A Mini Project Report
On
Creating your own Chatbot

Submitted in partial fulfillment of the requirements for the award of the degree
of
BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY
by
Mr. G.LOKESHPRASANTH (190171841L)

Under the guidance of

Mrs T.LAVANYA

Department of Information Technology
University College of Engineering,
KAKATIYA UNIVERSITY
Kothagudem, Bhadradri kothagudem Dist,
Telangana-507101
2018-2022

UNIVERSITY COLLEGE OF ENGINEERING
BHADRADRI KOTHAGUDEM
A constituent college of KU,Warangal , Approved by AICTE, New Delhi
EAMCET Code:KUCE/KUCESF
ECET Code:KUCE/KUCESF

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that project entitled "Creating your own Chatbot" is a bonafide work carried out by **Mr. G.Lokeshprasanth (190171841L)** in partial fulfillment for the award of Bachelor of Technology in Department of Information Technology, University College of Engineering, Kakatiya University, kothagudem during the year 2018-2022 under my supervision and guidance. The result embodied in this project work done has not been submitted to any University or Institute for the award of any Degree or Diploma

INTERNAL GUIDE

NAME:**Mrs T.LAVANYA**

DESIGNATION: **Assist Professor**

DEPARTMENT: **INF**

HEAD OF THE DEPARTMENT

NAME:**Dr. T.ARCHANA**

DESIGNATION:**Professor & Head**

DEPARTMENT: **INF**

PRINCIPAL

Dr. N.RAMANA

EXTERNAL EXAMINER

BATCH CERTIFICATE

DEPARTMENT OF INFORMATION TECHNOLOGY

This is to certified that the project entitled

Deep Surveillance with Deep Learning - Intelligent Video Surveillance Project

SNO	NAME	PIN
1	S.NIKHIL SAI	18017T1815
2	Y.RAVISHANKAR	190171822L
3	E.PRASHANTH	190171824L
4	B.SAI KIRAN	190171830L
5	U.THARUN	190171832L
6	G.LOKESHPRASANTH	190171841L

of 4th year B.tech in Information Technology in the academic year of **2018-2022**

INTERNAL GUIDE

NAME:**Mrs T.LAVANYA**

DESIGNATION: **Assist Professor**

DEPARTMENT: **INF**

HEAD OF THE DEPARTMENT

NAME:**Dr. T.ARCHANA**

DESIGNATION:**Professor & Head**

DEPARTMENT: **INF**

PRINCIPAL

Dr. N.RAMANA

EXTERNAL EXAMINER

ACKNOWLEDGMENT

I sincerely thank to Dr. N. RAMANA sir, principal of UNIVERSITY COLLEGE OF ENGINEERING, KAKATIYA UNIVERSITY for providing an opportunity to do our project work “Creating your own Chatbot”.

I sincerely thank to Head of Department of Information Technology Engineering, Dr. T.Archana mama for their guidance and encouragement in carrying out this project work.

I sincerely thank to Mentor, T. Lavanya madam for her guidance and encouragement in carrying out this project work.

I also thank to all staff members and my team members of UCEKU who rendered their help during the period of our project work.

CONTENT

ABSTRACT

The use of chatbots evolved numerous fields in recent years including marketing, supporting system, education, health care and entertainment. In this project we discuss motivations that drive the use of chatbot and clarify chatbots in variety of areas

Now a days its's need of an personal chatbot to remember our daily tasks. These are handle by systems and No human efforts are required like our personal supportes, so this type of chatbots are already implemented in so many areas and in AI technologies.

After clarifying necessary required concepts, we move on to a chatbot classification based on various categories such as area of knowledge they refer to, the need they serve and other.

Furthermore, we present general architecture of modern chatbots while also mentioning main platform for their creation. Our engagment with the subject so far reassures us of the prospects of chatbots and encourage us study the in a greater extent and depth.

1.INTRODUCTION

This document covers an introduction to the project including the context, a description of aims and objectives, a description of what has been achieved, contributions and the structure of the report.

