CSE1901 - Technical Answers to Real World Problems (TARP)

Project Report

VITian's Chatbot

By

20BCE1308	Sri Harshavardhan Palla
20BCE1968	Rukksana A

B. Tech Computer Science and Engineering

Submitted to

Dr. V. Viswanathan

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

April 2023

DECLARATION

I hereby declare that the report titled “VITian’s chatbot” submitted by Sri Harshavardhan Palla and Rukksana A to VIT Chennai is a record of bona-fide work undertaken by us under the supervision of Dr.V.Viswanathan, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai.

Signature of the Candidate

Sri Harshavardhan Palla
20BCE1308

Rukksana A
20BCE1968

CERTIFICATE

Certified that this project report entitled "VITian's Chatbot" is a bonafide work of Sri Harshavardhan Palla (20BCE1308) , Rukksana A (20BCE1968) and they carried out the Project work under my supervision and guidance for CSE1901 - Technical Answers to Real World Problems (TARP).

Dr V.Viswanathan

SCOPE, VIT Chennai

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr. V. Viswanathan, Professor, School of Computer Science and Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course. We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

SRI HARSHAVARDHAN PALLA RUKKSANA A

ABSTRACT

A chatbot is an artificial intelligence software that can simulate a conversation with a user in natural language through messaging applications, websites, mobile apps, or the telephone.Today almost every company has a chatbot deployed to engage with the users, mainly used for customer support . We created a multi-threaded web UI for chatbot using flask framework. Inputs are taken from the user through HTML form and after processing, chatbot gives the vital response. According to the user input , the server opens the required website automatically. Bot is designed where existing educational advising services have been uploaded and set up an online natural language interactive interface that will support a conversation and provide links which will direct users to the solutions . Multiple users can operate at the same time by logging in . Supporting the user to access the college websites according to their need reduces time consumption and difficulty level . This CHATBOT can be very well utilised by the users to guide them in any process.

CONTENTS

Declaration 1

Certificate 2

Acknowledgement 3

Abstract 4

1 Introduction

1.1 Objective and goal of the project . . . . …….6

1.2 Problem Statement. …. 6

1.3 Challenges. . . . . ……….….....7

1.4 Field work. . . . . ……………....7

2 Literature Survey 7

3 Requirements Specification

3.1 Software Requirements 13

4 System Design 14

5 Implementation of System 14

6 Results & Discussion 21

7 Conclusion and Future Work 25

8 References 25

1.**Introduction**

Nowadays, Speech and textual forms of conversation are the primary communication forms between humans and computers that occur through web applications. The purpose of a CHATBOT is to help the users by answering to the queries . CHATBOT is a computer program that processes a user’s natural- language input and generates relatively smart, affluent, and intelligent responses sent back to the user to provide guidance . CHATBOT help with human request and allow conversation throughout every part of a day and improve productivity by assuming control over all services where people are not required. However, the most significant advantage of using CHATBOT is that it can reach a wide- ranging users on a messaging system and automate customized messages in fraction of seconds . Some common examples of CHATBOT are Amazon chat customer service, HDFC bank bot EVA, etc. These CHATBOT has been utilized in several industries to convey specific information or perform tasks, used in telling the weather , making flight / railway reservations , answering educational-based queries, or purchasing any products.

1.1 Objective and goal of the project

The key point of this platform is its simplicity in use .By creating an account , he/she can utilize all the services .Our CHATBOT meets all the curriculum requirements , placement information , academic details , examination schedules , upcoming college events and other college related guidance for the students as well as college faculties . It also serves for the parents and visitors to help them with placements and admissions.

1.2 Problem Statement

Since the workload given to the students are relatively high , it becomes difficult for them to balance their tasks .Particularly the Freshers have to face all the new challenges at one go and it becomes a challenge for them . As a result, we came up with this idea of developing a CHATBOT that helps students/faculties and acts as a coach.

1.3 Challenges faced

Taking input from the user is one of the biggest challenges we faced so we went with the numerical input system and train the system was taking too much of time so we used flask framework since it runs the program parallelly and consumes less time when compared to normal computation.

1.4 Field Work

We have asked many of our juniors and ran a survey to find out the major problems they were facing, we have noticed that many of them are confused on which platform or website they have to go inorder to see and find out about different things and notices, so we have decided to come up with a single website using a chatbot which will guide the used to their need and help them speed up their work.

2. Literature Survey

1.An Automated Conversation System Using Natural Language Processing (NLP) Chatbot in Python

In 2022, Regin, Rajest, and Shynu published a study in the Central Asian Journal of Medical and Natural Science titled "An Automated Conversation System Using Natural Language Processing (NLP) Chatbot in Python." The creation of a chatbot utilising Python and natural language processing (NLP) methods is described in the paper.

The writers begin by giving a brief overview of chatbots and their significance in a number of industries, including healthcare. After that, they go on to detail the chatbot's architecture, which is made up of three parts: an input module, a processing module,

and an output module. The processing module examines the input and creates a response, whereas the input module pre-processes and receives user input. The user is then shown the response by the output module.
The authors use NLTK and TextBlob libraries for NLP tasks such as tokenization, part-of-speech tagging, and sentiment analysis. They also use a dataset of medical questions and answers to train their chatbot.

