Prashant Tripathi
0801CS211068

REPORT

Objectives Of Project

The Following Code will is based on the Vaccination Drive . The user can get all the important information needed regarding vaccination.

Information regarding all the Functions

1. main: This Function is the main executive function of the code . All other function calls will come under this functions. The main need of this function is due to profiling.

2. Get_OTP: This function will sent a OTP on the number provided by the user((imaginary)).

3. Verify_OTP: This Function will check whether the OTP entered by the user is correct or not.

4. ask_vaccination: This function will ask the use whether he/she wants to apply for vaccination. If user wants then the following function will ask for user details.

5. Apply_Vaccination: This function will ask for the general details of the user that will be needed for verification.

6. vaccine_name: This function will ask user for the vaccine he/she wants covishield or covaccine.

7. Schedule_Appointment: This Function will ask for aadhar number and get the present date and find the first dose date using datetime library function.

8. Set_Vaccination_Venue: This will will ask for user current location and then will generate the most nearest location for his preference.

9. first_dose_message: This function will generate a message containing all the necessary details of appointment.

10. guidelines_for_vaccination: This function will show the guidelines to the user regarding vaccination.

11. schedule_second_dose: This function will generate the date for second dose by datetime library.

12. Verification_for_second_dose: This function will verify that the correct user is trying to accessing the details or not.

13. second_dose_status: This function will show the user total days left for second dose vaccination.

14. second_dose_message: This function will generate a message containing all the necessary details of appointment.

15. first_dose_certificate: This function will generate a certificate for 1st dose.

16. second_dose_certificate: This function will generate a certificate for 2nd dose.

