Mini-Project Report
Advanced Programming Lab (ICT 3166)
ICT DEPARTMENT

Project Title: Vaccination Database System

Team Members: Aarush Dua, Anushree Bhat, Khushi Sinha
CCE Batch – 1

Sl#	Full Name	Reg#	Roll#	CCE section_name
1	Aarush Dua	190953037	14	A
2	Anushree Bhat	190953043	16	A
3	Khushi Sinha	190953005	3	A

INTRODUCTION

The world is in the midst of a COVID-19 pandemic. As WHO and partners work together on the response -- tracking the pandemic, advising on critical interventions, distributing vital medical supplies to those in need- they are racing to develop and deploy safe and effective vaccines.

In times of Covid-19, it is important for an immunization provider to ensure the safety and efficacy of vaccines. Vaccine administrating, observation of precautions, management and reporting of vaccine side effects etc, must be recorded.

However, many hospitals, such as the rural hospitals in India, lack the IT facilities to do so. Our project aims to solve this issue with a user friendly, basic registration system built using python concepts.

LITERATURE SURVEY

To have a better understanding of the modules used in our project, we used the following python documentation:

1. https://docs.python.org/3/library/tkinter.html

A graphics-based operating system interface that uses icons, menus and a mouse (to click on the icon or pull down the menus) to manage interaction with the system. The tkinter package ("Tk interface") is the standard Python interface to the Tcl/Tk GUI toolkit.

Tcl

Tcl is a dynamic interpreted programming language, just like Python. Though it can be used on its own as a general-purpose programming language, it is most commonly embedded into C applications as a scripting engine or an interface to the Tk toolkit.

Tk

Tk is a Tcl package implemented in C that adds custom commands to create and manipulate GUI widgets. Each Tk object embeds its own Tcl interpreter instance with Tk loaded into it. Tk's widgets are very customizable.

Ttk

Themed Tk (Ttk) is a newer family of Tk widgets that provide a much better appearance on different platforms than many of the classic Tk widgets.

2. https://docs.python.org/3/library/smtplib.html

Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers.

Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

We imported smptlib for the otp verification portal in our database system. When an individual registers using their email address and password, an OTP is sent to the entered email address. This was done to incorporate security features in the project.

METHODOLOGY

Multiple concepts of Python have been used to design the Covid-19 Database System. Broad description of the concepts used is: -

1.Database Connectivity: We used a module named SQLite 3 in SQLite to create a database. SQLite 3 is a self-contained, file-based SQL database. It comes bundled with Python and can be used in any of the Python applications without having to install any additional software.

2.GUI using Tkinter: It is a framework that is built into the Python Standard Library. t is famous for its simplicity and graphical user interface. It is open-source and available under the Python License. Certain widgets like Buttons e.g.: Login Button, were created using Tkinter.

3. Modules: For reusability of certain codes and organization of the python files, module concept was used. These modules are imported into the code when required, e.g.: User defined modules like **Menupage** and **function** and predefined modules like **math**, **smtplib** and **random**.

IMPLEMENTATION

Listed below are the detailed steps behind building the database.

1. SQLite 3 has been imported to create a database. Two tables have been created:

(a) To store Patient Information

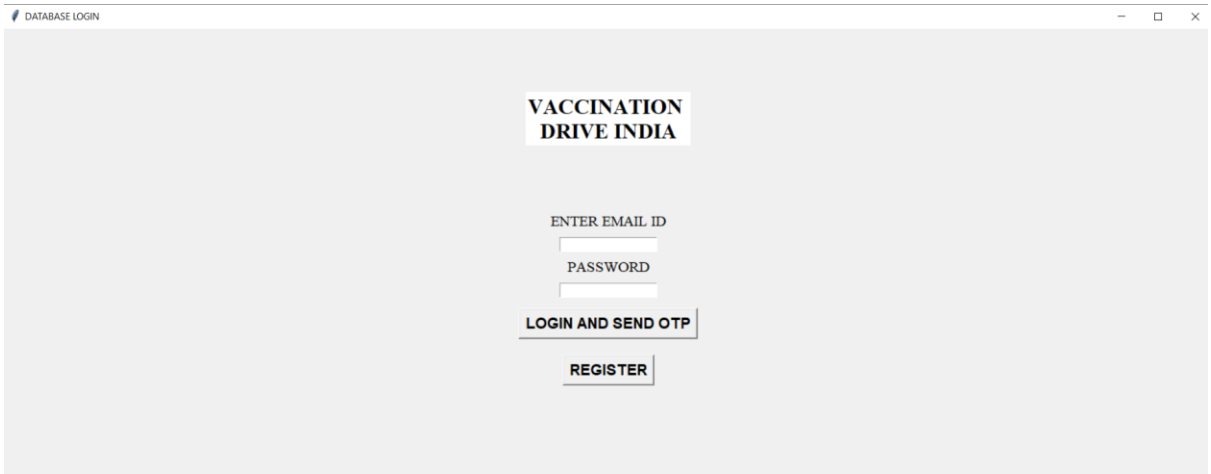
(b) To store Covid-19 Immunization Record of the Patient.

database.py - C:\Users\mksin\Downloads\Vaccination Management\Vaccination Manageme...

