

A
Mini Project Report

On

“CHATBOT FOR ARUNDATHI HOSPITAL”

Submitted in partial fulfillment of the
Requirements for the award of the degree of

Bachelor of Technology

In

Computer Science & Engineering – Data Science

By

S.Ritesh Dhyan - 20R21A6752

P.Sai Kumar - 20R21A6751

K.Sri Prabha - 20R21A6737

R.Vivek - 20R21A6750

Under the guidance of

Mr.P.Srinivasa Reddy

Assistant Professor

Department of Computer Science & Engineering(DS,AI ML,CS)



MLR

INSTITUTE OF TECHNOLOGY

(UGC AUTONOMOUS)

Affiliated to JNTUH, Approved by AICTE
Laxman Reddy Avenue, Dundigal, Hyderabad-500 043, Telangana, India



INDEX

College Certificate	i
Acknowledgement	ii
Declaration	iii
List of Figures	iv
Abstract	1
Chapter 1	
Introduction	2-3
1.1 Overview	2
1.2 Types of Chatbots	2
1.3 Applications	3
Chapter 2	
Literature Survey	4
2.1 Existing System	4
2.1.1 Limitations of Existing System	5
Chapter 3	
Proposed System	5-9
3.1 Proposed System	5
3.1.1 Advantages of Proposed System	5
3.2 System Requirements	6
3.2.1 Software Requirements	6
3.2.2 Hardware Requirements	6
3.3 Concepts used in proposed system	7-9
Chapter 4	
Architecture Design	10
4.1 Working of the chatbot	10-11

Chapter 5	
Modules	11-13
5.1 Tensorflow	11
5.2 Nltk	11
5.3 Tkinter	12
5.4 Json	12
5.5 Pickle	12
5.6 Random	12
5.7 Numpy	13
5.8 Pandas	13
5.9 Warnings	13
Chapter 6	
Implementation	14-18
Chapter 7	
Code	19-22
7.1 chatbot_py.py	19-20
7.2 Training_py.py	21-22
Chapter 8	
Output/Results	23-28
Chapter 9	
Conclusion	29
Chapter 10	
References	30

Department of Computer Science & Engineering(DS,AI ML,CS)

CERTIFICATE

This is to certify that the project entitled “**Chatbot for Arundathi Hospital**” has been submitted by **S. Ritesh Dhyan(20R21A6752), P. Sai Kumar(20R21A6751),K. Sri Prabha (20R21A6737),R. Vivek(20R21A6750)** in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

Internal Guide

Head of the Department

External Examiner

Department of Computer Science & Engineering(DS,AI ML,CS)

ACKNOWLEDGEMENT

There are many people who helped me directly and indirectly to complete my project successfully. I would like to take this opportunity to thank one and all. First of all I would like to express my deep gratitude towards my internal guide **Mr.P.Srinivasa Reddy Asst.Prof,Department of CSE(DS,AI ML,CS)** for his support in the completion of my dissertation. I wish to express my sincere thanks to **K.Sai Prasad HOD, Department of CSE(DS,AI ML,CS)** and also to principal **Dr. K.Srinivas Rao** for providing the facilities to complete the dissertation. I would like to thank all our faculty, coordinators and friends for their help and constructive criticism during the project period. Finally, I am very much indebted to our parents for their moral support and encouragement to achieve goals.

S. Ritesh Dhyan - 20R21A6752

P. Sai Kumar - 20R21A6751

K. Sri Prabha - 20R21A6737

R. Vivek - 20R21A6750

Department of Computer Science & Engineering(DS,AIIML,DS)

DECLARATION

I hereby declare that the project entitled “**Chatbot for Arundathi Hospital**” is the work done during the period from **August 2022 to December 2022** and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of technology in Computer Science and Engineering from MLR Institute of Technology, Hyderabad. The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

S. Ritesh Dhyan - 20R21A6752

P. Sai Kumar - 20R21A6751

K. Sri Prabha - 20R21A6737

R. Vivek - 20R21A6750

LIST OF FIGURES

Figure	Name of the figure	Page Number
1	NLP	7
2	Architecture Design	10
3	download py charm	14
4	choose installation path	14
5	choose start menu folder	15
6	Installation	15
7	Complete the setup	16
8	create a project	16
9	select the file	17
10	select chatbot project	17
11	package installation	18
12	check packages	18
13	Main Interface	23
14	color interface 1	24
15	color interface 2	24
16	color interface 3	25
17	color interface 4	25
18	conversation	26
19	file section	26
20	options section 1	27
21	options section 2	27
22	help section	28
23	about arubot	28
24	arubot developers	28

ABSTRACT

Chat bot[2] is an automated way of communication with the users in human understandable language. Chat bots[2] are used in applications such as e-commerce customer service, call centers and Internet gaming. Chat bots[2] are programs built to automatically engage with received messages. Chat bots[2] can be programmed to respond the same way each time, to respond differently to messages containing certain keywords and even to use machine learning[7] to adapt their responses to fit the situation. The chat bot which we are building is a task oriented[2] chatbot. A developing number of hospitals, nursing homes, and even private centers, presently utilize Chat bots[2] for human services. These bots connect with potential patients visiting the hospital, helping them discover specialists, facilities available in the hospital, doctors availability time, Specialized treatments and getting them access to the correct treatment. This healthcare chat bot[2] system will help hospitals to provide healthcare support 24 x 7, it answers deep as well as general questions. The Chat bot is designed using various packages from python such as tensor flow[9], nltk(Natural Language Tool Kit)[9], json[10] , tkinter[9] etc.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

A chat bot (conversational interface, AI agent) is a computer program that can understand human language and communicate with a user via a website or a messaging app. Chat bot can help users by delivering the consistent information about an organization which avoids dishing out irrelevant information to customers. Chat bots can be as simple as basic programs that answer a simple query with a single-line response, used as digital assistants that learn and evolve to deliver increasing levels of personalization as they gather and process information.

1.2 Types of Chatbots

There are two main types of chat bots:

- **Task-oriented** (declarative) chat bots are single-purpose programs that focus on performing one function. Using rules, NLP, and very little ML, they generate automated but conversational responses to user inquiries. Though they do use NLP so end users can experience them in a conversational way, their capabilities are fairly basic. These are currently the most commonly used chat bots.
- **Data-driven and predictive** (conversational) chat bots are often referred to as virtual assistants or digital assistance and they are much more sophisticated, interactive, and personalized than task-oriented chat bots. They apply predictive intelligence and analytics to enable personalization based on user profiles and past user behavior. Digital assistants can learn a user's preferences over time, provide recommendations. In addition to monitoring data and intent, they can initiate conversations. Examples like apple's Siri and Amazon's Alexa. The chatbot which we are creating is a task oriented type.

1.3 Applications :

- **Messaging apps** : Many companies use chatbots run on messaging apps or simply using SMS. They are used for B@C customer service. A 2017 study showed that 4% of companies used chatbots. According to a 2016 study, 80% of businesses said they intended to have one by 2020.
- **Company internal platforms** : Several companies are exploring ways to use chatbots internally, such as for Customer Service, Human Resources, and Internet-of-Things (IoT). In order to automate certain simple, yet time-consuming

processes when requesting sick leave, Overstock.com, for example, has launched a chatbot called Mila. Many large companies such as lloyds Banking Group, Royal Bank of Scotland, Renault and Citroën are now using automated online assistants instead of call centres with humans to provide a first point of contact. A SaaS chatbot ecosystem has steadily grown since Facebook's Mark Zucker berg announced that Messenger would allow chatbots into the app. A reference architecture for Intelligent Chatbots is being designed by IT architects in large companies, such as hospitals and aviation organizations, which unlock and share knowledge and experience more efficiently within the organization and reduce the errors made by expert service desks. Artificial intelligence, such as image moderation, natural language understanding, natural language generation, machine learning, and deep learning, is used to create Intelligent Chatbots.

