Computer Science Project

Vaccine Proximity Database Management

Name: Ram Prasath P

Class: XII – A10 (29)

Roll Number: 20632833

BONAFIDE CERTIFICATE

Certified that this project is a bonafide work of Master/Miss_____

Roll No._____of Class XII of **Maharishi Vidya Mandir Sr. Sec. School**, Chetpet, Chennai during the year 2021 – 2022.

Date: 05/02/2022 Teacher – in – charge

Submitted for SSCE Practical Examination held in the **COMPUTER SCIENCE** Laboratory at **Maharishi Vidya Mandir Sr. Sec. School**, Chetpet, Chennai.

DATE: INTERNAL EXAMINER

PRINCIPAL EXTERNAL EXAMINER

SEAL

ACKNOWLEDGEMENT

I express my sincere gratitude to our **PRINCIPAL, SHRI. G. HARIBABU** and **VICE PRINCIPAL**, **Smt V.SUNDARI** of our institution for their continuous support and encouragement.

I express my sincere thanks to my Computer Science Teachers, **Smt. T. SUJATHA** and **Smt. S. KRISHNAPRIYA** for helping me to complete this project successfully.

I would also like to thank our Laboratory Assistant **Smt. A. VIJAYA** for all the help extended to us.

Index

Problem Definition

Introduction

'Vaccination Planner' is an app designed through the integration of Python, Python's in-built module Tkinter and MySQL. This project was undertaken to increase the number of people vaccinated by making it easier for people to access the vaccine. We have utilized Python's in-built Tkinter module for its user-friendly GUI with the record maintenance provisions available on MySQL.

General Explanation

- The app is divided into 2 major subdivisions – one for vaccine camps who administer the vaccine, the other for people who want to get their vaccine (the vaccinee).
- Before entering amidst the sub-categories, it is essential to note that this app in no way is entitled to any organization but is only a medium for the various vaccine camps (Hospitals, NGO's etc.) to communicate with common people like us to ensure vaccination is provided at their disposal at the right time.
- When users open the app for the first time, they will be shown two options – Vaccinee and Vaccine Camp. Users are to choose according to their need.

- Choosing any of these options takes the user to a sign in and sign up page, where new users can create their account through the sign up page and existing users could log in to their account through the sign in page.
- While signing up users are shown the rules which they should follow to fill up their details.
- After users sign in to their accounts, they are shown their account details and the option to modify them if needed.
- Here, Vaccine Camp users can enter the vaccine stock available at their camp for the next day. They will not be allowed to enter or modify current day's vaccine stock as all eligible vaccinee will already be notified about the vaccine availability.
- Here, Vaccinee users can check whether they are eligible to get the vaccine for that day, if they are eligible, they will be shown the details of the vaccine camp from where they could get their vaccine.
- The eligibility of a vaccinee is calculated by prioritizing who needs the vaccine most by considering factors such as the job they do, their age, past medical conditions, the vaccine they took for their first dose (if any) and the date they got their first dose.
- All the details entered by the users are stored in a MySQL database in a usable form. All other functions are done by python with the help of inputs from the MySQL server.

Motive of the Project

As we are all aware of the ongoing global epidemic COVID 19, vaccinations are the only way to keep oneself safe in this atmosphere of delusional virus. As informed citizens of the Indian country with lukewarm technological awareness we decided to make a software application which not only benefits and supervises the vaccination process but also guides both retailers and common people to establish a secure network of quick hassle-free vaccinations.

Problem Analysis

Languages

Programming languages used in the making of this project are as follows,

- Python
  - Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
  - We made use of python to make use of its Tkinter GUI Module, take in input in a user-friendly way, run those input to get desired results and to display those result in a readable way to the user.
- MySQL
  - MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data.
  - We made use of MySQL for storing all the vaccinee data, vaccine camp data, the priority list and also to manipulate these data hassle-free.

Module

The modules used in the program are,

- mysql.connector
  - As in this project we used MySQL to keep track of all the records and databases, we needed seamless connection between Python and MySQL. That was achieved with the help of mysl.connector .
  - This allowed us to enter new user data, vaccine data, filter records and update records.
- PIL
  - Python Imaging Library (PIL) is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.
  - This was used to give a logo for our app.
- tkinter
  - Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI.
  - The clean and user-friendly user interface was made possible by the tkinter module.
- tkcalendar
  - This was used for getting dates from the users in an easy and intuitive way.

Functions

The functions defined and used in this app are listed below,

- connecting():

  This function is used for establishing a connection between Python and MySQL server.

- rootwin():

  This function is used to create the parent window for our app to work on and store some data which is required by the program.

- rootwin_destroy():

  This function is used to remove the widgets from the root window.

- dateformat (dfrmt):

  This function is used to convert dates (‘dfrmt’) from yyyy-mm-dd to dd-mm-yyyy and return the value.

- sign_frame(user='vaccinee', source='in', update='sign'):

  This function is used to create all the required widgets for the sign in, sign up and update page for both vaccinee and vaccine camp users. It takes value ‘user’ to differentiate between vaccinee and vaccine camp users, ‘source' to differentiate whether to show sign in or sign up page and ‘update’ to know whether to show update page.

- sign_frame_destroy():
  This function is used to remove widgets from the sign frame.

- signin_button(user=’vaccinee’):

  This function is used to check whether the entered sign in details are correct. If correct, redirects to the account information window. If not, shows a prompt to enter a valid set of details. It takes value ‘user’ to differentiate between vaccinee and vaccine camp users.

- signup_button(user='vaccinee', update='sign'):

  This function is used to create the widgets for the sign up or update page (depending upon the ‘update’ value), for user vaccinee or vaccine camps (as per the ‘user’ value).

- rbvacname_cond():
  This function is used to alter the signup or update page of the vaccinee by checking whether the vaccinee has taken their first dose.
- datacheck(user='vaccinee', update='sign'):

  This function is used to check whether the entered details in the signup or update page (depending upon the ‘update’ value) are of the correct form, for user vaccinee or vaccine camps (as per the ‘user’ value).

- signupconfirm_button(user='vaccinee', update='sign'):

  This function is used to enter the details given by the vaccinee or vaccine camp (as per the ‘user’ value), from the sign up or update page (depending upon the ‘update’ value), into their respective tables in the MySQL database.

- priority(job,age,mh):
  This function is used to determine the priority score for each vaccinee based on their user data. The priority list is made using this priority score. It is calculated by using the values ‘job’, ‘age’, ‘mh’. This list allows us to prioritize vaccine to the most vulnerable vaccinees at times of vaccine shortage.
- accinfo_frame (user='vaccinee'):

  This function is used to display all the details of the user vaccinee or vaccine camps (as per the ‘user’ value).
- vacsupply_frame(user=’vaccinee’):

  This function is used to show a vaccinee when they can get a vaccine, and allows a vaccine camp to enter how much vaccine they have available.
- updatevacsupply_button():

  This function is used to enter the details given by the vaccine camp regarding how much vaccine they have available into the MySQL database.
- mh(no):

  This function is used to create a string of medical conditions from the stored form of medical conditions (‘no’) in the MySQL database and returns it.
- vp():

  This function collects data from the MySQL database and determines which vaccinees get their vaccine for that day and stores that information in the database for later use.

Working

- First a root.mainloop() is established, to create a loop which gets input from the user interface.
- Then the connecting() function is called.
- Then the vp() function is called.
- Then the above mentioned user defined functions are called in the appropriate order to make the app work.

The following tables are created in a MySQL databse defined as ‘vp’,

```
mysql> use vp;
Database changed
mysql> desc vaccinee;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| ac        | varchar(12) | YES  |     | NULL    |       |
| pass      | varchar(50) | YES  |     | NULL    |       |
| pts       | int         | YES  |     | NULL    |       |
| loc       | varchar(50) | YES  |     | NULL    |       |
| vacname   | varchar(50) | YES  |     | None    |       |
| 1dose     | varchar(3)  | YES  |     | No      |       |
| 1dosedate | date        | YES  |     | NULL    |       |
| 2dose     | varchar(3)  | YES  |     | No      |       |
| 2dosedate | date        | YES  |     | NULL    |       |
| campno    | varchar(12) | YES  |     | NULL    |       |
| name      | varchar(50) | YES  |     | NULL    |       |
| job       | varchar(50) | YES  |     | NULL    |       |
| age       | varchar(3)  | YES  |     | NULL    |       |
| medhis    | varchar(50) | YES  |     | NULL    |       |
| sex       | varchar(6)  | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
15 rows in set (0.03 sec)
mysql> desc vaccamp;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| campno  | varchar(12)  | YES  |     | NULL    |       |
| pass    | varchar(50)  | YES  |     | NULL    |       |
| loc     | varchar(50)  | YES  |     | NULL    |       |
| name    | varchar(50)  | YES  |     | NULL    |       |
| address | varchar(100) | YES  |     | NULL    |       |
| contact | varchar(10)  | YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
mysql> desc vacsupply;
+-----------+-------------+------+-----+---------+-------
| Field     | Type        | Null | Key | Default | Extra
+-----------+-------------+------+-----+---------+-------
| supdate   | date        | YES  |     | NULL    |
| loc       | varchar(50) | YES  |     | NULL    |
| campno    | varchar(12) | YES  |     | NULL    |
| covidose  | int         | YES  |     | NULL    |
| covaxdose | int         | YES  |     | NULL    |
+-----------+-------------+------+-----+---------+-------
5 rows in set (0.00 sec)
```

Hardware and Software Requirement

Minimum System Requirements

- OS
    - Windows 7 or 10 or higher
    - Mac OS X 10.11 or higher
    - Linux RHEL 6/7 or higher
- Architecture: x86, 64-bit CPU (Intel / AMD Architecture)
- RAM: 4GB
- Free Disk Space: 8GB

Scope for Future Enhancement

Modifications and Future Improvements

- The prioritizing algorithm can be made even more sophisticated and accurate.
- If the supply of vaccine can be pre-determined better, we could schedule appointment months prior.
- The GUI could be updated to be even more sophisticated, and be visually pleasing. And be ported for many other applications involving appointment scheduling.
- Source code could be optimized in the future for better performance when a large number of users are using it at a time.
- Source could have been programmed to be even more modular and future proof. So that all modification needed in future could be implemented without changing the source code.
- Even more sophisticated and useful features can be implemented to make vaccination even more efficient.

