**PRESIDENCY SCHOOL**

**BANGALORE SOUTH**



**COVID-19 ASSISTANCE**

**Subject: COMPUTER SCIENCE**

**DONE BY:**

ADITYA TIWARI

VANISRI P

ZAINAB TABHA

XII ‘C’

2021-2022

CERTIFICATE

Name: ADITYA TIWARI Class: 12-C

Exam No: \_\_\_\_\_\_\_\_\_

This is certified to be the bonafide work of the student in the computer science laboratory during the academic year 2021-2022.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TEACHER INCHARGE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

EXAMINER’S SIGNATURE PRINCIPAL

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Institution Rubber stamp

**ACKNOWLEDGEMENT**

I wish to express my deep gratitude and sincere thanks to all my teachers for encouragement and the management for providing all facilities to successfully complete the project work.

I extend my sincere thanks to my principal, Mrs. J Bhuvaneshwari and my Computer Science teacher, Mrs. Anupama Ashok Narayanapur whose valuable guidance helped me not only successfully complete the project but also appreciate the beauty of the computer science.

I extend my gratitude to my parents and classmates for their valuable support and time.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Page No.** |
| **1.** | **System Hardware and Software Specifications** | **5** |
| **2.** | **Project Synopsis** | **6** |
| **3.** | **Design Work** | **12** |
| **4.** | **Coding** | **13** |
| **5.** | **Output** | **49** |
| **6.** | **Further Development Area** | **66** |
| **7.** | **Bibliography** | **67** |

**SYSTEM SOFTWARE AND HARDWARE SPECIFICATIONS**

**SOFTWARE**

The software used to run the program are :

* Windows 11.0.
* Python 3.9
* MYSQL 5.5

**HARDWARE**

The hardware used to run the project are :

* Intel Core i7
* 16GB RAM
* Windows 11 Operating System

**PROJECT SYNOPSIS**

Aim of ‘COVID-19 ASSISTANCE’:

The coronavirus (COVID-19) has resulted in an unprecedented crisis that has taken a great toll over our daily lives. The aim of our project named, Covid 19 Assistance, is to help understand the situation better by answering a few Frequently Asked Questions ( FAQs) , giving access to the nearby hospital beds, providing access to the latest Covid 19 statistics, and eligibility to take the vaccine..

Introduction:

The coronavirus disease (COVID-19) pandemic, which originated in the city of Wuhan, China, has quickly spread to various countries, with many cases having been reported worldwide. India, with a population of more than 1.34 billion—the second largest population in the world—will have difficulty in controlling the transmission of the virus. Multiple strategies would be highly necessary to handle the current outbreak; these include computational modeling, statistical tools, and quantitative analyses to control the spread.

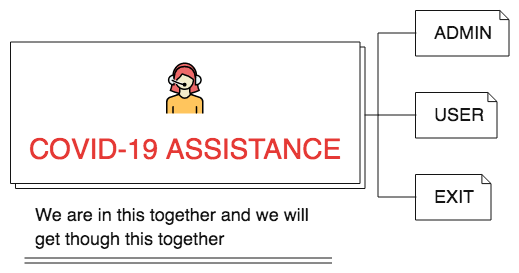
For most people, everything has changed in a short period of time. Many people are still unaware of the severity of the situation, and have a lot of questions unanswered. Our project focuses on disseminating fact-based information and dispelling myths. Vaccination is the most effective way to protect against the infectious disease, we have created a program to check if one is eligible to get the vaccine.

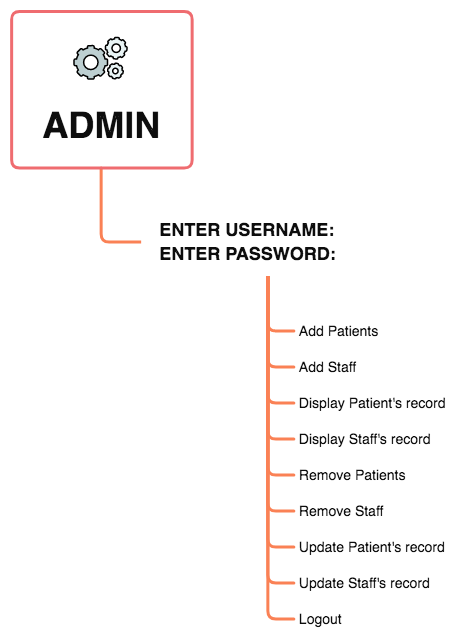
Due to the COVID-19 public health emergency, there was an increase in demand for hospital beds as well, we have created a program that gives one access to the nearby hospital beds by tracking their location. statistical models have also been extremely useful to many by tracing cases worldwide to individual countries, regions, cities, and specific areas within cities as people refer to these models in order to get a sense of the trajectory of this virus. We have also provided a portal that gives one the latest statistical data.

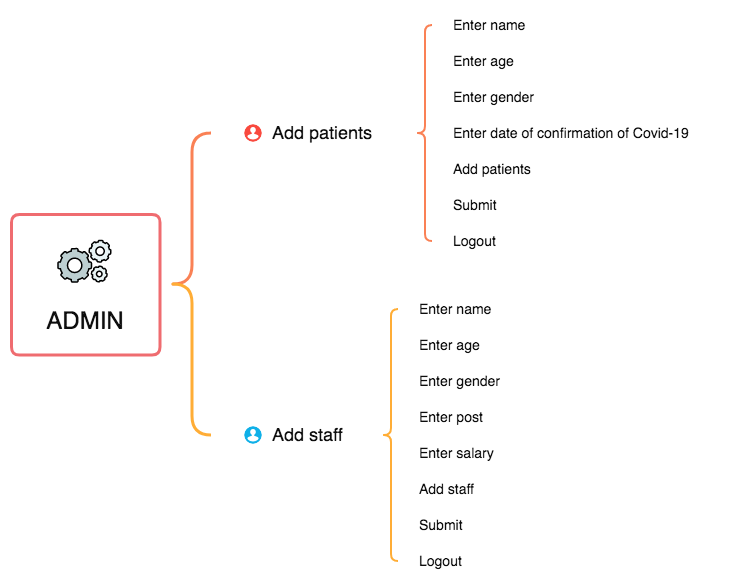
About the project:

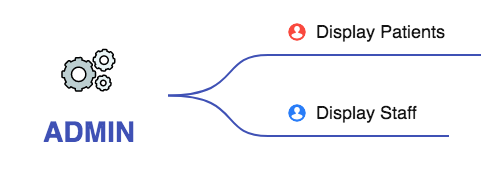
* Sharing Good Practices and Home Remedies: Home remedies are simple measures of symptom management for minor health complaints. In the crisis, sometimes it becomes difficult to get a patient admit due to lack of beds in the hospitals. In such a situation this application provides the user with some home remedies that can reduce the suffering of the patient and provide relief.
* Information provision: A health care center or a hospital record can be maintained using the application. The admin can add, display, remove and update patients and staff records.
* To spread awareness: The application also provides various safety guidelines one should follow to avoid infection.
* Locating Nearby Hospitals: In the case of medical help or an emergency the user can locate nearby hospitals in one click and directly find the google maps for the nearby health care centers.
* Self-Assessment: This is a questionnaire section, through which users can self-diagnose whether they suffer from corona virus. It helps to monitor the symptoms of the user, so that if there are any covid symptoms, the user can approach for a medical help soon. The user needs to answer every question regarding his/her current health condition. Based on the user’s answers, the app can determine whether you may have contracted the virus. In the case of positive assessment, the app will also provide guidance regarding user’s next step. Users can check the application to understand whether they should follow precautions or seek medical attention, based on the self-assess results.
* Covid-19 update: The update section allows the user to stay up to date regarding the number of cases, both locally and nationally. The accurate numbers can help the user to assess his/her risk further. Additionally, the availability of authorized updates prevents rumors and misinformation from spreading.

