Minor Project Report

On

Medical Assistant API integrated with Cloud

This Minor Project report is submitted in
Partial fulfillments of the requirements for fifth semester term work
in
Electronics & Telecommunication Engineering

Submitted by

Asita Orke (C 33)
Bhavesh Kakrotra (C 36)
Kartik Kamdi(C 35)
Taruna Pakhare (C 34)
Yash Deshmukh (C 32)
Yash Wargantiwar (C 31)

Guided by

Electronics &Telecommunication Engineering Department
G.H.RAISONI COLLEGE OF ENGINEERING
(An Autonomous Institute Affiliated to RTM Nagpur University)
Digdoh Hills, Nagpur-440016

2020-2021
October -2020

G.H. Raisoni College of Engineering
(An Autonomous Institute Affiliated to RTM Nagpur University)
Digdoh Hills, Nagpur-440016
Electronics & Telecommunication Engineering Department
2020-2021

Certificate

This is to certify that the Minor project titled

Medical Assistant API integrated with Cloud

is in partial fulfillment of the requirement for sixth semester term work in Electronics & Telecommunication Engineering Of Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur is a bonafide work carried out & completed under my guidance & supervision during the academic year

2020-2021

Submitted by
Asita Orke (C 33)
Bhavesh Kakrotra (C 36)
Kartik Kamdi(C 35)
Taruna Pakhare (C 34)
Yash Deshmukh (C 32)
Yash Wargantiwar (C 31)

Guided by
Dr Umesh Sahu

Head of the Department	**Incharge, Project Committee**
Dr. S. S. Salankar	Dr. K. A. Akant

ACKNOWLEDGEMENT

We wish to express my gratitude to our guide, **Dr Umesh Sahu** 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.net	

1. Chapters

2. Introduction

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.

3. Problem Statement:

2. This pandemic has changed the way of working for industries around the globe.
3. People are facing difficulties getting appointments from a doctor.
4. People are realising that many things can be done from home, hence people do not want to leave their homes in next few years on falling sick, and they might need something that would diagnose them on the fly.

4. Literature Review

2. It helps in quick contact with hospitals, doctors and pharmacies. In this pandemic bed availability is a big question this software helps in providing all the details of hospitals.
3. It also provides you with a list of medicines which should be used under this pandemic.
4. It also provides the data of patients suffering from corona and full analysis of the situation.
5. It is very helpful for the people who are in other parts of the world and they can provide proper guidance and facility to their family members especially in case of parents or grandparents.

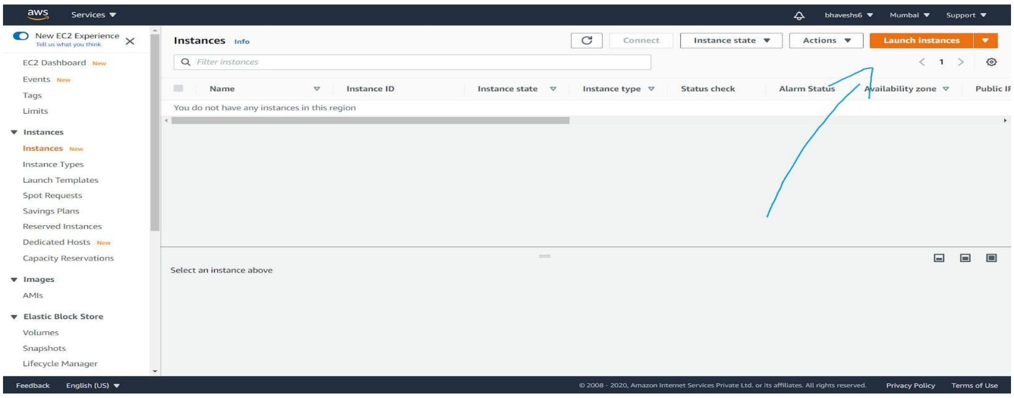
5. This software is very useful for the people of rural areas as they can know the status right from the treatment to the availability of beds in hospitals in the nearby cities.
6. This software avoids chaos while approaching for the information about the pandemic and also will help in preventing people from mislead, malfunction of government bodies and It will create an accountability on both the hospital and the administration to function properly , therefore this software will create a transparency between the common man and the administration.

7. Tools:

1. A Cloud Service i.e AWS, GCP etc...
2. Apache web server i.e httpd
3. ssh Client i.e Putty
4. CGI (Common Gateway Interface)
5. A Linux instance i.e RHEL8
6. Python3
7. HTML
8. CSS