Source Code

```
#importing necessary modules
from tkinter import * #for gui
from PIL import ImageTk, Image #for the program logo
from tkcalendar import DateEntry #for the interactive calendar
import mysql.connector as sql #for connecting mysql with python

#initialising root gui window
root=Tk()

#connecting to mysql server
def connecting():
    global connection, cursor
    connection=sql.connect(host='localhost',user='root',passwd='1234',database='vp')
    cursor=connection.cursor(buffered=True)
    rootwin()

#initialising gui
def rootwin():
    global root, label_welcome, button_vaccinee, button_vaccamp, curdate_int,
curdate_str, curdate_str_ori, tmrwdate_str, tmrwdate_str_ori
    root.title('Vacination Planner')
    root.iconbitmap('D:/RP/Study/Python/Python Codes/Project/Plus.ico')
    root.resizable(0, 0)
    label_welcome=Label(root, text='Vaccine Proximity Database Management')
    label_welcome.grid(row=0, column=1)
    button_vaccinee=Button(root, text='  Vaccinee  ', command=lambda:[sign_frame(),
sign_frame(source='up')])
    button_vaccinee.grid(row=1, column=0, padx=10, pady=10)
    button_vaccamp=Button(root, text='Vaccine Camp',
command=lambda:[sign_frame(user='vaccamp'), sign_frame(user='vaccamp',source='up')])
    button_vaccamp.grid(row=1, column=2, padx=10, pady=10)
    #required values
    exe='select current_date'
    cursor.execute(exe)
    curdate_str_ori=str(cursor.fetchone()[0])
    curdate_int=int(curdate_str_ori[0:4]+curdate_str_ori[5:7]+curdate_str_ori[8:10])
```

```
    curdate_str=dateformat(curdate_str_ori)
    exe='select date_add("{}", interval 1 day)'.format(curdate_str_ori, )
    cursor.execute(exe)
    tmrwdate_str_ori=str(cursor.fetchone()[0])
    tmrwdate_str=dateformat(tmrwdate_str_ori)


#removing welcome page
def rootwin_destroy():
    label_welcome.destroy()
    button_vaccinee.destroy()
    button_vaccamp.destroy()


#converts yyyy-mm-dd to dd-mm-yyyy
def dateformat(dfrmt):
    return dfrmt[8:10]+'-'+dfrmt[5:7]+'-'+dfrmt[0:4]


#initialising vaccinee's and vaccine camp's signin and signup page
def sign_frame(user='vaccinee', source='in', update='sign'):
    global user_signinid, user_signinpass, frame_signin, frame_signup, button_signup,
button_backroot
    rootwin_destroy()
    #signin page
    if source=='in':
        #vaccinee signin page
        frame_signin_txt='Vaccinee Sign In'
        label_signinid_txt='Aadhar Number'
        #vaccamp signin page
        if user=='vaccamp':
            frame_signin_txt='Vaccine Camp Sign In'
            label_signinid_txt='Camp Number'
        frame_signin=LabelFrame(root, text=frame_signin_txt, padx=5, pady=5)
        frame_signin.grid(column=0, row=0, padx=10, pady=10)
        user_signinid=Entry(frame_signin)
        user_signinid.grid(row=0, column=1, columnspan=2, padx=5, pady=5)
        user_signinpass=Entry(frame_signin, show='*')
        user_signinpass.grid(row=1, column=1, columnspan=2, padx=5, pady=5)
        label_signinid=Label(frame_signin, text=label_signinid_txt)
        label_signinid.grid(row=0, column=0)
```

```
        label_signinpass=Label(frame_signin, text='Password')
        label_signinpass.grid(row=1, column=0)
        button_signin=Button(frame_signin, text='Sign In',
command=lambda:signin_button(user))
        button_signin.grid(row=2, column=0, padx=10, pady=10, columnspan=3)
    #signup and update page
    else:
        #signup page
        if update=='sign':
            #vaccinee signup
            frame_signup_txt='Vaccinee Sign Up'
            #vaccamp signup
            if user=='vaccamp':
                frame_signup_txt='Vaccine Camp Sign Up'
            frame_signup=LabelFrame(root, text=frame_signup_txt, padx=5, pady=5)
            frame_signup.grid(column=0, row=1, padx=10, pady=10)
            button_signup=Button(frame_signup, text='          Sign Up          ',
command=lambda:signup_button(user))
            button_signup.grid(row=0, column=1, padx=10, pady=10)
            button_backroot=Button(frame_signup, text='          Back          ',
command=lambda:[sign_frame_destroy(), rootwin()])
            button_backroot.grid(row=1, column=1, padx=10, pady=10)
        #update page
        else:
            #vaccinee update page
            frame_signupdate_txt='Vaccinee Update'
            #vaccamp update page
            if user=='vaccamp':
                frame_signupdate_txt='Vaccine Camp Update'
            frame_signup=LabelFrame(root, text=frame_signupdate_txt, padx=5, pady=5)
            frame_signup.grid(column=0, row=1, padx=10,pady=10)


#signin and signup page
def sign_frame_destroy():
    frame_signin.destroy()
    frame_signup.destroy()
```

```
#code to be executed when the sigin button is pressed
def signin_button(user='vaccinee'):
    global cur_user_id
    #step for vaccinee sigin process
    if user=='vaccinee':
        exe='select ac, pass from vaccinee'
    #step for vaccamp sigin process
    else:
        exe='select campno, pass from vaccamp'
    cursor.execute(exe)
    creds=cursor.fetchall()
    #checking entered details with details in databases
    if (user_signinid.get(), user_signinpass.get()) in creds:
        cur_user_id=user_signinid.get()
        sign_frame_destroy()
        accinfo_frame(user)
    #message to be displayed when incorrect details entered
    else:
        incrt_signin_txt='Aadhar'
        if user=='vaccamp':
            incrt_signin_txt='Camp'
        messagebox.showerror('Incorrect '+incrt_signin_txt+' Number or Password',
'Enter a valid 12 digit '+incrt_signin_txt+' Number or Check whether the entered
details are correct.')


#code to be executed when the signup or update button is pressed
def signup_button(user='vaccinee', update='sign'):
    #vaccinee signup or update button
    if user=='vaccinee':
        global pat_signupaadhar, pat_signupname, pat_signupage, pat_signupmedhis1,
pat_signupmedhis2, pat_signupmedhis3, pat_signupmedhis4, pat_signupmedhis5,
pat_signupmedhis6, pat_signupmedhis7, pat_signupmedhis8, pat_signupmedhis9,
pat_signupmedhis10, pat_signuppass, button_signupconfirm, pat_signupsex, pat_signuploc,
pat_signupjob, pat_signupvacdose, rbvacname_cond1, rbvacname_cond2
        if update=='sign':
            frame_signin.destroy()
            button_signup.destroy()
            button_backroot.destroy()
```

```
            #entries to get details from the user as text
            pat_signupaadhar=Entry(frame_signup)
            pat_signupaadhar.grid(row=0,column=1,columnspan=2, padx=5, pady=5)
        pat_signupname=Entry(frame_signup)
        pat_signupname.grid(row=1,column=1,columnspan=2, padx=5, pady=5)
        #radiobutton to get details from the user as mcqs
        list_rbsex=[('Male','Male'),('Female','Female')]
        pat_signupsex=StringVar()
        pat_signupsex.set('Male')
        rb_sex_count=0
        for view_sex, val_sex in list_rbsex:
            Radiobutton(frame_signup, text=view_sex, variable=pat_signupsex,
value=val_sex, command=lambda:None).grid(row=2+rb_sex_count,column=1,columnspan=2)
            rb_sex_count+=1
        pat_signupage=Entry(frame_signup)
        pat_signupage.grid(row=4,column=1,columnspan=2, padx=5, pady=5)
        #drop down box to get details from the user from a list of options
        list_loc=['Tiruvallur','Sriperumbudur','Chennai North','Chennai South','Chennai
Central']
        pat_signuploc=StringVar()
        pat_signuploc.set('Tiruvallur')
        drop_loc=OptionMenu(frame_signup, pat_signuploc, *list_loc)
        drop_loc.grid(row=5, column=1, columnspan=2, padx=5, pady=5)
        list_job=['Health Workers','Staffs of Congregate Settings','School
Employee','Public Workers','IT','Others']
        pat_signupjob=StringVar()
        pat_signupjob.set('Health Workers')
        drop_job=OptionMenu(frame_signup, pat_signupjob, *list_job)
        drop_job.grid(row=6, column=1, columnspan=2, padx=5, pady=5)
        #check button to get details from the user as check marks
        pat_signupmedhis1=IntVar()
        pat_signupmedhis2=IntVar()
        pat_signupmedhis3=IntVar()
        pat_signupmedhis4=IntVar()
        pat_signupmedhis5=IntVar()
        pat_signupmedhis6=IntVar()
        pat_signupmedhis7=IntVar()
        pat_signupmedhis8=IntVar()
        pat_signupmedhis9=IntVar()
```

```
        pat_signupmedhis10=IntVar()

        Checkbutton(frame_signup, text='Cancer',
variable=pat_signupmedhis1).grid(row=7, column=1)

        Checkbutton(frame_signup, text='Chronic Kidney Disease',
variable=pat_signupmedhis2).grid(row=8, column=1)

        Checkbutton(frame_signup, text='Chronic Lung Disease',
variable=pat_signupmedhis3).grid(row=9, column=1)

        Checkbutton(frame_signup, text='Neurological Conditions',
variable=pat_signupmedhis4).grid(row=10, column=1)

        Checkbutton(frame_signup, text='Diabetes',
variable=pat_signupmedhis5).grid(row=11, column=1)

        Checkbutton(frame_signup, text='Pregnancy',
variable=pat_signupmedhis6).grid(row=12, column=1)

        Checkbutton(frame_signup, text='Heart Conditions',
variable=pat_signupmedhis7).grid(row=13, column=1)

        Checkbutton(frame_signup, text='HIV Infection',
variable=pat_signupmedhis8).grid(row=14, column=1)

        Checkbutton(frame_signup, text='Weakend Immune System',
variable=pat_signupmedhis9).grid(row=15, column=1)

        Checkbutton(frame_signup, text='Liver Disease',
variable=pat_signupmedhis10).grid(row=16, column=1)

        list_medhis=['Cancer', 'Chronic Kidney Disease', 'Chronic Lung Disease',
'Neurological Conditions', 'Diabetes', 'Pregnancy', 'Heart Conditions', 'HIV
Infection', 'Weakend Immune System', 'Liver Disease']

        list_rbvacdose=[('Yes','Yes'),('No','No')]

        #to check whether to give the option of choosing 1st dose vaccine name and date

        pat_signupvacdose=StringVar()

        pat_signupvacdose.set('Yes')

        rb_vacdose_count=0

        for view_vacdose, val_vacdose in list_rbvacdose:

            Radiobutton(frame_signup, text=view_vacdose, variable=pat_signupvacdose,
value=val_vacdose,
command=lambda:None).grid(row=17+rb_vacdose_count,column=1,columnspan=2)

            rb_vacdose_count+=1

        rbvacname_cond1=None

        rbvacname_cond2=True

        rbvacname_cond()

        button_checkvac=Button(frame_signup, text='Check',command=rbvacname_cond)

        button_checkvac.grid(row=18,column=3, padx=10, pady=10)

        pat_signuppass=Entry(frame_signup, show='*')

        pat_signuppass.grid(row=22,column=1,columnspan=2, padx=5, pady=5)
```

```
        if update=='sign':
            label_ac=Label(frame_signup, text='Aadhar Number')
            label_ac.grid(row=0,column=0)
        label_name=Label(frame_signup, text='Name')
        label_name.grid(row=1,column=0)
        label_sex=Label(frame_signup, text='Sex')
        label_sex.grid(row=2,column=0)
        label_age=Label(frame_signup, text='Age')
        label_age.grid(row=4,column=0)
        label_loc=Label(frame_signup, text='Location')
        label_loc.grid(row=5,column=0)
        label_job=Label(frame_signup, text='Job')
        label_job.grid(row=6,column=0)
        label_medh=Label(frame_signup, text='Medical History')
        label_medh.grid(row=7,column=0)
        label_vacdose=Label(frame_signup, text='Whether 1st Dose Administered')
        label_vacdose.grid(row=17,column=0)
        label_pass=Label(frame_signup, text='Password')
        label_pass.grid(row=22,column=0)
        if update=='sign':
            button_signupconfirm=Button(frame_signup,
text='Confirm',command=lambda:datacheck())
            button_signupconfirm.grid(row=23,column=1,columnspan=2, padx=10, pady=10)
        else:
            button_updateconfirm=Button(frame_signup,
text='Update',command=lambda:datacheck(update='update'))
            button_updateconfirm.grid(row=23,column=1,columnspan=2, padx=10, pady=10)
        messagebox.showinfo('Note','-> Aadhar Number: Enter a valid 12 digit natural
number.\n-> Name: Enter a valid name (should only contain alphabets and spaces, atleast
one alphabet and not more than 50 characters)\n-> Age: Enter a valid integer from 18 to
125.\n-> Whether 1st Dose Administered: If No, select No and click Check. If Yes,
select Yes and click Check.\n-> Date of 1st Dose: Enter a valid date from 16-01-2021 to
'+curdate_str+'.\n-> Password: Enter a valid password having a minimum of 8 characters
and a maximum of 50 characters.')
        if update=='sign':
            button_signin_frsignup=Button(frame_signup, text='Or Sign
In',command=lambda:[sign_frame_destroy(), sign_frame(), sign_frame(source='up')])
            button_signin_frsignup.grid(row=24,column=1, columnspan=2, padx=10,
pady=10)
```

```
        else:
            button_updatecancel=Button(frame_signup,
text='Cancel',command=lambda:[accinfo_frame(), frame_signup.destroy()])
            button_updatecancel.grid(row=24,column=1,columnspan=2, padx=10, pady=10)
    #vaccamp signup or update button
    else:
        global vc_signupcampno, vc_signupname, vc_signuploc, vc_signupaddress,
vc_signupcontact, vc_signuppass, vc_button_signupconfirm
        if update=='sign':
            frame_signin.destroy()
            button_signup.destroy()
            button_backroot.destroy()
            vc_signupcampno=Entry(frame_signup)
            vc_signupcampno.grid(row=0,column=1,columnspan=2, padx=5, pady=5)
        vc_signupname=Entry(frame_signup)
        vc_signupname.grid(row=1,column=1,columnspan=2, padx=5, pady=5)
        list_loc=['Tiruvallur','Sriperumbudur','Chennai North','Chennai South','Chennai
Central']
        vc_signuploc=StringVar()
        vc_signuploc.set('Tiruvallur')
        drop_loc=OptionMenu(frame_signup, vc_signuploc, *list_loc)
        drop_loc.grid(row=2, column=1, columnspan=2, padx=5, pady=5)
        vc_signupaddress=Entry(frame_signup)
        vc_signupaddress.grid(row=3,column=1,columnspan=2, padx=5, pady=5)
        vc_signupcontact=Entry(frame_signup)
        vc_signupcontact.grid(row=4, column=1, columnspan=2, padx=5, pady=5)
        vc_signuppass=Entry(frame_signup, show='*')
        vc_signuppass.grid(row=5,column=1,columnspan=2, padx=5, pady=5)
        if update=='sign':
            vc_label_campno=Label(frame_signup, text='Camp Number')
            vc_label_campno.grid(row=0,column=0)