- **Customer service** : Many high-tech banking groups are trying to combine computerized AI-primarily based totally answers along with chatbots into their customer support on the way to offer quicker and less expensive help to their customers who have become more and more more cushty with technology. In particular, chatbots can successfully behaviour a dialogue, typically changing different communicate equipment along with email, phone, or SMS. In banking, their important software is associated with short customer support answering not un usual place requests, in addition to transactional support.
- **Health Care** : Chatbots also are acting with inside the healthcare industry. A take a look at counseled that physicians with inside the United States believed that chatbots might be maximum useful for scheduling medical doctor appointments, finding fitness clinics, or presenting medicine information.
- **Travel and hospitality** : Just like e-commerce and retail, customers in travel and hospitality prefer personalized experiences (planning a vacation can get quite tedious!). Chatbots can provide a range of services in customer service from customizing itineraries to managing bookings and reservations. Be it hotel reservations or restaurant reservations, chatbots can take care of the entire booking process for your customers and send confirmations within seconds.
- **Ordering and delivery** : Famous restaurant chains like Burger King and Taco bell have deployed chatbots to help customers place their orders with minimal effort.

CHAPTER 2

LITERATURE SURVEY

Now a days the use of chat bot is popular in large of applications especially in systems that provide an intelligence support to user. There is no previous research on the topic which we have selected. So, we are the first team to do a project on the chat bot for Arundathi hospital. We are mainly concentrating on making the chat bot answer the basic queries regarding the hospital such as Location of the hospital, Facilities available in the hospital, Contact numbers of the hospital, Doctors of each department etc. We used machine learning approach to train the bot. To make this chatbot we used NLP i.e Natural language processing to pre process the data to the chat bot and make the bot understand the human language and respond back to the user in the same way. We trained the bot using json data set. The data in the json file is stored in the form of dictionary data structure. There are many other packages which we used to make the chat bot such as Numpy, pandas, Tensor flow, nltk, Tkinter, random, json. Further we would like to create a web application of the hospital and import this chat bot into the web application. We also want to make the chat bot more effective by adding some more features like booking an appointment for doctor consultancy, Showing the reports of patients if they have taken any tests in the hospital, Doctors live availability in the hospital

2.1 EXISTING SYSTEM

Slush : Customer expect an immediate response when they reach out to a company for any support. To deliver real-time and high-quality support to a large number of any support. To deliver real time and high quality support to a large number of customers is not an easy job. An AI enabled chat bot is the best option to handle the “n” number of conversations with 24/7 engagement.

Mariott : Here is a customer service chat bot example in the hospitality industry to get you started. Marriott used chat bot implementation ideas and made them available to guests via text message. Bots allow guests to request basic hotel services, essentially acting as an in-phone concierge. This exempts middleman involvement and enables requests to be met quickly and efficiently.

Babylon Health : The healthcare industry has made the best of the opportunity to capitalize on chatbots. The rise of health & fitness apps is the best chatbot example. Healthcare bots can help in personalizing the user experience based on the health needs of the user. Babylon Health is a well-known British online subscription service that has used bots to offer consultations based on personal medical history, and can even connect you with a live video consultation from a doctor.

2.1.1 LIMITATIONS OF EXISTING SYSTEM

- The bot might not understand every question asked by the user.
- Difficult to create.
- Limited Responses.

CHAPTER 3

PROPOSED SYSTEM

3.1 PROPOSED SYSTEM

The Proposed system is a basic chatbot which would answer the basic queries only regarding to the hospital. The questions can be such as location of the hospital, facilities available in the hospital, doctors availability, Multi speciality treatments in the hospital, about hospital, founder of the hospital etc. This chat bot uses Natural language processing to understand the human language and respond back in the same way.

3.1.1 ADVANTAGES OF PROPOSED SYSTEM

- Quick response to the questions.
- The Bot will help you to know about the hospital.
- 24/7 availability.
- Meet the Customer Expectations.
- Reduce User Service Costs
- Interacts almost similar to a human with a user.

3.2 SYSTEM REQUIREMENTS

3.2.1 HARDWARE REQUIREMENTS

A PC or Laptop with following Specifications

- CPU – INTEL i5 or above processor
- RAM – 8GB MINIMUM
- ROM – 256GB

3.2.2 SOFTWARE REQUIREMENTS

Python Programming Language and preinstalled libraries We used Windows operating System to run this project.

We used Py charm Community to create the virtual environment for the project.

Py charm Installation process

- Download the required setup file of Pycharm Community.
- Run the package and set a path where to install the Software.
- Now, begin the installation procedure similar to any other software packages. Once the installation is successful, PyCharm asks you to import settings of the existing package if any.

Install the required packages

Pip install numpy

Pip install pandas

Pip install Tensorflow

Pip install json

Pip install tkinter

Pip install pickle

Pip install nltk

Pip install random

All the above packages need to be installed in order to run the project successfully.

3.3 CONCEPTS USED IN PROPOSED SYSTEM

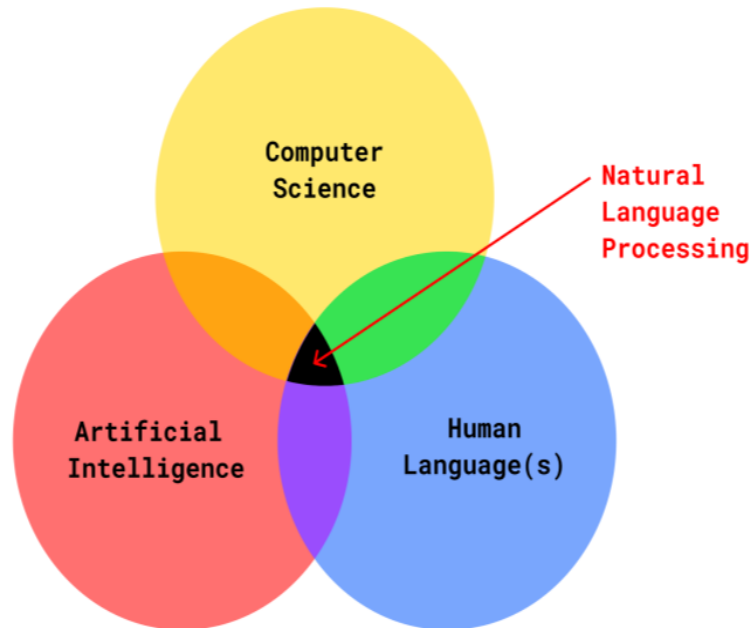


Fig 3.1 : NLP

We use Natural Language Processing to pre process the data before we train it to the machine.

The essence of Natural Language Processing lies in making computers understand the natural language. That's not an easy task though. Computers can understand the structured form of data like spreadsheets and the tables in the database, but human languages, texts, and voices form an unstructured category of data, and it gets difficult for the computer to understand it, and there arises the need for Natural Language Processing.

There's a lot of natural language data out there in various forms and it would get very easy if computers can understand and process that data. We can train the models in accordance with expected output in different ways. Humans have been writing for thousands of years, there are a lot of literature pieces available, and it would be great if we make computers understand that. But the task is never going to be easy. There are various challenges floating out there like understanding the correct meaning of the sentence, correct Named-Entity Recognition(NER), correct prediction of various parts of speech, coreference resolution(the most challenging thing in my opinion).

Computers can't truly understand the human language. If we feed enough data and

train a model properly, it can distinguish and try categorizing various parts of speech(noun, verb, adjective, supporter, etc...) based on previously fed data and experiences. If it encounters a new word it tried making the nearest guess which can be embarrassingly wrong few times.

It's very difficult for a computer to extract the exact meaning from a sentence. For example – The boy radiated fire like vibes. The boy had a very motivating personality or he actually radiated fire? As you see over here, parsing English with a computer is going to be complicated.

There are various stages involved in training a model. Solving a complex problem in Machine Learning means building a pipeline. In simple terms, it means breaking a complex problem into a number of small problems, making models for each of them and then integrating these models. A similar thing is done in NLP. We can break down the process of understanding English for a model into a number of small pieces.

It would be really great if a computer could understand that San Pedro is an island in Belize district in Central America with a population of 16, 444 and it is the second largest town in Belize. But to make the computer understand this, we need to teach computer very basic concepts of written language.

So let's start by creating an NLP pipeline. It has various steps which will give us the desired output(maybe not in a few rare cases) at the end.

Step 1: Sentence Segmentation

Breaking the piece of text in various sentences.

Step 2: Word Tokenization

Breaking the sentence into individual words called as tokens. We can tokenize them whenever we encounter a space, we can train a model in that way. Even punctuations are considered as individual tokens as they have some meaning.