6. Methodology

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

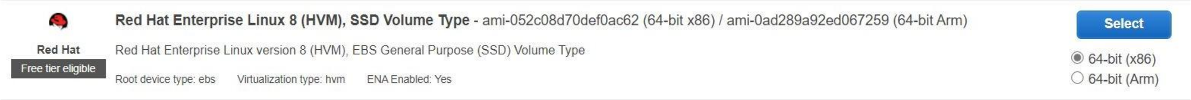


You do not have any instances in this region

Select an instance above

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

2. Select a linux image for our instance.

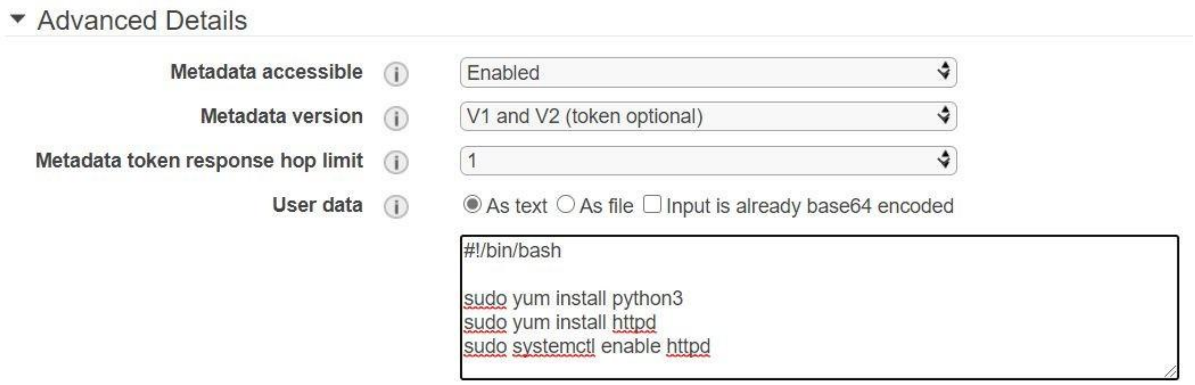


Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-052c08d70def0ac62 (64-bit x86) / ami-0ad289a92ed067259 (64-bit Arm)

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

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

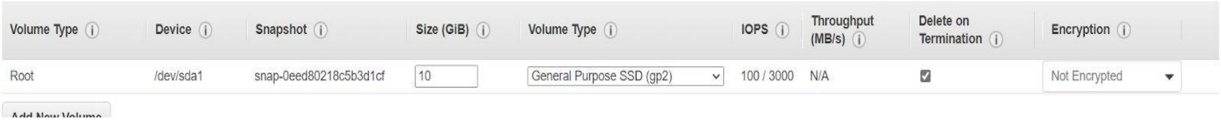


▼ Advanced Details

Metadata accessible	Enabled
Metadata version	V1 and V2 (token optional)
Metadata token response hop limit	1
User data	● As text ○ As file □ Input is already base64 encoded

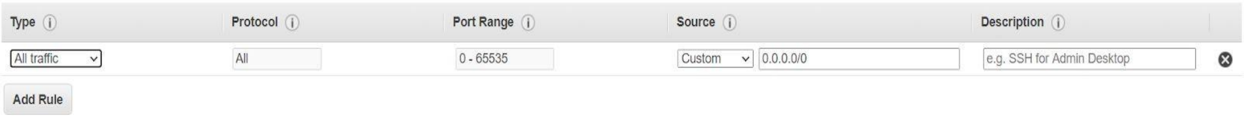
```
#!/bin/bash

sudo yum install python3
sudo yum install httpd
sudo systemctl enable httpd
```

4. Set volume to 10 GiB.



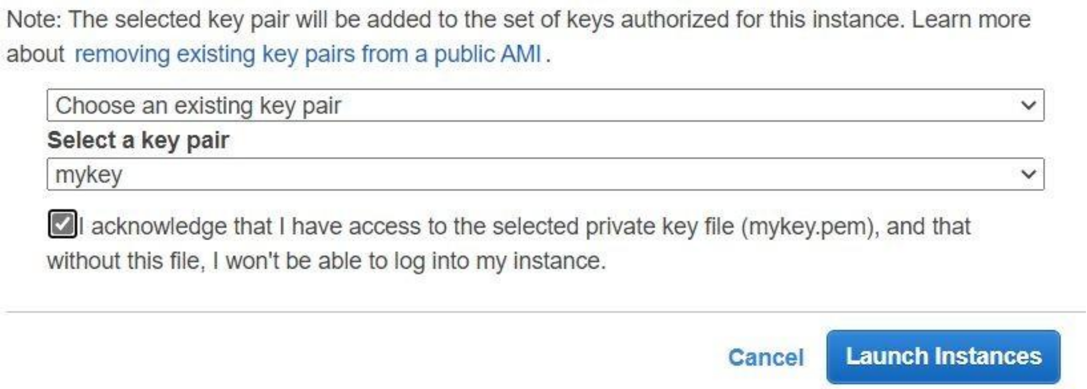
Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0eed80218c5b3d1cf	10	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted

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



Type	Protocol	Port Range	Source	Description
All traffic	All	0 - 65535	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

