### AI BASED CHATBOT ASSISTANT

**19ITPN6601 - INNOVATIVE AND CREATIVE PROJECT**

Submitted by

**THILAK RAJ G(20BIT026)**

**SANGEETH A (20BIT064)**

**ARUNGARAI SELVAN (20BIT072)**

***Inpartial fulfillment for the award of the degree of***

### Bachelor of Technology in InformationTechnology

### Dr. Mahalingam College of Engineering and Technology

### Pollachi-642003

**An Autonomous Institution**

### Affiliated to Anna University, Chennai - 600 025

### MAY 2023

**Dr. Mahalingam College of Engineering and Technology Pollachi -642003**

### An Autonomous Institution

**Affiliated to Anna University,Chennai-600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report, “**AI BASED CHATBOT ASSISTANT**”

is the bonafide work of

**THILAK RAJ G(20BIT026)**

**SANGEETH A (20BIT064)**

**ARUNGARAI SELVAN (20BIT072)**

Who carried out the project work under my supervision.

-------

**Mr. R. Govindaraj.,**

Assistant Professor

Information Technology

Dr. Mahalingam College of Engineering

and Technology,

NPT-MCET Campus Pollachi – 642003

India

**Dr. A.S. Muthanantha Murugavel**

Associate Professor &HoD i/c

InformationTechnology

Dr. Mahalingam College of Engineering

and Technology,

NPT-MCET Campus Pollachi – 642003

India

Submitted for the Autonomous End Semester Examination Project Viva-voce held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# ABSTRACT

**AI BASED CHATBOT ASSISTANT**

**ABSTRACT**

As a result of the rapid technology advancement and development in the fields of Artificial Intelligence and Machine Learning, the concepts of chatbot and other related are the lifesavers. The chatbot has integrated machine learning into it by depending on artificial intelligence, and it has gotten more complete and wider for numerous technical domains. This project is all about creating a chatbot based on the AI and ML technologies to generate responses requested as per the client or user. As a result, we will develop a chatbot for general use in relation to any industry information. The goal of this project is to create a chatbot that will make it easier to get any information. The project's objective is to include an API and chatbot feature. Yioop includes all of the essential functions of an online search portal. It contains a built-in account management system that enables the creation of groups with message boards. Users who belong to a group get access to the group feed. The first group owner is the user who creates the group. By thread, posts are arranged in groups with the most current activity at the top. By using the chatbot API provided by Yioop, programmers can build new chatbots with artificial intelligence or rules that can converse naturally with people on a group's feed page.

Weather chatbots and flight booking chatbots are two examples of chatbots that may be created using this API. Over the past few years, messaging apps have surpassed social networking sites in popularity. These days, people use messaging apps like Facebook Messenger, Skype, Viber, Telegram, and Slack, among others. Making other companies accessible via messaging services encourages proactive user conversation about their goods. Businesses can create a chatbot—a computer programme that can communicate like a human—to interact with a large number of people on such messaging services.

Two categories of chatbots exist:

• Simple set of regulations

• Computer learning

Chatbot with a small set of rules

These bots are severely constrained to a small number of words or commands. Only those texts or commands can cause them to react. The bot will not answer as planned if the user asks a question that is different from the predefined set of texts or commands because it does not understand or has not been trained to understand such questions. When compared to other types of bots, these ones are not extremely intelligent.

# ACKNOWLEDGEMENT

### ACKNOWLEDGEMENT

Apart from the efforts of us, the success of this project depends largely on the encouragement and guidelines of many others. We take this opportunity to praise the **almighty** and express our gratitude to the **people** who have been instrumental in the successful completion of our project.

We wish to acknowledge with thanks for the excellent encouragement given by the management of our college and we thank **Dr. C. Ramaswamy, M.E., Ph.D., FIV, Secretary, NIA Educational Institutions** for providing us with a plethora of facilities in the campus to complete our project successfully.

We wish to express our hearty thanks to **Dr. P. Govindasamy, M.E., Ph.D.,Principal**

of our college**,** for his constant motivation regarding our project work.

We heartily express our extreme gratefulness to **Dr. S. Ramakrishnan, M.E., Ph.D., Professor – IT & Dean–RI** for his instrumental in providing such wonderful support to us.

We extend my heartfelt gratitude to **Dr. A.S. Muthanantha Murugavel, M.E., Ph.D., Associate Professor & HoD i/c – IT** for his tremendous support and assistance in the completion of my project.

It is our primary duty to thank our **Project Coordinator, Mr. R. Govindaraj., /Assistant Professor, Information Technology** who is the backbone of all our project activities, for his consistent guidance and encouragement, which kept us fast and pro-active in our work. It’s her enthusiasm and patience that guided us through the right path.

We would like to show our greatest appreciation to **Mr. R. Govindaraj., /Assistant Professor.** We feel motivated and encouraged every time we attend his meeting. And the guidance from his broadens our minds to do the project within there stand enhanced knowledge to know more.

