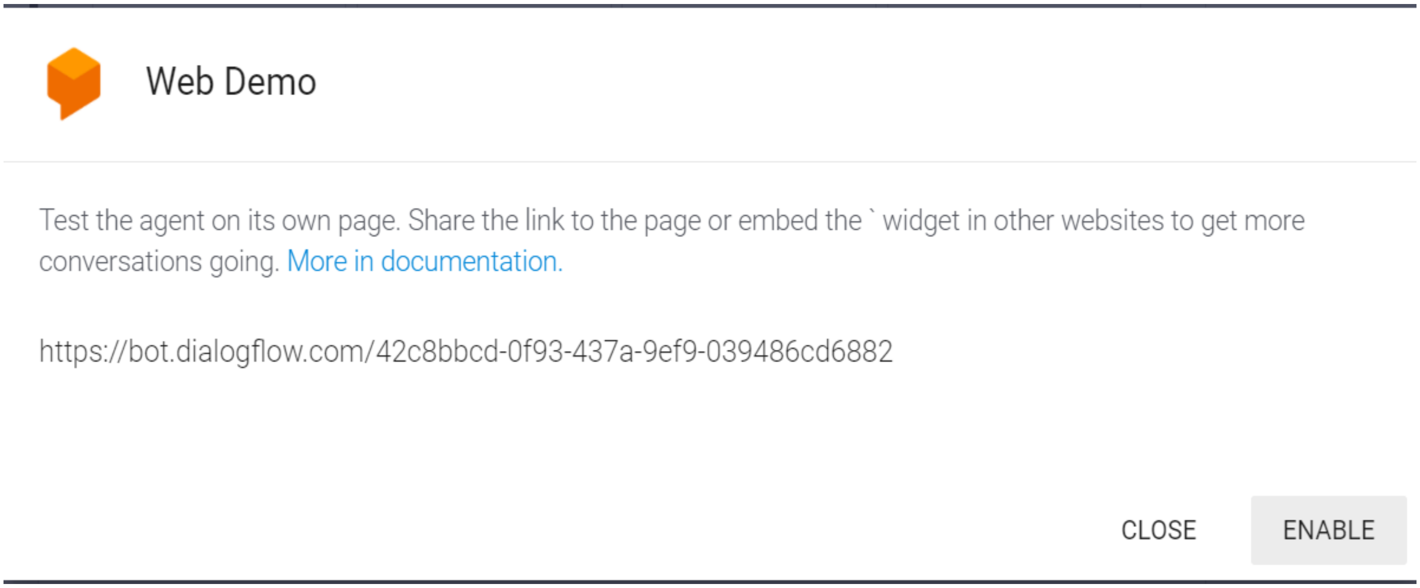


Fig – 15

- There is an integrations section below the fulfilment sawn in figure 15.
- This integrations section uses to upload our chatbot on some social media platform.
- As we can see there are some options where we can upload our chatbot.
- These options are Facebook messenger, Telegram, Line, Slack etc.
- I am going to upload this chatbot on Telegram.
- Before uploading let see web demo.
- Web demo is section in Integrations part sawn in figure 15.



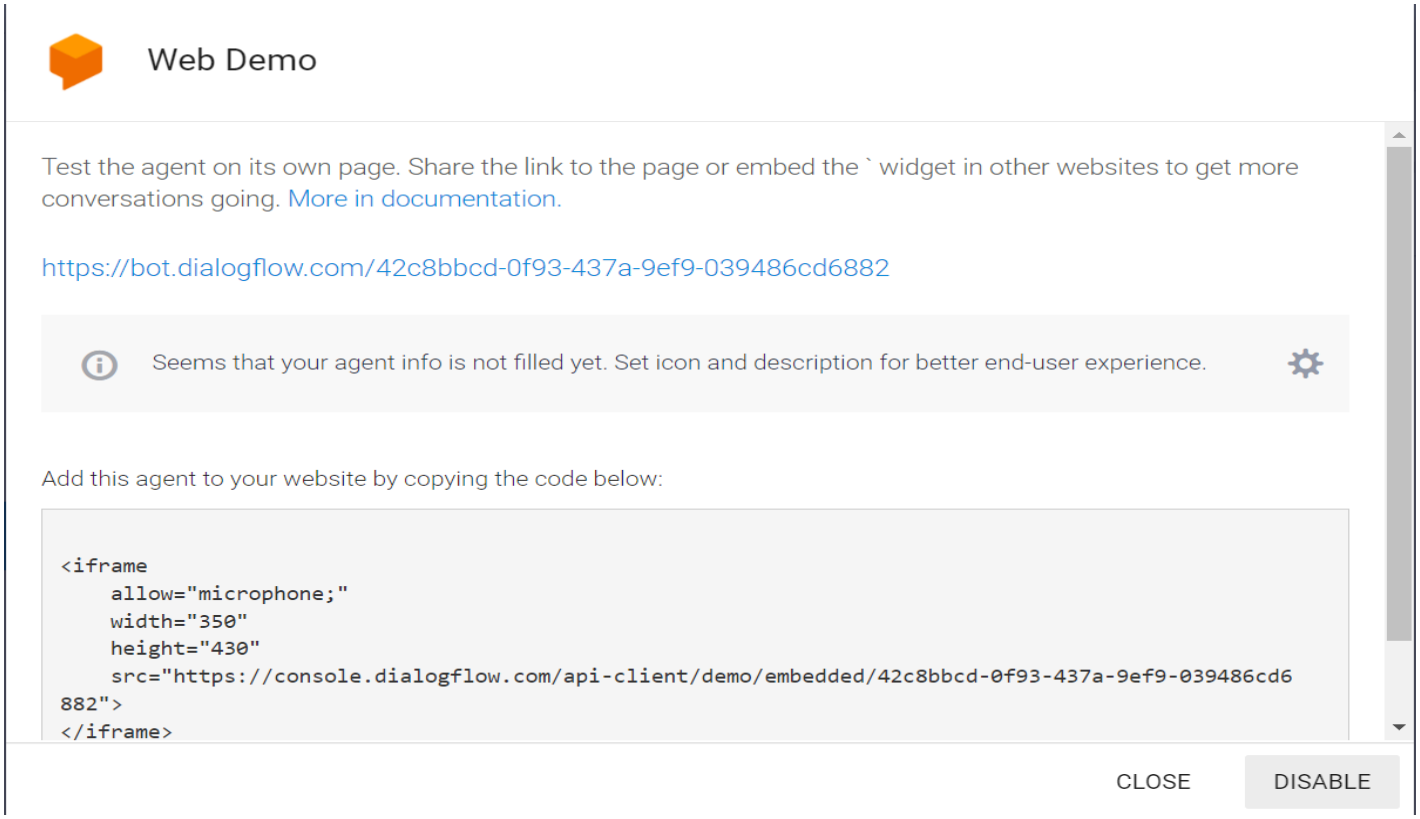
Web Demo

Test the agent on its own page. Share the link to the page or embed the ` widget in other websites to get more conversations going. More in documentation.

https://bot.dialogflow.com/42c8bbcd-0f93-437a-9ef9-039486cd6882

Fig – 16

❖ For starting the we have to enable the web demo sawn in figure 16.