6. Now launch the instance using a key pair.



Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

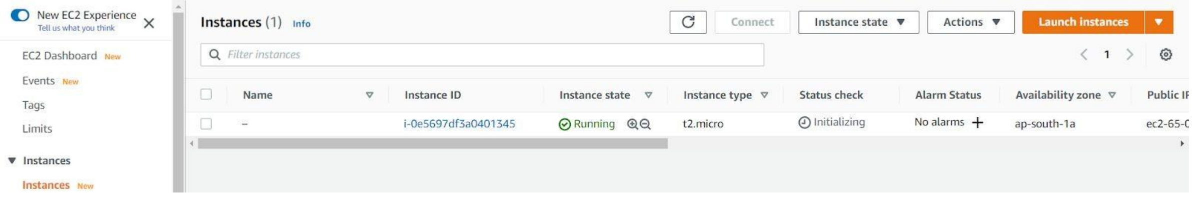
Choose an existing key pair

Select a key pair

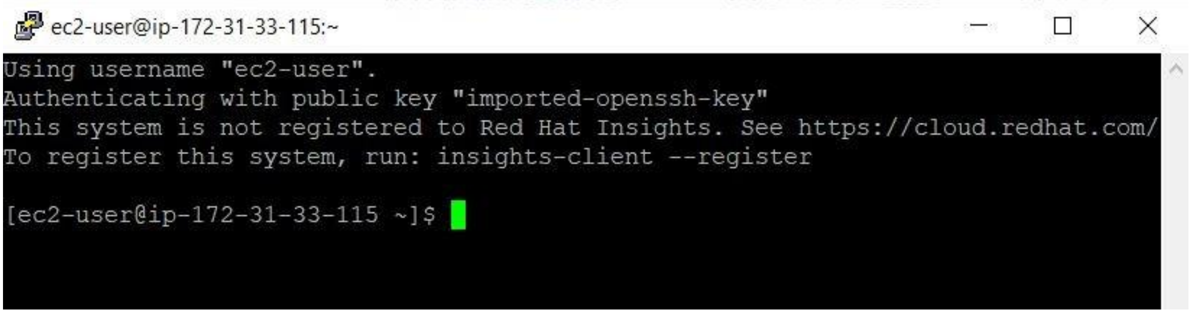
mykey

☑ I acknowledge that I have access to the selected private key file (mykey.pem), and that without this file, I won't be able to log into my instance.

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

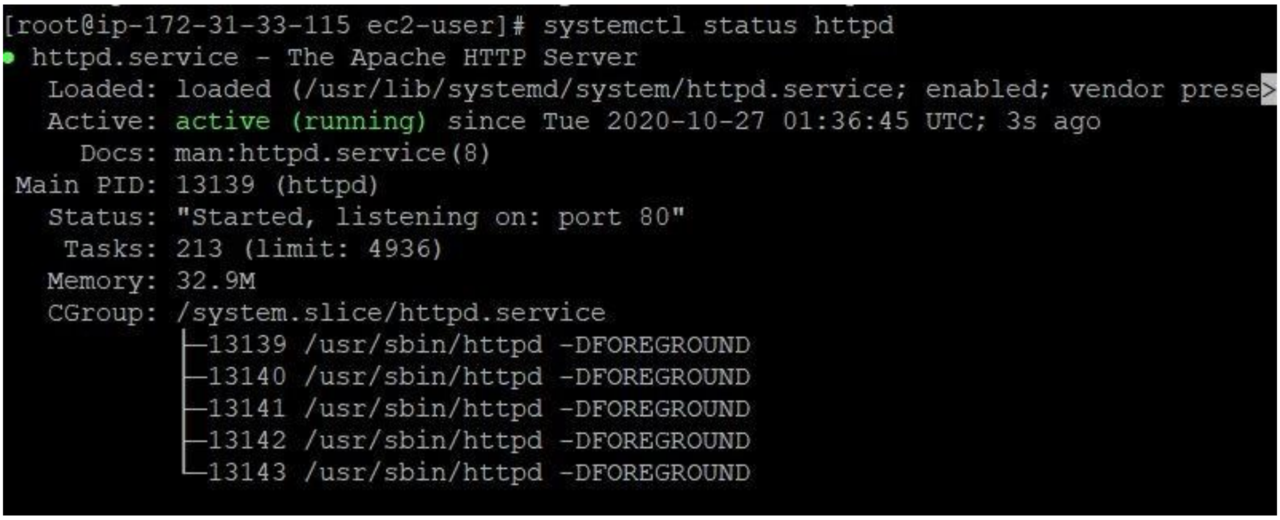


□	Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone	Public IF
□	–	i-0e5697df3a0401345	Running	t2.micro	Initializing	No alarms +	ap-south-1a	ec2-65-(



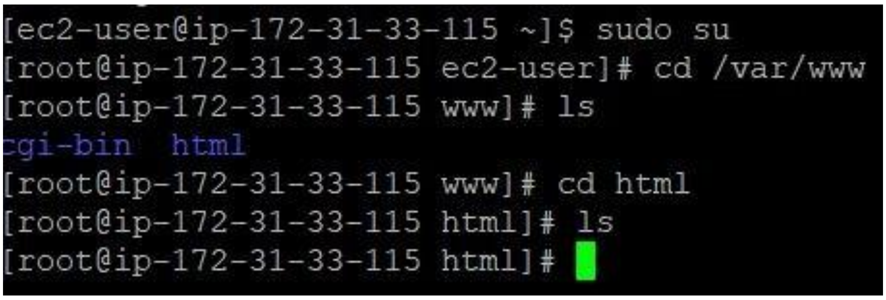
```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
This system is not registered to Red Hat Insights. See https://cloud.redhat.com/
To register this system, run: insights-client --register

[ec2-user@ip-172-31-33-115 ~]$ 
```