File Edit Format Run Options Window Help

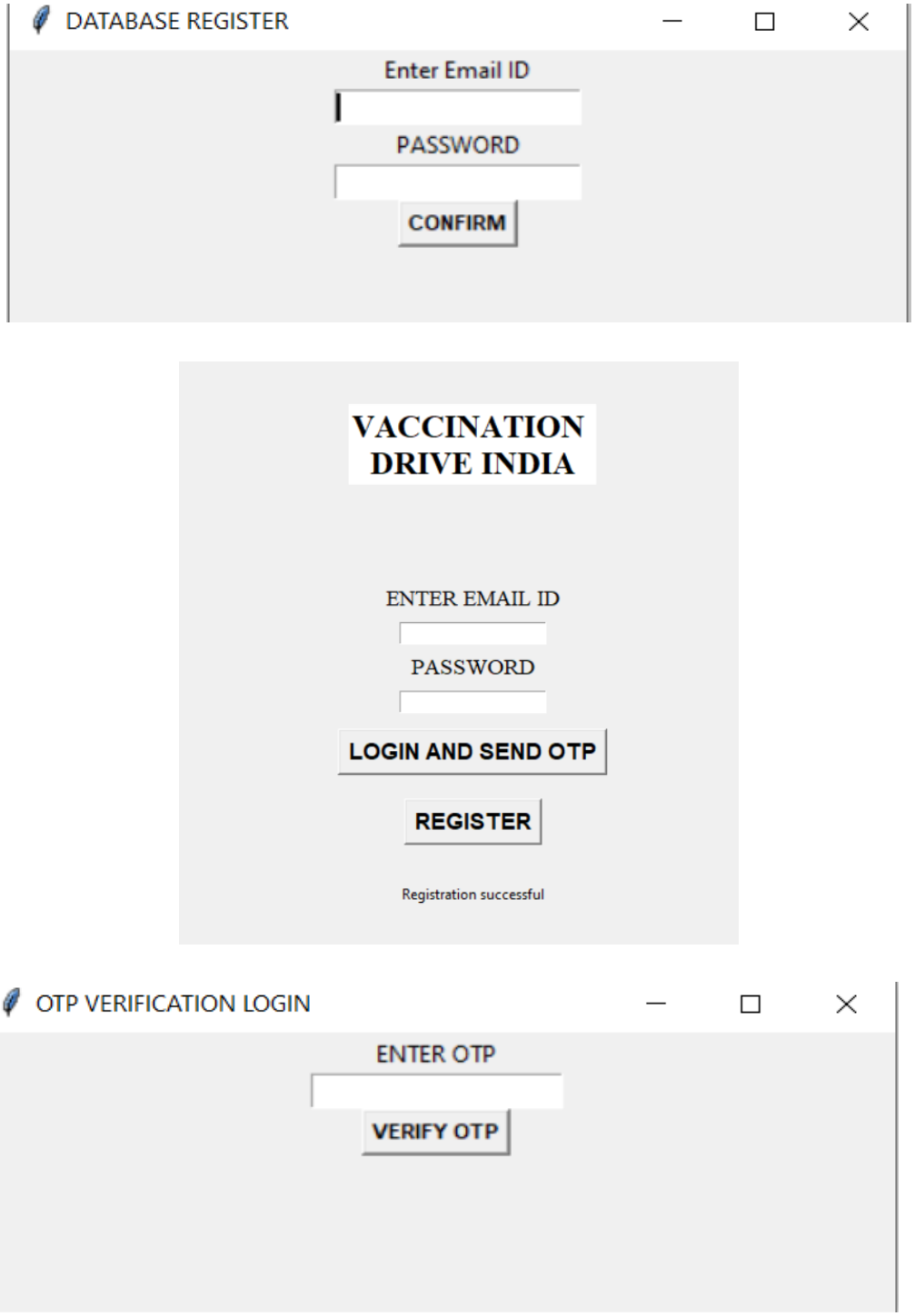
```
import sqlite3
conn=sqlite3.connect("vaccinedatabase.db")
print("DATABASE CONNECTION SUCCESSFUL")


conn.execute("Drop table if EXISTS PATIENT")
conn.execute("Drop table if EXISTS CONTACT_NO")
conn.execute("Drop table if EXISTS ROOM")
conn.execute("Drop table if EXISTS TREATMENT")
conn.execute("Drop table if EXISTS MEDICINE")

conn.execute("""Create table PATIENT
          (COWIN_ID int(10) primary key,
          NAME VARCHAR(20) not null,
          SEX varchar(10) not null,
          BLOOD_GROUP varchar(5) not null,
          DOB date not null,
          SIDE_EFFECTS varchar(100) not null,
          CONSULT_TEAM varchar(50) not null,
          CONTACT_NO varchar(20) not null    )""")
print("TABLE CREATED SUCCESSFULLY")

conn.execute("""CREATE TABLE COVID
             (COWIN_ID int(10) PRIMARY KEY,
              NODOSES int(15) not null,
              PRESCRIPTION varchar(15),
              FOREIGN KEY(COWIN_ID) REFERENCES PATIENT(COWIN_ID))
              """)
print("TABLE CREATED SUCCESSFULLY")
```

(a) Database Connection

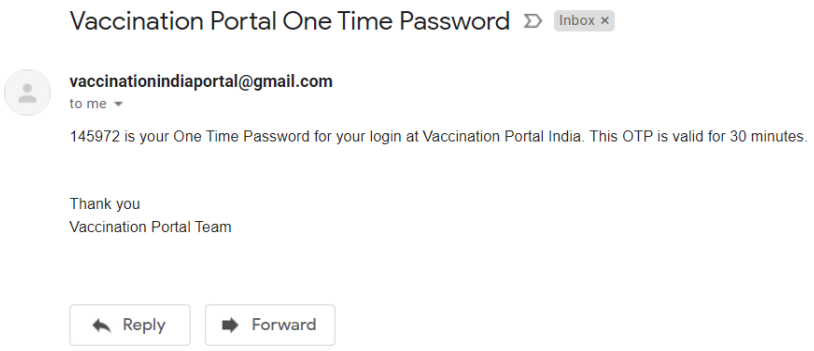


(b) Database Login - UI

2. The user first needs to register by clicking the REGISTER button. Once registered, a label " registration successful" appears.To incorporate security and authentication for Login, we have integrated an **OTP** (One Time Password) System. Upon entering Login details, OTP Verification Login button will show. The OTP will be sent to the email address used as username when logging in.