Step 3: Predicting Parts of Speech for each token

Predicting whether the word is a noun, verb, adjective, adverb, pronoun, etc. This will help to understand what the sentence is talking about. This can be achieved by feeding the tokens(and the words around it) to a pre-trained part-of-speech classification model. This model was fed a lot of English words with various parts of speech tagged to them so that it classifies the similar words it encounters in future in various parts of speech. Again, the models don't really understand the 'sense' of the words, it just

classifies them on the basis of its previous experience. It's pure statistics.

Step 4: Lemmatization

Feeding the model with the root word.

Step 5: Identifying stop words

There are various words in the English language that are used very frequently like 'a', 'and', 'the' etc. These words make a lot of noise while doing statistical analysis. We can take these words out. Some NLP pipelines will categorize these words as stop words, they will be filtered out while doing some statistical analysis. Definitely, they are needed to understand the dependency between various tokens to get the exact sense of the sentence. The list of stop words varies and depends on what kind of output are you expecting.

Step 6.1: Dependency Parsing

This means finding out the relationship between the words in the sentence and how they are related to each other. Just like we trained a Machine Learning model to identify various parts of speech, we can train a model to identify the dependency between words by feeding many words.

Step 6.2: Finding Noun Phrases

We can group the words that represent the same idea. We can use the output of dependency parsing to combine such words. Whether to do this step or not completely depends on the end goal, but it's always quick to do this if we don't want much information about which words are adjective, rather focus on other important details.

Step 7: Named Entity Recognition(NER)

NER systems look for how a word is placed in a sentence and make use of other statistical models to identify what kind of word actually it is

Step 8: Coreference Resolution:

it isn't possible to understand that both the tokens are same because it treats both the sentences as two different things while it's processing them. Pronouns are used with a high frequency in English literature and it becomes difficult for a computer to understand that both things are same.

CHAPTER 4

ARCHITECTURE DESIGN

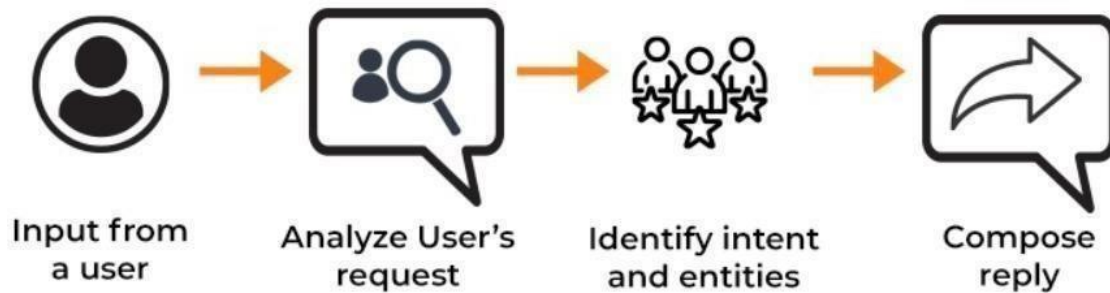


Fig 6.1 : Architecture Design

4.1 Working of the chatbot

Step1: The user enters some text.

Step2 : The input is analyzed by Natural Language Tool Kit(Natural Language Processing) which helps in assessing the intents and entities of the text entered by the user and create response based on the contextual analysis of the text.

Step 3 :Now it's time to really get into the nitty-gritty of how AI chat bots work. There are five major steps involved — tokenizing, normalizing, recognizing entities, dependency parsing, and generation — for the chatbot to read, interpret, understand, and formulate and send a response. Let's take a closer look.

- **Tokenizing:** The chatbot starts by chopping up text into pieces (also called 'tokens') and removing punctuation
- **Normalizing:** Next, the bot removes details that aren't relevant and converts words to their "normal" version, for example by making everything lowercase
- **Recognizing Entities:** Now that the words are all normalized, the chatbot seeks to identify which type of thing is being referred to. For example, it would identify North America as a location, 67% as a percentage, and Google as an organization
- **Dependency Parsing:** For the next step, the bot identifies the role each word plays in the sentence, such as noun, verb, adjective, or object

- **Generation:** Finally, the chatbot generates a number of responses using the information determined in all the other steps and selects the most appropriate response to send to the user.

Step 4 : If the Step 3 is accurately executed by the bot then it displays a correct output randomly from a list of responses.

CHAPTER 5 MODULES

The modules or libraries used in this project are :

5.1Tensorflow

TensorFlow is an open-source library for fast numerical computing. It was created and is maintained by Google and was released under the Apache 2.0 open source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API. Unlike other numerical libraries intended for use in Deep Learninglike Theano, TensorFlow was designed for use both in research and development and in production systems, not least of which is Rank Brain in Google search and the fun Deep Dream project. It can run on single CPU systems and GPUs, as well as mobile devices and large-scale distributed systems of hundreds of machines.

5.2Nltk

programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

5.3 Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

5.4 Json

The full form of JSON is Java Script Object Notation. It means that a script file which is made of text in a programming language, is used to store and transfer the data. Python supports JSON through a built-in package called json. To use this feature, we import the json package in Python script. The text in json is done through quoted – string which contains the value in key-value mapping with { }.

5.5 Pickle

Pickle in Python is primarily used in serializing and deserializing a Python object structure. In other words, it's the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network. The pickled byte stream can be used to re-create the original object hierarchy by unpickling the stream. This whole process is similar to object serialization in Java or .Net.

5.6 Random

Python Random Module is an in-built module of Python which is used to generate random numbers. The random module is used to generate the pseudo-random variables. It can be used to perform some action randomly such as to get a random number, selecting random elements from a list, shuffle elements randomly, etc.

5.7 Numpy

NumPy, which stands for Numerical Python, is a library consisting of Multidimensional array objects and a collection of routines for processing those arrays. Using Numpy, mathematical and logical operations on arrays can be performed. In this Python Numpy Tutorial, we will be learning about NumPy in Python, What is NumPy in Python, Data Types in NumPy, and more.

It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities.

5.8 Pandas

pandas is a fast, powerful, flexible and easy to use software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is a library which is built on the top of Numpy Library. Pandas has high performance.

It also provides time-series functionality and powerful group by functionality for performing split-apply-combine operations on datasets. It has a separate inbuilt data structures which is Series, Data frames and panels. Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState's ActivePython.

5.9 Warnings

Warnings are provided to warn the developer of situations that aren't necessarily exceptions. Usually, a warning occurs when there is some obsolete of certain programming elements, such as keyword, function or class, etc. A warning in a program is distinct from an error. Python program terminates immediately if an error occurs. Conversely, a warning is not critical. It shows some message, but the program runs. The warn() function defined in the 'warning' module is used to show warning messages. The warning module is actually a subclass of Exception which is a built-in class in Python.

CHAPTER 6

IMPLEMENTATION

Follow the below steps to download Py charm Application software.

Step 1 : Download the required package or executable from the official website PyCharm <https://www.jetbrains.com/pycharm/download/#section=window> Here you will observe two versions of package for Windows as shown in the screenshot given below –



Fig 8.1 : Download py charm

Step 2 : Download the community package (executable file) onto your system and mention a destination folder as shown below –

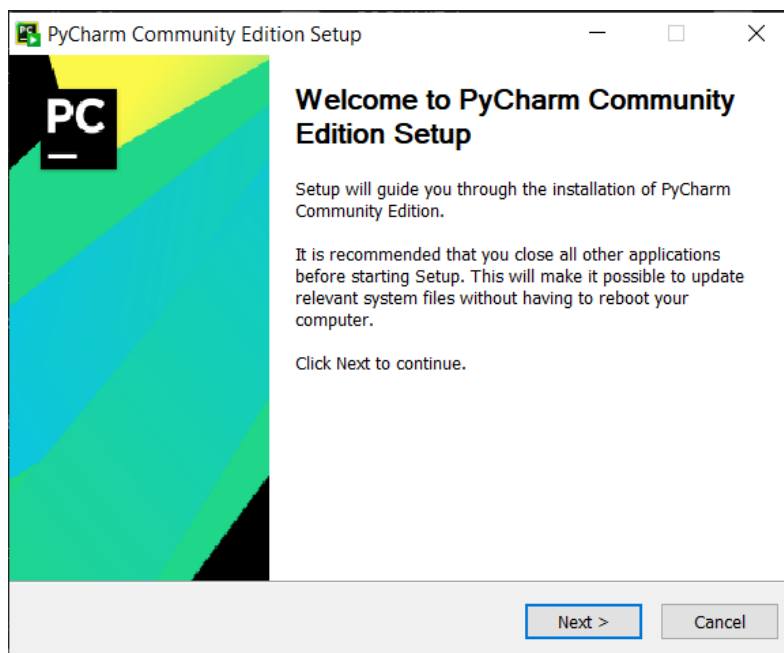


Fig 8.2 : Pycharm Setup

Step 3 : Now, begin the installation procedure similar to any other softwarepackage

