A REPORT ON

Hospital Healthbot

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

(Computer Science & Engineering)

SUBMITTED BY

Jasjit Singh Rudra

May 2021

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled "Whatsapp based Hostpital Healthbot" by "Jasjit Singh Rudra" in partial fulfillment of requirements for the award of degree of B.Tech. (CSE) submitted in the Department of CSE at Chandigarh Engineering College under PUNJAB TECHNICAL UNIVERSITY, JALANDHAR is an authentic record of my own work carried out during a period from March 2021 to May 2021 under the supervision of Mr. Rajeev Sharma.

Jasjit Singh Rudra

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.
The B.Tech Viva –Voce Examination of Jasjit Singh Rudra has been held on___and accepted.

Name & Sign of Project Guide
Designation

ABSTRACT

During these pandemic times, there has had been a lot of uncertainty and inconvenience over availing of services, especially in healthcare sector. Even in the biggest healthcare facility- PGIMER, we have to avail an appointment from their website and then have to wait for many days until we are finally contacted by a doctor. **Hospital Healthbot** is designed as a better approach to this system, and is more effective during the Covid 19 pandemic. The chatbot is WhatsApp based and hence can be used by anyone having a smartphone. Instead of having to continuously monitor a website for open available slots, this chatbot provides a whole interface for its users to avail any healthcare related facility.

The aim of this healthbot is to provide a supportive hospital functioning, which includes works of receptionist, chemist, diagnostics and provide other such supportive features such as Covid 19 vaccination status, etc. The functions of this chatbot are summarized below:-

A) See a doctor

Once a user sends the message 'hello', the chatbot starts functioning. The user is instructed to select a department he/she wants to visit. Based on their response, it then asks whether it is their first-time visiting or is it a follow-up. In case of follow-up, the user will be asked for their file reference number, and in case of first time, they would be asked their details after which they will be allotted an appointment.

B) Order Medicines

Through this functionality, the user can have medicines available to him/her at their doorstep. By typing 'order medicines', the chatbot will first confirm the doctor's prescription after which it will approve the request and deliver the so referred medicines to him/her.

C) Track Vaccination Status

The status of Covid-19 vaccines, whether the stock is available or not can be checked through the chatbot by simply typing 'Vaccination Status'.

D) Diagnostics

This functionality provides a diagnostic report based on symptoms provided by the patient. It takes symptoms in comma separated values, reads and compares them to diseases and in result gives the disease that is matching with most symptoms.

TECHNOLOGIES USED

Python: It is a very powerful, high level programming language used to design the Boomerang Application and program its various modules.

Flask: Flask is a micro web framework written in Python. It does not require particular tools or libraries. **It provides- D**evelopment server and debugger, integrated support for unit testing and RESTful request dispatching

Twilio: Twilio Messaging is an API to send and receive SMS, MMS, and OTT messages globally. It uses intelligent sending features to ensure messages reliably reach end users wherever they are.

ngrok: ngrok provides a real-time web UI where we can introspect all HTTP traffic running over your tunnels. Through it we can Replay any request against your tunnel with one click.

MySQL: The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful for storing information categorically.

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to Mrs.Jaspreet Kaur, Dept. of Computer Science & Engineering, CEC-CGC, Landran for her generous guidance, help and useful suggestions.

I express my sincere gratitude to Dr. Sukhpreet Kaur, HoD in Department of CSE, CEC-CGC, Landran for her stimulating guidance, continuous encouragement.

I also wish to extend my thanks to teachers for their insightful comments and constructive suggestions to improve the quality of this research work.

Jasjit Singh Rudra
1802061, 6 B1

List of Figures

CONTENTS

1. Introduction

Problem Definition

The COVID-19 Pandemic has impacted hospitals around the world. Many hospitals have scaled back or postponed non-emergency care. This has medical consequences for the people served by the hospitals, and it has financial consequences for the hospitals. Health and social systems across the globe are struggling to cope. The situation is especially challenging in humanitarian, fragile and low-income country contexts, where health and social systems are already weak. Health facilities in many places are closing or limiting services. Services to provide sexual and reproductive health care risk being sidelined, which will lead to higher maternal mortality and morbidity.

During the novel coronavirus (COVID-19) pandemic, institutions like the Centers for Disease Control and Prevention (CDC) and the World Health Organization (WHO) have begun utilizing chatbots to share information, suggest behavior, and offer emotional support. Chatbots are software programs that talk with people through voice or text in their natural language. They often come pre-installed on smartphones or home-based smart speakers. In recent years, chatbot use for health-related purposes has increased considerably, from supporting clinicians with clinical interviews and diagnosis to aiding consumers in self-managing chronic conditions. While promising, the use of chatbots may pose safety risks. Chatbots have varied widely in their responses to questions about physical health, suicide, intimate partner violence, substance abuse, and other sensitive conversations. In one study, about a third (29%) of chatbot responses to health questions could have caused harm, and about half of those (16%) could have resulted in death if acted upon. The COVID-19 pandemic puts in stark relief the potential for chatbots to help save lives.

On 11 March 2020, the WHO Director-General “rang the alarm bell loud and clear” by calling COVID-19 a pandemic. Globally and locally, control and prevention measures have been frustrated by myriad challenges. First, accurate information is crucial, but often unknown, or obscured by misinformation. Second, disease fear and confusion contribute to under-reporting of symptoms. Third, preventative strategies such as hand washing or social distancing are costly to disseminate and enforce. Fourth, infection countermeasures (e.g., social distancing and quarantine) are psychologically damaging. For example, the SARS outbreak in 2003 resulted in a “mental health catastrophe,” in which 59% of patients in a hospitalized cohort developed a diagnosable psychiatric disorder, most commonly post-traumatic stress disorder and depression. After 30 months, less than half of this cohort psychologically recovered. In this light, the WHO has called for “large-scale implementation of high-quality, non-pharmaceutical public health measures ” to help limit new cases, and safely triage those who may be infected. Normally, resources such as clinician attention or emergency department waiting areas are used at a rate the

healthcare system can handle. In a pandemic, the cost of these resources being spent inefficiently or contaminated can be catastrophic.