Prashant Tripathi
0801CS211068

CODE

```
# CODE FOR VACCINE REGISTRATION AND CERTIFICATE
# import date time library to get current time and date
import pdb
from datetime import datetime, timedelta
# this library is used to get current location
from geopy.geocoders import Nominatim
# this library is used to get random number
import random
# this library is used for profiling
import cProfile
loc = Nominatim(user_agent="GetLoc")
def main():
    global First_Name , Last_Name , location_for_appointment ,Aadhar_no
    global first_dose_date ,dob,age,vaccine
    Application_id=random.randint(1000,100000)
    print("Hey! Welcome To The Vaccination Drive...\n")
    print("Enter Your Phone Number To Login or Sign up")
    # getting phone number input from user

    phone_no = int(input())

    # this will generate a random OTP
    OTP_generate=random.randint(1000,100000)
    print("Your OTP for Login in vaccination is ", OTP_generate)
    '''
    This function is used to verify OTP entered by the user
    '''
    def Verify_OTP( otp):
        if(otp==OTP_generate):
            print("OTP verified!! Let's Continue ")
        else:
            print("INCORRECT OTP, Please click on Resend OTP To Try Again")
            Get_OTP()

    '''
    This function is used to take input for OTP from user
    '''
    def Get_OTP():
        print( ''' OTP Sent on The Given Phone Number
        Verify To Continue ''')
        global otp
        otp = int(input("Enter the OTP\n"))
        # calling the function to verify OTP
        Verify_OTP(otp)
        # calling the function to get OTP
    Get_OTP()
```

```
    '''
    Below function will generate your second dose certificate
    '''
    def second_dose_certificate(second_dose_date):
        print("\n")
        print('''        Certificate Of Covid - 19 Vaccination
''')
        print('''                    Second Dose
''')
        print("\n")
        print("APPLICATION ID: ", Application_id)
        print("\n")
        print("NAME: "+ First_Name +" "+ Last_Name)
        print("\n")
        print("Gender: "+ gender)
        print("\n")
        print("ID VERIFIED: ", Aadhar_no)
        print("\n")
        print("DATE OF BIRTH: "+ dob)
        print("\n")
        print("AGE: ", age)
        print("\n")
        print("Vaccination Status : Fully Vaccinated (1st Dose)")
        print("\n")
        print("Vaccination Details:")
        print("Vaccine Name: "+ vaccine)
        print("\n")
        print("Vaccine Type: Covid 19 vaccine, non- replacing viral vector")
        print("\n")
        print("Dose Number: 2")
        print("\n")
        print("Date Of First Dose: ", first_dose_date.strftime('%d-%m-%Y'))
        print("\n")
        print("Date of Second Dose: ", second_dose_date.strftime('%d-%m-%Y'))
        print("\n")
        print("Vaccination At : "+ str(location_for_appointment))

    '''
    Below function will count the number of days left for your second dose
    '''
    def second_dose_status():
        global days_left
        days_left = second_dose_date-presentday
         # finding days left by subtracting second dose day with
         # present day using date time library function
        print("Total Days left for Your Second Dose are", days_left)

    '''
    Below Function will take appllication id input from user
    if the entered id is correct then only user can do further process
```

```
    '''
    def Verification_for_second_dose():
        print("Enter the Application id provided during 1st dose")
        apply=int(input())
        # this condition is used to check entered number with application id
        if(apply==Application_id):
            print("You Can Apply On or After: \n",second_dose_date.strftime('%d
             # calling second dose staus function
            second_dose_status()

        else:
            print("Incorrect ID\n")

    '''
    Below function will schedule the second dose date
    '''
    def schedule_second_dose(vaccine,first_dose_date):
        global second_dose_date
        # if covishield vaacine is entered by user than second dose will be aft
        if(vaccine=="Covishield"):
            # time delta will show the date after 90 days from present day
            second_dose_date= first_dose_date +timedelta(90)
        # if covaccine vaacine is entered by user than second dose will be afte
        elif(vaccine=="Covaccine"):
            # time delta will show the date after 28 days from present day
            second_dose_date= first_dose_date +timedelta(28)

    '''
    Below function is used to get first dose certificate of the user
    '''