The paper presents the results of a user study conducted to evaluate the chatbot's performance. The study involved 50 participants who interacted with the chatbot and rated its performance on several metrics. The results showed that the chatbot performed well in terms of accuracy, relevance, and ease of use.
Overall, the paper provides a detailed description of the development of a chatbot using Python and NLP techniques. It demonstrates the potential of chatbots in healthcare and highlights the importance of evaluating their performance through user studies.

2. A platform for human-chatbot interaction using python

The paper explores the use of Python programming language to provide a platform for human-chatbot interaction. The platform is intended to allow natural and user-friendly conversation between users and chatbots. The Natural Language Toolkit (NLTK) library is used for natural language processing, and the Flask web framework is used to create a web-based interface. The platform also supports the integration of various chatbot frameworks, such as ChatterBot and Dialogflow, to give a broader set of features. The article outlines the platform's architecture and provides instructions on how to build it. To deliver a more efficient and personalised experience for customers, the platform has the potential to be employed in a range of applications, including customer service and educational settings.

3.Building chatbots with Python

"Building Chatbots with Python" by Raj, Raj, and Karkal is a comprehensive guide to building chatbots using Python programming language. The book covers the basics of chatbot development, including natural language processing, machine learning, and data analysis techniques. It provides step-by-step instructions for building chatbots using popular frameworks such as ChatterBot and TensorFlow. The authors also cover important topics such as chatbot deployment, testing, and maintenance. The book is suitable for both beginners and experienced developers who want to learn how to build chatbots from scratch. It provides practical examples and case studies to demonstrate the various techniques used in chatbot development. The book also provides insights into the future of chatbot technology and its potential applications in various industries, such as healthcare, finance, and retail. Overall, "Building Chatbots with Python" is a valuable resource for anyone interested in building chatbots using Python.

4. Deep learning techniques for implementation of chatbots.

The development of chatbots using deep learning techniques is covered in the study. It gives a general overview of the current chatbot systems and their shortcomings before explaining how deep learning techniques can be applied to get over these drawbacks. The many deep learning models utilised in chatbot development, including transformer models and recurrent neural networks (RNNs), are also covered in the article. The authors include a thorough discussion of the chatbot architecture that uses these models as well as a breakdown of the many phases of development, including data collecting, preprocessing, and training. They also emphasise the value of training data in enhancing chatbot functionality. The final section of the study discusses the obstacles and potential directions for chatbot development.Overall, the paper provides valuable insights into the use of deep learning techniques for chatbot development, and is useful for researchers and developers interested in the field of chatbots and artificial intelligence.

5.Generating and analyzing chatbot responses using natural language processing

In the research, natural language processing (NLP) techniques are used to generate and analyse chatbot responses. The authors initially give a summary of the current chatbot systems and their drawbacks before describing how NLP approaches can be applied to chatbot performance. They also go through the various rule-based and machine learning NLP models that are employed in chatbot creation. The next section of the study details the architecture of the chatbot that makes use of these models and explains the various steps that went into its creation, including feature extraction, data preprocessing, and answer production. The authors also emphasise the significance of assessing chatbot performance and give an outline of the many assessment criteria employed in the industry. The paper concludes by discussing the future prospects of chatbot development and the challenges that need to be addressed. Overall, the paper provides valuable insights into the use of NLP techniques for chatbot development, and is useful for researchers and developers interested in the field of chatbots and artificial intelligence.

6.Context Outlooker, and its Application to Question Answering and other Natural Language Processing Tasks.

The paper presents COOL (Context Outlooker), a novel approach to natural language processing that aims to enhance the contextual understanding of language models. COOL is designed to enable models to generate more accurate responses to queries by providing contextual information that may not be explicitly present in the text. The paper explains how COOL works, highlighting its key features, including the use of topic modeling, context expansion, and context filtering. The authors demonstrate the effectiveness of COOL on various natural language processing tasks, including question answering and sentiment analysis, showing that it outperforms existing state-of-the-art models. They also provide a detailed analysis of the impact of different features of COOL on its performance, providing insights into how the model can be further improved. The paper concludes by discussing the potential of COOL for future research in natural language processing and its applications in various domains. Overall, the paper presents a valuable contribution to the field of natural language processing, and the proposed COOL approach has the potential to improve the accuracy and performance of language models in various tasks.

7.Language model is all you need: Natural language understanding as question answering

The paper proposes a novel approach to natural language understanding (NLU) by treating it as a question-answering (QA) problem. The authors argue that the use of large pre-trained language models (such as BERT or GPT) has significantly improved the performance of NLU systems. However, they note that traditional NLU approaches still require specialized models and handcrafted features, which can be time-consuming and expensive to develop. The authors propose a unified framework for NLU based on QA, which involves training a language model to generate the answer to a given question, without the need for specialized models or features. They demonstrate the effectiveness of this approach on several benchmark datasets, showing that it outperforms traditional NLU approaches. The authors conclude by discussing the potential of this approach for future research in NLU and its applications in various fields. Overall, the paper provides valuable insights into the use of QA as a unified framework for NLU, and is useful for researchers and developers interested in the field of natural language processing.

8. Development of an e-commerce chatbot for a university shopping mall.

The paper presents the development of an e-commerce chatbot for a university shopping mall. The chatbot was designed to provide an interactive platform for students to purchase items from the mall without physically visiting the store. The authors describe the various components of the chatbot, including the natural language processing engine, the product database, the payment gateway, and the user interface. They also explain how the chatbot was trained to recognize and respond to user queries, providing examples of different types of queries and the corresponding responses generated by the chatbot.