Finally, we extend our heartfelt thanks to the enriched motivation and encouragement of our **parents, friends, and faculty members.** The facilities received from our institutions made our work easier. We are grateful to each and every one who is constantly helped and supported us to complete the project enthusiastically and successfully.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# TABLE OF CONTENTS

### TABLE OF CONTENTS

#### ChapterNo.

**Title PageNo.**

Abstract i

Acknowledgement ii

List of Figures iii

List of Abbreviations iv

1. [INTRODUCTION 1](#_TOC_250013)
   1. [1 Problem Definition 1](#_TOC_250012)

[1. 2 Project Overview/ Specifications 1](#_TOC_250011)

[1. 3 Hardware Specification 2](#_TOC_250010)

1. 4 Software Specification 2

1. EXISTING SYSTEM 3
   1. Existing System 3
   2. [Limitations in existing system 3](#_TOC_250009)
2. [PROPOSED SYSTEM 4](#_TOC_250008)
   1. Proposed methodology 4
   2. [Process flow diagram 7](#_TOC_250007)
3. [RESULTS 8](#_TOC_250006)

[4.1 Application areas 8](#_TOC_250005)

1. [CONCLUSIONS 9](#_TOC_250004)

[5.1 Future Works 9](#_TOC_250003)

1. REFERENCES 10

[6.1 Web References 10](#_TOC_250002)

1. [APPENDICES 11](#_TOC_250001)
   1. [Appendix-1 11](#_TOC_250000)
   2. Appendix-2 13

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **Figure No.** | **Title** | **Page No.** |
| Fig1 | Data Processing | 7 |
| Fig2  Fig3  Fig4  Fig5 | Home Page  Services  Response generation  Result | 11  11  12  12 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# LIST OF ABBREVIATIONS

### LIST OF ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| |  | | --- | | AI | | Artificial Intelligence |
| **ML** | Machine Learning |
| **NLP** | Natural Language Processing |
| **NLU** | Natural Language Understanding |
| **ASST.** | Assistant and Assistance |
| **GUI** | Graphical User Interface |
| **RF** | Radio Frequency |
| **GPS** | Global Positioning System |
| **DFD** | Data Flow Diagram |
| **UML** | Unified Modelling language |
| **SQL** | Structure Query Language |
| **DML** | Data Manipulation Language |
| **J2EE** | Java 2 Platform Enterprise Edition |
| J2SDK D | Java 2 SDKD |
| **CGI** | Common Gateway Interface |
| **HTTP** | Hypertext Transfer protocol |

# CHAPTER 1

# INTRODUCTION

## 1.INTRODUCTION

A chatbot is a computer programme that allows people to communicate with it using natural language and artificial intelligence methods like NLP (natural language processing), which makes the chatbot more responsive and trustworthy. The study of teaching computers to understand human languages is known as natural language processing (NLP). Without NLP, computer words are nothing more than a collection of meaningless symbols. The words and the grammar are not recognized by computers. NLP can be thought of as a "translator," converting human languages into information that computers can understand.

Computer science's field of machine learning (ML) "gives computers the ability to learn without being explicitly programmed." Instead of being predetermined by the programmer, the parameter of the formulas is calculated from the data. Regression and Classification are the two most popular uses of ML.

* 1. **PROBLEM DEFINITION**

Adaptation to new needs as part of an industrial endeavor to create a product, we must adhere to user requirements. The project's technical team is in charge of it, but the business team will use it, therefore the demand changed significantly in the middle of development after a meeting with the business team. The business team is looking for a straightforward bot that can make recommendations right away. We had to put our previous work in the past and start over.

#### PROJECT OVERVIEW AND SPECIFICATIONS

#### Machine learning begins with data, such as financial transactions, individuals, or photos. The information is collected and processed to be utilized as training data for the machine learning system. If the data is more than the software shows better results. After that, the developer selects a ML model to use, input the data, and train the system to find patterns or make predictions on its own. This system would help clients to make educated decisions about various aspects of domains. They employed decision trees, KNNs, Random Forests, and neural networks. The neural network had the highest accuracy of all of them.

**1.3. HARDWARE SPECIFICATION**

**Disk space :** 330MB for windows package extraction

**Graphics :** No specific graphics card be required. Hardware accelerated graphics

Card supporting OpenGL3.3 with 1 Gigabytes GPU memory

**RAM :** 2 Gigabytes with Simulink, 4 Gigabytes is required with Polyspace 4 Gigabytes

per core is recommended.

#### 1.4. SOFTWARE SPECIFICATION

**Tool :** Jupyter Notebook, VSCode

**Operating System**

**Language**

**:** Windows or Linux

**:**  Python, AIML

# CHAPTER 2

# EXISTING SYSTEM

## 2. EXISTING SYSTEM

#### 2.1. DESCRIPTION

#### Adapting to new requirements We must follow user needs as part of an industrial endeavour to build a product. The technical team of the project is in charge of it, but the business team will use it; as a result, during development, the need changed considerably following a meeting with the business team. The business team is searching for a simple bot that can offer suggestions immediately. We had to start afresh, putting our prior work in the past.

#### 2.2. LIMITATIONS IN EXISTING SYSTEM

* **LACK OF TRAINING DATA:** The performance of a machine learning model depends on the amount and quality of training data.
* We create some fake data for the machine learning model based on our everyday experience, which is extremely biased but accurate on the fake data.
* **UNSTABLE API VERSIONS:**  Since the API services we utilize are still being developed, we are unable to fix to a specific version of the API, hence the API may change over time.
* Additionally, there are discrepancies between the APIs' documentation and sample programs.
* **NOT FAMILIAR WITH THE PHP LANGUAGE AND .NET FRAMEWORK :**  None of the three of us had any prior experience with PHP.
* At the beginning of the project, programming in a new language in such a large framework is rather difficult for us. However, we are more accustomed to that by the later stages.