    def first_dose_certificate(second_dose_date,location_for_appointment):
        # taking application id from user for verification
        print("Enter application ID\n")
        h1=int(input())
        if(h1==Application_id):
            print("\n")
            print('''            Certificate Of Covid - 19 Vaccination
''')
            print('''                                                First Dose
''')
            print("\n")
            print("APPLICATION ID: ",Application_id)
            print("\n")
            print("NAME: "+ First_Name +" "+ Last_Name)
            print("\n")
            print("Gender: "+ gender)
            print("\n")
            print("ID VERIFIED: ",Aadhar_no)
            print("\n")
```

```
            print("DATE OF BIRTH: "+ dob)
            print("\n")
            print("AGE: ",age)
            print("\n")
            print("Vaccination Status : Half Vaccinated (1st Dose)")
            print("\n")
            print("Vaccination Details:")
            print("Vaccine Name: "+ vaccine)
            print("\n")
            print("Vaccine Type: Covid 19 vaccine, non- replacing viral vector"
            print("\n")
            print("Dose Number: 1")
            print("\n")
            print("Dose Date: ",first_dose_date.strftime('%d-%m-%Y'))
            print("\n")
            print("Apply For Second Dose After 2nd Dose Date: ",second_dose_dat
            print("\n")
            print("Vaccination At : "+ str(location_for_appointment))
        else:
            print("Incorrect ID, Try Again..")
            first_dose_certificate(second_dose_date)




    '''
    Below function will show the guidlines for vaccination for user
    '''

    def guidlines_for_vaccination():
        print('''                                    GUIDLINES FOR VACCINATION
''')
        print('''Make sure you have:

        1. A Mask that covers your nose and mouth and fits tightly and comforta
        2. Hand Sanitizer.
        3. The Notification you received about your appointment.
        4. Your ID''')
    '''
    Below function will generate a message confirming the appointment
    of user for second dose with date , time and location
    '''
    def second_dose_message(location_for_appointment,First_Name,Last_Name,dob,a
        print("\n")
        print('''YOUR 2nd DOSE APPOINTMENT IS BOOKED \n''')
        print("Below are the Following Detalis\n")
        print("APPLICATION ID: ",Application_id)
        print("\n")
        print("NAME: "+ First_Name +" "+ Last_Name)
        print("\n")
```

```
        print("Gender: "+ gender)
        print("\n")
        print("AADHAR_NUMBER: " ,Aadhar_no)
        print("\n")
        print("DATE_OF_BIRTH: "+ dob)
        print("\n")
        print("AGE: " ,age)
        print("\n")
        print("VACCINATION_VENUE: "+ str(location_for_appointment))
        print("\n")
        print("Time: 11AM - 3PM")
        print("\n")
        print("Second Dose Date: ", second_dose_date.strftime('%d-%m-%Y'))
        print("\n")
        print("Vaccine Name: "+ vaccine)
        print("\n")
        # Calling Guidlines for vaccination function
        guidlines_for_vaccination()

        '''
    Below function will generate a message confirming the appointment
    of user for first dose with date , time and location
    '''

    def first_dose_message(location_for_appointment ,First_Name ,Last_Name ,dob ,ag
        print("\n")
        print( ''' YOUR 1st DOSE APPOINTMENT IS BOOKED \n ''')
        print("Below are the Following Detalis\n")
        print("APPLICATION_ID: " ,Application_id)
        print("\n")
        print("NAME: "+ First_Name +" "+ Last_Name)
        print("\n")
        print("Gender: "+ gender)
        print("\n")
        print("AADHAR_NUMBER: " ,Aadhar_no)
        print("\n")
        print("DATE_OF_BIRTH: "+ dob)
        print("\n")
        print("AGE: " ,age)
        print("\n")
        print("VACCINATION_VENUE: "+ str(location_for_appointment))
        print("\n")
        print("Time: 11AM - 3PM")
        print("\n")
        print("First Dose Date: ", first_dose_date.strftime('%d-%m-%Y'))
        print("\n")
        print("Vaccine Name: "+ vaccine)
        print("\n")
        # Calling Guidlines for vaccination function
        guidlines_for_vaccination()
```

```
        # calling schedule second dose function
        schedule_second_dose(vaccine,first_dose_date)


    '''
    Below function will generate a new location from user's existing location
    '''
    def Set_Vaccination_Venue():
        global location_for_appointment
        print("Enter Your Location\n")
        # Taking current location as user input
