

DESIGN OF MACHINE LEARNING FOR LUNG TUMOR DETECTION

*Project Phase-I report submitted
in
partial fulfillment of requirement for the award of degree of*

**Bachelor of Engineering
in
Electronics and Telecommunication Engineering**

by

Mr. Yash Wargantiwar (C 31)

Mr. Yash Deshmukh (C 32)

Ms. Asita Orke (C 33)

Ms. Taruna Pakhare (C 34)

Mr. Kartik Kamdi (C 35)

Mr. Bhavesh Kakrotra (C 36)

Guide

Dr. Dilip Choudhary



Department of Electronics and Telecommunications Engineering
G H Raisoni College of Engineering, Nagpur

(An Autonomous Institute Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
NAAC "A+" Grade & NIRF Ranked 139th for 2020.
(Times new Roman 16 /12)

NOV 2021

Declaration

We, hereby declare that the project Phase-I report titled “Design of Machine Learning model for Lung Tumor Detection” submitted herein has been carried out by us towards partial fulfillment of requirement for the award of Degree of Bachelor of Engineering in Electronics and Telecommunication Engineering. The work is original and has not been submitted earlier as a whole or in part for the award of any degree / diploma at this or any other Institution / University.

We also hereby assign to G H Raisoni College of Engineering, Nagpur all rights under copyright that may exist in and to the above work and any revised or expanded derivatives works based on the work as mentioned. Other work copied from references, manuals etc. are disclaimed.

Place Nagpur

Date 22/11/2021

Yash Wargantiwar

Yash Deshmukh

Asita Orke

Taruna Pakhare

Kartik Kamdi

Bhavesh Kakrotra

Certificate

The Project Phase-I report entitled as "**Design of ML model for Lung Tumor Detection**" submitted by **Bhavesh Kakrotra , Asita Orke , Taruna Pakhare , Yash Deshmukh , Yash Wargantiwar , Kartik Kamdi** for the award of Degree of Bachelor of Engineering in Electronics and Telecommunication Engineering has been carried out under our supervision.

The work is comprehensive, complete and fit for evaluation.



RAISONI GROUP
Institute Guide — a vision beyond —
Dr. Dilip Choudhary
Assistant Professor

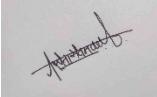
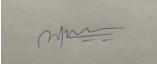
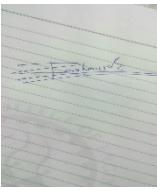
Department of Electronics and Telecommunication Engineering
G H R C E, Nagpur

Head
Department of Electronics and Telecommunication
Engineering
G H R C E, Nagpur

Director
G H R C E, Nagpur

ACKNOWLEDGEMENT

We wish to express my gratitude to our guide, **Dr. Dilip Choudhary**, Department of Electronics and Telecommunication for their active interest, constructive guidance and advice during every stage of this work. Their valuable guidance coupled with active and timely review of our work provided the necessary motivation for us to work on and successfully complete the project.

Sr No .	Name of Candidate	Mobile No.	Email Id	Signature
1	Asita Orke	9518914408	orke_asita.et@ghrce.raisoni.net	
2	Bhavesh Kakrotra	8553007699	kakrotra_bhavesh.et@ghrce.raisoni.net	
3	Kartik Kamdi	7083294355	kamdi_kartik.et@ghrce.raisoni.net	
4	Taruna Pakhare	9545448987	pakhare_taruna.et@ghrce.raisoni.net	
5	Yash Deshmukh	9579129863	deshmukh_yash.et@ghrce.raisoni.net	
6	Yash Wargantiwar	9370352362	wargantiwar_yash.et@ghrce.raisoni.ne	

Abstract

Cloud Healthcare API allows easy and standardized data exchange between healthcare applications and solutions built on amazon Cloud.

The Cloud Healthcare API provides a fully managed, highly scalable, enterprise-grade development environment for building clinical and analytics solutions securely.

It also provides seat availability, contacts of hospital, pharmacy contact, and medical help.

This software provides a single platform at which any person from any part of the world can know the status of the pandemic which the world is facing right now and can know the current situation even of his hometown.

Lung cancer is the second most diagnosed disease among a various kind of cancers which is available in the world.

In India there are roughly 70,275 individual cases in every year are determined to have lung malignancy. However, early identification and treatment can increase the survival rate. Usually Computed Tomography scan imaging is used in the medical field because of its high clarity and low noise.

Only CT scans cannot give proper interpretation to radiologist and the medical practitioner, therefore the Computer Aided Diagnosis system will be extremely useful for radiologists to detect the cancer precisely.

So Lung tumor detection is one advance feature for doctors in order to perform successful surgery.

LIST OF FIGURES

Sr. No.	Name of Figures	Page No
5.1	Flowchart of Chatbot and Tumor detection	19
5.2	Dashboard of Chatbot	20
5.3	CoWin app site inside the chatbot	21
6.1	Vaccination app site inside the chatbot	22
6.2	Diabetes checker in chatbot	23
6.3	BMI Checker in chatbot	24
6.4	Hospital Contact list in chatbot	25
6.5	Database for 3D tumor detection	26
6.6	Database of the Lung Tumor	27

Index

Sr.No.	Content	Page Number
1.	Introduction <ul style="list-style-type: none"> 1.1 Project Summary 1.2 Features 1.3 Software Required 	8-9
2.	Literature Review	10
3.	Methodology	11-15
4.	Platform Used <ul style="list-style-type: none"> 4.1. A Cloud Service i.e. AWS, GCP etc... 4.2. Apache web server i.e httpd 4.3. ssh Client i.e Putty 4.4. CGI (Common Gateway Interface) 4.5. A Linux instance i.e RHEL8 4.6. Python3 4.7. HTML 4.8. CSS 4.9. Deep learning 4.10. MATLAB 4.11. 3D Slicer 	16-18
5.	Design and Implementation	19
6.	Summery and Result	20-26
7.	Future Scopes	32
8.	References	33

INTRODUCTION

1.1 Project Summary

The majority of crops are rain-fed, with just about 45 percent of the area irrigated. It is estimated that roughly 55% of India's overall population lives in poverty. Agriculture is reliant on rain. As a result of the intense labour Changes in meteorological conditions, crop losses, shortages and excessive labour costs Pests, a lack of cash and agricultural supplies, and waste inputs, and a dreadful support price structure These are all of the issues that exist.

Crop yields have been improved using a variety of approaches including irrigation, pesticide use, and soil PH measurement. Irrigating fields using floods was an ancient practise. However, evaporation, infiltration of cultivated fields, and transpiration combined to lose half of the water provided. Drip and sprinkler irrigation developed as a result of excessive water waste during flood irrigation. Salt deposition at the nozzle, on the other hand, caused the pipes to become clogged with these procedures. In addition, the use of fertilisers and pesticides was expanded in order to improve the land's fertility and thus productivity. Manual fertiliser and pesticide spraying can harm the neurological system and lead to disorders that are lethal.

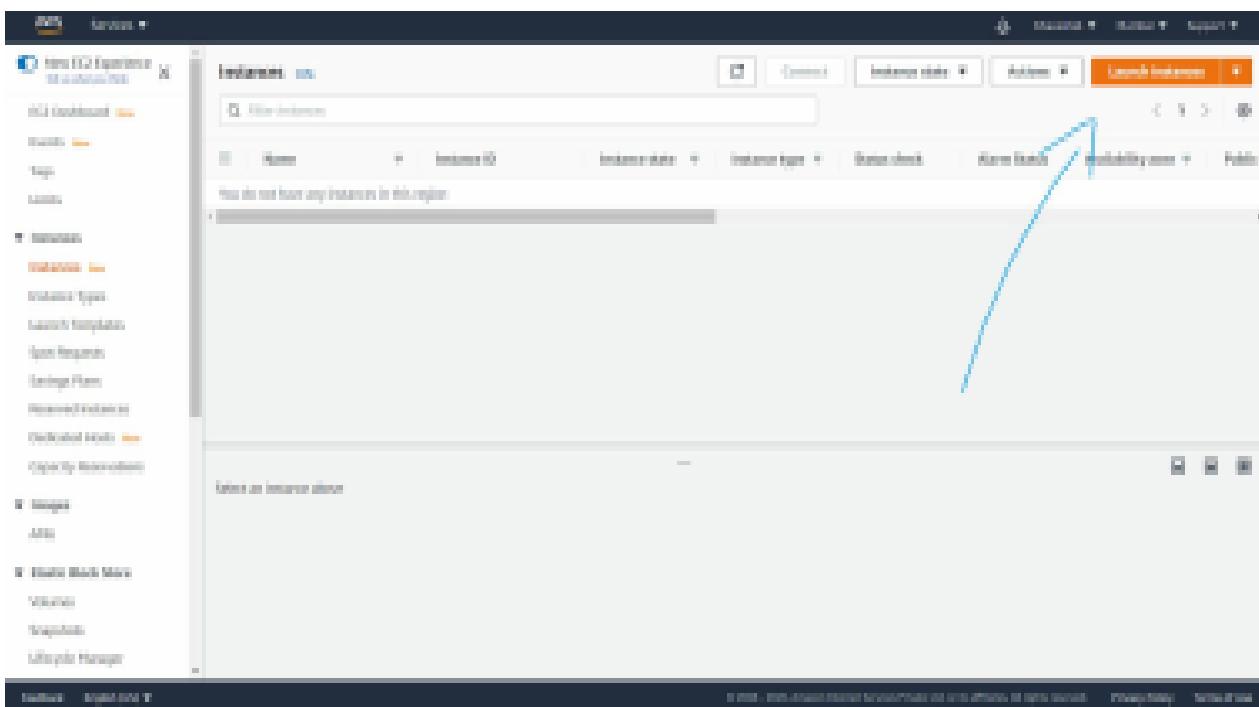
The development of smart irrigation has been based on the solution to these limitations. Smart irrigation allows you to irrigate the field, determine the PH value of the soil, and send the crop status to the farmer via GPS. Smart irrigation was used to build effective agriculture solutions. The extra space that smart irrigation takes up is mostly employed for hardware installations that cover bigger amounts of farmed land. The development of the UAV was based on this limitation.

