DEPARTMENT OF COMPUTER APPLICATIONS COCHIN

UNIVERSITY OF SCIENCE AND TECHNOLOGY

COCHIN-22

M.Sc COMPUTER SCIENCE

Specialization in Soft Computing

SUBMITTED BY

CHRISTIN MARIYA (32319010)

MIDHUNA P N (32319015)

NAHEELA P K (32319016)

SATHYAJITH K P (32319024)

PROJECT REPORT

CHATBOT FOR COVID-19 UPDATES

DECEMBER 2020

DEPARTMENT OF COMPUTER APPLICATIONS

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

COCHIN-22

CERTIFICATE

This is to certify that the mini project entitled "CHATBOT FOR COVID 19 UPDATES" is a bonafide record of the project work done by Christin Mariya (Reg No:32319010), Midhuna P N (Reg No:32319015) , Naheela P K (Reg No:32319016) & Sathyajith K P (Reg No:32319024) under supervision and guidance in partial fulfilment of the award of Degree of Masters of Science in Computer Science during the year 2019-2021 at Department of Computer Applications, Cochin University of science and Technology, Cochin.

Internal Examiner

Department of Computer Applications

CUSAT

Head of the department

Department of Computer Applications

CUSAT

DECLARATION

We hereby declare that the Report entitled “Chabot for Covid-19 Updates” in partial fulfillment of the requirements for the award of the Degree of Master of Computer Science is a record of bonafide work done by us under the supervision and guidance of Mrs. Archana Krishnan, Department of Computer Application, Cochin University of Science and Technology, Cochin.

Place: CUSAT
Date: 19/12/2020

Christin Mariya
Midhuna PN
Naheela P K
Sathyajith K P

ACKNOWLEDGEMENT

With great pleasure we hereby acknowledge that the help given to us by various individuals throughout the project/ the project itself is an acknowledgement to the inspiration driven and technical assistance contributed by many individuals.

We extend our sincere thanks to all the non-teaching staff for providing the necessary facilities and help. We thank Lord God, the almighty for His immeasurable blessing upon my life.

We are pleased our indebtedness to Dr. Sabu M K, Head of the Department, Department of Computer Applications, CUSAT for his gracious encouragement. And also, we are thankful to Mrs. Archana Krishnan, our project guide for her support.

We are obliged to the teaching staff for being helpful and co-operative during the period of project. We extend our heartfelt thankful to parents, friends and well wishes for their support and timely help.

TABLE OF CONTENT

LIST OF FIGURES

ABSTRACT

On 11 March 2020, WHO declared Novel Coronavirus Disease COVID-19 outbreak as a pandemic and reiterated the call for countries to take immediate actions and scale up response to treat, detect and reduce transmission to save people's lives. As of December 2020, there are more than 9 lakh confirmed Covid-19 positive cases in India. The counts are increasing all over the world. As more people are getting affected by this pandemic, the need for accurate and relevant information regarding Covid-19 and its update has also increased. There are many sources such as websites, television, newspapers, social media etc. But it would be easy and effective if there is a person from whom we can get answers for the questions regarding the pandemic by chatting with that person. But that is not feasible. This is where Artificial Intelligence gains its importance and the concept of chat bot arises.

Chabot are computer programs that can carry out human-like conversations with people using lightweight messaging app UI, language-based rules, or AI. The term "ChatterBot" was originally coined by Michael Mauldin. The first chatbots appeared in the 1960s. Chatbots currently operate on several channels, including the Internet, applications, and messaging platforms. They are useful in providing standard solutions to common problems. Chatbots help save time, human efforts, cost and provide efficient solutions (and keep improving from learning) from time to time. Interacting with everyone manually and resolving the problems can be a tedious task. This is where Chabot come into the picture. Initially, chatbots were only used as a tool that solved customers' queries, but today they have evolved into a personal companion. From recommending a product to getting feedback from the customers, chatbots can do everything.

In our project, a chatbot integrated with Telegram and Facebook messenger application with following features on hand tip without depending on any other media. The project is based on Artificial Intelligence and Natural Language Processing and built using Google dialog flow with the following features:

1. The bot would answer all the queries/FAQ related to Covid-19
2. The bot would show the statistics worldwide, Country-wise, Indian State, District, On Google Map using 3rd Party API
3. Able to send Report to User Email with Prevention Measure Attachments.
4. Able to show Help Desk, Live News, Government Announcements, Images and Videos related to Covid-19.
5. Able to Save User-Bot conversation on Database.

INTRODUCTION

The coronavirus outbreak occurred in our modern, highly connected, information-dense world. Yet, dissemination of accurate, up-to-date information about the spread of the disease remains a challenge. COVID outbreak has major consequences for society worldwide. People are rightly concerned and have many urgent questions. Traditional media (radio, TV and print channels) have shrinking audiences and the best require subscriptions for access. Several local and regional authorities have made text alerts available, but these are available just to those who register and aren't available in every locality. Younger audiences prefer social media over traditional channels but many social media channels have been challenged by fake news and privacy breaches, and aren't always fully reliable. The World Health Organization provides answers to frequently asked questions regarding the coronavirus on their website. However, people may have to search for a while before you have found the right answer to your question. It is vital that people are well informed about current measures. This way we can efficiently limit mass spread. A chatbot could perfectly handle this situation. With this thought we started to build a chatbot which gives daily COVID updates (country wise, state wise and district wise reports).