Chatbots may be uniquely useful in a pandemic, but challenges in information dissemination, symptom monitoring, behavior change, and mental health support are worthy of attention. Providing reliable evidence-based information is critical in a pandemic and two issues have material impact: conflicting advice between global and local authorities, and misinformation. Chatbot developers must decide whose voice to amplify and should provide reliable information from global sources like the WHO, while also coordinating with regional authorities. Both a feature and a challenge of chatbots is their ability to link users to third-party services that then gather and share data with unknown or unexpected consequences. If deployed for symptom screening, which is currently happening for COVID-19, constitutional and regulatory boundaries are tested by sharing health-related information between companies and governments. This concern is not theoretical, as both the United States and Israel have reportedly explored using digital contact tracing to understand infection vectors. Finally, although chatbots have demonstrated feasibility in behavior change and mental health treatment, they are untested in pandemics and have demonstrated limits in health crisis detection and response.

These challenges, if only addressed in real time during a crisis, may lead to erroneous outputs from a lack of testing. With more than a billion voice searches per month, any health-related mistakes, such as misidentifying key symptoms, would be amplified with extensive harmful repercussions. Additionally, medical and public health experts must inform what chatbots say, and how they say it. Translating medical information into advice for the public requires expertise and evaluation to prevent unintended consequences. Without proper design and deployment, and ongoing monitoring, chatbots may confuse rather than help users.

Project Overview and Specifications

Healthbot actively employees Python 3 (for application), WhatsApp application for its front end design. Flask and Selenium libraries have been used at the backend to implement various features in the chatbot such as sessions, automation, requestes and live functioning. The data obtained is stored using XAMPP database is used for storage of data. MySQL queries have been used for operations on data. Twilio console is used for providing a sandbox for WhatsApp. ngRok is used to create a live tunnel, which configures endpoint URLs.

Detailed Specifications are given below:

Software Requirements:

- Whatsapp installed
- Mozilla Firefox*, Google Chrome* or any other desktop web browser* -
  *(versions released after 2014) {Desktops}
- Pycharm IDE for running python GUI. (IDLE or any other Python interpreter may also be used)
- Windows 7 or above {Desktop}
- Android 4.0.3 or iOS 9 and above {Smartphone}

Hardware Requirements:

- Intel Pentium or above processor {Desktop}
- Snapdragon 365 or above processor {Smartphone}
- 512 Mb RAM or higher
- Mouse and Keyboard for inputs
- Stable internet connection

2 Literature Survey

Existing Systems

Chatbots are also appearing in the healthcare industry. A study suggested that physicians in the United States believed that chatbots would be most beneficial for scheduling doctor appointments, locating health clinics, or providing medication information.

Whatsapp recently tied up with the World Health Organisation (WHO) to make a chatbot service that answers users' questions on Covid-19.

As the number of health chatbots multiplies with incredible speed, here are some of the most promising bots that are shaping the future of AI in healthcare :

1) OneRemission

For cancer patients and cancer survivors, the app empowers them by providing a comprehensive list of diets, exercises, and post-cancer practices, curated by Integrative Medicine experts, so that they don't need to constantly rely on a doctor.

2) Youper

Youper's A.I. monitors and improves users' emotional health with quick personalized conversations using psychological techniques. To further help one improve their emotional health, the app features personalized meditations as well as the ability to track mood and monitor emotional health.

3) SafeDrugBot

Safedrugbot is a chat messaging service that offers assistant-like support to health professionals, doctors who need appropriate data about the use of drugs during breastfeeding.

4) Babylon Health

The company offers A.I. consultation based on personal medical history and common medical knowledge as well as live video consultation with a real doctor whenever a patient needs it.

5) Florence

The chatbot is basically a “personal nurse” in the color blue, and works on Facebook Messenger, Skype or Kik. “She” can remind patients to take their pills, which might be a handy feature for older patients. You just write the name of the medicine in chat, the number of times a day you must take it and at what time. Then, Florence sends you a message in chat every time you must take the pill.

Proposed System

Hospital Healthbot project is designed for ease in making doctor’s appointments for different hospital departments, follow up examinations, diagnostics, ordering medicines or checking vaccination status. Many people fail to book appointments on online web applications due to bad user interface, high website traffic, lack of information, etc.

To counter this problem, the chatbot is whatsapp based- an app used by almost all people living on earth. By typing a simple “Hello” message, users can access features such as booking an appointment, get delivered medicines based on prescription and check availability of vaccines. The chatbot undertakes the role of a receptionist, chemist as well as diagnose simpler diseases by taking a list of symptoms.

Healthbot application is designed using python, which is a very powerful and fast language. It uses Twilio console to connect to WhatsApp using Flask library as a bridge.

Feasibility Study

a) Technical Feasibility: Healthbot can run on any modern browser, whose versions are released in/after 2014, as they can support HTML 5 and its features. There are no specific hardware

requirements, but RAM higher than 512 Mb and Intel Pentium/Similar or above processor is recommended for desktops. For mobiles, Snapdragon 365 processor and Android OS 4.0.3 is the minimum requirement. A keyboard and Mouse is required for input.

b) Operational Feasibility: Healthbot has a self-explanatory user interface and should face minimum resistance from users. It uses whatsapp's interface, which has one of the best UI.

c) Economic Feasibility: There are minimal hardware and software costs for Healthbot as it is not system heavy. Cost of training and application maintenance would be average.

d) Legal Feasibility: Heathbot follows all W3 standards of design and thus should pass required quality tests/standards for software design.

e) Schedule Feasibility: A team size of 5 professionals for design/updates, 3 for maintenance and 10 for customer relations would be required during launch of project. A small team of medical professionals is also recommended.