UAV or unmanned aerial vehicle, as the name suggests, is an aircraft that does not require a pilot to operate it. The plane is usually operated using a remote control from the ground. It is also programmed to perform certain tasks using dynamic automation systems which the operator on the ground just needs to monitor to prevent the untoward incident from happening. A quadrotor or quadcopter is unique in the sense that the UAV performs Vertical Take-Off and Landing(VTOL). The main feature of an UAV is that individual motor control stabilizes the UAV instead of cyclic control.

LITERATURE REVIEW

- **Quadrotor design and control with application to autonomous flight:**
 - . A drone can be used to survey crops and collect data in the agriculture industry. As a result, a drone with a good camera will be necessary. Near real-time monitoring is possible with this type of UAV.
 - The batteries in multi-rotor systems endure for a shorter time. Fixed-wing drones, on the other hand, may spend significantly more time in the air. As a result of this feature, they are excellent tools for large-scale agricultural operators.
- **A quadrotor hovering robotic platform with a wireless multivariable control scheme:**
 - As a result of the problems of controlled flying, certain Quadrotor researchers faced a difficulty.
 - In this paper, we create an abstract flexible autonomous robot that can be controlled at many levels and give users from various backgrounds access to the features they want to utilise.

METHODOLOGY



1. Logging into amazon cloud console as a root user and selecting Launch instance from the ec2 dashboard.



2. Select a Linux image for our instance.

Advanced Details

Metadata accessible	Enabled
Metadata version	V1 and V2 (token optional)
Metadata token response hop limit	1
User data	<input checked="" type="radio"/> As text <input type="radio"/> As file <input type="checkbox"/> Input is already base64 encoded
<pre>#!/bin/bash sudo yum install python3 sudo yum install httpd sudo systemctl enable httpd</pre>	

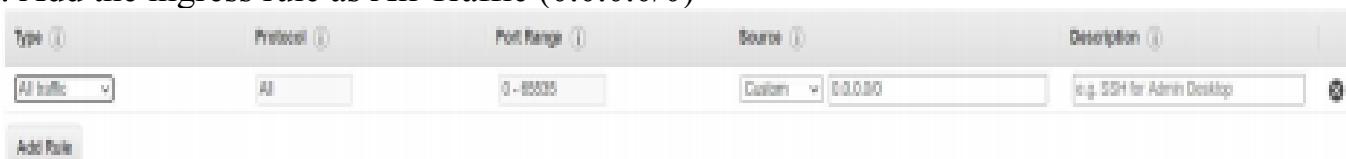
3. In the configure instance details tab we now type this into the user data field to run this

script after the launch.

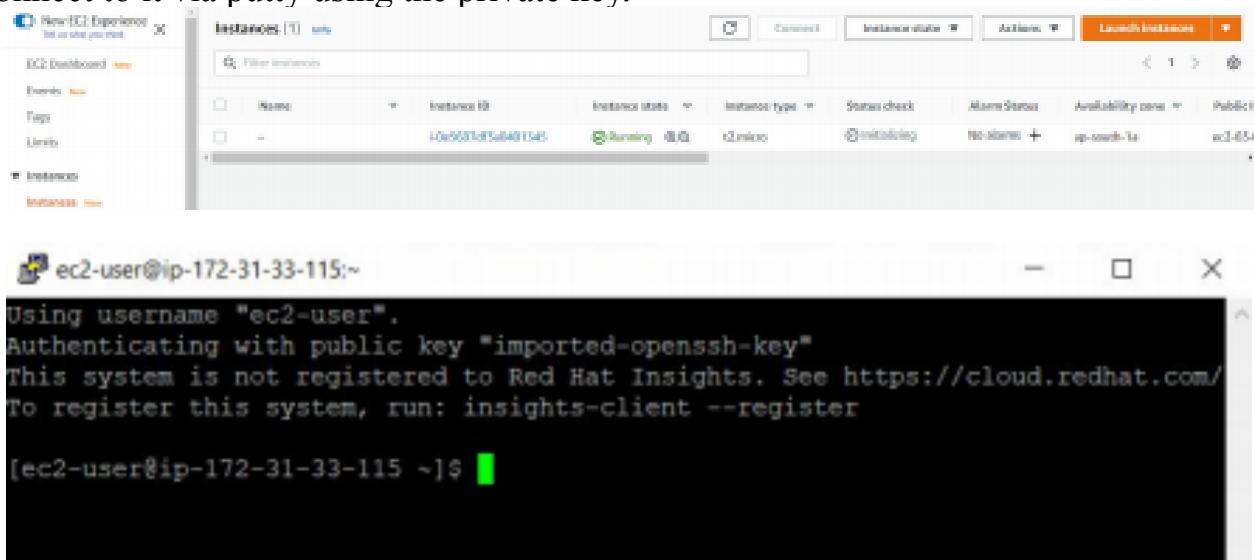


4. Set volume to 10 GiB.

5. Add the ingress rule as All Traffic (0.0.0.0/0)



7. Connect to it via putty using the private key.



8. [root@ip-172-31-33-115 ec2-user]# systemctl status httpd

```
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-10-27 01:36:45 UTC; 3s ago
     Docs: man:httpd.service(8)
 Main PID: 13139 (httpd)
    Status: "Started, listening on: port 80"
       Tasks: 213 (limit: 4936)
      Memory: 32.9M
     CGroup: /system.slice/httpd.service
             └─13139 /usr/sbin/httpd -DFOREGROUND
                 ├─13140 /usr/sbin/httpd -DFOREGROUND
                 ├─13141 /usr/sbin/httpd -DFOREGROUND
                 ├─13142 /usr/sbin/httpd -DFOREGROUND
                 ├─13143 /usr/sbin/httpd -DFOREGROUND
```

Check if our httpd apache server is running

9. Now we move to /var/www/html directory to create a webpage.

```
[ec2-user@ip-172-31-33-115 ~]$ sudo su  
[root@ip-172-31-33-115 ec2-user]# cd /var/www  
[root@ip-172-31-33-115 www]# ls  
cgi-bin  html  
[root@ip-172-31-33-115 www]# cd html  
[root@ip-172-31-33-115 html]# ls  
[root@ip-172-31-33-115 html]#
```

10. In this folder we create a new file “Chatbot” and run it

← → ⌂ Net secure | 65.0.93.53/chatbot



Hi! I'm Anna, your Chatbot.
Say hello!

11. We create the css file and a webpage that takes height and weight as an input.

```
[root@ip-172-31-33-115 html]# vim stylesheet.css  
[root@ip-172-31-33-115 html]# vim bmi  
[root@ip-172-31-33-115 html]# ls  
bmi  chatbot  stylesheet.css  
[root@ip-172-31-33-115 html]#
```

12. Now we create our API in /var/www/cgi-bin folder. We now create an API using python and call it hello.py

```

#!/usr/bin/env python3

print("content-type: text/html\n")

import subprocess as sp
import cgi

form = cgi.FieldStorage()
htt = form.getvalue("htt")
wtt = form.getvalue("wtt")

print("<!DOCTYPE html>")
print("<head><title>Anna, Your chatbot!</title>")
print("<link rel=\"stylesheet\" href=\"/stylesheet.css\">")
print("</head><body>")
print("<form action=\"/chatbot\">")
print("<img src=\"https://cdn2.iconfinder.com/data/icons/healthcare-medical-app/64/medical-robot-512.png\" style=\"width: 226px; height: 226px\">")
print("")
bmi = float(wtt)/((float(htt))**2)
print("<p>Your BMI is {}<p>".format(bmi))
if bmi < 18:
    print("<p>Consume more calories!<p>")
elif bmi > 25:
    print("Consume lesser calories!")
else:
    print("You are fit!")
print("<input type=\"submit\" value=\"Back\"></form>")
~
```

13. We also create another API and call it bmi.py

14. Lastly we give execute permissions to all files in our cgi-bin directory.

Using username "ec2-user".