Web Demo

Test the agent on its own page. Share the link to the page or embed the ` widget in other websites to get more conversations going. More in documentation.

https://bot.dialogflow.com/42c8bbcd-0f93-437a-9ef9-039486cd6882

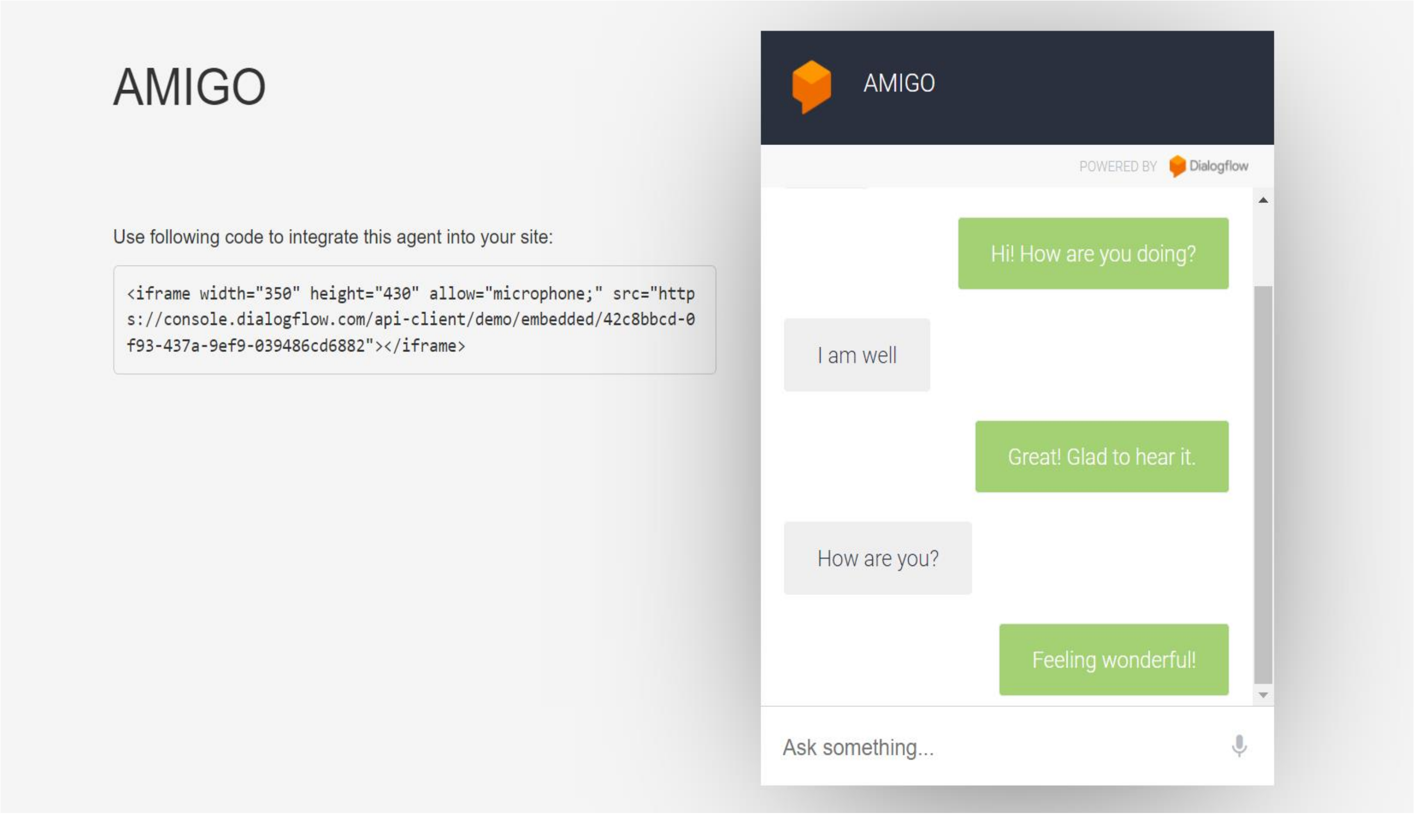
Seems that your agent info is not filled yet. Set icon and description for better end-user experience.

Add this agent to your website by copying the code below:

```
<iframe
    allow="microphone;"
    width="350"
    height="430"
    src="https://console.dialogflow.com/api-client/demo/embedded/42c8bbcd-0f93-437a-9ef9-039486cd6
882">
</iframe>
```

Fig - 17

- After enabling, a window appears having a URL in blue colour in figure 17.
- We have hit on that URL.