(a) OTP Verification



(b) OTP sent to email address

The **login.py file** contains the code for login page and OTP with the username and password present in the file.

```
*login.py - C:\Users\anush\OneDrive\Desktop\Vaccination Management\Vaccination Management\login.py (3.10.0)*
File Edit Format Run Options Window Help
import tkinter
from tkinter import *
from Menupage import menu
import math
import random
import smtplib
from email.message import EmailMessage

root=None
root3=None
otpbox=None
userbox=None
passbox=None
topframe=None
bottomframe=None
frame3=None
login=None
reg=None

users = {"username":"password"}

digits="0123456789"
OTP=""
for i in range(6):
    OTP+=digits[math.floor(random.random()*10)]
otp = OTP + " is your One Time Password for your login at Vaccination Portal India. This OTP is valid for 30 minutes.\n\n\nThank you\nVaccination Portal Team"
hey = otp
msg = EmailMessage()
msg.set_content(otp)

msg['Subject'] = 'Vaccination Portal One Time Password'


#command for login button
def GET():

    global userbox,passbox,loginverifylabel
    S1=userbox.get()
    S2=passbox.get()
    frame = Frame()
    frame.pack(pady=3)
    loginverifylabel.pack(fill=BOTH)
    if S1 in users.keys():
        if S2 == users[S1]:
            server = smtplib.SMTP_SSL("smtp.gmail.com", 465)
            server.login("vaccinationindiaportal@gmail.com", "vaccination@2021")
            msg['From'] = "vaccinationindiaportal@gmail.com"
            msg['To'] = S1
            server.send_message(msg)
            server.quit()
            Entry1() #if username matches password

        else:
            loginverifylabel["text"] = "Invalid username or password ! "
    else:
        loginverifylabel["text"] = "Invalid username or password ! "
```

```
def Entry1():
    global otpbox, verify, topframe, bottomframe, image_1,heading
    root3 = tkinter.Tk()
    root3.geometry("450x150")
    topframe = tkinter.Frame(root3)
    topframe.pack()
    bottomframe = tkinter.Frame(root3)
    bottomframe.pack()
    heading = tkinter.Label(topframe, text="WELCOME TO VACCINATION DRIVE INDIA", bg='white', fg='orange',font='Times 20 bold italic')
    otp = tkinter.Label(bottomframe, text="ENTER OTP")
    otpbox = tkinter.Entry(bottomframe, show="*")
    login1 = tkinter.Button(bottomframe, text="VERIFY OTP", command=GET1, font="arial 8 bold")
    otp.pack()
    otpbox.pack()
    login1.pack()
    root3.title("OTP VERIFICATION LOGIN")
    root3.mainloop()

def GET1():
    global otpbox,error
    $3 = str(otpbox.get())
    if($3==OTP):
        menu()
    else:
        error=tkinter.Label(bottomframe,text="Wrong Id / Password \n TRY AGAIN",fg="red",font="bold")
        error.pack()

def register():
    global regbox,reguserbox,passwbox,regbox,topframe,bottomframe,registerverifylabel
    root4 = tkinter.Tk()
    root4.geometry("450x150")
    topframe = tkinter.Frame(root4)
    topframe.pack()
    bottomframe = tkinter.Frame(root4)
    bottomframe.pack()
    heading = tkinter.Label(topframe, text="WELCOME TO VACCINATION DRIVE INDIA", bg='white', fg='orange',font='Times 20 bold italic')
    reguser = tkinter.Label(topframe, text="Enter Email ID")
    regbox = tkinter.Entry(topframe)
    passw = tkinter.Label(bottomframe, text="PASSWORD")
    passwbox = tkinter.Entry(bottomframe, show="*")
    regis = tkinter.Button(bottomframe, text="CONFIRM", command=registerverify, font="arial 8 bold")
    reguser.pack()
    regbox.pack()
    passw.pack()
    passwbox.pack()
    regis.pack()
    root4.title("DATABASE REGISTER")
    registerverifylabel = Label(reg)
    registerverifylabel.pack()
    root4.mainloop()

def registerverify():
    if regbox.get() in users:
        frame=Frame()
        frame.pack(pady=2)
        registerverifylabel["text"]="This Username Is Taken"
    else:
        users[regbox.get()] = passwbox.get()
        frame=Frame()
        frame.pack(pady=2)
        registerverifylabel["text"]="Registration successful"


#LOGIN PAGE WINDOW
def Entry():
    global userbox,passbox,otpbox,login,topframe,bottomframe,image_1,loginverifylabel
    root = tkinter.Tk()
    width= root.winfo_screenwidth()
    height= root.winfo_screenheight()
    root.geometry("%dx%d" % (width, height))
    heading = tkinter.Label(topframe, text="VACCINATION \nDRIVE INDIA",bg='white',fg='Black',font='Times 20 bold')
    heading.place(x=width, y=height, anchor="center")
    username=tkinter.Label(topframe,text="ENTER EMAIL ID",font="Times 14")
    userbox = tkinter.Entry(topframe)
    password=tkinter.Label(bottomframe,text="PASSWORD",font="Times 14")
    passbox = tkinter.Entry(bottomframe,show="*")
    login = tkinter.Button(bottomframe, text="LOGIN AND SEND OTP", command=GET,font="arial 14 bold")
    reg = tkinter.Button(bottomframe, text="REGISTER", command=register,font="arial 14 bold")
    heading.pack(pady=80)
    username.pack(pady=3)
    userbox.pack(pady=3)
    password.pack(pady=3)
    passbox.pack(pady=3)
    login.pack(pady=10)
    reg.pack(pady=10)
    root.title("DATABASE LOGIN")
    loginverifylabel = Label(root)
    loginverifylabel.pack()
    root.mainloop()

Entry()
```