8. Check if aur httpd apache server is running



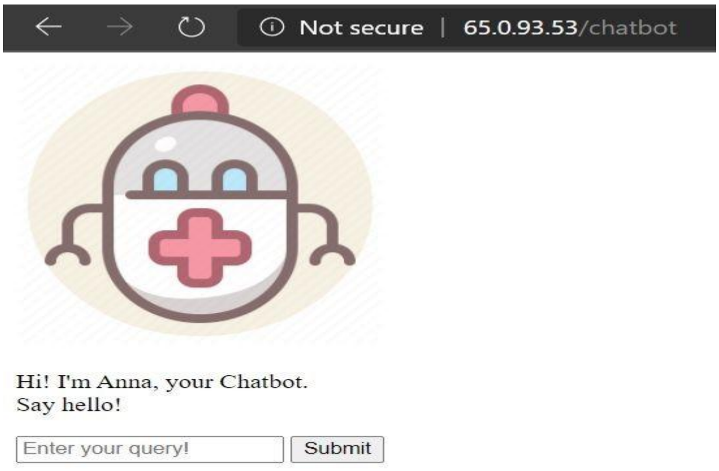
```
[root@ip-172-31-33-115 ec2-user]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor prese>
   Active: active (running) since Tue 2020-10-27 01:36:45 UTC; 3s ago
     Docs: man:httpd.service(8)
 Main PID: 13139 (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
```

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



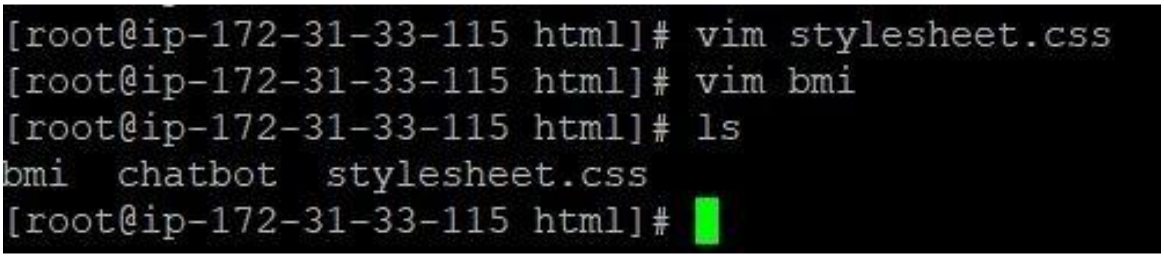
```
[ec2-user@ip-172-31-33-115 ~]$ sudo su
[root@ip-172-31-33-115 ec2-user]# cd /var/www
[root@ip-172-31-33-115 www]# ls
cgi-bin  html
[root@ip-172-31-33-115 www]# cd html
[root@ip-172-31-33-115 html]# ls
[root@ip-172-31-33-115 html]#
```

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



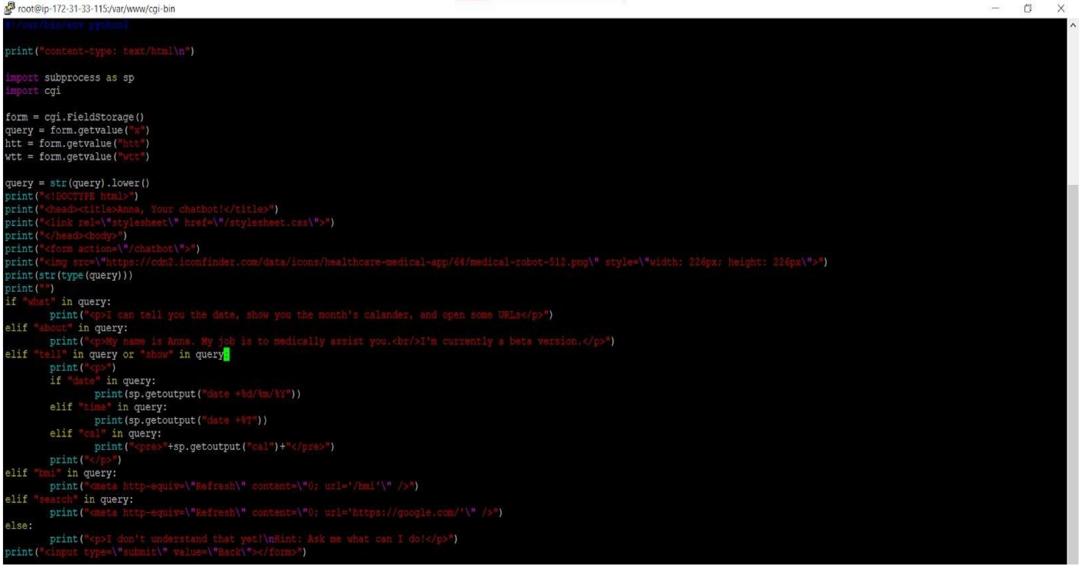
Hi! I'm Anna, your Chatbot.
Say hello!