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/42c8bbcd-0f93-437a-9ef9-039486cd6882"></iframe>
```

Hi! How are you doing?

I am well

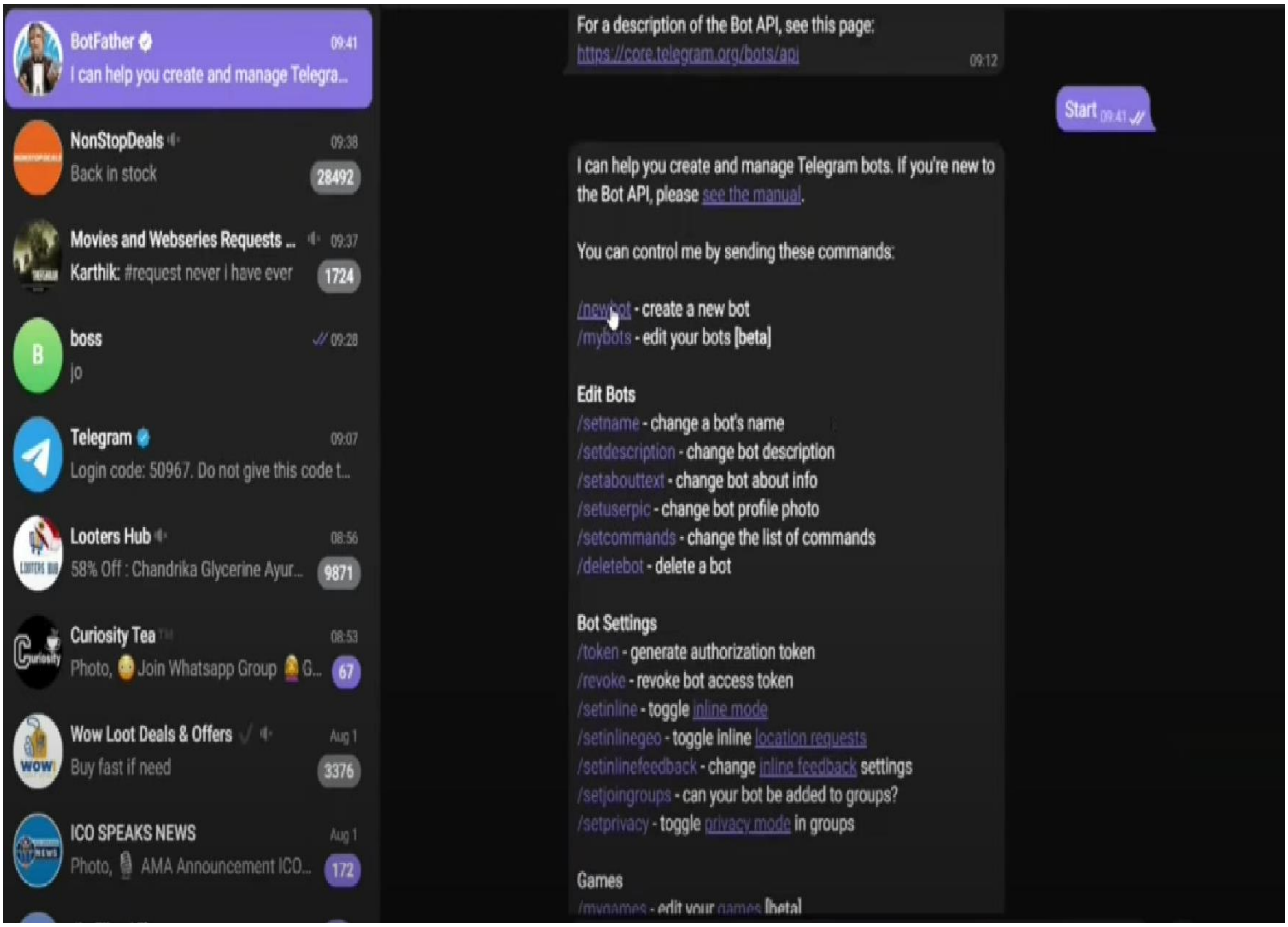
Great! Glad to hear it.

How are you?

Feeling wonderful!

Fig – 18

- After hitting on the URL, a window appears just like sawn in the figure 18.
- Here we can test our chatbot, by giving instruction to it.
- There is section written *“use the following code to integrate this agent into your site.”*
- We can use this code to embed this chatbot in our website if we have one.
- Now come to the uploading part of chatbot on telegram.
- For that we have to open telegram on chrome.
- Then login using our account.
- After logged in search for BotFather.
- Then open the chat window of BotFather.
- Type start in chat window.


BotFather 09:41
I can help you create and manage Telegra...
NonStopDeals 09:38
Back in stock 28492
Movies and Webseries Requests ... 09:37
Karthik: #request never i have ever 1724
boss 09:28
jo
Telegram 09:07
Login code: 50967. Do not give this code t...
Looters Hub 08:56
58% Off : Chandrika Glycerine Ayur... 9871
Curiosity Tea 08:53
Photo, Join Whatsapp Group G... 67
Wow Loot Deals & Offers Aug 1
Buy fast if need 3376
ICO SPEAKS NEWS Aug 1
Photo, AMA Announcement ICO... 172
Start 09:41

For a description of the Bot API, see this page:
https://core.telegram.org/bots/api 09:12

I can help you create and manage Telegram bots. If you're new to the Bot API, please see the manual.

You can control me by sending these commands:

/newbot - create a new bot
/mybots - edit your bots [beta]

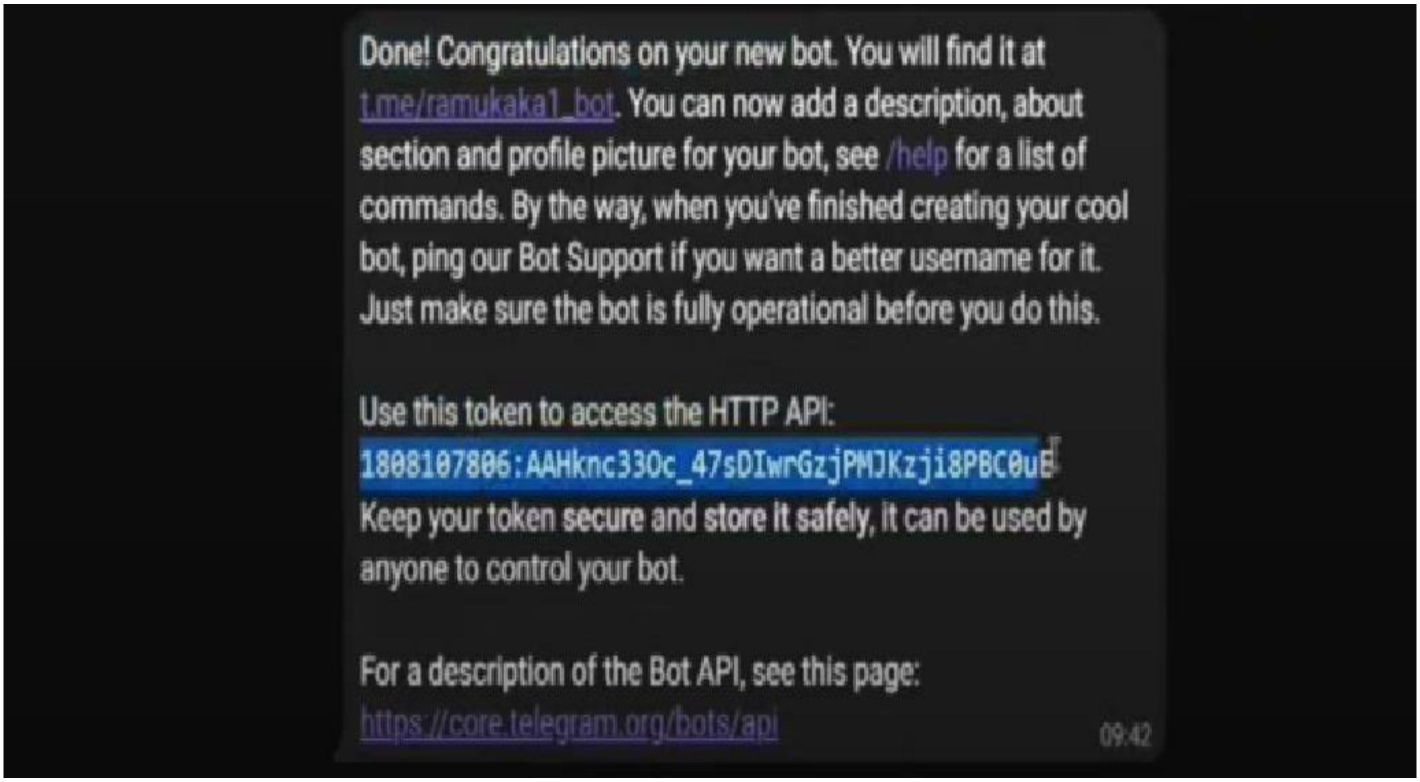
Edit Bots
/setname - change a bot's name
/setdescription - change bot description
/setabouttext - change bot about info
/setuserpic - change bot profile photo
/setcommands - change the list of commands
/deletebot - delete a bot

Bot Settings
/token - generate authorization token
/revoke - revoke bot access token
/setinline - toggle inline mode
/setinlinegeo - toggle inline location requests
/setinlinefeedback - change inline feedback settings
/setjoingroups - can your bot be added to groups?
/setprivacy - toggle privacy mode in groups

Games
/mygames - edit your games [beta]

Fig – 19

