TERM- I

A PROJECT SYNOPSIS

On

COVID Vaccination Survey

Submitted By

1. **Mr. Yash Bhake**
2. **Mr. Ajinkya A Deshpande**
3. **Mr. Udbhav Dwivedi**

2021-2022

Under the Guidance of

Mrs. Shahjahan Shaikh

Submitted in partial fulfilment of the requirement for qualifying

XII/CBSE/2021-2022/Board Examination

Apeejay School

Nerul, Navi Mumbai

CERTIFICATE

This is to certify that the Project Entitled **COVID Vaccination Survey** undertaken at the **Apeejay School, Nerul, Navi Mumbai** by Mr Yash Bhake in partial fulfilment of the requirement for qualifying TERM-I COMPUTER SCIENCE XII/CBSE/2021-2022 Examination. It is further certified that he/she has completed all required phases of the Project.

Signature of Internal Guide

Signature of Principal

School Seal

TABLE OF CONTENTS

Introduction

Our project is titled ‘COVID Vaccination Survey’. The Covid-19 pandemic has altered the world’s health sector drastically. Thus in the year 2021 vaccines were made available to the people of our country. But these vaccines are needed to be monitored so as to make them more efficient in the future. Our project is an attempt to design an efficient interactive menu oriented interface which monitors the symptoms/side effects after taking a particular vaccine. The information collected will be stored in a database, and various plots have been created using this information, which can help catch any trend or any flaw and may help Vaccine brands Aware of The effects of the vaccines on the people. It is designed in a way that the user is able to understand the system with ease. It asks the user to register his/her name for his/her chosen vaccine, after which it gives the user a unique registration number. After taking the shot it asks for symptoms (if any).It then takes all the inputs given, stores them in the database and plots the graphs accordingly.

The main aim of this project is to make the healthcare sector aware of the symptoms of the vaccines and help them make the vaccines more efficient. We assure that ourproject works in an coherent and hassle-free way. The whole process is also very time efficient.

MySQL tables used in the program:

i) userinfo
ii) vaccination

Literature Survey

This project is inspired from the Government of India app Aarogya Setu, which amidst this pandemic, has played a vital role. This app has made the Vaccination process much simpler and smooth.

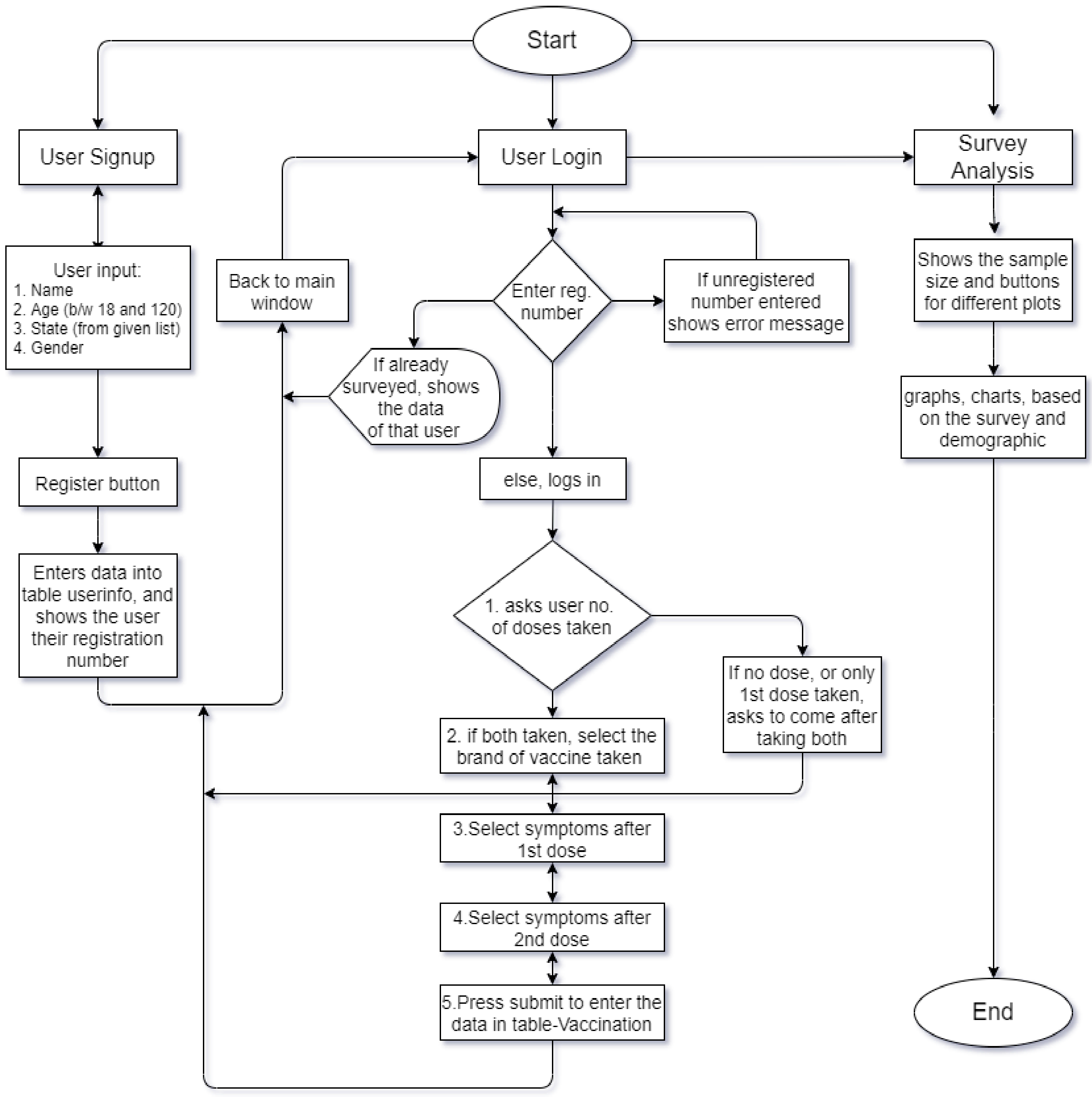
This app is an immense, in the whole covid scenario, We got very much inspired by its well functioning, and ability to analyze data collected from a huge population, and seamlessly providing the user with various graphs and plots.

Hardware and Software requirement

1. HARDWARE REQUIREMENTS:
   o x86 64-bit CPU (Intel / AMD architecture)
   o 4 GB RAM,5 GB free disk space

2. SOFTWARE REQUIREMENTS:
   o Operating system:
      Linux- Ubuntu 16.04 to 17.10
      Windows 7 to 10 with 2GB RAM (4GB preferable)
   o Python IDLE(3.8-32\64bit)
   o MySQL
   o Matplotlib and Pillow Python library.

Flow chart of the project


Start
User Signup
User Login
Survey Analysis
User input:
1. Name
2. Age (b/w 18 and 120)
3. State (from given list)
4. Gender
Back to main window
Enter reg. number
If unregistered number entered shows error message
Shows the sample size and buttons for different plots
If already surveyed, shows the data of that user
graphs, charts, based on the survey and demographic
Register button
else, logs in
Enters data into table userinfo, and shows the user their registration number
1. asks user no. of doses taken
If no dose, or only 1st dose taken, asks to come after taking both
2. if both taken, select the brand of vaccine taken
3.Select symptoms after 1st dose
4.Select symptoms after 2nd dose
5.Press submit to enter the data in table-Vaccination
End

Description of the Packages/Modules used

1) MySQL Connector:
MySQL Connector/Python enables Python programs to access MySQL databases using an API that is compliant with the Python.
It is written in pure Python and does not have any dependencies except for the Python Standard Library.