        your_location=input()
        print("Your Appointement Request is Being Processed . . Please Wait . . ")
        getloc = loc.geocode(your_location)
        # generating some random loactions according to user preference
        getloc1 = loc.geocode("Vijay nagar indore")
        getloc2 = loc.geocode("abhay prashal indore")
        getloc3 = loc.geocode("rajwada indore")
        getloc4 = loc.geocode("Geeta bhawan indore")
        getloc5 = loc.geocode("Regal square indore")

        locations=[getloc1,getloc2,getloc3,getloc4,getloc5]
        rand_idx = random.randrange(len(locations))
        random_num = locations[rand_idx]
        location_for_appointment=(random_num)
    # calling first dose message function
        first_dose_message(location_for_appointment,First_Name,Last_Name,dob,ag

        '''
    Below function will schedule appointment for first dose
    '''
    def Schedule_Appointment():
        global Aadhar_no,first_dose_date,presentday
        # asking for user aadhar number
        print("Enter Your Aadhar Number:\n")
        Aadhar_no=int(input())
        presentday = datetime.now()
        # Fetching present day date and time
        ## Setting first dose date after 3 days from present day by date time
        first_dose_date=presentday + timedelta(3)
        # calling set vaccination venue function

        Set_Vaccination_Venue()
    '''
    Below Function will ask for vaccine name that user wants
    '''
    def get_vaccine_name():
        global vaccine
        print("Enter The Vaccine You Want: Covishield or Covaccine:\n")
        vaccine=input()
```

```
        # calling schedule appointment function after that
        Schedule_Appointment()

    '''
    Below function will ask for general details for appointment
    '''

    def Apply_Vaccination():
        global First_Name ,Last_Name,dob,age,Application_id,gender

        # this will generate random id using random library function
        print("Enter your First Name\n")
        # Asking for user name
        First_Name=input()
        print("Enter your Last Name\n")
        Last_Name=input()
        # asking for gender
        print("Your Gender\n")
        gender=input()
        print("Enter Your Date Of Birth\n")
         # asking date of birth
        dob=input()

        print("Enter your Age\n")
        # Asking age input
        age=int(input())
        # Calling get vaccine name function
        get_vaccine_name()




    '''
    Below function will ask user whether he/she wants to apply for vaccination
    '''
    def ask_vaccination():
        global Application_id
        print("DO you want to Apply For 1st Dose Vaccination [Yes/No]")
        s = input()

        if(s=="yes"):
            # calling Apply Vaccination function

            Apply_Vaccination()
        else:
            print("Do you want to apply for second dose [Yes/No]\n")
            s2=input()
            if(s2=="yes"):
                # asking for application ID
                print("Enter First Dose Application Id\n")
```

```
                    s1=int(input())
                    if(s1==Application_id):
                        Apply_Vaccination()
                    else:
                        print("You have not applied for 1st dose , First apply for
                        ask_vaccination()
                else:
                    print("Stay Healthy , Stay Safe ...\n")
                    exit()

    ask_vaccination()
    # calling ask vaccination function
    print("\n")
    print("Your 1st Dose Vaccination is Done!!!!\n")
    global second_dose_date
    print("You can apply for Second Dose after ",second_dose_date.strftime('%d-
    print("For 1st Dose Certificate click [ y or n]\n")
    # asking user whether he wants 2st dose certificate
    pdb.set_trace()
    inp=input()
    if(inp=="y"):
        first_dose_certificate(second_dose_date,location_for_appointment)
        # calling first dose certificate

    print("Do You Want To Check Second Dose Staus:")
    second_inp=input()
    if(second_inp=="y"):
        Verification_for_second_dose()

    print("Do you want to apply for Second dose [y/n]\n")
    # asking whether want to apply for second dose
    s3=input()
    if(s3=="y"):
        second_dose_message(location_for_appointment,First_Name,Last_Name,dob,a
    else:
        print("Stay safe ...")
        exit()

    print("To Download certifacte of your Second Dose click [y/n]\n")
    # asking user whether he wants 2st dose certificate
    in2=input()
    if(in2=="y") :
        second_dose_certificate(second_dose_date)
cProfile.run('main()')
```