# CHAPTER 3

# PROPOSED SYSTEM

## PROPOSED SYSTEM

#### 3.1 PROPOSED METHODOLOGY:

#### The proposed approach will allow users to give input for the system and generate responses based on the requested information. Processes that have carried out in this project are as follows,

#### AIML SCRIPTING: We built an AIML file that solely manages the load aiml b pattern. When we give the bot that command, it will attempt to load basic\_chat.aiml. If we don't make it, it won't work. What you can include in basic chat.aiml is listed below. We'll compare two fundamental patterns and answers.

#### CREATING A STARTUP FILE: As the primary entry point for loading AIML files, a startup file called std-startup.xml is typically created. In this instance, we'll make a simple file that matches a single pattern and executes a single action. In order to have our aiml brain load in response, we want to match the pattern load aiml b.The basic\_chat.aiml file will be created in a moment..

#### Feature Engineering: Using domain expertise, feature engineering extracts features (characteristics, traits, and attributes) from raw data. The idea is to use these extra attributes to improve the quality of ML results.

#### Training set: A training set is a data set that contains data that has been labelled. Both input and output vectors are included. The model is trained using supervised machine learning algorithms using this dataset.

#### Testing set: A testing set is a data set that is devoid of labelled data. It predicts the outcome with the assistance of the training data set. It is unaffected by the training data set.

#### Machine Learning Algorithm: Machine learning prediction algorithms necessitate extremely accurate estimation based on previously learned data. Predictive analytics historical information is the use of data, statistical methods,and machine learning approaches to forecast future results. The goal is to go beyond simply understanding what happened to providing the best feasible remedy and a prediction of what will happen next.

#### A. K-Nearest Neighbor Classifier KNN is a type of supervised machine learning that can be used to solve a wide range of issues. Classification and regression are examples of challenges that can be addressed. The number of nearest neighbours to a newly forecasted unknown variable is represented by the symbol K. The distance between the data points is calculated using the Euclidean distance formula.

#### Euclidean Distance b/w A and B = √ (X2 -X1)2 + (Y2-Y1)2. (1)

#### B. Decision Tree Decision Trees (DTs) comes under supervised learning for classification and regression. A tree representation is used to solve the problem, with each leaf node representing a class label. and the tree's interior node represents attributes.

#### Entropy: H(S) = -∑ Pi(S) log2 Pi(S) (2)

#### Information Gain: IG(S,A) = H(S) - ∑v € Values(A)( | Sv |/S) H(Sv) (3)

#### C. Naive Bayes The Bayes theorem is used to create a simple probablistic classifier called Naive Bayes. Naive Bayes classifiers assume that the value of one feature is independent of the value of any other feature given the class variable.

#### P(A|B) = (P(B|A) \* P(A) )/P(B) (4)

#### D. Random Forest Random Forest is an ensemble learning method that creates a huge number of distinct models to solve classification, regression, and other issues. At training time, decision trees are used. Algorithm of random forest makes decision trees based on various data samples and then forecasts data from each subset, then votes on it gives the system a better solution RF uses the bagging method for data training, which improves the correctness of the outcome. ∞

#### Gini Index =1-∑ (Pi)2 i=1 = 1-[(P+) 2+ (P-) 2] (5)

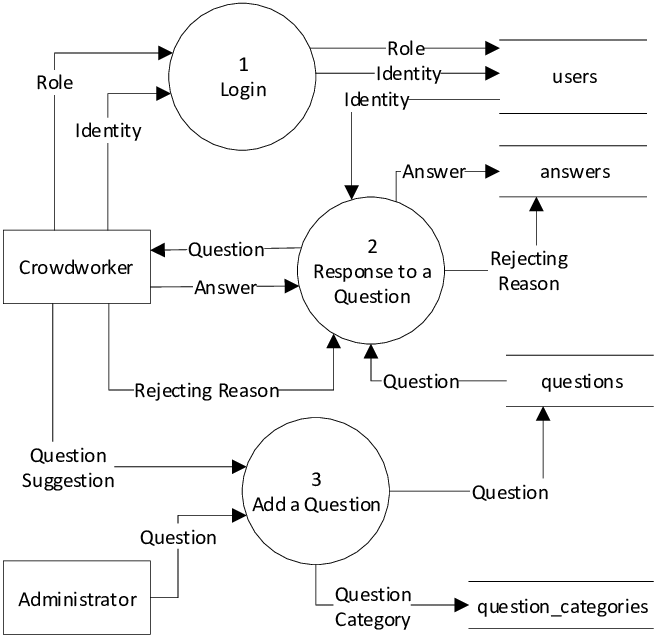
#### E. Gradient Boosting Gradient Boosting is also comes under supervised machine learning technique for solving classification and regression issues. It's a poor prediction model with an ensemble. As with previous boosting approaches, a gradient-boosted trees model is built stage by stage.

#### Performance Analysis: Performance analysis is a specialized subject that uses systemic objectives to improve performance and decision-making.

#### Debugging : Plugin calls and other information that is not accessible to the user will not by default be displayed in the conversation. You can view the request and response by clicking the down arrow on the plugin name after interacting with the plugin in order to obtain a more thorough understanding of how the model is communicating with your plugin.

#### A typical model call to the plugin will start with a message from the model that contains JSON-like arguments that are delivered to the plugin, then come a message from the model that uses the information the plugin has returned.By selecting "Settings" and then "Open plugin devtools," you can also access the developer console if you are creating a localhost plugin.