2) OS:
The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. We have used this for getting the image address from the folder where the .py file is stored

3) Pillow:
Python pillow library is used to image class within it to show the image. The image modules that belong to the pillow package have a few inbuilt functions such as load images or create new images, etc. We have used this module just for some igame resizing and applying some as wallpaper, for a userfriendly look, this is totally optional, so our code doesn't depend on the availability of this module on any other devices

4) Matplotlib:
Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, Qt etc. We have used this awesome library for creating graphs and plots of the data we collected using this survey, the graphs in this module are interactive too.

5) NumPy:
NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. It comes under matplotlib, and some of matpotlib's coding requires NumPy arrays.

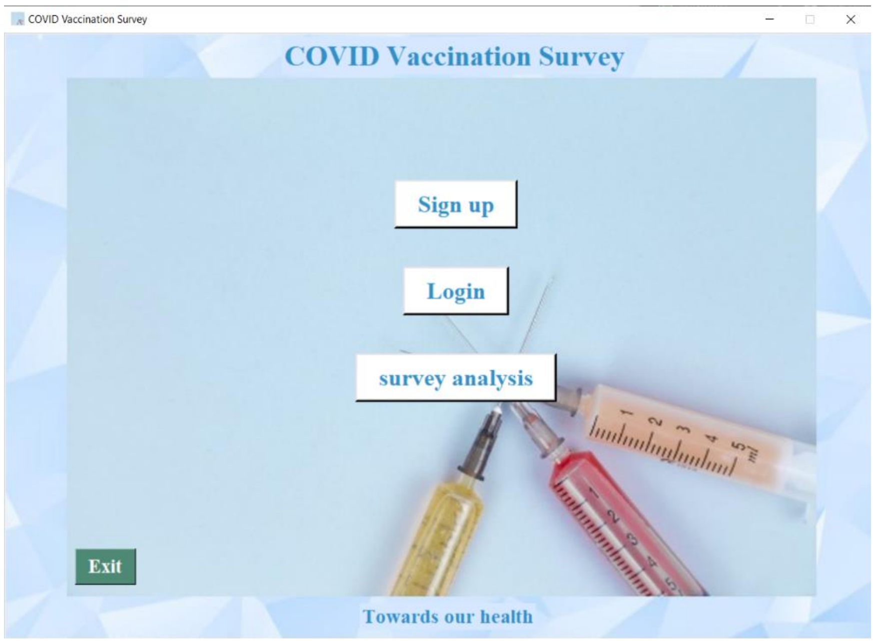
6) Tkinter:
Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Our user interface is built using this library.