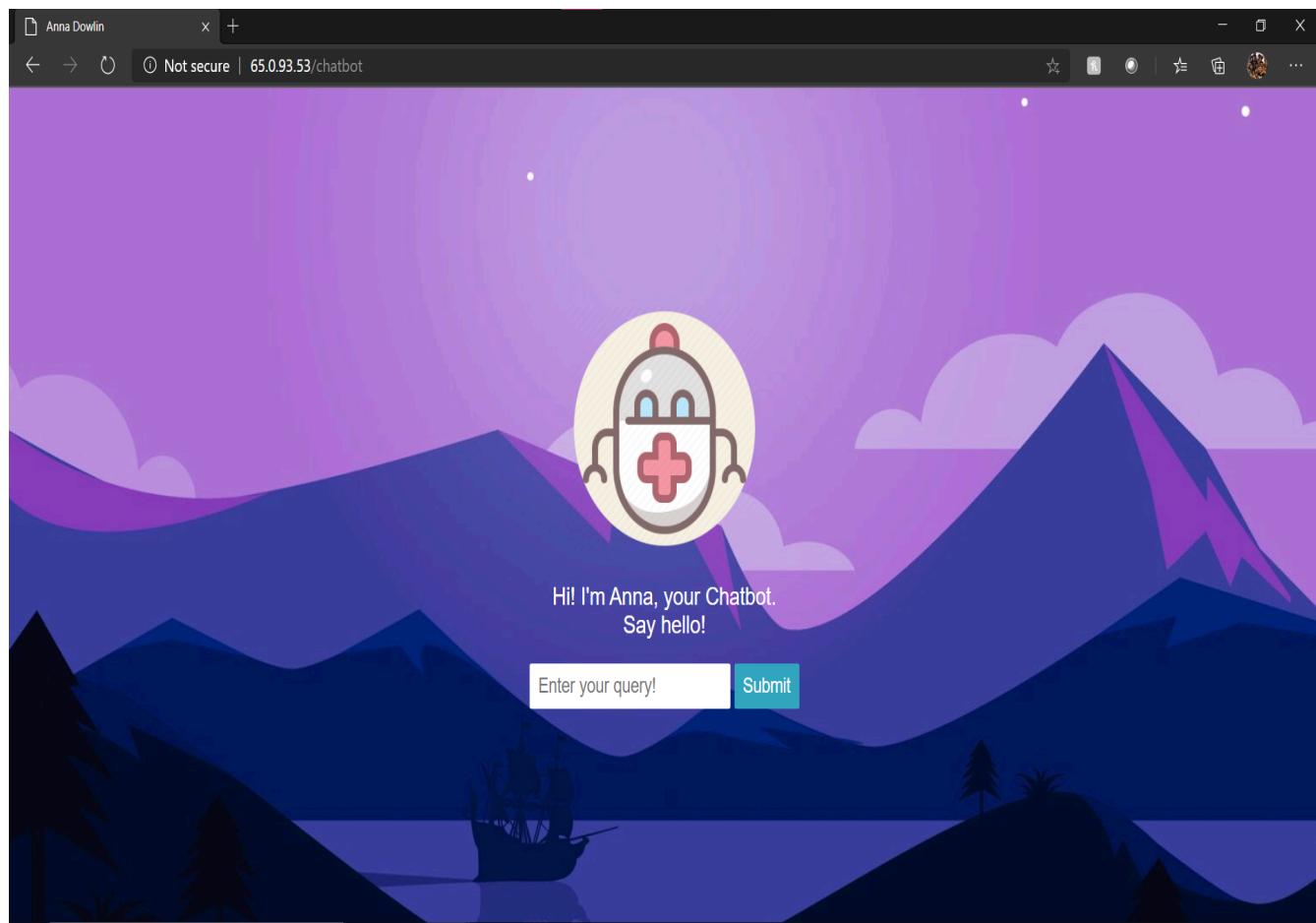
Authenticating with public key "imported-openssh-key"

This system is not registered to Red Hat Insights. See <https://cloud.redhat.com/>
To register this system, run: insights-client --register

Last login: Tue Oct 27 02:19:22 2020 from 106.193.218.97

```
[ec2-user@ip-172-31-33-115 ~]$ sudo su
[root@ip-172-31-33-115 ec2-user]# cd /var/www/cgi-bin
[root@ip-172-31-33-115 cgi-bin]# vim hello.py
[root@ip-172-31-33-115 cgi-bin]# vim bmi.py
[root@ip-172-31-33-115 cgi-bin]# chmod +x *
[root@ip-172-31-33-115 cgi-bin]#
```

15. We now check if our chatbot is running.



PLATFORM USED

Cloud Service_AWS

- Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.
- These cloud computing web services provide a variety of basic abstract technical infrastructure and distributed computing building blocks and tools. One of these services is Amazon Elastic Compute Cloud (EC2), which allows users to have at their disposal a virtual cluster of computers, available all the time, through the Internet.



- AWS's virtual computers emulate most of the attributes of a real computer, including hardware central processing units (CPUs) and graphics processing units (GPUs) for processing; local/RAM memory; hard-disk/SSD storage; a choice of operating systems; networking; and pre-loaded application software such as web servers, databases, and customer relationship management (CRM).
- The AWS technology is implemented at server farms throughout the world, and maintained by the Amazon subsidiary.
- Fees are based on a combination of usage (known as a "Pay-as-you-go" model), hardware, operating system, software, or networking features chosen by the subscriber required availability, redundancy, security, and service options. Subscribers can pay for a single virtual AWS computer, a dedicated physical computer, or clusters of either.

- As part of the subscription agreement, Amazon provides security for subscribers' systems. AWS operates from many global geographical regions including 6 in North America.
- Amazon markets AWS to subscribers as a way of obtaining large scale computing capacity more quickly and cheaply than building an actual physical server farm.
- All services are billed based on usage, but each service measures usage in varying ways. As of 2017, AWS owns 33% of all cloud (IaaS, PaaS) while the next two competitors Microsoft Azure and Google Cloud have 18%, and 9% respectively, according to Synergy Group.

Common Gateway Interface (CGI) **Common Gateway Interface**



- In computing, Common Gateway Interface (CGI) is an interface specification that enables web servers to execute an external program, typically to process user requests. Such programs are often written in a scripting language and are commonly referred to as CGI scripts, but they may include compiled programs. A typical use case occurs when a Web user submits a Web form on a web page that uses CGI. The form's data is sent to the Web server within an HTTP request with a URL denoting a CGI script.

Python3

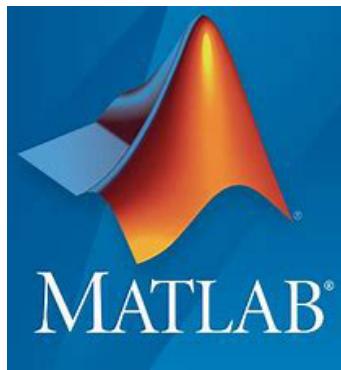
- Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

- Python is named after a TV Show called "Monty Python's Flying Circus" and not after Python-the snake. Python 3.0 was released in 2008. Although this version is supposed to be backward incompatible, later on many of its important features have been backported to be compatible with version 2.7.



MATLAB

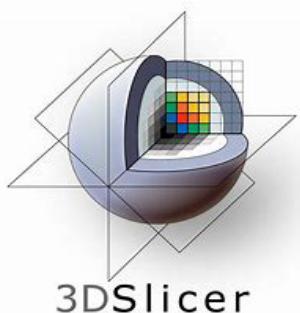
- MATLAB (an abbreviation of "MATrix LABoratory") is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks.
- MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.
- Although MATLAB is intended primarily for numeric computing, an optional toolbox uses the MuPAD symbolic engine allowing access to symbolic computing abilities.



- An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

3D Slicer

- 3D Slicer is a free open source software (BSD-style license) that is a flexible, modular platform for image analysis and visualization. 3D Slicer is extended to enable development of both interactive and batch processing tools for a variety of applications.
- 3D Slicer provides image registration, processing of DTI (diffusion tractography), an interface to external devices for image guidance support, and GPU-enabled volume rendering, among other capabilities. 3D Slicer has a modular organization that allows the addition of new functionality and provides a number of generic features not available in competing tools.