#### 3.2 PROCESS FLOW DIAGRAM:



**Fig1 Data Proccessing**

# CHAPTER 4

# RESULTS

## 4. Results and Discussions:

Findings from testing the most recent prototype point to the fact that participants had no issues receiving information from a chatbot as opposed to a human. The fact that the chatbot cited a source for the information it supplied may have contributed to their perception that the information they received was not less reliable. Investigating how the participants interacted with the chatbot and how they discussed it afterwards has been interesting. Our research suggests that a chatbot might provide a viable substitute for a kind, supportive companion for freshmen at a new school.

Due of the project's size, we were not able to test as many users or redesign the chatbot as thoroughly as we would have desired. This affects the reliability of our study. While developing the chatbot, we have touched on some theory throughout the project, but this also has to have a bigger focus for greater legitimacy. We cannot say that consumers trust a chatbot as much as they trust a human being, despite the fact that the participants in this project trusted the information provided. There are other biases in our project, one of which is that every student we involved in it already knew a significant portion of the solution the prototype could offer.

# CHAPTER 5

# CONCLUSION

## 5. CONCLUSION

The newest apps are Chabot’s! As mentioned in the aforementioned deliverables, this project enhances Yioop's usefulness by bringing Chabot’s' strength to the platform. In Yioop, Chabot’s can provide a human touch to various elements and make the discussion enjoyable. And they give information and carry out tasks for the people they engage with with full attention. All of the deliverables have the aforementioned functionality implemented and put into the Yioop code. By putting the aforementioned deliverables into practise, I was able to provide Yioop some basic Chabot capabilities. Specifically, setting up accounts for bot users with the bot parameters described in Deliverable 2 and activating a bot whenever a user requests it in a post.

I want to improve the system that has been created so far in CS298. The next stage in creating chatbots is to assist users in facilitating their work and interacting with computers through the use of natural language or a set of rules. Future Yioop chatbots will be able to recall previous interactions and draw lessons from them to respond to new ones thanks to machine learning technology. Speaking with many bot users and multiple users would be difficult.

## 5.1 FUTURE WORKS

What has been accomplished with chatbots thus far has limitations. Data processing and retrieval restrictions prevent chatbots from reaching their full potential. We don't lack the computational processing ability to do so. However, there is a constraint to "How" we do it. The retail consumer market is one of the most prominent examples. Because of the nature of their needs, retail customers are primarily interested in engaging with humans. They do not want bots to handle their requests and answer appropriately..

# CHAPTER 6

# REFERENCES

## 6. REFERENCES

[1] Bayan Abu Shawar and Eric Atwell, 2007 “Chatbots: Are they Really Useful?”

[2] LDV Forum - GLDV Journal for Computational Linguistics and Language

Technology.

[3] http://www.ldv-forum.org/2007\_Heft1/Bayan\_Abu-Shawar\_and\_Eric\_Atwell.pdf

[4] Bringing chatbots into education: Towards natural language negotiation of open

learner models. Know.-Based Syst. 20, 2 (Mar. 2007), 177-185.

[5] Intelligent Tutoring Systems: Prospects for Guided Practice and Efficient

Learning. Whitepaper for the Army's Science of Learning Workshop, Hampton,

VA. Aug 1-3, 2006.

[6] http://en.wikipedia.org/wiki/Chatterbot

[7] ALICE. 2002. *A.L.I.C.E AI* Foundation, http://www.alicebot.org/

[8] "Chatbots - Artificial Solutions." Chatbots | Conversational AI Platform for Enterprise - Teneo. Accessed 8 Nov. 2020.

[9] "How Do Bots and Chatbots Work." https://www.cxtoday.com/contact-centre/how-do-bots-and-chatbots-work/

[10] "Building Chatbots - A comparison of Rasa-NLU and Dialogflow." 17 Apr. 2018, Building Chatbots - A comparison of Rasa-NLU and Dialogflow. Accessed 8 Nov. 2020.

[11] "Jonathan Wheat" Jonathan Wheat - DEV Community. Accessed 10 Nov. 2020.

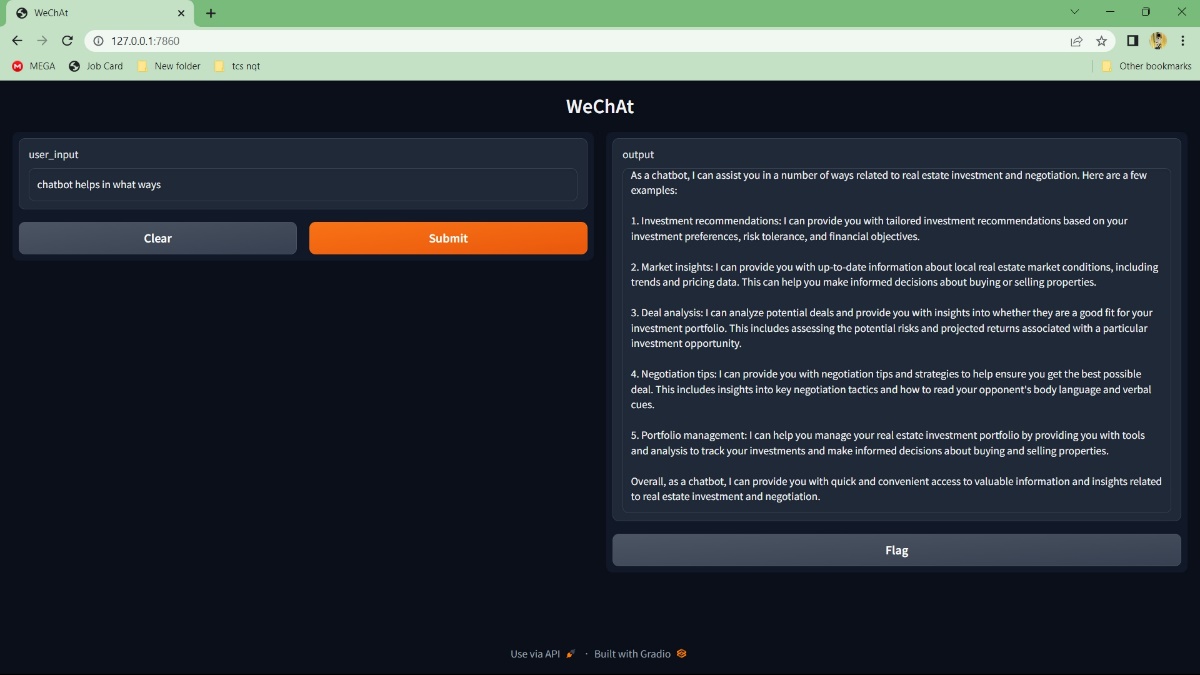
[12] Understanding the Basics of Rasa - Open source conversational AI https://smazee.com/blog/basics-of-rasa