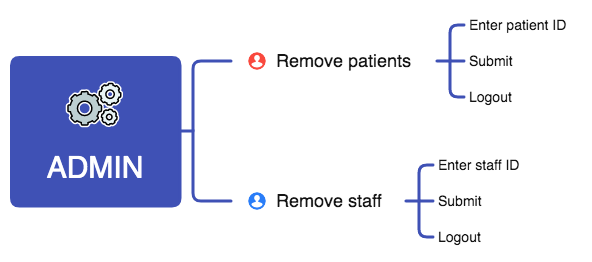
**STRUCTURE OF COVID-19 ASSISTANCE**

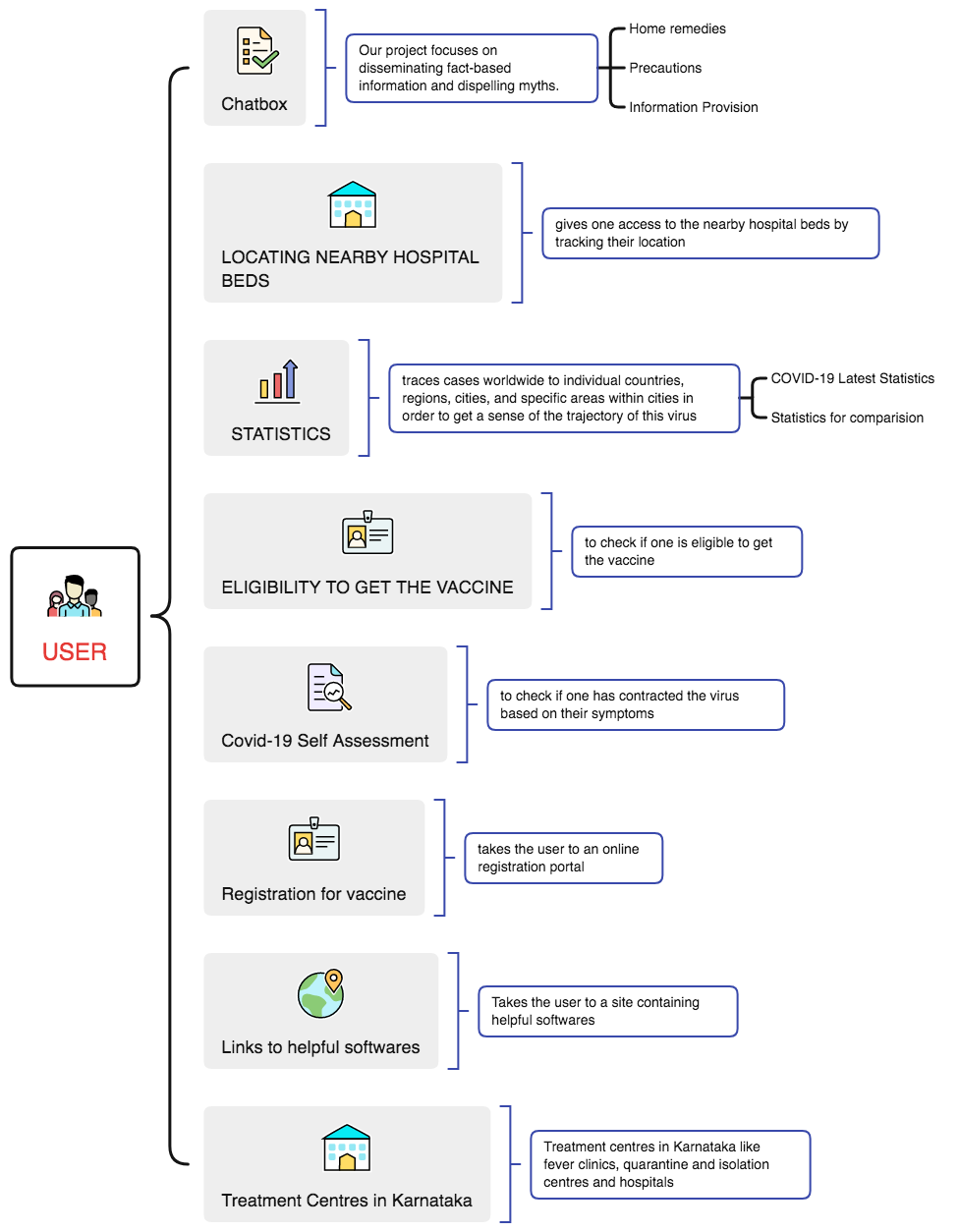


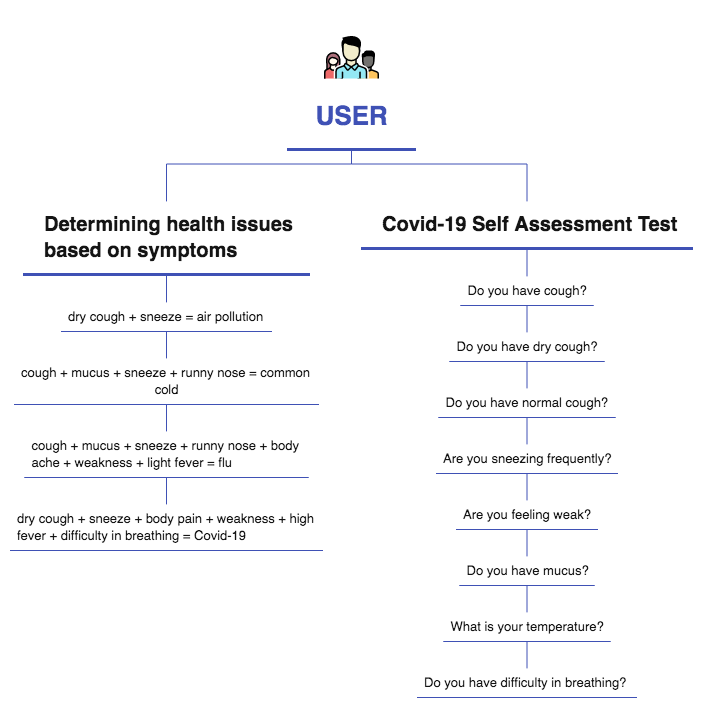












**DESIGN WORK**

**Libraries Used**

* **TKINTER:** Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.
* **MATPLOTLIB:** **Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB.**
* **COVID:** A new python library which tells the COVID-19 related information (country-wise) and it shows that how many cases of confirmed, active, deaths and recovered are found in that particular country.
* **WEB-BROWSER:** It provides a high-level interface to allow displaying web-based documents to users. Under most circumstances, simply calling the open() function from this module will do the right thing.
* **PIL:** Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors. As of 2019, Pillow development is supported by Tidellift.
* **MYSQL.CONNECTOR:** MySQL connector enables Python programs to access MYSQL databases, using an API is compliant with Python Database API Specification v2.

CODING

#Importing the required modules

from tkinter import \*

from PIL import ImageTk,Image

import webbrowser

from tkinter import messagebox

root = Tk()

#Window for admin

#First to check the entered password and then let the admin proceed

def openNewWindowA():

newWindow = Toplevel(root)

newWindow.title("ADMIN:")

newWindow.geometry("700x500")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

l1=Label(newWindow,

text ="Welcome to Admin",

fg="#ffcc66",bg="#141414")

l1.place(x=300,y=140)

#To check the password entered by the admin

def checkpswd():

x1=inputtxt1.get()

if x1=="123":

lbl = Label(newWindow, text = "You may proceed",

fg="#ffcc66",bg="#141414")

lbl.place(x=320,y=380)

printButton2 = Button(newWindow,

text = "Enter", fg="#ffcc66",bg="#141414",border=10,

command = openWindowADMIN)

printButton2.place(x=400,y=300)

else:

lbl = Label(newWindow, text = "Please enter the correct password",

fg="#ffcc66",bg="#141414")

lbl.place(x=280,y=420)

l1=Label(newWindow,

text ="Username:",

fg="#ffcc66",bg="#141414")

l1.place(x=150,y=200)

l1=Label(newWindow,

text ="Password:",

fg="#ffcc66",bg="#141414")

l1.place(x=150,y=250)

inputtxt = Text(newWindow,

height = 1,

width = 25)

inputtxt.place(x=250,y=200)

inputtxt1 = Entry(newWindow,show="\*")

inputtxt1.place(x=250,y=250,

height=27, width=230)

printButton1 = Button(newWindow,

text = "Check", fg="#ffcc66",bg="#141414", border=10,

command = checkpswd)

printButton1.place(x=300,y=300)

#Connecting mysql to python

import mysql.connector

mydb=mysql.connector.connect(host="localhost", user="root", password="mysql")

#To check if the database is connected with python program

#if mydb.is\_connected():

# print("Yes")

#Creating mycursor object

mycursor=mydb.cursor()

#Creating database

mycursor.execute("create database if not exists covid\_management")

#Using the database

mycursor.execute("use covid\_management")

#Creating the required tables

mycursor.execute("create table if not exists staff(staff\_id char(4), name varchar(20), age char(3), gender varchar(7), post varchar(20), salary char(7))")

mycursor.execute("create table if not exists patients(patient\_id char(4), name varchar(20), age char(3), gender varchar(7), confirmation\_date char(10))")

#Window under Admin

def openWindowADMIN():

newWindow = Toplevel(root)

newWindow.title("ADMIN:")

newWindow.geometry("700x700")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

printButton1 = Button(newWindow,

text = "Add Patients",fg="#25dae9",bg="#141414",

command=WindowaddPatients)

printButton1.place(x=300,y=20)

printButton2 = Button(newWindow,

text = "Add Staff",fg="#25dae9",bg="#141414",

command=Windowaddstaff)

printButton2.place(x=300,y=60)

printButton3= Button(newWindow,

text = "Display Patients Record",

fg="#25dae9",bg="#141414",

command=display\_patient\_record)

printButton3.place(x=300,y=100)

printButton4= Button(newWindow,

text = "Display Staff's Record",

fg="#25dae9",bg="#141414",

command=DispStaffrec)

printButton4.place(x=300,y=140)

printButton5= Button(newWindow,

text = "Remove Patients",

fg="#25dae9",bg="#141414",

command=RemovePatients)

printButton5.place(x=300,y=180)

printButton6= Button(newWindow,

text = "Remove Staff",

fg="#25dae9",bg="#141414",

command=RemoveStaff)

printButton6.place(x=300,y=220)

printButton7= Button(newWindow,

text = "Update Patient's Record",

fg="#25dae9",bg="#141414",

command=UpdatePatientRec)

printButton7.place(x=300,y=260)

printButton8= Button(newWindow,

text = "Update Staff's Record",

fg="#25dae9",bg="#141414",

command=UpdateStaffRec)

printButton8.place(x=300,y=300)

printButton10= Button(newWindow,

text = "LOGOUT",fg="#25dae9",bg="#141414",

command=newWindow.destroy)

printButton10.place(x=300,y=380)

#Adding the patient to the database

def add\_patients():

sql='INSERT INTO patients (patient\_id, name, age, gender, confirmation\_date) VALUES (%s,%s,%s,%s,%s)'

p\_id=inputtxt.get()

p\_name=inputtxt1.get()

p\_age=inputtxt2.get()

p\_gender=inputtxt3.get()

p\_date=inputtxt4.get()

T=(p\_id, p\_name, p\_age, p\_gender,p\_date)

val1=T

mycursor.execute(sql,T)

mydb.commit()

messagebox.showinfo('Output',

'Data of Patient has been saved successfully')

#Window for add patients

def WindowaddPatients():

newWindow = Toplevel(root)

newWindow.title("Add Patients:")

newWindow.geometry("700x300")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

l=Label(newWindow, text="Enter Patient ID:", fg="#f86263",bg="#141414")

l.place(x=1,y=0)

global inputtxt

global inputtxt1

global inputtxt2

global inputtxt3

global inputtxt4

inputtxt = Entry(newWindow)

inputtxt.place(x=150,y=1,height = 27,

width = 150)

inputtxt1 = Entry(newWindow)

inputtxt1.place(x=150,y=40,height = 27,

width = 150)

patient\_id=inputtxt.get()

name=inputtxt1.get()

l1=Label(newWindow, text="Enter name:",fg="#f86263",bg="#141414")

l1.place(x=0,y=40)

l2=Label(newWindow, text="Enter Age:", fg="#f86263",bg="#141414")

l2.place(x=0,y=80)

inputtxt2 = Entry(newWindow)

inputtxt2.place(x=150,y=80,height = 27,

width = 150)

age=inputtxt2.get()

l3=Label(newWindow, text="Enter Gender:",fg="#f86263",bg="#141414")

l3.place(x=0,y=120)

inputtxt3 = Entry(newWindow)

inputtxt3.place(x=150,y=120,height = 27,

width = 150)

gender=inputtxt3.get()

l4=Label(newWindow, text="Enter date of confirmation of Covid-19(YYYY-MM-DD):",

fg="#f86263",bg="#141414")

l4.place(x=0,y=160)

inputtxt4 = Entry(newWindow)

inputtxt4.place(x=360,y=160,height = 27,

width = 150)

date=inputtxt4.get()

printButton6 = Button(newWindow,

text = "SUBMIT",fg="#f86263",bg="#141414",

command=add\_patients)

printButton6.place(x=20,y=240)

printButton7 = Button(newWindow,

text = "LOGOUT",fg="#f86263",bg="#141414",

command=newWindow.destroy)

printButton7.place(x=300,y=240)