Context:

Although the admissions process works properly as it is, it is very difficult and time consuming to contact a member of staff of the university. However, the problem would be partially solved if the applicant could talk to a convincing chatbot, able to respond to their concerns with information about admissions, booking accommodation, paying fees in instalments and what pre-sessional courses are on offer.

The chatbot should be able to communicate with a user in a way similar to the following:

- ChatBot: Hello how can I help you?
- User: What is the minimum score of IELTS required for entry into MSc Computer Science?
- ChatBot: The minimum requirement is a total score of 6.0 with no less that 5.5 in any section.
- User: What other test is accepted by the department? ChatBot: We also accept TOEFL and the certificate of Proficiency of the University of Cambridge.
- User: What are the entry requirements for the MSc in Computer Security?
- ChatBot: To enter the programme, you need to have at least an Upper Second Class (2.1) degree or an international equivalent in Computer Science or a closely related discipline and a solid foundation in programming.

Aims and Objectives:

The aim of this project is to contribute to the solution of the problem of direct communication between applicants and the university.

The main objectives of the project are as follows:

Database: To develop a database were all the relevant information about questions, answers, keywords, logs and feedback will be stored.

Algorithm: To develop a keyword matching algorithm and a string distance comparison algorithm and combine them in order to retrieve the best possible answer.

Interface: To develop a web interface which aims to give the ability to potential students and their families to submit questions in a chatbot and get convincing replies. The web system can be hosted on any computer that supports an operating system on which the Java Virtual machine will be able to load. Furthermore the Hibernate, Spring, Apache Wicket, Lingpipe, the Jazzy spelling, the Stanford NLP parser and Google search libraries should be available.

2.OBJECTIVES OF PROJECT

Chatbots today mimic human conversations. Thanks to their learning ability and 24/7 presence. They can optimize communication and create real engagement. The client must build a creative and user-friendly interface for communication. Over-burdening your chatbot with traits and crafting it to ace all undertakings will probably set you up for disappointment.

A chatbot can communicate with a real person behaving like a human. Let's list down objectives and purpose of chatbots.

You can create chatbots for any business the same as you recruit a person for any department of your company. Whether you are a:

- Wedding Planner
- Insurance Assistant
- Education Consultant
- Legal Assistant
- A real estate business
- Recruiter
- Travel Agency
- Hospital or a Beautician!

3.SOFTWARE REQUIREMENTS AND SPECIFICATIONS

Software requirements:
Programming language: Python language
Software: Python IDLE 3.10 version
OS: Windows 8, 10, 11(Home Single)
Online Interpreter: replit.com
Front End: HTML, CSS, JS
Back End: Python
Prerequisites: Web Development, Python, Deep Learning, Machine Learning

Hardware requirements:
Processor: i3, i5, i7
RAM: 1GB RAM
Memory: 10 GB

Functional Requirements

User management:

- User just go through with the website and click on the project button which provided in the Web page for output then the page redirect to console of replit.
- If user wants to learn more project, then just click on the Learn more button, then the page will redirect to documentation.

Implementation:

- The project is implemented in web browser. The Windows OS is used as the platform