        vc_label_name=Label(frame_signup, text='Name')
        vc_label_name.grid(row=1,column=0)
        vc_label_loc=Label(frame_signup, text='Location')
        vc_label_loc.grid(row=2,column=0)
        vc_label_address=Label(frame_signup, text='Address')
        vc_label_address.grid(row=3,column=0)
        vc_label_contact=Label(frame_signup, text='Contact')
        vc_label_contact.grid(row=4,column=0)
```

```
        vc_label_pass=Label(frame_signup, text='Password')

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

        if update=='sign':

            vc_button_signupconfirm=Button(frame_signup,
text='Confirm',command=lambda:datacheck(user='vaccamp'))

            vc_button_signupconfirm.grid(row=6,column=1,columnspan=2, padx=10, pady=10)

        else:

            vc_button_updateconfirm=Button(frame_signup,
text='Update',command=lambda:datacheck(user='vaccamp', update='update'))

            vc_button_updateconfirm.grid(row=6,column=1,columnspan=2, padx=10, pady=10)

        messagebox.showinfo('Note','-> Camp Number: Enter a valid 12 digit natural
number.\n-> Name: Enter a valid name (should only contain alphabets and spaces, atleast
one alphabet and not more than 50 characters)\n-> Address: Enter a valid string atleast
1 character long and a maximum length of 100.\n-> Contact: Enter a valid 10 digit
natural number.\n-> Password: Enter a valid password having a minimum of 8 characters
and a maximum of 50 characters.')

        if update=='sign':

            vc_button_signin_frsignup=Button(frame_signup, text='Or Sign
In',command=lambda:[sign_frame_destroy(), sign_frame(user='vaccamp'),
sign_frame(user='vaccamp', source='up')])

            vc_button_signin_frsignup.grid(row=7,column=1, columnspan=2, padx=10,
pady=10)

        else:

            vc_button_updatecancel=Button(frame_signup,
text='Cancel',command=lambda:[accinfo_frame(user='vaccamp'), frame_signup.destroy()])

            vc_button_updatecancel.grid(row=7,column=1,columnspan=2, padx=10, pady=10)


#function to check whether to give the option of choosing 1st dose vaccine name and
date

def rbvacname_cond():

    global rbvacname_cond1, rbvacname_cond2, label_vacname, pat_signupvacname,
list_rbvacname_del, label_vacdate, cal_vacdate

    if (pat_signupvacdose.get()=='Yes' and rbvacname_cond1!='Y') or rbvacname_cond2:

        label_vacname=Label(frame_signup, text='Vaccine Name')

        label_vacname.grid(row=19,column=0)

        list_rbvacname=[('Covishield','Covishield'),('Covaxin','Covaxin')]

        pat_signupvacname=StringVar()

        pat_signupvacname.set('Covishield')

        rb_vacname_count=0

        list_rbvacname_del=[]

        for view_vacname, val_vacname in list_rbvacname:

            rb=Radiobutton(frame_signup, text=view_vacname, variable=pat_signupvacname,
value=val_vacname, command=lambda:None)

            rb.grid(row=19+rb_vacname_count,column=1,columnspan=2)
```

```
            rb_vacname_count+=1
            list_rbvacname_del.append(rb)
        rbvacname_cond1='Y'
        rbvacname_cond2=False
        label_vacdate=Label(frame_signup, text='Date of 1st Dose')
        label_vacdate.grid(row=21,column=0, padx=5, pady=5)
        cal_vacdate=DateEntry(frame_signup, locale='en_US', date_pattern='yyyy/MM/dd')
        cal_vacdate.grid(row=21,column=1,columnspan=2, padx=5, pady=5)
    elif pat_signupvacdose.get()=='No' and rbvacname_cond1!='N' and
rbvacname_cond1!=None:
        for w in list_rbvacname_del:
            w.destroy()
        label_vacname.destroy()
        label_vacdate.destroy()
        cal_vacdate.destroy()
        rbvacname_cond1='N'

#to check whether the given data is in the correct form
def datacheck(user='vaccinee', update='sign'):
    #checking data entered in vaccinee signup or update page
    if user=='vaccinee':
        exe='select ac from vaccinee;'
        cursor.execute(exe)
        db_aadharlist=cursor.fetchall()
        name_cond=True
        if pat_signupname.get().strip()=='':
            name_cond=False
        else:
            for i in pat_signupname.get().strip():
                if i.isalpha() or (i==' '):
                    pass
                else:
                    name_cond=False
                    break
        if pat_signupvacdose.get()=='Yes':
            giv_date=str(cal_vacdate.get_date())
            giv_date=int(giv_date[0:4]+giv_date[5:7]+giv_date[8:10])
```

```
        if update=='sign' and (len(pat_signupaadhar.get())!=12 or (not
pat_signupaadhar.get().isdigit())):

            messagebox.showerror('Invalid Aadhar Number','Enter a valid 12 digit
natural number')

        elif update=='sign' and ((pat_signupaadhar.get(),) in db_aadharlist):

            messagebox.showerror('Account already present', 'There is already an
account created with the entered Aadhar Number')

        elif len(pat_signupname.get().strip())>50 or (not name_cond):

            messagebox.showerror('Invalid Name','Enter a valid name (should only
contain alphabets and spaces, atleast one alphabet and not more than 50 characters)')

        elif (not pat_signupage.get().isdigit()) or int(pat_signupage.get())>125 or
int(pat_signupage.get())<18 :

            messagebox.showerror('Invalid Age','Enter a valid integer from 18 to 125')

        elif pat_signupvacdose.get()=='Yes' and (20210115>giv_date or
giv_date>curdate_int):

            messagebox.showerror('Invalid Date', 'Enter a valid date from 16-01-2021 to
'+curdate_str+'.')

        elif len(pat_signuppass.get())<8 or len(pat_signuppass.get())>50:

            messagebox.showerror('Invalid Password','Enter a valid password having a
minimum of 8 characters and a maximum of 50 characters')

        else:

            signupconfirm_button(user, update)

    #checking data entered in vaccamp signup or update page

    else:

        exe='select campno from vaccamp'

        cursor.execute(exe)

        db_campnolist=cursor.fetchall()

        vc_name_cond=True

        if vc_signupname.get().strip()=='':

            vc_name_cond=False

        else:

            for i in vc_signupname.get().strip():

                if i.isalpha() or (i==' '):

                    pass

                else:

                    vc_name_cond=False

                    break

        if update=='sign' and (len(vc_signupcampno.get())!=12 or (not
vc_signupcampno.get().isdigit())):

            messagebox.showerror('Invalid Camp Number','Enter a valid 12 digit natural
number')
```

```
        elif update=='sign' and ((vc_signupcampno.get(),) in db_campnolist):

            messagebox.showerror('Account already present', 'There is already an
account created with the entered Camp Number')

        elif len(vc_signupname.get().strip())>50 or (not vc_name_cond):

            messagebox.showerror('Invalid Name','Enter a valid name (should only
contain alphabets and spaces, atleast one alphabet and not more than 50 characters)')

        elif len(vc_signupaddress.get().strip())>100 or
len(vc_signupaddress.get().strip())==0:

            messagebox.showerror('Invalid Address','Enter a valid string atleast 1
character long and a maximum length of 100')

        elif (len(vc_signupcontact.get())!=10 or (not
vc_signupcontact.get().isdigit())):

            messagebox.showerror('Invalid Contact','Enter a valid 10 digit natural
number')

        elif len(vc_signuppass.get())<8 or len(vc_signuppass.get())>50:

            messagebox.showerror('Invalid Password','Enter a valid password having a
minimum of 8 characters and a maximum of 50 characters')

        else:

            signupconfirm_button(user, update)


#updating entered data in signup or update page into mysql database

def signupconfirm_button(user='vaccinee', update='sign'):

    #entering data in vaccinee signup or update page into mysql database

    if user=='vaccinee':

        #medical history stored as ones and zeroes for simplicity


pat_signupmedhis=str(pat_signupmedhis1.get())+str(pat_signupmedhis2.get())+str(pat_sign
upmedhis3.get())+str(pat_signupmedhis4.get())+str(pat_signupmedhis5.get())+str(pat_sign
upmedhis6.get())+str(pat_signupmedhis7.get())+str(pat_signupmedhis8.get())+str(pat_sign
upmedhis9.get())+str(pat_signupmedhis10.get())

        job_dict={'Health Workers':1,'Staffs of Congregate Settings':2,'School
Employee':3,'Public Workers':4,'IT':5,'Others':6}


priority(job_dict[pat_signupjob.get()],int(pat_signupage.get()),pat_signupmedhis.count(
'1'))

        #updating entered data in vaccinee update page into mysql database

        if update=='update':

            #updating entered data in vaccinee update page into mysql database where
the vaccinee did get their 1st dose

            if pat_signupvacdose.get()=='Yes':

                cal_vacdate1=str(cal_vacdate.get_date())

                exe='select date_add("{}", interval 84
day)'.format(str(cal_vacdate.get_date()),)

                cursor.execute(exe)
```

```
                cal_vacdate2=cursor.fetchone()[0]

                exe='update vaccinee set pass="{}", pts={}, loc="{}", vacname="{}",
1dose="{}", 1dosedate="{}", 2dosedate="{}", name="{}", job="{}", age="{}", medhis="{}",
sex="{}" where ac="{}";'.format(pat_signuppass.get(), pts, pat_signuploc.get(),
pat_signupvacname.get(), pat_signupvacdose.get(), cal_vacdate1, cal_vacdate2,
pat_signupname.get().strip(), pat_signupjob.get(), pat_signupage.get(),
pat_signupmedhis, pat_signupsex.get(), cur_user_id)

                cursor.execute(exe)

            #updating entered data in vaccinee update page into mysql database where
the vaccinee didn't get their 1st dose

            elif pat_signupvacdose.get()=='No':

                exe='update vaccinee set pass="{}", pts={}, loc="{}", name="{}",
job="{}", age="{}", medhis="{}", sex="{}" where ac="{}";'.format(pat_signuppass.get(),
pts, pat_signuploc.get(), pat_signupname.get().strip(), pat_signupjob.get(),
pat_signupage.get(), pat_signupmedhis, pat_signupsex.get(), cur_user_id)

                cursor.execute(exe)

        #entering data in vaccinee update page into mysql database

        else:

            #entering data in vaccinee update page into mysql database where the
vaccinee did get their 1st dose

            if pat_signupvacdose.get()=='Yes':

                cal_vacdate1=str(cal_vacdate.get_date())

                exe='select date_add("{}", interval 84
day)'.format(str(cal_vacdate.get_date()),)

                cursor.execute(exe)

                cal_vacdate2=cursor.fetchone()[0]

                attributes=(pat_signupaadhar.get() ,pat_signuppass.get(), pts,
pat_signuploc.get(), pat_signupvacname.get(), pat_signupvacdose.get(), cal_vacdate1,
cal_vacdate2, pat_signupname.get().strip(), pat_signupjob.get(), pat_signupage.get(),
pat_signupmedhis, pat_signupsex.get())

                exe='insert into vaccinee (ac, pass, pts, loc, vacname, 1dose,
1dosedate, 2dosedate, name, job, age, medhis, sex) values {};'.format(attributes,)

                cursor.execute(exe)

            #entering data in vaccinee update page into mysql database where the
vaccinee didn't get their 1st dose

            elif pat_signupvacdose.get()=='No':

                attributes=(pat_signupaadhar.get() , pat_signuppass.get(), pts,
pat_signuploc.get(), pat_signupname.get().strip(), pat_signupjob.get(),
pat_signupage.get(), pat_signupmedhis, pat_signupsex.get())

                exe='insert into vaccinee (ac, pass, pts, loc, name, job, age, medhis,
sex) values {};'.format(attributes,)

                cursor.execute(exe)
```

```
        #going back to previous page
        if update=='update':
            messagebox.showinfo('Info', 'Updated Details added to Database')
            frame_signup.destroy()
            accinfo_frame()
        else:
            messagebox.showinfo('Info', 'Details added to Database')
            frame_signup.destroy()
            sign_frame()
            sign_frame(source='up')
        #running vaccine distribution program again for new user
        vp()
    #entering data in vaccamp signup or update page into mysql database
    else:
        #updating entered data in vaccamp update page into mysql database
        if update=='update':
            exe='update vaccamp set pass="{}", loc="{}", name="{}", address="{}",
contact="{}" where campno="{}";'.format(vc_signuppass.get(), vc_signuploc.get(),
vc_signupname.get().strip(), vc_signupaddress.get(), vc_signupcontact.get(),
cur_user_id)
            cursor.execute(exe)
        #entering data in vaccamp signup page into mysql database
        else:
            attributes=(vc_signupcampno.get(), vc_signuppass.get(), vc_signuploc.get(),
vc_signupname.get().strip(), vc_signupaddress.get(), vc_signupcontact.get())
            exe='insert into vaccamp (campno, pass, loc, name, address, contact) values
{};'.format(attributes,)
            cursor.execute(exe)
        #going back to previous page
        if update=='update':
            messagebox.showinfo('Info', 'Updated Details added to Database')
            frame_signup.destroy()
            accinfo_frame(user='vaccamp')
        else:
            messagebox.showinfo('Info', 'Details added to Database')
            frame_signup.destroy()
            sign_frame(user='vaccamp')
            sign_frame(user='vaccamp', source='up')
```

```
    #ordering table vaccinee in mysql as per their priority
    exe='alter table vaccinee order by pts DESC'
    cursor.execute(exe)
    connection.commit()

#gives scores according to priority
def priority(job,age,mh):
    global pts
    pts=0
    crit=0
    if job==1:
        pts+=12000
        crit=1
    elif age>=75:
        pts+=11000+age-74
        crit=2
    elif 65<=age<=74 and mh>=2:
        pts+=10000+age-64+((mh-1)*10)
        crit=2
    elif job==2:
        pts+=9000
        crit=1
    elif age>=65 and mh>=1:
        pts+=8000+age-64+(mh*10)
        crit=2
    elif age>=65:
        pts+=7000+age-64
        crit=2
    elif mh>=2:
        pts+=6000+((mh-1)*10)
        crit=2
    elif job==3:
        pts+=5000
        crit=1
    elif job==4:
        pts+=4000
        crit=1
```

```
elif 18<=age<=64 and mh>=1:
    pts+=3000+age-15+(mh*10)
    crit=2
elif job==5:
    pts+=2000
    crit=1
elif job==6:
    pts+=1000
    crit=1
if crit==2:
    if job==1:
        pts+=600
    elif job==2:
        pts+=500
    elif job==3:
        pts+=400
    elif job==4:
        pts+=300
    elif job==5:
        pts+=200
    elif job==6:
        pts+=100
elif crit==1:
    if age>=75:
        pts+=600+age-74
    elif 65<=age<=74 and mh>=2:
        pts+=500+age-64+((mh-1)*10)
    elif age>=65 and mh>=1:
        pts+=400+age-64+(mh*10)
    elif age>=65:
        pts+=300+age-64
    elif mh>=2:
        pts+=200+((mh-1)*10)
    elif 18<=age<=64 and mh>=1:
        pts+=100+age-15+(mh*10)
```

```
#to display account info of users
def accinfo_frame(user='vaccinee'):
    global frame_accinfo
    #frame name for vaccinee
    if user=='vaccinee':
        frame_accinfo_txt='Vaccinee Acount Information'
        exe='select * from vaccinee where ac="{}";'.format(cur_user_id, )
    #frame name for vaccamp
    else:
        frame_accinfo_txt='Vaccine Camp Acount Information'
        exe='select * from vaccamp where campno="{}";'.format(cur_user_id, )
    cursor.execute(exe)
    cur_user_info=cursor.fetchone()
    frame_accinfo=LabelFrame(root, text=frame_accinfo_txt, padx=5, pady=5)
    frame_accinfo.grid(padx=10,pady=10)
    #vaccinee details
    if user=='vaccinee':
        #to convert ones and zeroes of medical history back into strings
        mh_str=mh(cur_user_info[13])
        if mh_str=='':
            mh_str='None'
        label_accvaccinee1=Label(frame_accinfo, text='\nAadhar Number:
'+cur_user_info[0]+'\n\nName: '+cur_user_info[10]+'\n\nSex:
'+cur_user_info[14]+'\n\nAge: '+cur_user_info[12]+'\n\nLocation:
'+cur_user_info[3]+'\n\nJob: '+cur_user_info[11]+'\n\nMedical History: '+mh_str)
        label_accvaccinee1.grid()
        if cur_user_info[7]=='Yes':
            label_accvaccinee2=Label(frame_accinfo, text='\nVaccine Status:
'+cur_user_info[4]+' 2nd Dose Completed'+'\n\nDate of 1st Dose:
'+dateformat(str(cur_user_info[6]))+'\n\nDate of 2nd Dose:
'+dateformat(str(cur_user_info[8])))
        elif cur_user_info[5]=='Yes':
            label_accvaccinee2=Label(frame_accinfo, text='\nVaccine Status:
'+cur_user_info[4]+' 1st Dose Completed'+'\n\nDate of 1st Dose:
'+dateformat(str(cur_user_info[6]))+'\n\nEligibile for 2nd Dose from:
'+dateformat(str(cur_user_info[8]))+'\n')
        else:
            label_accvaccinee2=Label(frame_accinfo, text='\nVaccine Status: Needs to
get 1st Dose\n')
        label_accvaccinee2.grid()
        #user should not update after getting vaccine even once with app
        exe='select campno from vaccinee where ac="{}";'.format(cur_user_id,)
```

```
        cursor.execute(exe)

        exist_vaccinee=cursor.fetchone()[0]

        if exist_vaccinee==None:

            button_accinfoupdate=Button(frame_accinfo, text='Update Details',
command=lambda:[frame_accinfo.destroy(),sign_frame(source='up',
update='update'),signup_button(update='update')])

            button_accinfoupdate.grid(padx=10, pady=10)

        button_accinfocheck=Button(frame_accinfo, text='Check for Vaccine',
command=lambda:[vacsupply_frame()])

        button_accinfocheck.grid(padx=10, pady=10)

        button_accinfologout=Button(frame_accinfo, text='Log
Out',command=lambda:[messagebox.showinfo('Info','Successfully logged out'),
sign_frame(), sign_frame(source='up'), frame_accinfo.destroy()])

        button_accinfologout.grid(padx=10, pady=10)

    #vaccamp details

    else:

        label_accvaccamp=Label(frame_accinfo, text='\nCamp Number:
'+cur_user_info[0]+'\n\nName: '+cur_user_info[3]+'\n\nLocation:
'+cur_user_info[2]+'\n\nAddress: '+cur_user_info[4]+'\n\nContact:
'+cur_user_info[5]+'\n')

        label_accvaccamp.grid()

        button_accinfoupdate=Button(frame_accinfo, text='Update
Details',command=lambda:[frame_accinfo.destroy(),sign_frame(user='vaccamp',
source='up', update='update'),signup_button(user='vaccamp', update='update')])

        button_accinfoupdate.grid(padx=10, pady=10)

        vc_button_accinfosupply=Button(frame_accinfo, text='Update Vaccine Stock',
command=lambda:[vacsupply_frame(user='vaccamp')])

        vc_button_accinfosupply.grid(padx=10, pady=10)

        button_accinfologout=Button(frame_accinfo, text='Log
Out',command=lambda:[messagebox.showinfo('Info','Successfully logged out'),
sign_frame(user='vaccamp'), sign_frame(user='vaccamp', source='up'),
frame_accinfo.destroy()])

        button_accinfologout.grid(padx=10, pady=10)


#frame for checking or entering vaccinee availability

def vacsupply_frame(user='vaccinee'):

    #frame for vaccinee to check vaccine availability

    if user=='vaccinee':

        frame_accinfo.destroy()

        frame_vacsupply=LabelFrame(root, text='Vaccine Availability', padx=5, pady=5)

        frame_vacsupply.grid(padx=10, pady=10)

        exe='select 1dose, 1dosedate, 2dose, 2dosedate, campno, vacname from vaccinee
where ac="{}";'.format(cur_user_id, )

        cursor.execute(exe)
```

```
        vac_status=cursor.fetchone()

        exe='select loc, name, address, contact from vaccamp where
campno="{}";'.format(vac_status[4], )

        cursor.execute(exe)

        vc_info=cursor.fetchone()

        if vac_status[0]=='No':

            txt='\n1st Dose Vaccine Not Available Yet. Try again tomorrow\n'

        elif vac_status[0]=='Yes' and vac_status[2]=='No':

            if vac_status[4]==None:

                txt='\n2nd Dose Vaccine Not Available Yet. Try again tomorrow\n'

            else:

                txt='\n1st Dose of '+vac_status[5]+' at Camp '+vac_status[4]+' on
'+str(vac_status[1])+'\n\nCamp Info\n\nName: '+vc_info[1]+'\n\nLocation:
'+vc_info[0]+'\n\nAddress: '+vc_info[2]+'\n\nContact: '+vc_info[3]+'\n'

        elif vac_status[0]=='Yes' and vac_status[2]=='Yes':

            txt='\n2nd Dose of '+vac_status[5]+' at Camp '+vac_status[4]+' on
'+str(vac_status[3])+'\n\nCamp Info\n\nName: '+vc_info[1]+'\n\nLocation:
'+vc_info[0]+'\n\nAddress: '+vc_info[2]+'\n\nContact: '+vc_info[3]+'\n'

        label_vacstat=Label(frame_vacsupply, text=txt)

        label_vacstat.grid()

        button_back=Button(frame_vacsupply, text='Back',
command=lambda:[frame_vacsupply.destroy(), accinfo_frame()])

        button_back.grid(padx=10, pady=10)

    #frame for vaccamp to enter vaccine stock

    else:

        global vc_covidose, vc_covaxdose, vc_frame_vacsupply

        frame_accinfo.destroy()

        vc_frame_vacsupply=LabelFrame(root, text='Vaccine Stock', padx=5, pady=5)

        vc_frame_vacsupply.grid(padx=10, pady=10)

        exe='select supdate, covidose, covaxdose from vacsupply where (campno="{}" and
(supdate="{}" or supdate="{}"));'.format(cur_user_id, curdate_str_ori,
tmrwdate_str_ori)

        cursor.execute(exe)

        existing_sup=cursor.fetchall()

        tocovi=tocovax=tmrwcovi=tmrwcovax=0

        for rec in existing_sup:

            if str(rec[0])==curdate_str_ori:

                tocovi=rec[1]

                tocovax=rec[2]
```

```
            elif str(rec[0])==tmrwdate_str_ori:

                tmrwcovi=rec[1]

                tmrwcovax=rec[2]

        vc_label_exist=Label(vc_frame_vacsupply, text='\nVaccine Supply for the day
'+curdate_str+':\n\nCovishield: '+str(tocovi)+'\n\nCovaxin:
'+str(tocovax)+'\n\n\nVaccine Supply for the day '+tmrwdate_str+':\n\nCovishield:
'+str(tmrwcovi)+'\n\nCovaxin: '+str(tmrwcovax)+'\n\n')

        vc_label_exist.grid(row=0, column=0, columnspan=3)

        vc_label_vacsupply=Label(vc_frame_vacsupply, text='Enter vaccine supply details
for the day '+tmrwdate_str+':')

        vc_label_vacsupply.grid(row=1, column=0, columnspan=3)

        vc_covidose=Entry(vc_frame_vacsupply)

        vc_covidose.grid(row=2,column=1,columnspan=2, padx=5, pady=5)

        vc_covaxdose=Entry(vc_frame_vacsupply)

        vc_covaxdose.grid(row=3,column=1,columnspan=2, padx=5, pady=5)

        vc_label_covidose=Label(vc_frame_vacsupply, text='Covishield')

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

        vc_label_covaxdose=Label(vc_frame_vacsupply, text='Covaxin')

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

        vc_button_updatevacsupply=Button(vc_frame_vacsupply, text='Update Stock',
command=updatevacsupply_button)

        vc_button_updatevacsupply.grid(columnspan=3, padx=10, pady=10)

        messagebox.showwarning('Important',"Could only update supply details for
tomorrow. Changes of supply for tomorrow should be made today itself. Tomorrow's supply
can't be changed tomorrow.")

        vc_button_cancelvacsupply=Button(vc_frame_vacsupply, text='Cancel',
command=lambda:[vc_frame_vacsupply.destroy(), accinfo_frame(user='vaccamp')])

        vc_button_cancelvacsupply.grid(columnspan=3, padx=10, pady=10)


#entering vaccine stocks to table vacsupply

def updatevacsupply_button():

    #checking whether entered details are correct

    if (not vc_covidose.get().isdigit()):

        messagebox.showerror('Invalid Covishield Dose','Enter a valid integer')

    elif (not vc_covaxdose.get().isdigit()):

        messagebox.showerror('Invalid Covaxin Dose','Enter a valid integer')

    else:

        exe='select campno from vacsupply where
supdate="{}";'.format(tmrwdate_str_ori,)

        cursor.execute(exe)

        existing_vc=cursor.fetchall()
```

```
        #updating vaccine supply for an existing supply date
        if (cur_user_id, ) in existing_vc:
            messagebox.showinfo('Info','Details updated to database successfully')
            exe='update vacsupply set covidose={}, covaxdose={} where (supdate="{}" and
campno="{}");'.format(int(vc_covidose.get()), int(vc_covaxdose.get()),
tmrwdate_str_ori, cur_user_id)
            cursor.execute(exe)
            connection.commit()
        #entering fresh vaccine supply for a supply date
        else:
            messagebox.showinfo('Info','Details added to database successfully')
            exe='select loc from vaccamp where campno="{}";'.format(cur_user_id, )
            cursor.execute(exe)
            vc_loc=cursor.fetchone()[0]
            vac_dose=(tmrwdate_str_ori, vc_loc, cur_user_id, int(vc_covidose.get()),
int(vc_covaxdose.get()))
            exe='insert into vacsupply values {};'.format(vac_dose, )
            cursor.execute(exe)
            connection.commit()
        vc_frame_vacsupply.destroy()
        accinfo_frame(user='vaccamp')
        #running vaccine distribution program again for updated vaccine stock
        vp()


#converts medicl history ones and zeroes to strings
def mh(no):
    list_medhis=['Cancer', 'Chronic Kidney Disease', 'Chronic Lung Disease',
'Neurological Conditions', 'Diabetes', 'Pregnancy', 'Heart Conditions', 'HIV
Infection', 'Weakend Immune System', 'Liver Disease']
    s=''
    for i in range(len(no)):
        if no[i]=='1':
            s+=list_medhis[i]+', '
    return s[:len(s)-2]


#function to distribute vaccines to vaccinee according to priority and vaccine
availability
def vp():
    exe='select loc, campno, covidose, covaxdose from vacsupply where
supdate="{}";'.format(curdate_str_ori,)
    cursor.execute(exe)
```

```
    list_vacsupply=cursor.fetchall()
    exe='select loc, ac, vacname from vaccinee where (1dose="No" or (2dose="No" and
2dosedate<=current_date()));'
    cursor.execute(exe)
    list_vaccinee=cursor.fetchall()
    #to stop function if no eligible vaccinees are present
    if list_vaccinee==[]:
        return
    #creating dictionary where key is location and value is details of vaccamp in given
location
    dict_vacsupply={}
    for rec in list_vacsupply:
        if rec[0] not in dict_vacsupply:
            dict_vacsupply[rec[0]]=[list(rec[1:4])]
        else:
            dict_vacsupply[rec[0]].append(list(rec[1:4]))
    #creating dictionary where key is location and value is details of vaccinee in
given location
    dict_vaccinee={}
    for rec in list_vaccinee:
        val_vaccinee=list(rec[1:3])
        val_vaccinee.append('No')
        if rec[0] not in dict_vaccinee:
            dict_vaccinee[rec[0]]=[val_vaccinee]
        else:
            dict_vaccinee[rec[0]].append(val_vaccinee)
    #loop for vaccine distribution
    for loc in dict_vacsupply:
        for vc in dict_vacsupply[loc]:
            if loc in dict_vaccinee:
                for rec in dict_vaccinee[loc]:
                    if rec[2]=='Yes':
                        continue
                    novac=False
                    if rec[1]=='Covishield':
                        index=1
                    elif rec[1]=='Covaxin':
                        index=2
```

```
                    else:
                        novac=True
                    if not novac and vc[index]-1>=0:
                        exe='update vaccinee set 2dose="Yes", 2dosedate="{}",
campno="{}" where ac="{}";'.format(curdate_str_ori, vc[0], rec[0])
                        cursor.execute(exe)
                        connection.commit()
                        vc[index]=vc[index]-1
                        rec[2]=('Yes')
                    elif novac:
                        for index in [1, 2]:
                            if vc[index]-1>=0:
                                if index==1:
                                    vac='Covishield'
                                else:
                                    vac='Covaxin'
                                exe='select date_add("{}", interval 84
day)'.format(curdate_str_ori,)
                                cursor.execute(exe)
                                vacdate2=str(cursor.fetchone()[0])
                                exe='update vaccinee set vacname="{}", 1dose="Yes",
1dosedate="{}", 2dosedate="{}", campno="{}" where ac="{}";'.format(vac,
curdate_str_ori, vacdate2, vc[0], rec[0])
                                cursor.execute(exe)
                                connection.commit()
                                vc[index]=vc[index]-1
                                rec[2]=('Yes')
                                break
                #updating vacsupply for accurate results when function is run again
                exe='update vacsupply set covidose={}, covaxdose={} where (campno="{}"
and supdate="{}");'.format(vc[1], vc[2], vc[0], curdate_str_ori)
                cursor.execute(exe)
                connection.commit()
connecting()
#running vaccine distribution program
vp()
#starting tkinter loop to interact with the gui
root.mainloop()
```