- The interactive visualization capabilities of 3D Slicer include the ability to display arbitrarily oriented image slices, build surface models from image labels, and hardware accelerated volume rendering.

DESGINE / IMPLEMENTATION

Flowchart of Lung Tumor Detection

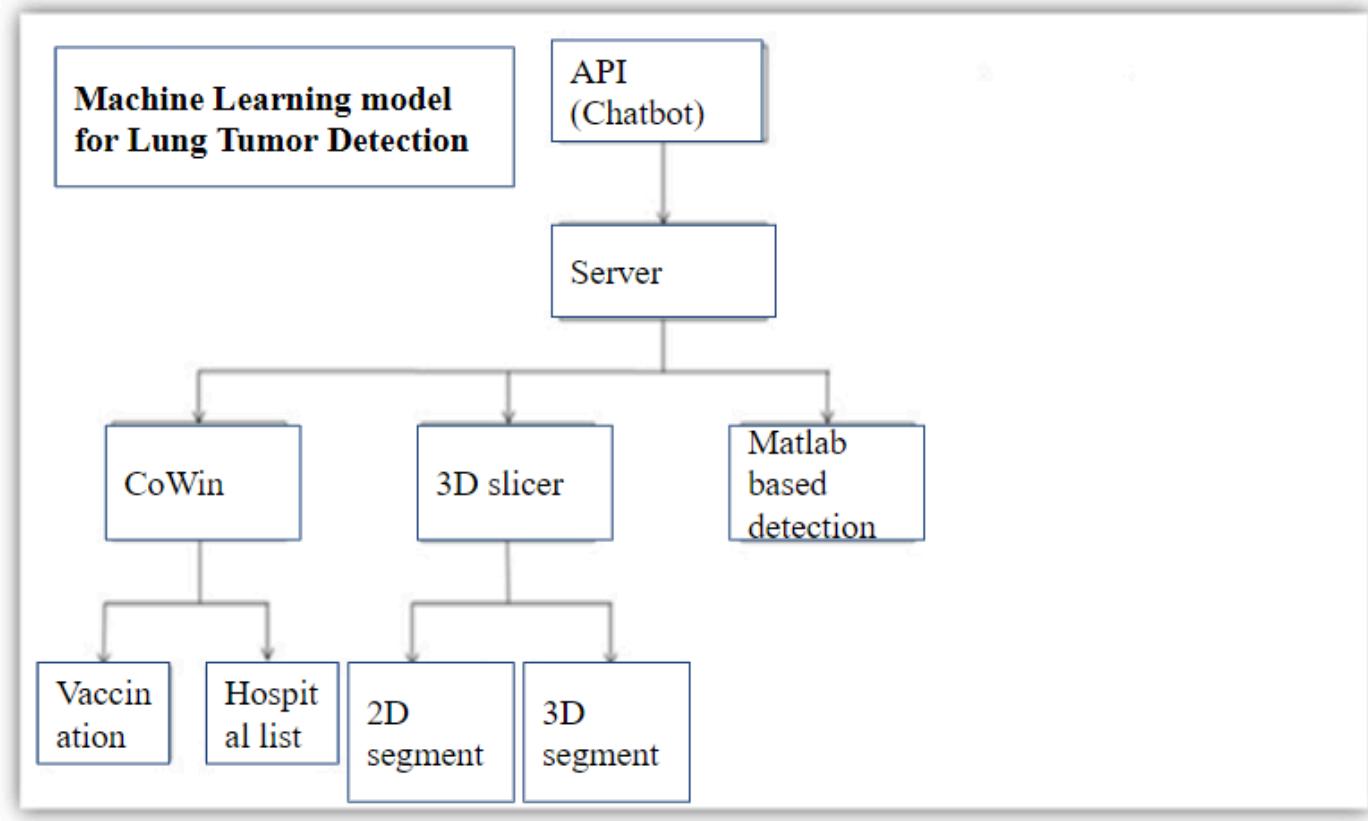
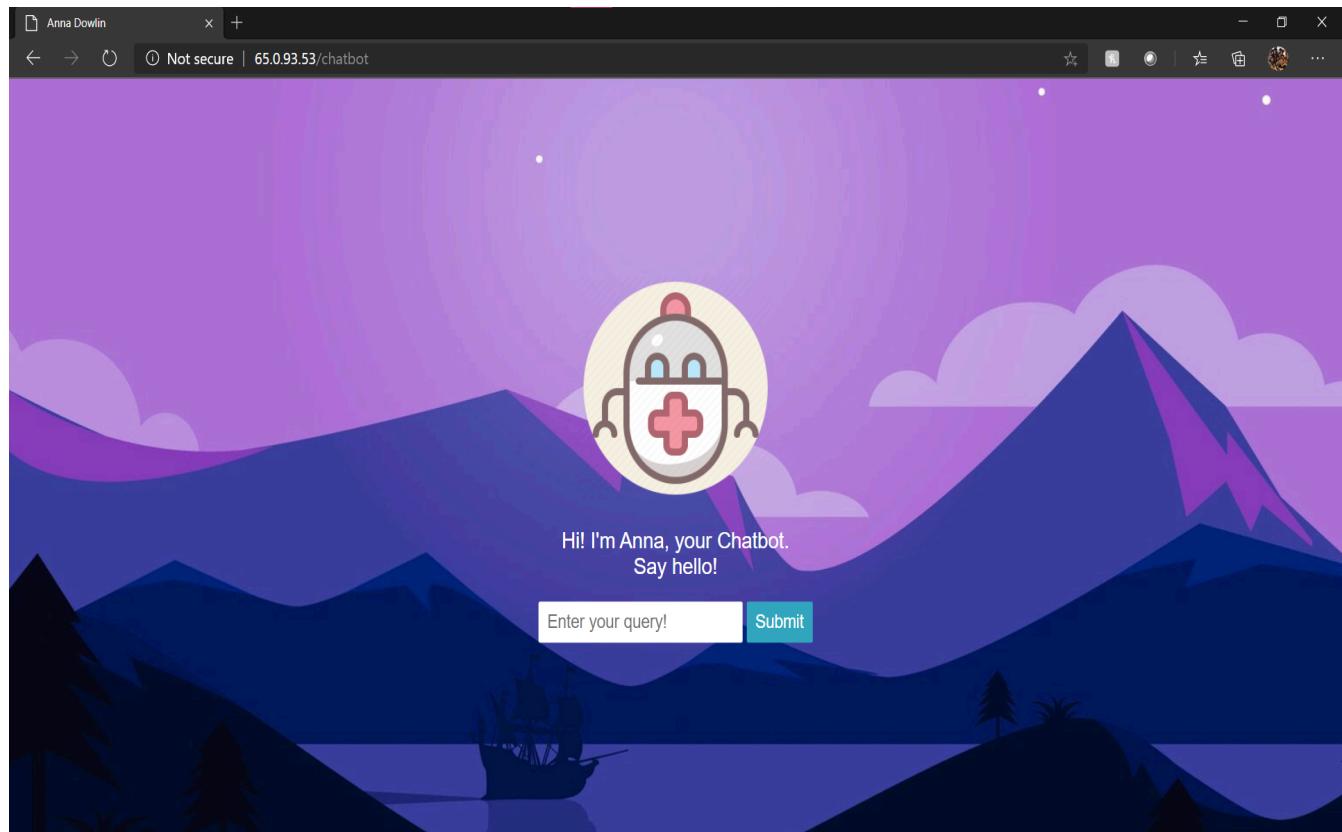