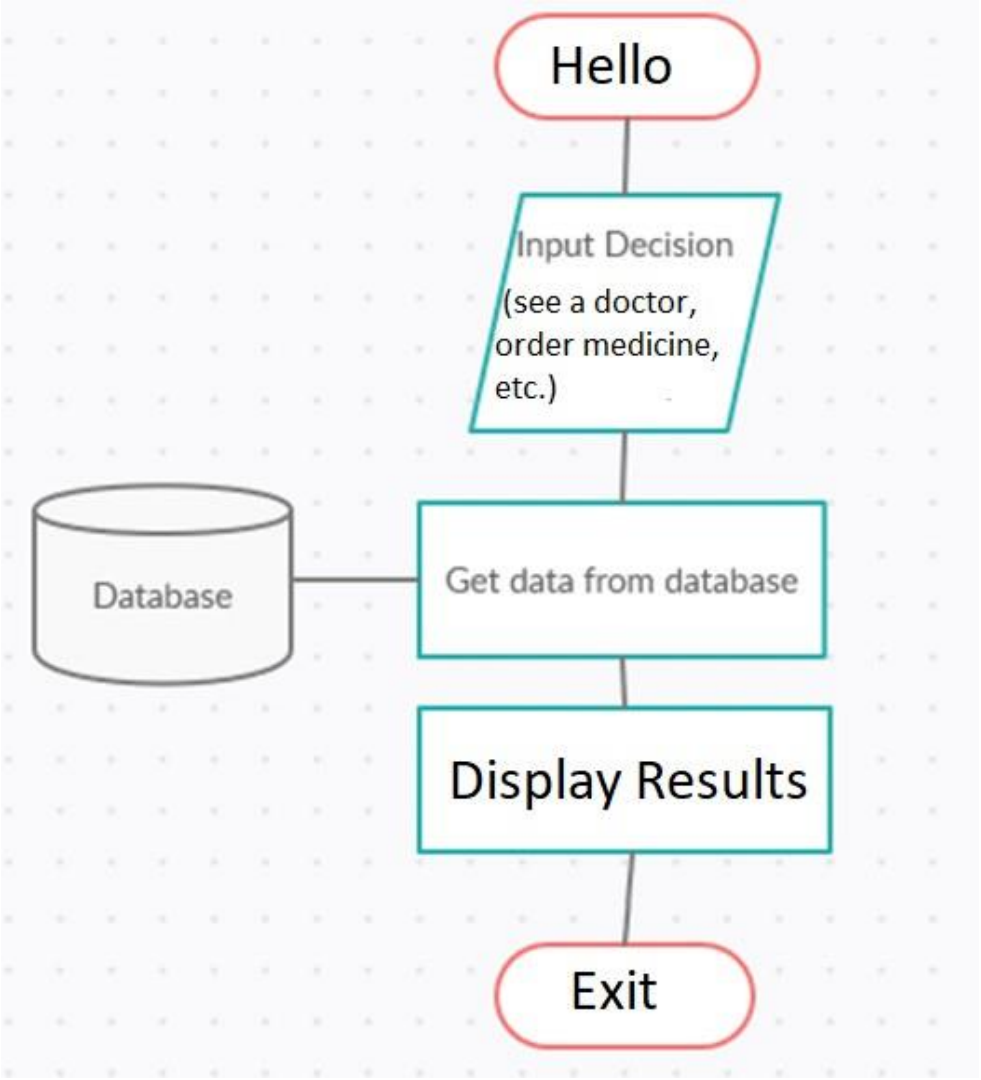
3. System Analysis/ Design

Requirement Specification

Please refer to page 2

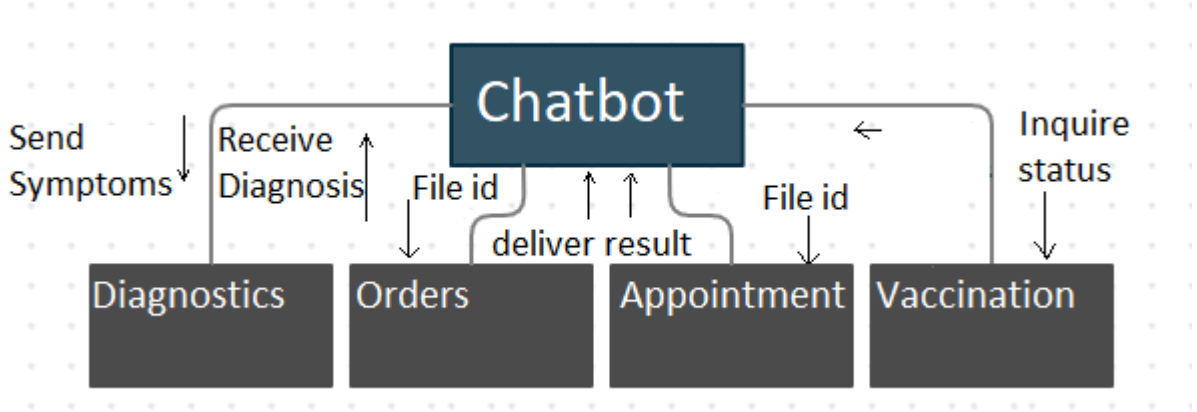
Flowcharts / DFDs

1) Flowchart



Flow Chart- Chatbot

2) DFD



DFD Chatbot

Design and Test Steps / Criteria

Healthbot was designed using Python to program its interactivity. MySQL was used to handle database queries and WhatsApp was used for User interface.

Test Steps for website included:

a) Verification of Input- Under this step, form inputs were tested, and it was checked whether form action was correct and required input field were being used properly. Input fields such as phone number were checked upon that they can only have length of 10 characters.
b) Database verification- Under this step, inputs inserted into the tables was checked. For example: checking of primary keys, whether two users can have same e-mail id, etc.)
c) Testing of SQL queries: Whether correct MySQL queries have been used.