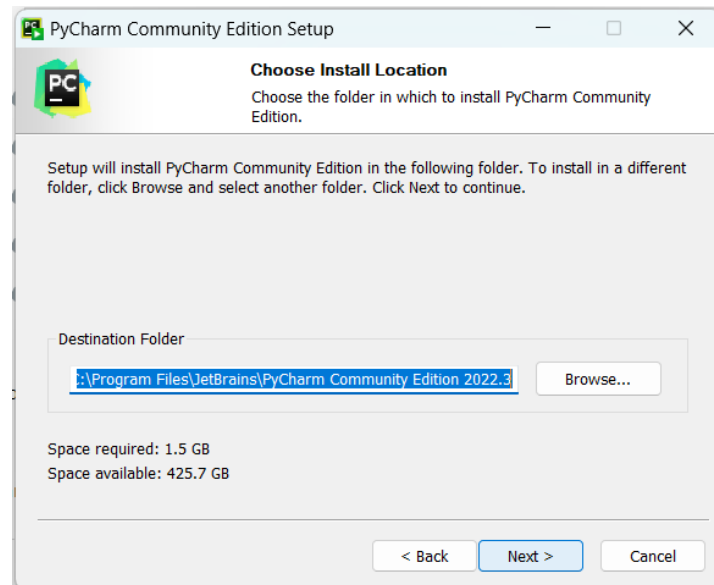


Fig 8.3 : Choose start Menu Folder

Step 4 : let the installation process complete

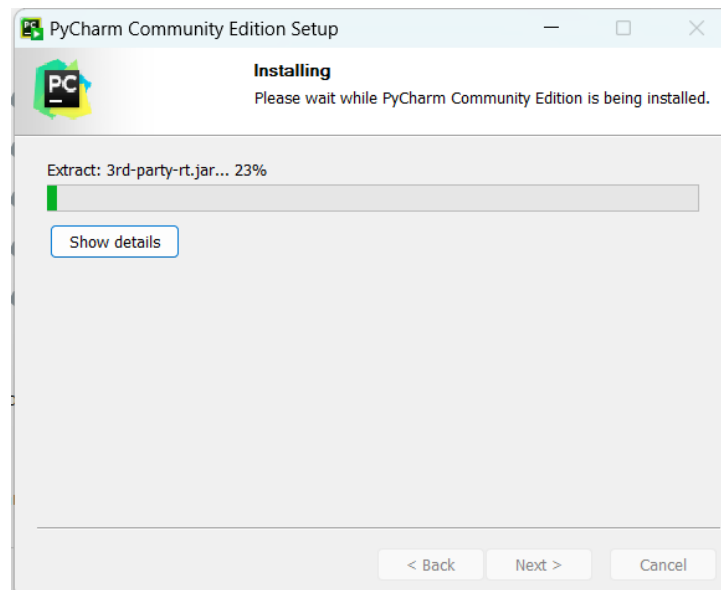


Fig 8.4 : Installing

Step 5 : Once the installation is completed click on reboot now to complete the whole installation process. The Reboot will hardly take 5 minutes of time.

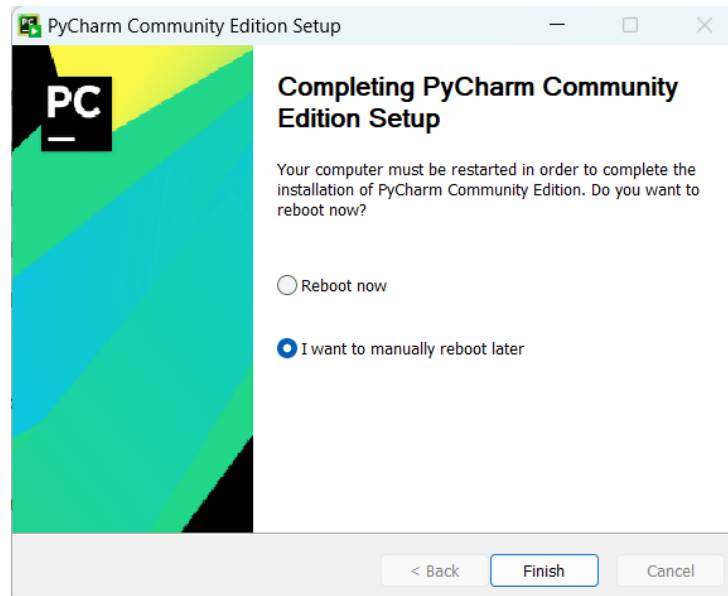


Fig 8.5 : Complete the setup

Step 6 : Open the Application and click on create a new project

- Once you click on it you will get a page like this.
- Then select the python interpreter or new virtual environment.
- After selecting the python interpreter click on create.

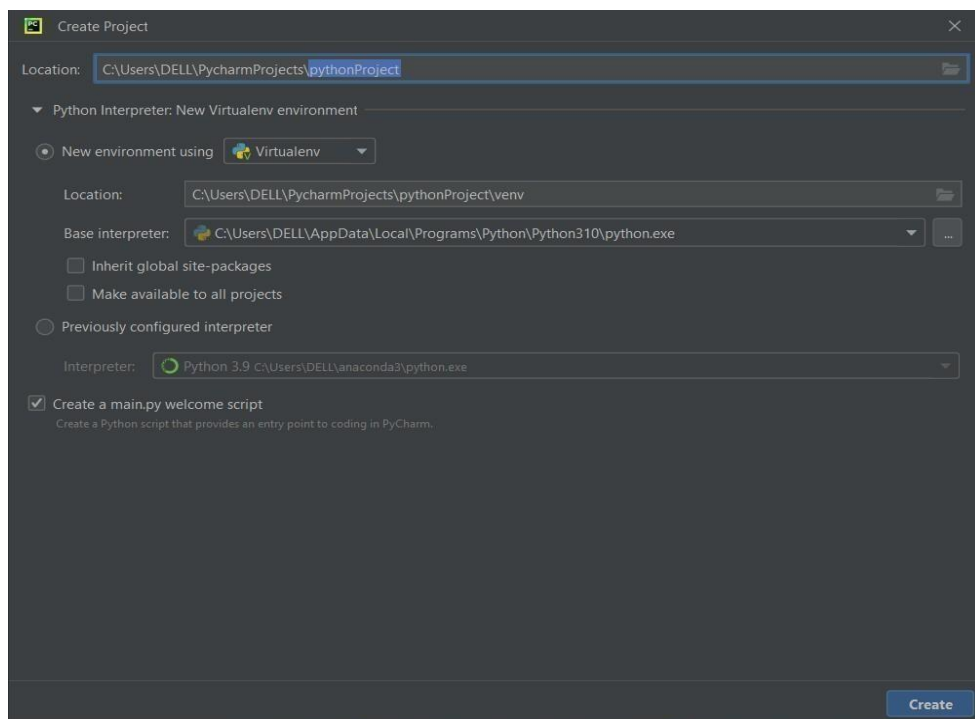


Fig 8.6 : create a project

Step 7 : At the top most left corner click on file.

- Click on the open

- A new tab will be opened where you need to select the file where the whole chatbot scripts are present.