Our chatbot should be able to answer any questions regarding the coronavirus. Advances in Natural Language Processing have enabled conversational AI technologies and widened their reach, leading to tools such as Siri, Alexa, and Google Home that are part of many consumers' everyday lives. The intuitive interface of chatbots presents a low-friction approach to disseminate critical information to vast populations. And Chatbots, like websites, are available 24/7.

Additionally, chatbots offer strong potential for curated information. The information can be customized to the needs and symptoms of the individual. Response to specific questions can be provided in an interactive manner, more rapidly than traditional online search methods. The information is also adaptable to local guidelines and regulations, based on the location of the user. Anyone with access to the chatbots can do so from a range of devices (online computer, smart phone, or analogue phone in some cases). Widespread adoption of chatbots for COVID-19 information can also reduce the burden on hospital call centres.

Natural Language Processing in Chatbot

Our conversational bot is AI-powered and based on a server less architecture. It is embedded with Natural Language Processing (NLP) and Natural Language Understanding (NLU) to understand the user's query and return respective responses. This mainly focuses on the four core NLP tasks: information retrieval, named entity recognition, literature-based discovery, and question answering. NLP facilitates to read, decode, understand, and make sense of the human languages.

Dialog flow for conversation

By using the Google Dialogflow we can easily add chat functionality to our chatbot using Google's chat widget or our own chat interface. The Dialogflow Telegram integration provides a customizable chat dialog for our COVID chatbot so that it can be used easily. In the Dialogflow there will be intents which match the user query and after processing the request user gets a response.

Rapid API for live data

To show the Covid-19 statistics to the chatbot we used Rapid API. In this API, we will be able to get all the information needed to track the spread of the virus. Much of this information comes directly from top medical institutions, such as the COVID-19 API website, which directly reports John Hopkins CSSE data. Information pulled from this API return in JSON format for simple data integration.

LITERATURE REVIEW

With the evolution of AI and Natural Language processing, chat bots have been developed for various purposes. Many business enterprises and websites use chatbot so that their customers can ask their queries at any time without waiting. The users also prefer chat bots over human customer care because chatbots never get tired or irritated with multiple queries and chatbots are able to give accurate results in less time and in a convenient way. During the pandemic, the statistics of Covid-19, precautions, government announcement etcetera are the most relevant data that people searched for.

The World Health Organization has released a chat bot integrated with WhatsApp, Facebook messenger and Viber. It provides the statistics of Covid-19 updates. But the chat bot is not well trained as it cannot even understand the similarity of “Hai” and “Hello”. The users need to be more specific on what they have to type to the WHO chat bot. The WHO chatbot works on the basis of keywords and numbers only. The concept of natural language understanding is not evident in the working of that chat bot.

In our system, we have trained the chat bot in such a way that they can easily recognize the small talks. Using Natural Language Understanding, our chat bot can understand that “Hai” and “hello” indicates the same meaning. This makes our chat bot more user friendly. Also, they don’t have an option to send the Covid-19 details to the user’s email. There are some other chat bots that are developed for Covid-19 updates. Some of them only provide the statistics while some others don't undertake the concept of natural language understanding. So in our system, we implemented natural language understanding so that users can chat with our bot and get desired results with a casual talk.

SYSTEM REQUIREMENTS

HARDWARE REQUIRMENTS

- RAM: 8 GB RAM
- PROCESSOR: Intel Core i3 CPU
- DISPLAY: 720 Display
- HARD DISK: 1TB
- PROCESSOR SPEED: 2Ghz

SOFTWARE REQUIRMENTS

- **Windows 10**

  Windows 10 is a Microsoft operating system for personal computers, tablets, embedded devices and internet of things devices.

- **Flask**

  A micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. Extensions exist for object-relational mappers, form validation, and upload handling, various open authentication technologies and several common frameworks related tools. Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.

- **Google DialogFlow**

  It is a natural language understanding platform that makes it easy to design and integrate a conversational user interface into your mobile app, web application, device, bot, interactive voice response system, and so on. Using Dialogflow, you can provide new and engaging ways for users to interact with your product.

- **Rapid API**

  Rapid API is the world's largest API Marketplace used by over one million developers to discover and connect to thousands of APIs. Using Rapid API, developers can search and test the APIs, subscribe, and connect to the APIs all with a single account, single API key and single SDK. Engineering teams also use Rapid API to share internal APIs and micro service documentation.

- **Postman**

  Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so we can create better APIs—faster. It is a great tool when trying to dissect RESTful APIs made by others or test ones we have made. It offers a sleek user interface with which to make HTML requests, without the hassle of writing a bunch of code just to test an API's functionality.

- **Ngrok**

  Ngrok is a cross-platform application that enables us to expose a local development server to the Internet with minimal effort. The software makes our locally-hosted web server appear to be hosted on a subdomain of ngrok.com, meaning that no public IP or domain name on the local machine is needed. Similar functionality can be achieved with Reverse SSH Tunneling, but this requires more setup as well as hosting of our own remote server.