Healthbot was designed using Python with external libraries such as Flask which were tested using print statements to observe results.

Algorithms and Pseudo Code

Patient Registration

a) Algorithm

1) Start
2) Get USER credentials
3) Register credentials to database.
4) Exit.

b) Pseudo Code

```
Input
(Age,Name,address,
phone);

If (Name OR Age OR Address OR PHONE == NULL)
      { Print("Please enter all credentials"); }
Else
{
 Get mysqli_connection ("localhost","root","","hospital");
  Insert into table(Age,Name,Address,Phone);

 Print ("You have been registered");
 }
```

Patient Follow Up

1) Start
2) Get USER file id
3) Apply linear search and get matching credentials.
4) If, no id match, display Wrong ID
5) Else, 'get appointment'
6) Exit

b) Pseudo Code

```
Input ID;
```

```
If (ID == NULL)
      { Print(“Please enter ID”); }
Else
{
 Get mysqli_connection ("localhost","root","","hospital");
 Select credentials where id == id;
                if(Selected rows == 1)
                {
                        Get
                appointment;        }
                Else
                { Print (“Invalid ID”); }

              Return;
              }
```

Medicines Delivery

1) Start
2) Get USER file id
3) Apply linear search and get matching credentials.
4) If, no id match, display Wrong ID
5) Else, ‘get medicines’
6) Exit

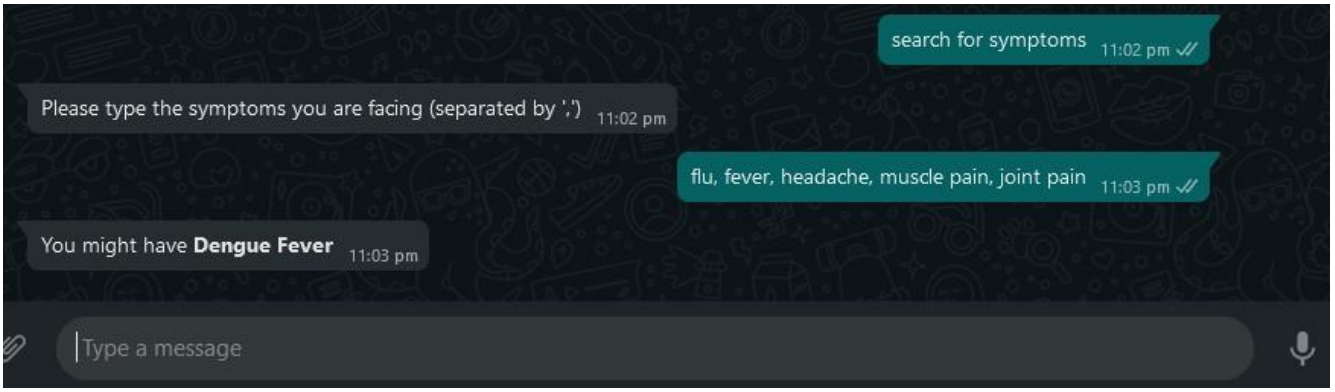
b) Pseudo Code

```
Input ID;

If (ID == NULL)
      { Print(“Please enter ID”); }
Else
{
 Get mysqli_connection ("localhost","root","","hospital");
 Select credentials where id == id;
                if(Selected rows == 1)
                {
                    Get medicines;
                    }
              Else
               { Print (“Invalid ID”); }

Return;
}
```

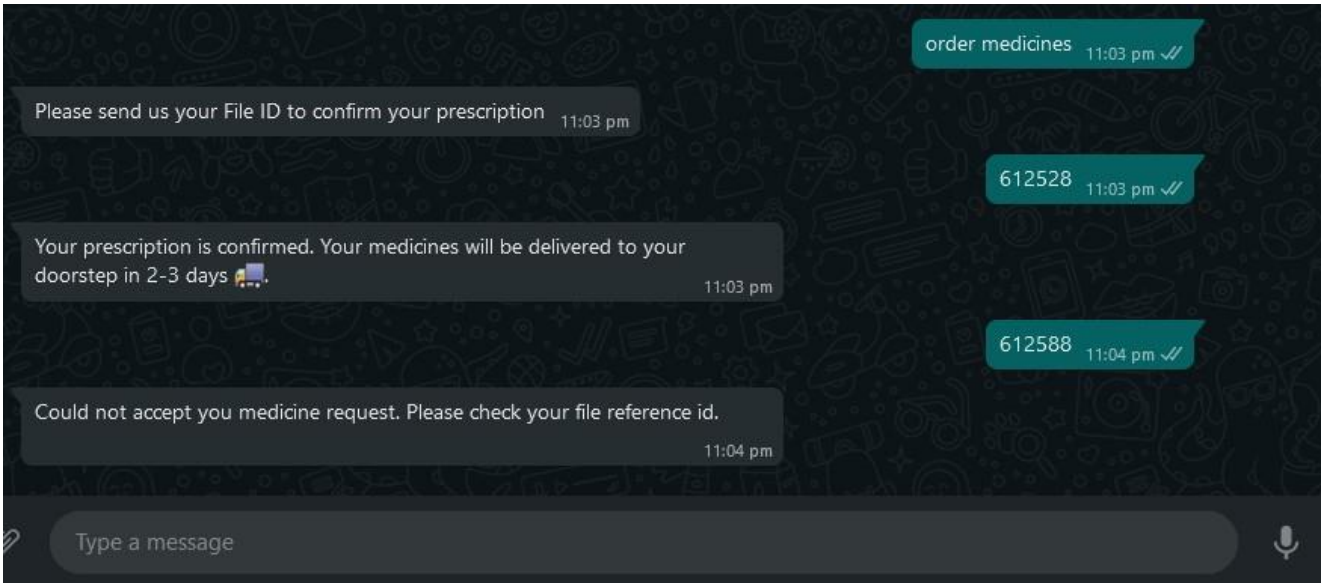
Testing Process

a) For verification of Input- Different types of inputs, such as alphabets in phone numbers, not using ‘,’ in symptoms, etc. were used in order to prevent unwanted inputs by users.
b) For database verification- Insertion of NULL values, duplicate values (for primary key), etc. were inputted to achieve desired stored results in tables.
c) SQLqueries: To check whether correct MySQL queries have been used, output statements were inserted in various parts of code to check out whether each query gets the desired result.

