**PROJECT REPORT ON**

**VACCINE MANAGEMENT SYSTEM**

**ROLL NO : 12676355**

**CLASS : XII Science**

**SUBJECT : COMPUTER SCIENCE**

**SUB CODE : 083**

**SESSION : 2021-22**

**PROJECT GUIDE: Mrs. Sayani Hazra Pal**

**PGT (CS)**

# download (1).png

# CERTIFICATE

This is to certify that Anindita Chakraborty, Roll No: 12676355 has successfully completed the project work entitled **VACCINE MANAGEMENT SYSTEM** the subject Computer Science (083) laid down in the regulations of CBSE for Class XII.

INTERNAL EXAMINER PRINCIPAL

TERM-1

INTERNAL EXAMINER EXTERNAL EXAMINER

TERM-2

**ACKNOWLEDGEMENT**

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I express deep sense of gratitude to almighty God for giving me strength for the successful completion of the project.

I express my heartfelt gratitude to my parents for constant encouragement while carrying out this project.

I gratefully acknowledge the contribution of the individuals who contributed in bringing this project up to this level, who continues to look after me despite my flaws,

I express my deep sense of gratitude to the luminary The Principal, D.A.V Public School who has been continuously motivating and extending their helping hand to us.

I express my sincere thanks to Mrs. Sayani Hazra Pal, a guide a mentor all the above a friend who critically reviewed my project and helped in solving each and every problem that occurred during the implementation of this project

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. I am grateful for their constant support and help.

Anindita Chakraborty

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS [ T O C ]** | | |
| **SER** | **DESCRIPTION** | **PAGE NO.** |
| 01 | INTRODUCTION | 2 |
| 02 | OBJECTIVES OF THE PROJECT | 3 |
| 03 | SYSTEM DESIGN and FLOW CHART | 04 |
| 04 | SYSTEM REQUIREMENTS | 11 |
| 05 | MODULE USED, DATA FILE USED | 12 |
| 06 | SOURCE CODE AND CONNECTIVITY WITH SQL | 13 |
| 07 | OUTPUT | 20 |
| 08 | SCOPE | 17 |
| 09 | LIMITATIONS | 18 |
| 10 | CONCLUSION | 19 |
| 11 | BIBLIOGRAPHY | 20 |

**INTRODUCTION**

In the dawn of the outbreak of a global pandemic, our administration authorities and the healthcare officials have encountered a situation where vaccines are to be provided to thousands of people every day. In response to this situation, we planned upon developing an application that offers a hand of help to the healthcare officials and administration centers who are working relentlessly for the welfare of our masses. This project is a sustainable approach to the creation of an application that supports planning and effective record management for Covid-19 vaccination clinics and centers so as to make sure that maximum efficiency is generated out of minimal efforts and time is saved.

The program is designed for the purpose of effective data management system with the help of data file handing and Python-SQL connectivity. The program also involves the use of GUI to make the program convenient and intuitive to use.

Finally, every effort has been made to make this program as user friendly as possible, from the menus and submenus which allow users to intuitively find what they want, to the clear instructions and error messages that guide the user.

**OBJECTIVES OF THE PROJECT**

The main objective of our Vaccine Management System named ‘Vaxer’ is to manage mass vaccination, including vaccine distribution and administration. It will help the state and local public health officials manage vaccine distribution at scale and expedite vaccine administration for a large population. The solution is designed to provide real-time access to vaccine administration data to support decision-making, including support for outreach campaigns and distribution efforts.

SYSTEM DESIGN AND FLOWCHART

Then we begin with the design phase of the system.

System Design is a solution, a ‘HOW TO’ approach to

the creation of a new system. It translates system

requirements into ways by which they can be made

operational. It is a translation from a user-oriented

document to a document oriented program. For

that, it provides the understanding of procedural details

necessary for the implementation. Here we use

flowchart to supplement the working for the new system.

The system should be thus made reliable, durable and

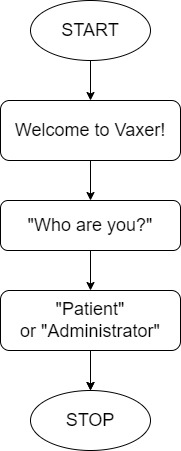
above all should have least possible maintenance cost.

It should overcome all the drawbacks of the old existing

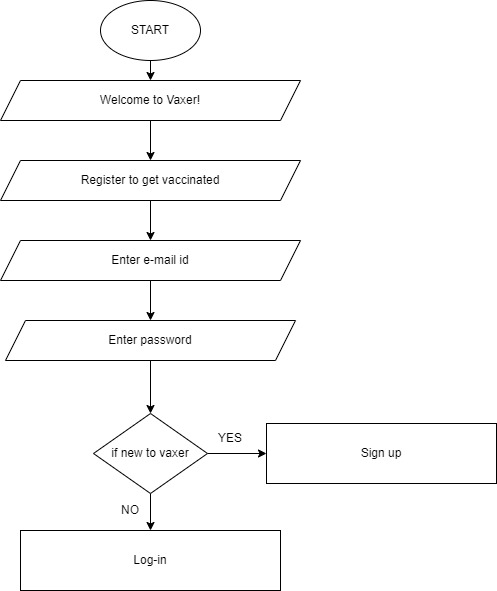
system and most important of all, meet the user

requirements.