OUTPUT

```
PS C:\Users\HP> python -u "c:\Users\HP\Desktop\finalll.py"
Hey! Welcome To The Vaccination Drive...

Enter Your Phone Number To Login or Sign up
9131618018
Your OTP for Login in vaccination is  7041
 OTP Sent on The Given Phone Number
       Verify To Continue
Enter the OTP
7041
OTP verified!! Let's Continue
DO you want to Apply For 1st Dose Vaccination [Yes/No]
yes
Enter your First Name

Prashant
Enter your Last Name

Tripathi
Your Gender

M
Enter Your Date Of Birth

15 Apr 2003
Enter your Age

19
Enter The Vaccine You Want: Covishield or Covaccine:

Covaccine
Enter Your Aadhar Number:

902046447402
Enter Your Location

sgsits
Your Appointement Request is Being Processed . Please Wait..
```

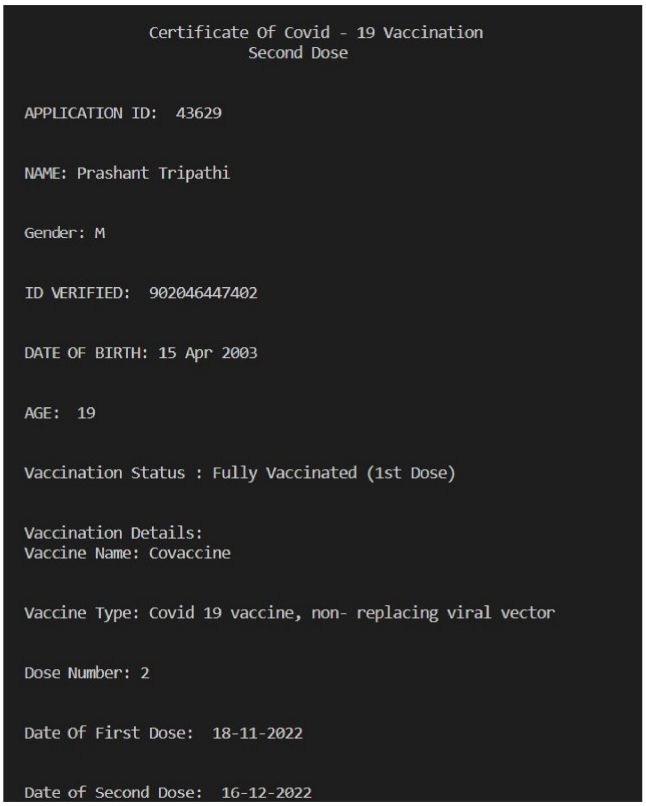
```
 YOUR 1st DOSE APPOINTMENT IS BOOKED

Below are the Following Detalis

APPLICATION ID:  43629


NAME: Prashant Tripathi


Gender: M


AADHAR NUMBER:  902046447402


DATE OF BIRTH: 15 Apr 2003


AGE:  19


VACCINATION VENUE: Rajwada Palace, Sarafa Street, Martand Chowk, Indore, Indore District, Madhya Pradesh, 452001, India


Time: 11AM - 3PM


First Dose Date:  18-11-2022


Vaccine Name: Covaccine
```

```
                                    GUIDLINES FOR VACCINATION
Make sure you have:

        1. A Mask that covers your nose and mouth and fits tightly and comfortably.
        2. Hand Sanitizer.
        3. The Notification you received about your appointment.
        4. Your ID


Your 1st Dose Vaccination is Done!!!!

You can apply for Second Dose after  16-12-2022
For 1st Dose Certificate click [ y or n]

y
Enter application ID

43629


                        Certificate Of Covid - 19 Vaccination
                                    First Dose


APPLICATION ID:  43629


NAME: Prashant Tripathi


Gender: M


ID VERIFIED:  902046447402


DATE OF BIRTH: 15 Apr 2003
```

```
AGE:  19


Vaccination Status : Half Vaccinated (1st Dose)


Vaccination Details:
Vaccine Name: Covaccine


Vaccine Type: Covid 19 vaccine, non- replacing viral vector


Dose Number: 1


Dose Date:  18-11-2022


Apply For Second Dose After 2nd Dose Date:  16-12-2022


Vaccination At : Rajwada Palace, Sarafa Street, Martand Chowk, Indore, Indore District, Madhya Pradesh, 452001, India
Do You Want To Check Second Dose Staus:
y
Enter the Application id provided during 1st dose
43629
You Can Apply On or After:
 16-12-2022
Total Days left for Your Second Dose are 31 days, 0:00:00
Do you want to apply for Second dose [y/n]

y
```

```
 YOUR 2nd DOSE APPOINTMENT IS BOOKED

Below are the Following Detalis

APPLICATION ID:  43629


NAME: Prashant Tripathi


Gender: M


AADHAR NUMBER:  902046447402


DATE OF BIRTH: 15 Apr 2003


AGE:  19


VACCINATION VENUE: Rajwada Palace, Sarafa Street, Martand Chowk, Indore, Indore District, Madhya Pradesh, 452001, India


Time: 11AM - 3PM


Second Dose Date:  15-12-2022


Vaccine Name: Covaccine
```

```
                                GUIDLINES FOR VACCINATION
Make sure you have:

        1. A Mask that covers your nose and mouth and fits tightly and comfortably.
        2. Hand Sanitizer.
        3. The Notification you received about your appointment.
        4. Your ID
To Download certifacte of your Second Dose click [y /n]

y
```