#Adding the staff to the database

def add\_staff():

sql='INSERT INTO staff (staff\_id, name, age, gender, post, salary) VALUES (%s,%s,%s,%s,%s,%s)'

s\_id=inputtxt.get()

s\_name=inputtxt1.get()

s\_age=inputtxt2.get()

s\_gender=inputtxt3.get()

s\_post=inputtxt4.get()

s\_salary=inputtxt5.get()

T=(s\_id, s\_name, s\_age, s\_gender,s\_post,s\_salary)

val1=T

mycursor.execute(sql,T)

mydb.commit()

messagebox.showinfo('Output', 'Data of Staff has been saved successfully')

#Window for add staff

def Windowaddstaff():

newWindow = Toplevel(root)

newWindow.title("Add Staff:")

newWindow.geometry("700x350")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

l=Label(newWindow, text="Enter Staff ID:", fg="#f86263",bg="#141414")

l.place(x=1,y=0)

global inputtxt

global inputtxt1

global inputtxt2

global inputtxt3

global inputtxt4

global inputtxt5

inputtxt = Entry(newWindow)

inputtxt.place(x=150,y=1,height = 27,

width = 150)

staff\_id=inputtxt.get()

inputtxt1 = Entry(newWindow)

inputtxt1.place(x=150,y=40,height = 27,

width = 150)

l1=Label(newWindow, text="Enter name:",fg="#f86263",bg="#141414")

l1.place(x=0,y=40)

name=inputtxt1.get()

l2=Label(newWindow, text="Enter Age:", fg="#f86263",bg="#141414")

l2.place(x=0,y=80)

inputtxt2 = Entry(newWindow)

inputtxt2.place(x=150,y=80,height = 27,

width = 150)

age=inputtxt2.get()

l3=Label(newWindow, text="Enter Gender:", fg="#f86263",bg="#141414")

l3.place(x=0,y=120)

inputtxt3 = Entry(newWindow)

inputtxt3.place(x=150,y=120,height = 27,

width = 150)

gender=inputtxt3.get()

l4=Label(newWindow, text="Enter post:",fg="#f86263",bg="#141414")

l4.place(x=0,y=160)

inputtxt4 = Entry(newWindow)

inputtxt4.place(x=150,y=160,height = 27,

width = 150)

post=inputtxt4.get()

l5=Label(newWindow, text="Enter salary:", fg="#f86263",bg="#141414")

l5.place(x=0,y=200)

inputtxt5 = Entry(newWindow)

inputtxt5.place(x=150,y=200,height = 27,

width = 150)

salary=inputtxt5.get()

printButton7 = Button(newWindow,

text = "SUBMIT", fg="#f86263",bg="#141414",command=add\_staff)

printButton7.place(x=20,y=300)

printButton8 = Button(newWindow,

text = "LOGOUT",fg="#f86263",bg="#141414",

command=newWindow.destroy)

printButton8.place(x=300,y=300)

#Displaying patients record in the form of table

def display\_patient\_record():

newWindow = Toplevel(root)

newWindow.title("Display Patient's record:")

newWindow.geometry("500x200")

mycursor.execute("SELECT \* FROM patients")

myresult=mycursor.fetchall()

e=Label(newWindow,width=10,text='ID',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=0)

e=Label(newWindow,width=10,text='Name',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=1)

e=Label(newWindow,width=10,text='Age',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=2)

e=Label(newWindow,width=10,text='Gender',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=3)

e=Label(newWindow,width=10,text='Date',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=4)

i=1

for x in myresult:

for j in range(len(x)):

e = Label(newWindow,width=10, text=x[j],

borderwidth=2,relief='ridge', anchor="w")

e.grid(row=i, column=j)

i=i+1

#Displaying staff's record in the form of table

def DispStaffrec():

newWindow = Toplevel(root)

newWindow.title("Display Staff's record:")

newWindow.geometry("500x200")

mycursor.execute("SELECT \* FROM staff")

myresult=mycursor.fetchall()

e=Label(newWindow,width=10,text='ID',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=0)

e=Label(newWindow,width=10,text='Name',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=1)

e=Label(newWindow,width=10,text='Age',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=2)

e=Label(newWindow,width=10,text='Gender',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=3)

e=Label(newWindow,width=10,text='Post',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=4)

e=Label(newWindow,width=10,text='Salary',borderwidth=2, relief='ridge',anchor='w',bg='yellow')

e.grid(row=0,column=5)

i=1

for x in myresult:

for j in range(len(x)):

e = Label(newWindow,width=10, text=x[j],

borderwidth=2,relief='ridge', anchor="w")

e.grid(row=i, column=j)

i=i+1

#Removing patients from the database

def remove\_patients():

rec=inputtxt5.get()

mycursor.execute("select patient\_id from patients")

result=tuple(mycursor.fetchall())

length=len(result)

for i in result:

for item in i[0:6:length+1]:

if int(item)==int(rec):

mycursor.execute("delete from patients where patient\_id=%s",(rec,))

mydb.commit()

messagebox.showinfo('Output', 'Patient has been removed successfully')

else:

messagebox.showerror('Output', "Entered Patient ID doesn't exist")

#Window for removing patients

def RemovePatients():

newWindow = Toplevel(root)

newWindow.title("Remove Patients:")

newWindow.geometry("450x100")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

l1=Label(newWindow, text="Enter Patient ID:", fg="#f86263",bg="#141414")

l1.place(x=1,y=0)

global inputtxt5

inputtxt5 = Entry(newWindow)

inputtxt5.place(x=180,y=0,

height=27,width=150)

printButton1 = Button(newWindow,

text = "SUBMIT",fg="#f86263",bg="#141414",command=remove\_patients)

printButton1.place(x=20,y=50)

printButton2 = Button(newWindow,

text = "LOGOUT",fg="#f86263",bg="#141414",

command=newWindow.destroy)

printButton2.place(x=300,y=50)

#Removing staff from the database

def remove\_staff():

rec=inputtxt.get()

mycursor.execute("select staff\_id from staff")

result=tuple(mycursor.fetchall())

length=len(result)

for i in result:

for item in i[0:6:length+1]:

if int(item)==int(rec):

mycursor.execute("delete from staff where staff\_id=%s",(rec,))

mydb.commit()

messagebox.showinfo('Output', 'Staff has been removed successfully')

else:

messagebox.showerror('Output', "Entered Staff ID doesn't exist")

#Window for removing staff

def RemoveStaff():

newWindow = Toplevel(root)

newWindow.title("Remove Staff:")

newWindow.geometry("450x100")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

l1=Label(newWindow, text="Enter Staff ID:", fg="#f86263",bg="#141414" )

l1.place(x=1,y=0)

global inputtxt

inputtxt = Entry(newWindow)

inputtxt.place(x=180,y=0,height = 27,

width = 150)

printButton1 = Button(newWindow,

text = "SUBMIT", fg="#f86263",bg="#141414",command=remove\_staff)

printButton1.place(x=20,y=50)

printButton2 = Button(newWindow,

text = "LOGOUT", fg="#f86263",bg="#141414",

command=newWindow.destroy)

printButton2.place(x=300,y=50)

#Updating patient's record in the database

def update\_patients():

rec=inputtxt.get()

sql="UPDATE patients SET name=%s,age=%s,gender=%s,confirmation\_date=%s where patient\_id=%s"

name=inputtxt1.get()

age=inputtxt2.get()

gender=inputtxt3.get()

date=inputtxt4.get()

T=(name,age,gender,date,rec)

mycursor.execute(sql,T)

mydb.commit()

messagebox.showinfo('Output', 'Multiple columns updated successfully')

#Window for updating patient's record

def UpdatePatientRec():

newWindow = Toplevel(root)

newWindow.title("Update Patient's record:")

newWindow.geometry("500x100")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

global inputtxt

text=Label(newWindow, text="Specify the ID of the patient to be updated:",

fg="#f86263",bg="#141414")

text.place(x=1,y=0)

inputtxt = Entry(newWindow)

inputtxt.place(x=350,y=0,height = 27,

width = 150)

printButton = Button(newWindow,

text = "SUBMIT",fg="#f86263",bg="#141414",command=UpdatePatientDetails)

printButton.place(x=200,y=50)

#Taking the details from admin for patient's record

def UpdatePatientDetails():

newWindow = Toplevel(root)

newWindow.title("Update Patient's record:")

newWindow.geometry("550x500")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

l1=Label(newWindow, text="Enter name:",fg="#f86263",bg="#141414")

l1.place(x=0,y=40)

global inputtxt1

global inputtxt2

global inputtxt3

global inputtxt4

inputtxt1 = Entry(newWindow)

inputtxt1.place(x=150,y=40,height = 27,

width = 150)

l2=Label(newWindow, text="Enter Age:", fg="#f86263",bg="#141414")

l2.place(x=0,y=80)

inputtxt2 = Entry(newWindow)

inputtxt2.place(x=150,y=80,height = 27,

width = 150)

l3=Label(newWindow, text="Enter Gender:",fg="#f86263",bg="#141414")

l3.place(x=0,y=120)

inputtxt3 = Entry(newWindow)

inputtxt3.place(x=150,y=120,height = 27,

width = 150)

l4=Label(newWindow, text="Enter date of confirmation of Covid-19(YY/MM/DD):",

fg="#f86263",bg="#141414")

l4.place(x=0,y=160)

inputtxt4 = Entry(newWindow)

inputtxt4.place(x=360,y=160,height = 27,

width = 150)

printButton6 = Button(newWindow,

text = "SUBMIT",fg="#f86263",bg="#141414",

command=update\_patients)

printButton6.place(x=20,y=240)

printButton7 = Button(newWindow,

text = "LOGOUT",fg="#f86263",bg="#141414",

command=newWindow.destroy)

printButton7.place(x=300,y=240)