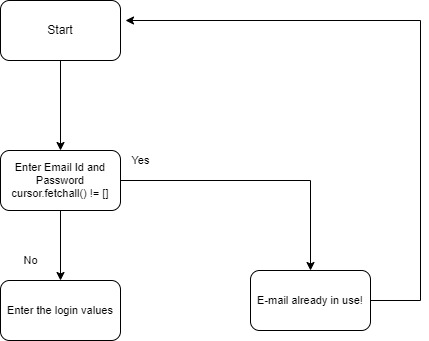
GREET

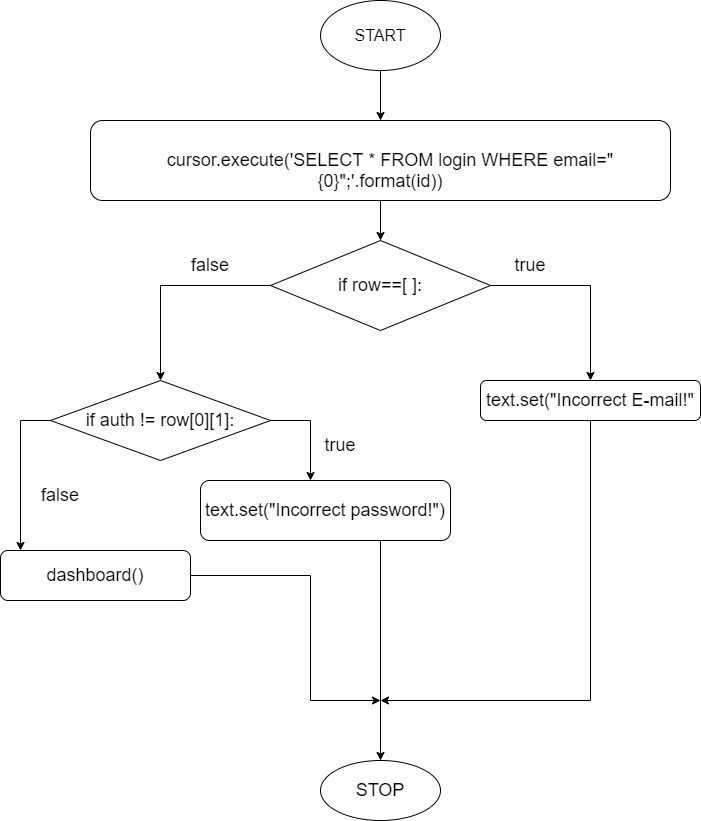


PATIENT

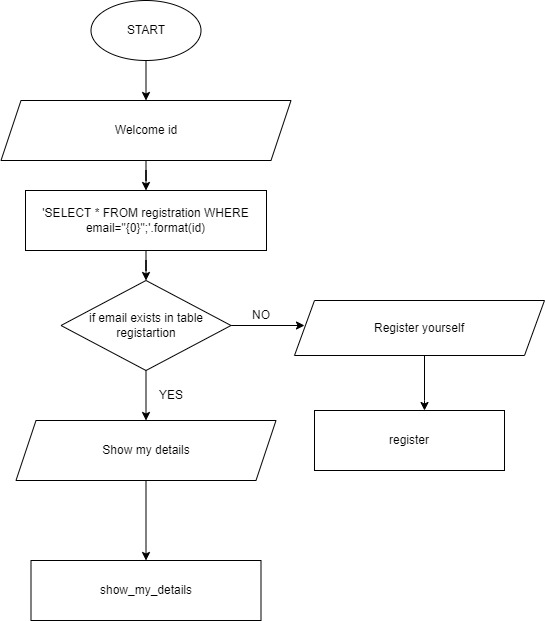


SIGN-UP

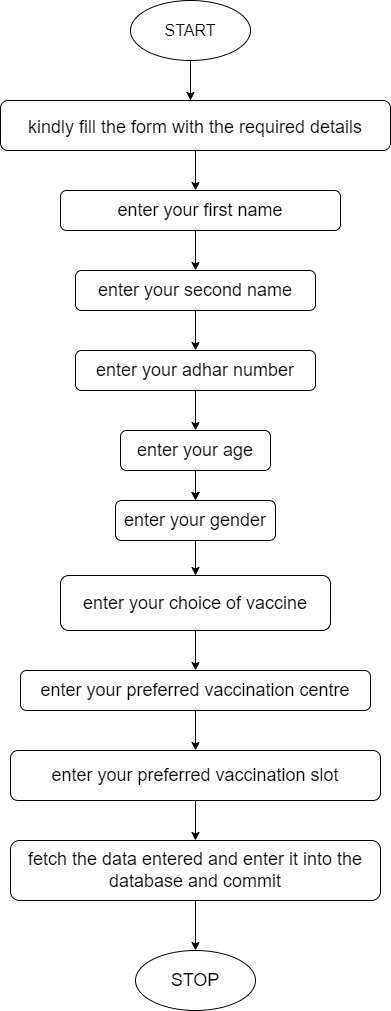
  
LOG-IN



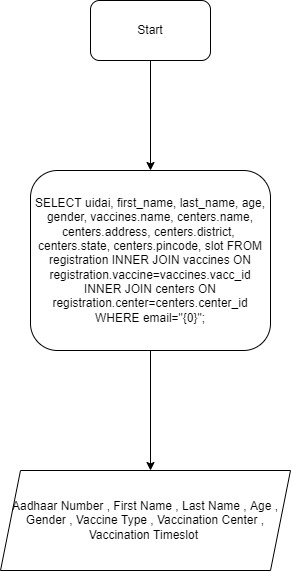
DASHBOARD



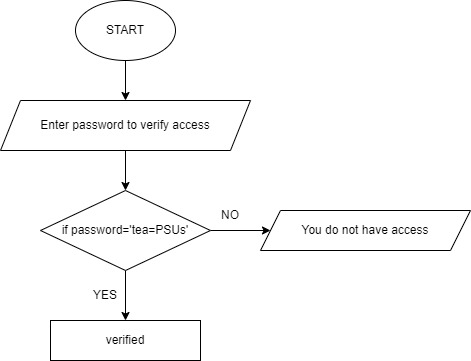
REGISTER



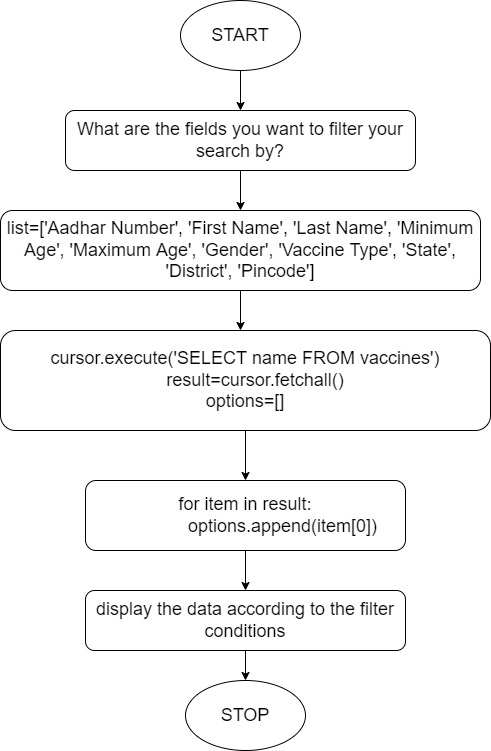
SHOW MY DETAILS



ADMIN



VERIFIED



**HARDWARE REQUIREMENTS**

I. PROCESSOR : x86 CPU 2 GHz+

II. RAM : 4GB+

**SOFTWARE REQUIREMENTS**

I. Operating System- Windows/Linux/Mac OS

II. Python 3.8.x

III. SQL

.

**MODULE USED DATA FILE USED**

**Data File Used:**

● Binary Files

**Modules Used:**

● pickle: to work with binary files

● tkinter: to create Graphical user Interface(GUI)

● sqlconnector: to connect python with sql

**SOURCE CODE**

# %% md

# # Vaxer

# ## Vaccination Management System

# %% md

# First we will import the packages needed

# %% imports

from tkinter import \*

import tkinter.ttk as ttk

import pickle

import mysql.connector as sql

# %% database connector

db=sql.connect(host='localhost', user='root', password='') # password is needed to be put by the user

cursor=db.cursor()

# %% create the database

def createdb():

source=open('dbcreate.sql')

lines=source.readlines()

for line in lines:

cursor.execute(line)

# %% update static data tables

def update():

files=('vaccines.dat', 'centers.dat')

for file in files:

table=open(file, 'rb')

while True:

try:

data=pickle.load(table)

try:

if file=='vaccines.dat':

cursor.execute('INSERT INTO vaccines VALUES ({0}, "{1}", "{2}");'.format(data[0], data[1], 'Y' if data[2]=='Available' else 'N'))

else:

cursor.execute('INSERT INTO centers VALUES ({0}, "{1}", "{2}", "{3}", "{4}", {5});'.format(data[0], data[1], data[2], data[3], data[4], data[5]))

except sql.errors.IntegrityError:

pass

except EOFError:

break

db.commit()

# %% tkinter root

root=Tk()

root.geometry("1540x800")

root.title("Vaxer")

# %% scrollbar

scrollbar=Scrollbar(root).grid(column=10,sticky="NS")

root.configure(bg= "ghost white")

# %% mainframe

mainframe=Frame(root, borderwidth= 25,bg="ghost white")

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

# %% resizability

root.resizable(True, True)

root.columnconfigure(0, weight=1)

root.columnconfigure(0, weight=1)

ttk.Sizegrip(root).grid(row=2, sticky='SE')

# %% clearing a window

def clear(frame):

for widget in frame.winfo\_children():

widget.destroy()

# %% first window

def greet():

clear(mainframe)

Label(mainframe, text="Welcome to Vaxer!", borderwidth=15, bg="ghost white", font= ("Bodoni MT", 28,"bold"), padx=25, pady=15).grid(row=0)

Label(mainframe, text="Who are you?", borderwidth=20,bg="ghost white", font= ("Bodoni MT", 28, "bold"), padx=25, pady=15).grid(row=1)

buttonframe=Frame(mainframe, bg="lavender")

buttonframe.grid(row=2)

Button(buttonframe, text="Patient", command=patient, padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 18,"bold")).grid(column=0, row=0)

Button(buttonframe, text="Administrator", command=admin, padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 18,"bold")).grid(column=1, row=0)

# %% restart button

Button(root, text="Home", command=greet, padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 18,"bold")).grid(row=1)

# %% Patient greet window

def patient():

clear(mainframe)

Label(mainframe, text="Welcome to Vaxer!", borderwidth=15, bg="ghost white",font= ("Bodoni MT", 28,"bold")).grid(row=0)

Label(mainframe, text="Register to get vaccinated", borderwidth=15, bg="ghost white",font= ("Bodoni MT", 28,"bold")).grid(row=0)

fieldframe=Frame(mainframe, bg="lavender")

fieldframe.grid(row=1)