The authors evaluate the performance of the chatbot using various metrics, including accuracy and response time, showing that it performs well and is able to handle a wide range of queries from users. The paper concludes by discussing the potential of the chatbot for improving the shopping experience of students and the broader implications of chatbots for e-commerce. Overall, the paper provides a valuable example of how chatbots can be used in the context of e-commerce and is useful for researchers and practitioners interested in developing similar applications.

9.Implementation of database using python flask framework

In the study, the Python Flask framework is used to implement a database. Python-based Flask is a micro web framework that offers resources for creating online applications, including tools, libraries, and technologies.The writers go over the Flask framework's architecture and how a database can be implemented using it. They explain how to make tables, create a database structure, and insert data into the database. Additionally, they describe how to build a web interface for communicating with the database using Flask.

The paper serves as a helpful resource for individuals who are new to Flask and database development because it offers a step-by-step tutorial for constructing a database using Flask. Additionally, it offers pictures and examples of code to aid readers in understanding the implementation process.

Overall, the paper is a valuable resource for developers who want to learn how to use Flask to implement a database in their web applications.

10. An Overview of Chatbot Structure and Source Algorithms.

Overall, the paper is a valuable resource for developers who want to learn how to use Flask to implement a database in their web applications.The paper gives a general overview of the chatbots' source and structural algorithms. It goes through the value of chatbots in contemporary communication systems, including how they can improve user experience and offer quick, individualized service

The three primary parts of a chatbot are natural language processing, dialogue management, and response production, according to the authors. They also go through the various varieties of chatbots, such as rule-based, retrieval-based, and generative chatbots.The study examines the machine learning algorithms used in chatbot development, including support vector machines, decision trees, and neural networks. The usage of deep learning methods, such as convolutional neural networks and recurrent neural networks, is also covered by the writers.

The paper concludes by highlighting the importance of continuous learning and improvement in chatbots, as well as the potential of incorporating additional technologies, such as virtual and augmented reality, to enhance chatbot functionality.

Overall, the paper provides a comprehensive overview of the structure and source algorithms used in chatbots, making it a useful resource for researchers and practitioners in the field of chatbot development.

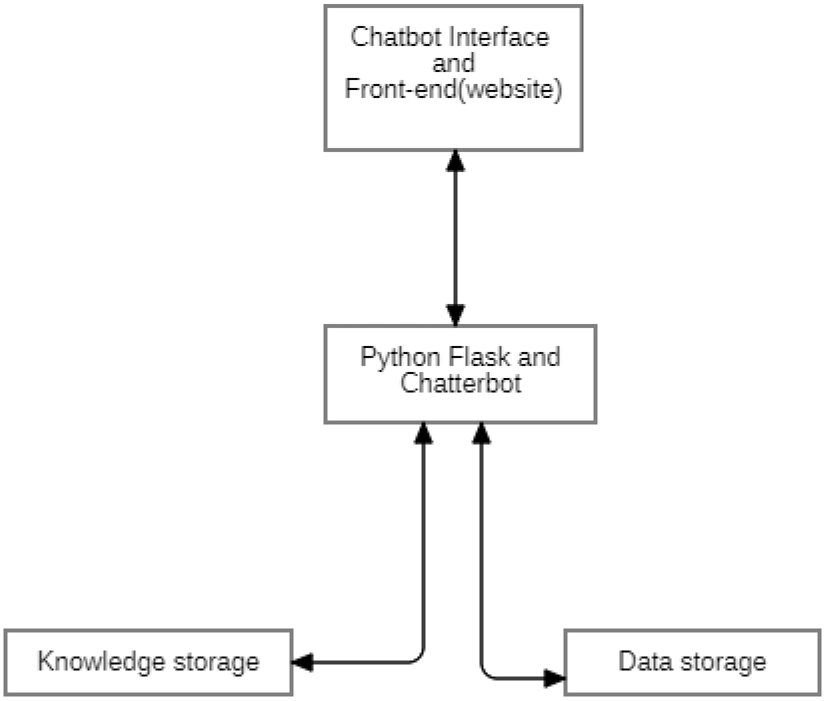
3.Requirements Specification

3.1 Software Requirements

- Python
- MySQL
- HTML
- CSS
- JAVASCRIPT
- Flask framework

4.System Design

The chatbot is run using flask and is deployed using the chatterbot extension in python methodology diagram for the chatbot is given below.


Chatbot Interface and Front-end(website)
Python Flask and Chatterbot
Knowledge storage
Data storage

5.Implementation of System

We have used python to implement the back-end of our project since it has the flask framework which supports multithreading and multiprocessing. Flask framework is multiprocessing by default and doesn't need any extra commands for the activation of multithreading. HTML along with CSS and JAVASCRIPT are used for the front-end of the project to make it look better and to provide the user a user friendly experience. We have written two codes for the whole program one for establishing all the connections and one to run the chatbot.

In the connections code the python and MySQL are connected, MySQL is used for signup and login of the users, it helps store the credentials of the user in the database and also stores the suggestions given by the users.