Coding

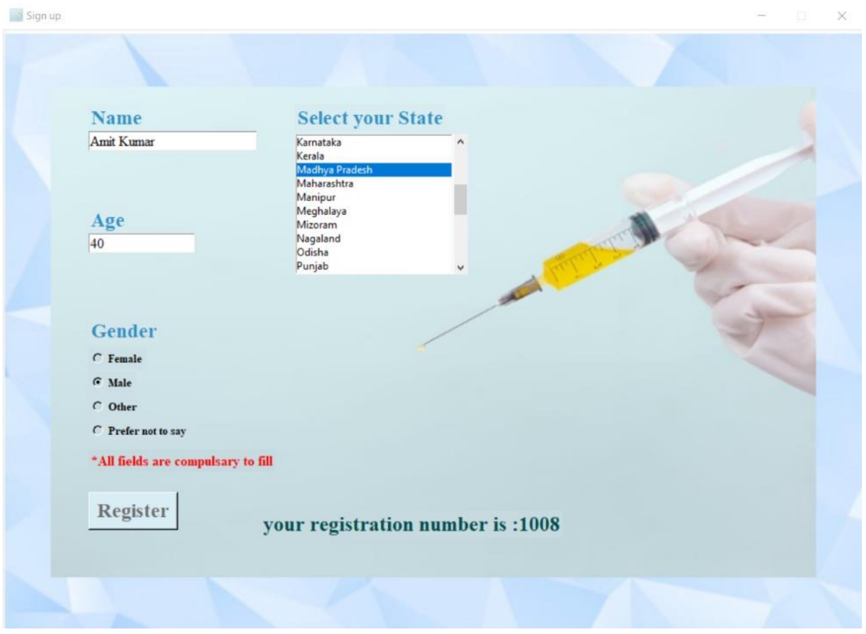
```
import numpy as np
import mysql.connector as connector
import os
from tkinter import *
from PIL import ImageTk, Image
from matplotlib import pyplot as plt
from tkinter import messagebox as mb
conn = connector.connect(user="root",
                    password="root",
                    host = "localhost",
                    database="computer")
cursor = conn.cursor(buffered = True)
#To prevent unread result found error
#buffering -> temporary storage, the whole db data is fetched and stored temporarily.
#table creation (userinfo and vaccination)
#userinfo table creation
"""cursor.execute('''create table IF NOT EXISTS userinfo(
            regno integer AUTO_INCREMENT PRIMARY KEY,
            name varchar(255),
            age integer NOT NULL DEFAULT 18,
            gender enum ('Female', 'Male', 'Other', 'Prefer not to say'),
                state enum ("Andhra Pradesh","Arunachal Pradesh
                ","Assam","Bihar","Chhattisgarh","Goa","Gujarat","Haryana",
                            "Himachal Pradesh","Jammu and
                Kashmir","Ladakh","Jharkhand","Karnataka","Kerala","Madhya Pradesh",

                "Maharashtra","Manipur","Meghalaya","Mizoram","Nagaland","Odisha","Punj
                ab","Rajasthan",
                            "Sikkim","Tamil Nadu","Telangana","Tripura","Uttar     Pradesh","Uttarakhand","West
                Bengal",
                            “Andaman and Nicobar Islands","Chandigarh","Dadra and Nagar Haveli","Daman and Diu",
                            "Lakshadweep","National Capital Territory of Delhi","Puducherry"))''')
cursor.execute('alter table userinfo AUTO_INCREMENT=1001')
#vaccination table creation
cursor.execute('''create table IF NOT EXISTS vaccination(
            entryno integer AUTO_INCREMENT PRIMARY KEY,
            regno integer REFERENCES userinfo (regno) ON DELETE CASCADE ON UPDATE CASCADE,
            vaccinebrand enum ('Covaxin', 'Covishield', 'Sputnik V', 'Moderna', 'Pfizer') NOT NULL,
           symptoms enum ("1. Feeling sick (nausea)", "1. Feeling tired (fatigue)","1. Fever","1. cough and/or cold",
"1. itching/rash/red bumps at injection site","1. pain in chest","1. breathlessness","1. persistent abdominal pain","1. seizures",
"1. severe and persistent headache","1. weakness/paralysis of limbs or any side of the body","1. persistent vomiting",
 "1. blurred vision or pain in eyes","1. change in mental status,      confusion","1. Decreased appetite","1. Other","1. none",
"2. Feeling sick (nausea)","2. Feeling tired (fatigue)","2. Fever","2. cough and/or cold",
"2. itching/rash/red bumps at injection site","2. pain in chest","2. breathlessness","2. persistent abdominal pain","2.
seizures",
“2. severe and persistent headache","2. weakness/paralysis of limbs or any side of the body","2. persistent vomiting",
"2. blurred vision or pain in eyes","2. change in mental status, confusion","2. Decreased appetite","2. Other", "2.
none"))''')
# 1. means symptoms after 1st dose, 2. means symptoms after 2nd dose
cursor.execute('alter table vaccination AUTO_INCREMENT=1')
cursor.execute('''alter table vaccination add foreign key(regno) references
userinfo(regno) on delete cascade on update cascade''')
#initiating table userinfo with sample data.
cursor.execute("insert ignore into userinfo (regno, name, age, gender, state) values(1001,'Yash
Bhake',18,'Male','Maharashtra')")
cursor.execute("insert ignore into userinfo (regno, name, age, gender, state) values(1002,'Ajinkya
Deshpande',18,'Male','Maharashtra')")
cursor.execute("insert ignore into userinfo (regno, name, age, gender, state) values(1003,'Rashmi
Banerjee',25,'Female','West Bengal')")
conn.commit()"""
# main window
root = Tk()
root.title("COVID Vaccination Survey")
root.resizable(0,0) #disables maximise btn
bgimg1 = Image.open(r'{}'.format(str(os.getcwd())+"/vaccimg1.png"))# ======defining image and its path, and
inserting it in a label (photolabel)
resized = bgimg1.resize((1000, 700))# resizing image to required size.
bgimg = ImageTk.PhotoImage(resized)#pillow command to get the image after applying changes
photolabel = Label(root, image=bgimg).place(x=0, y=0)
root.iconphoto(False, bgimg)#changing the tk symbol on top left of the window
root.geometry("1000x700")
heading = Label(root, text="COVID Vaccination Survey", font=("times", 25, "bold",), fg = "#3090C7",bg ="#d6e8ff" )
subheading = Label(root, text="Towards our health", font=("times", 17, "bold"),fg = "#3090C7", bg="#d6e8ff")
heading.place(relx=0.32, y=5)
subheading.place(relx=0.41, rely=0.94)
def exitroot():
    qn = mb.askquestion("Exit", "Are you sure you want to exit?")
    if qn=='yes':
        root.destroy()
root.protocol("WM_DELETE_WINDOW", exitroot)# will execute exitroot fn on pressing X button
nameval=0
ageval=0
stateval=0
genderval=0
regnoval=0
sym1lst=0
sym2lst=0
#USER SIGN UP
def usersignup():
    global nameval, ageval, stateval, genderval
    root.iconify()#minimises root window
    signupwin = Toplevel(root)# creating new subwindow
    signupwin.title("Sign up")
    signupwin.geometry("1000x700")
    signupwin.resizable(0,0)
    bgimg2 = Image.open(r'{}'.format(str(os.getcwd())+"/vaccimg2.png"))
    resized = bgimg2.resize((1000, 700))
    bgimg = ImageTk.PhotoImage(resized)
    photolabel = Label(signupwin, image=bgimg).place(x=0, y=0)
    signupwin.iconphoto(False, bgimg)
    #dropdown menu for state
    #stateval = StringVar() erase this after some time
    statelist = ["Andhra Pradesh","Arunachal Pradesh
","Assam","Bihar","Chhattisgarh","Goa","Gujarat","Haryana",
            "Himachal Pradesh","Jammu and Kashmir","Ladakh","Jharkhand","Karnataka","Kerala","Madhya
Pradesh",
            "Maharashtra","Manipur","Meghalaya","Mizoram","Nagaland","Odisha","Punjab","Rajasthan",
            "Sikkim","Tamil Nadu","Telangana","Tripura","Uttar Pradesh","Uttarakhand","West Bengal",
            "Andaman and Nicobar Islands","Chandigarh","Dadra and Nagar Haveli","Daman and Diu",
            "Lakshadweep","National Capital Territory of Delhi","Puducherry"]
    Label(signupwin, text="Select your State",font=("times", 17, "bold"),fg = "#3090C7", bg="#d9edf1").place(relx=0.34,
rely=0.125)
    #scrollbar for statelistbox
    myframe = Frame(signupwin,height = 12, width = 15)
    scrollbar = Scrollbar(myframe, orient = VERTICAL)
    statelistbox = Listbox(myframe, yscrollcommand=scrollbar.set, width = 30, exportselection = False)#exportselection
ensures that even when focus is not on the
    statelistbox.pack(side = LEFT)                                   #listbox, the selection(s) remain intact
    scrollbar.configure(command = statelistbox.yview)
    scrollbar.pack(side=RIGHT, fill=Y)
    myframe.place(relx=0.34, rely=0.175)
    for i in statelist:
        statelistbox.insert(END,i)
 #name
    name = StringVar# StringVar is a datatype defined for&by tkinter, it is similar to str
    nameentry = Entry(signupwin, textvariable = name, font=("times", 12), bg = "white", width = 24)
    nameentry.delete(0, END)
    nameentry.insert(0, "Enter your name")
    global a
    a = 1
    def entryerase1(event):                #this loop if for not deleting the default text on clicking the entry box the 2nd time
        global a
        if a<2:
            nameentry.delete(0, END)
            a+=1
        else:
            pass
    nameentry.bind('<Button-1>', entryerase1)
    nameentry.place(relx = 0.10, rely = 0.17)
        Label(signupwin, text="Name",font=("times", 17, "bold"),fg = "#3090C7",    bg="#daf0f2").place(relx = 0.10, rely
      = 0.125)
      #age
    age = IntVar
        ageentry = Entry(signupwin, textvariable = age, font=("times", 12), bg = "white", width = 15)
    ageentry.delete(0, END)
    ageentry.insert(0, "Enter your age")
    global b
    b = 1
    def entryerase2(event):
        global b
        if b<2:
            ageentry.delete(0, END)
            b+=1
        else:
            pass
    ageentry.bind('<Button-1>', entryerase2)
    ageentry.place(relx = 0.10, rely = 0.34)
        Label(signupwin, text="Age",font=("times", 17, "bold"),fg = "#3090C7", bg="#d9edf1").place(relx = 0.10, rely =
      0.295)
      #gender
    gender = StringVar(signupwin, 4)
    gendervalues = {1:"Female", 2:"Male", 3:"Other", 4:"Prefer not to say"}
    for (num, value) in gendervalues.items():
            Radiobutton(signupwin, text = value, variable = gender, value = num, font=("times", 10, "bold"),
                        selectcolor = "#c7dae0", bg = "#cde0e6").place(relx=0.1, rely=(num/25+0.49))
    Label(signupwin, text="Gender",font=("times", 17, "bold"),fg = "#3090C7", bg="#d3e6ea").place(relx = 0.1, rely =
      0.48)
Label(signupwin, text="*All fields are compulsary to fill",font=("times", 12, "bold"),fg = "#ff0000",
      bg="#c7dae0").place(relx = 0.1, rely = 0.70)
    cursor.execute("select regno from userinfo")
    rows = cursor.fetchall()
    x = (rows)[cursor.rowcount-1][0]+1
    #cursor.execute("select * from userinfo"
#USER LOGIN
def loginpwd():
    global regnoval, bgimg
    regnowin = Toplevel(root)
    regnowin.title("Registration no.")
    regnowin.geometry("320x82+400+360")
    regnowin.configure(bg = "#bedaec")
    regnowin.resizable(0,0)
    regnowin.iconphoto(False, bgimg)
Label(regnowin, text = """Please enter your registration number""",font=("times", 13), fg = "#008080", bg =
"#bedaec").pack()
    pressenterlbl = Label(regnowin, text = """Press enter""",font=("times", 13),fg = "#008080", bg = "#bedaec")