Label(fieldframe, text="Email",bg="lavender",font= ("Bodoni MT", 18), padx=25, pady=15).grid(row=0)

Entry(fieldframe, textvariable=email, bg="ghost white",bd=3, relief=SUNKEN).grid(row=0, column=1)

Label(fieldframe, text="Password",bg="lavender",font= ("Bodoni MT", 18),padx=25, pady=15).grid(row=1)

Entry(fieldframe, textvariable=password, show="\u2022", bg="ghost white",bd=3, relief=SUNKEN).grid(row=1, column=1)

Label(mainframe, text="New to Vaxer?",bg="ghost white",font= ("Bodoni MT", 18, "bold"),padx=25, pady=15).grid(row=2)

Button(mainframe, text="Sign up", command=sign\_up, padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 15)).grid(row=2, column=1)

Label(mainframe, text="Have an account?",bg="ghost white",font= ("Bodoni MT", 18,"bold"),padx=25, pady=15).grid(row=3)

Button(mainframe, text="Log in", command=log\_in, padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 15)).grid(row=3, column=1)

# %% global log-in credentials

email=StringVar()

password=StringVar()

#global user variable

id=''

# %% Sign up System

def sign\_up():

global id

id=email.get()

auth=password.get()

cursor.execute('SELECT \* FROM login WHERE email="{0}";'.format(id))

if cursor.fetchall() != []:

Label(mainframe, text="E-mail already in use!").grid(row=4)

else:

cursor.execute('INSERT INTO login VALUES ("{0}", "{1}");'.format(id, auth))

db.commit()

dashboard()

# %% Log\_in System

def log\_in():

global id

id=email.get()

auth=password.get()

cursor.execute('SELECT \* FROM login WHERE email="{0}";'.format(id))

row=cursor.fetchall()

text=StringVar()

msg=Label(mainframe, textvariable=text,bg="lavender",font= ("Bodoni MT", 14)).grid(row=4)

if row==[]:

text.set("Incorrect E-mail!")

else:

if auth != row[0][1]:

text.set("Incorrect password!")

else:

dashboard()

# %% Dashboard

def dashboard():

clear(mainframe)

Label(mainframe, text='Welcome '+id, bg="ghost white",font= ("Bodoni MT", 28,"bold"),padx=25, pady=15).grid(row=0)

cursor.execute('SELECT \* FROM registration WHERE email="{0}";'.format(id))

result=cursor.fetchall()

if result==[]:

Button(mainframe, text='Register yourself', command=register, padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 18,"bold")).grid(row=1)

else:

Button(mainframe, text='Show my details', command=show\_my\_details, padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 18,"bold")).grid(row=1)

# %% registtration details

def show\_my\_details():

cursor.execute('SELECT uidai, first\_name, last\_name, age, gender, vaccines.name, centers.name, centers.address, centers.district, centers.state, centers.pincode, slot FROM registration INNER JOIN vaccines ON registration.vaccine=vaccines.vacc\_id INNER JOIN centers ON registration.center=centers.center\_id WHERE email="{0}";'.format(id))

result=cursor.fetchall()

clear(mainframe)

list=['Aadhar Number', 'First Name', 'Last Name', 'Age', 'Gender', 'Vaccine Type']

i=0

for item in list:

Lf=LabelFrame(mainframe, text=item,bg="lavender",font= ("Bodoni MT", 16,"bold"),padx=25, pady=15)

L=Label(Lf, text=result[0][i],bg="ghost white",font= ("Bodoni MT", 14), width=25)

Lf.grid(column=0, row=i)

L.grid()

i+=1

Lf=LabelFrame(mainframe, text='Vaccination Center',bg="lavender",font= ("Bodoni MT", 16,"bold"),padx=25, pady=15)

L=Label(Lf, text=result[0][6]+'\n'+result[0][7]+'\n'+result[0][8]+'\n'+result[0][9]+'-'+str(result[0][10]),bg="ghost white",font= ("Bodoni MT", 14),width=25)

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

L.grid()

Lf=LabelFrame(mainframe, text='Vaccination Timeslot',bg="lavender",font= ("Bodoni MT", 16,"bold"),padx=25, pady=15)

r=int(result[0][11])-1

L=Label(Lf, text='{0}:00-{1}:00'.format(9+2\*r,11+2\*r),bg="ghost white",font= ("Bodoni MT", 14),width=25)

Lf.grid(column=1, row=1)

L.grid()

# %% Register

def register():

clear(mainframe)

Label(mainframe,bg="lavender",font= ("Bodoni MT", 16),padx=25, pady=15, text=' Welcome {0}, kindly fill the form with the required details'.format(id)).grid(row=0)

fieldframe=Frame(mainframe,bg="lavender")

fieldframe.grid(row=1)

list=['first name', 'last name', 'Aadhar number', 'age', 'gender', 'choice of vaccine', 'preferred vaccination center', 'preferred vaccination slot']

i=0

for item in list:

Label(fieldframe,bg="lavender",font= ("Bodoni MT", 16),padx=25, pady=15, text='Enter your '+item).grid(row=i, column=0)

i+=1

Entry(fieldframe, textvariable=f\_name,bg="ghost white",bd=3, relief=SUNKEN).grid(row=0, column= 1 )

Entry(fieldframe, textvariable=l\_name,bg="ghost white",bd=3, relief=SUNKEN).grid(row=1, column= 1 )

Entry(fieldframe, textvariable=uidai,bg="ghost white",bd=3, relief=SUNKEN).grid(row=2, column= 1 )

Scale(fieldframe, from\_=0, to=100, variable=age, orient=HORIZONTAL,bg="ghost white").grid(row=3, column=1)

Radiobutton(fieldframe, text="Male", variable=gender, value='M',bg="ghost white").grid(row=4,column=1)

Radiobutton(fieldframe, text="Female", variable=gender, value='F',bg="ghost white").grid(row=4, column=2)

cursor.execute('SELECT name FROM vaccines WHERE status="Y"')

result=cursor.fetchall()

options=[]

vaccine.set("Choose the vaccine")

for item in result:

options.append(item[0])

OptionMenu(fieldframe, vaccine, \*options).grid(row=5, column= 1)

cursor.execute('SELECT name, address, pincode FROM centers')

result=cursor.fetchall()

options=[]

center.set("Choose the center")

for item in result:

options.append(item[0]+'\n'+item[1]+'\n'+str(item[2]))

OptionMenu(fieldframe, center, \*options).grid(row=6, column= 1)

list=[("9:00AM - 11:00AM", '1'), ("11:00AM - 1:00PM", '2'), ("1:00PM - 3:00PM", '3'), ("3:00PM - 5:00PM", '4')]

i=7

for (item, value) in list:

Radiobutton(fieldframe, text=item, variable=slot, value=value).grid(row=i, column= 1 )

i+=1

Button(fieldframe, text='Continue', command=fetch, padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 16)).grid(row=11, column=1)

# %% global variables for registration

uidai=StringVar()

f\_name= StringVar()

l\_name=StringVar()

age= IntVar()

gender= StringVar()

vaccine= StringVar()

center= StringVar()

slot = StringVar()

district=StringVar()

state=StringVar()

age\_max=IntVar()

age\_min=IntVar()

pincode=StringVar()

center\_name=StringVar()

# %% registration processing

def fetch():

cursor.execute('SELECT vacc\_id FROM vaccines WHERE name="{0}"'.format(vaccine.get()))

vacc=cursor.fetchall()[0][0]

cursor.execute('SELECT center\_id FROM centers WHERE CONCAT\_WS(CHAR(10 USING UTF8), name, address, pincode)="{0}"'.format(center.get()))

cent=cursor.fetchall()[0][0]

cursor.execute('INSERT INTO registration VALUES ({0}, "{1}", "{2}", {3}, "{4}", {5}, {6}, "{7}", "{8}")'.format(uidai.get(), f\_name.get(), l\_name.get(), age.get(), gender.get(), vacc, cent, slot.get(), id))

db.commit()

dashboard()

# %% Admin access

def admin():

clear(mainframe)

Label(mainframe, text='Enter password to verify access',bg="lavender",font= ("Bodoni MT", 18), padx=25, pady=15).grid(row=0)