Fig 5.2: Flowchart of Lung Tumor Detection

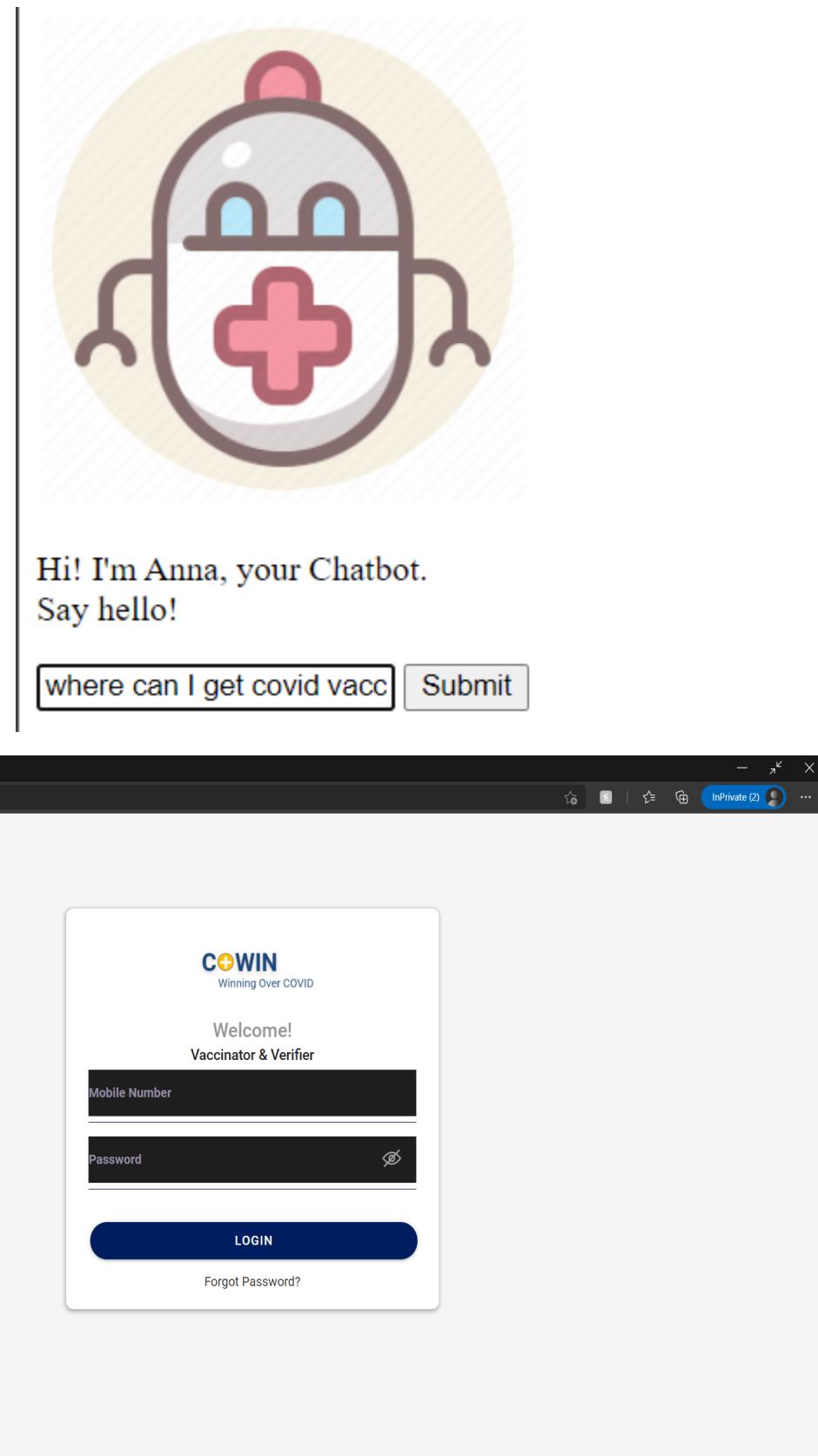
- The Healthcare API is created using AWS. The API contains a chatbot server which gives access to all the other features of Chatbot.
- After accessing the server search engine, it would take you to the Cowin app which provides either vaccine registration facility or Hospital contact list facility.
- The server takes you to 3D lung segmentation for the doctor to perform surgery using the segmentation process.
- The final access is detection of tumors by providing MRI images. This detection is provided to patients in order to scan their MRI and detect the tumor.

Server

Fig 5.3: Dashboard of Chatbot



SUMMARY OF RESULT



Cowin application in Chatbot

Fig.5.3. Cowin application inside Chatbot

Covid-19 vaccines +

https://www.who.int/emergencies/diseases/novel-coronavirus-2019/covid-19-vaccines

InPrivate (2) ...

 World Health Organization

Health Topics Countries Newsroom Emergencies Data About Us

Home / Emergencies / Diseases / Coronavirus disease (COVID-19) / COVID-19 vaccines