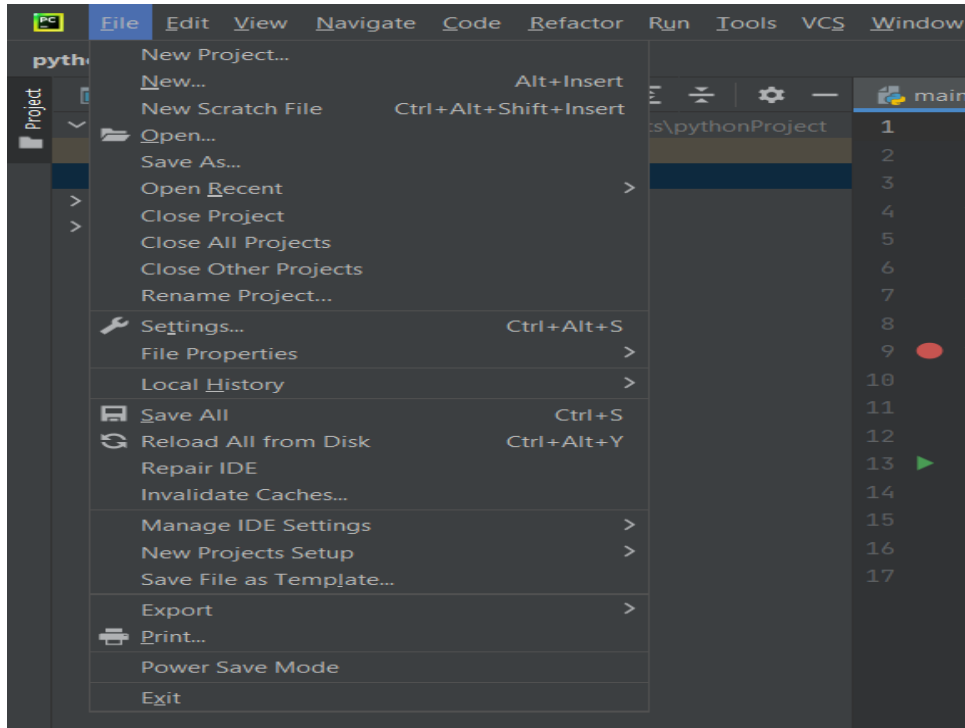


Fig 8.7 : select the file

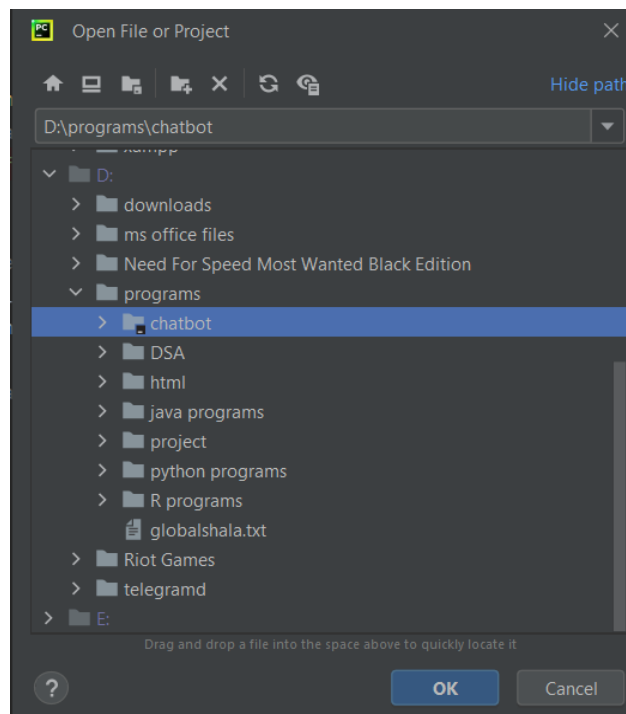


Fig 8.8 : select chatbot

Click on ok and the chatbot scripts will be included in the virtual environment.

Step 8 : After importing the project into the environment. Install all the required packages. Refer to Modules page to know what packages need to be installed for this project.

Use the command **pip install package_name** to install the required package in the terminal.

```

Terminal: C:\Windows\system32\cmd.exe
Try the new cross-platform PowerShell https://aka.ms/pscore6

File D:\programs\project\chatbot\venv\Scripts\activate.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.
+ CategoryInfo          : SecurityError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : UnauthorizedAccess

PS D:\programs\project\chatbot> pip install numpy
Requirement already satisfied: numpy in c:\users\dell\appdata\local\programs\python\python310\lib\site-packages (1.23.3)
WARNING: You are using pip version 22.0.4; however, version 22.3.1 is available.
You should consider upgrading via the 'C:\Users\Dell\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.

PS D:\programs\project\chatbot>

```

Fig 8.9 : package installation

Step 9 : Once you have installed all the packages you can check if the packages are installed in the environment by clicking on the python packages option in the bottom.

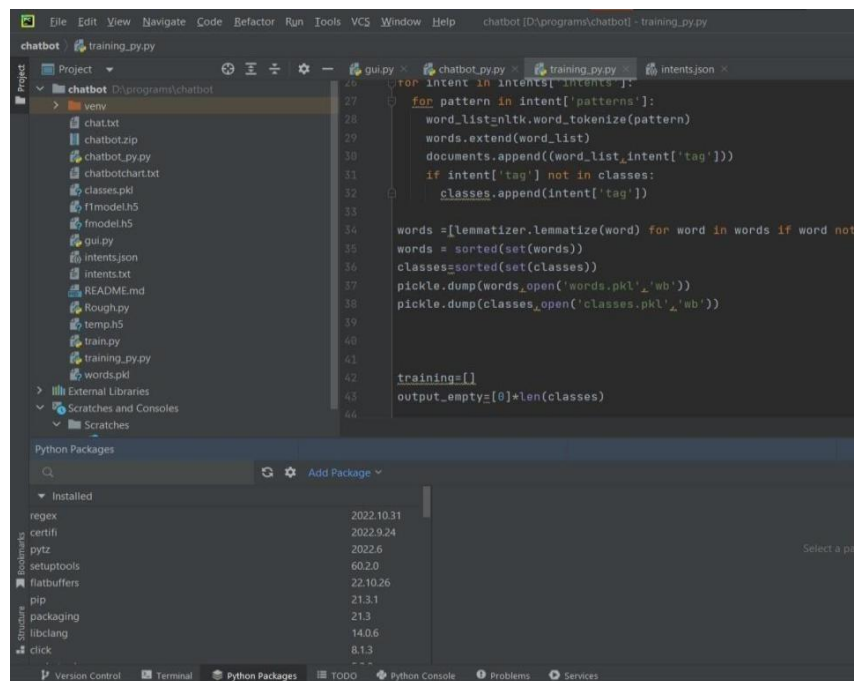


Fig 8.10 : check packages

Do not ignore the above steps, every step is crucial. So, follow the above steps and install the packages properly in order to run the project without errors.

Once you are down with the above process you can run the gui.py file to start conversation with the chatbot.

CHAPTER 7

CODE

7.1 chatbot_py.py file

This python file consists of pre processing the data to the chatbot using Natural Language processing library i.e nltk which makes the machine understand the human language and respond back in the same way.

Code :

```
import warnings
warnings.filterwarnings('ignore')
import nltk
import random
import numpy as np
import json
import pickle
from nltk.stem import WordNetLemmatizer
from tensorflow.keras.models import load_model
lemmatizer=WordNetLemmatizer()
nltk.download('punkt')
with open('intents.json') as json_file:
    intents = json.load(json_file)
words=pickle.load(open('words.pkl','rb'))
classes=pickle.load(open('classes.pkl','rb'))
model=load_model('finalmodel.h5')
def clean_up_sentence(sentence):
    sentence_words=nltk.word_tokenize(sentence)
    sentence_words=[lemmatizer.lemmatize(word) for word in sentence_words]
    return sentence_words
def bag_of_words(sentence):
    sentence_words=clean_up_sentence(sentence)
    bag=[0]*len(words)
    for w in sentence_words:
        for i,word in enumerate(words):
            if word == w:
                bag[i]=1
```

```

    return np.array(bag)
def predict_class(sentence):
    bow=bag_of_words(sentence)
    res=model.predict(np.array([bow]), verbose=0)[0]
    ERROR_THRESHOLD=0.75
    results=[[i,r] for i,r in enumerate(res) if r> ERROR_THRESHOLD]
    results.sort(key=lambda x:x[1],reverse=True)
    return_list=[]
    for r in results:
        return_list.append({'intent': classes[r[0]], 'probability':str(r[1])})
    return return_list
def get_response(intents_list,intents_json):
    tag=intents_list[0]['intent']
    list_of_intents=intents_json['intents']
    for i in list_of_intents:
        if i['tag']==tag:
            result=random.choice(i['responses'])
            break
    return result
print("GO! BOT IS RUNNING")
while True:
    message=input("")
    ints=predict_class(message)
    # print(ints)
    res=get_response(ints,intents)
    print(res)

```

7.2 Training_py.py

This python file consists of the code which trains the data or feeds the data to the machine using tensor flow library to make the machine respond to the questions asked by the user. We use machine learning approach to train the machine.