[13] Rasa Architecture Overview https://rasa.com/docs/rasa/arch-overview/

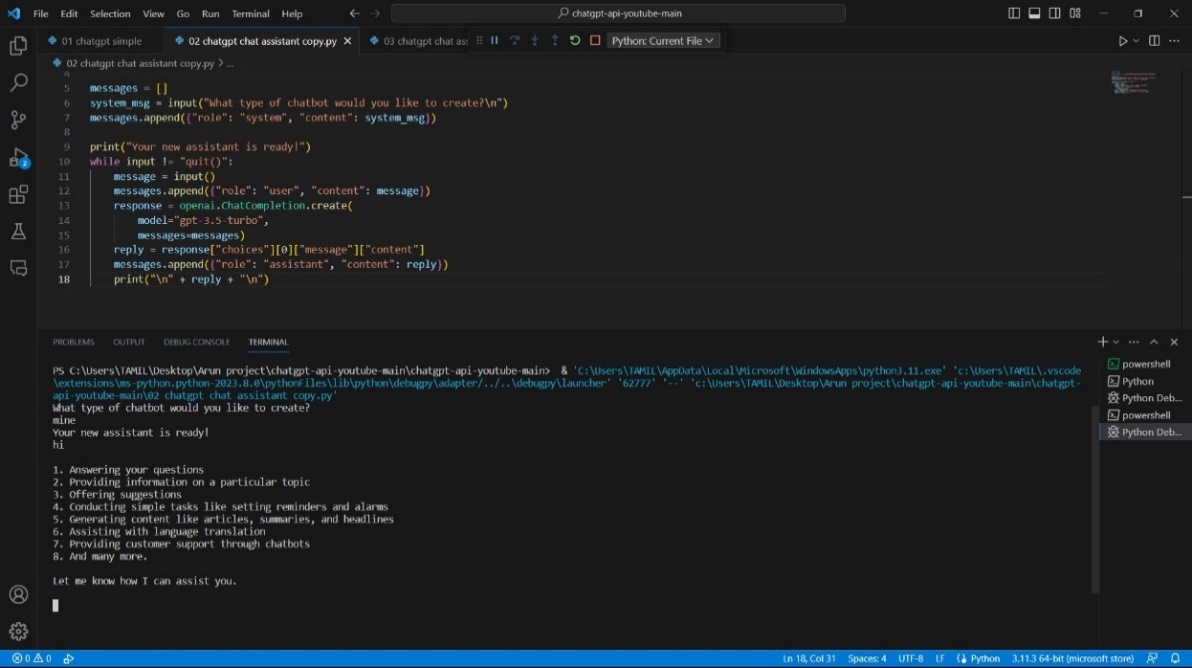
## 7. APPENDICES

### 7.1. APPENDIX-1

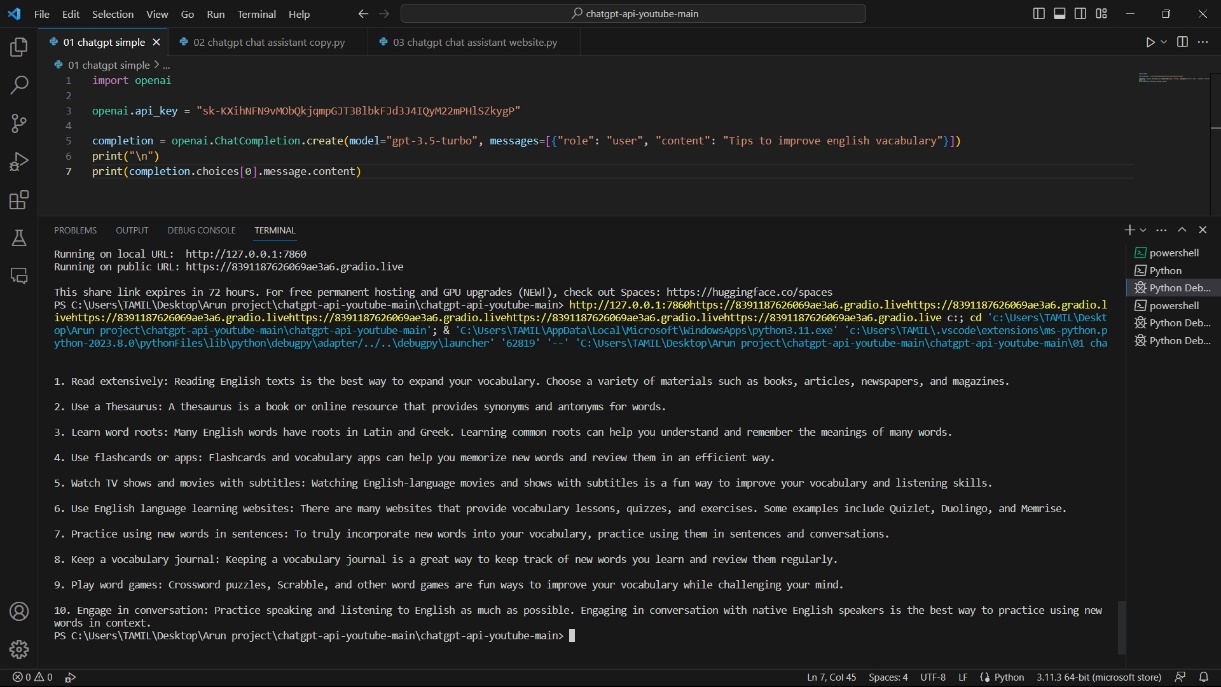
#### SCREENSHOTS



**Fig1 Chatbot interface**



**Fig2 command line panel**



**Fig 3 simple interface**

### 7.2. APPENDIX-2

#### CODING

#### Main.html

#### {% load static %}

#### <!DOCTYPE html>

#### <html lang="en">

#### <head>

#### <meta charset="UTF-8">

#### <meta http-equiv="X-UA-Compatible" content="IE=edge">

#### <meta name="viewport" content="width=device-width, initial-scale=1.0">

#### <title>Home Page CRS</title>

#### <link rel="stylesheet" href="{% static 'bootstrap/bootstrap.min.css' %}">

#### <script src="{% static 'bootstrap/bootstrap.min.js' %}"></script>

#### <style>

#### body {

#### line-height: 1;

#### background-image: url(../static/images/bg1.jpeg);

#### background-size: cover;

#### background-repeat: no-repeat;

#### background-position: center center;

#### background-attachment: fixed;

#### /\* adjust this value to create space for the navbar \*/

#### }

#### /\* style for nav \*/

#### nav {

#### background-color: #333;

#### overflow: overlay;

#### padding: 0;

#### height: auto;

#### }

#### nav ul {

#### list-style-type: none;

#### margin: 0;

#### padding: 0;

#### }

#### nav li {

#### float: left;

#### }

#### nav li a {

#### display: inline-block;

#### color: white;

#### text-align: right;

#### padding: 14px 16px;

#### text-decoration: none;

#### }

#### nav li a:hover {

#### background-color: #111;

#### }

#### .nav-container {

#### text-align: center;

#### background-color: #333;

#### padding: 16px;

#### }

#### .nav-container ul {

#### list-style: none;

#### padding: 0;

#### margin: 0;

#### }

#### .nav-container li {

#### display: inline-block;

#### margin-right: 20px;

#### }

#### .nav-container li:last-child {

#### margin-right: 0;

#### }

#### @media (max-width: 768px) {

#### body {

#### line-height: 1.2;

#### }

#### .nav-container {

#### text-align: right;

#### }

#### .nav-container li {

#### display: block;

#### margin: 10px 0;

#### }

#### nav-container ul {

#### display: inline-block;

#### }

#### }

#### h5 {

#### text-align: center;

#### margin-top: 30px;

#### margin-bottom: auto;

#### }

#### h1 {

#### text-align: center;

#### margin-top: 20px;

#### }

#### h3 {

#### text-align: center;

#### margin-top: 10px;

#### }

#### .container {

#### margin-top: 40px;

#### }

#### /\* service boxes \*/

#### .services-container {

#### display: flex;

#### justify-content: space-between;

#### flex-wrap: wrap;

#### max-width: 700px;

#### margin: 0 auto;

#### padding: 10px;

#### }

#### .service-box {

#### width: 45%;

#### max-width: 500px;

#### margin-bottom: 20px;

#### border: 1px solid #ccc;

#### border-radius: 20px;

#### padding: 17px;

#### text-align: center;

#### background-color: white;

#### }

#### @media (max-width: 500px) {

#### .service-box {

#### width: 200px;

#### white-space: nowrap;

#### overflow: hidden;

#### text-overflow: ellipsis;

#### width: 500px;

#### }

#### .service-box p {

#### font-size: 5px;

#### }

#### }

#### @media (max-width: 500px) {

#### .service-box {

#### width: 200px;

#### white-space: nowrap;

#### overflow: hidden;

#### text-overflow: ellipsis;

#### width: 500px;

#### }

#### .service-box p {

#### font-size: 