METHODOLOGY

In this project, waterfall model is used as it is easy to implement and best suited for this application.

Waterfall Model:

It is a simplest model which states that the phases are organized in a linear order. In this model, a project begins with feasibility analysis. Upon successfully demonstrating the feasibility of a project, the requirements analysis and project planning begins. The design starts after the requirements analysis is complete, and coding begins after the design is complete. Once the programming is completed, the code is integrated and testing is done. After this, the regular operation and maintenance of the system takes place.

Natural language processing (NLP) is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language. NLP draws from many disciplines, including computer science and computational linguistics, in its pursuit to fill the gap between human communication and computer understanding. It is the technology used to aid computers to understand the human's natural language. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation. NLP is commonly used for text mining, machine translation, and automated question answering.

Chatbots are software applications that use artificial intelligence & natural language processing to understand what a human wants, and guides them to their desired outcome with as little work for the end user as possible. Like a virtual assistant for your customer experience touchpoints. It is a software application used to conduct an on-line chat conversation via text or text-to-speech, providing direct contact with a live human agent. Chatbots currently operate on several channels, including the Internet, applications, and messaging platforms. Chatbots help save time, human efforts, cost and provide efficient solutions (and keep improving from learning) from time to time.

Tokens are the building blocks of Natural Language; the most common way of processing the raw text happens at the token level. In order to get our computer to understand any text, we need to break that word down in a way that our machine can understand. That's where the concept of tokenization in Natural Language Processing (NLP) comes in. **Tokenization** is the process of

breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing such as parsing or text mining. Tokenization is useful both in linguistics (where it is a form of text segmentation), and in computer science, where it forms part of lexical analysis.

For the simplification of various search queries, **Stemming** and **Lemmatization** are the strategies used in NLP. These are the text normalizing and text mining procedures in the field of Natural Language Processing that are applied to adjust text, words, documents for more processing. The process of reducing inflection towards their root forms are called Stemming, this occurs in such a way that depicting a group of relatable words under the same stem, even if the root has no appropriate meaning. Lemmatization is a method responsible for grouping different inflected forms of words into the root form, having the same meaning. It is similar to stemming, in turn, it gives the stripped word that has some dictionary meaning.

Dialog Flow is a natural language understanding platform that makes it easy to design and integrate a conversational user interface into your mobile app, web application, device, bot, interactive voice response system, and so on. Using Dialogflow, you can provide new and engaging ways for users to interact with your product. Dialogflow can analyse multiple types of input from your customers, including text or audio inputs (like from a phone or voice recording). It can also respond to your customers in a couple of ways, either through text or with synthetic speech.

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common frameworks related tools. Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications. Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common frameworks related tools.

Rapid API is the world's largest API Marketplace — used by over one million developers to discover and connect to thousands of APIs. Using Rapid API, developers can search and test the APIs, subscribe, and connect to the APIs — all with a single account, single API key and single SDK. Engineering teams also use Rapid API to share internal APIs and micro service documentation. Rapid API is an API Marketplace for developers to find, connect, and manage their API connections. Find the APIs that you need for your project, embed the API into your app, and track usage of all your APIs through a single dashboard. If you have an API you've created, use Rapid API to make it available to over 1 million developers already utilizing APIs through Rapid API.

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster. Postman is a popular API client that makes it easy for developers to create, share, test and document APIs. This is done by allowing users to create and save simple and complex HTTP/s requests, as well as read their responses. Postman is an interactive and automatic tool for verifying the APIs of your project. Postman is a Google Chrome app for interacting with HTTP APIs. It presents you with a friendly GUI for constructing requests and reading responses. It works on the backend, and makes sure that each API is working as intended. It has the ability to make various types of HTTP requests (GET, POST, PUT, PATCH), saving environments for later use, converting the API to code for various languages (like JavaScript, Python).

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program. NoSQL is used as an alternative to traditional relational databases. NoSQL databases are quite useful for working with large sets of distributed data. MongoDB is a tool that can manage document-oriented information, store or retrieve information. MongoDB uses JSON-like documents with optional schemas. MongoDB supports various forms of data. It is one of the many non-relational database technologies.