- After typing start in chat box, a window appears just like in figure 19.
- It gives us options what we wanted to do.
- Like Edit Bots, Bot Setting, New Bot.
- We click on newbot, because we are making new chatbot.
- Then it asks to provide the name of chatbot.
- We will type the name in chat box.
- Make sure your chatbot name contain bot in the end.
- Now it asks for the username of that chatbot like: *"trios_bot"*.
- If the username is available then it assigns to our chatbot otherwise it asks for another username.



Done! Congratulations on your new bot. You will find it at t.me/ramukaka1_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
1808107806:AAHknc33Oc_47sDIwrGzjPMJKzji8PBC0uE
Keep your token secure and store it safely, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:
https://core.telegram.org/bots/api

09:42

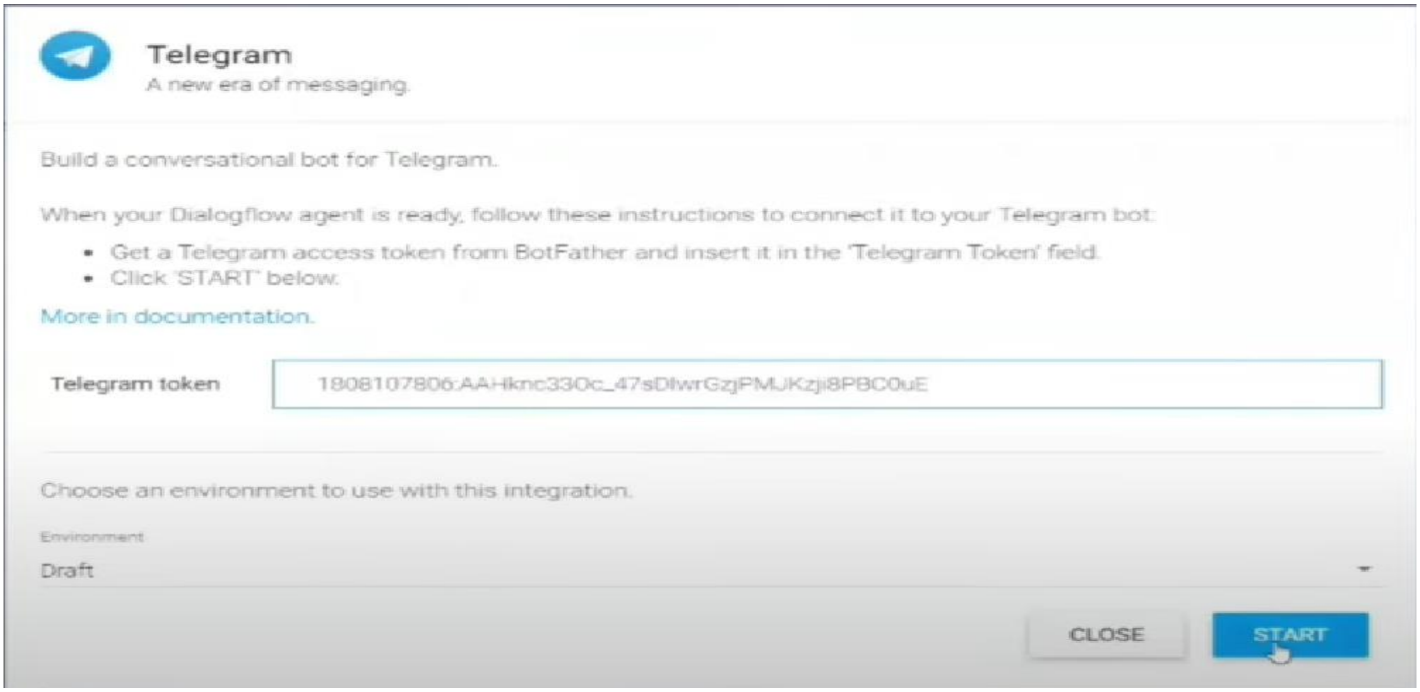
Fig – 20

- After successful naming and selecting username for the chatbot it gives us access token as sawn in figure 20.

- Copy this access code.

- Go to Dialogflow.

- Open the telegram in integrations section**.**



Telegram

A new era of messaging

Build a conversational bot for Telegram.

When your Dialogflow agent is ready, follow these instructions to connect it to your Telegram bot:

- Get a Telegram access token from BotFather and insert it in the 'Telegram Token' field.
- Click 'START' below.

More in documentation.

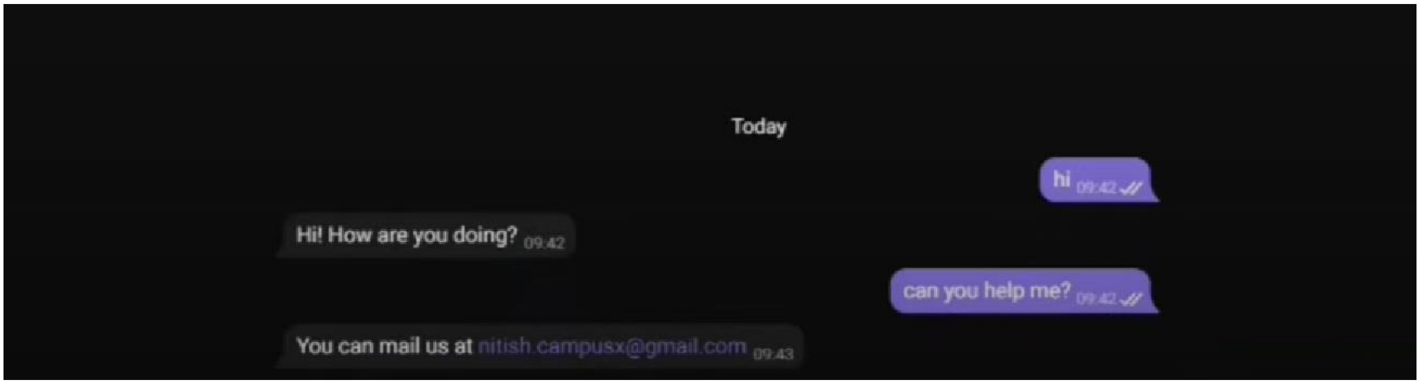
Telegram token: 1808107806:AAHknc330c_47sDlwrGzjPMJKzji8PBC0uE

Choose an environment to use with this integration.

Environment: Draft

Fig – 21

- ❖ Paste the code in Telegram token part.
- ❖ Tap on start.
- ❖ After that the uploading of chatbot is done.



Today

hi 09:42

Hi! How are you doing? 09:42

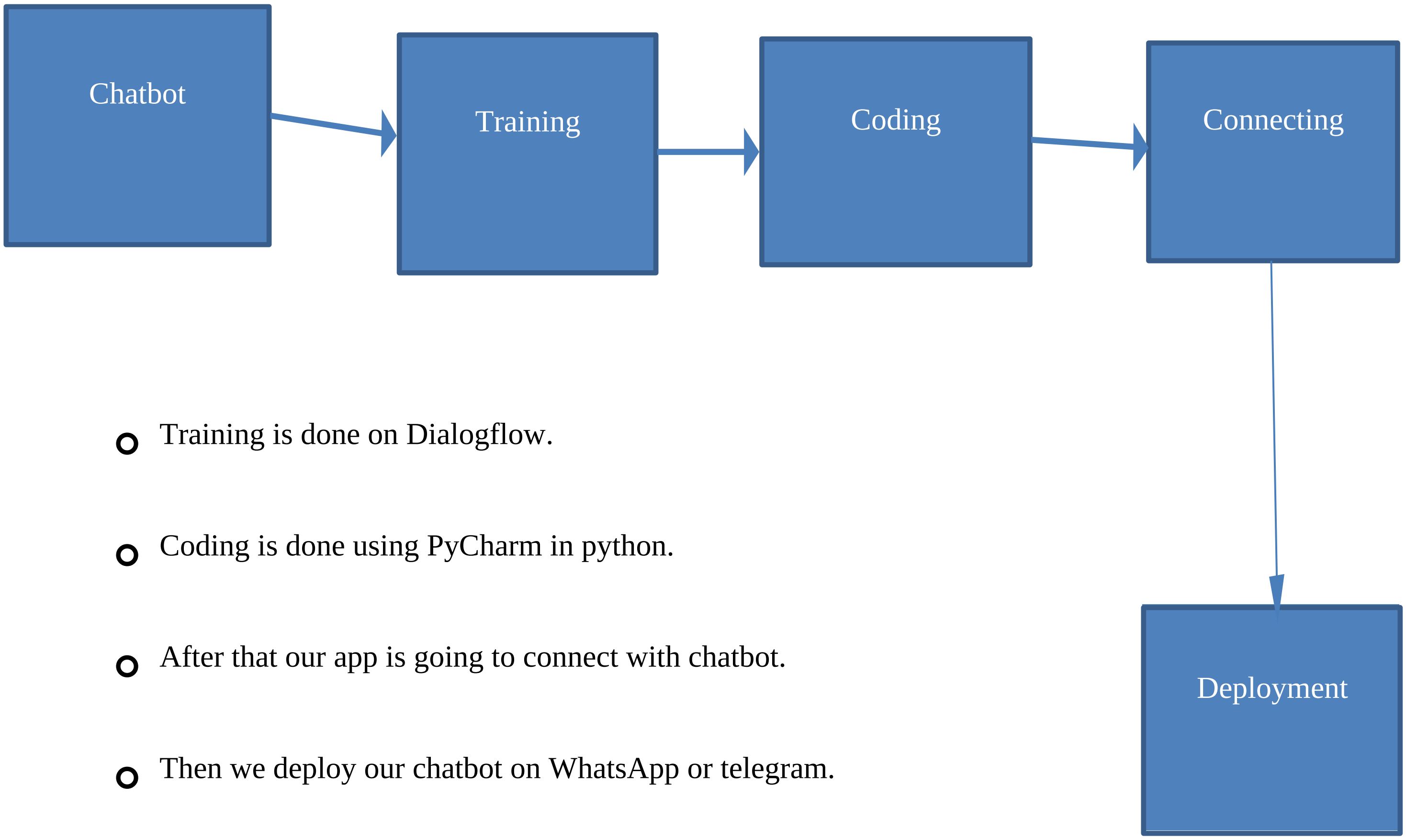
can you help me? 09:42

You can mail us at nitish.campusx@gmail.com 09:43

Fig - 22

- ❖ Now search the name as you provide to your chatbot.
- ❖ Your chatbot is ready to work.

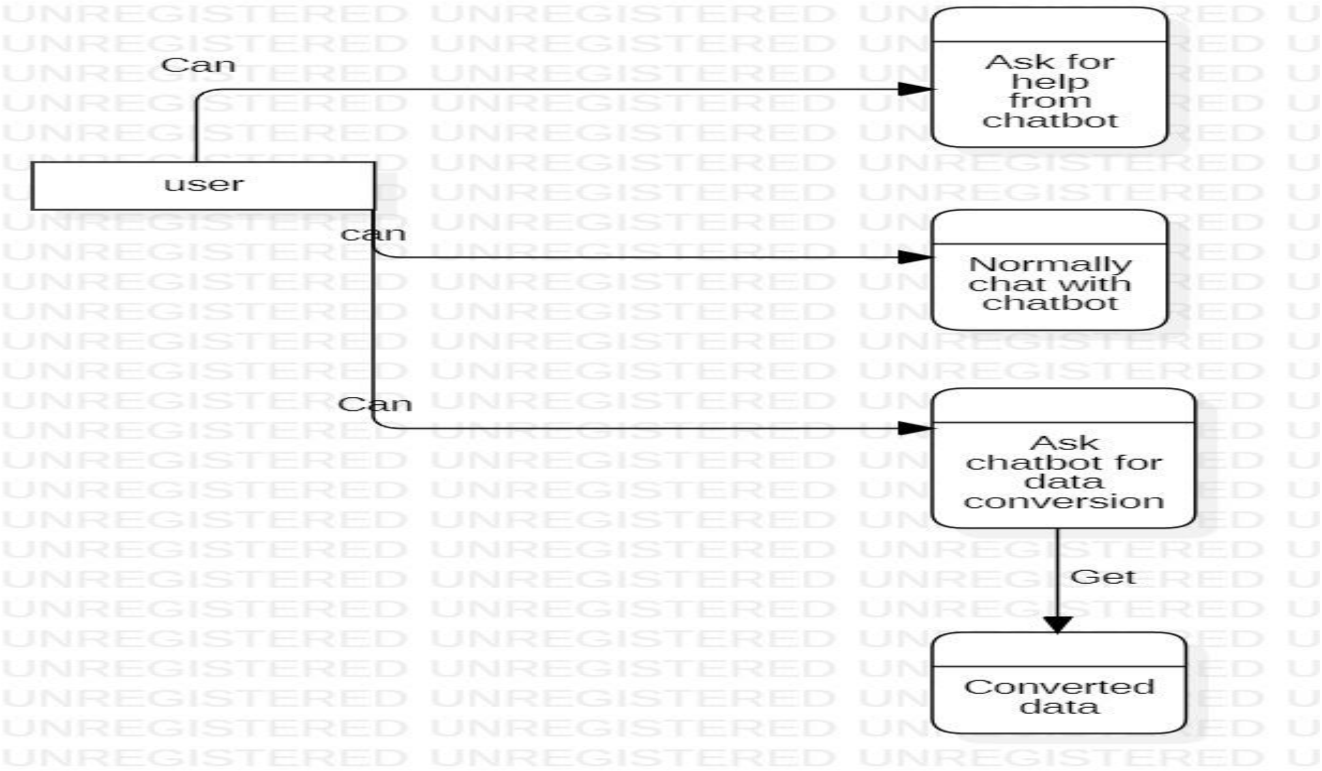
DESIGN



- Training is done on Dialogflow.
- Coding is done using PyCharm in python.
- After that our app is going to connect with chatbot.
- Then we deploy our chatbot on WhatsApp or telegram.