COVID-19 vaccines

Vaccination application in Chatbot

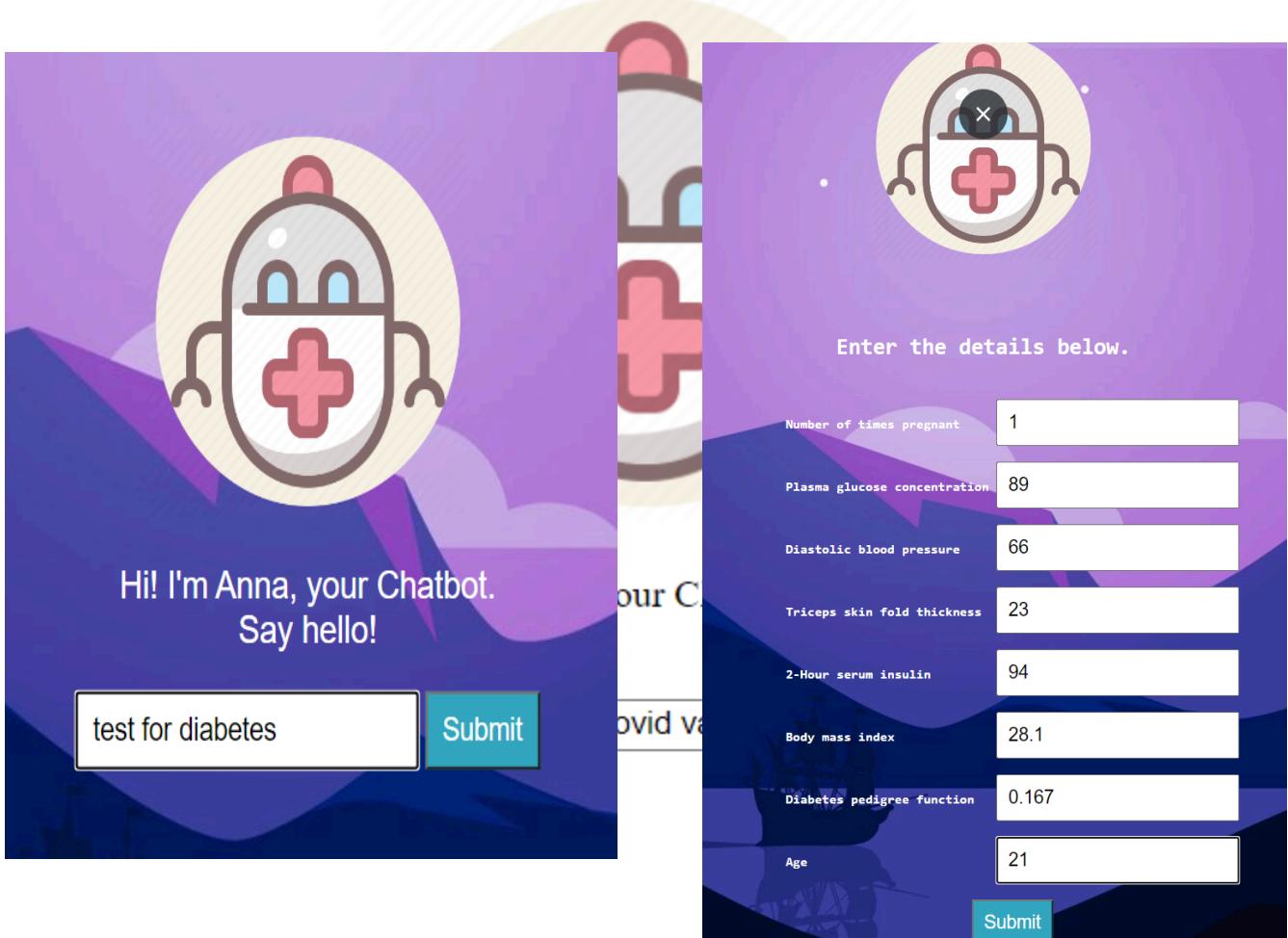


Fig.6.1. Vaccination application site inside Chatbot Diabetes Checker in Chatbot

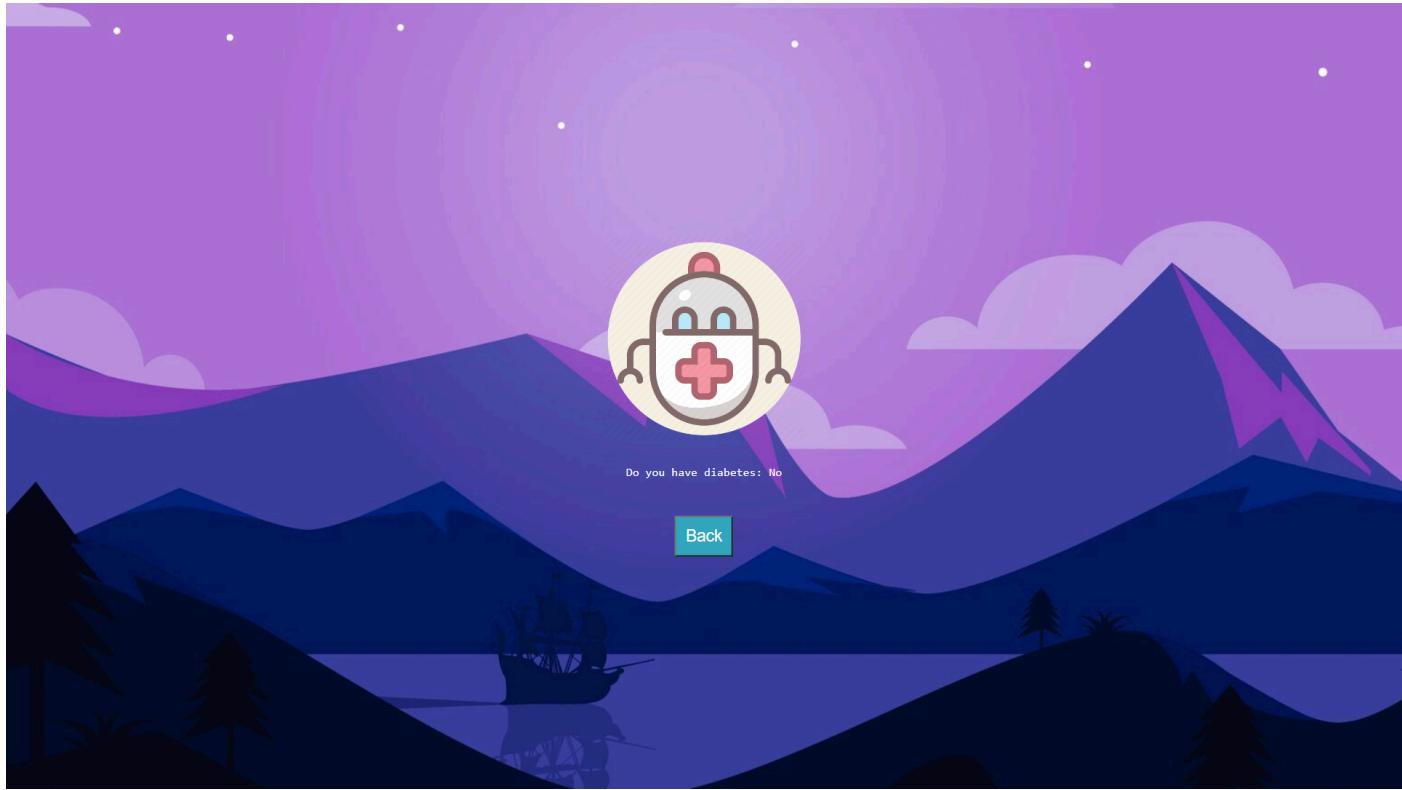
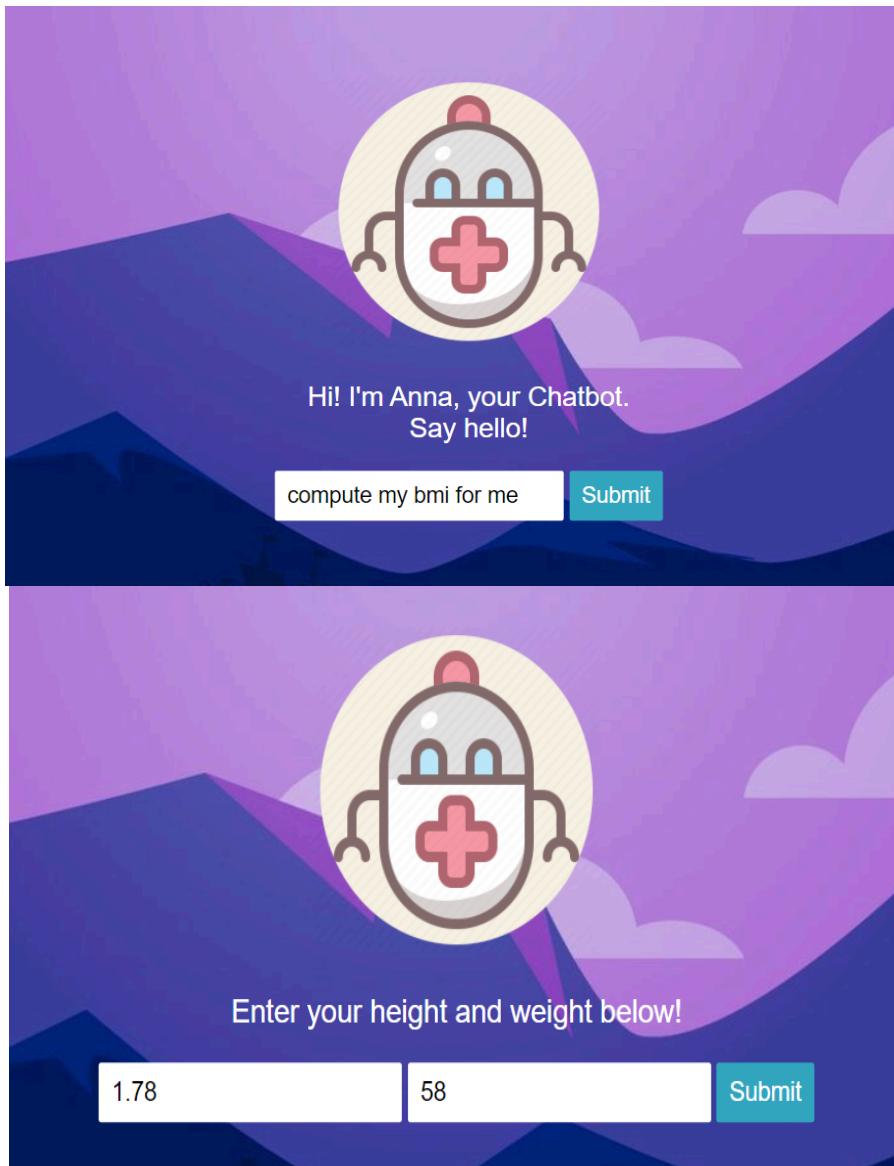