Entry(mainframe, textvariable=password, show="\u2022",bg="ghost white",bd=3, relief=SUNKEN).grid(row=1)

Button(mainframe, text="Verify", command=verify,padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 18, "bold")).grid(row=2)

# %% verify

def verify():

key='tea=PSUs'

if password.get()!=key:

Label(mainframe, text='You do not have access',bg="lavender",font= ("Bodoni MT", 18), padx=25, pady=15).grid(row=3)

else:

verified()

# %% verified

def verified():

clear(mainframe)

Label(mainframe, bg="lavender",font= ("Bodoni MT", 16),padx=25, pady=15, text="What are the fields you want to filter your search by?").grid(row=0)

fieldframe=Frame(mainframe,bg="ghost white",width=25)

fieldframe.grid(row=1)

list=['Aadhar Number', 'First Name', 'Last Name', 'Minimum Age', 'Maximum Age', 'Gender', 'Vaccine Type', 'Center Code', 'Center Name', 'State', 'District', 'Pincode']

i=0

for item in list:

Label(fieldframe, text=item,bg="lavender").grid(row=i, column=0)

i+=1

a=[uidai, f\_name, l\_name, district, state, vaccine, gender, center, center\_name, pincode]

for item in a:

item.set('any')

age\_min.set(0)

age\_max.set(100)

Entry(fieldframe, textvariable=uidai, bg="ghost white", bd=3, relief=SUNKEN).grid(column=1, row=0)

Entry(fieldframe, textvariable=f\_name, bg="ghost white", bd=3, relief=SUNKEN).grid(column=1, row=1)

Entry(fieldframe, textvariable=l\_name, bg="ghost white", bd=3, relief=SUNKEN).grid(column=1, row=2)

Scale(fieldframe, from\_=0, to=100, variable=age\_min, orient=HORIZONTAL,bg="ghost white").grid(row=3, column=1)

Scale(fieldframe, from\_=0, to=100, variable=age\_max, orient=HORIZONTAL,bg="ghost white").grid(row=4, column=1)

Radiobutton(fieldframe, text="Male", variable=gender, value='M',bg="ghost white").grid(row=5,column=1)

Radiobutton(fieldframe, text="Female", variable=gender, value='F',bg="ghost white").grid(row=5, column=2)

cursor.execute('SELECT name FROM vaccines')

result=cursor.fetchall()

options=[]

for item in result:

options.append(item[0])

OptionMenu(fieldframe, vaccine, \*options).grid(row=6, column= 1)

Entry(fieldframe, textvariable=center, bg="ghost white", bd=3, relief=SUNKEN).grid(column=1, row=7)

Entry(fieldframe, textvariable=center\_name, bg="ghost white", bd=3, relief=SUNKEN).grid(column=1, row=8)

states=('West Bengal', 'Jharkhand', 'Odisha', 'Bihar') #fill below

OptionMenu(fieldframe, state, \*states).grid(row=9, column= 1)

Entry(fieldframe, textvariable=district, bg="ghost white", bd=3, relief=SUNKEN).grid(column=1, row=10)

Entry(fieldframe, textvariable=pincode, bg="ghost white", bd=3, relief=SUNKEN).grid(column=1, row=11)

Button(mainframe, text='Filter', command=display, padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 18, "bold")).grid(row=2)

# %% display the records

def display():

clear(mainframe)

columns=('Aadhar No.', 'First Name', 'Last Name', 'Age', 'Gender', 'Vaccine', 'Center Code', 'Center Name', 'State', 'District', 'Pincode')

table=ttk.Treeview(mainframe, columns=columns, show='headings')

for item in columns:

table.heading(item, text=item)

table.column(item, stretch=True, width=140)

table.grid(row=0, column=0, sticky='nsew')

cursor.execute('TRUNCATE TABLE records;')

cursor.execute('INSERT INTO records SELECT uidai, first\_name, last\_name, age, gender, vaccines.name, centers.center\_id, centers.name, centers.state, centers.district, centers.pincode FROM registration INNER JOIN vaccines ON registration.vaccine=vaccines.vacc\_id INNER JOIN centers ON registration.center=centers.center\_id WHERE age BETWEEN {0} AND {1};'.format(age\_min.get(), age\_max.get()))

a=[uidai, f\_name, l\_name, district, state, vaccine, gender, center, center\_name, pincode]

c\_name=['uidai', 'first\_name', 'last\_name', 'gender', 'vaccine\_name', 'center\_id', 'center\_name', 'center\_state', 'center\_district', 'center\_pincode']

for item in a:

k=item.get()

if k != 'any':

cursor.execute('DELETE FROM records WHERE NOT {0}="{1}";'.format(c\_name[a.index(item)], k))

cursor.execute('SELECT \* FROM records;')

result=cursor.fetchall()

for row in result:

table.insert('', END, values=row)

yscroll=Scrollbar(mainframe, orient=VERTICAL, command=table.yview)

table.configure(yscroll=yscroll.set)

yscroll.grid(row=0, column=1, sticky='ns')

Button(mainframe, text='Back', command=verified, padx=30, pady=5, bg="lavender",bd=3, relief=RAISED,font= ("Bodoni MT", 18, "bold")).grid(row=2, column=0)

createdb()

update()

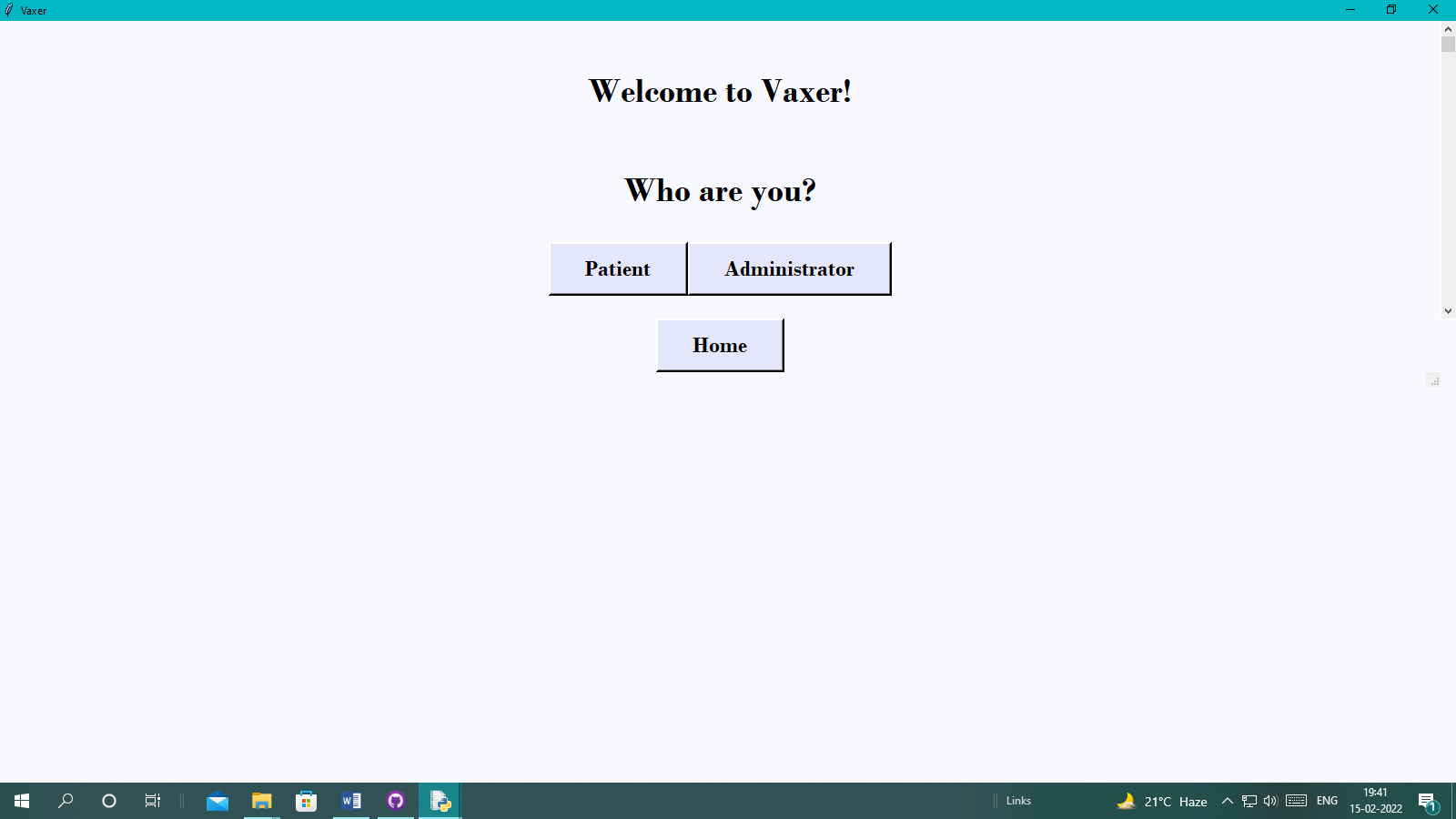
greet()

mainloop()

db.close()