Output

- The MySQL database ‘vp’ has the following tables,

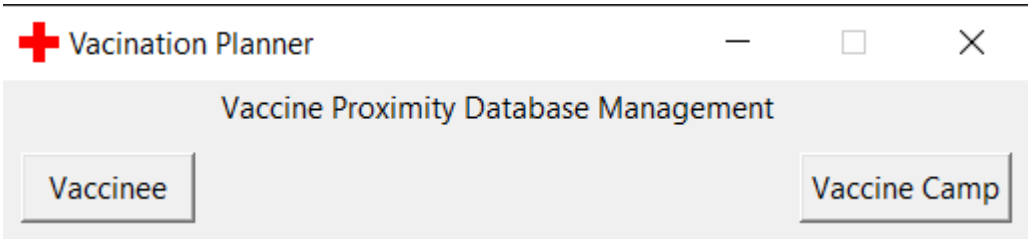
```
mysql> select current_date();
+----------------+
| current_date() |
+----------------+
| 2022-01-03     |
+----------------+
1 row in set (0.00 sec)

mysql> select * from vaccinee;
+--------------+----------+------+---------------+-----------+-------+-------------+-------+-------------+---------+-----------------+-----------------------------+------+-----------+--------+
| ac           | pass     | pts  | loc           | vacname   | 1dose | 1dosedate   | 2dose | 2dosedate   | campno  | name            | job                         | age  | medhis    | sex    |
+--------------+----------+------+---------------+-----------+-------+-------------+-------+-------------+---------+-----------------+-----------------------------+------+-----------+--------+
| 666666666666 | 66666666 | 12116 | Tiruvallur    | Covaxin   | Yes   | 2021-01-20  | No    | 2021-04-14  | NULL    | Aditi           | Health Workers              | 21   | 0001000000 | Female |
| 222222222222 | 22222222 | 12000 | Tiruvallur    | Covaxin   | Yes   | 2021-02-03  | No    | 2021-04-28  | NULL    | Anush           | Health Workers              | 21   | 0000000000 | Male   |
| 444444444444 | 44444444 | 11406 | Tiruvallur    | Covaxin   | Yes   | 2021-06-20  | No    | 2021-09-12  | NULL    | Kavya           | School Employee             | 80   | 0000000000 | Female |
| 123456789333 | 12345333 | 11401 | Tiruvallur    | Covishield | Yes   | 2021-12-01  | No    | 2022-02-23  | NULL    | Three           | School Employee             | 75   | 0000000000 | Female |
| 123456789111 | 12345111 | 10116 | Tiruvallur    | Covishield | Yes   | 2021-09-01  | No    | 2021-11-24  | NULL    | One             | Others                      | 70   | 0000100010 | Female |
| 123456789444 | 12345444 |  9220 | Tiruvallur    | Covishield | Yes   | 2022-01-01  | No    | 2022-03-26  | NULL    | Four            | Staffs of Congregate Settings | 50   | 0000101010 | Male   |
| 888888888888 | 88888888 |  9000 | Tiruvallur    | Covaxin   | Yes   | 2021-09-06  | No    | 2021-11-29  | NULL    | Lux             | Staffs of Congregate Settings | 19   | 0000000000 | Female |
| 123456789222 | 12345222 |  8311 | Tiruvallur    | Covishield | Yes   | 2021-11-01  | No    | 2022-01-24  | NULL    | Two             | Public Workers              | 65   | 0000001000 | Male   |
| 123456789555 | 55555555 |  5125 | Tiruvallur    | None      | No    | NULL        | No    | NULL        | NULL    | Five            | School Employee             | 30   | 0010000000 | Female |
| 123456789666 | 12345666 |  4130 | Tiruvallur    | None      | No    | NULL        | No    | NULL        | NULL    | Six             | Public Workers              | 35   | 0000010000 | Male   |
| 123456789888 | 12345888 |  3135 | Tiruvallur    | None      | No    | NULL        | No    | NULL        | NULL    | Eight           | Others                      | 40   | 1000000000 | Male   |
| 777777777777 | 77777777 |  2000 | Tiruvallur    | Covaxin   | Yes   | 2021-06-05  | No    | 2021-08-28  | NULL    | Ram Prasath P   | IT                          | 21   | 0000000000 | Male   |
| 111111111111 | 11111111 |  2000 | Chennai North | Covishield | Yes   | 2021-10-01  | No    | 2021-12-24  | NULL    | B Taarun        | IT                          | 21   | 0000000000 | Male   |
| 333333333333 | 33333333 |  2000 | Chennai Central | None    | No    | NULL        | No    | NULL        | NULL    | Soham Banerjee  | IT                          | 21   | 0000000000 | Male   |
| 555555555555 | 55555555 |  2000 | Chennai South | None      | No    | NULL        | No    | NULL        | NULL    | Hari Hara Sudhan V S | IT                     | 21   | 0000000000 | Male   |
+--------------+----------+------+---------------+-----------+-------+-------------+-------+-------------+---------+-----------------+-----------------------------+------+-----------+--------+
15 rows in set (0.00 sec)

mysql> select * from vaccamp;
+--------------+----------+---------------+-------------+-----------------------------------------------+------------+
| campno       | pass     | loc           | name        | address                                       | contact    |
+--------------+----------+---------------+-------------+-----------------------------------------------+------------+
| 777777777777 | 77777777 | Tiruvallur    | RP Camp     | No. 7, Street 7, Tiruvallur, Chennai - 77     | 7777777777 |
| 111111111111 | 11111111 | Chennai North | T Camp      | No. 1, Street 1, Chennai North, Chennai - 11  | 1111111111 |
| 333333333333 | 33333333 | Chennai Central | SB Camp   | No. 3, Street 3, Chennai Central, Chennai - 33 | 3333333333 |
| 555555555555 | 55555555 | Chennai South | HHS Camp    | No. 5, Street 5, Chennai South, Chennai - 55  | 5555555555 |
| 222222222222 | 22222222 | Tiruvallur    | A Camp      | No. 2, Street 2, Tiruvallur, Chennai - 22     | 2222222222 |
+--------------+----------+---------------+-------------+-----------------------------------------------+------------+
5 rows in set (0.00 sec)

mysql> select * from vacsupply;
+------------+------------+--------------+----------+-----------+
| supdate    | loc        | campno       | covidose | covaxdose |
+------------+------------+--------------+----------+-----------+
| 2022-01-03 | Tiruvallur | 777777777777 |        0 |         9 |
| 2022-01-03 | Tiruvallur | 222222222222 |        2 |         5 |
+------------+------------+--------------+----------+-----------+
2 rows in set (0.00 sec)
```