regnopwd = StringVar
    regentry = Entry(regnowin, textvariable = regnopwd, font=("times", 12), bg = "white", width = 24, show = "*")
    regentry.pack()
    cursor.execute("select regno from userinfo")
    rows1=cursor.fetchall()
    regnolist1 = []
    for i in range(0,cursor.rowcount):
        regnolist1.append(str(rows1[i][0]))
    cursor.execute("select distinct regno from vaccination")
    rows2=cursor.fetchall()
    regnolist2=[]
    for i in range(0,cursor.rowcount):
        regnolist2.append(str(rows2[i][0]))
        def enter(event):# event is necessary fotr the bind method to work
        global regnoval
        regnoval = regentry.get()
        def loginpwddestroy():
            regnowin.destroy()
        def invalidregno():
            mb.showwarning("Error", """Please enter your correct registration number
      no registration done with this number""")
        if regnoval in regnolist1 and regnoval not in regnolist2:# means the user has registered but not been surveyed
            regentry.delete(0, END)
            regentry.insert(0, "")
            loginpwddestroy()
            userlogin()
        elif regnoval in regnolist1 and regnoval in regnolist2:# means the user has registered as well as been surveyed
            qn = mb.askquestion("Thank you","""Your response has been recorded, Thank you.
        Do you want to see your response?""")
            if qn=='yes':
                regnowin.iconify()
                cursor.execute("select * from userinfo where regno = '"+regnoval+"'")
                vals = cursor.fetchone()
                nameval, ageval, genderval, stateval = vals[1], vals[2], vals[3], vals[4]
                cursor.execute("select vaccinebrand from vaccination where regno = '"+regnoval+"'")
                vaccval = cursor.fetchall()[0][0]
                cursor.execute("select symptoms from vaccination where regno = '"+regnoval+"'")
                vals2 = cursor.fetchall()
                sym1val = []
                sym2val = []
                for i in vals2:
                    if i[0][0]=="1":
                        sym1val.append(i[0][3:])
                    else:
                        sym2val.append(i[0][3:])
                if sym1val == ["1. none"]:
                    sym1val = "none"
                if sym2val == ["2. none"]:
                    sym2val = "none"
                infowin = Toplevel(regnowin)
                infowin.title("Info")
                infowin.geometry("1000x313+400+360")
                infowin.configure(bg = "white")
                infowin.resizable(0,0)
                infowin.iconphoto(False, bgimg)
                Label(infowin, height = 20, width = 141, borderwidth = 5, relief = RIDGE, bg = "white").place(x = 0, y = 0)
                #SURVEY ANALYSIS
def surveyanalysis():
    global symlist, sym1list, regnos
    root.iconify()
    symlist = ["nausea", "fatigue","Fever","Cough/ cold","Itching/rash","Pain in chest",
               "Breathlessness","abdominal pain","Seizures","headache",
               "Weakness","vomiting","Blurred vision",
               "confusion","Decreased appetite","Other", "None"]
    sym1list = ["1. Feeling sick (nausea)", "1. Feeling tired (fatigue)","1. Fever","1. cough and/or cold",
                "1. itching/rash/red bumps at injection site","1. pain in chest","1. breathlessness","1. persistent abdominal
pain","1. seizures",
                "1. severe and persistent headache","1. weakness/paralysis of limbs or any side of the body","1. persistent
vomiting",
                "1. blurred vision or pain in eyes","1. change in mental status, confusion","1. Decreased appetite","1. Other",
"1. none"]
    svanwin = Toplevel(root)
    svanwin.title("Stats")
    svanwin.geometry("1000x700")
    svanwin.resizable (0,0)
    bgimg4 = Image.open(r'{}'.format(str(os.getcwd())+"/vaccimg4.png"))
    resized = bgimg4.resize((1000, 700))
    bgimg = ImageTk.PhotoImage(resized)
    photolabel = Label(svanwin, image=bgimg).place(x=0, y=0)
    svanwin.iconphoto(False, bgimg)
    cursor.execute("select distinct regno from vaccination")
    regnos = len(cursor.fetchall())
    Label(svanwin, text = "Survey Statistics", font = ("times",22), bg = "#d3e6ff", fg = "#008080").place(relx = 0.4, rely =
0.02)
    Label(svanwin, text = "Sample size: {}".format(regnos), font = ("times",18), bg = "#60cae0", fg =
"#003232").place(relx = 0.1, rely = 0.1)
   def sym_vs_freq():
        global sym1freq, sym2freq, regnos
 #y values for 1st bar (symptoms after dose 1)
        sym1freq = []
        sym2freq = []
        for i in sym1list:
                cursor.execute("select regno from vaccination where symptoms = '1. "+str(i)[3:]+"'")#this query will make
            a table having one particular symptom only
            cursor.fetchall()# this will have the table
            sym1freq.append(cursor.rowcount)
            # this will count the number of rows in fetchall() i.e. the frequency of that symptom
             #y values for 1st bar (symptoms after dose 2)
        for i in sym1list:
            cursor.execute("select regno from vaccination where symptoms = '2. "+str(i[3:])+"'")
            cursor.fetchall()
            sym2freq.append(cursor.rowcount)
        symfreq = sym1freq+sym2freq
        xindex = np.arange(len(symlist))
        figure, ax = plt.subplots(figsize = (12,7))
            sym1bar = ax.bar(xindex-0.21, sym1freq, color = "#008080", label = "Symptoms after 1st dose", width = 0.4)
            sym2bar = ax.bar(xindex+0.21, sym2freq, color = "#AEEEEE", label = "Symptoms after 2nd dose", width = 0.4)
        ax.set_title("Frequency of various symptoms after taking vaccination")
        ax.set_xlabel("Symptoms")
        ax.set_ylabel("Frequency")
        ax.bar_label(sym1bar)
        ax.bar_label(sym2bar)
        plt.xticks(ticks = xindex, labels = symlist, rotation = 70)
        plt.legend()#displays the iindex to the graph
        figure.tight_layout()# fits the graph in the window
        plt.ylim([0, regnos-30])
        plt.show()
      def states_vs_no():
              statelist = ["Andhra Pradesh","Arunachal Pradesh
           ","Assam","Bihar","Chhattisgarh","Goa","Gujarat","Haryana",
            "Himachal Pradesh","Jammu and Kashmir","Ladakh","Jharkhand","Karnataka","Kerala","Madhya
           Pradesh",
           "Maharashtra","Manipur","Meghalaya","Mizoram","Nagaland","Odisha","Punjab","Rajasthan",
           "Sikkim","Tamil Nadu","Telangana","Tripura","Uttar Pradesh","Uttarakhand","West Bengal",
           "Andaman and Nicobar Islands","Chandigarh","Dadra and Nagar Haveli","Daman      and Diu",
       "Lakshadweep","National Capital Territory of Delhi","Puducherry"]
       # x values for bar graph
        states = ['AP','AR','AS','BH','CT','GA','GJ','HR',"HP",
               "JK","LA","JH","KA","KL","MP",'MH','MN',
               "ML","MZ","NL","OD","PB","RJ", "SK",'TN','TG','TR','UP',
               'UT','WB','AN','CH',"DH",'DD','LD','DL','PY']
        # y values (frequency of vaccine doses)
        statefreq=[]
        for i in statelist:
            cursor.execute("select regno from userinfo where state='" + str(i) + "'")
            cursor.fetchall()
            statefreq.append(cursor.rowcount)
        xindex = np.arange(len(statelist))
        plt.figure(figsize=(15, 7))
        for i in range(0,37):
            plt.text(i, statefreq[i], str(round(statefreq[i]*100/sum(statefreq),1))+"%", ha = 'center', size = 7)
        plt.bar(xindex,statefreq,color="#008078",label="Vaccine taken",width=0.5)
        plt.title("Patients vaccinated per state")
        plt.xlabel("State Name")
        plt.ylabel("No of Pateints Vaccinated")
        plt.xticks(ticks=xindex, labels=states)
        plt.ylim([0,8])
        plt.legend()
        plt.tight_layout()
        plt.show()
    def sym_vs_nosym():
        cursor.execute("""select regno from vaccination where symptoms = '1. none' and regno          not in
                    (select regno from vaccination where symptoms = '2. none')""")
        cursor.fetchall()
        nosym1ct = cursor.rowcount
         cursor.execute("""select regno from vaccination where symptoms = '2. none' and regno not in
                    (select regno from vaccination where symptoms = '1. none' )""")
        cursor.fetchall()
        nosym2ct = cursor.rowcount
        cursor.execute("""select regno from vaccination where symptoms = '1. none' and regno in
                    (select regno from vaccination where symptoms = '2. none')""")
        cursor.fetchall()
        nosymct = cursor.rowcount
        cursor.execute("""select distinct regno from vaccination where regno not in
                    (select regno from vaccination where symptoms like '___none')""")
        cursor.fetchall()
        symct = cursor.rowcount
       #print(nosym1ct, nosym2ct, nosymct, symct)
        lbls = ["Symptoms only after 2nd dose", "Symptoms only after 1st dose", "No symptoms after both the doses",
             "Symptoms after both the doses"]
        plt.figure(figsize = (14,7))
        plt.pie([nosym1ct, nosym2ct, nosymct, symct], labels = lbls, explode=(0.01, 0.01, 0.01, 0.01),
              autopct='%1.2f%%', colors = ["#236B8E", "#0276FD", "#82CFFD", "#008080"])
        plt.title("Symptoms vs no symptoms")
        #autopct shows percentages of the information displayed on the pie chart, explode
        plt.legend(loc = "lower right")
        plt.tight_layout()
        plt.show()
```