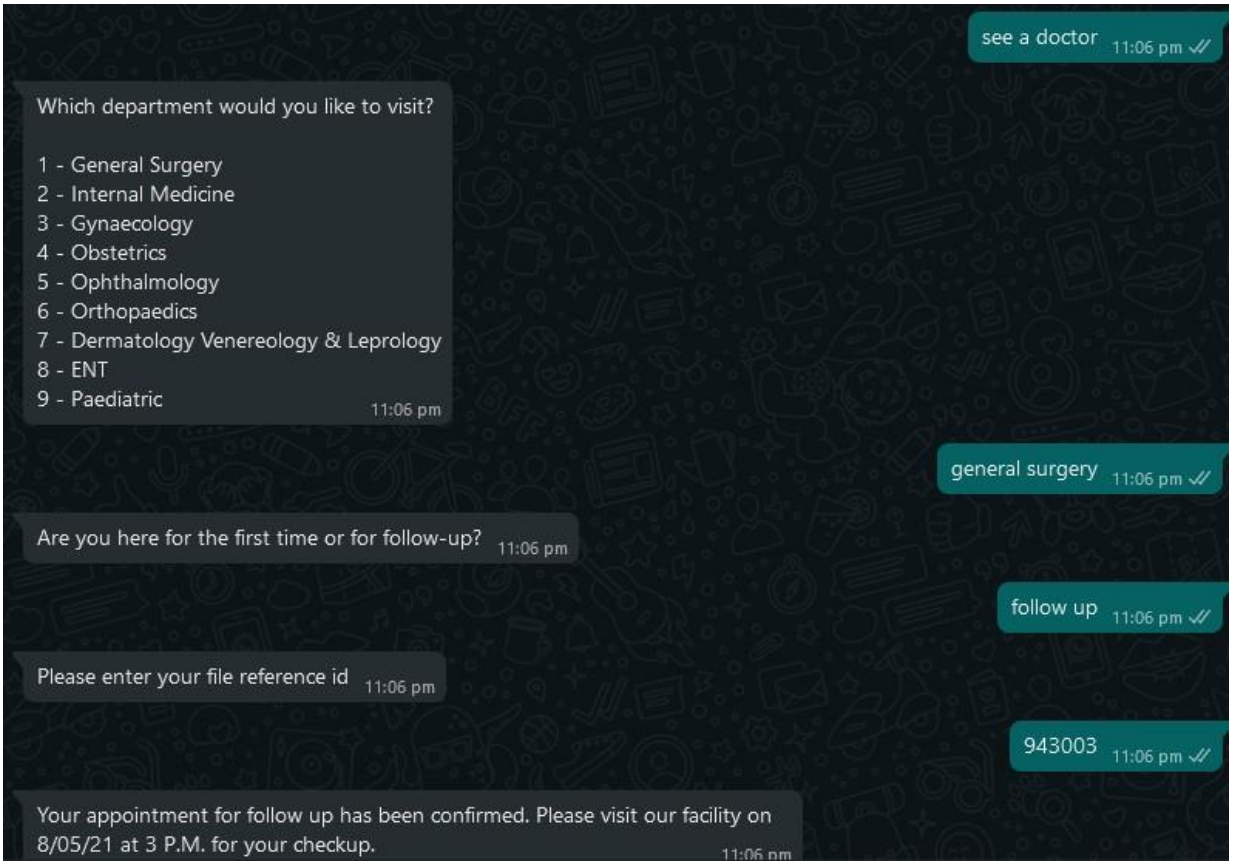
Testing for Whatsapp UI



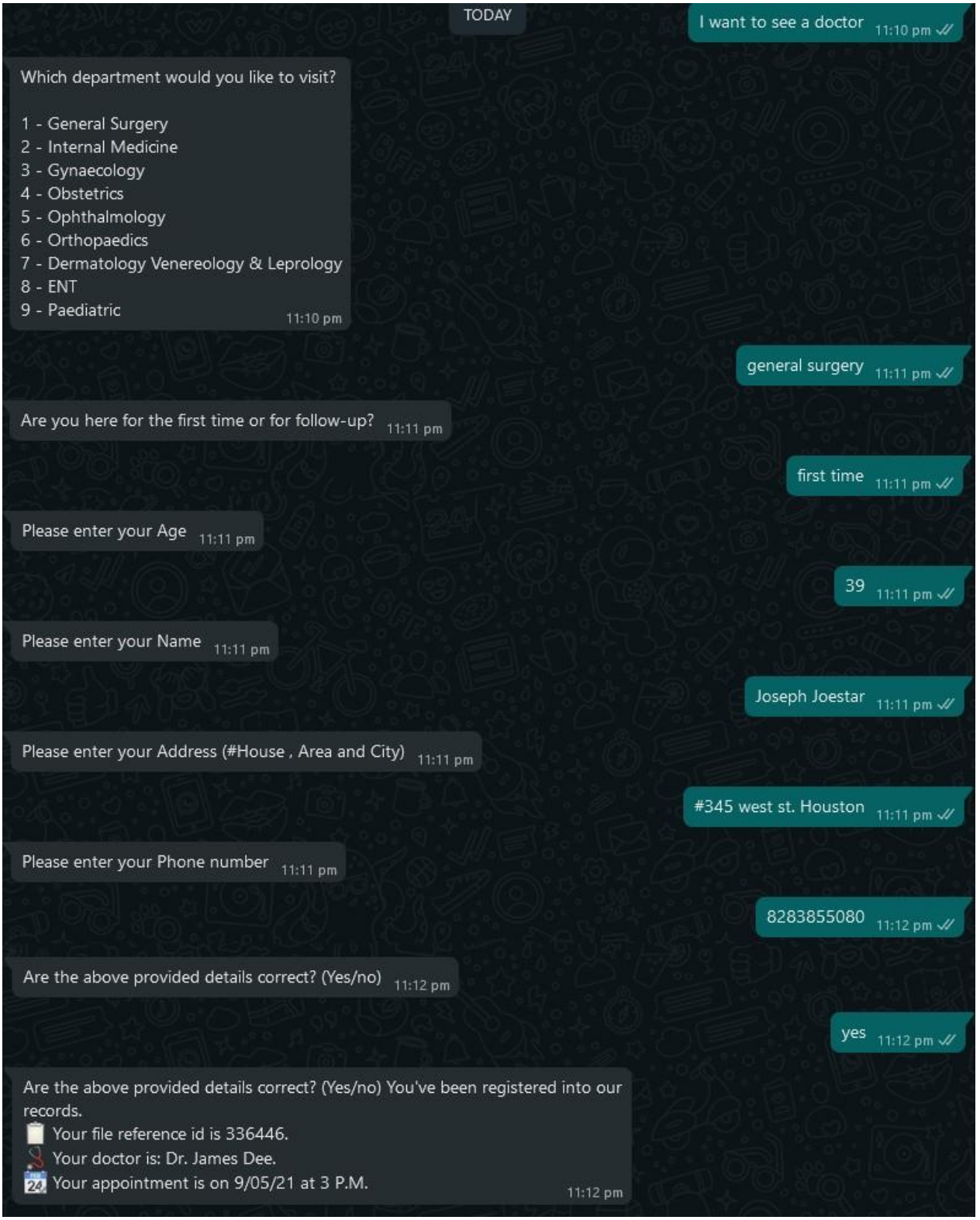
Different results were tested, based on different diseases to check diagnostics



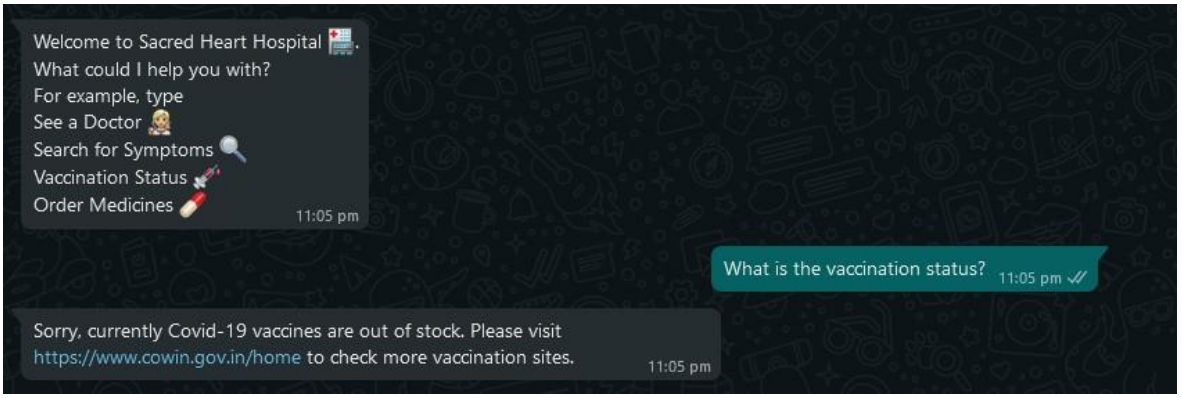
Correct and Incorrect file IDs were given as input to check the functioning of ordering medicines



Similar approach as above was used in follow up examinations for patients



For patients registration, each credential and its corresponding if statement was checked such as type (integer or string), length and format was checked by matching with entries in database