3. Once verified, the login will be a success and the **menu portal** will show.

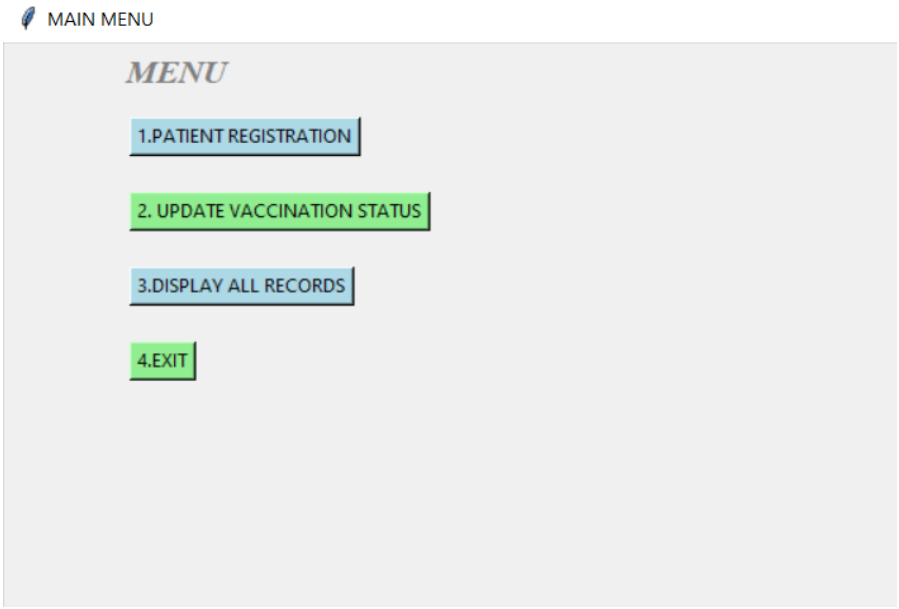
   The Menu portal is divided into 4 sections:

(a) **Patient Registration –** COWIN ID, Name, Patient Name, Sex, DOB, Blood Group, Dose Number, Prescription, Contact Number, Consulting Team/Doctor, Side Effects.

(b) **Update Vaccination** – The above information can be updated as required.

(c) **Display all records** – The above information can be displayed, fetching records of the patient.