5px;

#### line-height: 1.5;

#### }

#### }

#### .service-box img {

#### width: 100%;

#### height: auto;

#### margin-bottom: 10px;

#### }

#### .service-box h2 {

#### font-size: 24px;

#### margin-bottom: 10px;

#### }

#### .service-box p {

#### font-size: 16px;

#### line-height: 1.5;

#### }

#### /footer/

#### .footer-container {

#### background-color: #333;

#### color: white;

#### }

#### .footer-container p {

#### color: white;

#### }

#### h4 {

#### color: #007bff;

#### }

#### .col-sm-4 {

#### padding: 16px 70px;

#### margin: 0px 102px;

#### text-align: center;

#### }

#### .row {

#### place-content: center;

#### }

#### .start {

#### background-color: #f8f8f8;

#### padding: 40px;

#### border: 1px solid#ccc;

#### }

#### h3 {

#### font-size: 24px;

#### font-weight: bold;

#### color: #333;

#### margin-bottom: 0;

#### }

#### p {

#### font-size: 16px;

#### color: #666;

#### margin-bottom: 0;

#### text-align: center;

#### }

#### h5 {

#### font-size: 18px;

#### font-weight: bold;

#### color: #333;

#### margin-bottom: 0px;

#### }

#### .start {

#### overflow: hidden;

#### /\* add clearfix hack to contain floated images \*/

#### }

#### .startimg.left {

#### float: left;

#### margin-right: 20px;

#### /\* add some spacing between images \*/

#### margin-right: 30px;

#### /\* add some spacing between images \*/

#### max-width: 30%;

#### /\* set a maximum width for the image \*/

#### height: auto;

#### }

#### .startimg.right {

#### float: right;

#### margin-left: 20px;

#### /\* add some spacing between images \*/

#### margin-left: 30px;

#### /\* add some spacing between images \*/

#### max-width: 30%;

#### /\* set a maximum width for the image \*/

#### height: auto;

#### /\* set height to auto to maintain aspect ratio \*/

#### }

#### @media screen and (max-width: 600px) {

#### .startimg.left,

#### .startimg.right {

#### float: none;

#### /\* remove float on smaller screens \*/

#### max-width: 100%;

#### /\* set max width to 100% to fill container \*/

#### margin: 10px auto;

#### /\* add some spacing and center align \*/

#### }

#### }

#### h6 {

#### align-items: center;

#### }

#### </style>

#### </head>

#### <body>

#### <div class="nav-container">

#### <ul>

#### <li><a href="{% url 'home' %}">Home</a></li>

#### <li><a href="{% url 'predictor' %}">Services</a></li>

#### <li><a href="{% url 'about' %}">About</a></li>

#### <li><a href="#">Contact</a></li>

#### </ul>

#### </div>

#### </div>

#### </div>

#### </footer>

#### </body>

#### </html>

#### Extension.json

{

    "recommendations": [

        "danielsanmedium.dscodegpt"

    ]

}

**01 CHATBOT CHAT ASSISTANT COPY**

import openai

openai.api\_key = "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"

completion = openai.ChatCompletion.create(model="gpt-3.5-turbo", messages=[{"role": "user", "content": "Tips to improve english vacabulary"}])

print("\n")

print(completion.choices[0].message.content)

**02 CHATBOT CHAT ASSISTANT COPY**

import openai

openai.api\_key = "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"

messages = []

system\_msg = input("What type of chatbot would you like to create?\n")

messages.append({"role": "system", "content": system\_msg})

print("Your new assistant is ready!")

while input != "quit()":

    message = input()

    messages.append({"role": "user", "content": message})