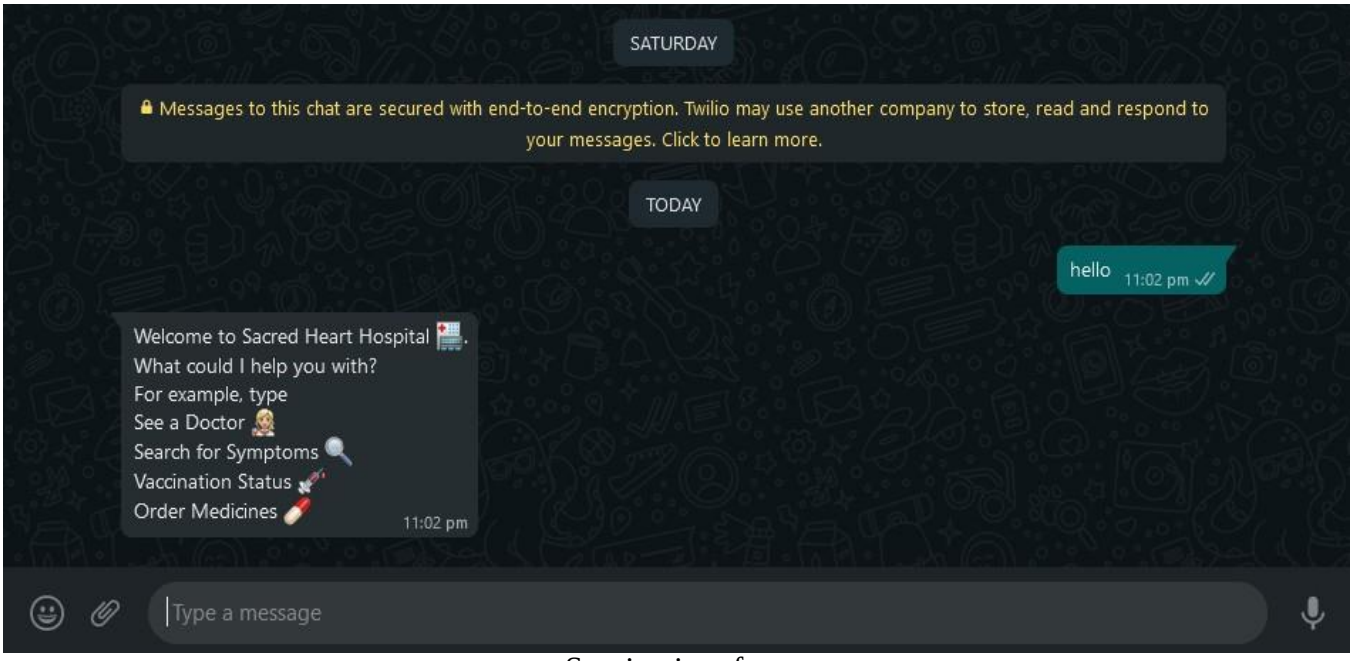


Vaccination status was checked and hyperlink's working was tested by connecting to CoWin's platform

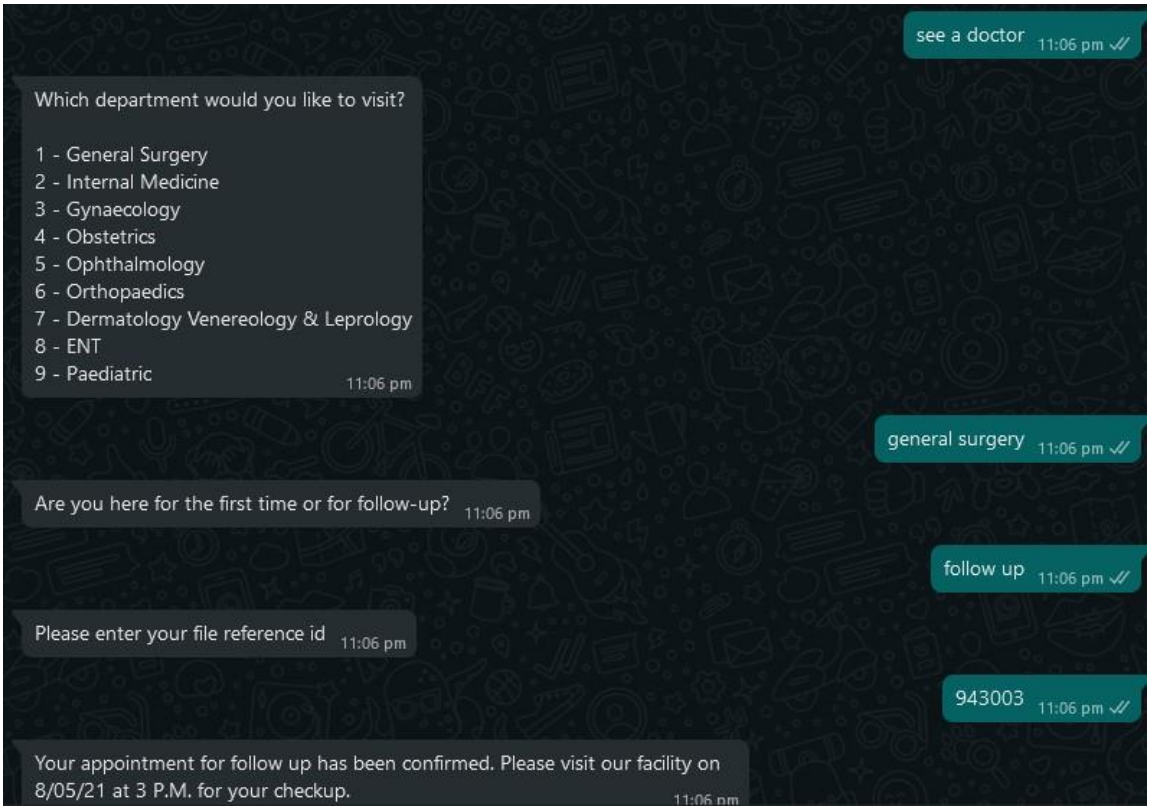
id	name	age	address	file_id	phone
1	el Papito del Rico	34	#6969 420th street, Corona, Panama	56743	8978978970
21	geralt	43	#121 khael morgen	612528	7698972162
22	christian cage	67	#565 montreal	702622	9620269816
24	kratos	37	#4323 crete	329234	6747533129
25	freddy	48	#4232 bikini bottom	462686	8613528386
26	frank	30	#121 ikaros	672647	7029849100
28	jeniffer	46	#565 montreal	943003	9364787858
29	joseph joestar	39	#345 west st. houston	336446	7310169354