- When the program is run, the connecting() function is called,



- Then the vp() function is called which changes the database as,

```
mysql> select * from vaccinee;
+-------------+----------+------+-----------------+-----------+-------+-------------+-------+-------------+-------------+-----------------------+-------------------------------+------+-----------+--------+
| ac          | pass     | pts  | loc             | vacname   | 1dose | 1dosedate   | 2dose | 2dosedate   | campno      | name                  | job                           | age  | medhis    | sex    |
+-------------+----------+------+-----------------+-----------+-------+-------------+-------+-------------+-------------+-----------------------+-------------------------------+------+-----------+--------+
| 666666666666 | 66666666 | 12116 | Tiruvallur      | Covaxin   | Yes   | 2021-01-20  | Yes   | 2022-01-03  | 777777777777 | Aditi                 | Health Workers                | 21   | 0001000000 | Female |
| 222222222222 | 22222222 | 12000 | Tiruvallur      | Covaxin   | Yes   | 2021-02-03  | Yes   | 2022-01-03  | 777777777777 | Anush                 | Health Workers                | 21   | 0000000000 | Male   |
| 444444444444 | 44444444 | 11406 | Tiruvallur      | Covaxin   | Yes   | 2021-06-20  | Yes   | 2022-01-03  | 777777777777 | Kavya                 | School Employee               | 80   | 0000000000 | Female |
| 123456789333 | 12345333 | 11401 | Tiruvallur      | Covishield | Yes   | 2021-12-01  | No    | 2022-02-23  | NULL         | Three                 | School Employee               | 75   | 0000000000 | Female |
| 123456789111 | 12345111 | 10116 | Tiruvallur      | Covishield | Yes   | 2021-09-01  | Yes   | 2022-01-03  | 222222222222 | One                   | Others                        | 70   | 0000100010 | Female |
| 123456789444 | 12345444 |  9220 | Tiruvallur      | Covishield | Yes   | 2022-01-01  | No    | 2022-03-26  | NULL         | Four                  | Staffs of Congregate Settings | 50   | 0000101010 | Male   |
| 888888888888 | 88888888 |  9000 | Tiruvallur      | Covaxin   | Yes   | 2021-09-06  | Yes   | 2022-01-03  | 777777777777 | Lux                   | Staffs of Congregate Settings | 19   | 0000000000 | Female |
| 123456789222 | 12345222 |  8311 | Tiruvallur      | Covishield | Yes   | 2021-11-01  | No    | 2022-01-24  | NULL         | Two                   | Public Workers                | 65   | 0000001000 | Male   |
| 123456789555 | 55555555 |  5125 | Tiruvallur      | Covaxin   | Yes   | 2022-01-03  | No    | 2022-03-28  | 777777777777 | Five                  | School Employee               | 30   | 0010000000 | Female |
| 123456789666 | 12345666 |  4130 | Tiruvallur      | Covaxin   | Yes   | 2022-01-03  | No    | 2022-03-28  | 777777777777 | Six                   | Public Workers                | 35   | 0000010000 | Male   |
| 123456789888 | 12345888 |  3135 | Tiruvallur      | Covaxin   | Yes   | 2022-01-03  | No    | 2022-03-28  | 777777777777 | Eight                 | Others                        | 40   | 1000000000 | Male   |
| 777777777777 | 77777777 |  2000 | Tiruvallur      | Covaxin   | Yes   | 2021-06-05  | Yes   | 2022-01-03  | 777777777777 | Ram Prasath P         | IT                            | 21   | 0000000000 | Male   |
| 111111111111 | 11111111 |  2000 | Chennai North   | Covishield | Yes   | 2021-10-01  | No    | 2021-12-24  | NULL         | B Taarun              | IT                            | 21   | 0000000000 | Male   |
| 333333333333 | 33333333 |  2000 | Chennai Central | None      | No    | NULL        | No    | NULL        | NULL         | Soham Banerjee        | IT                            | 21   | 0000000000 | Male   |
| 555555555555 | 55555555 |  2000 | Chennai South   | None      | No    | NULL        | No    | NULL        | NULL         | Hari Hara Sudhan V S  | IT                            | 21   | 0000000000 | Male   |
+-------------+----------+------+-----------------+-----------+-------+-------------+-------+-------------+-------------+-----------------------+-------------------------------+------+-----------+--------+
15 rows in set (0.00 sec)

mysql> select * from vacsupply;
+------------+------------+--------------+----------+-----------+
| supdate    | loc        | campno       | covidose | covaxdose |
+------------+------------+--------------+----------+-----------+
| 2022-01-03 | Tiruvallur | 777777777777 |        0 |         1 |
| 2022-01-03 | Tiruvallur | 222222222222 |        1 |         5 |
+------------+------------+--------------+----------+-----------+
2 rows in set (0.00 sec)
```

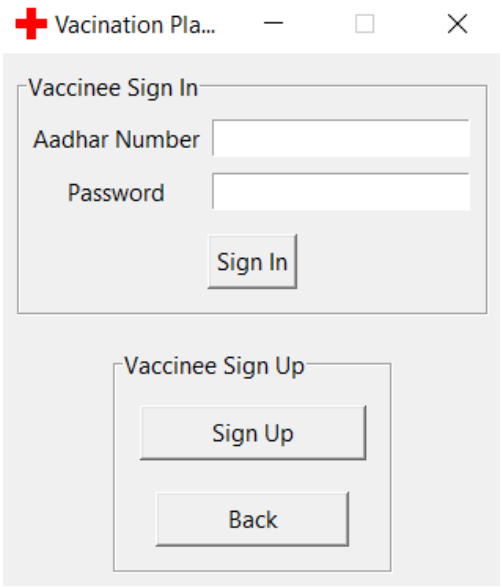
- **Steps for a vaccinee to sign up, then sign in and then finally check vaccine availability or update vaccine stock,**
    1. Click the button according to which user you are in the startup window, which shows Fig 1.
    2. Then click ‘Sign Up’, which then shows Fig 2.
    3. Read the given instructions and click ‘OK’, which then shows Fig 3.
    4. Fill up your details according to the instructions given and the click ‘Confirm’, which then shows Fig 4.
    5. Click ‘OK’, which takes you back to Fig 1.
    6. Now enter your details into sign in page and click ‘Sign in’, which if correct shows you Fig 5.
    7. Now as a vaccinee to check whether you are eligible for a vaccine click ‘Check for Vaccine’, which then shows your eligibility in Fig 6.
    8. Now for a vaccine camp user to enter the available vaccine stock click ‘Update Vaccine Stock’, which takes you to Fig 6.

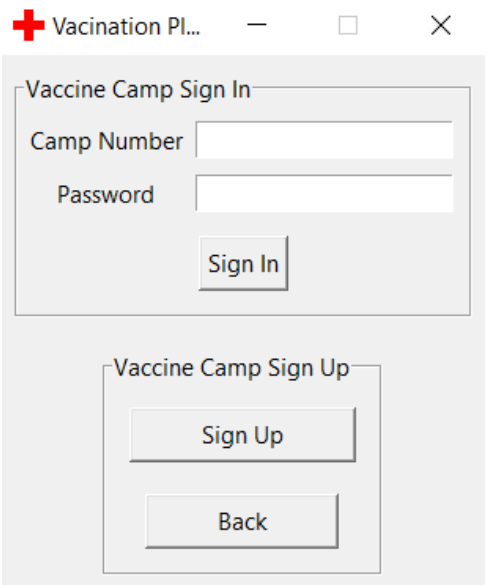
- Now read the instructions in Fig 6 and then click ‘OK’, which takes you to Fig 7.
- Enter the details correctly and click ‘Update Stock’, which shows you Fig 8.
- Click ‘OK’, which takes you back to Fig 5.
- You can click ‘Update Vaccine Stock’ again to check the values you entered.

Note: You cannot update as a vaccinee after you receive a vaccine through this app. Otherwise you can update your details which is similar to the sign up process. Check Vaccine Camp Fig 5.

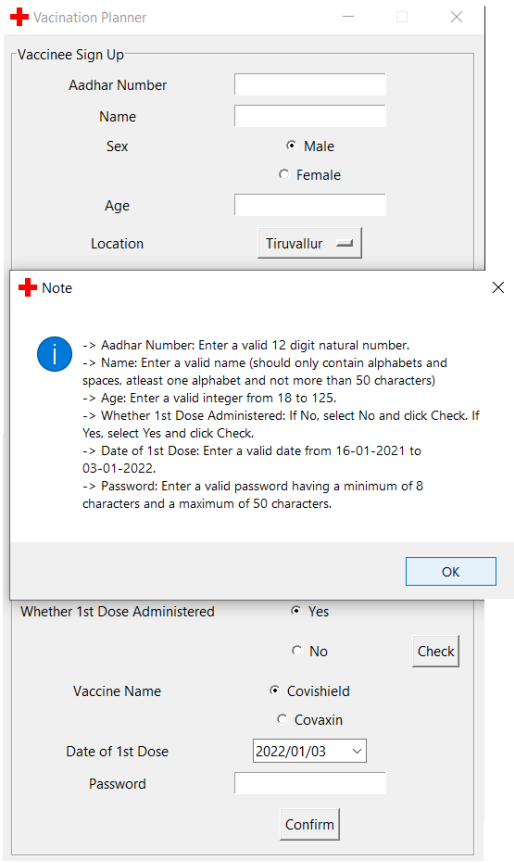
- After Step 5, the MySQL database is changed as shown in Fig 9.1 (vaccinee) and Fig 9.2 (vaccine camp).
- After Step 8.3, the MySQL database is changed as shown in Fig 10.

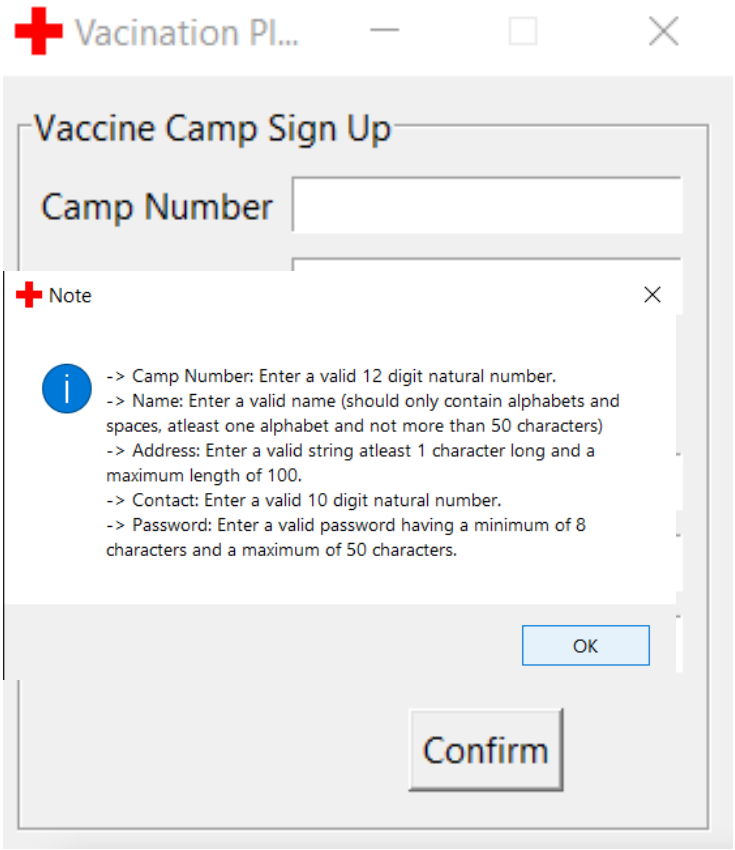
- **Fig 1**



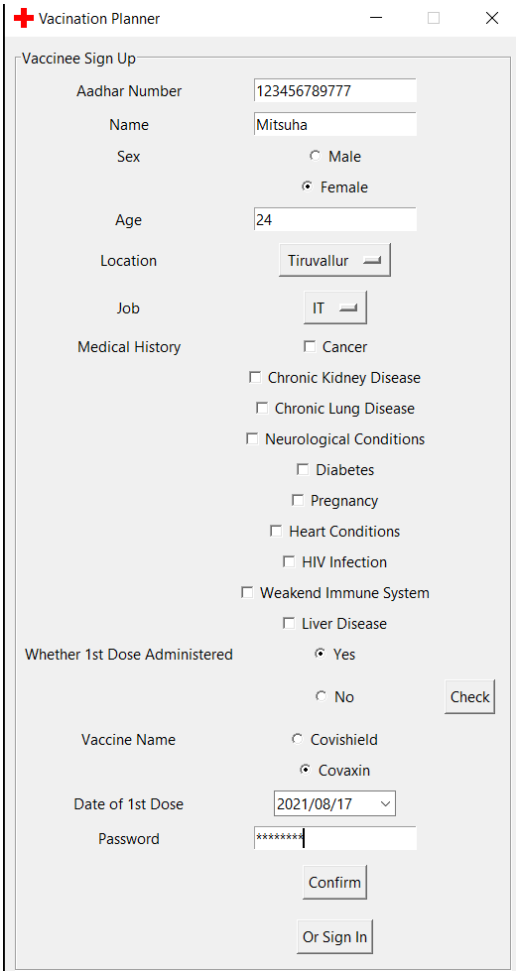


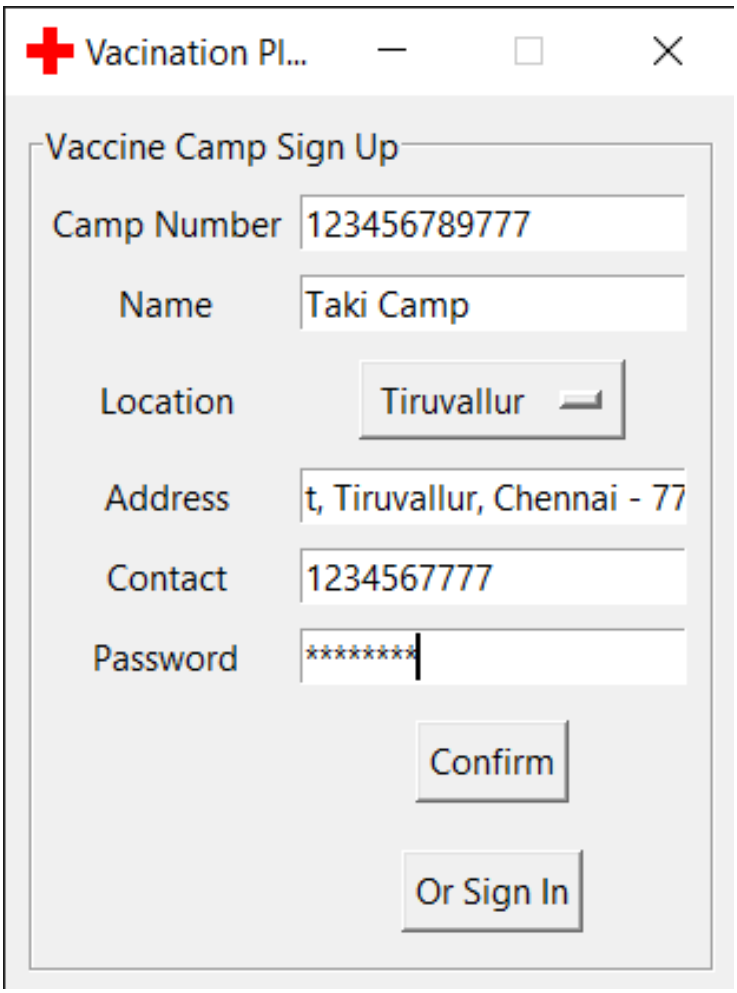
- **Fig 2**


Vacination Planner
Vaccinee Sign Up
Aadhar Number
Name
Sex
Male
Female
Age
Location
Tiruvallur
Note
-> Aadhar Number: Enter a valid 12 digit natural number.
-> Name: Enter a valid name (should only contain alphabets and spaces, atleast one alphabet and not more than 50 characters)
-> Age: Enter a valid integer from 18 to 125.
-> Whether 1st Dose Administered: If No, select No and click Check. If Yes, select Yes and click Check.
-> Date of 1st Dose: Enter a valid date from 16-01-2021 to 03-01-2022.
-> Password: Enter a valid password having a minimum of 8 characters and a maximum of 50 characters.
OK
Whether 1st Dose Administered
Yes
No
Check
Vaccine Name
Covishield
Covaxin
Date of 1st Dose
2022/01/03
Password
Confirm