#Updating staff's record in the database

def update\_staff():

rec=inputtxt.get()

sql="UPDATE staff SET name=%s,age=%s,gender=%s,post=%s, salary=%s where staff\_id=%s"

name=inputtxt1.get()

age=inputtxt2.get()

gender=inputtxt3.get()

post=inputtxt4.get()

salary=inputtxt5.get()

T=(name,age,gender,post,salary,rec)

mycursor.execute(sql,T)

mydb.commit()

messagebox.showinfo('Output', 'Multiple columns updated successfully')

#Window for updating patient's record

def UpdateStaffRec():

newWindow = Toplevel(root)

newWindow.title("Update Staff's record:")

newWindow.geometry("500x100")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

global inputtxt

text=Label(newWindow, text="Specify the name of the staff to be updated:",

fg="#f86263",bg="#141414")

text.place(x=1,y=0)

inputtxt = Entry(newWindow)

inputtxt.place(x=350,y=0,height = 27,

width = 150)

printButton = Button(newWindow,

text = "SUBMIT",fg="#f86263",bg="#141414",command=UpdateStaffDetails)

printButton.place(x=200,y=50)

#Taking the details from admin for patient's record

def UpdateStaffDetails():

newWindow = Toplevel(root)

newWindow.title("Update Staff's record:")

newWindow.geometry("550x500")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

l1=Label(newWindow, text="Enter name:",fg="#f86263",bg="#141414")

l1.place(x=0,y=40)

global inputtxt1

global inputtxt2

global inputtxt3

global inputtxt4

global inputtxt5

inputtxt1 = Entry(newWindow)

inputtxt1.place(x=150,y=40,height = 27,

width = 150)

l2=Label(newWindow, text="Enter Age:", fg="#f86263",bg="#141414")

l2.place(x=0,y=80)

inputtxt2 = Entry(newWindow)

inputtxt2.place(x=150,y=80,height = 27,

width = 150)

l3=Label(newWindow, text="Enter Gender:", fg="#f86263",bg="#141414")

l3.place(x=0,y=120)

inputtxt3 = Entry(newWindow)

inputtxt3.place(x=150,y=120,height = 27,

width = 150)

l4=Label(newWindow, text="Enter post:",fg="#f86263",bg="#141414")

l4.place(x=0,y=160)

inputtxt4 = Entry(newWindow)

inputtxt4.place(x=150,y=160,height = 27,

width = 150)

l5=Label(newWindow, text="Enter salary:", fg="#f86263",bg="#141414")

l5.place(x=0,y=200)

inputtxt5 = Entry(newWindow)

inputtxt5.place(x=150,y=200,height = 27,

width = 150)

printButton7 = Button(newWindow,

text = "SUBMIT", fg="#f86263",bg="#141414",

command=update\_staff)

printButton7.place(x=20,y=300)

printButton8 = Button(newWindow,

text = "LOGOUT",fg="#f86263",bg="#141414",

command=newWindow.destroy)

printButton8.place(x=300,y=300)

#Defining Logout

def logout():

newWindow = Toplevel(root)

newWindow.title("Logout")

newWindow.geometry("800x200")

#Window for user

def openNewWindowU():

newWindow = Toplevel(root)

newWindow.title("USER:")

newWindow.geometry("700x700")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

label=Label(newWindow,

text ="-Welcome to User-",fg="#25dae9",bg="#141414", font="Impact 30")

label.place(x=190,y=100)

printButton1 = Button(newWindow,

text = "Chatbot",fg="#25dae9",bg="#141414", border=0,

command=FAQ)

printButton1.place(x=290,y=200)

printButton2 = Button(newWindow,

text = "Locating nearby hospital beds",

fg="#25dae9",bg="#141414",border=0,

command=hospitalbeds)

printButton2.place(x=240,y=250)

printButton3= Button(newWindow,

text = "Eligibiltiy to get the vaccine",

fg="#25dae9",bg="#141414",border=0,

command=Eligibility)

printButton3.place(x=250,y=300)

printButton4= Button(newWindow,

text = "Covid-19 Self Assessment",

fg="#25dae9",bg="#141414",border=0,

command=SAssessment)

printButton4.place(x=250,y=350)

printButton5= Button(newWindow,

text = "Registration for vaccine",

fg="#25dae9",bg="#141414",border=0,

command=registration)

printButton5.place(x=250,y=400)

printButton6= Button(newWindow,

text = "Links to helpful softwares",

fg="#25dae9",bg="#141414",border=0,

command=helpfulsoftwares)

printButton6.place(x=250,y=450)

printButton7= Button(newWindow,

text = '''Treatment centres in Karnataka like fever clinics,

quarantine & isolation centres and hospitals''',

fg="#25dae9",bg="#141414",border=0,

command=TreatmentCentres)

printButton7.place(x=190,y=500)

printButton8 = Button(newWindow,

text = "Statistics",fg="#25dae9",bg="#141414", border=0,

command=statistics)

printButton8.place(x=290,y=550)

#For displaying Covid-19 statistics

def statistics():

newWindow = Toplevel(root)

newWindow.title("Statistics")

newWindow.geometry("300x300")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

printButton1 = Button(newWindow,

text = "Covid-19 Latest Statistics(WHO)",fg="#25dae9",bg="#141414",

command=stats)

printButton1.place(x=50,y=20)

printButton2 = Button(newWindow,

text = "Statistics for comparision",fg="#25dae9",bg="#141414",

command=statsforcomparision)

printButton2.place(x=75,y=60)

#Statistics provided by WHO

def stats():

webbrowser.open('https://covid19.who.int/', new=2)

#For comparing statistics between different countries

def statsforcomparision():

newWindow = Toplevel(root)

newWindow.title("Get Covid-19 Data Country Wise")

newWindow.geometry("400x400")

def showdata():

from matplotlib import pyplot as plt

import matplotlib.patches as mpatches

from covid import Covid

covid = Covid()

country=[]

cases = []

confirmed = []

deaths = []

latitude=[]

longitude=[]

root.update()

countries = data.get()

country\_names=countries.strip()

country\_names=country\_names.replace(" ", ",")

country\_names=country\_names.split(",")

for x in country\_names:

cases.append(covid.get\_status\_by\_country\_name(x))

for y in cases:

country.append(y["country"])

confirmed.append(y["confirmed"])

deaths.append(y["deaths"])

latitude.append(y["latitude"])

longitude.append(y["longitude"])

confirmed\_patch = mpatches.Patch(color='green', label='confirmed')

deaths\_patch = mpatches.Patch(color='black', label='deaths')

plt.legend(handles=[confirmed\_patch, deaths\_patch])

for x in range(len(country\_names)):

plt.bar(country\_names[x], confirmed[x], color='green')

plt.bar(country\_names[x], deaths[x], color='black')

plt.title('Current Covid Cases')

plt.xlabel('Country Name')

plt.ylabel('Cases(in millions)')

plt.show()

Label(newWindow, text="Enter all countries names\nfor whom you want to get\ncovid-19 data", font="Consolas 15 bold").pack()

Label(newWindow, text="Enter country name:").pack()

data = StringVar()

data.set("Seperate country names using comma or space(not both)")

entry = Entry(newWindow, textvariable=data, width=50).pack()

Button(newWindow, text="Get Data", command=showdata).pack()

#For registration for covid vaccines

def registration():

webbrowser.open('https://selfregistration.cowin.gov.in/',

new=2)

#Providing links for helpful softwares for users

def helpfulsoftwares():

webbrowser.open('https://covid19.karnataka.gov.in/new-page/softwares/en',

new=2)

#For finding fever clinics, quarantine and isolation centers and designated hospitals in Karnataka

def TreatmentCentres():

webbrowser.open('https://covid19.karnataka.gov.in/treatment/en',

new=2)

#Creating Chatbot

def FAQ():

#Creating chat window

newWindow = Toplevel(root)

newWindow.title("Chatbot")

def send():

send="You => "+e.get()

txt.insert(END,"\n"+send)

if(e.get() in ["hello", "hi", "greetings", "sup", "what's up","hey"]):

txt.insert(END,"\n"+'''Bot => Hi. My name is Robo. I will help you with some information on Covid-19

1)For home remedies, enter 1

2)To know about preventive measures, enter 2

3)For FAQs & more information about variants, symptoms, testing & travel,

enter 3

4)To know about vaccines, enter 4

5)Guidance for managing patients with covid-19, including clinical guidance,

home & hospitl care, enter 5

6)For science & research, enter 6

7)For government of karnataka instructions, enter 8

8)For awareness, enter9

9)For fake news buster, enter 10

Enter exit once you are done''')

#Mentioning conditions

elif (e.get()=="1"):

webbrowser.open('https://www.medicalnewstoday.com/articles/coronavirus-home-remedies', new=2)

elif (e.get()=="2"):

txt.insert(END,"\n"+"Bot =>",

"The ways that you can prevent yourself from COVID-19 are:\n",

"1. Wear a suitable mask which covers your nose & mouth completely.\n",

"2. Wash your hands for atleast 20 seconds.\n",

"3. Try to reduce/avoid contact with other people.\n",

"4. Maintain a social distance with other people (preferably 6 ft distance)\n",

"5. Try to stay in open,ventilated and isolated places.\n",

"6. Get VACCINATED at the earliest from a nearby hospital.\n",

"7. Cover your nose and mouth with a tissue paper when you cough or sneeze")

elif (e.get()=="3"):

webbrowser.open('https://www.cdc.gov/coronavirus/2019-ncov/faq.html', new=2)

elif (e.get()=="4"):

webbrowser.open('https://www.cdc.gov/coronavirus/2019-ncov/vaccines/index.html', new=2)

elif (e.get()=="5"):

webbrowser.open('https://www.cdc.gov/coronavirus/2019-nCoV/hcp/index.html', new=2)

elif (e.get()=="6"):

webbrowser.open('https://www.cdc.gov/coronavirus/2019-ncov/science/science-and-research.html', new=2)

elif (e.get()=="7"):

webbrowser.open('https://covid19.karnataka.gov.in/samanya\_suchanegalu/en', new=2)

elif (e.get()=="8"):

webbrowser.open('https://covid19.karnataka.gov.in/iec/en', new=2)

elif (e.get()=="9"):

webbrowser.open('https://covid19.karnataka.gov.in/gallery/STOP%20FAKE%20NEWS/en', new=2)

elif (e.get()=="exit"):

txt.insert(END,"\n"+"Bot => Hope you found all the relevant information. Take care, bye.")

elif(e.get()=="How are you"):

txt.insert(END,"\n"+"Bot => Fine and you?")

elif (e.get()=="fine"):

txt.insert(END,"\n"+"Bot => Nice to hear")

else:

txt.insert(END,"\n"+"Bot => Sorry I didn't get it")

e.delete(0,END)

txt=Text(newWindow)

txt.grid(row=0, column=0, columnspan=2)

e=Entry(newWindow,width=100)

send=Button(newWindow, text="Send", command=send).grid(row=1,column=1)

e.grid(row=1, column=0)

#For finding nearby hospital beds

def hospitalbeds():

webbrowser.open('https://www.google.com/maps/search/hospital+beds+near+me',

new=2)

#To check eligibility for vaccination

def Eligibility():

newWindow = Toplevel(root)

newWindow.title("Eligibiltiy to get the vaccine")

newWindow.geometry("700x300")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

label=Label(newWindow,text="Enter age(Enter in terms of two digits)",

fg="#ffcc66",bg="#141414")

label.pack()

def checkage():

x1=inputtxt.get(1.0,"end-1c")

if x1 >= "15":

lbl = Label(newWindow, text = '''You are eligible for the vaccination!