    response = openai.ChatCompletion.create(

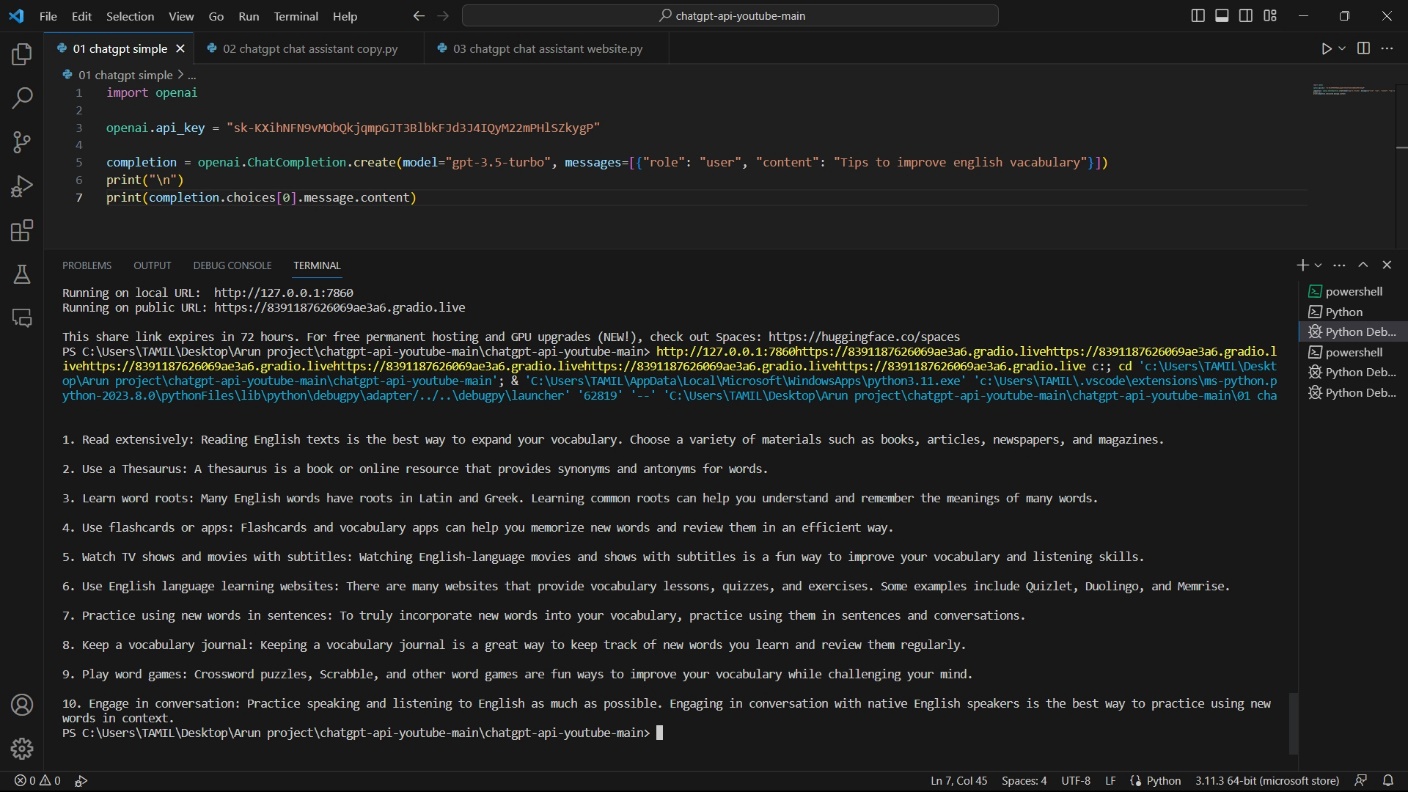
        model="gpt-3.5-turbo",

        messages=messages)

    reply = response["choices"][0]["message"]["content"]

    messages.append({"role": "assistant", "content": reply})

    print("\n" + reply + "\n")

****

**03 CHATGPT CHAT ASSISTANT WEBSITE**

import openai

import gradio

openai.api\_key = "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"

messages = [{"role": "system", "content": "You are a financial experts that specializes in real estate investment and negotiation"}]