Output Windows



MAIN WINDOW



Name: Amit Kumar

Age: 40

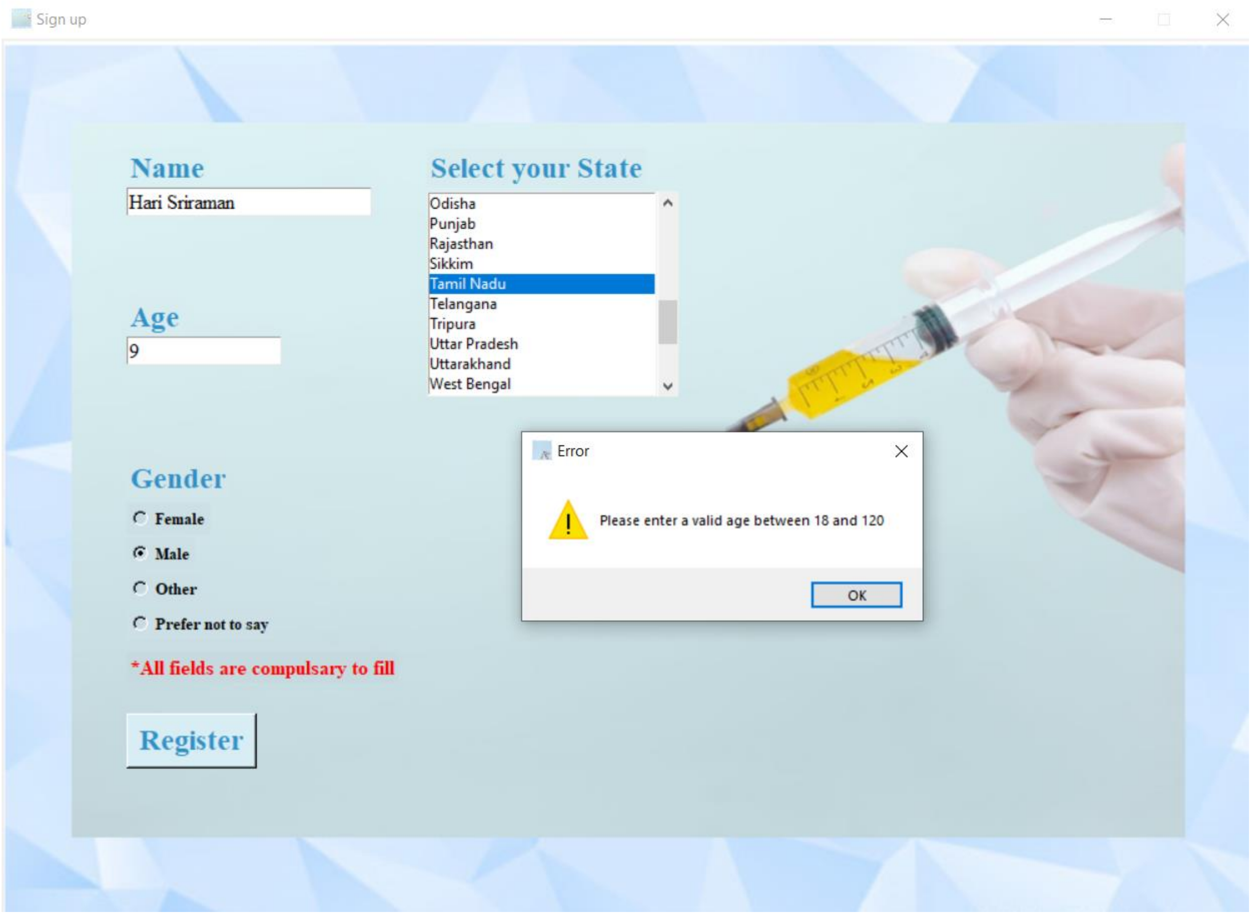
Select your State: Karnataka, Kerala, Madhya Pradesh (selected), Maharashtra, Manipur, Meghalaya, Mizoram, Nagaland, Odisha, Punjab

Gender: Female [✓ Male] Other Prefer not to say

*All fields are compulsary to fill

your registration number is :1008

REGISTRATION WINDOW



Sign up

Name

Hari Sriraman

Age

9

Gender

◯ Female

● Male

◯ Other

◯ Prefer not to say

*All fields are compulsary to fill

Register

Select your State

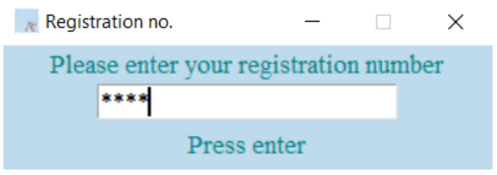
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal

Error

Please enter a valid age between 18 and 120

OK

ONE OF THE VALIDATIONS



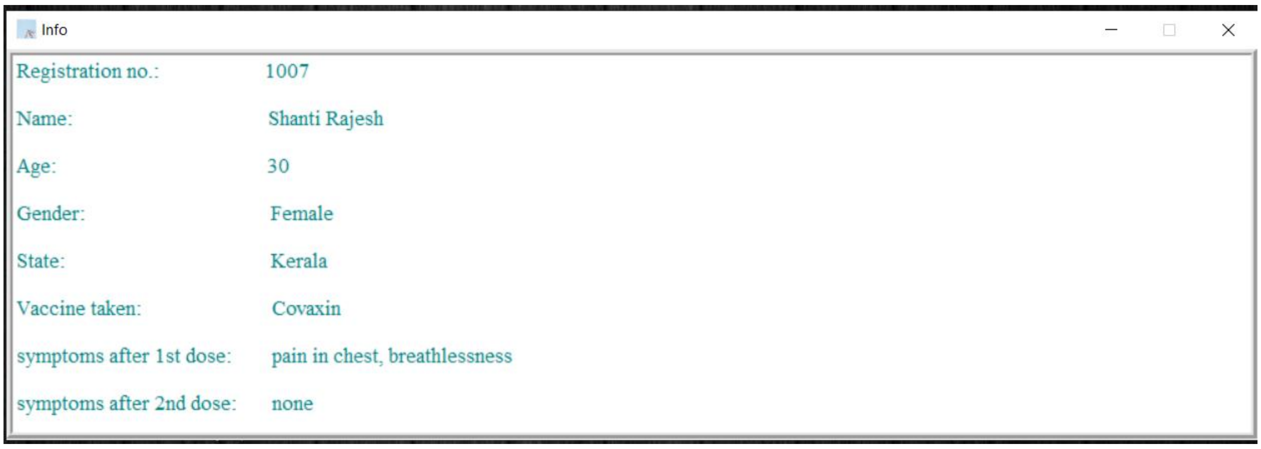
Registration no.

Please enter your registration number

****

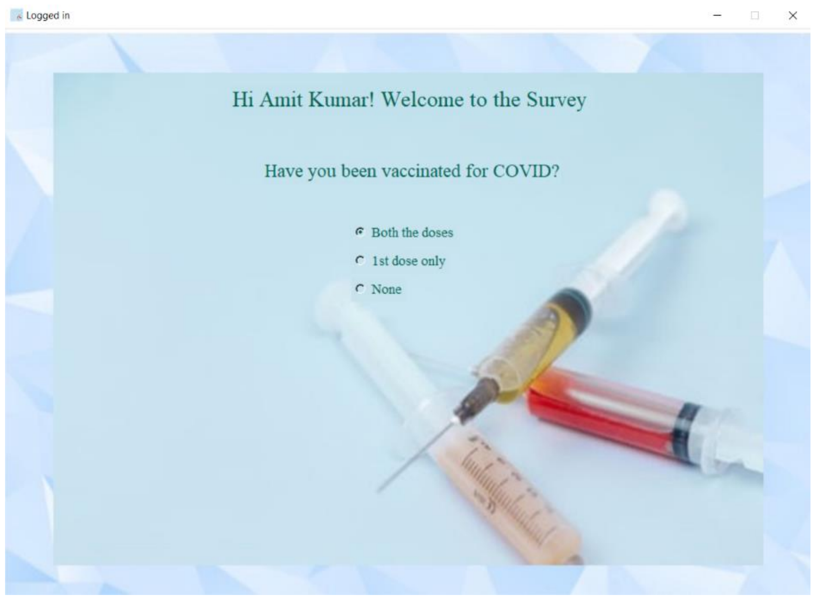
Press enter

LOGIN WINDOW



Registration no.:	1007
Name:	Shanti Rajesh
Age:	30
Gender:	Female
State:	Kerala
Vaccine taken:	Covaxin
symptoms after 1st dose:	pain in chest, breathlessness
symptoms after 2nd dose:	none

IF THE PERSON HAS BEEN SURVEYED

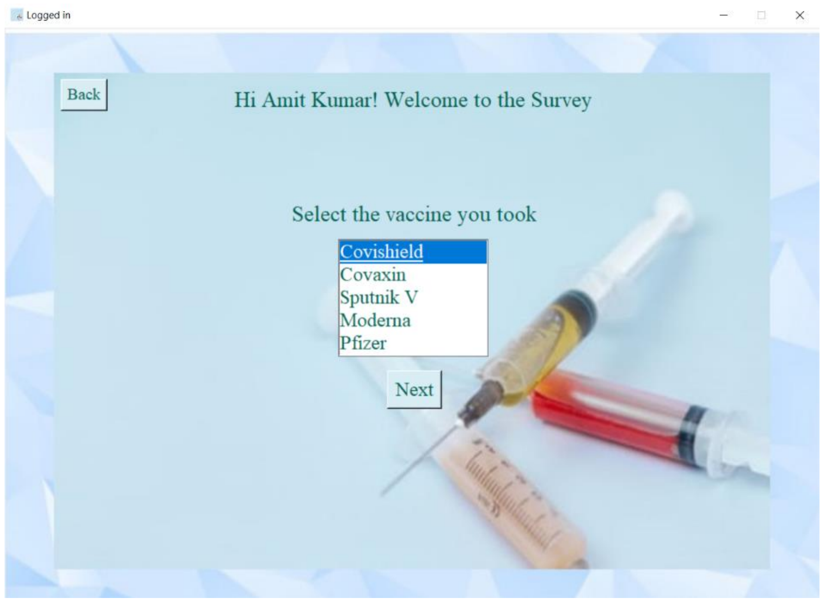


Hi Amit Kumar! Welcome to the Survey

Have you been vaccinated for COVID?