Please get registered online to get a SMS on your registered mobile number

regarding the due date,place and time of vaccination''',

fg="#ffcc66",bg="#141414")

lbl.pack()

else:

lbl = Label(newWindow, text = '''You are not eligible for the vaccination.

Till then please take care of getting exposed to the virus

Stay Safe and Wear your Masks.''',

fg="#ffcc66",bg="#141414")

lbl.pack()

inputtxt = Text(newWindow,

height = 1,

width = 25)

inputtxt.pack()

printButton1 = Button(newWindow,

text = "Check", fg="#ffcc66",bg="#141414",

command = checkage)

printButton1.pack()

#Covid-19 Self-Assessment for user

def SAssessment():

newWindow = Toplevel(root)

newWindow.title("Covid-19 Self Assessment")

newWindow.geometry("700x300")

newWindow.configure(bg="#141414")

newWindow.option\_add('\*Font', 'Impact')

l1=Label(newWindow, text= "Please select one option based on your symptoms",

fg="#ffcc66",bg="#141414")

l1.pack()

def print\_selection():

if (var.get() == 1):

l.config(text='''Nothing to worry, it's just due to Air pollution

You have not contracted Covid-19,

If you feel like something is wrong, you just need to rest.

If your symptoms get worse,please consult your doctor''')

if (var.get() == 2):

l.config(text=''' Nothing to worry, it's just a common cold

You have not contracted Covid-19,

If you feel like something is wrong, you just need to rest.

If your symptoms get worse,please consult your doctor''')

if (var.get() == 3):

l.config(text='''Nothing to worry, it's just a flu

You have not contracted Covid-19,

If you feel like something is wrong, you just need to rest.

If your symptoms get worse,please consult your doctor''')

if (var.get() == 4):

l.config(text='''You have contracted Covid-19, Please follow the regulations.

Quarantine yourself,

and if the symptoms get worse please visit a nearby hospital.

Stay safe!''')

var = IntVar()

c1 = Radiobutton(newWindow, text='Dry cough/Sneeze',variable=var,value=1,

fg="#ffcc66",bg="#141414",

command=print\_selection)

c1.pack()

c2 = Radiobutton(newWindow, text='Cough/Mucus/Sneeze/Runny nose',variable=var,

value=2, fg="#ffcc66",bg="#141414",

command=print\_selection)

c2.pack()

c3 = Radiobutton(newWindow,

text='Cough/Mucus/Sneeze/Runny nose/Body ache/Weakness/Light fever',

variable=var,value=3,fg="#ffcc66",bg="#141414",

command=print\_selection)

c3.pack()

c4 = Radiobutton(newWindow,

text='Dry cough/Sneeze/Body pain/Weakness/High fever/Difficulty in breathing',

fg="#ffcc66",bg="#141414",

variable=var,value=4,command=print\_selection)

c4.pack()

l = Label(newWindow, bg="#141414", width=50, text='',fg="#f86263")

l.pack()

#Creating windows for Covid-19 Assistance

def MyClick():

newWindow = Toplevel(root)

newWindow.title("COVID-19 ASSISTANCE")

newWindow.geometry("800x500")

newWindow.configure(bg="#141414")

def bttn(x,y,text,bcolor,fcolor,font,cmd):

mybutton=Button(newWindow,width=100,height=4,text=text,fg=bcolor,

bg=fcolor,border=0,activeforeground=fcolor,

activebackground=bcolor,font=font,command=cmd)

mybutton.place(x=x,y=y)

bttn(0,85,"ADMIN","#ffcc66","#141414","Impact",openNewWindowA)

bttn(0,185,"USER","#25dae9","#141414","Impact",openNewWindowU)

bttn(0,285,"EXIT","#f86263","#141414","Impact",newWindow.destroy)

root.geometry("700x500")

load=Image.open("Doc\\COVID-19(1) bg.png")

render=ImageTk.PhotoImage(load)

img=Label(root,image=render)

img.place(x=0, y=0)

img1= PhotoImage(file="Doc\\bhomepg.png")

btn = Button(root, image=img1,bd=0,command=MyClick)

btn.place(x=400,y=200)

root.mainloop()