DESIGN

PROJECT WORK FLOW

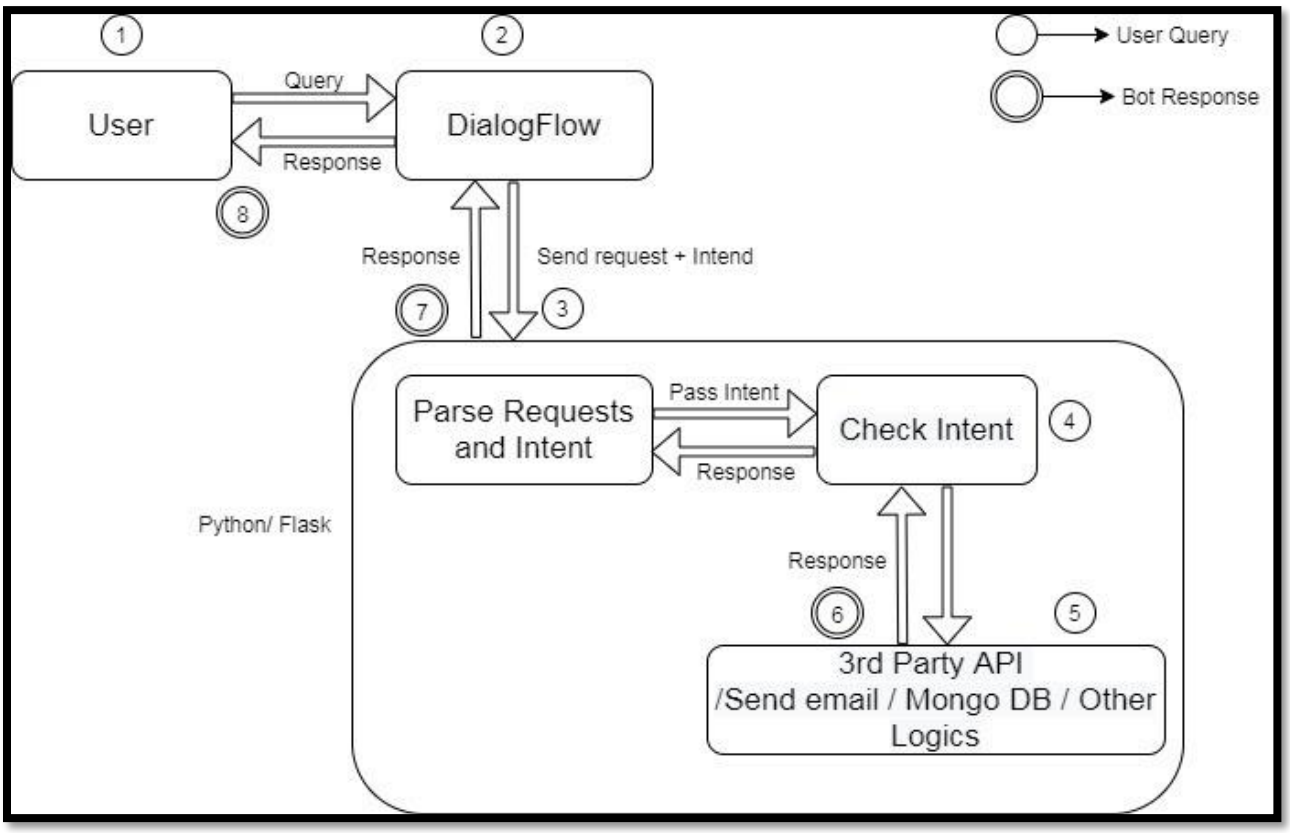

Figure 1: Project Work Flow

WORKING DESCRIPTION

Our basic architecture is given in the above image. The user queries to the chatbot in the form of natural language (text). Based on the semantic meaning of the user query, the entities are detected, which is then mapped to the respective intent on Dialogflow. Then it sends the request and the intent to Python Flask web API. Inside the Python Flask API there are different functions to perform different tasks. So now our control is in the web API. Now the Python API parse the request and intent then pass it to further functions. These functions decide whether it has to perform some actions like send email, connect to database (MongoDB), 3rd party API actions etc. After performing the task, the function will return the response from Python Flask to Dialogflow. Then the Dialogflow will return the response to our user. This is how our chatbot works.