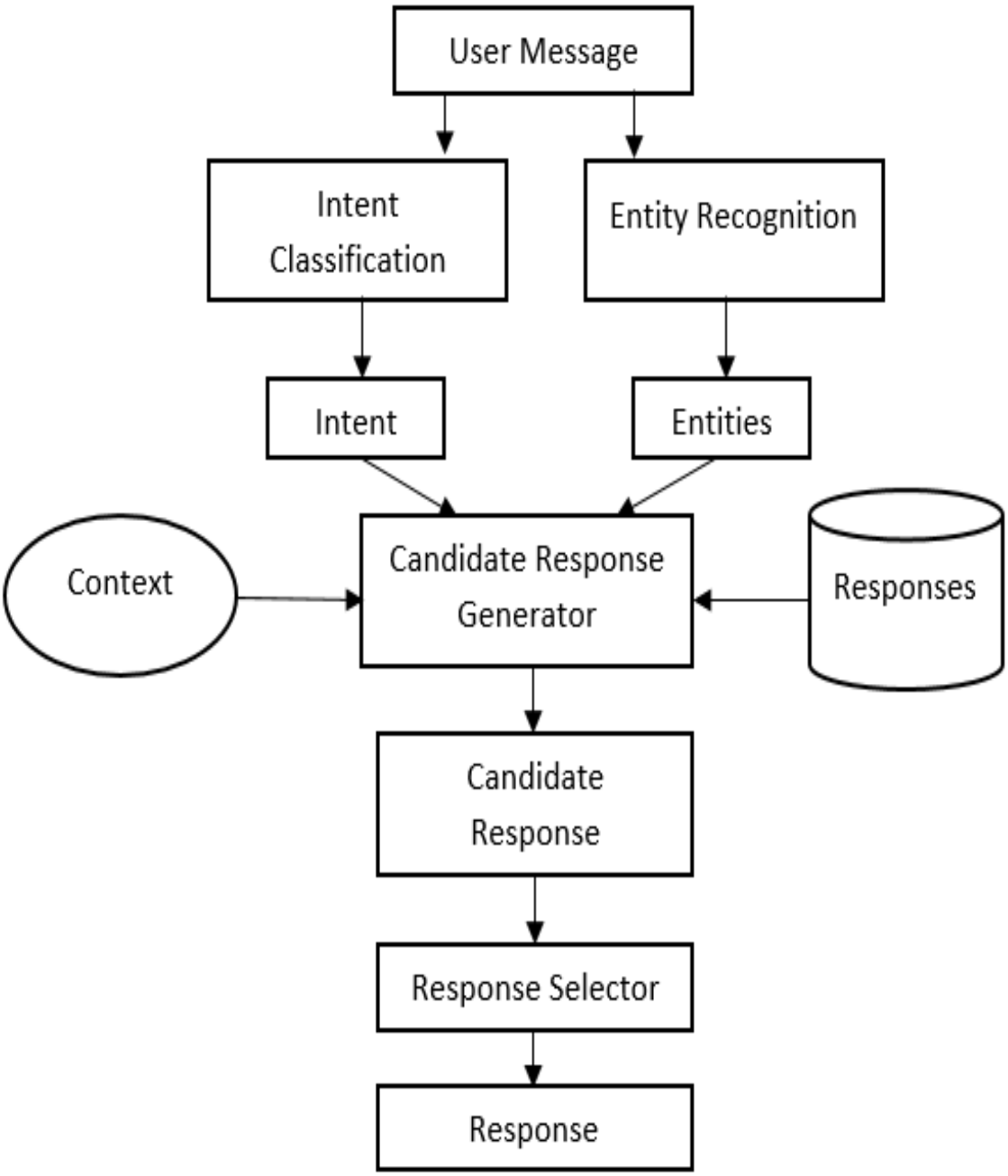
Interface:

- The user interface is based on the web browser. The page is developed by using HTML, CSS
- The interface designed at a flexible front-end communication to provide the user with clear information in navigating a user-friendly interface is planned.

4. METHODOLOGY

1.Chatbot Architecture:

An architecture of Chatbot requires a candidate response generator and response selector to give the response to the user's queries through text, images, and voice. The architecture of the Chatbot is shown in the below figure.



In the above figure, user messages are given to an intent classification and entity recognition.

A. **Intent:** An intent in the above figure is defined as a user's intention, example the intent of the word "Good Bye" is to end the conversation similarly, the intent of the word "What are some good Chinese restaurants" the intent would be to find a restaurant.

B. **Entity:** An entity in the Chatbot is used to modifies an intent and there are three types of entities they are system entity, developer entity and session entity.
C. **Candidate Response Generator:** The candidate response generator in the Chatbot do the calculations using different algorithms to process the user request. Then the result of these calculations is the candidate's response.
D. **Response Selector:** The response selector in the Chatbot used to select the word or text according to the user queries to give a response to the users which should work better.