The connection code is given below:-

```
from chatbot import chatbot
from flask import Flask, render_template, request,session,logging,url_for,redirect,flash
from flask_recaptcha import ReCaptcha
import mysql.connector
import os

app = Flask(__name__)
recaptcha = ReCaptcha(app=app)
app.secret_key=os.urandom(24)
app.static_folder = 'static'

app.config.update(dict(
        RECAPTCHA_ENABLED = True,
        RECAPTCHA_SITE_KEY =
"6LdbAx0aAAAAAANl04WHtDbraFMufACHccHbn09L",
        RECAPTCHA_SECRET_KEY =
"6LdbAx0aAAAAAMmkgBKJ2Z9xsQjMD5YutoXC6Wee"
))

recaptcha=ReCaptcha()
recaptcha.init_app(app)

app.config['SECRET_KEY'] = 'bot'

#database connectivity
```

```
conn=mysql.connector.connect(host='localhost',port='3306',user='root',password='',database='query')
cur=conn.cursor()

# Google recaptcha - site key :
6LdbAx0aAAAAANl04WHtDbraFMufACHccHbn09L
# Google recaptcha - secret key :
6LdbAx0aAAAAMmkgBKJ2Z9xsQjMD5YutoXC6Wee

@app.route("/index")
def home():
        if 'id' in session:
        return render_template('index.html')
        else:
        return redirect('/')



@app.route('/')
def login():
        return render_template("login.html")

@app.route('/register')
def about():
        return render_template('register.html')

@app.route('/forgot')
def forgot():
        return render_template('forgot.html')

@app.route('/login_validation',methods=['POST'])
def login_validation():
        email=request.form.get('email')
        password=request.form.get('password')

        cur.execute("""SELECT * FROM `users` WHERE `email` LIKE '{}' AND
`password` LIKE '{}'""".format(email,password))
        users = cur.fetchall()
        if len(users)>0:
        session['id']=users[0][0]
        flash('You were successfully logged in')
        return redirect('/index')
        else:
        flash('Invalid credentials !!!')
        return redirect('/')
        # return "The Email is {} and the Password is {}".format(email,password)
        # return render_template('register.html')

@app.route('/add_user',methods=['POST'])
def add_user():
        name=request.form.get('name')
        email=request.form.get('uemail')
        password=request.form.get('upassword')

        #cur.execute("UPDATE users SET password='{}'WHERE name =
'{}'".format(password, name))
        cur.execute("""INSERT INTO  users(name,email,password)
VALUES('{}','{}','{}')""".format(name,email,password))
        conn.commit()
        cur.execute("""SELECT * FROM `users` WHERE `email` LIKE
'{}'""".format(email))
        myuser=cur.fetchall()
        flash('You have successfully registered!')
        session['id']=myuser[0][0]
        return redirect('/index')

@app.route('/suggestion',methods=['POST'])
def suggestion():
        email=request.form.get('uemail')
        suggesMess=request.form.get('message')

        cur.execute("""INSERT INTO  suggestion(email,message)
VALUES('{}','{}')""".format(email,suggesMess))
        conn.commit()
        flash('You suggestion is succesfully sent!')
        return redirect('/index')

@app.route('/add_user',methods=['POST'])
def register():
        if recaptcha.verify():
        flash('New User Added Successfully')
        return redirect('/register')
        else:
        flash('Error Recaptcha')
        return redirect('/register')



@app.route('/logout')
def logout():
        session.pop('id')
   return redirect('/')

@app.route("/get")
def get_bot_response():
        userText = request.args.get('msg')
        return str(chatbot.get_response(userText))

if __name__ == "__main__":
        # app.secret_key=""
        app.run()
```

First we have imported all the required libraries in python. Some of the libraries which we imported are:-

- Flask
- Tensorflow
- Chatterbot
- MySQL connector

Then we have connected python to MySQL then we have taken different functions for login, logout, signup, chatbot responses etc. this code is mainly for the connection between python and MySQL and to call the chat bot code.

The second code that we used in the project is for the chatbot, which is the main attribute for our project, the chat bot works on a basic principle it should display a menu and ask the user to choose one of it and the bot should respond to the choice which was taken by the user. We have made this chat bot for student enquiry and all the links which are required by the user are stored in it so that user can be redirected to whatever page he wants to with the guidance from the chatbot.

The code is given below:-

```
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer
import spacy
spacy.load('en_core_web_sm')
# from spacy.lang.en import English
from chatterbot.trainers import ChatterBotCorpusTrainer
```