INTENT FLOW IN DIALOGFLOW

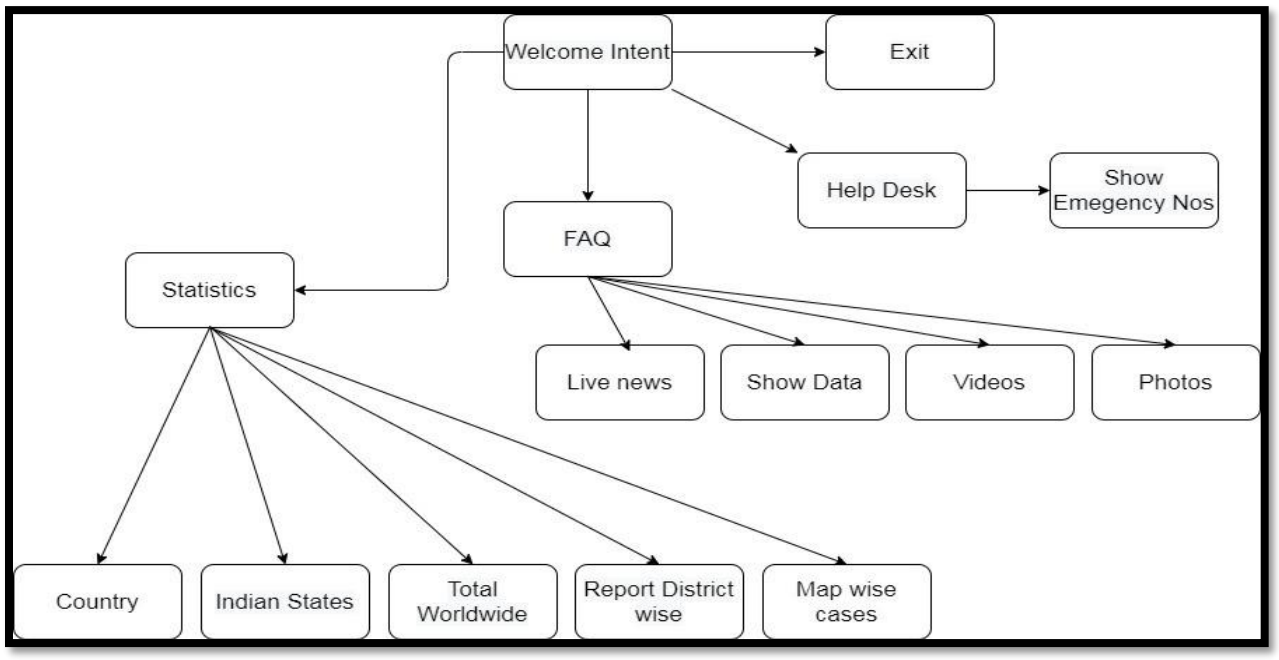

Figure 2: Intent Flow in Dialog Flow

This picture represents the intent flow in the Dialogflow. Intents are the actions your user wants to execute. It determines where conversation will go and what an agent should do. They are more like root verbs in the dialogue. We need to train the agents to recognize the intents from the phrases or conversations. Also, we need to add training phrases, then train and test them. In the diagram we can see there is welcome intent. There are four intents connected to the welcome intent that are: Statistics, FAQ, helpdesk or exit. They are having different Childs which are also intents but connected to the parent intent. And few intents like send email or end conversation are mentioned in the figure because they are direct intents for our Covid-19 Updates bot. The user is not dependent on any of these options. User can ask for email after these options anytime.

We have also used entities to extract useful facts from the users. It helps to identify who, what, when and where. We can also give synonyms for entities in Dialog Flow. Then we need to map entities with the intent. Time, date, country etcetera is some of the system entities. In order to avoid repetition during a conversation, we might need to store information. This is done using Context. It gives a follow up conversation. There is an input context and output context. When the agents have to access external system or data stores, we can write code in the fulfillment section. In the fulfillment sections we can map the actions.