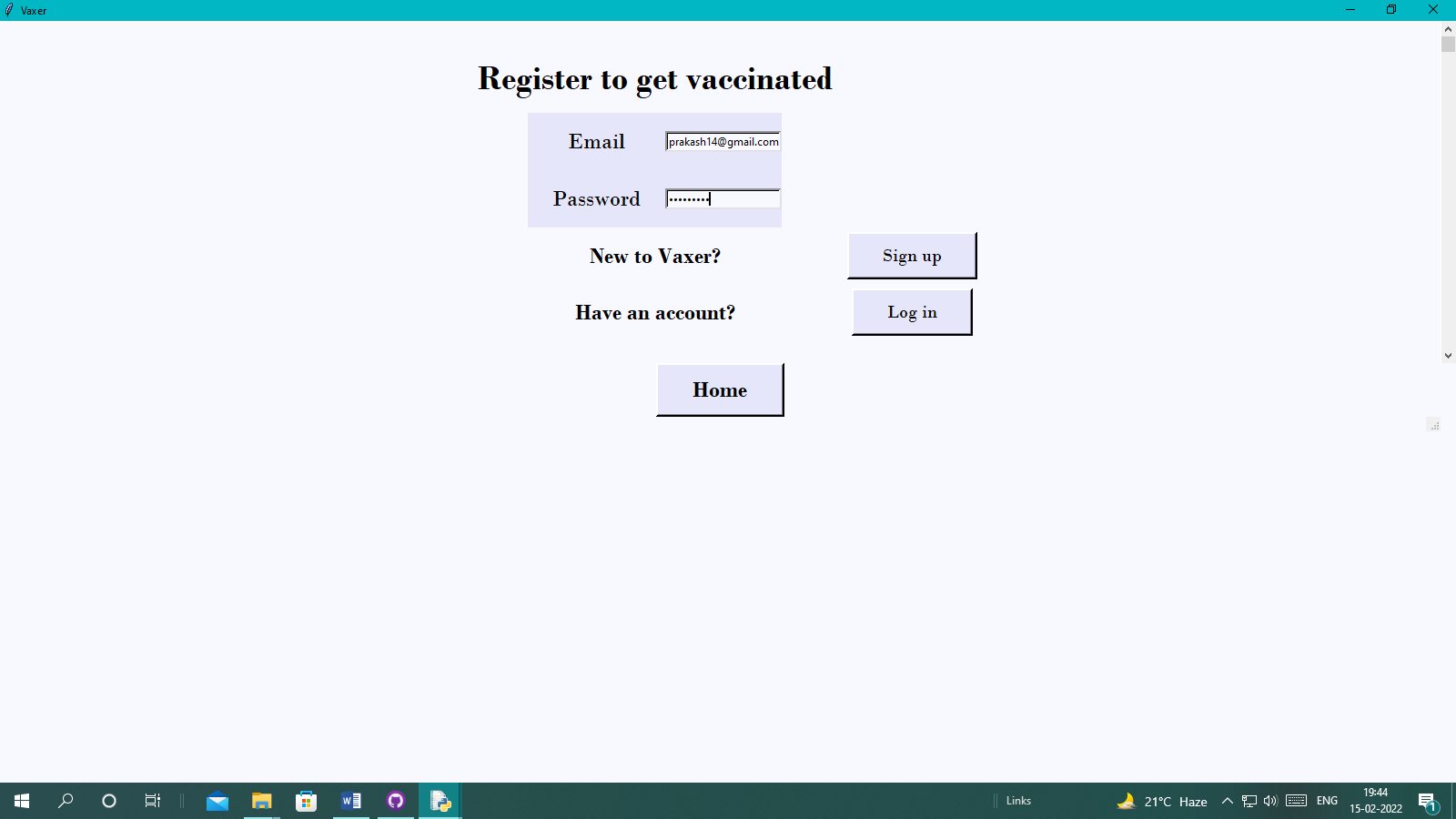
**OUTPUT**

**Greeting window**

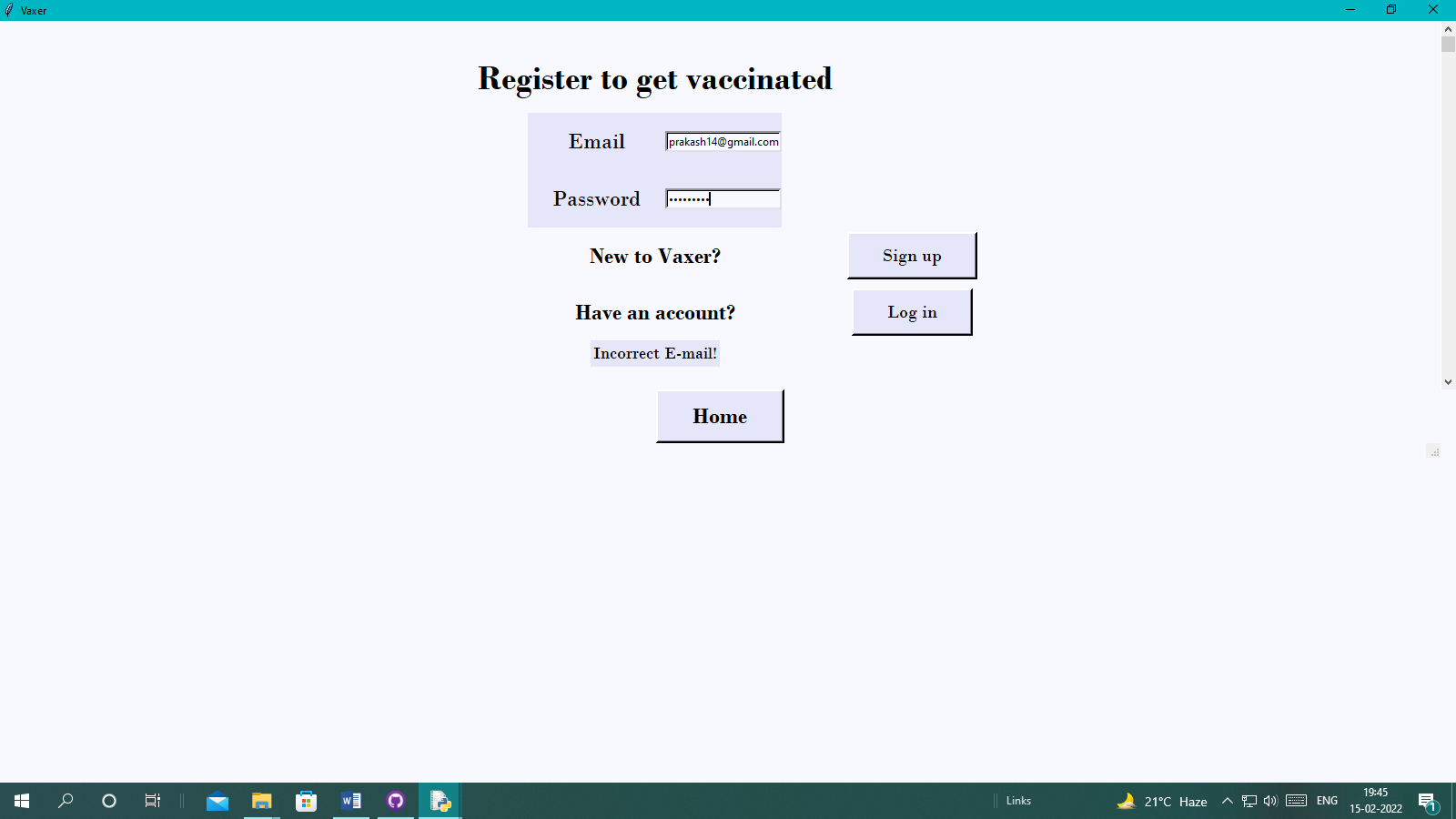


**Patient window**

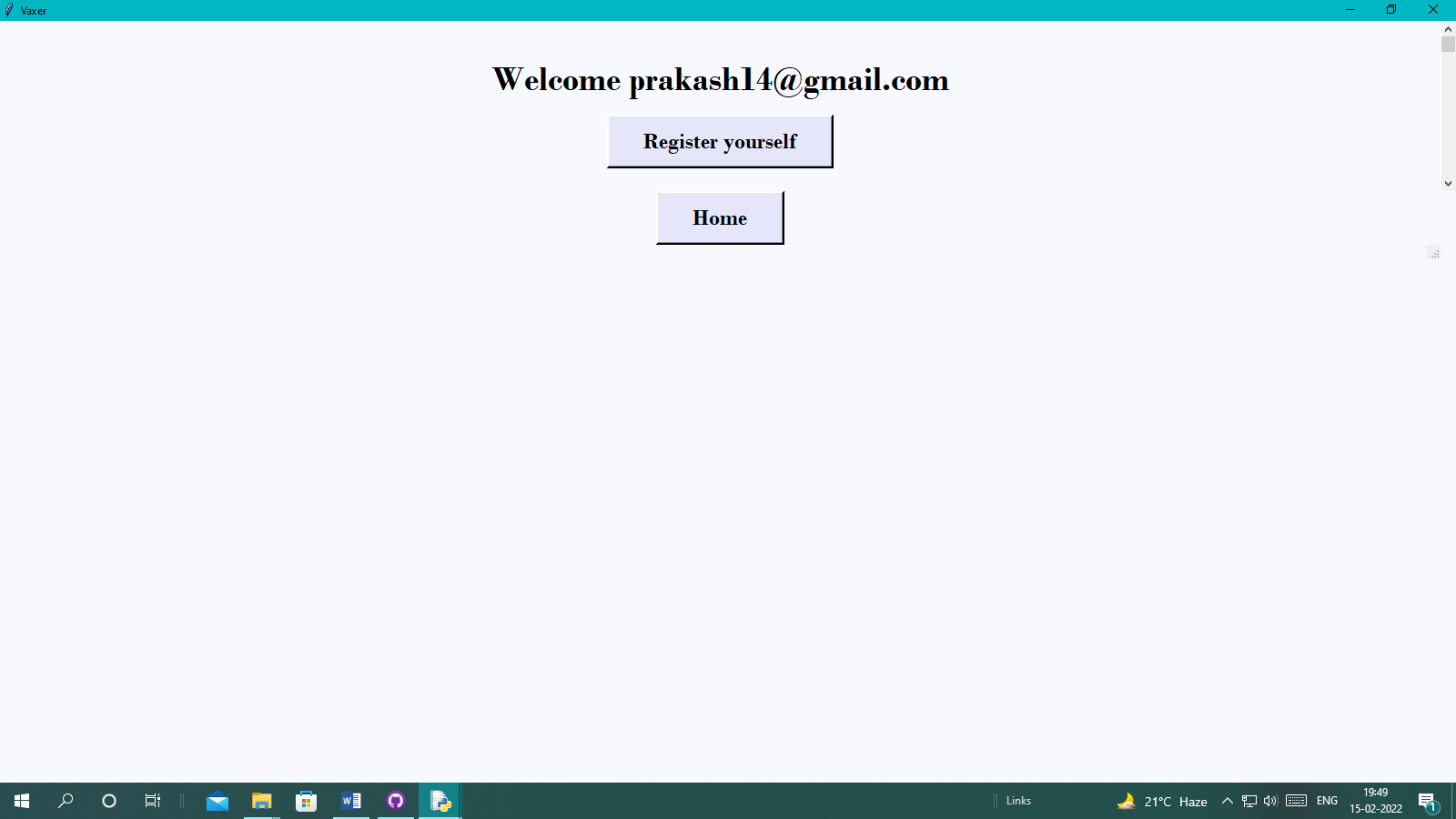