Certificate Of Covid - 19 Vaccination
Second Dose

APPLICATION ID: 43629

NAME: Prashant Tripathi

Gender: M

ID VERIFIED: 902046447402

DATE OF BIRTH: 15 Apr 2003

AGE: 19

Vaccination Status : Fully Vaccinated (1st Dose)

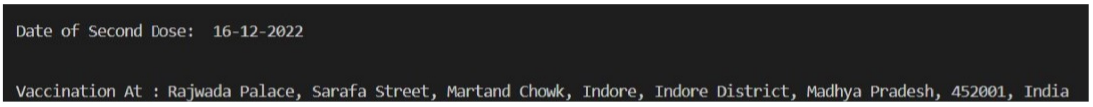
Vaccination Details:
Vaccine Name: Covaccine

Vaccine Type: Covid 19 vaccine, non- replacing viral vector

Dose Number: 2

Date Of First Dose: 18-11-2022

Date of Second Dose: 16-12-2022



Date of Second Dose: 16-12-2022

Vaccination At : Rajwada Palace, Sarafa Street, Martand Chowk, Indore, Indore District, Madhya Pradesh, 452001, India

PROFILING

```
      7488 function calls (7380 primitive calls) in 67.190 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:100(acquire)
   3/1    0.000    0.000    0.004    0.004 <frozen importlib._bootstrap>:1022(_find_and_load)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:125(release)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:165(__init__)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:169(__enter__)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:173(__exit__)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:179(_get_module_lock)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:198(cb)
   4/1    0.000    0.000    0.003    0.003 <frozen importlib._bootstrap>:233(_call_with_frames_removed)
    32    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:244(_verbose_message)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:357(__init__)
     5    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:391(cached)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:404(parent)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:412(has_location)
     2    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:48(_new_module)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:492(_init_module_attrs)
     3    0.000    0.000    0.001    0.000 <frozen importlib._bootstrap>:564(module_from_spec)
   3/1    0.000    0.000    0.004    0.004 <frozen importlib._bootstrap>:664(_load_unlocked)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:71(__init__)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:746(find_spec)
     3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:826(find_spec)
    12    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:893(__enter__)
    12    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap>:897(__exit__)
     3    0.000    0.000    0.001    0.000 <frozen importlib._bootstrap>:921(_find_spec)
   3/1    0.000    0.000    0.004    0.004 <frozen importlib._bootstrap>:987(_find_and_load_unlocked)
     2    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:1040(__init__)
     2    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:1065(get_filename)
     2    0.000    0.000    0.001    0.001 <frozen importlib._bootstrap_external>:1070(get_data)
     2    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:1089(path_stats)
     1    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:1163(__init__)
     1    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:1174(create_module)
     1    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:1182(exec_module)
    27    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:119(<listcomp>)
     4    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:132(_path_split)
    12    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:134(<genexpr>)
     8    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:1356(_path_importer_cache)
     3    0.000    0.000    0.001    0.000 <frozen importlib._bootstrap_external>:1399(_get_spec)
    11    0.000    0.000    0.001    0.000 <frozen importlib._bootstrap_external>:140(_path_stat)
     3    0.000    0.000    0.001    0.000 <frozen importlib._bootstrap_external>:1431(find_spec)
```

```
  11    0.000    0.000    0.001    0.000 <frozen importlib._bootstrap_external>:140(_path_stat)
   3    0.000    0.000    0.001    0.000 <frozen importlib._bootstrap_external>:1431(find_spec)
   3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:150(_path_is_mode_type)
   3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:1531(_get_spec)
   6    0.000    0.000    0.001    0.000 <frozen importlib._bootstrap_external>:1536(find_spec)
   3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:159(_path_isfile)