2.Shorter Response

In today's fast-paced world, with attention spans growing shorter every second, no one has the time to read out long conversations. Jutting in complicated languages and lengthy dialogues will make the chatbot seem tedious. Though your bot is capable of handling long messages and sending responses to the user, we need a mechanism to ensure that the bot is interactive and capable of responding to diverse and yet common queries that the user might have.

The Approach

When it comes to general usage, a bot should reply to the query in advance during interactions involving single-line or two-line messages. Be imaginative. Keep it basic; the bot must be clear about the next step. To achieve this, we need to train the bot to reply quicker for frequently asked messages.

3.Bot Humanization

There is a fine line between a decent bot and an incredible bot, and the latter is possible only if you give your bot a genuine identity (named as human).

It is not merely enough to pack a sequence of answers and algorithms with a human touch. Never neglect to humanize your bot, as it can leave your potential customer with mixed feelings. Users prefer to have a human conversation, irrespective of the knowledge that they are chatting with a chatbot.

The Approach

You can give a human personality to your bot with a cool title. Discover a particular and personalized name for your bot so that your users can find it easily. Additionally, educate your bot about its representation. Ensure your bot has specific information about its personification, especially when users attempt to get some info about your bot name, age, or its central goal. Ensure to keep your chatbot holistic in approach.

4. Design the conversation

Chatbot conversations are designed to attract customers. But when this is not executed correctly, it can be taxing on the customer experience. Most chatbots redirect to the Live agent quickly, wherever there are in-depth queries and conversations required. This can help retain the customers' interest.

The Approach

Conversational chatbots are now enabling you to comprehend your customer's demands better and collect more significant information to make the interaction between your bot & customer more open and easier. We need to monitor the use-cases in clients' current activities and save communication flows.

5. Poor escalation protocol

Undoubtedly, a chatbot can communicate with the clients and help them to settle on an active choice by demonstrating them the data from the database and contingent upon the information given by clients.

The bots can likewise get to the client's expressed objective, keeping in mind that the end goal is to allow adequate search and results. They can gather and present pre-chosen reviews to clients amid an interaction, given the keyword found in the data.

However, if any procedure is extremely customized and profitable, for example, marketing, then it may be helpful also to keep a human in the string. It is vital to comprehend how and where are you going to utilize your chatbot for effective business practices.

The Approach

Chatbots can be utilized for various reasons. But one fundamental reason that organizations adapt to chatbots is the reliable customer service and customer engagement. Chatbots enhance user experiences and increase the value of the organization. At the point that a customer has a positive involvement with an organization, they will, in the end, turn into a trustworthy customer.

Keep in mind that customer experiences remain the critical factor that will add to the success or failure of your chatbot procedure. As chatbots lead towards the standard, many accepted methods for UX and development will rise. Probably a few of the above precautions will be valuable to you in terms of recognizing the issues that divert from your bots UX.

6.SOURCE CODE

HTML code:

```
<!DOCTYPE html>
<html>
<head>
	<title>MINI PROJECTS</title>
	<link rel="stylesheet" type="text/css" href="style.css">
	<link rel="icon" href="icon.png">

</head>
<body>
	<header>
		<div class="main">
			<div class="logo">

			</div>
			<ul>

			</ul>
		</div>


		<div class="button">
			<a target="_blank" href="https://replit.com/@Lokeshprasanth/ChatBot?v=1" class="btn">CHATBOT PROJECT</a>
			<a target="_blank" href="" class="btn">VIEW PROJECT 2</a>
		</div>
	</header>
<div class="title">
```

```
            <h1><small>UNIVERSITY COLLEGE OF ENGINEERING</small></h1>
            <p>(KAKATIYA UNIVERSITY)</p>
             </div>
</body>
</html>
```