**Case 1: user does not have an account created**



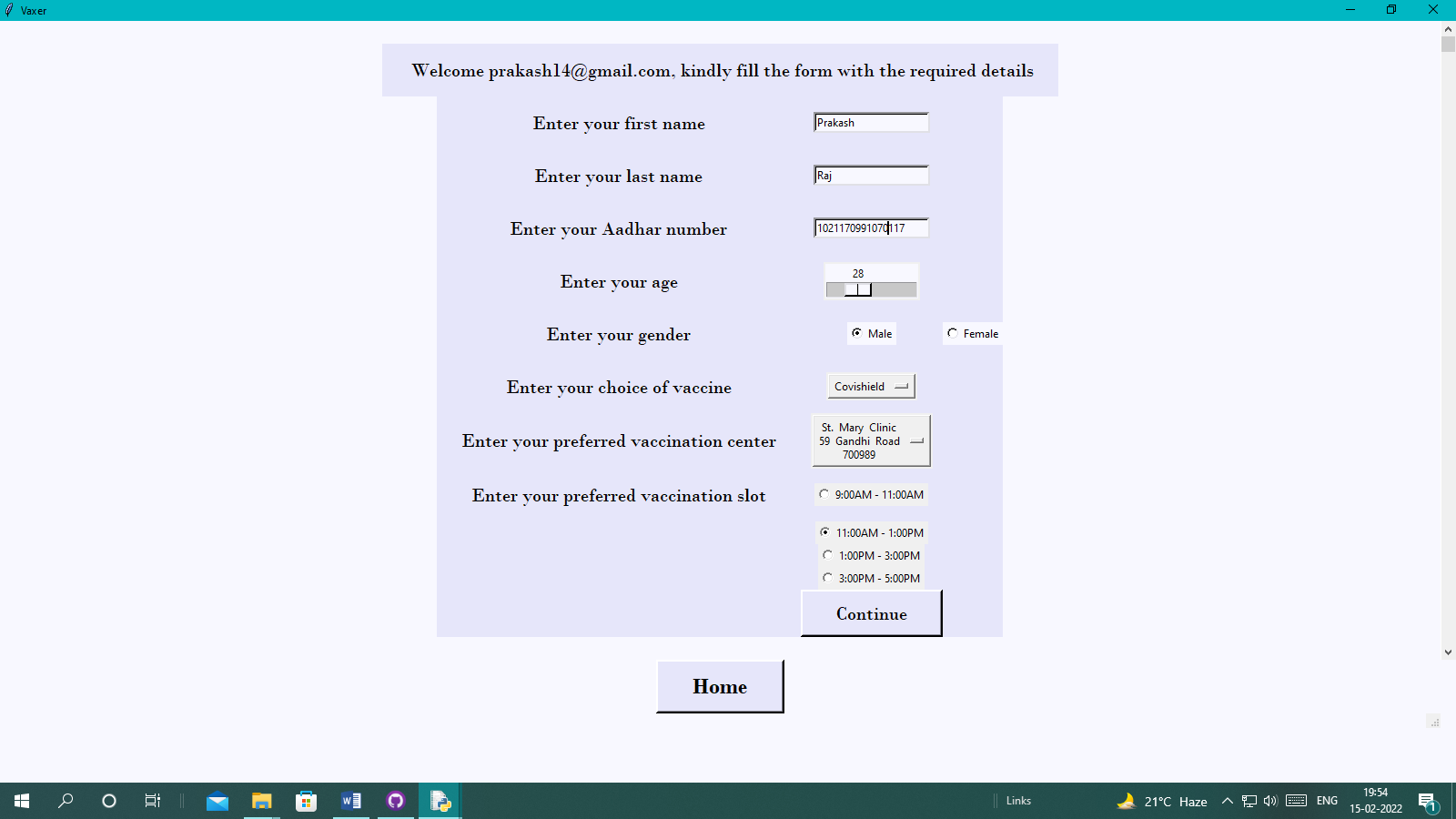
**If user clicks on login, error message pops up since account does not exist**