Data Flow Diagram

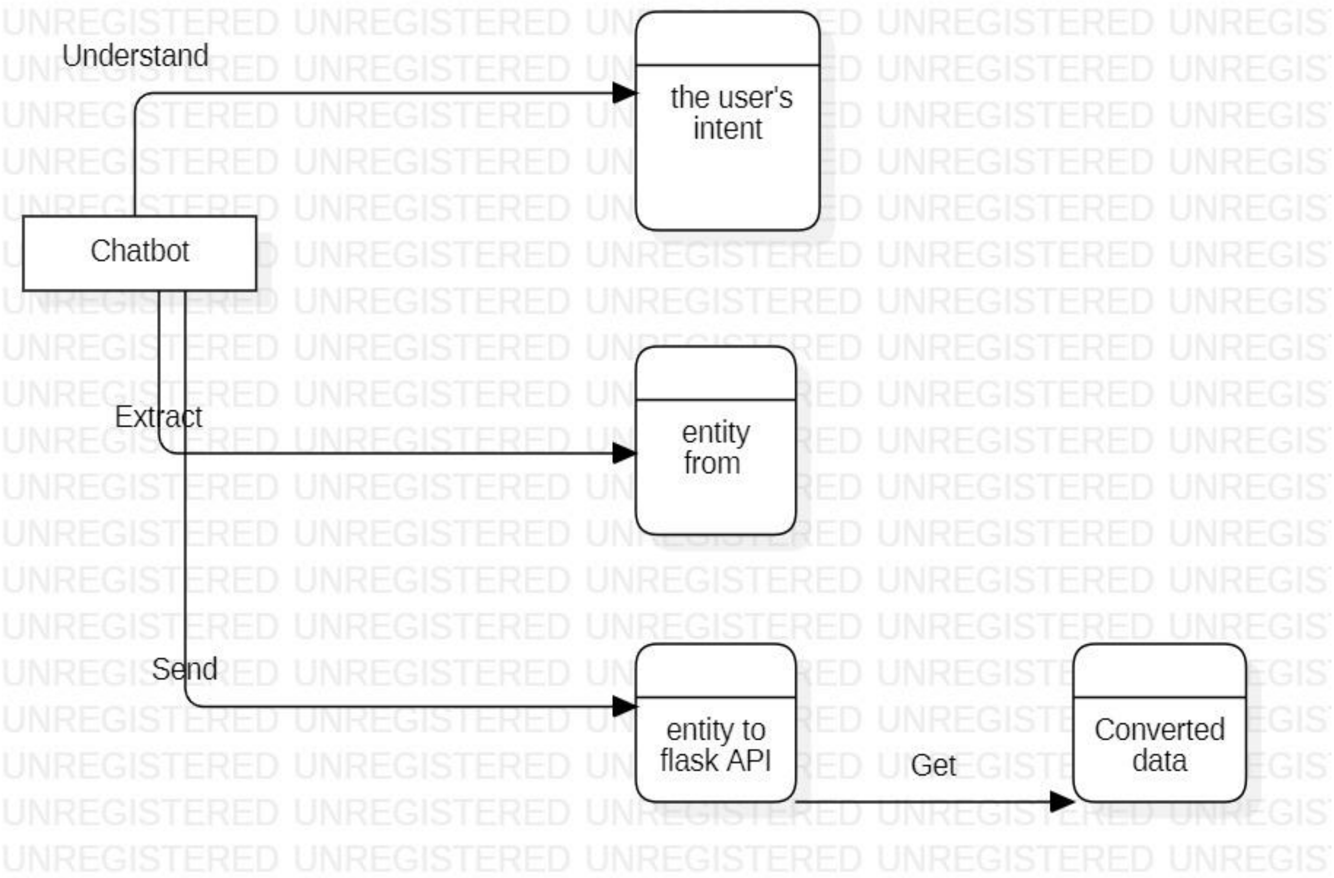
A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.



user
Can
Ask for help from chatbot
can
Normally chat with chatbot
Can
Ask chatbot for data conversion
Get
Converted data

Rules for creating DFD

- The name of the entity should be easy and understandable without any extra assistance (like comments).
- The processes should be numbered or put in ordered list to be referred easily.
- The DFD should maintain consistency across all the DFD levels.
- A single DFD can have maximum processes up to 9 and minimum 3 processes.



Advantages of DFD

- ❖ It helps us to understand the functioning and the limits of a system.
- ❖ It is a graphical representation which is very easy to understand as it helps visualize contents.
- ❖ Data Flow Diagram represent detailed and well explained diagram of system components.
- ❖ It is used as the part of system documentation file.
- ❖ Data Flow Diagrams can be understood by both technical or nontechnical person because they are very easy to understand.

Disadvantages of DFD

- ❖ At times DFD can confuse the programmers regarding the system.
- ❖ Data Flow Diagram takes long time to be generated, and many times due to this reasons analysts are denied permission to work on it.

Activity Diagram:

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as −

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

How to Draw an Activity Diagram?

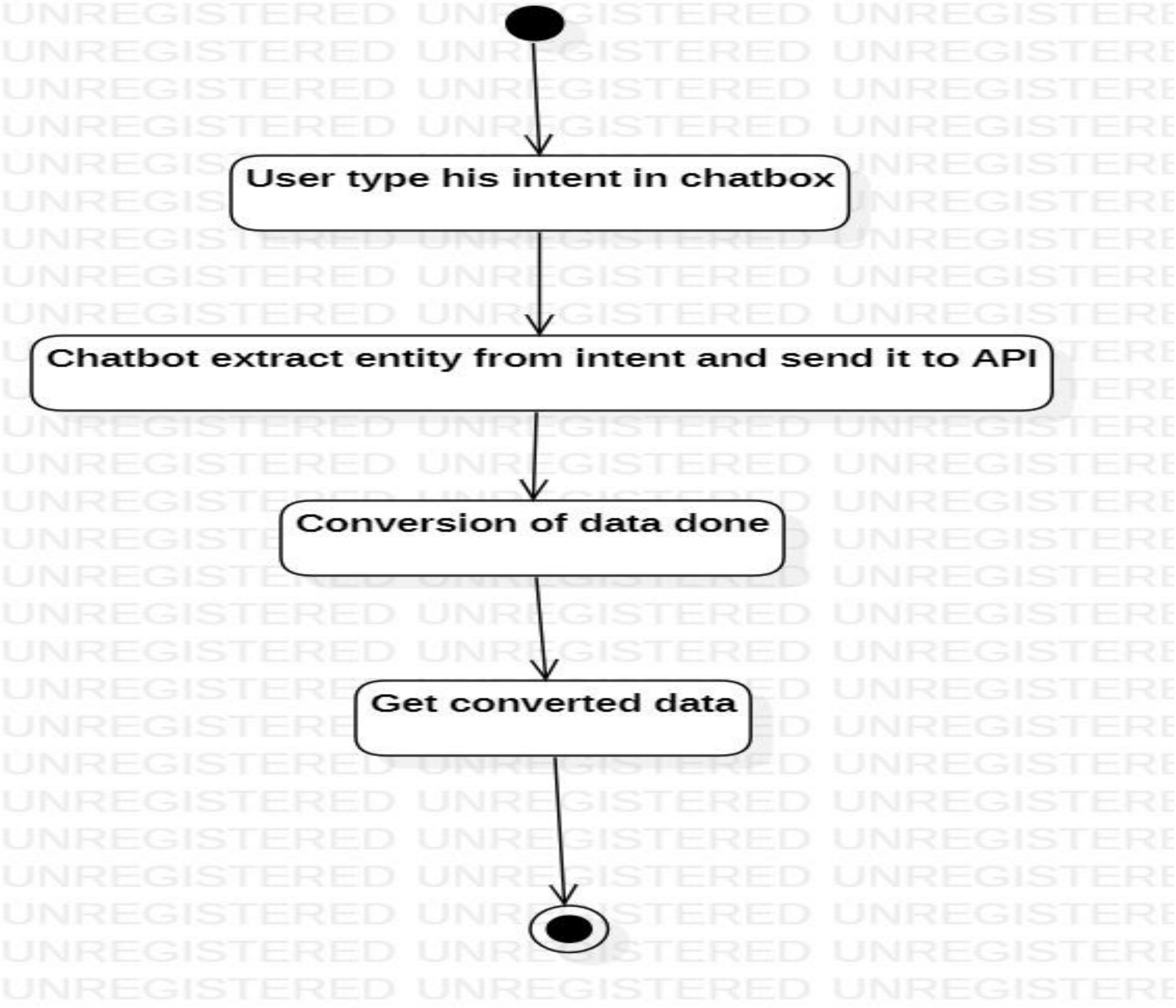
Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc.

Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

Before drawing an activity diagram, we should identify the following elements −

- Activities
- Association
- Conditions
- Constraints

Once the above-mentioned parameters are identified, we need to make a mental layout of the entire flow. This mental layout is then transformed into an activity diagram.


User type his intent in chatbox
Chatbot extract entity from intent and send it to API
Conversion of data done
Get converted data

Where to Use Activity Diagrams?

The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.

Activity diagram is suitable for modelling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