(d) **Exit**



(a) Menu Portal

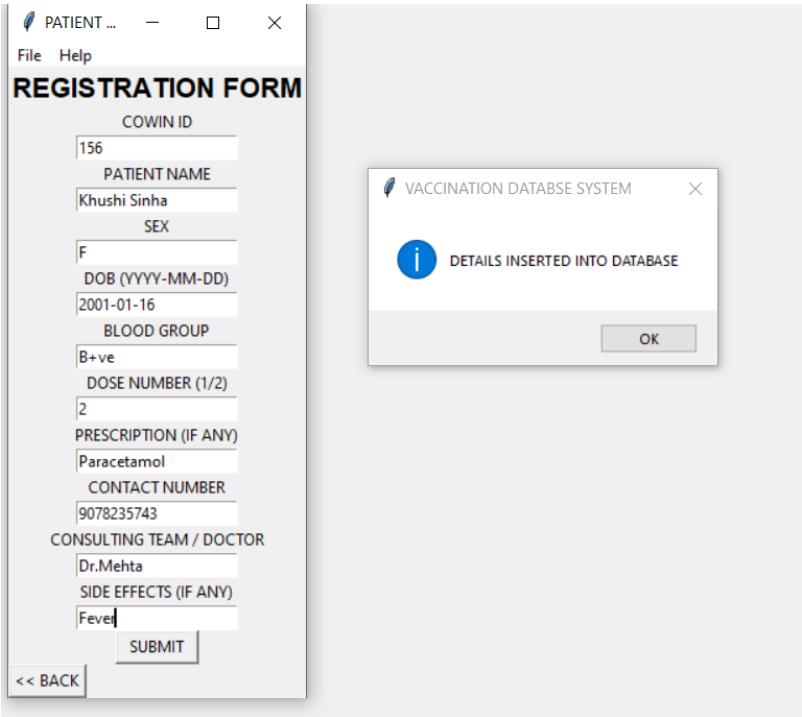
The **Menupage.py** file contains the code for menu, menu buttons and patient forms.

```
import tkinter
import sqlite3
import tkinter.messagebox
from function import P_display
from function import P_UPDATE

conn=sqlite3.connect("vaccinedatabase.db")
print("DATABASE CONNECTION SUCCESSFUL")

#variables
root1=None
rootp=None
pat_ID=None
pat_name=None
pat_dob=None
pat_effects=None
pat_sex=None
pat_BG=None
pat_contact=None
pat_doses=None
pat_pres=None
pat_CT=None


#EXIT for MENU
def ex():
    global root1
    root1.destroy()

#MENU BUTTONS
def menu():
    global root1,button1,button2,button3,button4,button5,m,button6
    root1=tkinter.Tk()
    width= root1.winfo_screenwidth()
    height= root1.winfo_screenheight()
    root1.geometry("%dx%d" % (width, height))
    root1.title("MAIN MENU")
    m=tkinter.Label(root1,text="MENU",font='Times 16 bold italic',fg='grey')
    button1=tkinter.Button(root1,text="1.PATIENT REGISTRATION",command=PAT,bg='light blue',fg='black')
    button2 = tkinter.Button(root1, text="2. UPDATE VACCINATION STATUS",bg='light green',fg='black',command=P_UPDATE)
    button3 = tkinter.Button(root1, text="3.DISPLAY ALL RECORDS",bg='light blue',fg='black',command=P_display)
    button4 = tkinter.Button(root1, text="4.EXIT",bg='light green',fg='black',command=ex)
    m.place(x=75,y=5)
    button1.pack(side=tkinter.TOP)
    button1.place(x=80,y=50)
    button2.pack(side=tkinter.TOP)
    button2.place(x=80,y=100)
```

```
    button3.pack(side=tkinter.TOP)
    button3.place(x=80,y=150)
    button4.pack(side=tkinter.TOP)
    button4.place(x=80, y=200)
    root1.mainloop()

p=None
#input patient form
def IN_PAT():
    global pp1, pp2, pp3, pp4, pp5, pp6, pp7, pp8, pp9, pp10,ce1,conn
    conn=sqlite3.connect("vaccinedatabase.db")
    conn.cursor()
    pp1=pat_ID.get()
    pp2=pat_name.get()
    pp3=pat_sex.get()
    pp4=pat_BG.get()
    pp5=pat_dob.get()
    pp6=pat_dose.get()
    pp7=pat_pres.get()
    pp8=pat_effects.get()
    pp9=pat_CT.get()
    pp10=pat_contact.get()
    conn.execute('INSERT INTO PATIENT VALUES(?,?,?,?,?,?,?,?)',(pp1,pp2,pp3,pp4,pp5,pp8,pp9,pp10,))
    conn.execute('INSERT INTO COVID VALUES (?,?,?)',(pp1,pp6,pp7,))
    tkinter.messagebox.showinfo("VACCINATION DATABSE SYSTEM","DETAILS INSERTED INTO DATABASE")
    conn.commit()


#exit from patient form
def EXO():
    rootp.destroy()

#function for patient form help
def nothing():
    print("CONTACT DATABASE HEAD : Hospitalmanagement@gmail.com 8973465732 ")

def nothing1():
    print("MADE BY AARUSH,ANUSHREE AND KHUSHI")
#PATIENT FORM
back=None
SEARCH=None
DELETE=None
UPDATE=None

def PAT():
    global pat_effects, pat_BG, pat_contact, pat_pres, pat_dose, pat_dob, pat_contact, pat_ID, pat_name, pat_sex,pat_CT
    global rootp,regform,id,name,dob,sex,contact,dose,effects,ct,pres,bg,SUBMIT,menubar,filemenu,back,SEARCH,DELETE,UPDATE
    rootp=tkinter.Tk()
    rootp.title("PATIENT VACCINATION FORM")
    menubar=tkinter.Menu(rootp)
    filemenu=tkinter.Menu(menubar,tearoff=0)
    filemenu.add_command(label="NEW",command=PAT)
    filemenu.add_separator()
    filemenu.add_command(label="EXIT", command=EXO)
    helpmenu=tkinter.Menu(menubar,tearoff=0)
    helpmenu.add_command(label="HELP",command=nothing)
    helpmenu.add_command(label="ABOUT",command=nothing1)
    menubar.add_cascade(label="File", menu=filemenu)
    menubar.add_cascade(label="Help", menu=helpmenu)
    rootp.config(menu=menubar)
    regform=tkinter.Label(rootp,text="REGISTRATION FORM",font="Arial 16 bold")
    id=tkinter.Label(rootp,text="COWIN ID")
    pat_ID=tkinter.Entry(rootp)
    name=tkinter.Label(rootp,text="PATIENT NAME")
    pat_name = tkinter.Entry(rootp)
    sex=tkinter.Label(rootp,text="SEX")
    pat_sex=tkinter.Entry(rootp)
    dob=tkinter.Label(rootp, text="DOB (YYYY-MM-DD)")
    pat_dob=tkinter.Entry(rootp)
    bg=tkinter.Label(rootp, text="BLOOD GROUP")
    pat_BG=tkinter.Entry(rootp)
    dose=tkinter.Label(rootp, text="DOSE NUMBER (1/2)")
    pat_dose=tkinter.Entry(rootp)
    pres=tkinter.Label(rootp, text="PRESCRIPTION (IF ANY)")
    pat_pres=tkinter.Entry(rootp)
    contact=tkinter.Label(rootp, text="CONTACT NUMBER")
    pat_contact = tkinter.Entry(rootp)
    ct=tkinter.Label(rootp,text="CONSULTING TEAM / DOCTOR")
    pat_CT=tkinter.Entry(rootp)
    effects=tkinter.Label(rootp, text="SIDE EFFECTS (IF ANY)")
    pat_effects=tkinter.Entry(rootp)
    back=tkinter.Button(rootp,text="<< BACK",command=menu)
    SUBMIT=tkinter.Button(rootp,text="  SUBMIT  ",command=IN_PAT,)
    regform.pack()
    id.pack()
```