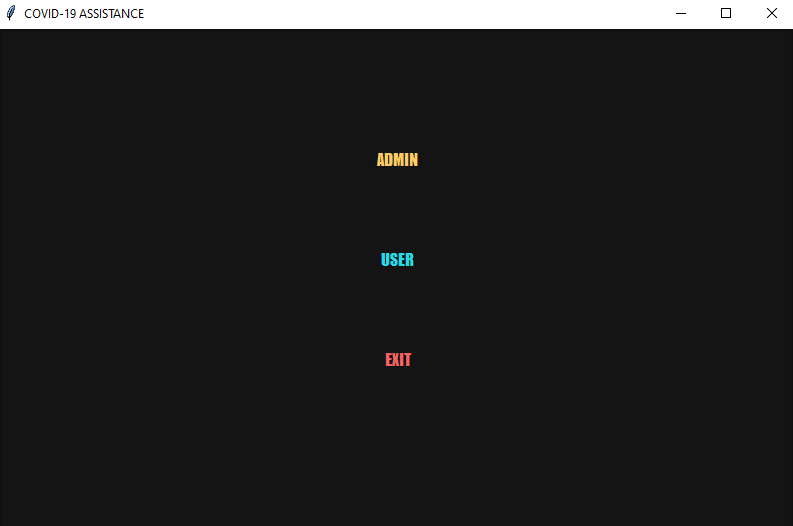
**OUTPUT**

**MAIN WINDOW:**

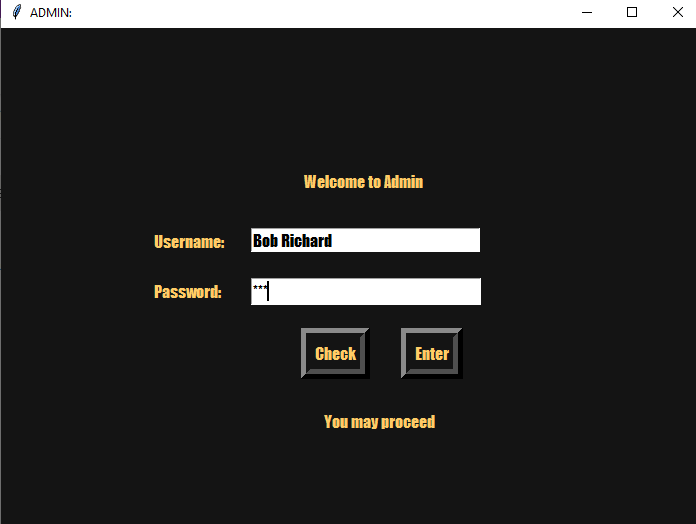




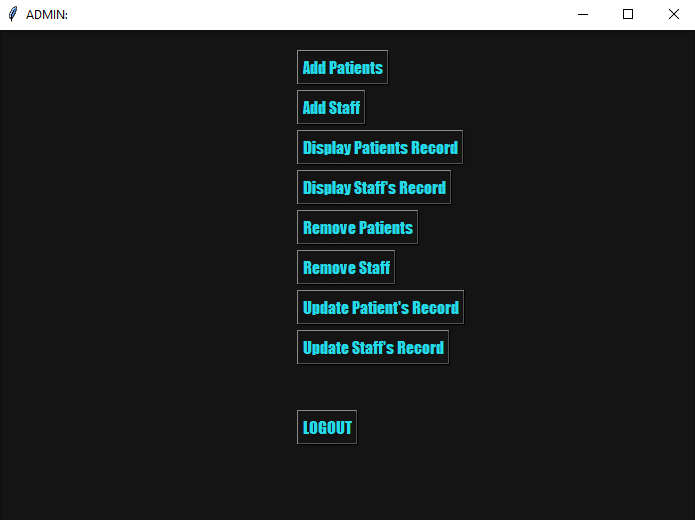
**MENU SELECTION:**



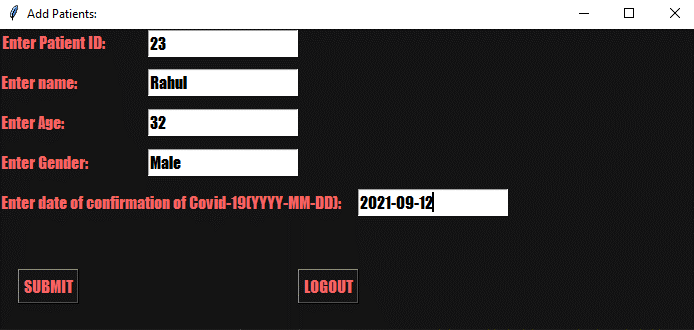
**USERNAME & PASSWORD (ADMIN):**

****

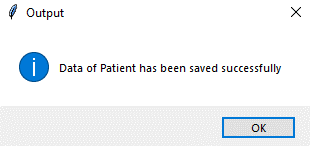
**ADMIN MENU SELECTION:**



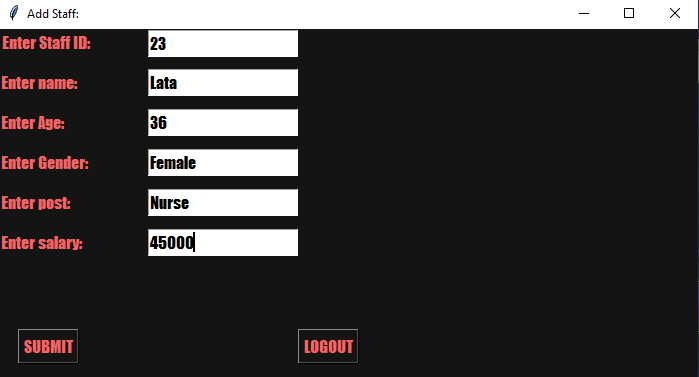
**ADD PATIENTS:**



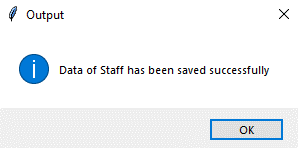
**ADD PATIENT OUTPUT:**



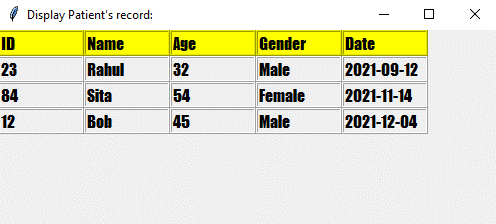
**ADD STAFF:**



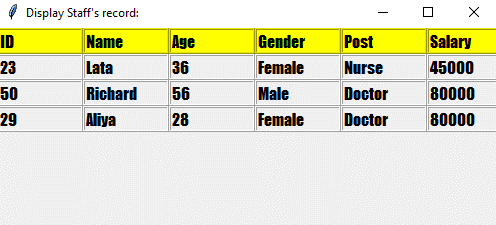
**ADD STAFF OUTPUT:**



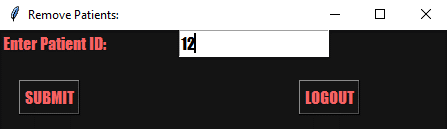
**DISPLAY PATIENT’S RECORD:**



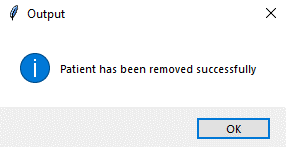
**DISPLAY STAFF’S RECORD:**



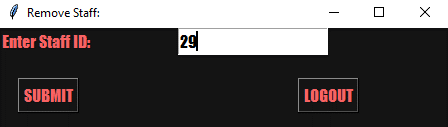
**REMOVE PATIENT:**



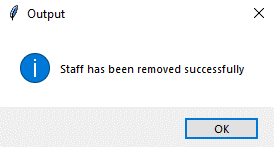
**OUTPUT:**



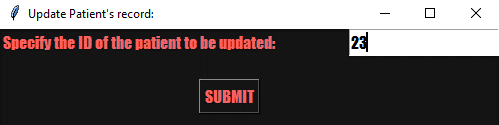
**REMOVE STAFF:**

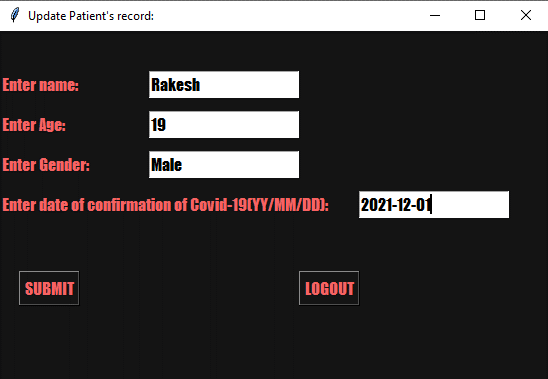


**OUTPUT:**

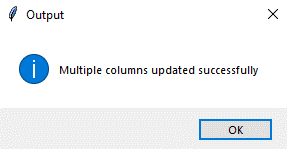


**UPDATE PATIENT’S RECORD:**

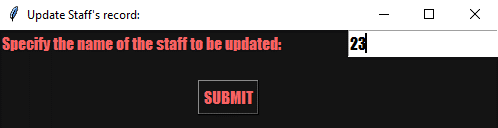


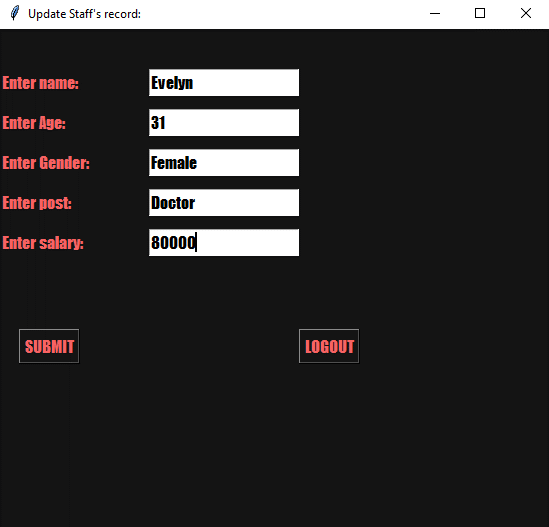


**OUTPUT:**

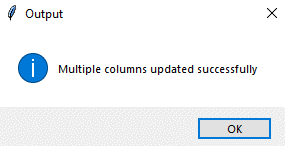


**UPDATE STAFF’S RECORD:**

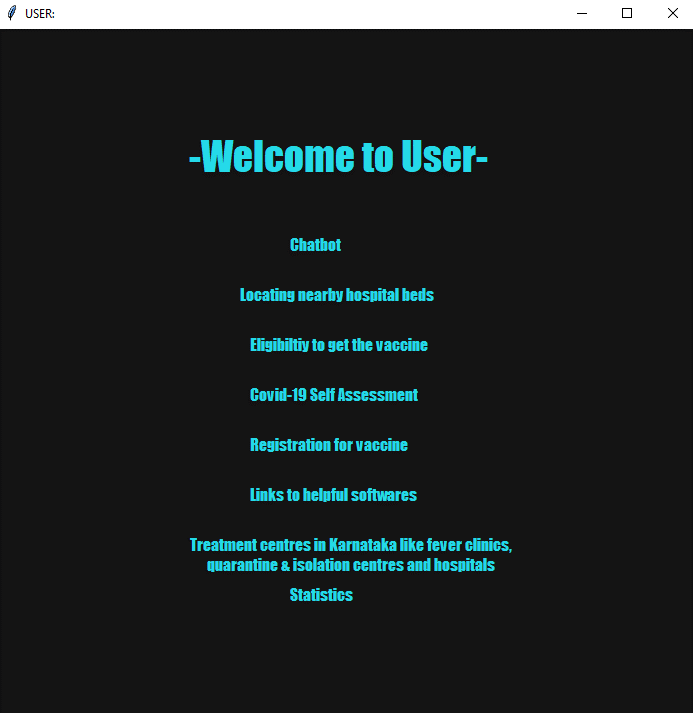