The following database was used to check entries

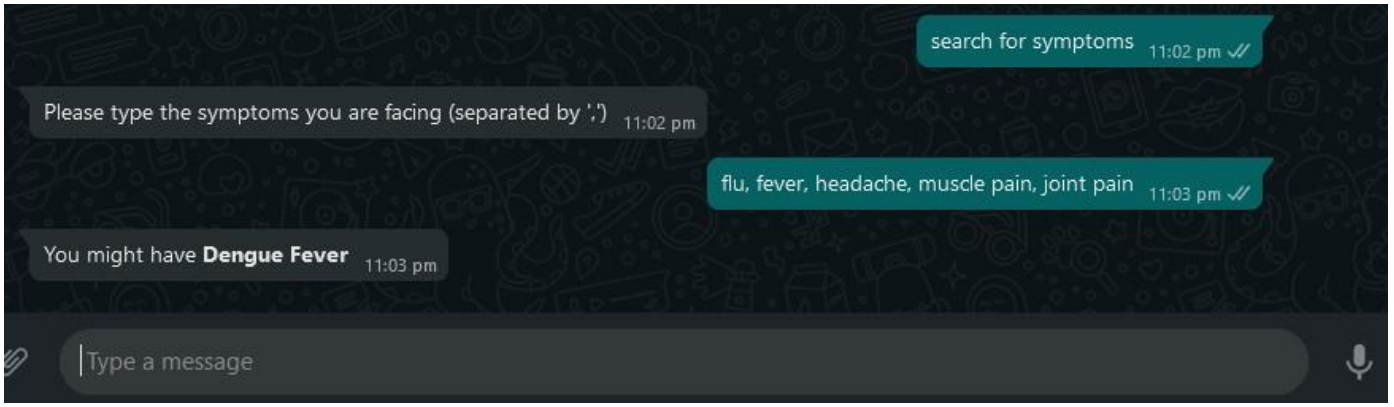
4 Results/Outputs



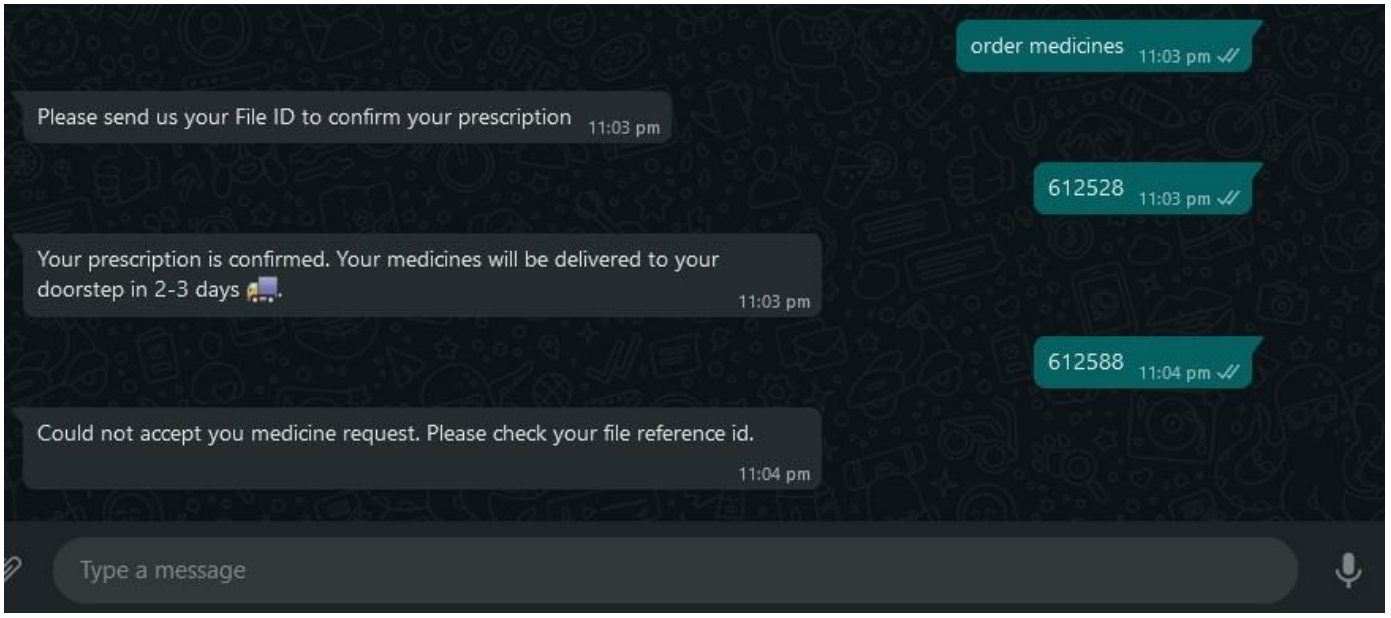
Starting interface



Follow up request output



Diagnostics Output



Output for both correct and incorrect file reference IDs

5. Conclusions / Recommendations

The COVID-19 pandemic has placed enormous stress on both healthcare systems and researchers, and people all across the sector are looking for digital solutions to help ease the pressure. Chatbots are one technology that could speed up how people interact with researchers and find medical information online – and they have already been embraced by several companies in the industry.

Conversational technology (and chatbots) has matured significantly, especially over the past few months as the COVID-19 pandemic has spread. Healthcare companies have been running pilots with chatbot technology for the past few years and many were moving out of pilot stage and into frontline, advanced healthcare workflow automation.

An effective chatbot delivers outcomes by helping patients get something done without requiring human intervention. It must deliver a simple, low friction experience or nobody will use it.

In such situations, chatbots would be most beneficial for scheduling doctor appointments, locating health clinics, or providing medication information.

Healthbot puts the following technologies to use : -

1. **Python** :Python is an interpreted high level general purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
2. **Flask** : 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. 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.
3. **Twilio Whatsapp API** : It is the fastest way to integrate two-way messaging on WhatsApp into your web application.
4. **MySQL**: It provides an open-source relational database management system.
5. **ngrok**: It is used to create a live tunnel, which configures endpoint URLs.

Use of these technologies are recommended as they serve many advantages over the others. Thus, by having a simpler, easy to maintain system, initial success and customer satisfaction can be achieved, and the chatbot itself can serve as framework for future projects.

References

I. MySQL Documentation (https://dev.mysql.com/doc/)
II. Python Documentation (https://docs.python.org/3/)
III. ngrok Documentation (https://ngrok.com/docs)
IV. Twilio Documentation (https://www.twilio.com/docs)
V. Flask Documentation (https://flask.palletsprojects.com/en/1.1.x/)
VI. StackOverflow (https://stackoverflow.com/)
VII. GeekforGeeks (https://www.geeksforgeeks.org/)
VIII. Google Images
IX. YouTube (https://www.youtube.com/)