- [✓ Both the doses]
- 1st dose only
- None

ELSE, LOGIN PAGE 1

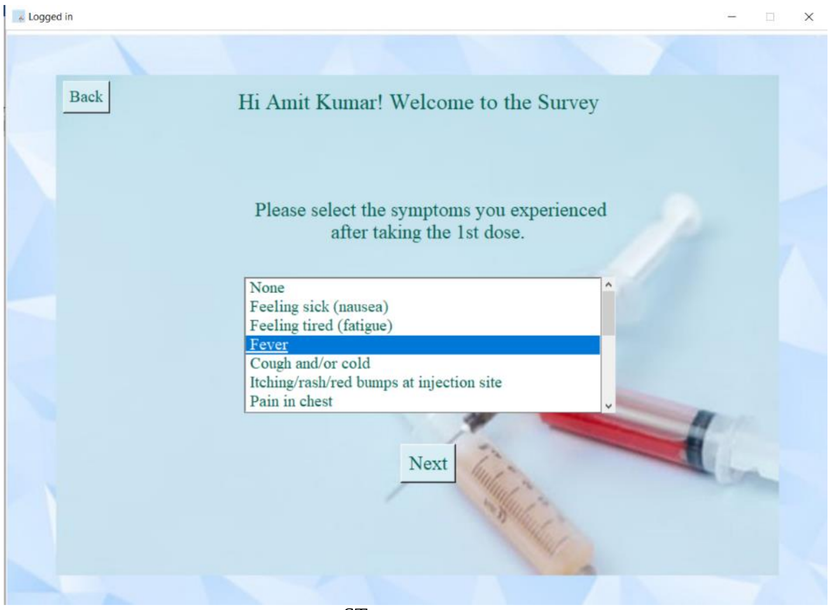


Hi Amit Kumar! Welcome to the Survey

Select the vaccine you took

- [✓ Covishield]
- Covaxin
- Sputnik V
- Moderna
- Pfizer

LOGIN PAGE 2

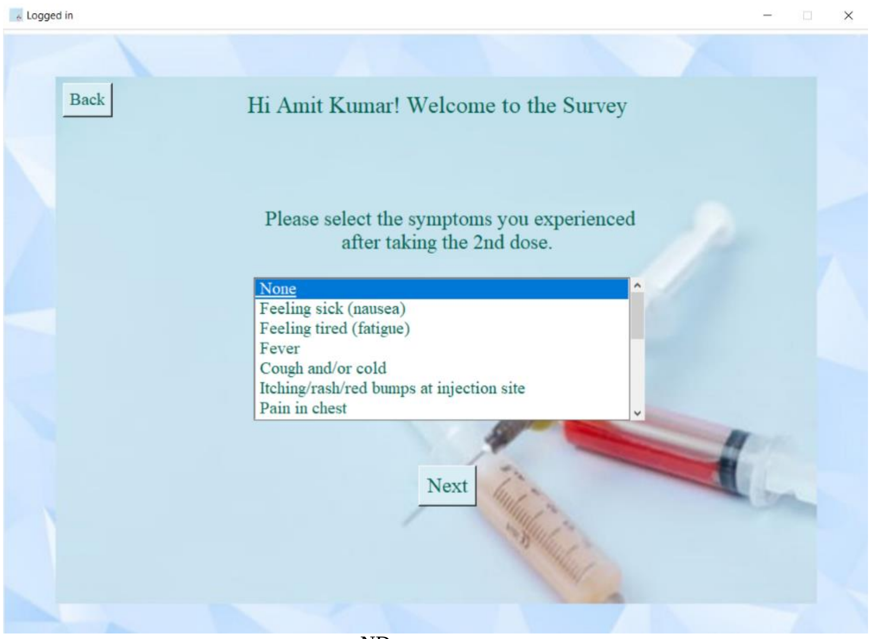


Hi Amit Kumar! Welcome to the Survey

Please select the symptoms you experienced after taking the 1st dose.

None
Feeling sick (nausea)
Feeling tired (fatigue)
Fever
Cough and/or cold
Itching/rash/red bumps at injection site
Pain in chest

SYMPTOMS AFTER 1ST DOSE (multiple can be selected)

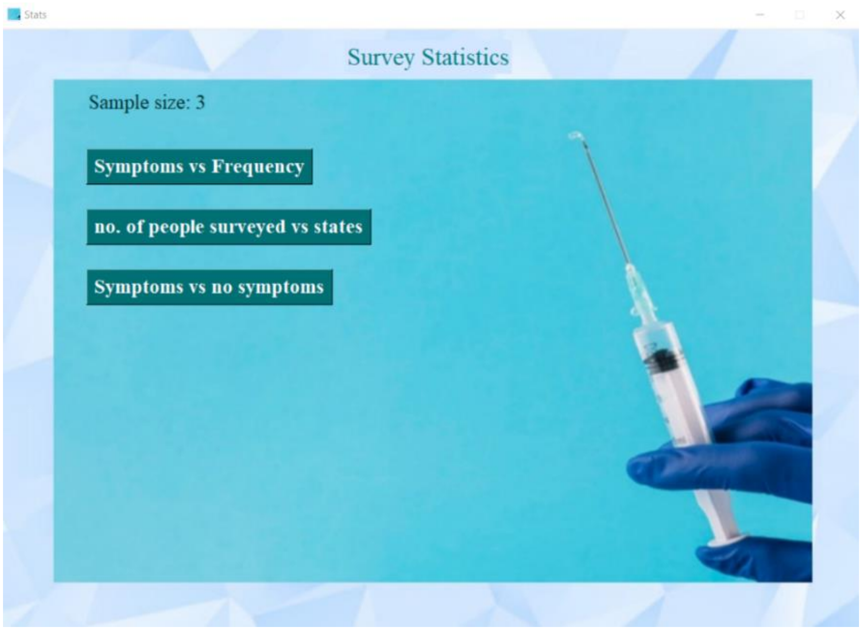


Hi Amit Kumar! Welcome to the Survey

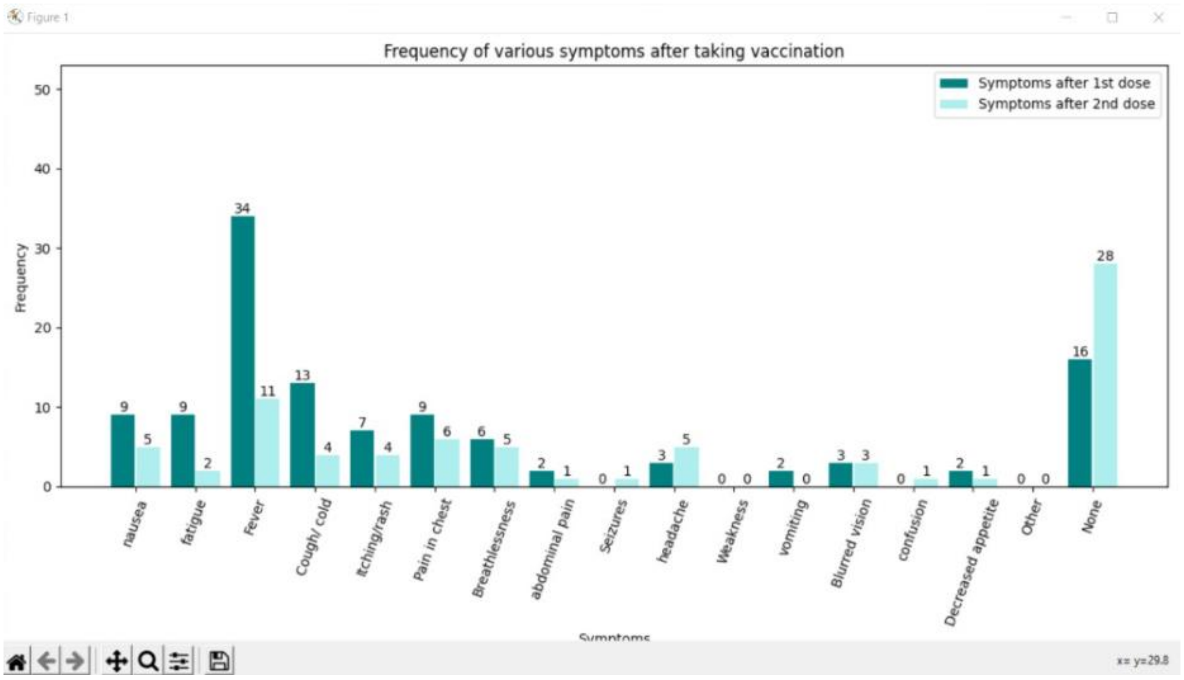
Please select the symptoms you experienced after taking the 2nd dose.