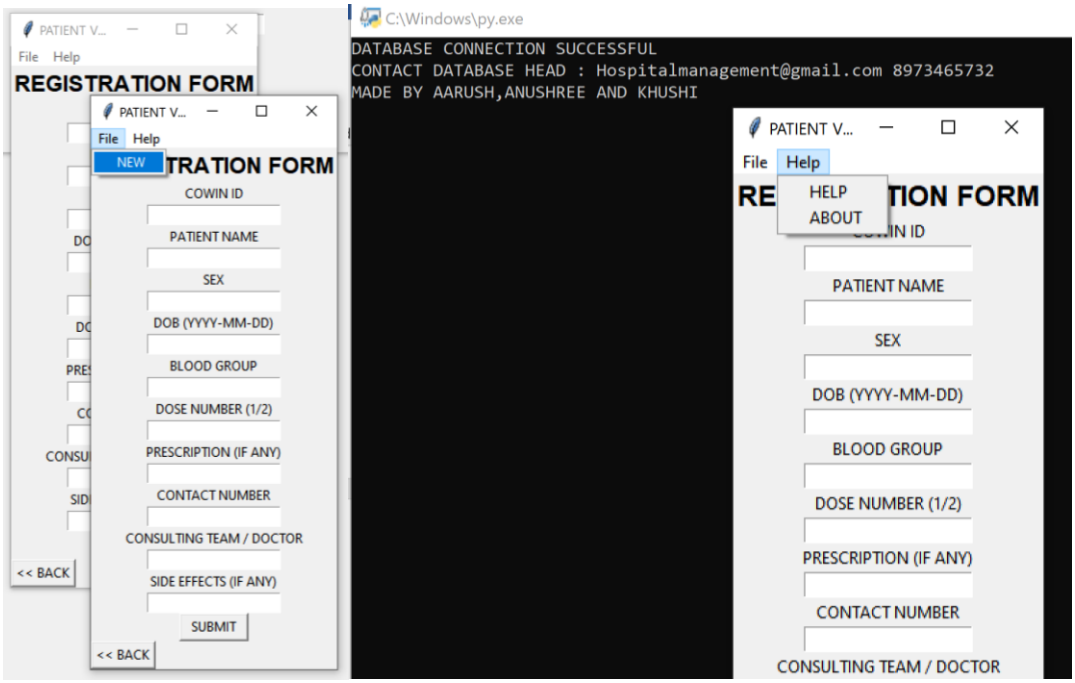
```
pat_ID.pack()
name.pack()
pat_name.pack()
sex.pack()
pat_sex.pack()
dob.pack()
pat_dob.pack()
bg.pack()
pat_BG.pack()
dose.pack()
pat_dose.pack()
pres.pack()
pat_pres.pack()
contact.pack()
pat_contact.pack()
ct.pack()
pat_CT.pack()
effects.pack()
pat_effects.pack()
SUBMIT.pack()
back.pack(side=tkinter.LEFT)
rootp.mainloop()
```

4. To **Register** for the Vaccine, one would click one Patient Registration Option. A Registration form would show. When details are entered, it is saved to the database as shown below.



a) Registration form

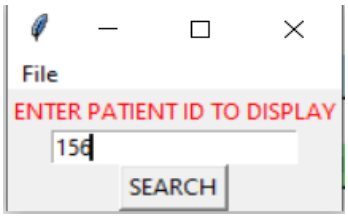
The menubar in the registration form is to make a new registration form and for the help and about functionalities. The help button prints the contact information and the about functionality here, shows who made the database system.



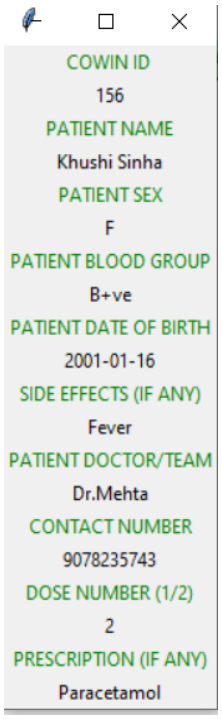
b) Menubar in the registration form

5. To **Update** or **Display** the records, the respective options may be selected. For Example, If one wanted to display the previously registered form, the 'Display All Records' button may be clicked and it will appear as shown below. The originally entered COWIN ID (here: Patient ID) must be entered.

.