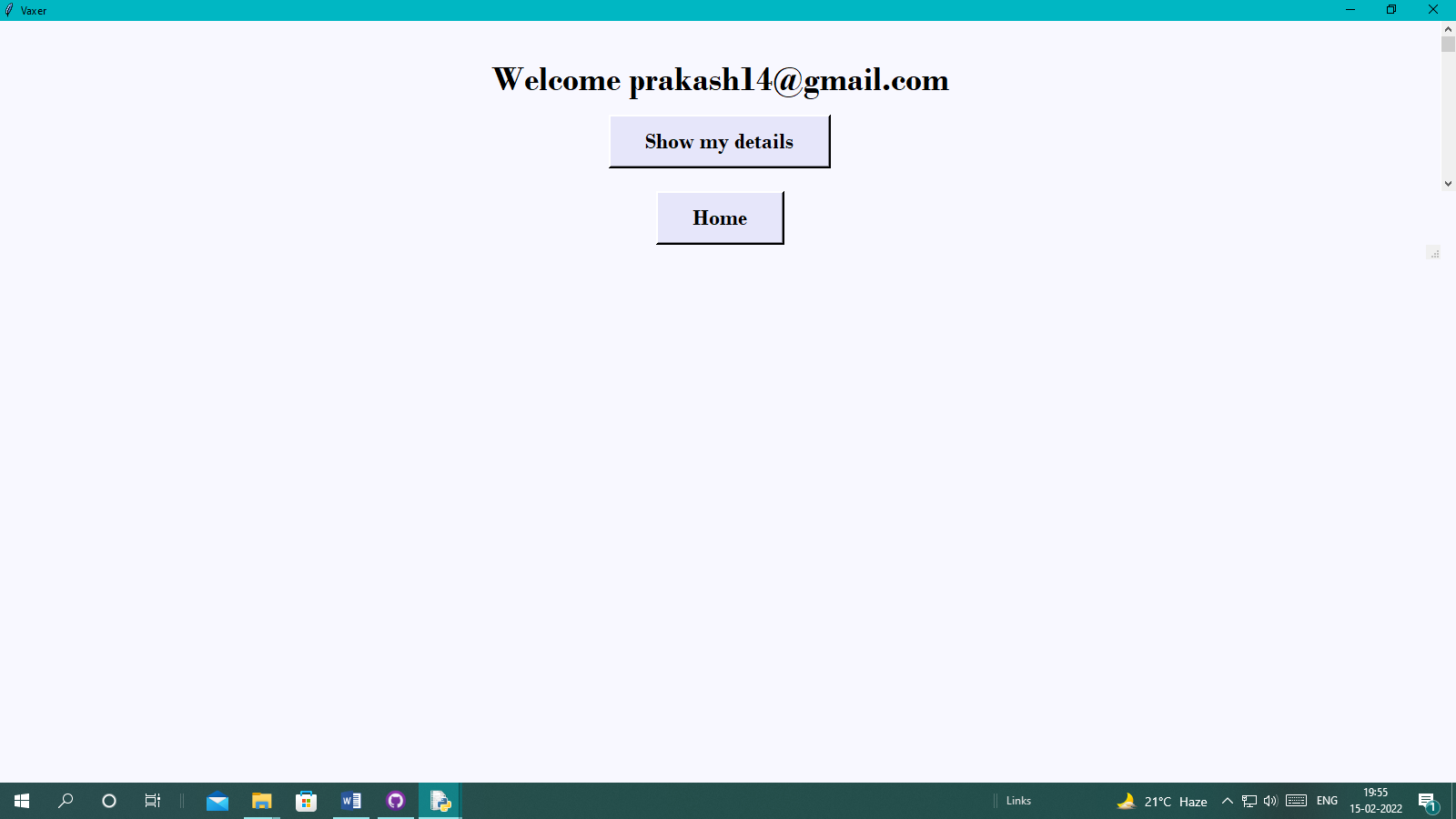
**If user clicks on sign-up, the user is directed to the dashboard**



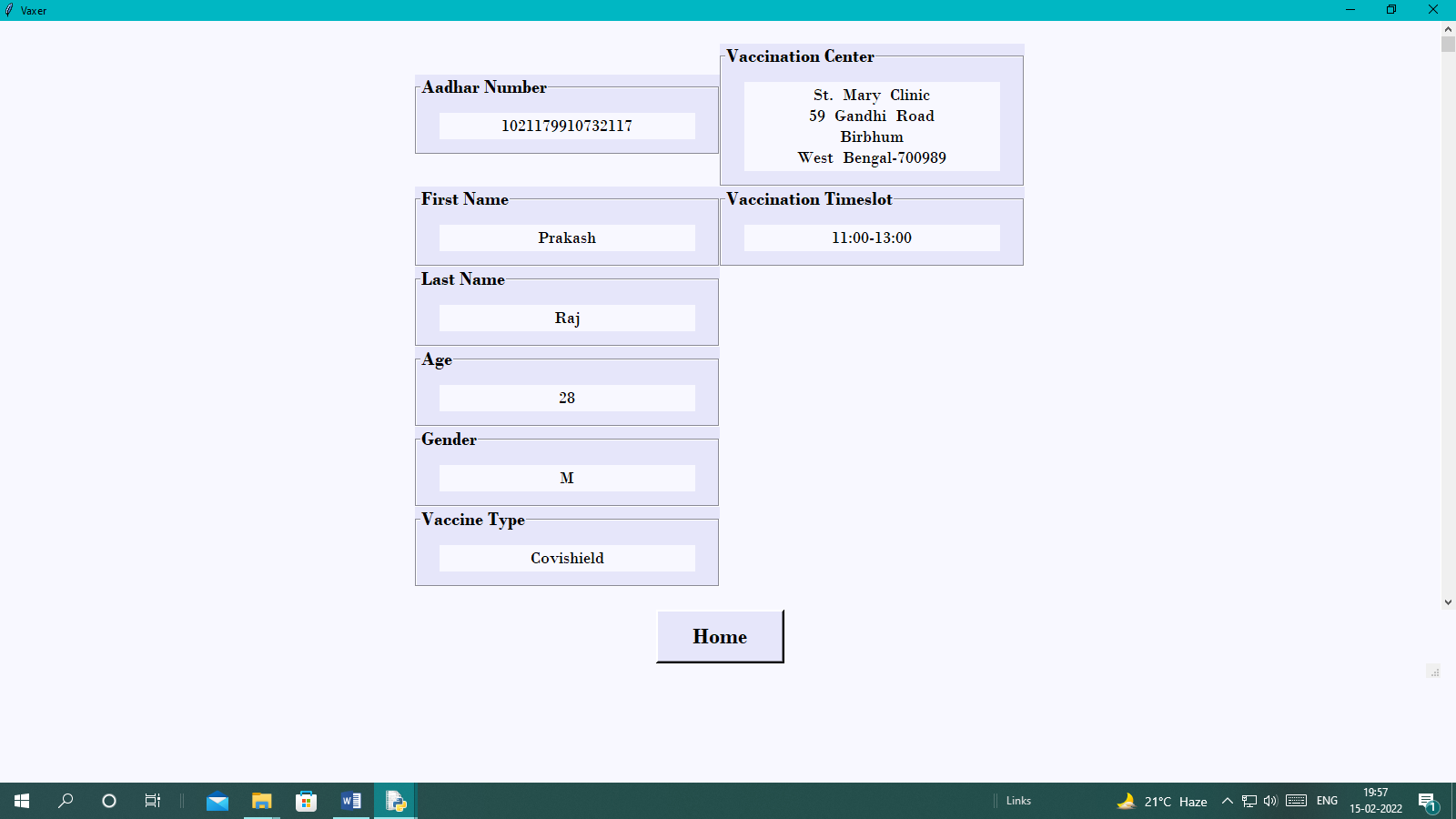
**For registering, user has to click on Register yourself and fill in the details. Clicking on Home, directs the user to the greeting window**