def CustomChatGPT(user\_input):

    messages.append({"role": "user", "content": user\_input})

    response = openai.ChatCompletion.create(

        model = "gpt-3.5-turbo",

        messages = messages

    )

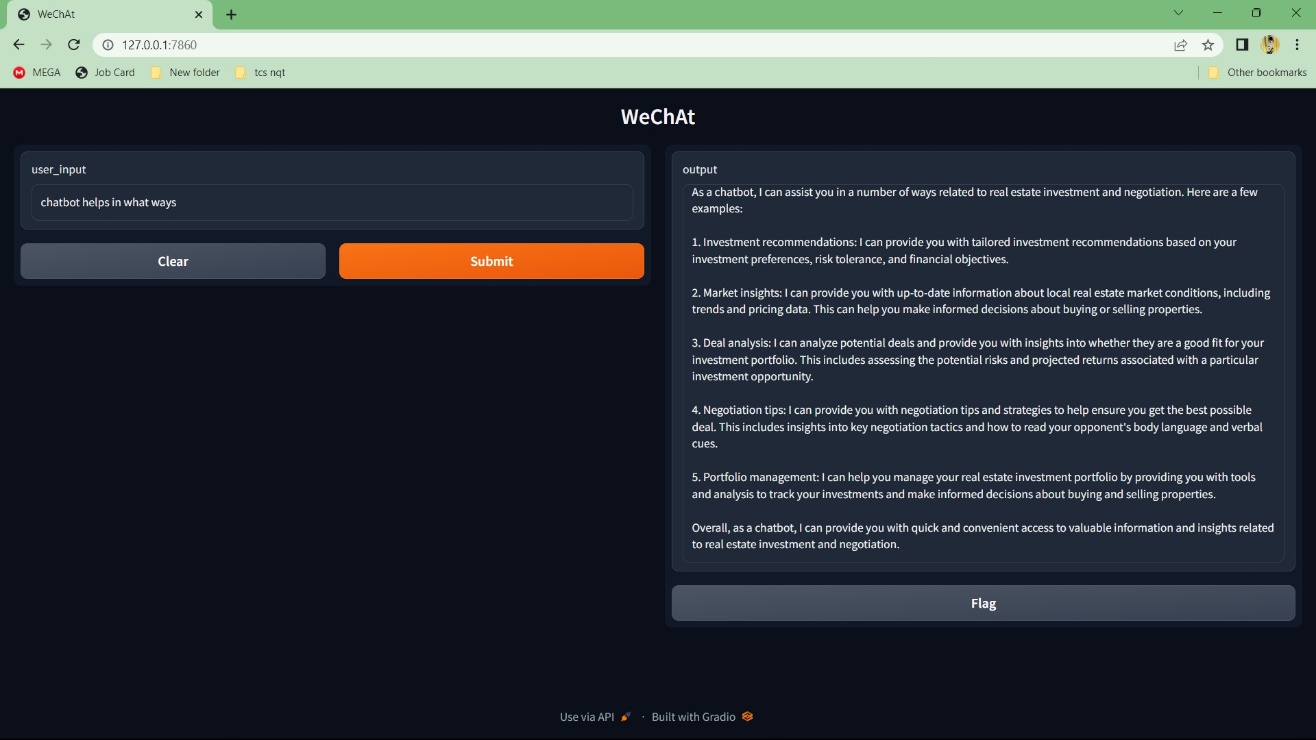
    ChatGPT\_reply = response["choices"][0]["message"]["content"]

    messages.append({"role": "assistant", "content": ChatGPT\_reply})

    return ChatGPT\_reply

demo = gradio.Interface(fn=CustomChatGPT, inputs = "text", outputs = "text", title = "WeChAt")

demo.launch(share=True)

****