CSS code:

```
*{
        margin: 0;
        padding: 0;
        font-family: Century Gothic;
}
header{
        background-image:linear-gradient(rgba(0,0,0,0.1),rgba(0,0,0,0.1)), url(2.jpg);
        height: 100vh;
        background-size: cover;
        background-position: center;
}
ul
{
        float: right;
        list-style-type: none;
        margin: 20px;
}
ul li{
        display: inline-block;
}
ul li a{
        text-decoration: none;
```

```
        color: #ffff;
        padding: 5px 20px;
        border: 1px solid #FFF;
        transition: 0.5s ease;
}
ul li a:hover{
        background-color: #fff;
        color: #000;
}
ul li.active a{
        background-color: #fff;
        color: #000;
}
.main{
        max-width: 1200px;
        margin: auto;
}
.logo img{
    left: 100%;
        width: 110px;
        height: auto;
}
.title{
        position: absolute;
        top: 52%;
        left: 50%;
        transform: translate(-50%,-50%);
}
.title h1{
    top: auto;
        color: #fff;
        font-size: 50px;
    text-align: center;
```

```
    font-family: sans-serif;
    text-shadow: 0 0 4px #000, 0 0 6px #000;
}
.title p{
        color: #fff;
        font-size: 20px;
    text-align: center;
    text-shadow: 0 0 3px #000, 0 0 5px #000;
}
.button{
        position: absolute;
        top: 66%;
        left: 50%;
        transform: translate(-50%,-50%);

}
.btn{
        border: 1px solid #fff;
        padding: 10px 30px;
        color: #fff;
        text-decoration: none;
        transition: 0.6s ease;
}
.btn:hover{
        background-color: #000;
        color: #fff;
}
```

Python code:

Main.py

```
import re
import long_responses as long


def message_probability(user_message, recognised_words, single_response=False, required_words=[]):
    message_certainty = 0
    has_required_words = True

    # Counts how many words are present in each predefined message
    for word in user_message:
        if word in recognised_words:
            message_certainty += 1

    # Calculates the percent of recognised words in a user message
    percentage = float(message_certainty) / float(len(recognised_words))

    # Checks that the required words are in the string
    for word in required_words:
        if word not in user_message:
            has_required_words = False
            break

    # Must either have the required words, or be a single response
    if has_required_words or single_response:
        return int(percentage * 100)
```

```
    else:
        return 0


def check_all_messages(message):
    highest_prob_list = {}

    # Simplifies response creation / adds it to the dict
    def response(bot_response, list_of_words, single_response=False, required_words=[]):
        nonlocal highest_prob_list
        highest_prob_list[bot_response] = message_probability(message, list_of_words, single_response, required_words)

    # Responses -------------------------------------------------------------------------------------
    response('Hello!', ['hello', 'hi', 'hey', 'sup', 'heyo'], single_response=True)
    response('See you!', ['bye', 'goodbye'], single_response=True)
    response('I\'m doing fine, and you?', ['how', 'are', 'you', 'doing'], required_words=['how'])
    response('You\'re welcome!', ['thank', 'thanks'], single_response=True)
    response('Thank you!', ['i', 'love', 'code', 'palace'], required_words=['code', 'palace'])

    # Longer responses
    response(long.R_ADVICE, ['give', 'advice'], required_words=['advice'])
    response(long.R_EATING, ['what', 'you', 'eat'], required_words=['you', 'eat'])
```

```
    best_match = max(highest_prob_list, key=highest_prob_list.get)
    # print(highest_prob_list)
    # print(f'Best match = {best_match} | Score: {highest_prob_list[best_match]}')

    return long.unknown() if highest_prob_list[best_match] < 1 else best_match


# Used to get the response
def get_response(user_input):
    split_message = re.split(r'\s+|[,;?!.-]\s*', user_input.lower())
    response = check_all_messages(split_message)
    return response


# Testing the response system
while True:
    print('Bot: ' + get_response(input('You: ')))
```