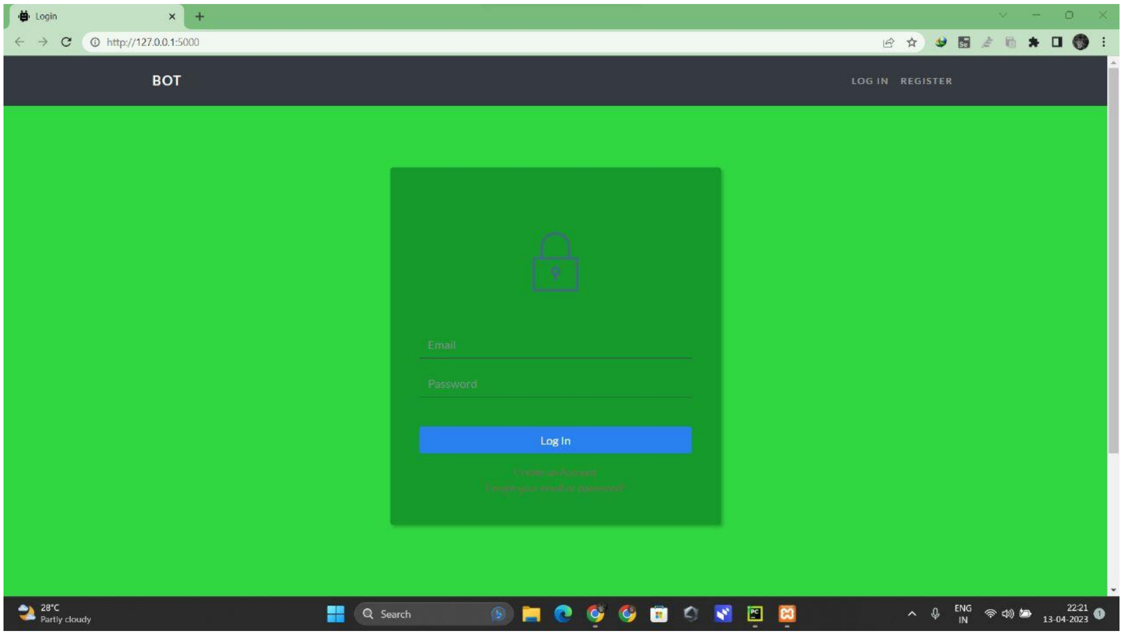
```
# Creating ChatBot Instance
chatbot = ChatBot('<b>BOT</b>')

# nlp = spacy.load("en_core_web_sm")

chatbot = ChatBot(
      'ChatBot',
  storage_adapter='chatterbot.storage.SQLStorageAdapter',
      logic_adapters=[
      {
      'import_path': 'chatterbot.logic.BestMatch',
      'default_response': "Hi there, Welcome to VIT👋 If you need any assistance, I'm always here.Go ahead and write the number of any query. 😃✨<b><br><br> Which of the following user groups do you belong to? <br><br>1.&emsp;Student's Section Enquiry.</br>2.&emsp;Faculty Section Enquiry. </br>3.&emsp;Parent's Section Enquiry.</br>4.&emsp;Visitor's Section Enquiry.</br><br>",
      'maximum_similarity_threshold': 0.90
      }
      ],
  database_uri='sqlite:///database.sqlite3'
)

trainer = ListTrainer(chatbot)
```

First we have imported the chatter bot and spacy libraries and the chatbot responses are written after that, The code automatically creates a database so that the chatbot is easily trained and does not require any manual coding of the database.