Activity diagram can be used for −

- Modelling work flow by using activities.
- Modelling business requirements.
- High level understanding of the system's functionalities.
- Investigating business requirements at a later stage.

Class Diagram:

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Purpose of Class Diagrams

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as −

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

How to Draw a Class Diagram?

Class diagrams are the most popular UML diagrams used for construction of software applications. It is very important to learn the drawing procedure of class diagram.

Class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top-level view.

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represents the whole system.

The following points should be remembered while drawing a class diagram −

- The name of the class diagram should be meaningful to describe the aspect of the system.
- Each element and their relationships should be identified in advance.
- Responsibility (attributes and methods) of each class should be clearly identified
- For each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated.
- Use notes whenever required to describe some aspect of the diagram. At the end of the drawing, it should be understandable to the developer/coder.
- Finally, before making the final version, the diagram should be drawn on plain paper and reworked as many times as possible to make it correct.

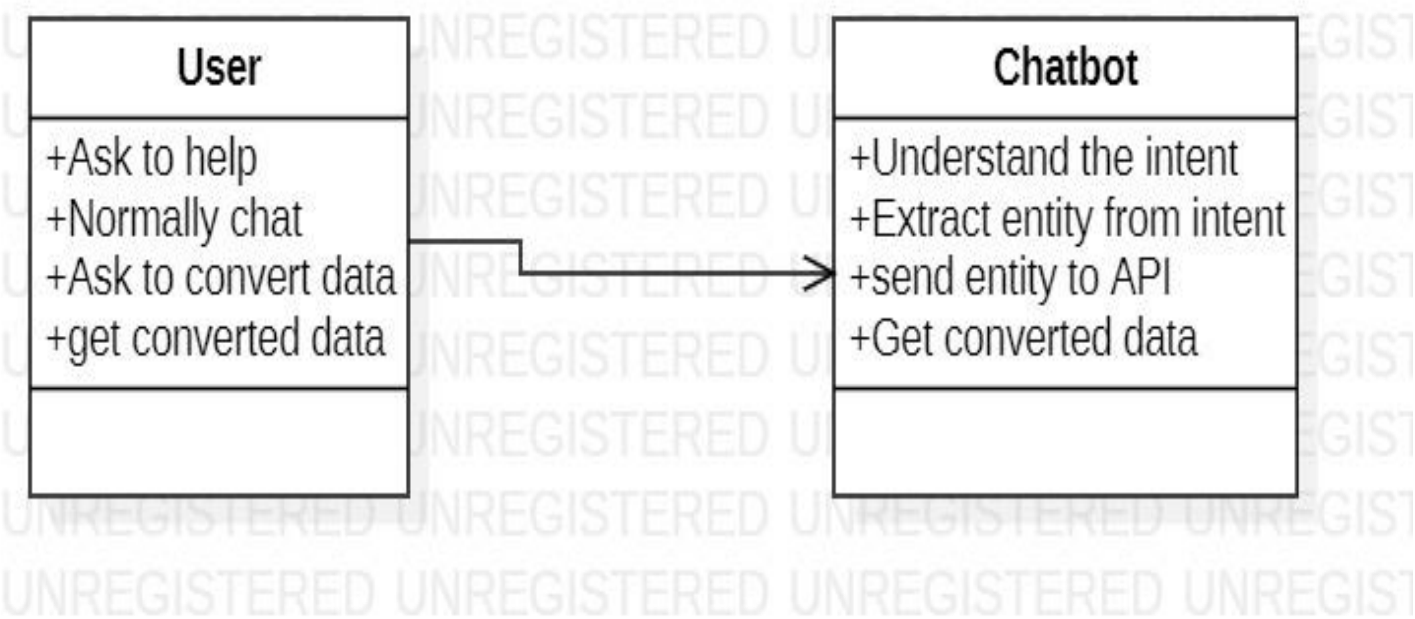
Where to Use Class Diagrams?

Class diagram is a static diagram and it is used to model the static view of a system. The static view describes the vocabulary of the system.

Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.

Generally, UML diagrams are not directly mapped with any object-oriented programming languages but the class diagram is an exception.

Class diagram clearly shows the mapping with object-oriented languages such as Java, C++, etc. From practical experience, class diagram is generally used for construction purpose.



In a nutshell it can be said, class diagrams are used for −

- Describing the static view of the system.
- Showing the collaboration among the elements of the static view.