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



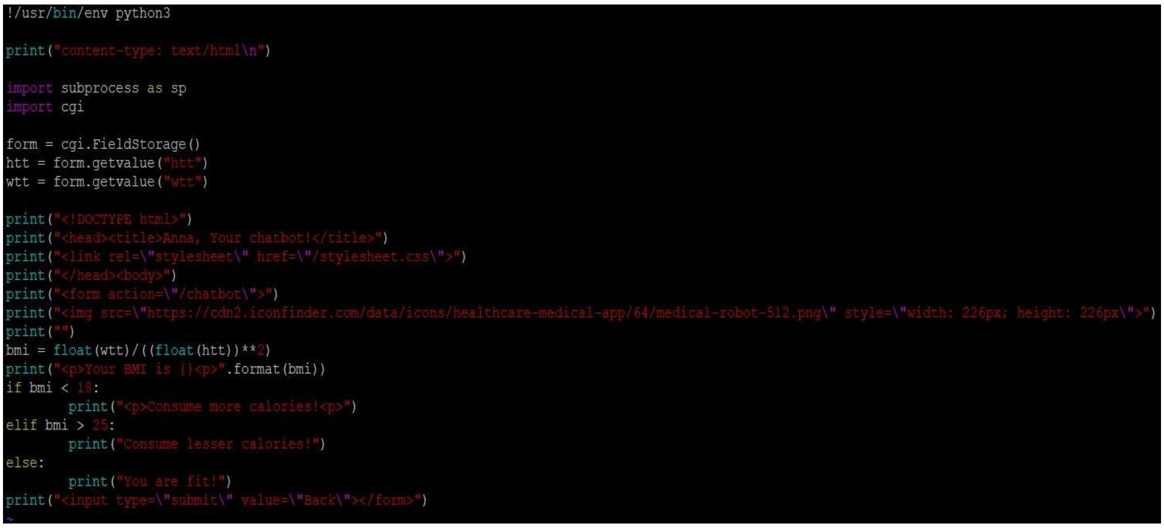
```
[root@ip-172-31-33-115 html]# vim stylesheet.css
[root@ip-172-31-33-115 html]# vim bmi
[root@ip-172-31-33-115 html]# ls
bmi   chatbot   stylesheet.css
[root@ip-172-31-33-115 html]# 
```

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



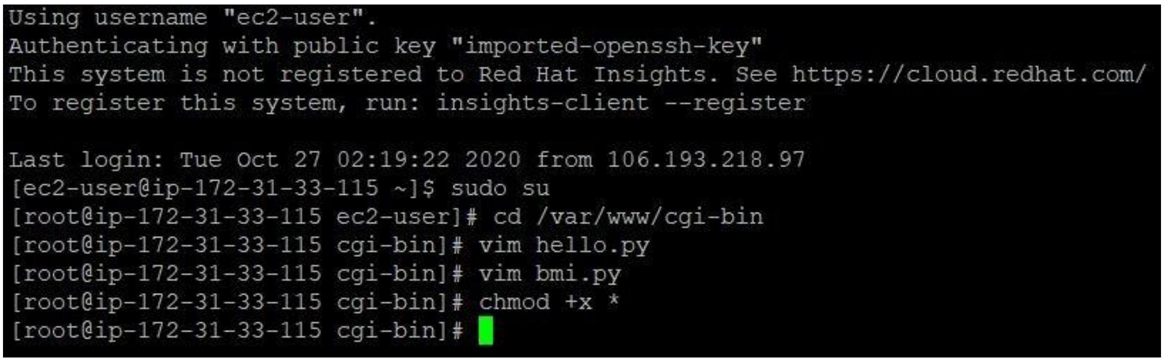
```
#!/usr/bin/env python3

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

import subprocess as sp
import cgi

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

query = str(query).lower()
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(str(type(query)))
print("")
if "what" in query:
        print("<p>I can tell you the date, show you the month's calander, and open some URLs</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/'\" />")
else:
        print("<p>I don't understand that yet!\nHint: Ask me what can I do!</p>")
print("<input type=\"submit\" value=\"Back\"></form>")
```