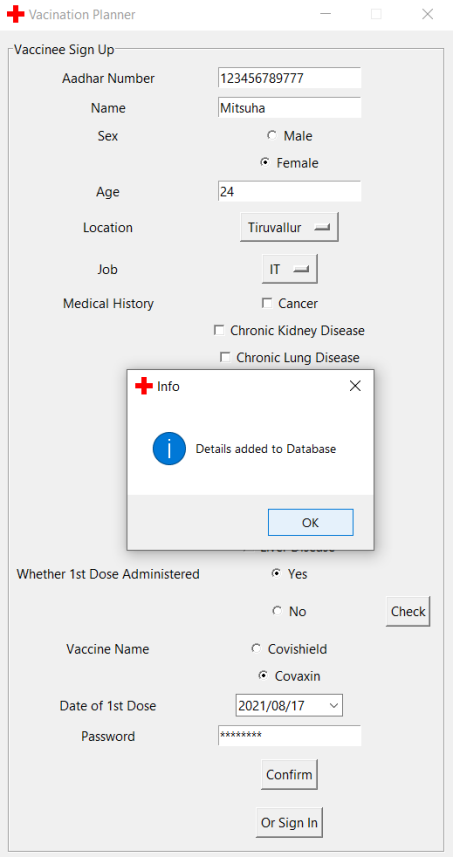

Vacination Pl...
Vaccine Camp Sign Up
Camp Number
Note
-> Camp Number: Enter a valid 12 digit natural number.
-> Name: Enter a valid name (should only contain alphabets and spaces, atleast one alphabet and not more than 50 characters)
-> Address: Enter a valid string atleast 1 character long and a maximum length of 100.
-> Contact: Enter a valid 10 digit natural number.
-> Password: Enter a valid password having a minimum of 8 characters and a maximum of 50 characters.
OK
Confirm

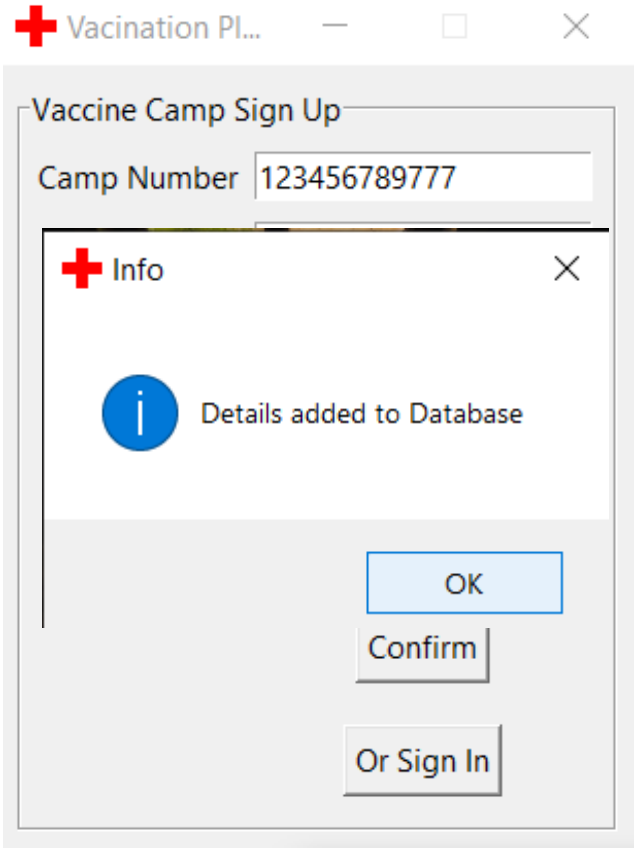
- **Fig 3**


Vacination Planner
Vaccinee Sign Up
Aadhar Number
123456789777
Name
Mitsuha
Sex
Male
Female
Age
24
Location
Tiruvallur
Job
IT
Medical History
Cancer
Chronic Kidney Disease
Chronic Lung Disease
Neurological Conditions
Diabetes
Pregnancy
Heart Conditions
HIV Infection
Weakend Immune System
Liver Disease
Whether 1st Dose Administered
Yes
No
Check
Vaccine Name
Covishield
Covaxin
Date of 1st Dose
2021/08/17
Password
Confirm
Or Sign In

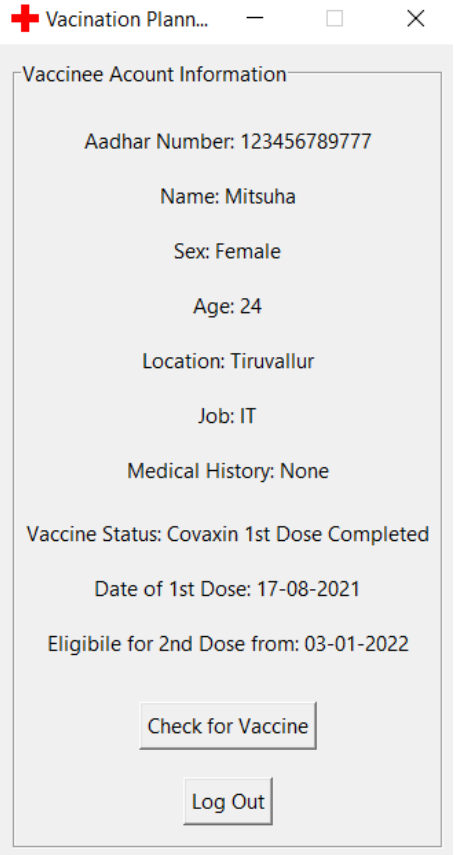

Vacination Pl...
Vaccine Camp Sign Up
Camp Number
123456789777
Name
Taki Camp
Location
Tiruvallur
Address
t, Tiruvallur, Chennai - 77
Contact
1234567777
Password
Confirm
Or Sign In

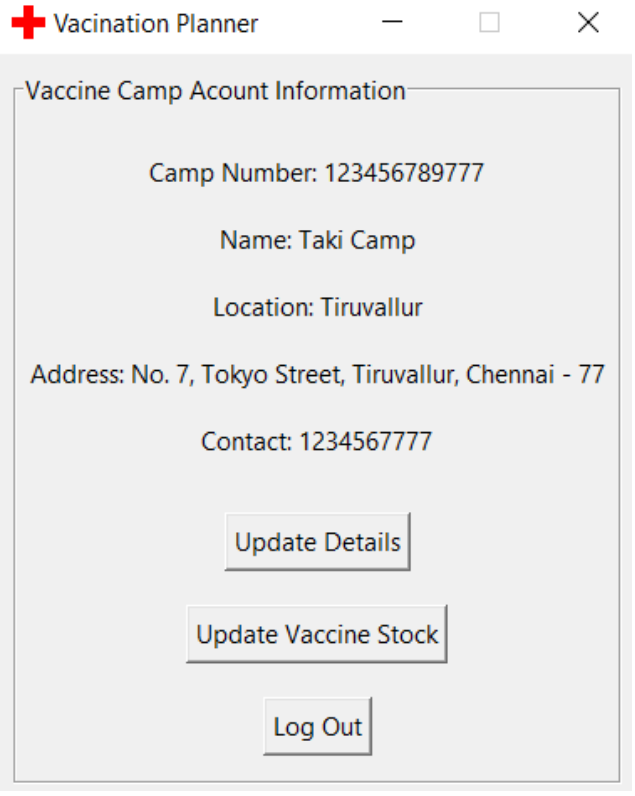
- **Fig 4**


Vacination Planner
Vaccinee Sign Up
Aadhar Number
12345678977
Name
Mitsuha
Sex
Male
Female
Age
24
Location
Tiruvallur
Job
IT
Medical History
Cancer
Chronic Kidney Disease
Chronic Lung Disease
Info
Details added to Database
OK
Whether 1st Dose Administered
Yes
No
Check
Vaccine Name
Covishield
Covaxin
Date of 1st Dose
2021/08/17
Password
****
Confirm
Or Sign In

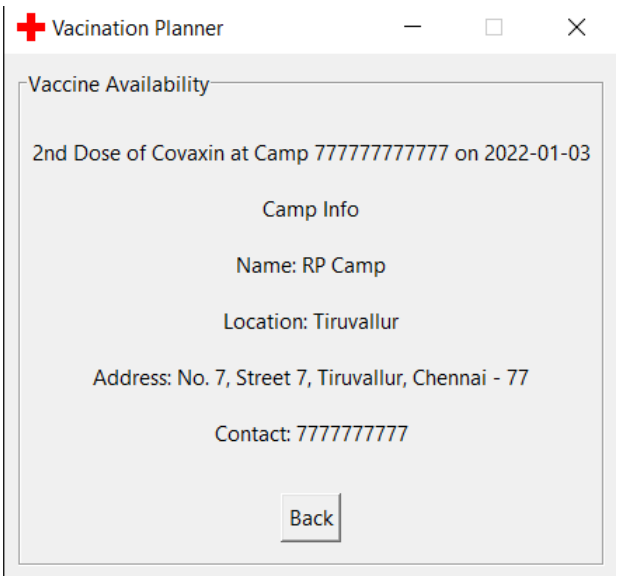

Vacination Pl...
Vaccine Camp Sign Up
Camp Number
123456789777
Info
Details added to Database
OK
Confirm
Or Sign In

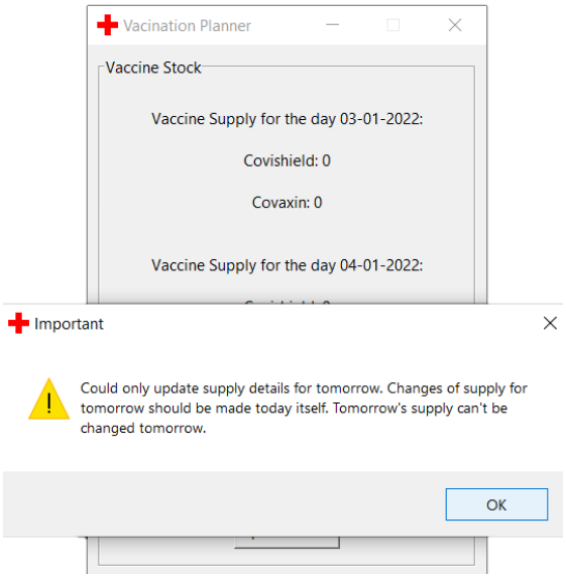
- **Fig 5**


Vacination Plann...
Vaccinee Acount Information
Aadhar Number: 123456789777
Name: Mitsuha
Sex: Female
Age: 24
Location: Tiruvallur
Job: IT
Medical History: None
Vaccine Status: Covaxin 1st Dose Completed
Date of 1st Dose: 17-08-2021
Eligibile for 2nd Dose from: 03-01-2022
Check for Vaccine
Log Out