SCREENSHOTS

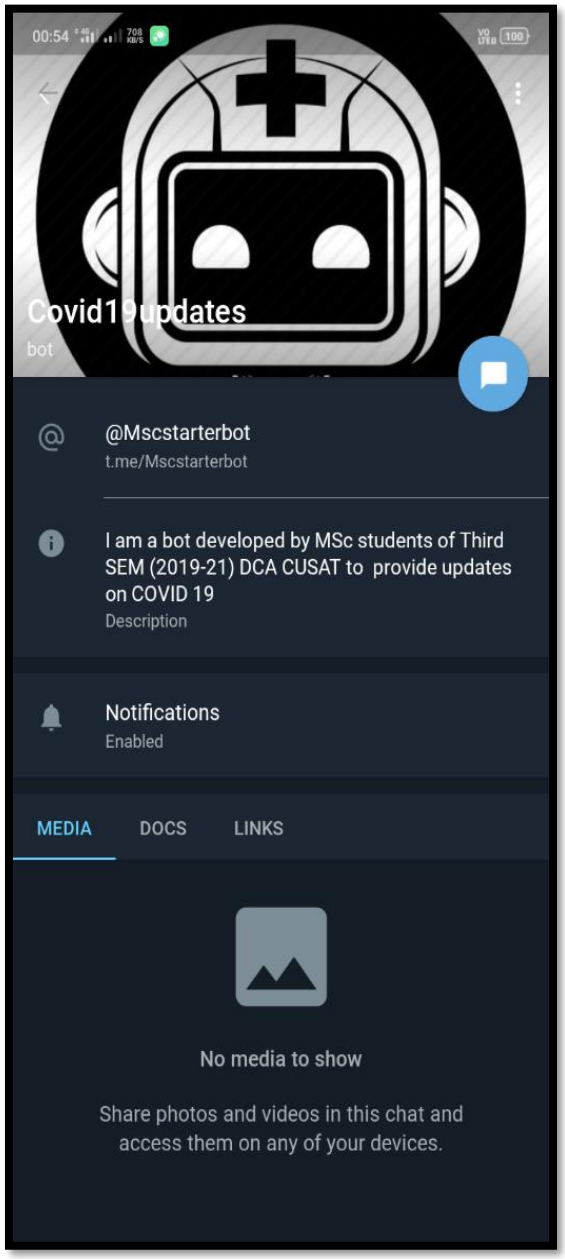

Figure 3: The telegram integrated Chabot's contact screen view

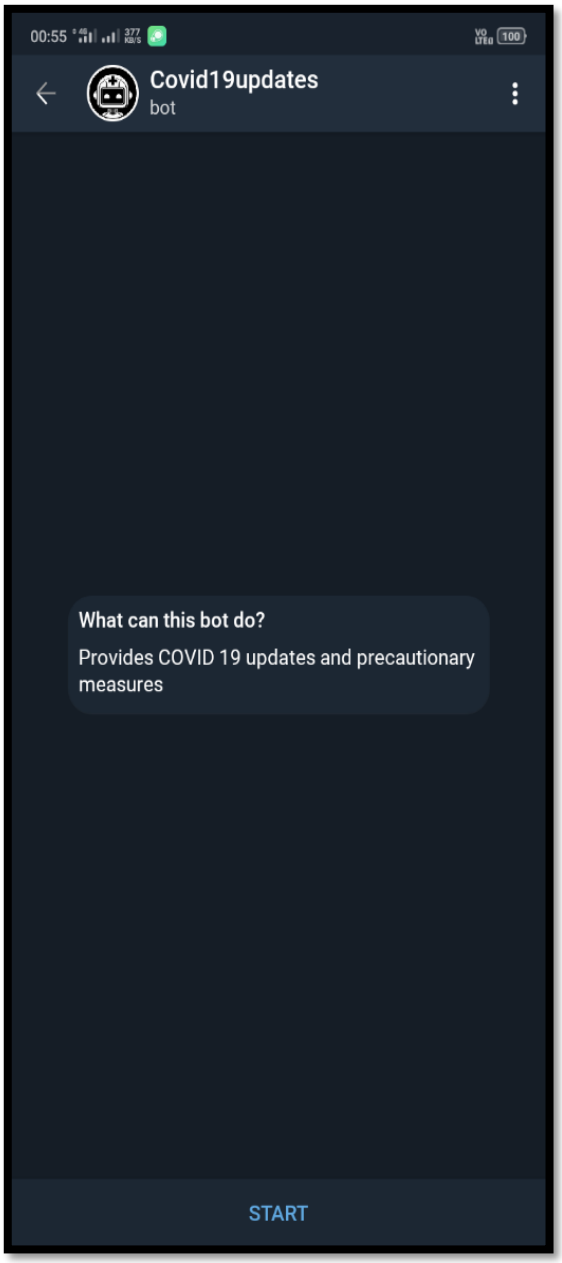

Figure 4: Welcome screen

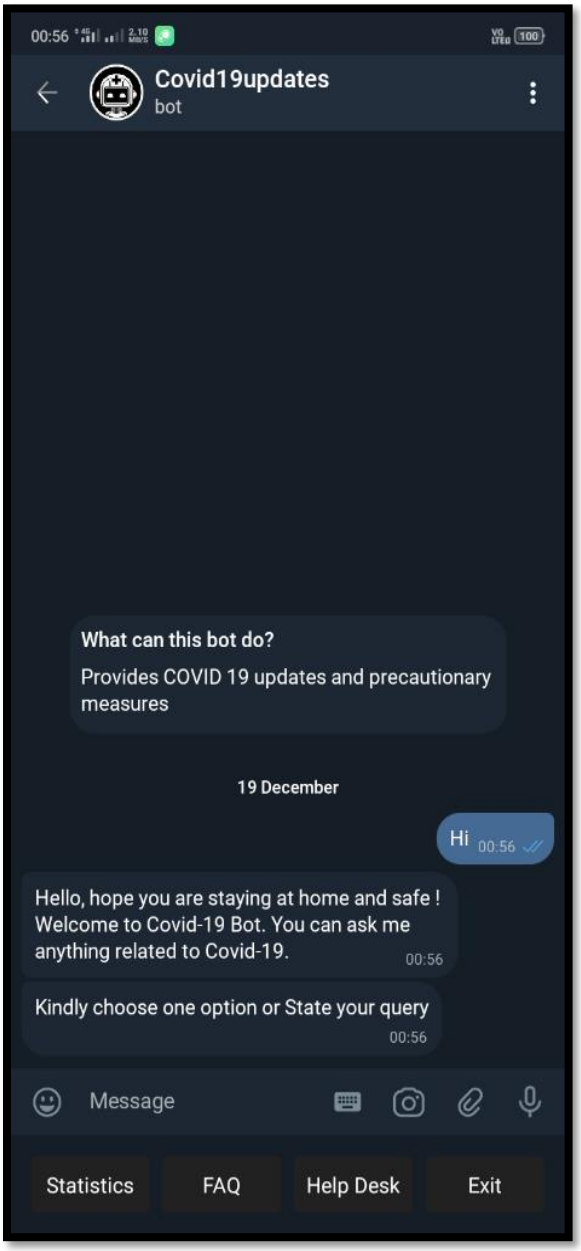

Figure 5: Chatbot is invoked with the "Hi" message and starts to answer

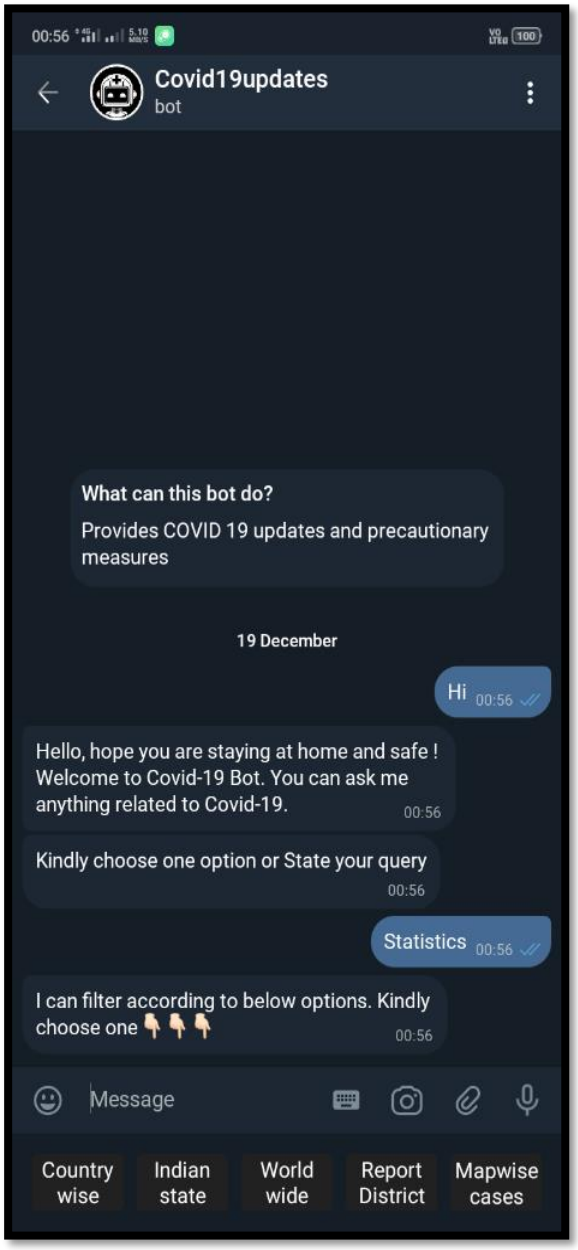

Figure 6: Chatbot answering user query, also provides options for accuracy

Figure 7: Displays national level report and asking for detailed report via mail

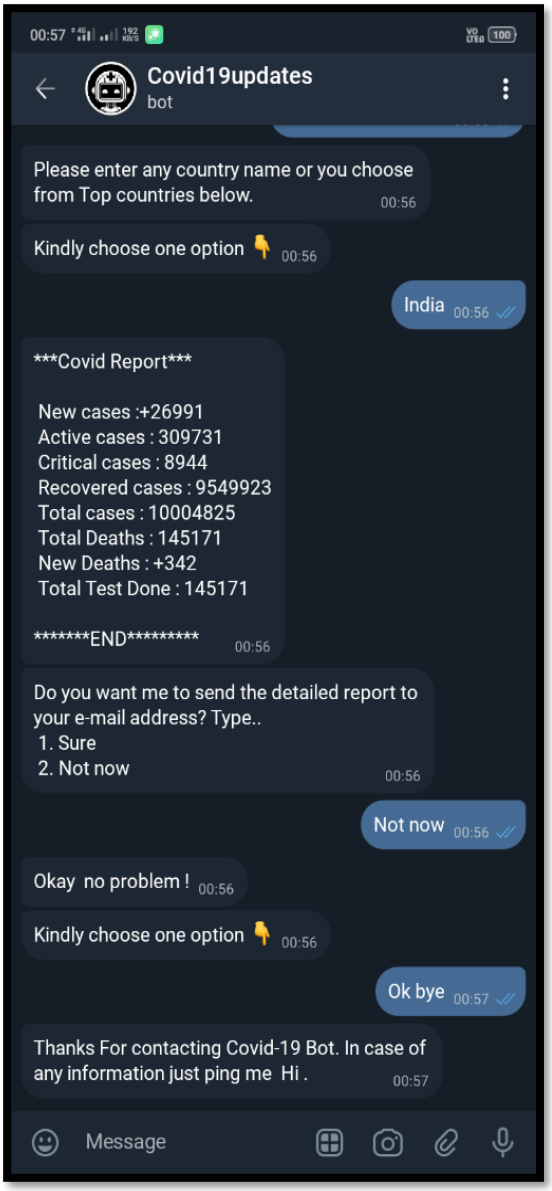

Figure 8: Understands natural language and making user friendly chats

TESTING

TESTING

INTRODUCTION

Testing is a process of creating a program with the explicit intention of finding error that is making the program fail. Successful test then, is one that finds an as yet undiscovered error. It makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. It is the stage of implementation, which ensures that the system works accurately and effectively before the live operation commences. It is a confirmation that all are correct and an opportunity to show users that the system must be tested and show that the system will operate successfully and produce expected results under expected conditions.

During testing the program to be tested is executed with set of test data and the output of the test data is evaluated to determine if the program is performing as expected. A series of testing are performed for the proposed system before the system is ready.

UNIT TESTING

Unit testing focuses on the modules independently of one another to locate errors. In this application each module is tested individually and may caught errors such as save same data two times, does not clear the fields after the save, exceeding the length of textboxes etc. Also validation is not properly done. These errors may clear under unit testing.

INTEGRATION TESTING

Integration testing is a systematic testing for construction of the program structure while at the same time conducting tests to uncover Instead of testing the system as a whole, unit testing focuses on the module that makes up the system. Each module is taken up individually and tested for correctness in coding and logic One specified target of integration testing is the interface, whether errors associated with interfacing. The objective is to take until-tested modules and build a program structure that has been dictated by design.