Fig.6.2. Diabetes Checker in Chatbot



BMI Checker in Chatbot

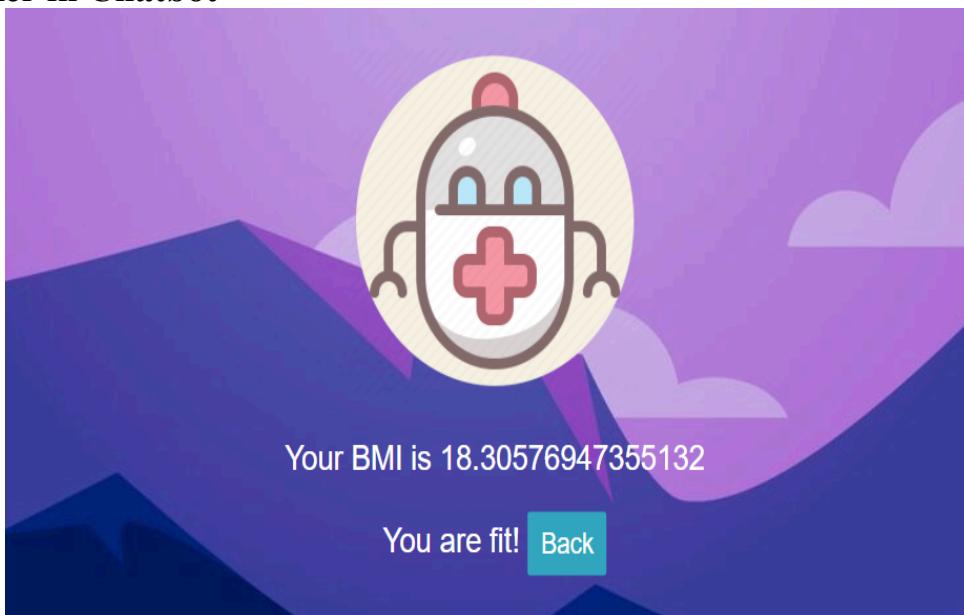


Fig.6.3. BMI Checker in Chatbot

Hospital Contact list in Chatbot

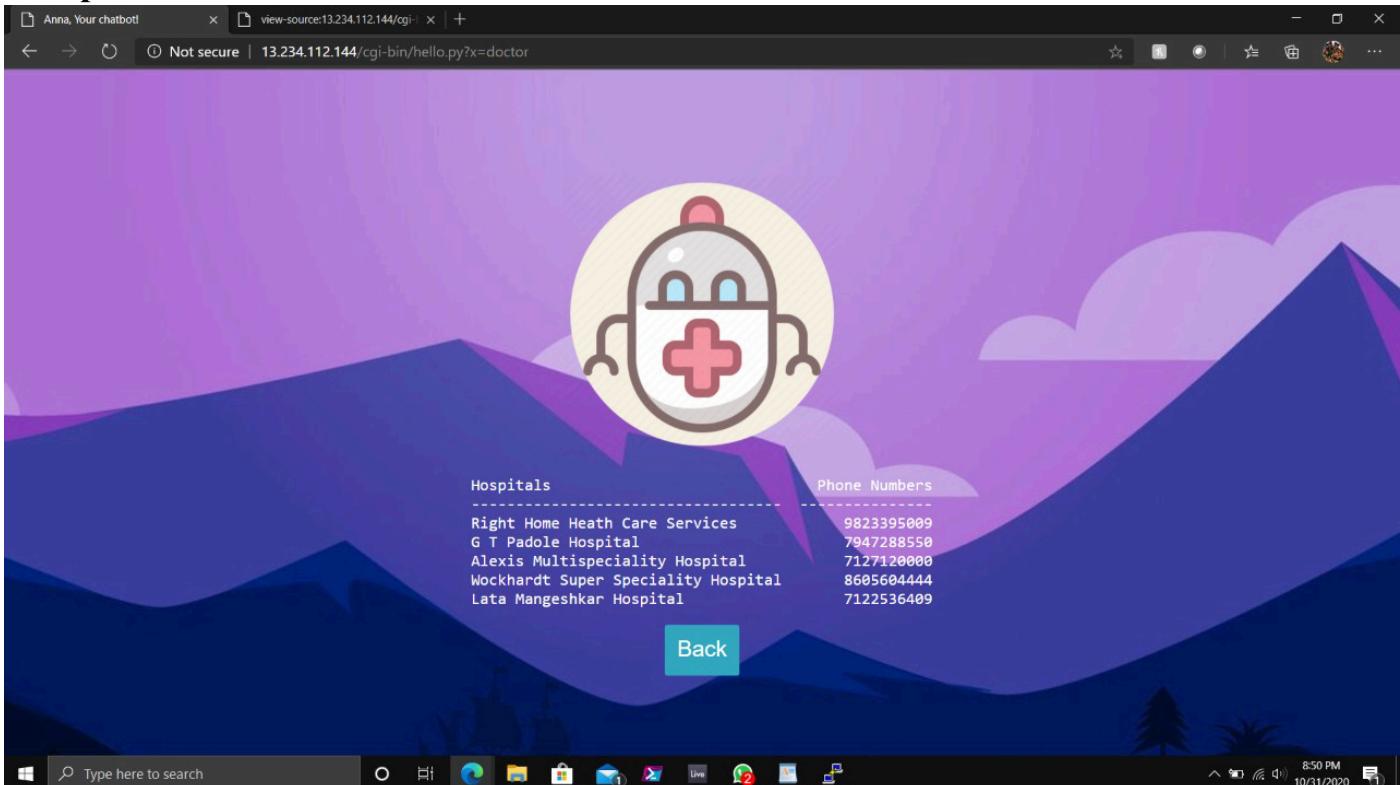


Fig.6.4. Hospital Contact list



Fig.6.4.1. Covid Update

Database for 3D Tumor Detection

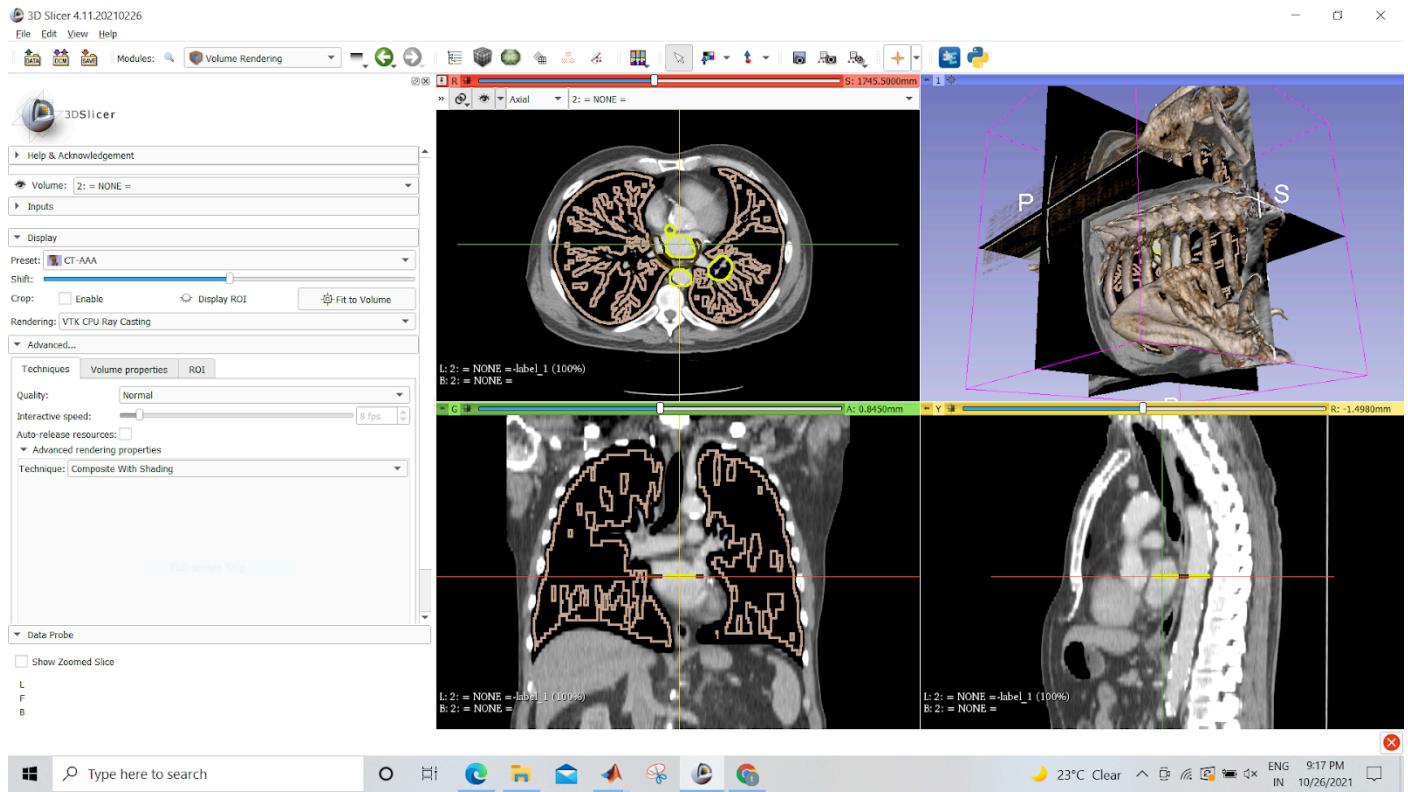


Fig.6.5. 3D Tumor Detection

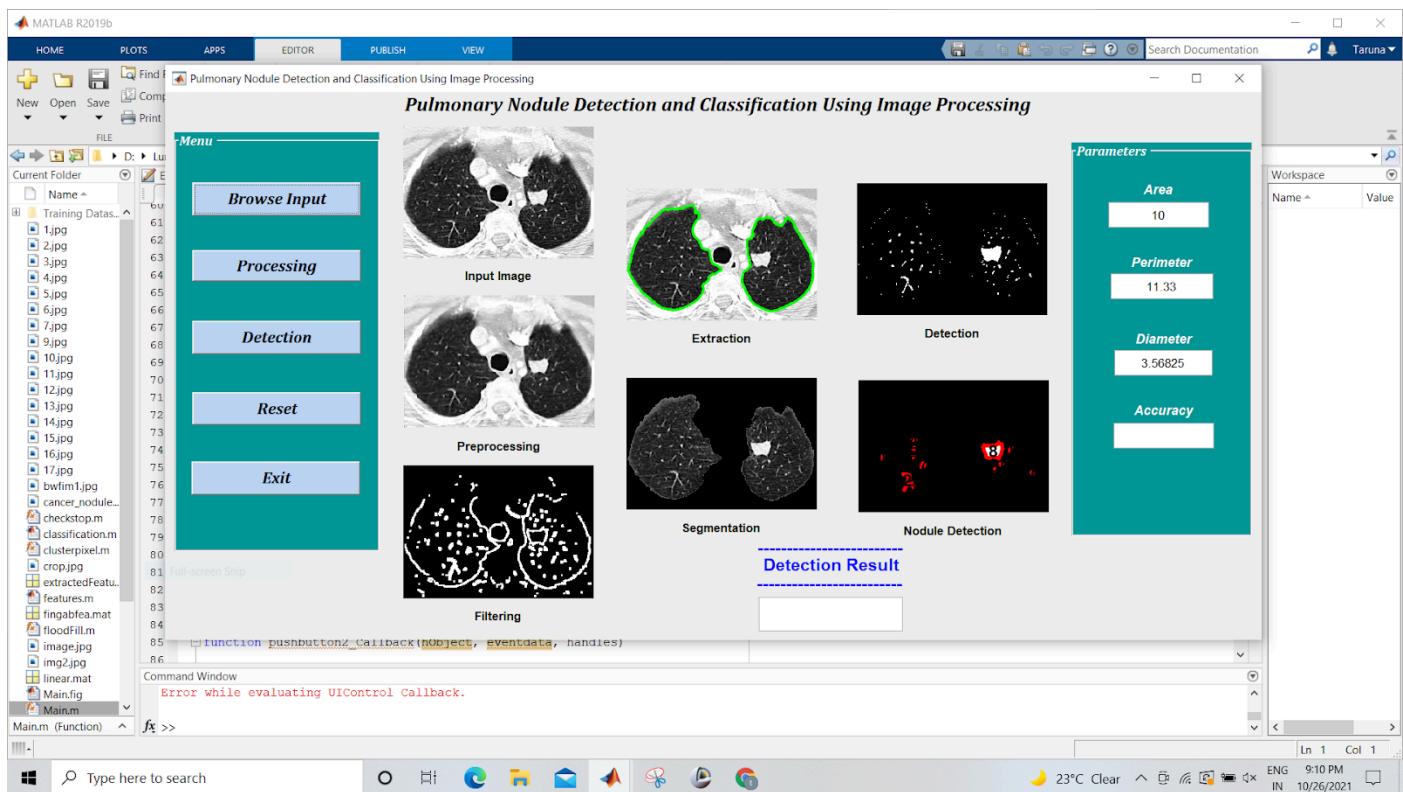


Fig.6.6. Lung Tumor Detection

CODES AND IMPLEMENTATION

Create the files as follows after installing and configuring httpd with these commands:

1. sudo yum install httpd
2. sudo systemctl enable httpd
3. systemctl start httpd

File path: /var/www/html/bmi

```
<!DOCTYPE html>
<head>
<title>Anna Dowlin</title>
<link rel="stylesheet" href="/stylesheet.css">
</head>
<body>
<form action="/cgi-bin/bmi.py">

<p>Enter your height and weight below!</p>
<input type="text" placeholder="Enter height in meters" name="htt">
<input type="text" placeholder="Enter weight in kg" name="wtt"> <input
type="submit">
</form>
</body>
```

File path: /var/www/html/chatbot

```
<!DOCTYPE html>
<head>
<title>Anna Dowlin</title>
<link rel="stylesheet" href="/stylesheet.css">
</head>
<body>
<form action="/cgi-bin/hello.py">

<p>Hi! I'm Anna, your Chatbot.<br>Say hello!</p>
<input type="text" placeholder="Enter your query!" name="x">
<input type="submit">
```

```
</form>
</body>
```

File path: var/www/html/stylesheet.css

```
body {
    text-align: center;
    background-image: url("https://wallpaperaccess.com/full/1261770.jpg");
    background-position:center;
    background-repeat: no-repeat;
    background-attachment: fixed;
    color: white;
    font-family: helvetica;
}
p
{
    font-size: 22px;
}
input
{
    border:0;
    padding:10px;
    font-size:18px;
}
input[type="submit"]{
    background:rgb(50, 169, 193);
    color:white;}
form{
    position: fixed; /* or absolute */
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}
```


File path: /var/www/cgi-bin/bmi.py

```
#!/usr/bin/env python3
//the line above is used to tell linux instance that the script should be executed using
//python3 interpreter

// the line below tells the cgi that the content will be html code
print("content-type: text/html\n")

//importing necessary modules
import subprocess as sp
import cgi

// we take the values from the input tag using the FieldStorage class form =
cgi.FieldStorage()
htt = form.getvalue("htt")
wtt = form.getvalue("wtt")

//now we begin printing our html code
print("<!DOCTYPE html>")
print("<head><title>Anna, Your chatbot!</title>")
print("<link rel=\"stylesheet\" href=\"/stylesheet.css\">")
print("</head><body>")
print("<form action=\"/chatbot\">")
print("<img"
src="https://cdn2.iconfinder.com/data/icons/healthcare-medical-app/64/medical-robo-512.png" style="width: 226px; height: 226px\">")
print("")
bmi = float(wtt)/((float(htt))**2)
print("<p>Your BMI is {}<p>".format(bmi))
// here we use if-else to change the html code
if bmi < 18:
    print("<p>Consume more calories!<p>")
elif bmi > 25:
    print("Consume lesser calories!")
else:
    print("You are fit!")
print("<input type=\"submit\" value=\"Back\"></form>")
```

File path: /var/www/cgi-bin/hello.py (The main API)

```
#!/usr/bin/env python3
//the line above is used to tell linux instance that the script should be executed using
//python3 interpreter

// the line below tells the cgi that the content will be html code
print("content-type: text/html\n")

//importing necessary modules
import subprocess as sp
import cgi
from tabulate import tabulate

// we take the values from the input tag using the FieldStorage class form =
cgi.FieldStorage()