- Describing the functionalities performed by the system.
- Construction of software applications using object-oriented languages.

Use Case Diagram:

To model a system, the most important aspect is to capture the dynamic behaviour. Dynamic behaviour means the behaviour of the system when it is running/operating.

Only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

Hence to model the entire system, a number of use case diagrams are used.

Purpose of Use Case Diagrams

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows −

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

How to Draw a Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analysed, the functionalities are captured in use cases.

We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

Functionalities to be represented as use case

- Actors
- Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system.

Where to Use a Use Case Diagram?

There are five diagrams in UML to model the dynamic view of a system. Now each and every model has some specific purpose to use. Actually, these specific purposes are different angles of a running system.

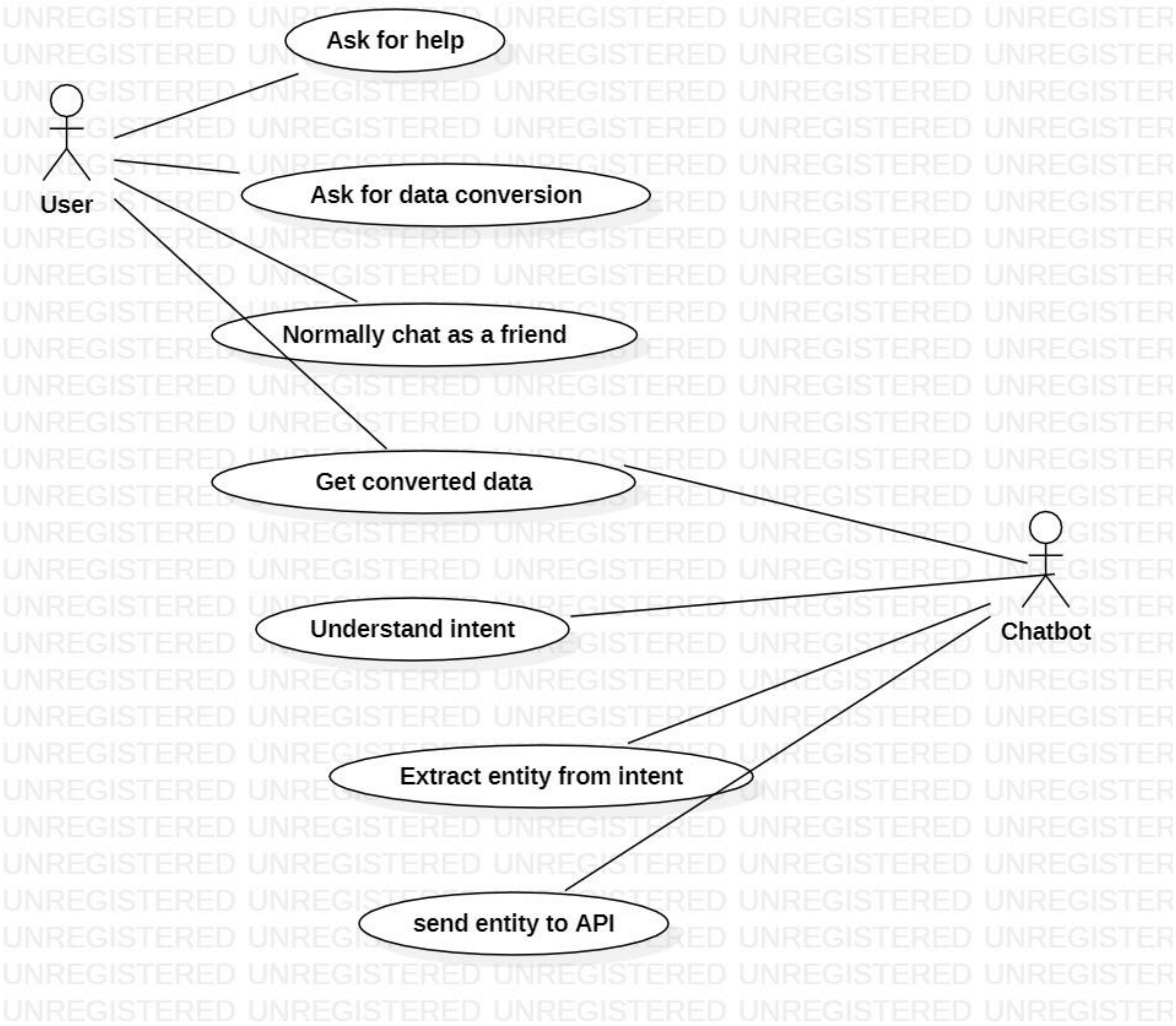
To understand the dynamics of a system, we need to use different types of diagrams. Use case diagram is one of them and its specific purpose is to gather system requirements and actors.

Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented. Use case diagram can be imagined as a black box where only the input, output, and the function of the black box is known.

These diagrams are used at a very high level of design. This high-level design is refined again and again to get a complete and practical picture of the system. A well-structured use case also describes the pre-condition, post condition, and exceptions. These extra elements are used to make test cases when performing the testing.

Although use case is not a good candidate for forward and reverse engineering, still they are used in a slightly different way to make forward and reverse engineering. The same is true for reverse engineering. Use case diagram is used differently to make it suitable for reverse engineering.