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



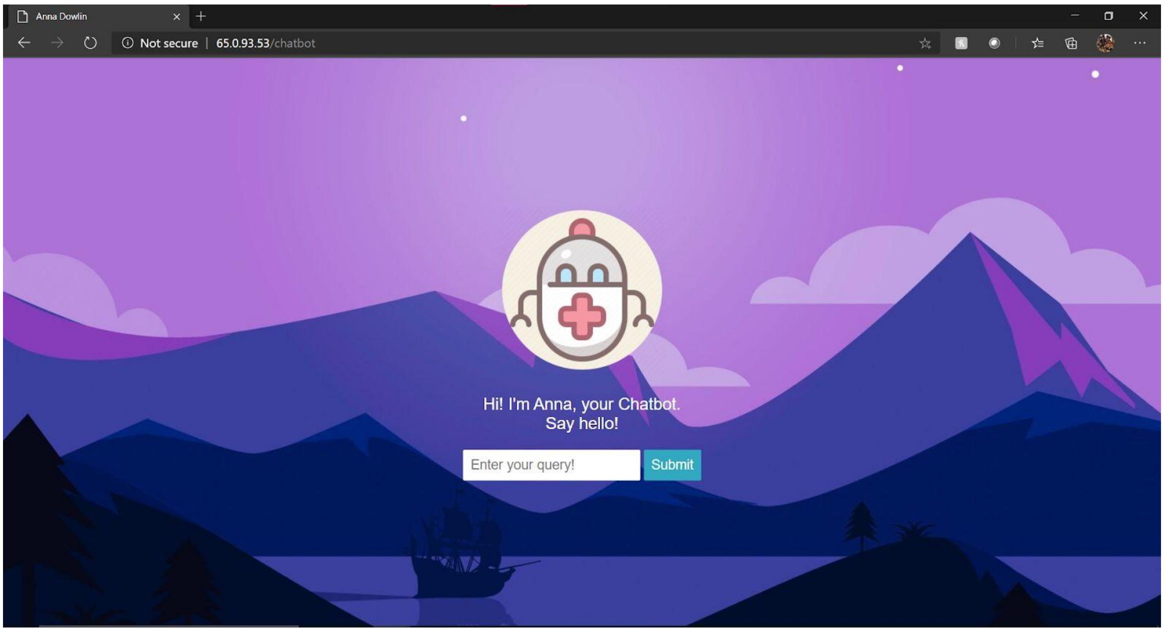
```
!/usr/bin/env python3

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>")
~
```