Long_responses.py

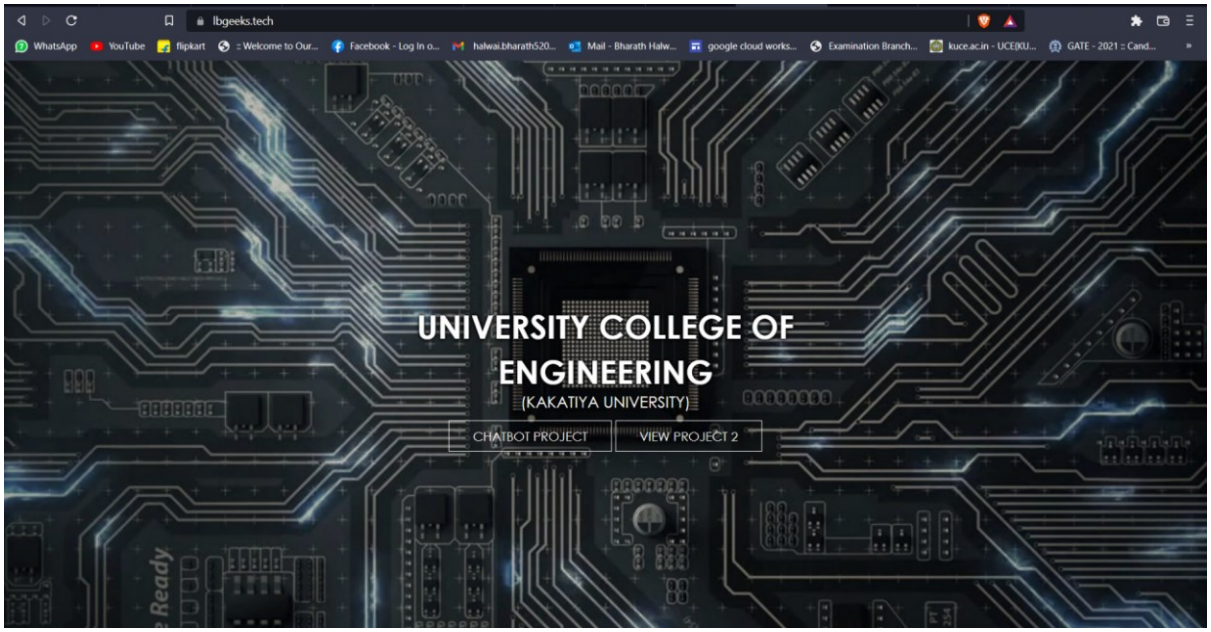
```
import random

R_EATING = "I don't like eating anything because I'm a bot obviously!"
R_ADVICE = "If I were you, I would go to the internet and type exactly what you wrote there!"


def unknown():
    response = ["Could you please re-phrase that? ",
                "...",
                "Sounds about right.",
                "What does that mean?",
                "I dont understand"][
        random.randrange(5)]
    return response
```