   3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:172(_path_isabs)
   4    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:380(cache_from_source)
   3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:510(_get_cached)
   2    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:542(_check_name_wrapper)
   2    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:585(_classify_pyc)
   2    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:618(_validate_timestamp_pyc)
   6    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:67(_relax_case)
   2    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:670(_compile_bytecode)
   3    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:721(spec_from_file_location)
   6    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:84(_unpack_uint32)
   2    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:874(create_module)
 2/1    0.000    0.000    0.004    0.004 <frozen importlib._bootstrap_external>:877(exec_module)
   2    0.000    0.000    0.002    0.001 <frozen importlib._bootstrap_external>:950(get_code)
  27    0.000    0.000    0.000    0.000 <frozen importlib._bootstrap_external>:96(_path_join)
   6    0.000    0.000    0.000    0.000 <string>:1(<lambda>)
   1    0.000    0.000   67.190   67.190 <string>:1(<module>)
   1    0.000    0.000    0.000    0.000 __init__.py:1307(disable)
   6    0.000    0.000    0.000    0.000 __init__.py:1455(debug)
   1    0.000    0.000    0.000    0.000 __init__.py:1710(getEffectiveLevel)
   6    0.000    0.000    0.000    0.000 __init__.py:1724(isEnabledFor)
   1    0.000    0.000    0.000    0.000 __init__.py:219(_acquireLock)
   1    0.000    0.000    0.000    0.000 __init__.py:228(_releaseLock)
   6    0.000    0.000    0.000    0.000 __init__.py:299(loads)
   1    0.000    0.000    0.000    0.000 __init__.py:43(normalize_encoding)
   1    0.000    0.000    0.004    0.004 __init__.py:71(search_function)
   3    0.000    0.000    0.000    0.000 __init__.py:89(find_spec)
   2    0.000    0.000    0.000    0.000 __init__.py:96(<lambda>)
   8    0.000    0.000    0.000    0.000 _collections_abc.py:315(__subclasshook__)
  24    0.000    0.000    0.000    0.000 _collections_abc.py:409(__subclasshook__)
   6    0.000    0.000    0.000    0.000 _collections_abc.py:816(get)
   6    0.000    0.000    0.000    0.000 _parseaddr.py:201(quote)
  42    0.000    0.000    0.000    0.000 _policybase.py:281(_sanitize_header)
  42    0.000    0.000    0.000    0.000 _policybase.py:293(header_source_parse)
  42    0.000    0.000    0.000    0.000 _policybase.py:311(header_fetch_parse)
  21    0.000    0.000    0.000    0.000 abc.py:117(__instancecheck__)
32/8    0.000    0.000    0.000    0.000 abc.py:121(__subclasscheck__)
   6    0.000    0.000    5.881    0.980 adapters.py:285(get_json)
   6    0.000    0.000    5.881    0.980 adapters.py:294(get_text)
```

```
   6    0.000    0.000    0.000    0.000 ssl.py:1113(_check_connected)
   6    0.000    0.000    1.867    0.311 ssl.py:1121(read)
   6    0.000    0.000    0.000    0.000 ssl.py:1199(send)
   6    0.000    0.000    0.000    0.000 ssl.py:1226(sendall)
   6    0.000    0.000    1.868    0.311 ssl.py:1263(recv_into)
   6    0.001    0.000    0.002    0.000 ssl.py:1331(_real_close)
   6    0.000    0.000    1.899    0.317 ssl.py:1335(do_handshake)
   6    0.000    0.000    0.002    0.000 ssl.py:488(__new__)
   6    0.000    0.000    0.000    0.000 ssl.py:499(_encode_hostname)
   6    0.000    0.000    1.901    0.317 ssl.py:507(wrap_socket)
   6    0.000    0.000    0.000    0.000 ssl.py:562(set_alpn_protocols)
  12    0.000    0.000    0.054    0.004 ssl.py:573(_load_windows_store_certs)
   6    0.000    0.000    0.054    0.009 ssl.py:587(load_default_certs)
  12    0.000    0.000    0.000    0.000 ssl.py:728(verify_mode)
   6    0.000    0.000    0.000    0.000 ssl.py:736(verify_mode)
   6    0.000    0.000    0.057    0.009 ssl.py:741(create_default_context)
   1    0.000    0.000    0.001    0.001 stringprep.py:1(<module>)
   2    0.000    0.000    0.000    0.000 types.py:176(__get__)
  12    0.000    0.000    0.000    0.000 utils.py:222(unquote)
   6    0.000    0.000    0.000    0.000 utils.py:260(decode_params)
  42    0.000    0.000    0.000    0.000 utils.py:51(_has_surrogates)
```