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]#
```

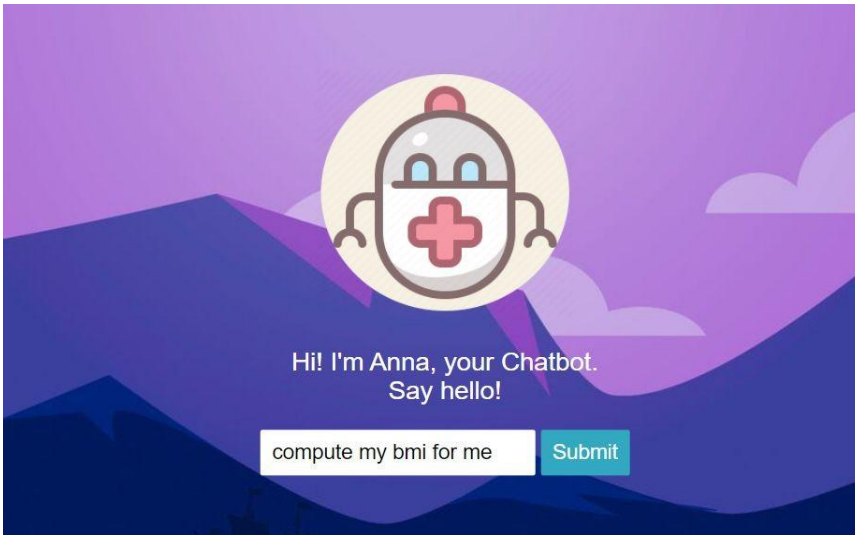
15.We now check if our chatbot is running.



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

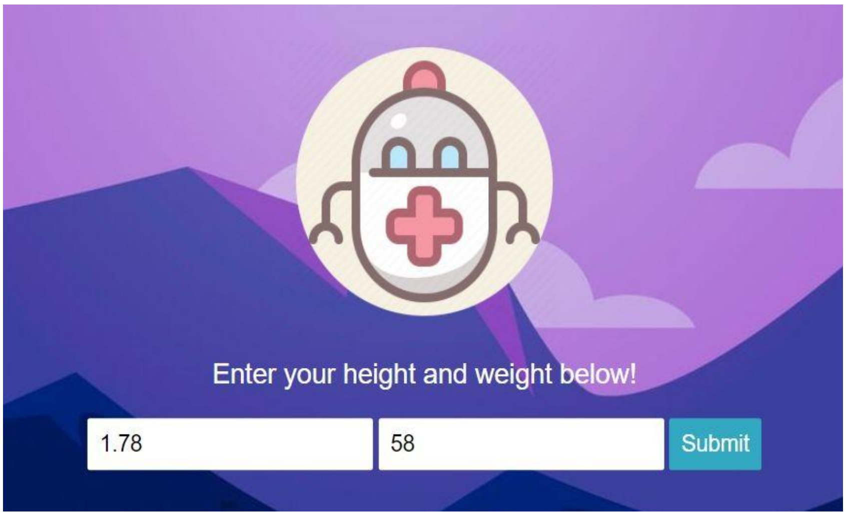
7. Results:

Calculating the BMI:



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

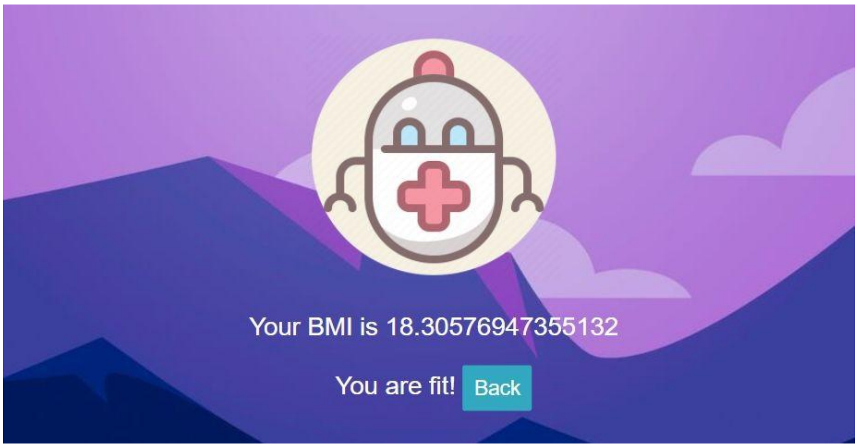
compute my bmi for me



Enter your height and weight below!

1.78

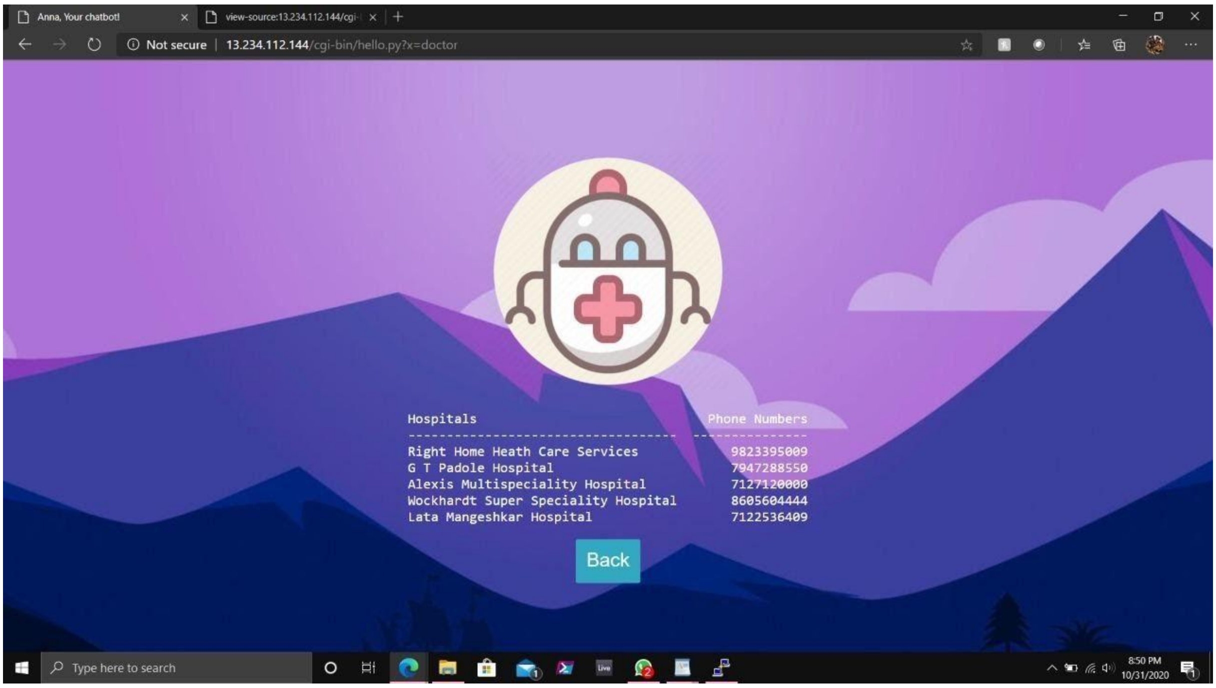
58



Your BMI is 18.30576947355132

You are fit!

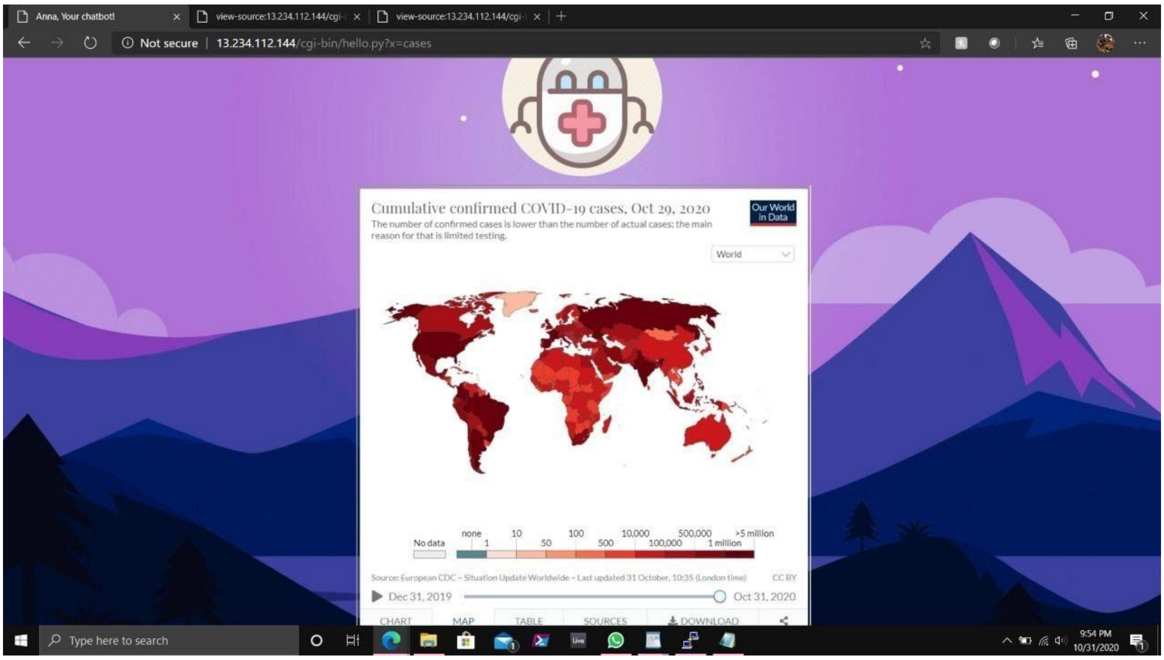
Contact list of Hospitals



Anna, Your chatbot!
view-source:13.234.112.144/cgi-
Not secure
13.234.112.144/cgi-bin/hello.py?x=doctor
Type here to search
8:50 PM
10/31/2020