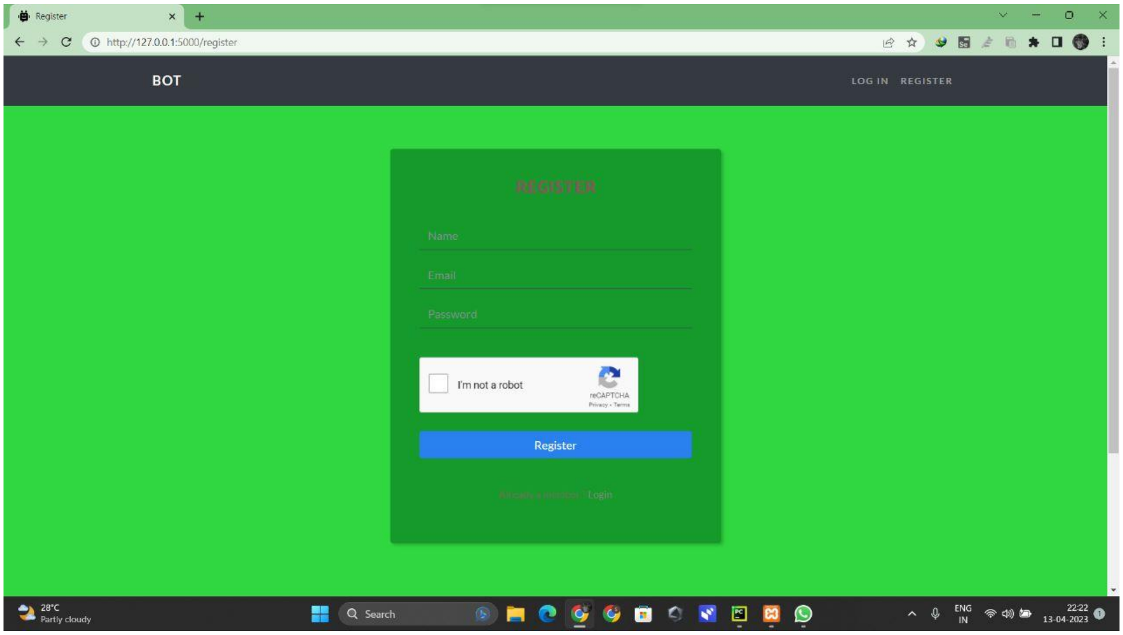
6. Results and Discussion

Website Screenshots:-

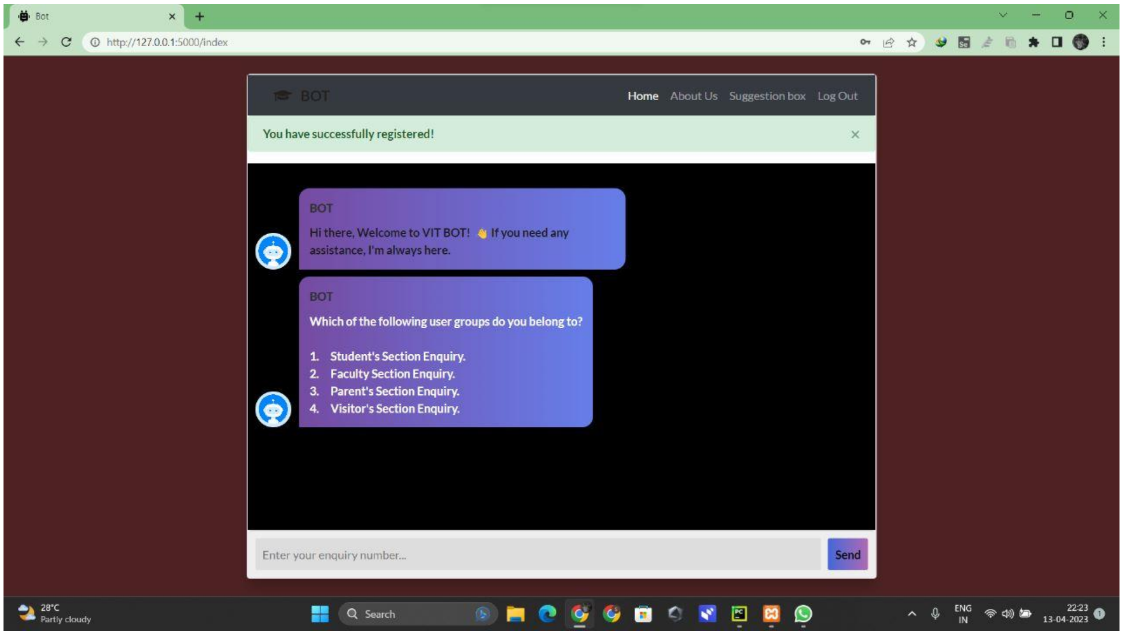
Login:-



Signup:-



Chatbot:-



You have successfully registered!

BOT

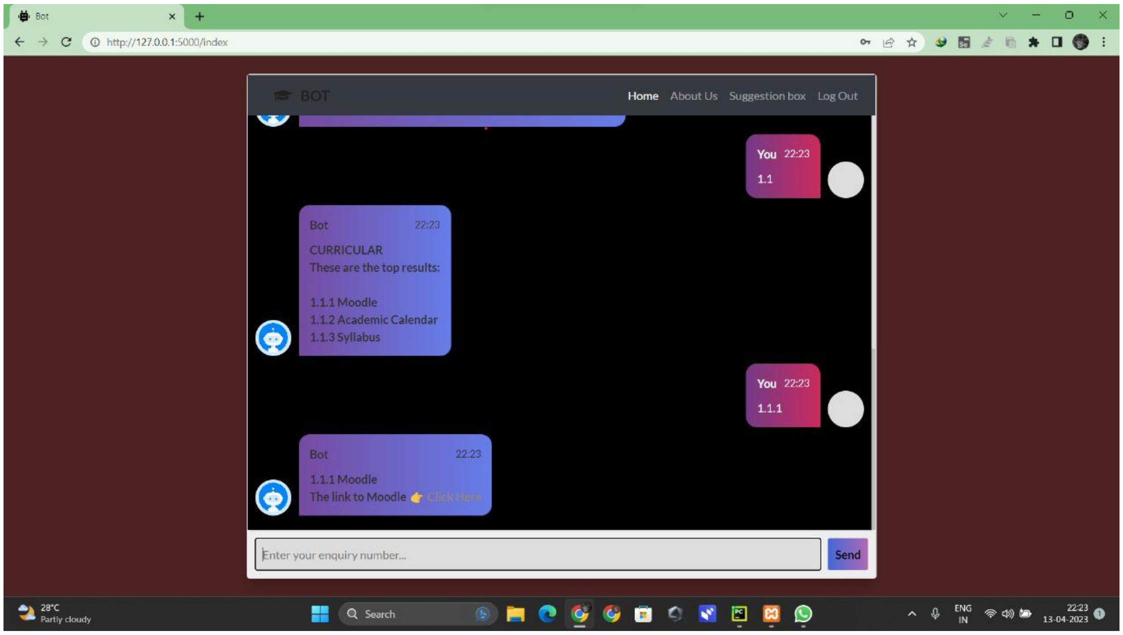
Hi there, Welcome to VIT BOT! 👋 If you need any assistance, I'm always here.

BOT

Which of the following user groups do you belong to?

1. Student's Section Enquiry.
2. Faculty Section Enquiry.
3. Parent's Section Enquiry.
4. Visitor's Section Enquiry.

Enter your enquiry number...



You 22:23

1.1

Bot 22:23

CURRICULAR
These are the top results:

1.1.1 Moodle
1.1.2 Academic Calendar
1.1.3 Syllabus

You 22:23

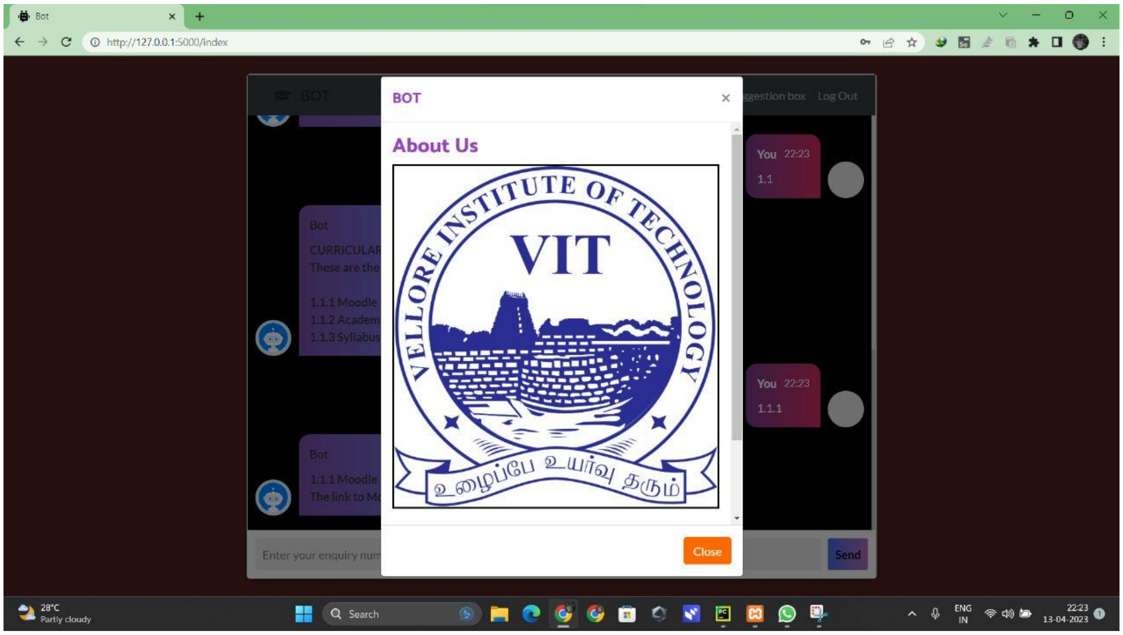
1.1.1

Bot 22:23

1.1.1 Moodle
The link to Moodle 👉 Click Here

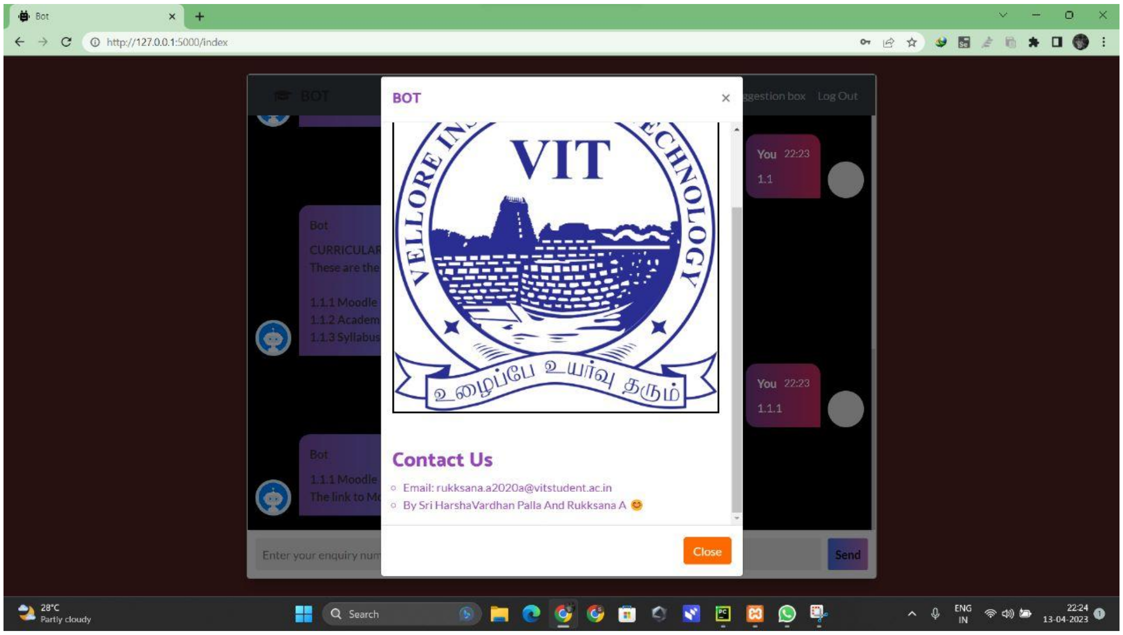
Enter your enquiry number...