Code :

```
import random
import json
import pickle

import nltk
import pandas as pd
nltk.download('punkt')
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Activation,Dropout
from tensorflow.keras.optimizers import SGD, Adam
lemmatizer=WordNetLemmatizer()
with open('intents.json') as json_file:
intents = json.load(json_file)
words=[]
classes=[]
documents=[]
ignore_letters=['?','!','.',',']
for intent in intents['intents']:
    for pattern in intent['patterns']:
        word_list=nltk.word_tokenize(pattern)
        words.extend(word_list)
        documents.append((word_list,intent['tag']))
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
words =[lemmatizer.lemmatize(word) for word in words if word not in ignore_letters]
words = sorted(set(words))
classes=sorted(set(classes))
pickle.dump(words,open('words.pkl','wb'))
```

```

pickle.dump(classes,open('classes.pkl','wb'))
training=[]
output_empty=[0]*len(classes)
for document in documents:
    bag=[]
    word_patterns=document[0]
    words = [lemmatizer.lemmatize(word) for word in words if word and word not in
ignore_letters]

    for word in words:
        bag.append(1) if word in word_patterns else bag.append(0)
        output_row=list(output_empty)
        output_row[classes.index(document[1])]=1
        training.append([bag,output_row])
random.shuffle(training)
training=np.array(training)
train_x=list(training[:,0])
train_y=list(training[:,1])
print(len(train_x))
model=Sequential()
model.add(Dense(256,input_shape=(len(train_x[0]),),activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]),activation='softmax'))

sgd=SGD(lr=0.01,decay=1e-6,momentum=0.9,nesterov=True)
model.compile(loss='categorical_crossentropy',optimizer=sgd,metrics=['accuracy'])
hist = model.fit(np.array(train_x),np.array(train_y),epochs=200,batch_size=5,verbose=1)
model.save('fmodel.h5', hist)
print("Training Done")

```

CHAPTER 8

OUTPUT

1. To run this project run the gui.py file which runs the interface where the user and chatbot communication happens.

We named this chatbot as Arubot.