```
Hospitals                                      Phone Numbers
-----------------------------------    ----------------
Right Home Heath Care Services                 9823395009
G T Padole Hospital                            7947288550
Alexis Multispeciality Hospital                7127120000
Wockhardt Super Speciality Hospital            8605604444
Lata Mangeshkar Hospital                       7122536409
```

Update about COVID

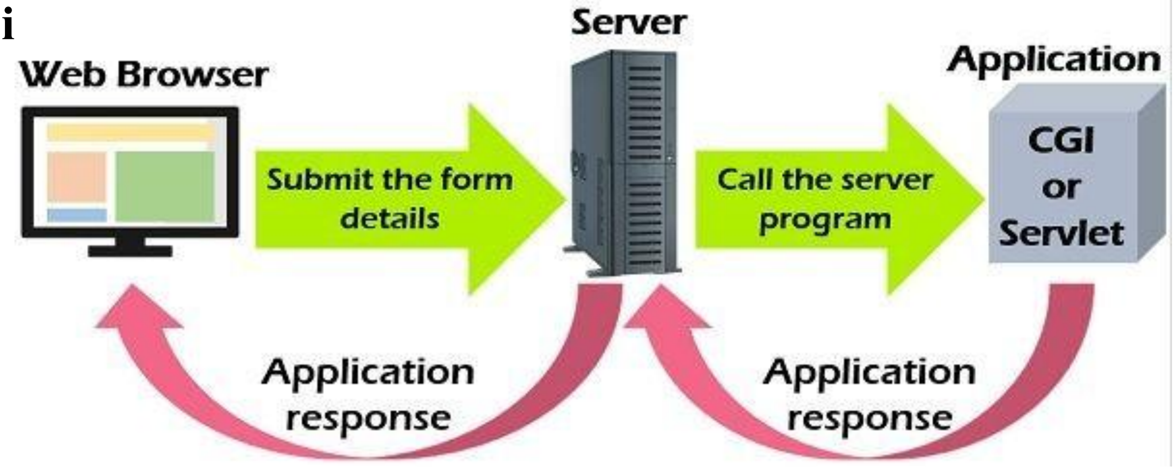


Cumulative confirmed COVID-19 cases, Oct 29, 2020

The number of confirmed cases is lower than the number of actual cases; the main reason for that is limited testing.

Source: European CDC – Situation Update Worldwide – Last updated 31 October, 10:35 (London time)

8. Desi



9. 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">
    <img
src="https://cdn2.iconfinder.com/data/icons/healthcare-medical-app/64/medical-r
obot-512.png" style="width: 226px; height: 226px">
    <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">
    <img
src="https://cdn2.iconfinder.com/data/icons/healthcare-medical-app/64/medical-r
obot-512.png" style="width: 226px; height: 226px">
    <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-robot-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-robot-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>")
```

10. Proposed work for next semester.

1. This project could be used in emergency situations by contacting hospitals and providing emergency medical help.
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 analysing solutions.

11. References:

1. https://ourworldindata.org/grapher/total-cases-covid-19?tab=map
2. https://www.researchgate.net/publication/274527941_A_history_and_future_of_Web_APIs
3. https://ieeexplore.ieee.org/abstract/document/7582836
4. https://ieeexplore.ieee.org/document/8057500
5. https://ieeexplore.ieee.org/document/7195557
6. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8510800
7. https://ieeexplore.ieee.org/document/7079135/
8. https://ieeexplore.ieee.org/document/4142302/

12. Video:

Video of chatbot\

13. Conclusion:

Our project focuses on Health of our people so to prioritize this we made this healthcare device which will help in focusing on health with other work. We learned how API can be used in making software based on health and also how software basic is all focused on API and cloud integrated together.