In forward engineering, use case diagrams are used to make test cases and in reverse engineering use cases are used to prepare the requirement details from the existing application.



Use case diagrams can be used for –

- Requirement analysis and high-level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.

Result, Conclusion & Future Scope

The purpose of making this project is to provide everyone an easy way to the problem of data conversion. So, the result of this project is a chatbot which can convert any type of data into our desirable data.

And finally, it came up with a friendly chatbot. The chatbot is friendly because we can chat with it as friend also that is why the name of bot is AMIGO which is a Spanish word means friend.

In future, it can convert the distances, lengths from one form to another like meters to centimetres and many more option will be there.

List of Figures

REFERENCES

1. “Recruitment Chatbots”, International Research Journal of Engineering and Technology (IRJET), vol. 5, Issue: 08, Aug 2018[1].
   Authors: Akash Balachandar, Anusha D Kulkarni

2. Implementation of Chatbot in Online Commerce, and Open Innovation.
   Authors: María D. Illescas-Manzano, Noé Vicente López, Nuno Afonso González and Carmen Cristofol Rodríguez.

3. Literature survey on various chatbot.
   Authors: Harsha Pariyani, Anshika Sinha, Preeti Bhat, Roshni Rote, Asst. Prof. N. A. Mulla.

4. “Intelligent chatbot for easy web-analytics”.
   Author: Ravi R.

5. “Task-based Interaction Chatbot”, EEE521 final year project Report school of computing, Engineering & Intelligent System[4]. Authors: Dr. Kevin Curran, Dr. Daniel Kelly

6. “Classification Technique of Interviewer-Bot Result using Naïve Bayes an Phrase Reinforcement Algorithms,” International Journal of Emerging Technologies in Learning (iJET), 13(02), 33-47, 2018[2]. Authors: Sarosa, M., Junus, M., Hoesny, M. U., Sari, Z., & Fatnuriyah, M.

7. Deshpande, A., Shahane, A., Gadre, D., Deshpande, M., & Joshi, P. M, 2017. A Survey of various chatbot implementation techniques. International Journal of Computer Engineering and Applications, Volume XI, Special Issue, ISSN 2321-3469.

8. .https://chatbotsmagazine.com/what-is-the-working-of-a-chatbot-e99e6996f51c

9. https://en.wikipedia.org/wiki/Chatbot