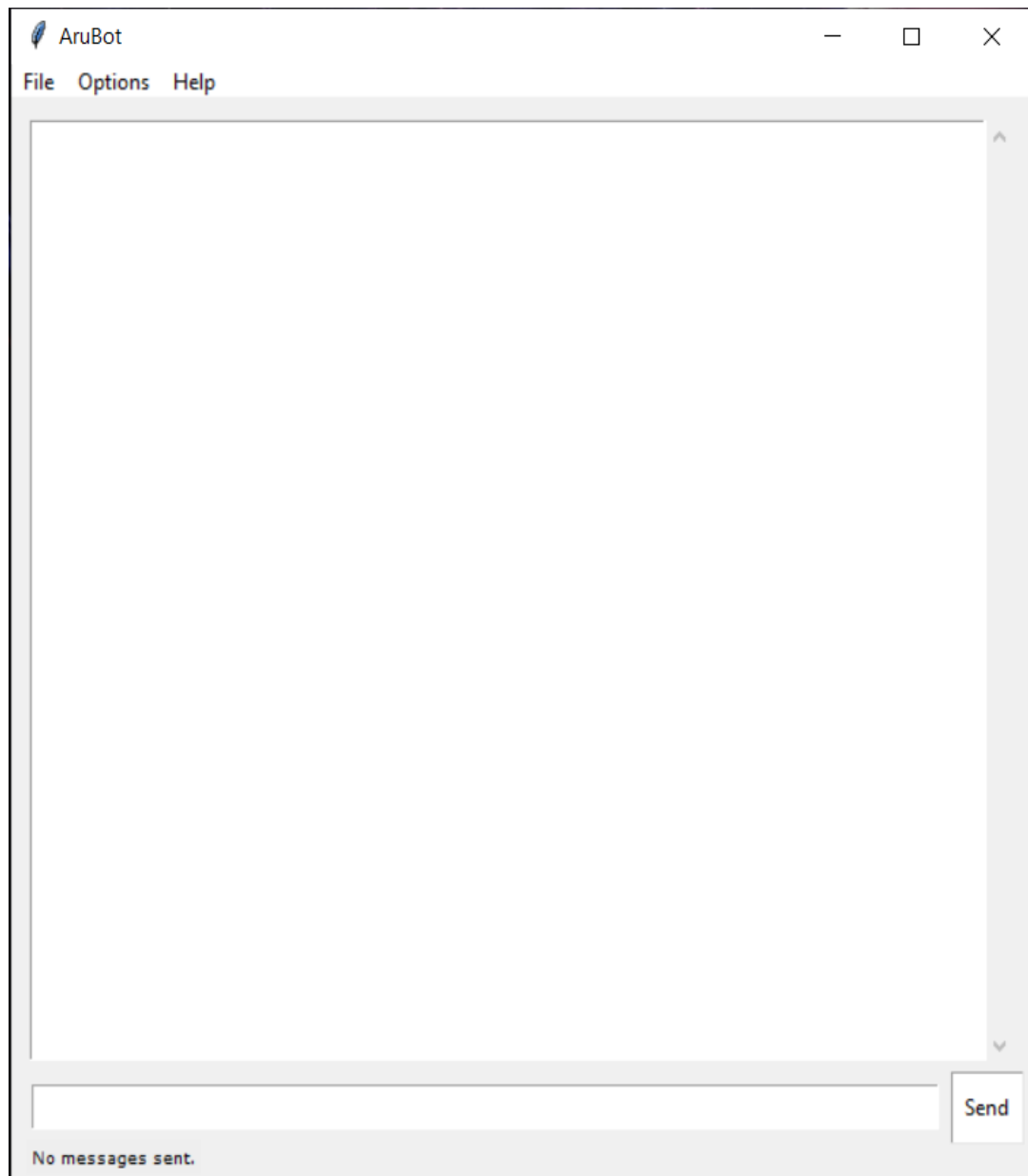


Fig 10.1 : Main interface

2. We also have different colors interfaces.

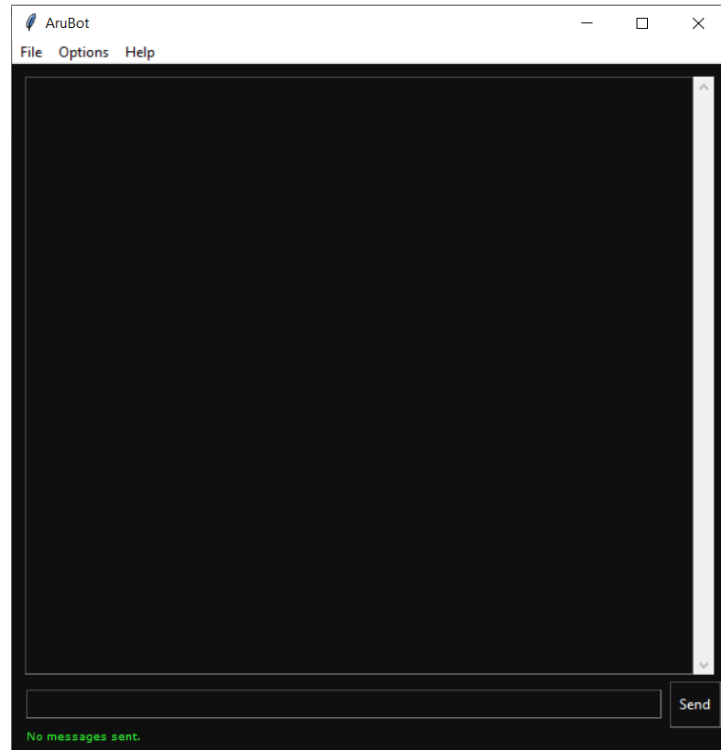


Fig 10.2 : color interface 1

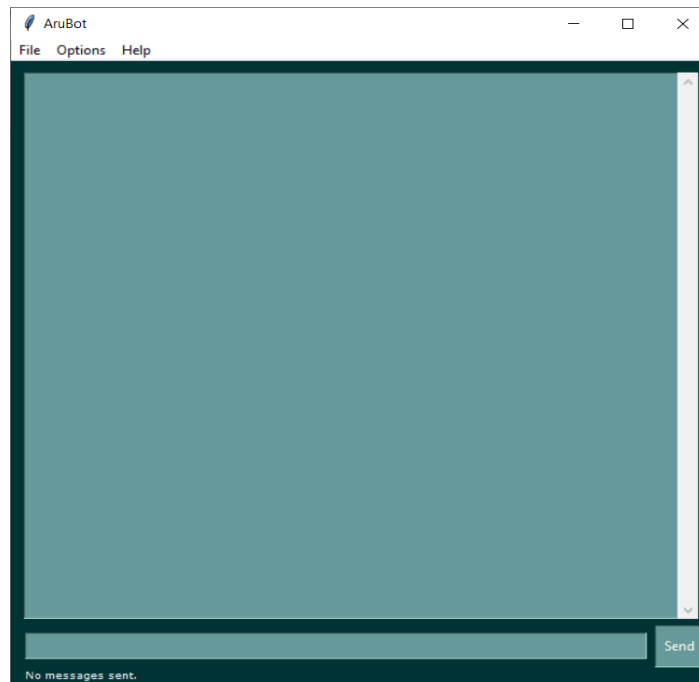


Fig 10.3 : color interface 2

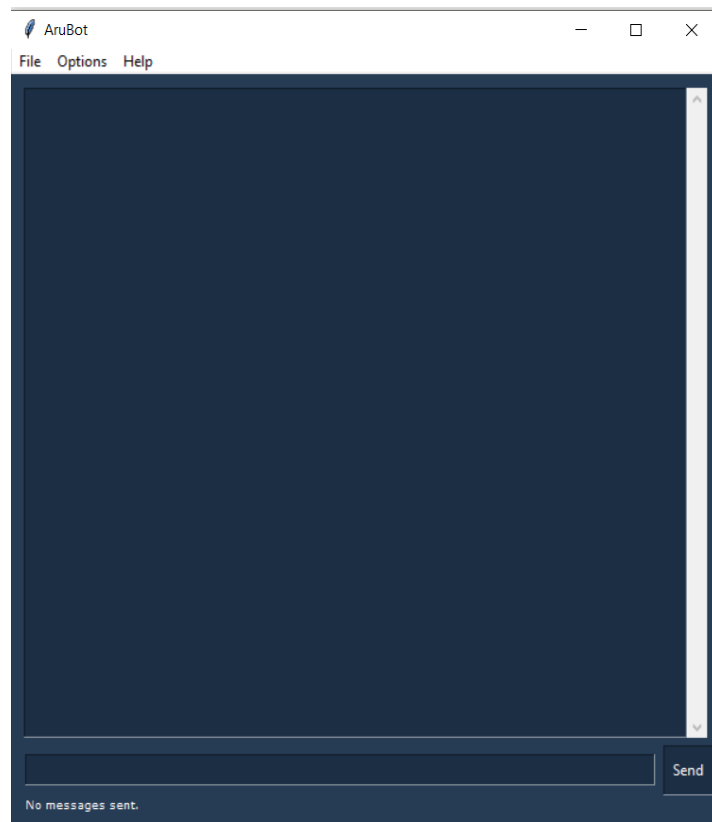


Fig 10.4 : color interface 3

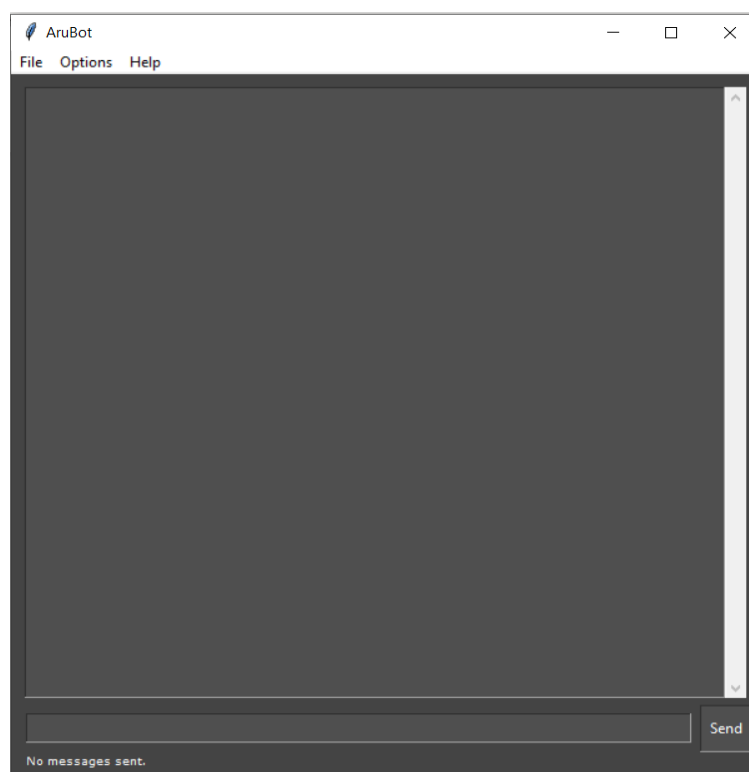


Fig 10.5 : color interface 4

3. In the interface you can see that the user enters a question and the chatbot answers according to the way it is trained.

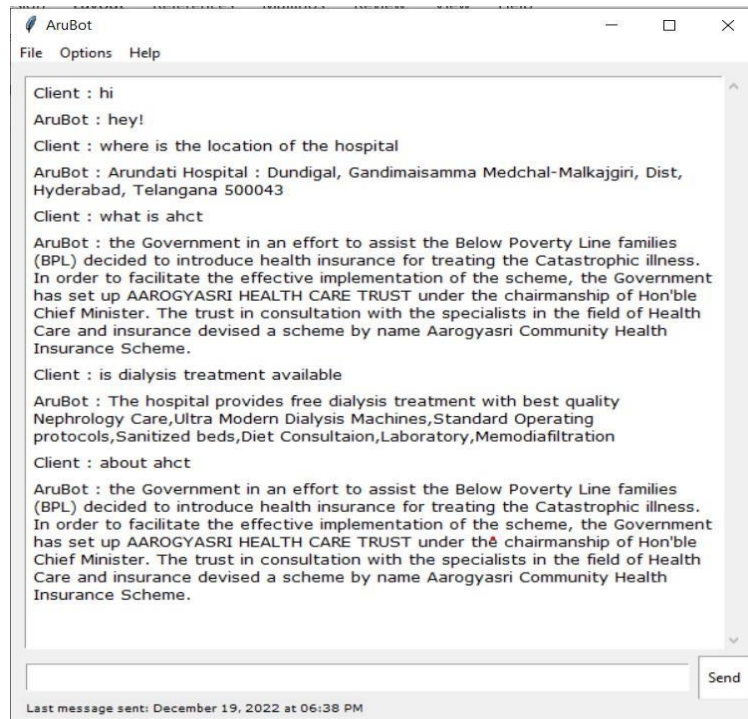


Fig 10.6 : conversation

4. We also have various other options in the interface like File, Options, Help

- The File option consists of Exit and clear chat. When we click on the Exit option the interface will close and if we click on clear chat the whole chat will be cleared.