DEBUGGING

```
Your OTP for Login in vaccination is  13225
> c:\users\hp\desktop\finalll.py(34)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(37)main()
-> def Get_OTP():
(Pdb) n
> c:\users\hp\desktop\finalll.py(45)main()
-> Get_OTP()
(Pdb) n
 OTP Sent on The Given Phone Number
        Verify To Continue
Enter the OTP
13225
OTP verified!! Let's Continue
> c:\users\hp\desktop\finalll.py(47)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(50)main()
-> def second_dose_certificate(second_dose_date):
(Pdb) n
> c:\users\hp\desktop\finalll.py(82)main()
-> '''
(Pdb) p phone_no
1
(Pdb) n
> c:\users\hp\desktop\finalll.py(85)main()
-> def second_dose_status():
(Pdb) n
> c:\users\hp\desktop\finalll.py(92)main()
-> '''
(Pdb) b 67
Breakpoint 1 at c:\users\hp\desktop\finalll.py:67
(Pdb) p presentday
*** NameError: name 'presentday' is not defined
(Pdb) n
> c:\users\hp\desktop\finalll.py(97)main()
-> def Verification_for_second_dose():
(Pdb) step
```

```
(Pdb) b 101
Breakpoint 2 at c:\users\hp\desktop\finalll.py:101
(Pdb) n
> c:\users\hp\desktop\finalll.py(112)main()
-> def schedule_second_dose(vaccine,first_dose_date):
(Pdb) n
> c:\users\hp\desktop\finalll.py(123)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(126)main()
-> def first_dose_certificate(second_dose_date,location_for_appointment):
(Pdb) n
> c:\users\hp\desktop\finalll.py(168)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(172)main()
-> def guidlines_for_vaccination():
(Pdb) n
> c:\users\hp\desktop\finalll.py(180)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(184)main()
-> def second_dose_message(location_for_appointment,First_Name,Last_Name,dob,age,Aadhar_no,second_dose_date,vaccine,Application_id,gender)
(Pdb) n
> c:\users\hp\desktop\finalll.py(216)main()
-> def first_dose_message(location_for_appointment,First_Name,Last_Name,dob,age,Aadhar_no,first_dose_date,vaccine,Application_id,gender):
(Pdb) n
> c:\users\hp\desktop\finalll.py(246)main()
-> '''
(Pdb)
```

```
> c:\users\hp\desktop\finalll.py(246)main()
-> '''
(Pdb) b 165
*** Blank or comment
(Pdb) n
> c:\users\hp\desktop\finalll.py(249)main()
-> def Set_Vaccination_Venue():
(Pdb) n
> c:\users\hp\desktop\finalll.py(273)main()
-> def Schedule_Appointment():
(Pdb) n
> c:\users\hp\desktop\finalll.py(285)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(288)main()
-> def get_vaccine_name():
(Pdb) n
> c:\users\hp\desktop\finalll.py(295)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(299)main()
-> def Apply_Vaccination():
(Pdb) n
> c:\users\hp\desktop\finalll.py(324)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(327)main()
-> def ask_vaccination():
(Pdb) n
> c:\users\hp\desktop\finalll.py(352)main()
-> ask_vaccination()
(Pdb) n
DO you want to Apply For 1st Dose Vaccination [Yes/No]
n
Do you want to apply for second dose [Yes/No]

n
Stay Healthy, Stay Safe...
```

```
-> print("Your OTP for Login in vaccination is ", OTP_generate)
(Pdb) p phone_no
9131618018
(Pdb) n
Your OTP for Login in vaccination is  88241
> c:\users\hp\desktop\finalll.py(25)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(28)main()
-> def Verify_OTP( otp):
(Pdb) b 78
Breakpoint 1 at c:\users\hp\desktop\finalll.py:78
(Pdb) p Application_id
42388
(Pdb) n
> c:\users\hp\desktop\finalll.py(34)main()
-> pdb.set_trace()
(Pdb) n
> c:\users\hp\desktop\finalll.py(35)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(38)main()
-> def Get_OTP():
(Pdb) n
> c:\users\hp\desktop\finalll.py(46)main()
-> Get_OTP()
(Pdb) n
 OTP Sent on The Given Phone Number
        Verify To Continue
Enter the OTP
56
INCORRECT OTP, Please click on Resend OTP To Try Again
 OTP Sent on The Given Phone Number
        Verify To Continue
Enter the OTP
56
INCORRECT OTP, Please click on Resend OTP To Try Again
 OTP Sent on The Given Phone Number
```

```
-> def guidlines_for_vaccination():
(Pdb) p otp
30622
(Pdb) p Application_id
70061
(Pdb) p presentday
*** NameError: name 'presen□day' is not defined
(Pdb) n
> c:\users\hp\desktop\finalll.py(181)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(185)main()
-> def second_dose_message(location_for_appointment,First_Name,Last_Name,dob,age,Aadhar_no,second_dose_date,vaccine,Application_id,gender):
(Pdb) n
> c:\users\hp\desktop\finalll.py(217)main()
-> def first_dose_message(location_for_appointment,First_Name,Last_Name,dob,age,Aadhar_no,first_dose_date,vaccine,Application_id,gender):
(Pdb) n
> c:\users\hp\desktop\finalll.py(247)main()
-> '''
(Pdb) n
> c:\users\hp\desktop\finalll.py(250)main()
-> def Set_Vaccination_Venue():
(Pdb) n
> c:\users\hp\desktop\finalll.py(274)main()
-> def Schedule_Appointment():
(Pdb) n
> c:\users\hp\desktop\finalll.py(286)main()
-> '''
(Pdb) b 250
Breakpoint 1 at c:\users\hp\desktop\finalll.py:250
(Pdb) p presentdat
*** NameError: name 'presen□dat' is not defined
(Pdb) b 330
Breakpoint 2 at c:\users\hp\desktop\finalll.py:330
(Pdb) n
> c:\users\hp\desktop\finalll.py(289)main()
-> def get_vaccine_name():
(Pdb) n
```

```
> c:\users\hp\desktop\finalll.py(362)main()
-> inp=input()
(Pdb) p First_name
*** NameError: name 'First_name' is not defined
(Pdb) p First_Name
'prashant'
(Pdb) p Last_Name
'tripathi'
(Pdb) p dob
'12 apr'
(Pdb) n
n
> c:\users\hp\desktop\finalll.py(363)main()
-> if(inp=="y"):
(Pdb) n
> c:\users\hp\desktop\finalll.py(367)main()
-> print("Do You Want To Check Second Dose Staus:")
(Pdb) n
Do You Want To Check Second Dose Staus:
> c:\users\hp\desktop\finalll.py(368)main()
-> second_inp=input()
(Pdb) n
n
> c:\users\hp\desktop\finalll.py(369)main()
-> if(second_inp=="y"):
(Pdb) n
> c:\users\hp\desktop\finalll.py(372)main()
-> print("Do you want to apply for Second dose [y/n]\n")
(Pdb) p Application_id
9398
```

Statiscal Information

1. Starting Data: 11/11/2022
2. End Date: 15/11/2022
3. Total Line Of Code: 350
4. Total Days Taken: 5 Days
5. Language Used: Python
6. Program Title: Vaccination Drive
7. Degubber Used: pdb
8. Profiler Used: cProfile