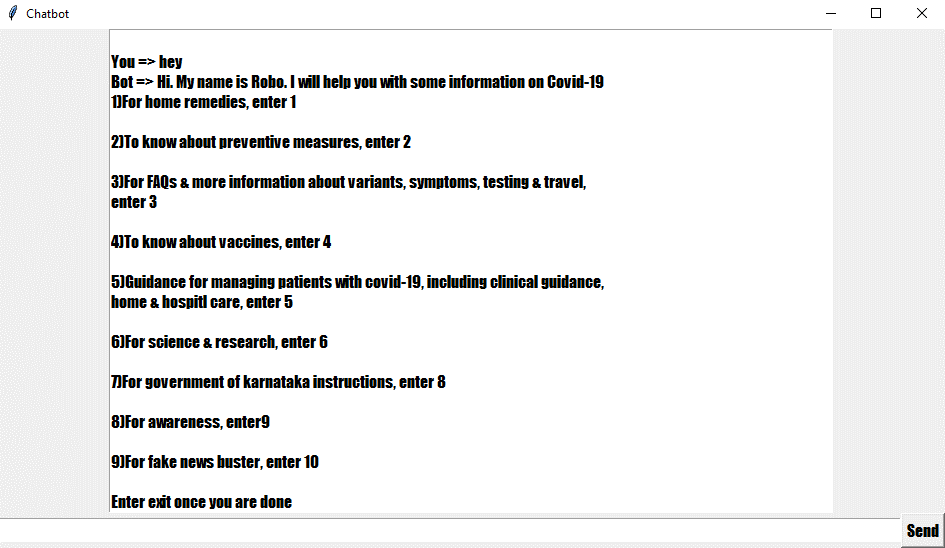
**OUTPUT:**



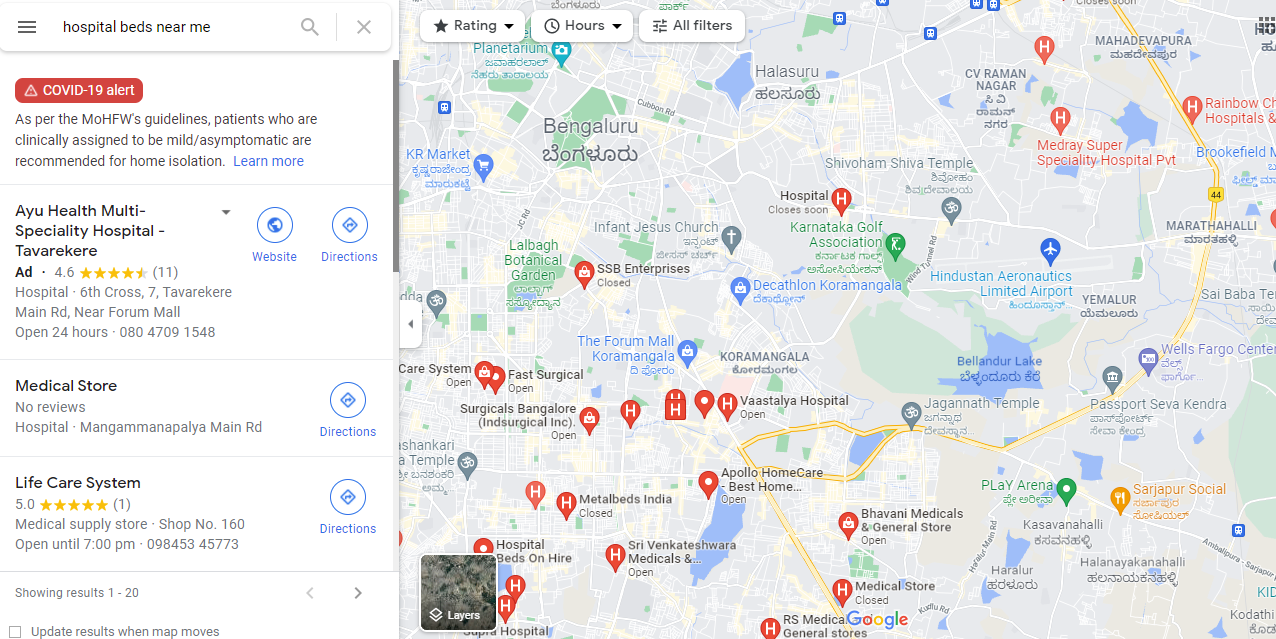
**USER MENU SELECTION:**



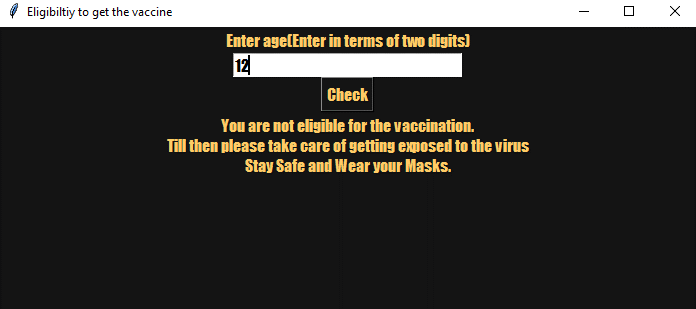
**CHATBOT:**

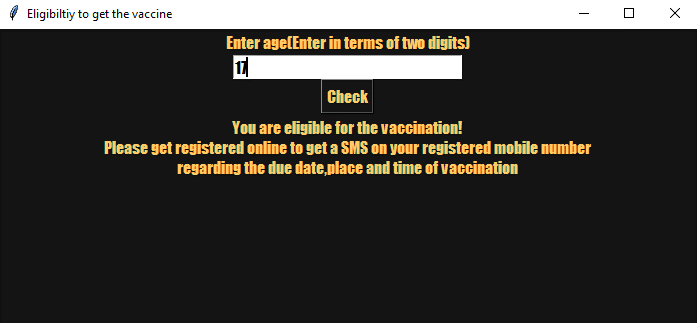


**LOCATING HOSPITAL BEDS:**

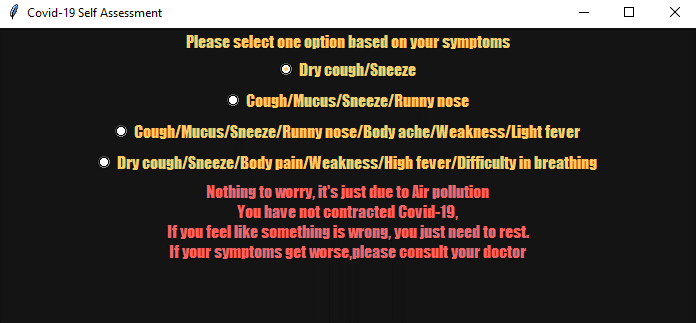


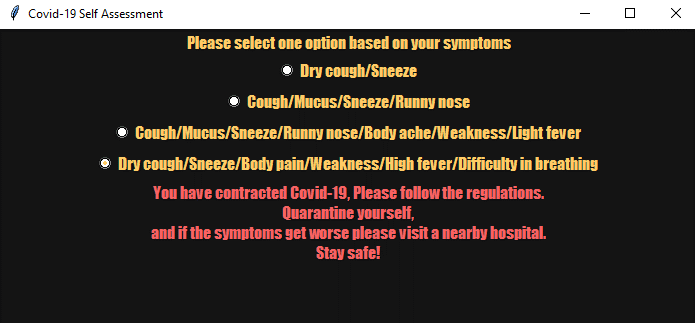
**ELIGIBILITY TO GET VACCINE:**

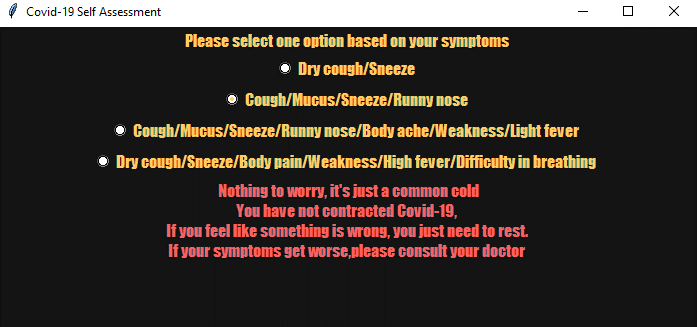


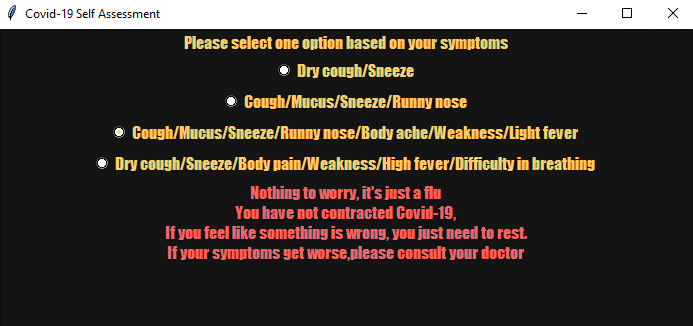


**COVID-19 SELF ASSESMENT:**

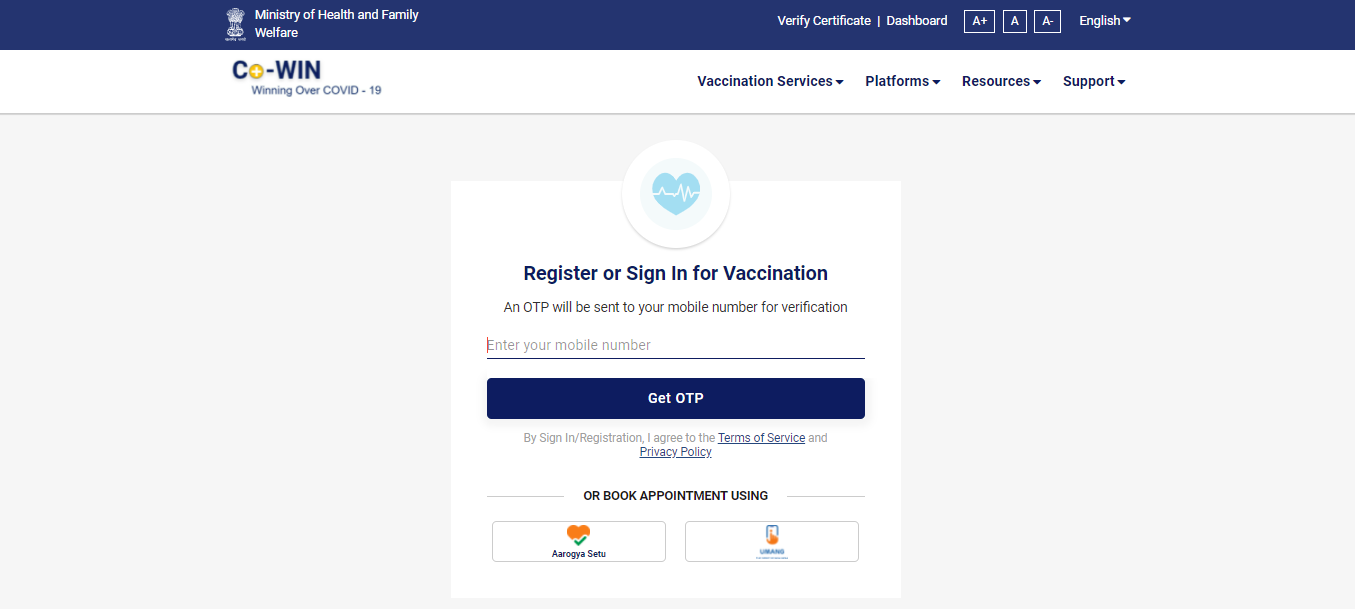




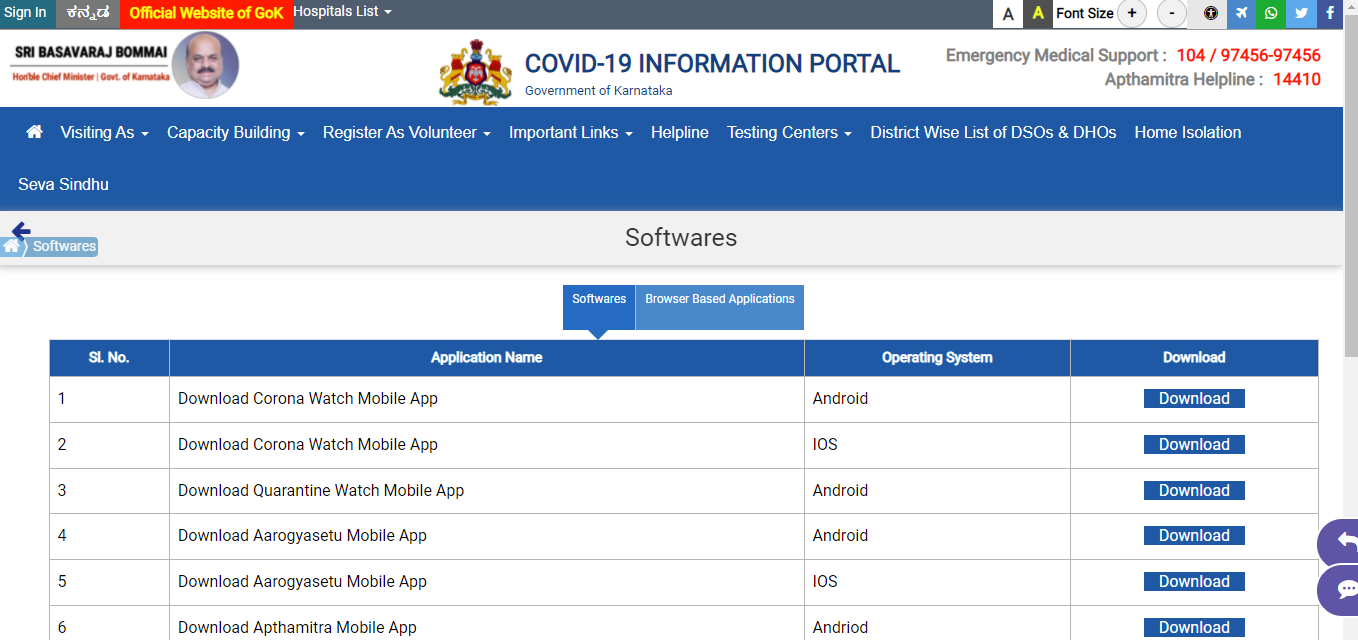




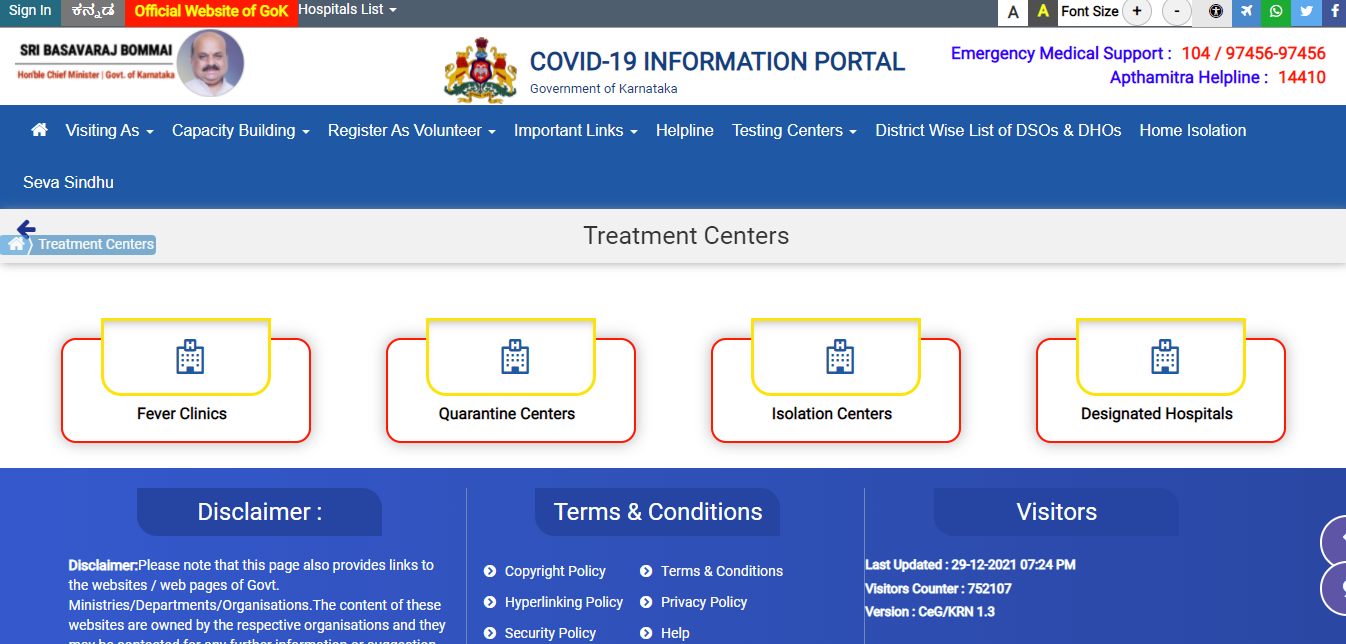
**REGISTRATION FOR VACCINE:**



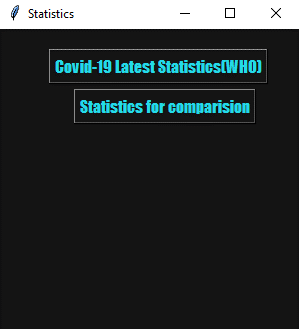
**HELPFUL SOFTWARE:**



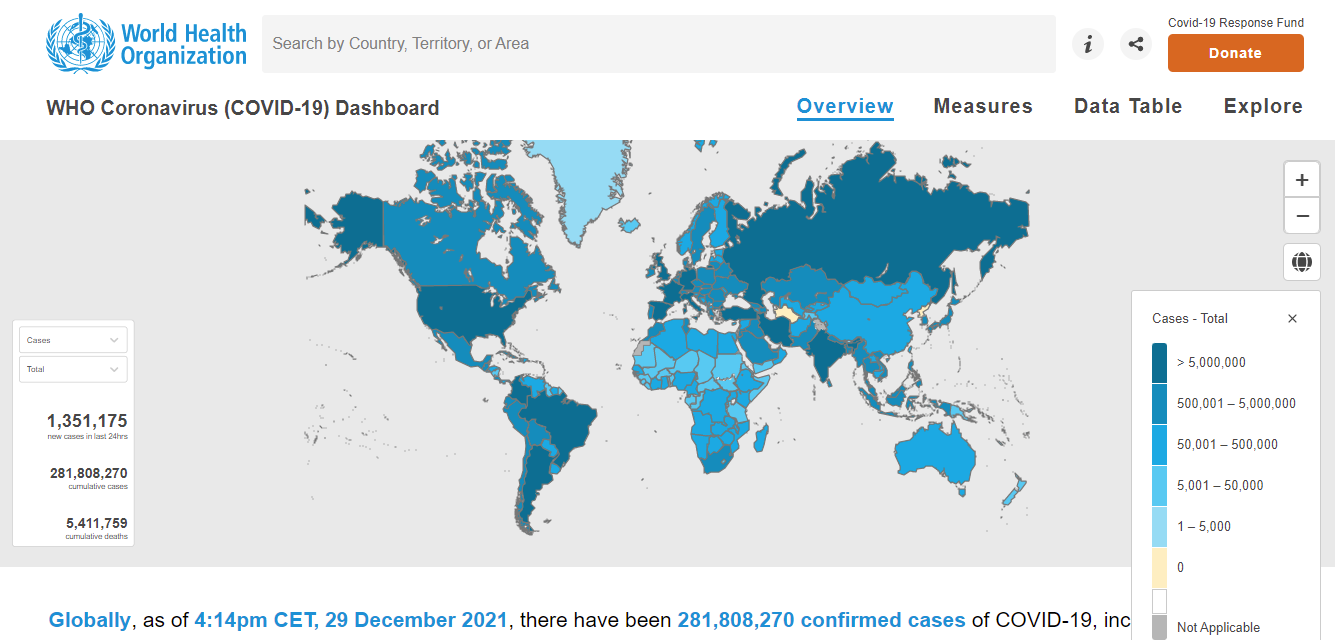
**TREATEMENT CENTERS:**

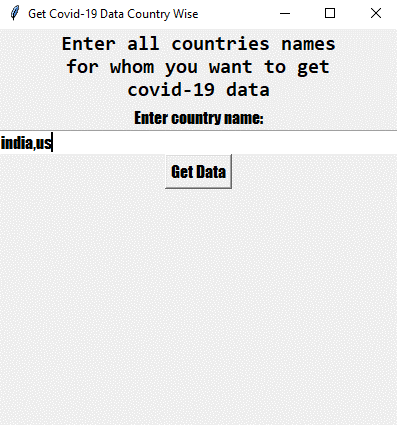


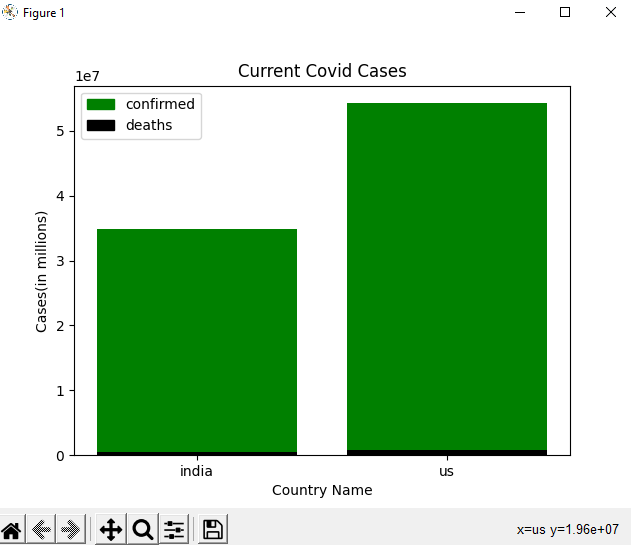
**STATISTICS MENU:**



**WHO (LATEST STATISTICS):**



**STATISTICS FOR COMPARISION:**



**FURTHER DEVELOPMENT AREA**

* **Solves a very specific problem**: As the pandemic was not expected and the government, hospitals and health centers were not prepared for it, there lacks a proper application that the hospitals and health centers can use to maintain their patients and staff record. The application built by us provides