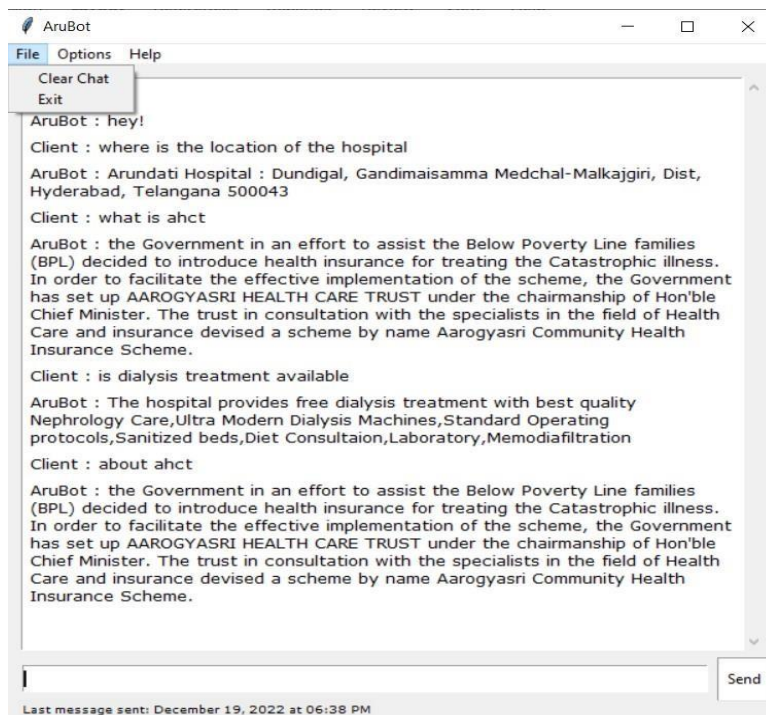


Fig 10.7 : file section of interface

5. In the options tab we can see different font styles and theme styles.

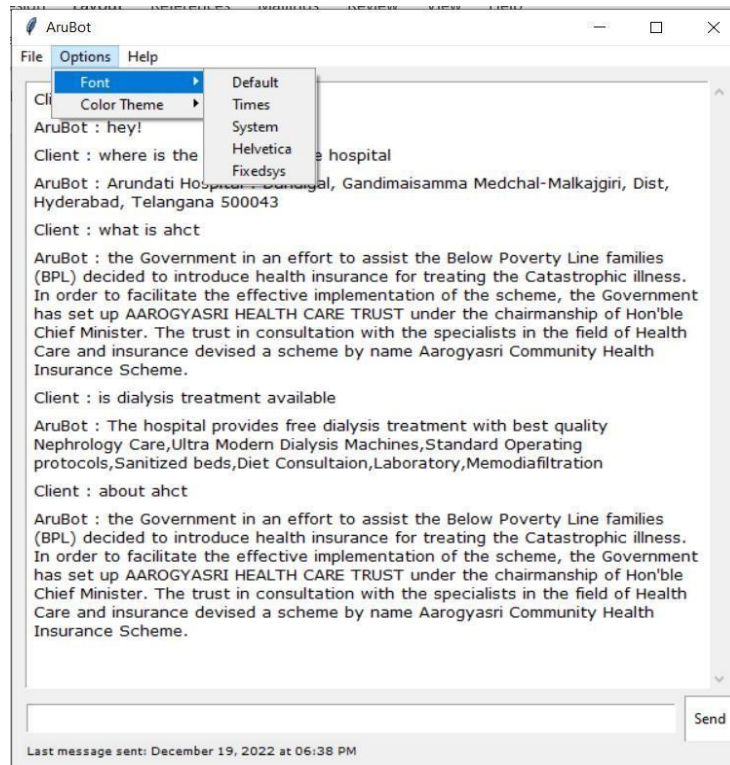


Fig 10.8 : options section 1

Different Themes are also available.

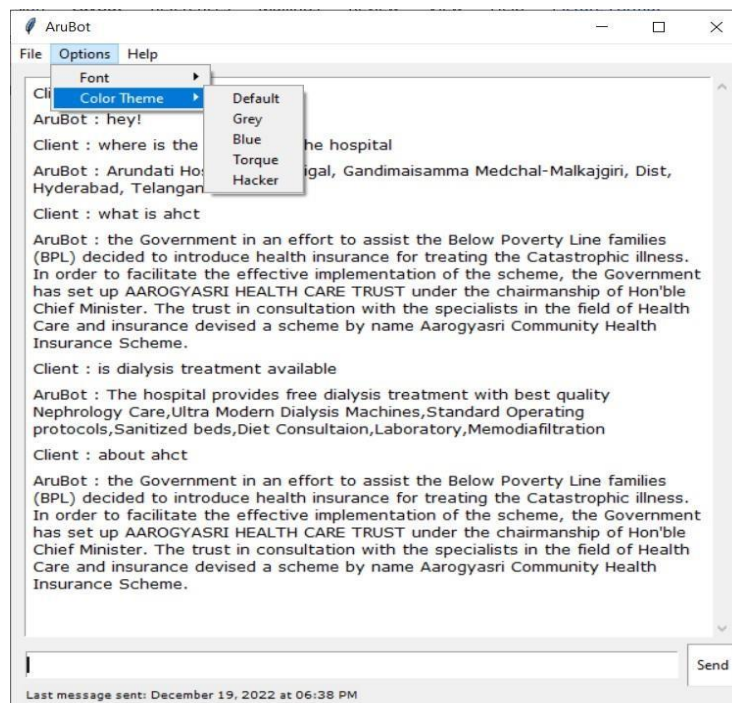


Fig 10.9 : options section 2

6. In the help option you will be able to see the Arubot developers and about arubot option.

7. If we click on Arubot developers the team members list is shown and if you click on about arubot we will be able to see about the bot.

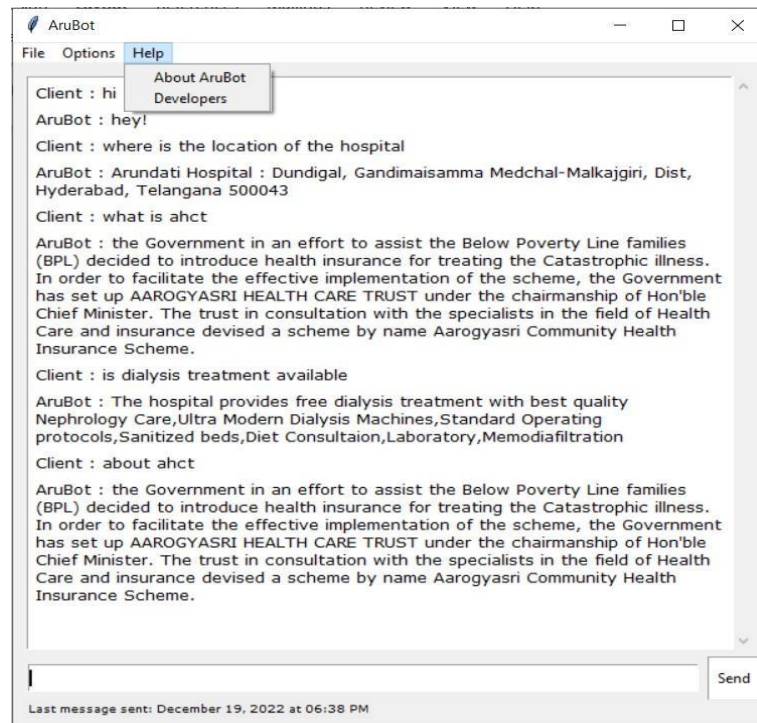


Fig 10.10 : Help section

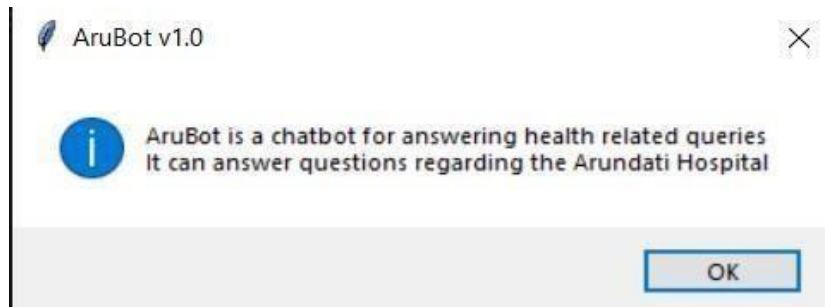


Fig 10.11 : about arubot



Fig 10.12 : arubot developers

CHAPTER 9

CONCLUSION

Chatbots are going to explode and can be really dominating in future. Chatbots can provide a new and flexible way for users. They are giving AI something better to do. Chatbots results in smart conversation and is advancing at an unprecedented rate with each new development. They can be used in various domains like a chatbot for company or organizations, a chatbot for web application, a chatbot for hospital etc.

Chatbots are more effective than people in reaching out to the users via messaging apps or websites. They have the potential to become a useful information gathering tool in the near future. Chatbots are artificial intelligence that can provide highly tailored communication to the user while minimizing the workload of employees. The organizations or companies can easily offer high-quality support and conflict resolution any time of day, and for a large quantity of users simultaneously.

CHAPTER 9

REFERENCES

- [1] Chatbots : An Introduction and Easy Guide to make your Own –Oisin Muldowney.
- [2] Designing Bots : Creating Conversational Experiences – Amir Shevat
- [3] Conversational Interfaces : Principles of Successful Bots, Chatbots & Messaging apps – Mariya Yao
- [4] Designing Conversational Interfaces – Alper Cugun
- [5] Building Chatbots with Python : Using Natural Language Processing and Machine Learning - Sumit Raj
- [6] Chatbots : 100 Successful Business and Brand bots by Adelyn Zhou
- [7] The Conversational Interface : Talking to Smart Devices by Mc Tear, Michael, Callejas.