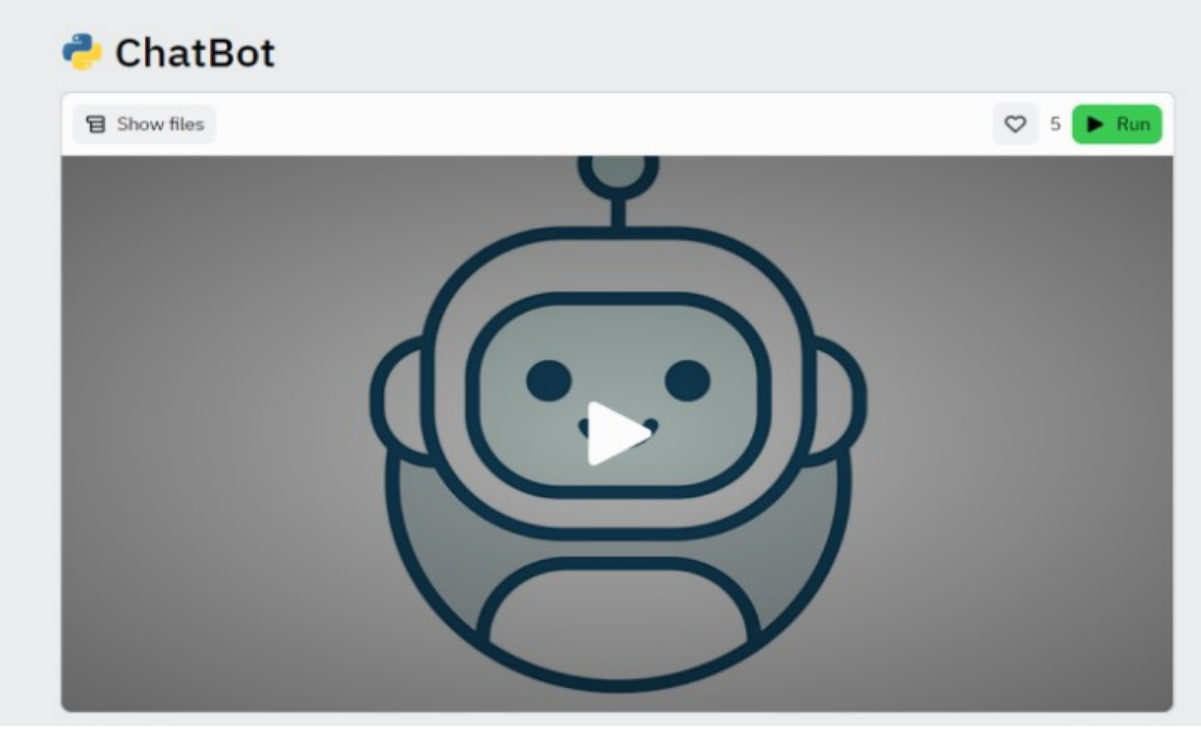
7.OUTPUT SCREENS

Screen-1:

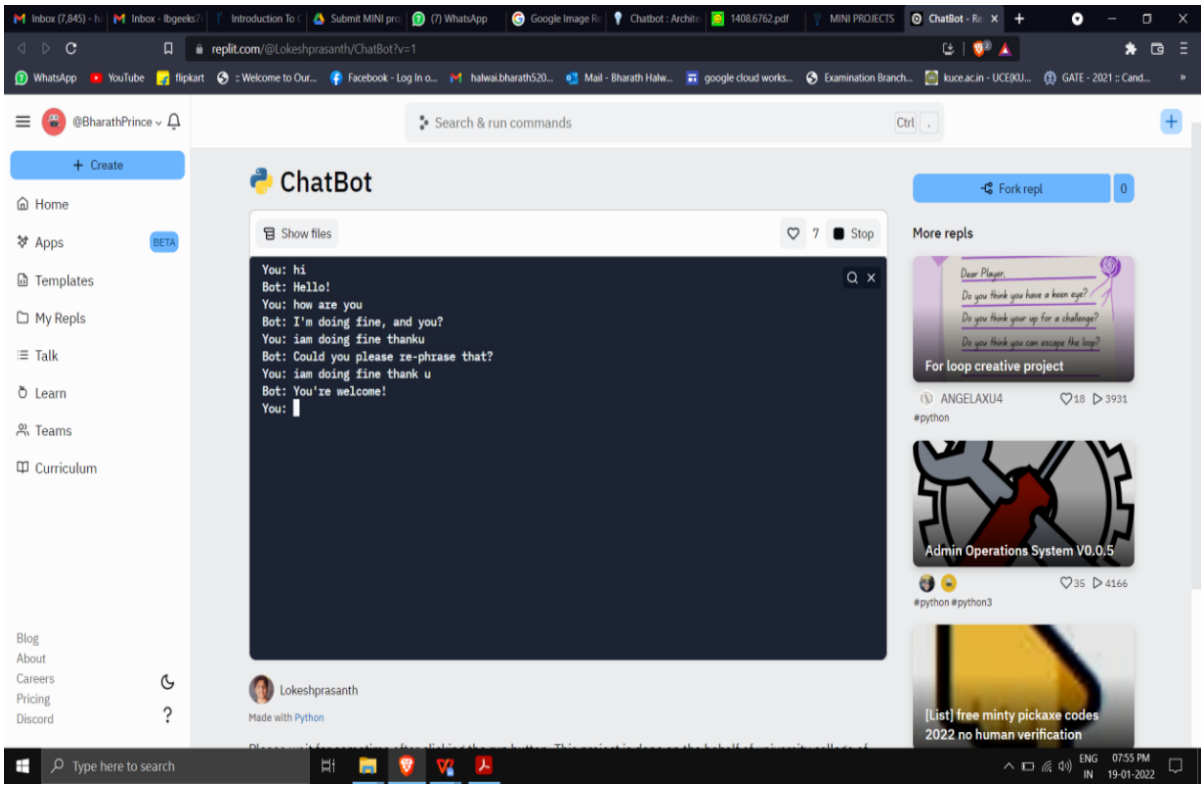


Screen-2:

Screen-3:



Screen-4:



Chatbot Testing

When it comes to testing, we need to clearly understand that the flows are implemented properly in the Chatbot application. In short, we need to derive the use-cases that help users to understand the flows easily rather than referring to the Requirements Document.

As a tester, we will be looking out the functionality bugs at the initial releases, then concentrating more on Usability & UI issues. Testing is not about finding and reporting bugs; we serve as the gatekeepers before the product releases. Hence, we need to shape the application according to the normal user perspective in terms of Design & Usability.

Furthermore, live users expect the chatbot application to be responsive. So, we should perform compatibility testing across all browsers and devices.
Once this is done, you need to perform "user testing" with a closed user group – perhaps, your company staff, client staff, or a subset of real users – to identify as many unexpected inputs as possible. Test automation may help with functional testing, but at the moment, there are no shortcuts for testing conversational logic against real humans.

-

Click here to view the live Demo

Chatbot Challenges

Some of the challenges are

- Security
- Understanding user sentiments and emotions in case of voice bots
- Language specialization
- Nonstandard languages

Advantages:

The advantages are

✧ Less cost
✧ 24/7 Availability
✧ Learning and updating
✧ It manages multiple clients
✧ It is easy to use
✧ Human effort is less

Disadvantages:

Some of the disadvantages are

✧ It takes more time for installing the app
✧ Complex interface

APPLICATIONS

The applications of Chatterbot are shown in the below

- Chatbot's for entertainment: Jokebot, Quotebot, Dinner ideas bot, Ruuh, Zo, Genius, etc
- Chatbot's for health: Webot, Meditatebot, Health tap, etc
- Chatbot's for news and weather: CNN, Poncho, etc

The Chabot improves customer services, because of this improvement the benefits of the Chatbot are increasing day by day. In today's world messaging has become one of the popular means of communication, whether it is a text message or through messaging apps. The Chabot's are used in different fields for different purposes, because of these different types of businesses are being developed Chabot's. The Chatbots are designed and developed based on customer queries and their languages.

9.CONCLUSION

In this project, we have introduced a chatbot that is able to interact with users. This chatbot can answer queries in the textual user input. For this purpose, AIML with program-o has been used. The chatbot can answer only those questions which he has the answer in its AIML dataset. So, to increase the knowledge of the chatbot, we can add the APIs of Wikipedia, Weather Forecasting Department, Sports, News, Government and a lot more.

In such cases, the user will be able to talk and interact with the chatbot in any kindof domain. Using APIs like Weather, Sports, News and Government Services, the chatbot will be able to answer the questions outside of its dataset and which are currently happening in the real world.

The next step towards building chatbots involves helping people to facilitate their work and interact with computers using natural language or using their set of rules. Future Such chatbots, backed by machine-learning technology, will be able to remember past conversations and learn from them to answer new ones. The challenge would be conversing with the various multiple bot users and multiple users.

10.REFERENCES

- ✓ http://en.wikipedia.org/wiki/Chatterbot

- ✓ Kumar, M Naveen, PC Linga Chandar, A Venkatesh Prasad, and K Sumangali (2016). "Android based educational Chatbot for visually impaired people". In: International Conference on Computational Intelligence and Computing Research (ICCIC), 2016 IEEE. IEEE, pp. 1–4.
- ✓ https://arxiv.org/ftp/arxiv/papers/1408/1408.6762.pdf

- ✓ https://www.irjet.net/archives/V6/i11/IRJET-V6I1174.pdf

- ✓ https://www.impigertech.com/resources/blogs/chatbots-and-how-to-test-them