a user-friendly interface that can be used by both the hospital or an organization and the individuals who are the patients and others who just want to use it for assistance, information and monitoring.
* **Access and completion**: The Covid-19 Application provides with a database under which patients and staff records are maintained. The hospital or health center who are the main admin/user have to maintain it frequently. Sometimes in the case of emergencies, the records are not updated. As a result, the data displayed by the application will not be accurate. The data provided by the application is not synced with other hospitals or health centers.

It is not verified and it can only be used by the health centers for maintaining their personal record for their own reference.

**BIBLIOGRAPHY**

1. COVID-19 REMEDIES (GOOGLE)
2. FAQs RELATED ON COVID-19 (GOOGLE)
3. PREETI ARORA CLASS 12 COMPUTER SCIENCE WITH PYTHON
4. SUMITA ARORA CLASS 12 COMPUTER SCIENCE WITH PYTHON
5. COVIDBEDMBMC.IN

LINK: ( <https://www.covidbedmbmc.in/> )

1. COVID19.KARNATAKA.GOV.IN

LINK: ( <https://covid19.karnataka.gov.in/english> )

1. SELFREGISTRATION.COWIN.GOV.IN

LINK: ( <https://selfregistration.cowin.gov.in/> )

1. COVID19.WHO.INT

LINK: ( <https://covid19.who.int/> )