None
Feeling sick (nausea)
Feeling tired (fatigue)
Fever
Cough and/or cold
Itching/rash/red bumps at injection site
Pain in chest

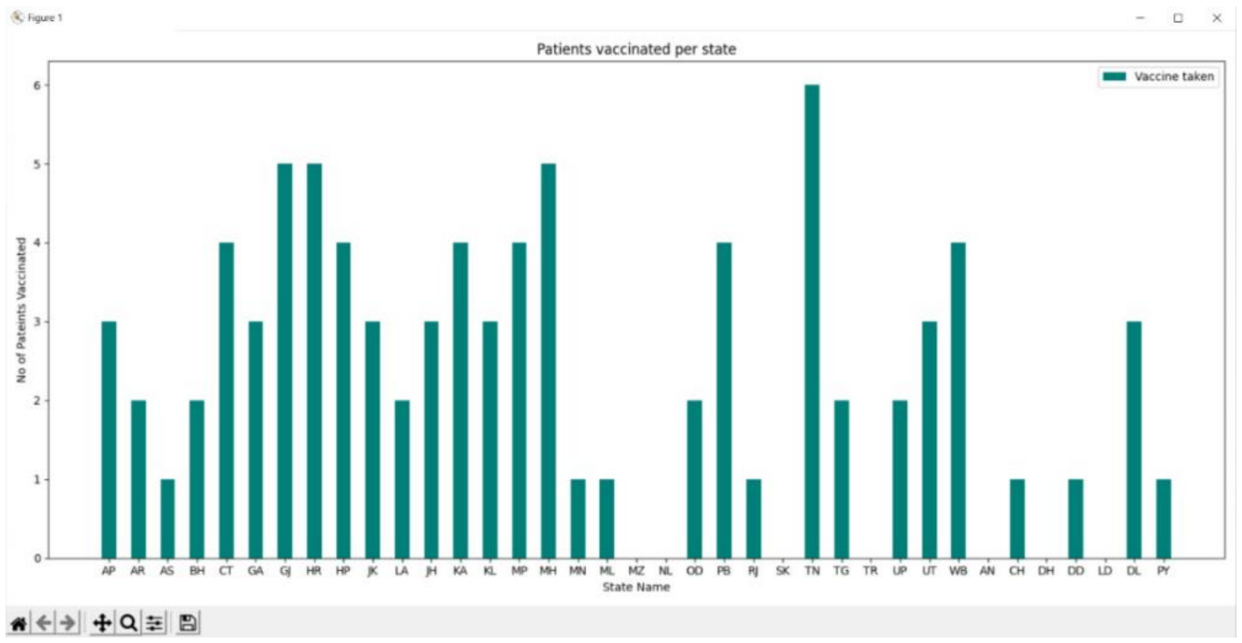
SYMPTOMS AFTER 2ND DOSE (multiple can be selected)



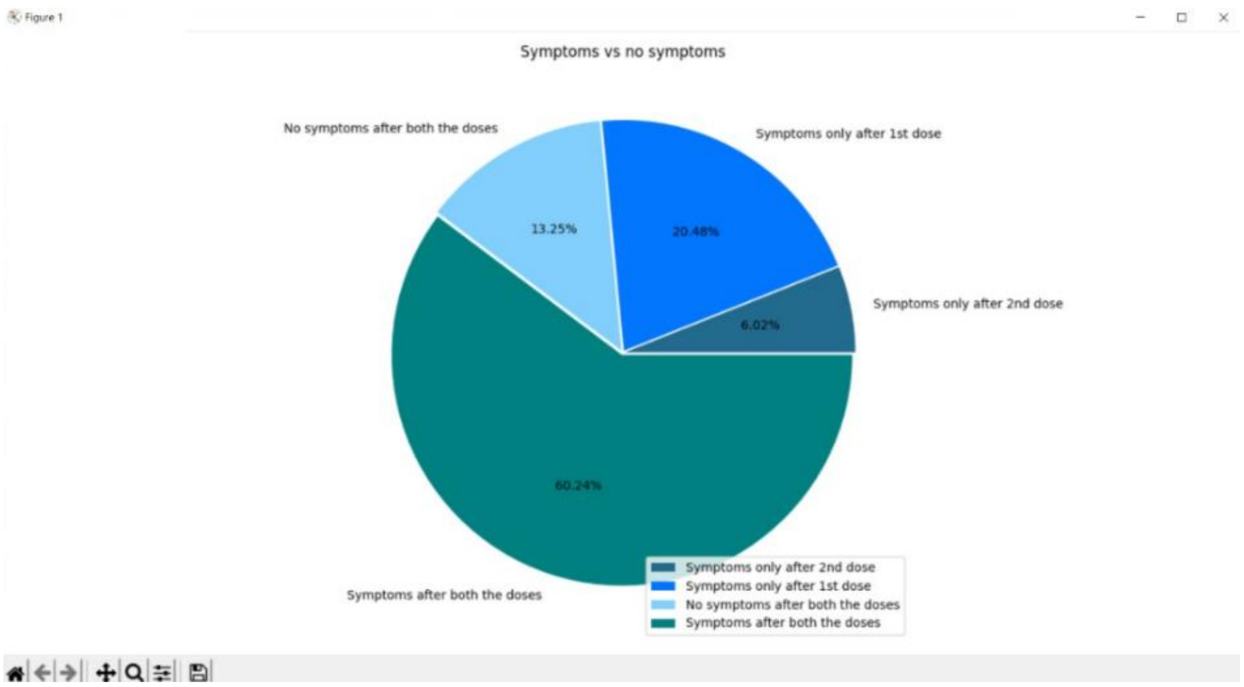
SURVEY ANALYSIS WINDOW



GRAPH: FREQ VS SYMPTOMS



GRAPH VACCINATION VS STATE



PIE CHART FOR SYMPTOMS

Conclusion

This project is a very user-friendly and shows the status of the vaccine available to the common man. Survey helps the person to choose the appropriate vaccine for him/her and for their loved ones. It also helps the Pharmaceutical Companies for analysing and manipulating their vaccines. The project is inspired by the current situation of the world. Thus our project “COVID Vaccination Survey” an efficient interactive menu oriented interface which monitors the symptoms/side effects after taking a particular vaccine. The information collected is stored in a database, and various plots have been created using this information, which can help catch any trend or any flaw. This project is can be seen as a template for surveys and its analysis, our project provides a jist of it, which may be further developed into a more practical and purposeful application that can help the nation.

Future Scope

The project can be a template for better larger practical projects, which may have a great impact in this crisis. Though it has a few limitations such as it can run on specific devices (i.e. it cant run on the mobile phones which are generally used by all the people.) . This program is not compatible with the Internet, it runs only on local Computers with compatible Operating System having MySQL, Matplotlib and Pillow Python libraries installed. These limitations can be rectified and the program can be developed further for mass use which may help us understand and analyse the biological threats that may arise in the future and their cures.

Bibliography

- Computer Science with Python for class XII by Sumita Arora
- www.geeksforgeeks.com
- www.tutorialspoint.com
- www.stackoverflow.com
- www.youyube
- www.python-course.eu
- www.javapoint.com