query = form.getvalue("x")

query = str(query).lower()
//now we begin printing our html code
print("<!DOCTYPE html>")
print("<head><title>Anna, Your chatbot!</title>")
print("<link rel=\"stylesheet\" href=\"/stylesheet.css\">")
print("</head><body>")
print("<form action=\"/chatbot\">")
print("<img")
src="https://cdn2.iconfinder.com/data/icons/healthcare-medical-app/64/medical-r
obot-512.png" style="width: 226px; height: 226px">")
print("")
// here we use if-else to change the html code
if "what" in query:
    print("<p>I can tell you the date, time, show you the month's calendar, calculate
your bmi, show some covid stats and give hospital numbers.</p>")
elif "about" in
query:
    print("<p>My name is Anna. My job is to medically assist you.<br/>I'm currently a beta
version.</p>")
elif "tell" in query or "show" in query:
    print("<p>")
if "date" in query:
    print(sp.getoutput("date +%d/%m/%Y"))
elif "time" in query:
    print(sp.getoutput("date +%T"))
elif "cal" in query:
    print("<pre>" +sp.getoutput("cal")+"</pre>")
```

```
print("</p>")
elif "bmi" in query:
    print("<meta http-equiv=\"Refresh\" content=\"0; url='/bmi'\" />") elif
"search" in query:
    print("<meta http-equiv=\"Refresh\" content=\"0; url='https://google.com/'\" />") elif
"doctor" in query:
    ls_hptl = zip(['Right Home Health Care Services', 'G T Padole Hospital', 'Alexis
Multispeciality Hospital', 'Wockhardt Super Speciality Hospital', 'Lata Mangeshkar
Hospital'], [9823395009, 7947288550, 7127120000, 8605604444, 7122536409])
    print("<pre>{}</pre>".format(tabulate(ls_hptl, headers = ["Hospitals", "Phone
Numbers"])))
elif "cases" in query:
    print("<iframe
src=\"https://ourworldindata.org/grapher/total-cases-covid-19?tab=map\""
width="100%" height="600px"></iframe>")
else:
    print("<p>I don't understand that yet!\nHint: Ask me what can I do!</p>")
    print("<input type=\"submit\" value=\"Back\"></form>")
```

FUTURE SCOPE

- 1.This Webapp can be made into an Android/iOS app since the REST API has already been applied.
- 2.It will also help in connecting to various pharmacies across the country to get immediate medicine.
- 3.It will also help in keeping updated about medical problems faced in the country and also help in analyzing solutions
- 4.Speech recognition and text to speech can be applied to go completely hands free, using python modules pyttsx3 and speech_recognition.
- 5.Use of NLP or Natural Language Processing.Adding a list of nearby hospitals using location trackers in devices.
- 6.Arduino based tumor detection

REFRENCES

1. F. Thung, "API recommendation system for software development," 2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE), 2016, pp. 896-899.
2. H. Nindito, A. R. Condrobimo and A. V. Dian Sano, "Rule based classification for diseases articles using oracle database 12c," 2017 International Conference on Information Management and Technology (ICIMTech), 2017, pp. 1-5, doi: 10.1109/ICIMTech.2017.8273500.
3. S. Narula, A. Jain and Prachi, "Cloud Computing Security: Amazon Web Service," 2015 Fifth International Conference on Advanced Computing & Communication Technologies, 2015, pp. 501-505, doi: 10.1109/ACCT.2015.20.
4. P. Katiyar and K. Singh, "A Comparative study of Lung Cancer Detection and Classification approaches in CT images," 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), 2020, pp. 135-142, doi: 10.1109/SPIN48934.2020.9071240.
5. N. Albayrak, A. Özdemir and E. Zeydan, "An overview of artificial intelligence based chatbots and an example chatbot application," 2018 26th Signal Processing and Communications Applications Conference (SIU), 2018, pp. 1-4, doi: 10.1109/SIU.2018.8404430.