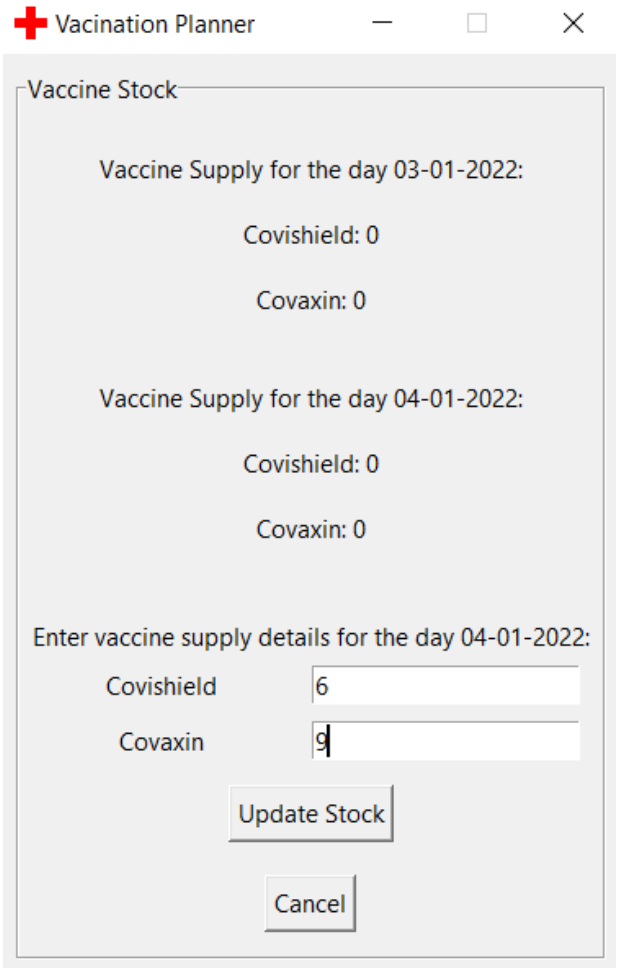

Vacination Planner
Vaccine Camp Acount Information
Camp Number: 123456789777
Name: Taki Camp
Location: Tiruvallur
Address: No. 7, Tokyo Street, Tiruvallur, Chennai - 77
Contact: 1234567777
Update Details
Update Vaccine Stock
Log Out

- **Fig 6**

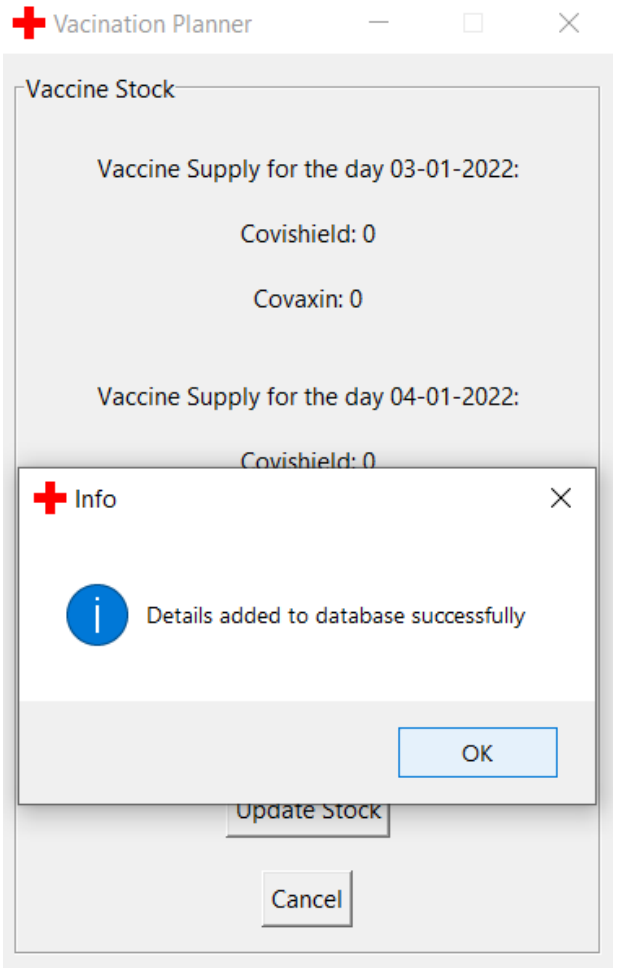




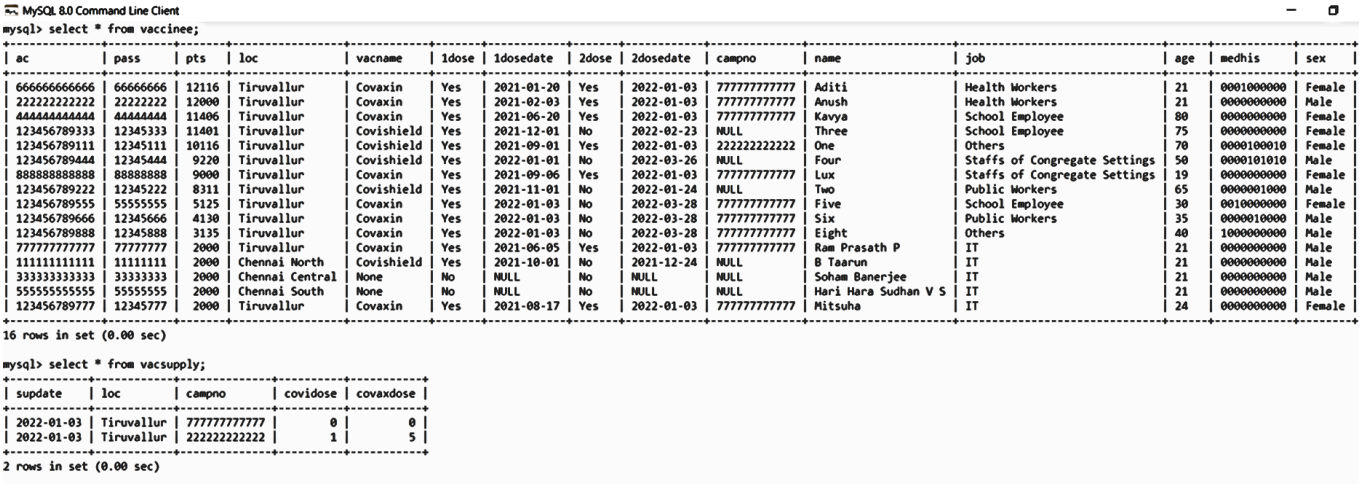
- **Fig 7**



- **Fig 8**



- **Fig 9.1**



MySQL 8.0 Command Line Client

```
mysql> select * from vaccinee;
+-------------+----------+------+-----------+-----------+-------+-----------+-------+-----------+-------------+-----------------------------+-----+------------+--------+
| ac          | pass     | pts  | loc       | vacname   | 1dose | 1dosedate | 2dose | 2dosedate | campno      | name                        | job                         | age | medhis     | sex    |
+-------------+----------+------+-----------+-----------+-------+-----------+-------+-----------+-------------+-----------------------------+-----------------------------+-----+------------+--------+
| 666666666666 | 66666666 | 12116 | Tiruvallur | Covaxin   | Yes   | 2021-01-20 | Yes   | 2022-01-03 | 777777777777 | Aditi                       | Health Workers              | 21  | 0001000000 | Female |
| 222222222222 | 22222222 | 12000 | Tiruvallur | Covaxin   | Yes   | 2021-02-03 | Yes   | 2022-01-03 | 777777777777 | Anush                       | Health Workers              | 21  | 0000000000 | Male   |
| 444444444444 | 44444444 | 11406 | Tiruvallur | Covaxin   | Yes   | 2021-06-20 | Yes   | 2022-01-03 | 777777777777 | Kavya                       | School Employee             | 80  | 0000000000 | Female |
| 123456789333 | 12345333 | 11401 | Tiruvallur | Covishield | Yes   | 2021-12-01 | No    | 2022-02-23 | NULL        | Three                       | School Employee             | 75  | 0000000000 | Female |
| 123456789111 | 12345111 | 10116 | Tiruvallur | Covishield | Yes   | 2021-09-01 | Yes   | 2022-01-03 | 222222222222 | One                         | Others                      | 70  | 0000100010 | Female |
| 123456789444 | 12345444 |  9220 | Tiruvallur | Covishield | Yes   | 2022-01-01 | No    | 2022-03-26 | NULL        | Four                        | Staffs of Congregate Settings | 50  | 0000101010 | Male   |
| 888888888888 | 88888888 |  9000 | Tiruvallur | Covaxin   | Yes   | 2021-09-06 | Yes   | 2022-01-03 | 777777777777 | Lux                         | Staffs of Congregate Settings | 19  | 0000000000 | Female |
| 123456789222 | 12345222 |  8311 | Tiruvallur | Covishield | Yes   | 2021-11-01 | No    | 2022-01-24 | NULL        | Two                         | Public Workers              | 65  | 0000001000 | Male   |
| 123456789555 | 55555555 |  5125 | Tiruvallur | Covaxin   | Yes   | 2022-01-03 | No    | 2022-03-28 | 777777777777 | Five                        | School Employee             | 30  | 0010000000 | Female |
| 123456789666 | 12345666 |  4130 | Tiruvallur | Covaxin   | Yes   | 2022-01-03 | No    | 2022-03-28 | 777777777777 | Six                         | Public Workers              | 35  | 0000010000 | Male   |
| 123456789888 | 12345888 |  3135 | Tiruvallur | Covaxin   | Yes   | 2022-01-03 | No    | 2022-03-28 | 777777777777 | Eight                       | Others                      | 40  | 1000000000 | Male   |
| 777777777777 | 77777777 |  2000 | Tiruvallur | Covaxin   | Yes   | 2021-06-05 | Yes   | 2022-01-03 | 777777777777 | Ram Prasath P               | IT                          | 21  | 0000000000 | Male   |
| 111111111111 | 11111111 |  2000 | Chennai North | Covishield | Yes   | 2021-10-01 | No    | 2021-12-24 | NULL        | B Taarun                    | IT                          | 21  | 0000000000 | Male   |
| 333333333333 | 33333333 |  2000 | Chennai Central | None    | No    | NULL      | No    | NULL      | NULL        | Soham Banerjee              | IT                          | 21  | 0000000000 | Male   |
| 555555555555 | 55555555 |  2000 | Chennai South | None    | No    | NULL      | No    | NULL      | NULL        | Hari Hara Sudhan V S        | IT                          | 21  | 0000000000 | Male   |
| 123456789777 | 12345777 |  2000 | Tiruvallur | Covaxin   | Yes   | 2021-08-17 | Yes   | 2022-01-03 | 777777777777 | Mitsuha                     | IT                          | 24  | 0000000000 | Female |
+-------------+----------+------+-----------+-----------+-------+-----------+-------+-----------+-------------+-----------------------------+-----------------------------+-----+------------+--------+
16 rows in set (0.00 sec)

mysql> select * from vacsupply;
+------------+-----------+-------------+----------+-----------+
| supdate    | loc       | campno      | covidose | covaxdose |
+------------+-----------+-------------+----------+-----------+
| 2022-01-03 | Tiruvallur | 777777777777 |        0 |         0 |
| 2022-01-03 | Tiruvallur | 222222222222 |        1 |         5 |
+------------+-----------+-------------+----------+-----------+
2 rows in set (0.00 sec)
```

- **Fig 9.2**

```
mysql> select * from vaccamp;
+--------------+----------+-----------------+-----------+-----------------------------------------------+------------+
| campno       | pass     | loc             | name      | address                                       | contact    |
+--------------+----------+-----------------+-----------+-----------------------------------------------+------------+
| 777777777777 | 77777777 | Tiruvallur      | RP Camp   | No. 7, Street 7, Tiruvallur, Chennai - 77     | 7777777777 |
| 111111111111 | 11111111 | Chennai North   | T Camp    | No. 1, Street 1, Chennai North, Chennai - 11  | 1111111111 |
| 333333333333 | 33333333 | Chennai Central | SB Camp   | No. 3, Street 3, Chennai Central, Chennai - 33 | 3333333333 |
| 555555555555 | 55555555 | Chennai South   | HHS Camp  | No. 5, Street 5, Chennai South, Chennai - 55  | 5555555555 |
| 222222222222 | 22222222 | Tiruvallur      | A Camp    | No. 2, Street 2, Tiruvallur, Chennai - 22     | 2222222222 |
| 123456789777 | 12345777 | Tiruvallur      | Taki Camp | No. 7, Tokyo Street, Tiruvallur, Chennai - 77 | 1234567777 |
+--------------+----------+-----------------+-----------+-----------------------------------------------+------------+
6 rows in set (0.00 sec)
```

- **Fig 10**

```
mysql> select * from vacsupply;
+------------+------------+--------------+----------+-----------+
| supdate    | loc        | campno       | covidose | covaxdose |
+------------+------------+--------------+----------+-----------+
| 2022-01-03 | Tiruvallur | 777777777777 |        0 |         0 |
| 2022-01-03 | Tiruvallur | 222222222222 |        1 |         5 |
| 2022-01-04 | Tiruvallur | 123456789777 |        6 |         9 |
+------------+------------+--------------+----------+-----------+
3 rows in set (0.00 sec)
```

Bibliography

- Computer Science with Python [Textbook XI] by Sumita Arora
- Computer Science with Python [Textbook XII] by Sumita Arora
- https://youtube.com/playlist?list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV
- https://www.health.nd.gov/covid-19-vaccine-priority-groups