About us:-



BOT

About Us

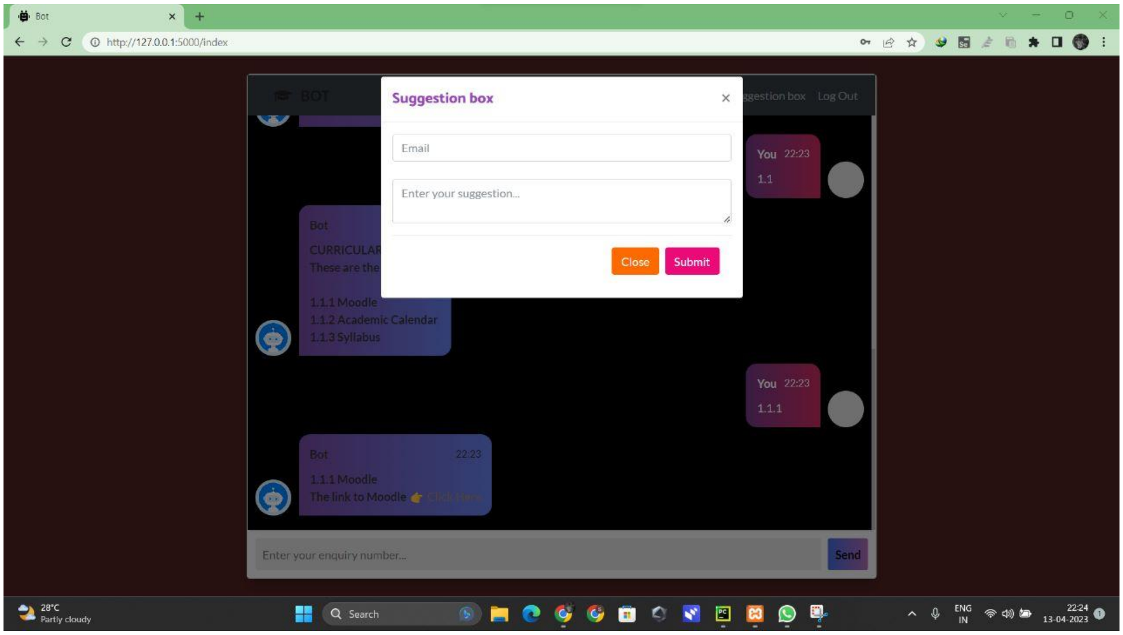


BOT

Contact Us

- Email: rukksana.a2020a@vitstudent.ac.in
- By Sri HarshaVardhan Palla And Rukksana A 😊

Suggestion Box:-



You 22:23
1.1

Bot
CURRICULAR
These are the

1.1.1 Moodle
1.1.2 Academic Calendar
1.1.3 Syllabus

You 22:23
1.1.1

Bot 22:23
1.1.1 Moodle
The link to Moodle 👉 Click Here

The code is run using the command python run server.py the connection code which combines everything is the server.py for this project, As expected when the command is given the link is generated and we are able to access the chatbot by giving the email id and password or by creating a new login if you are a new user.

The chatbot is giving the proper responses and is providing links to redirect the user to the desired webpage according to his requirement, the user can give his suggestions in the suggestion box and its getting stored in the database.

The about us tab is also working without any glitches. When the program has is exited the thread IDs where the objects have been created are printed in the terminal since flask is used in the project it is run in parallel using multiple threads and the code is also set to run in parallel in the pycharm editor.

7. Conclusion and Future Work

The Main Objective of the project is to develop a chatbot that will be used to identify answers related to users submitted questions using parallel computing . The chatbot is developed with the help of flask which will help easy the coding and provide better results. College Enquiry Chatbot is helpful in guiding students with correct and most up to date sources of information. It is advantageous for international applicants for queries such as fee payment and academic matters. Students can get the information at their fingertips rather than visiting college office. It improves efficiency by taking over tasks for which humans are not essential.

The future work of the chatbot is to improve its efficiency, provide a better and friendly UI for the user and to implement more modules.

8. References

1.Regin, R., Rajest, S. S., & Shynu, T. (2022). An Automated Conversation System Using Natural Language Processing (NLP) Chatbot in Python. Central Asian Journal of Medical and Natural Science, 3(4), 314-336.

2.Kohli, B., Choudhury, T., Sharma, S., & Kumar, P. (2018, August). A platform for human-chatbot interaction using python. In 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 439-444). IEEE.

3.Raj, S., Raj, K., & Karkal. (2019). Building chatbots with Python. Apress.

4.Karri, S. P. R., & Kumar, B. S. (2020, January). Deep learning techniques for implementation of chatbots. In 2020 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-5). IEEE.

5.Aleedy, M., Shaiba, H., & Bezbradica, M. (2019). Generating and analyzing chatbot responses using natural language processing. International Journal of Advanced Computer Science and Applications, 10(9).

6.Zhu, Fangyi, See-Kiong Ng, and Stéphane Bressan. "COOL, a Context Outlooker, and its Application to Question Answering and other Natural Language Processing Tasks." arXiv preprint arXiv:2204.09593 (2022).

7.Namazifar, M., Papangelis, A., Tur, G., & Hakkani-Tür, D. (2021, June). Language model is all you need: Natural language understanding as question answering. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7803-7807). IEEE.

8.Oguntosin, V., & Olomo, A. (2021). Development of an e-commerce chatbot for a university shopping mall. Applied Computational Intelligence and Soft Computing, 2021, 1-14.

9.Singh, M., Verma, A., Parasher, A., Chauhan, N., & Budhiraja, G. (2019). Implementation of database using python flask framework. International Journal of Engineering and Computer Science, 8(12), 24890-24893.

10.Eltahir, A. M., Abdulla, H., Platos, J., & Snasel, V. (2022, July). An Overview of Chatbot Structure and Source Algorithms. In 2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC) (pp. 91-98). IEEE.

11.Duckett, J. (2014). Web design with HTML, CSS, JavaScript and jQuery set (Vol. 1). IN: Wiley.

12.Deshmukh, S., Mane, D., & Retawade, A. (2019, March). Building a single page application web front-end for E-learning site. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 985-987). IEEE.

13. R. Anand, K. Arunachalam, and K. Raja. (2015)An Intelligent Chatbot for e-Learning Platforms Using AIML .

14.Gupta, S., Borkar, D., De Mello, C., & Patil, S. (2015). An e-commerce website based chatbot. *International Journal of Computer Science and Information Technologies*, *6*(2), 1483-1485.