VALIDATION TESTING

In this application proper validation is done for email id, phone number, name etc. The submit button is enabled only if the value entered is according to that particular validation. All the fields must be filled otherwise it will be shown below field in red color. Validation is also providing for login. User name and password are validated. If they do not match, it is defined access there by providing a strong security.

TEST CASES

All test cases are carried out in the built-in format of google dialog flow.

RESULT AND ANALYSIS

CONCLUSION

In this project, we have created a chatbot that can give the Covid-19 updates, precautions and safety measures through telegram. It is embedded with Natural Language Processing (NLP) and Natural Language Understanding (NLU) to understand the user's query and return respective responses. Users can have a pleasant conversation with our chatbot in a casual way and can ask about the Covid-19 statistics. Users can even request for getting the details into their mail. So users need not have to search for the details into various websites and they can get the details through a simple chat. Users can also ask for government announcements regarding Covid-19, live news etc.

1. The bot would answer all the queries/FAQ related to Covid-19
2. The bot would show the statistics worldwide, Country-wise, Indian State, District, On Google Map using 3rd Party API
3. Able to send Report to User Email with Prevention Measure Attachments.
4. Able to show Help Desk, Live News, Government Announcements, Images and Videos related to Covid-19.
5. Able to Save User-Bot conversation on Database.

FUTURE SCOPE

Currently, we have integrated our chatbot with telegram only. It can also be integrated with WhatsApp and many other social media platforms. It can also be deployed on public host. Also this project can be further extended with functionalities like giving details of healthcare facilities nearby their location, asking for symptoms and giving personalized health care instructions to the Covid-19 patients based on the symptoms. It can be developed with languages other than English. And could also respond to audio messages.

REFERENCE

[1] Medium Dialog Flow Restaurant Chatbot Tutorial, Adi Cucolaș, Apr 14, 2018
https://chatbotslife.com/dialogflow-restaurant-bot-tutorial-d29b665eb975

[2] Medium How to Create a Chatbot with Google Dialog flow, Cobus Greyling, Jan 31 2020
https://cobusgreyling.medium.com/how-to-create-a-chatbot-with-google-dialogflow-60616c2b802f

[3] Coursera: Building Conversational Experiences with Dialog flow by Google Cloud
https://www.coursera.org/learn/conversational-experiences-dialogflow/home

[4] Dialog Flow Documentation
https://cloud.google.com/dialogflow/docs

[5] Building and deploying a chatbot by using Dialog flow (overview)
https://cloud.google.com/solutions/building-and-deploying-chatbot-dialogflow

APPENDIX

CODE SNIPPETS

1. Python flask implementation

```python
from flask import Flask, render_template, request, jsonify, make_response
from flask_cors import CORS, cross_origin
import requests
import pymongo
import json
import os
from saveConversation import Conversations
from DataRequests import MakeApiRequests
from sendEmail import EMailClient
from pymongo import MongoClient

app = Flask(__name__)  # initialising the flask app with the name 'app'

# geting and sending response to dialogflow
@app.route('/webhook', methods=['POST'])
@cross_origin()
def webhook():
    req = request.get_json(silent=True, force=True)
    res = processRequest(req)
    res = json.dumps(res, indent=4)
    print(res)
    r = make_response(res)
    r.headers['Content-Type'] = 'application/json'
    return r
```

2. Intent Response

```python
if intent == 'covid_searchcountry':
    cust_country = parameters.get("geo-country")
    if(cust_country=="United States"):
        cust_country = "USA"

    fulfillmentText, deaths_data, testsdone_data = makeAPIRequest(cust_country)
    webhookresponse = "***Covid Report*** \n\n" + " New cases :" + str(fulfillmentText.get('new')) + \
                      "\n" + " Active cases : " + str(
        fulfillmentText.get('active')) + "\n" + " Critical cases : " + str(fulfillmentText.get('critical')) + \
                      "\n" + " Recovered cases : " + str(
        fulfillmentText.get('recovered')) + "\n" + " Total cases : " + str(fulfillmentText.get('total')) + \
                      "\n" + " Total Deaths : " + str(deaths_data.get('total')) + "\n" + " New Deaths : " + str(
        deaths_data.get('new')) + \
                      "\n" + " Total Test Done : " + str(deaths_data.get('total')) + "\n\n*******END********* \n "
    print(webhookresponse)
    #log.saveConversations(sessionID, cust_country, webhookresponse, intent, db)
    #log.saveCases( "country", fulfillmentText, db)
```

3. API connection, email function & local hosting

```
def makeAPIRequest(query):
    api = MakeApiRequests.Api()

    if query == "world":
        return api.makeApiWorldwide()
    if query == "state":
        return api.makeApiRequestForIndianStates()
    else:
        return api.makeApiRequestForCounrty(query)


def prepareEmail(contact_list):
    mailclient = EMailClient.GMailClient()
    mailclient.sendEmail(contact_list)


'''if __name__ == '__main__':
    port = int(os.getenv('PORT'))
    print("Starting app on port %d" % port)
    app.run(debug=False, port=port, host='0.0.0.0')'''
if __name__ == "__main__":
    app.run(port=5000, debug=True)# running the app on the local machine on port 8000
```

****