(a)Display Button



(b)Display of the records

The **function.py** file contains the code for the search, update and display of patient records

```
import tkinter
import sqlite3
import tkinter.messagebox
conn=sqlite3.connect("vaccinedatabase.db")
#variables
rootU=None
rootD=None
rootS=None
head=None
inp_s=None
searchB=None
#display/search button

def Search_button():
    global inp_s,entry,errorS,t,i,q,dis1,dis2,dis3,dis4,dis5,dis6,dis7,dis8,dis9,dis10
    global l1,l2,l3,l4,l5,l6,l7,l8,l9,l10
    c1=conn.cursor()
    inp_s=entry.get()
    rootS = tkinter.Tk()
    width= rootS.winfo_screenwidth()
    height= rootS.winfo_screenheight()
    p=list(c1.execute('select * from PATIENT where COWIN_ID=?',(inp_s,)))
    if (len(p)==0):
        errorS=tkinter.Label(rootS,text="PATIENT RECORD NOT FOUND")
        errorS.pack()
    else:
        t=c1.execute('SELECT * FROM PATIENT NATURAL JOIN COVID where COWIN_ID=?',(inp_s,));
        for i in t:
            l1=tkinter.Label(rootS,text="COWIN ID ",fg='green')
            dis1=tkinter.Label(rootS,text=i[0])
            l2=tkinter.Label(rootS,text="PATIENT NAME",fg='green')
            dis2=tkinter.Label(rootS,text=i[1])
            l3=tkinter.Label(rootS,text="PATIENT SEX",fg='green')
            dis3=tkinter.Label(rootS,text=i[2])
            l4=tkinter.Label(rootS,text="PATIENT BLOOD GROUP",fg='green')
            dis4=tkinter.Label(rootS,text=i[3])
            l5=tkinter.Label(rootS,text="PATIENT DATE OF BIRTH",fg='green')
            dis5=tkinter.Label(rootS,text=i[4])
            l6=tkinter.Label(rootS,text="SIDE EFFECTS (IF ANY)",fg='green')
            dis6=tkinter.Label(rootS,text=i[5])
            l7=tkinter.Label(rootS,text="PATIENT DOCTOR/TEAM",fg='green')
            dis7=tkinter.Label(rootS,text=i[6])
            l8=tkinter.Label(rootS,text="CONTACT NUMBER",fg='green')
            dis8=tkinter.Label(rootS,text=i[7])
            l9=tkinter.Label(rootS,text="DOSE NUMBER (1/2)",fg='green')
            dis9=tkinter.Label(rootS,text=i[8])
            l10=tkinter.Label(rootS,text="PRESCRIPTION (IF ANY)",fg='green')
```

```
            dis10=tkinter.Label(rootS,text=r[9])
            l1.pack()
            dis1.pack()
            l2.pack()
            dis2.pack()
            l3.pack()
            dis3.pack()
            l4.pack()
            dis4.pack()
            l5.pack()
            dis5.pack()
            l6.pack()
            dis6.pack()
            l7.pack()
            dis7.pack()
            l8.pack()
            dis8.pack()
            l9.pack()
            dis9.pack()
            l10.pack()
            dis10.pack()
            conn.commit()


def eXO():
    rootS.destroy()

##search window
def P_display():
    global rootS,head,inp_s,entry,searchB
    rootS=tkinter.Tk()
    rootS.title("DISPLAY")
    head=tkinter.Label(rootS,text="ENTER PATIENT ID TO DISPLAY",fg="red")
    entry=tkinter.Entry(rootS)
    searchB=tkinter.Button(rootS,text='SEARCH',command=Search_button)
    menubar= tkinter.Menu(rootS)
    filemenu = tkinter.Menu(menubar, tearoff=0)
    filemenu.add_command(label="NEW", command=P_display)
    filemenu.add_separator()
    filemenu.add_command(label="EXIT", command=eXO)
    menubar.add_cascade(label="File", menu=filemenu)
    rootS.config(menu=menubar)
    head.pack()
    entry.pack()
    searchB.pack()
    rootS.mainloop()
```

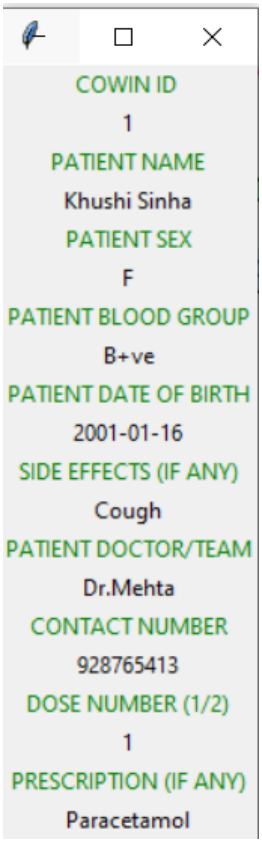
```
inp_d=None
entry1=None
errorD=None
disd1=None



##variables for update

pat_ID=None
pat_name=None
pat_dob=None
pat_effects=None
pat_sex=None
pat_BG=None
pat_contact=None
pat_doses=None
pat_pres=None
pat_CT=None

def up1():
    global u1, u2, u3, u4, u5, u6, u7, u8, u9, u10, ue1, conn
    conn.cursor()
    u1 = pat_ID.get()
    u2 = pat_name.get()
    u3 = pat_sex.get()
    u4 = pat_dob.get()
    u5 = pat_BG.get()
    u6 = pat_doses.get()
    u7 = pat_pres.get()
    u8 = pat_contact.get()
    u9 = pat_CT.get()
    u10 = pat_effects.get()
    conn = sqlite3.connect("vaccinedatabase.db")
    p = list(conn.execute("Select * from PATIENT where COWIN_ID=?", (u1,)))
    if len(p) != 0:
        conn.execute('UPDATE PATIENT SET NAME=?,SEX=?,DOB=?,BLOOD_GROUP=?,SIDE_EFFECTS=?,CONSULT_TEAM=?,CONTACT_NO=? where COWIN_ID=?', ( u2, u3, u4, u5, u10, u9, u8,u1,))
        conn.execute('UPDATE COVID  set NODOSES =?,PRESCRIPTION=? WHERE COWIN_ID=?', ( u6, u7,u1,))
        tkinter.messagebox.showinfo("VACCINATION DATABSE SYSTEM", "DETAILS UPDATED INTO DATABASE")
        conn.commit()

    else:
        tkinter.messagebox.showinfo("VACCINATION DATABSE SYSTEM", "PATIENT IS NOT REGISTERED")

labelu=None
bu1=None
```

```
    rootU.destroy()

##-----PATIENT UPDATE SCREEN -----##
def P_UPDATE():
    global pat_effects, pat_BG, pat_doses, pat_pres, pat_CT, pat_dob, pat_contact, pat_ID, pat_name, pat_sex
    global rootU, regform, id, name, dob, sex, contact, ct, effects, pres, doses, bg, SUBMIT, menubar, filemenu, p1f, p2f,HEAD
    rootU = tkinter.Tk()
    rootU.title("UPDATE WINDOW")
    menubar = tkinter.Menu(rootU)
    filemenu = tkinter.Menu(menubar, tearoff=0)
    filemenu.add_command(label="NEW", command=P_UPDATE)
    filemenu.add_separator()
    filemenu.add_command(label="EXIT", command=EXITT)
    rootU.config(menu=menubar)
    menubar.add_cascade(label="File", menu=filemenu)
    HEAD=tkinter.Label(rootU,text="ENTER NEW DETAILS TO UPDATE",bg='black',fg='white')
    id = tkinter.Label(rootU, text="COWIN ID")
    pat_ID = tkinter.Entry(rootU)
    name = tkinter.Label(rootU, text="PATIENT NAME")
    pat_name = tkinter.Entry(rootU)
    sex = tkinter.Label(rootU, text="SEX")
    pat_sex = tkinter.Entry(rootU)
    dob = tkinter.Label(rootU, text="DOB (YYYY-MM-DD)")
    pat_dob = tkinter.Entry(rootU)
    bg = tkinter.Label(rootU, text="BLOOD GROUP")
    pat_BG = tkinter.Entry(rootU)
    doses = tkinter.Label(rootU, text="DOSE NUMBER")
    pat_doses = tkinter.Entry(rootU)
    pres =  tkinter.Label(rootU, text="PRESCRIPTION (IF ANY)")
    pat_pres = tkinter.Entry(rootU)
    contact = tkinter.Label(rootU, text="CONTACT NUMBER")
    pat_contact = tkinter.Entry(rootU)
    ct = tkinter.Label(rootU, text="CONSULTING TEAM / DOCTOR")
    pat_CT = tkinter.Entry(rootU)
    effects = tkinter.Label(rootU, text="SIDE EFFECTS (IF ANY)")
    pat_effects = tkinter.Entry(rootU)
    SUBMIT=tkinter.Button(rootU,text="SUBMIT",command=up1)
    HEAD.pack()
    id.pack()
    pat_ID.pack()
    name.pack()
    pat_name.pack()
    sex.pack()
    pat_sex.pack()
    dob.pack()
    pat_dob.pack()
```

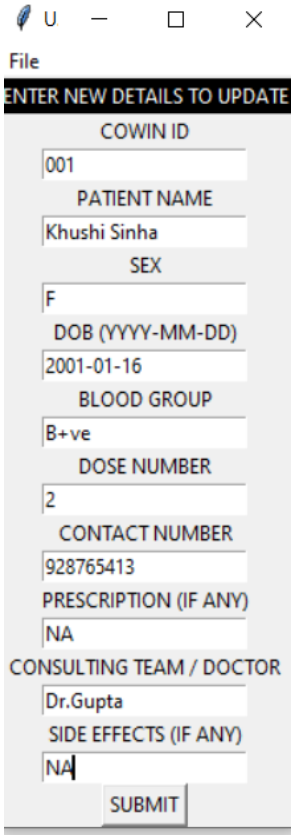
```
contact.pack()
pat_contact.pack()
pres.pack()
pat_pres.pack()
ct.pack()
pat_CT.pack()
effects.pack()
pat_effects.pack()
SUBMIT.pack()
rootU.mainloop()
```

RESULT

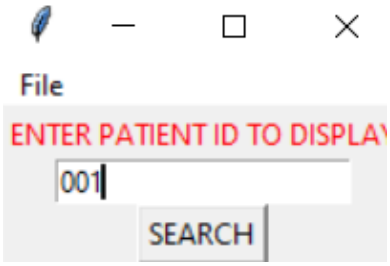
The Database is successfully able to input details, store it, update it and display as shown below with double security features (i.e. Password and OTP through mail)



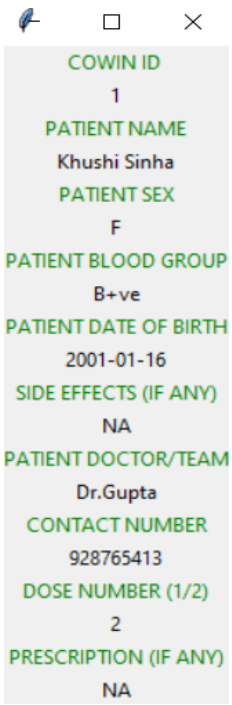
(a) Registration (1 dose)



(b) Update (2 doses)



(c) Enter ID to Display



(d)Updated data & change of doctor)

CONCLUSION

It is important to record the Covid-19 Vaccination doses given to the public in the country. It helps keep a track and understand the immunity pattern .With proper immunization records, it is easier to identify how many are yet to be vaccinated and follow the procedure to vaccinate them, thereby curbing the spread of the Novel Coronavirus. We aim to build on this project further so we can maybe host it and actually put to use. This can be used as a template for future pandemics or any vaccination management in the future as well.

REFERENCES

https://docs.python.org/3/

https://aka-trip.medium.com/otp-verification-using-python-598959e84f78

https://www.geeksforgeeks.org/python-gui-tkinter/

https://docs.python.org/3/tutorial/modules.html

https://docs.python.org/3/library/smtplib.html

https://stackoverflow.com/questions/tagged/python