**Clicking on Continue will direct the user to the following window**

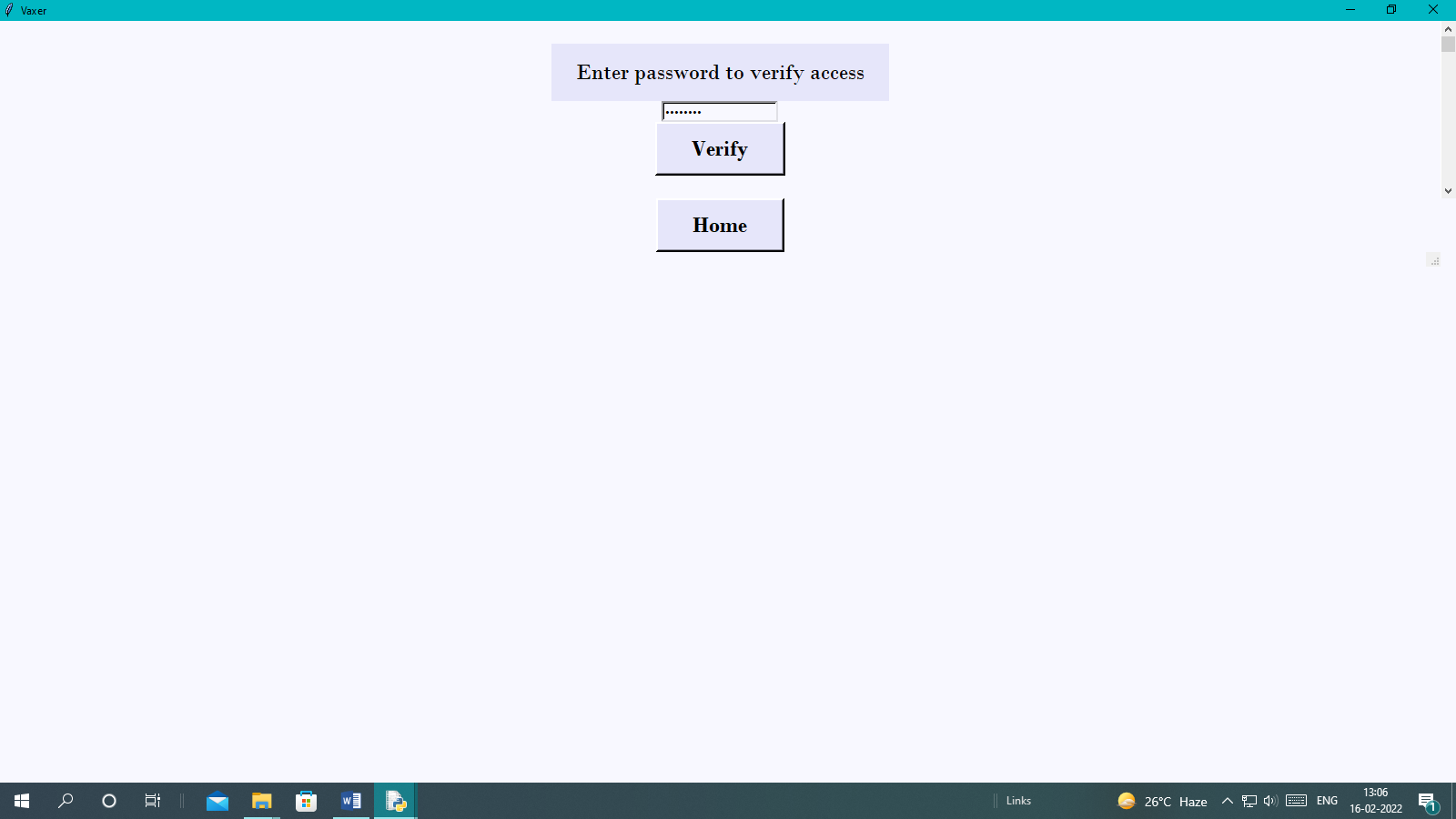


**Clicking on Show my details will allow the user to view the details that the user has filled in**

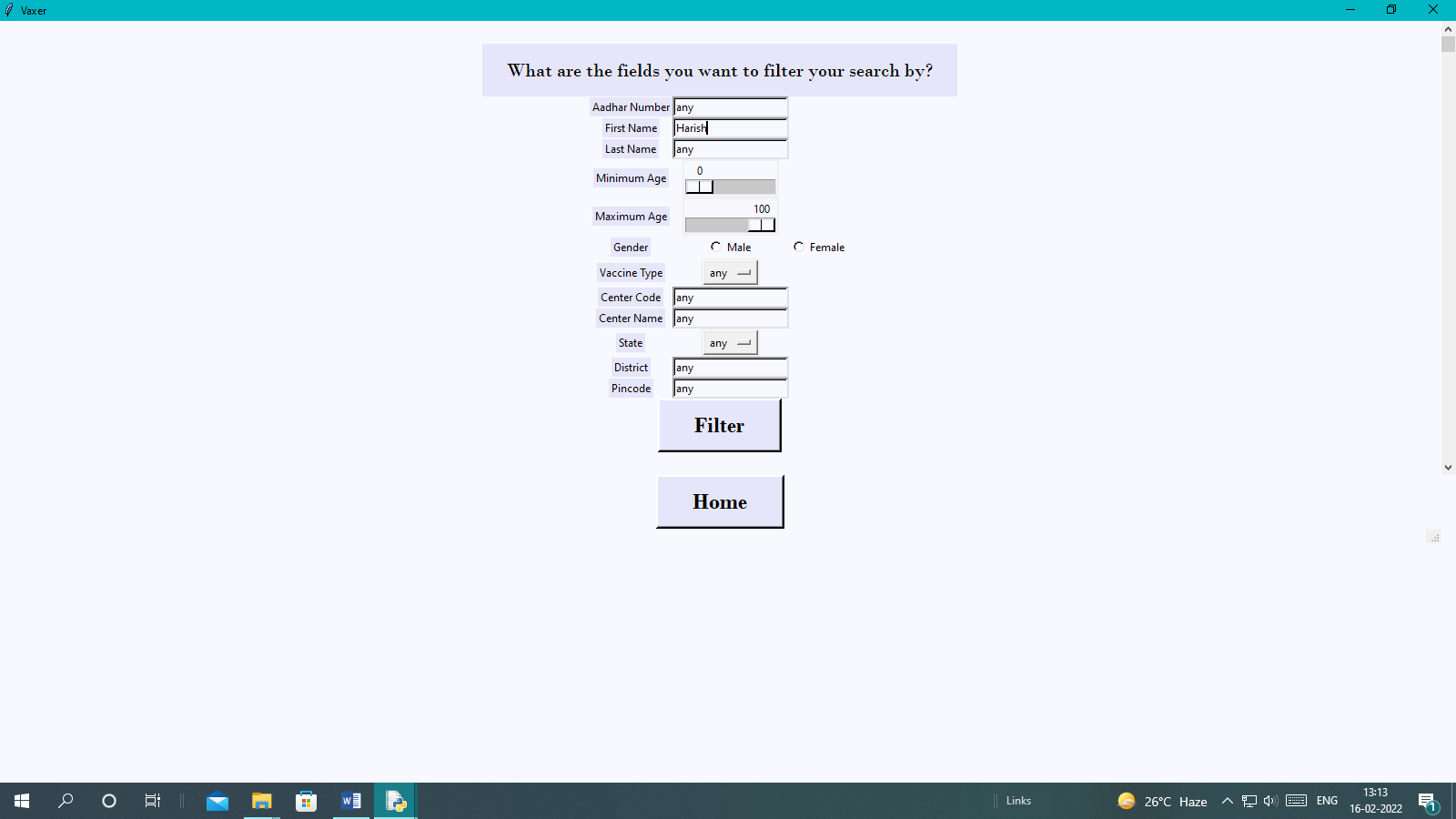
**If user already has filled in the details, he/she/they can login to view their details**

**Administration window**

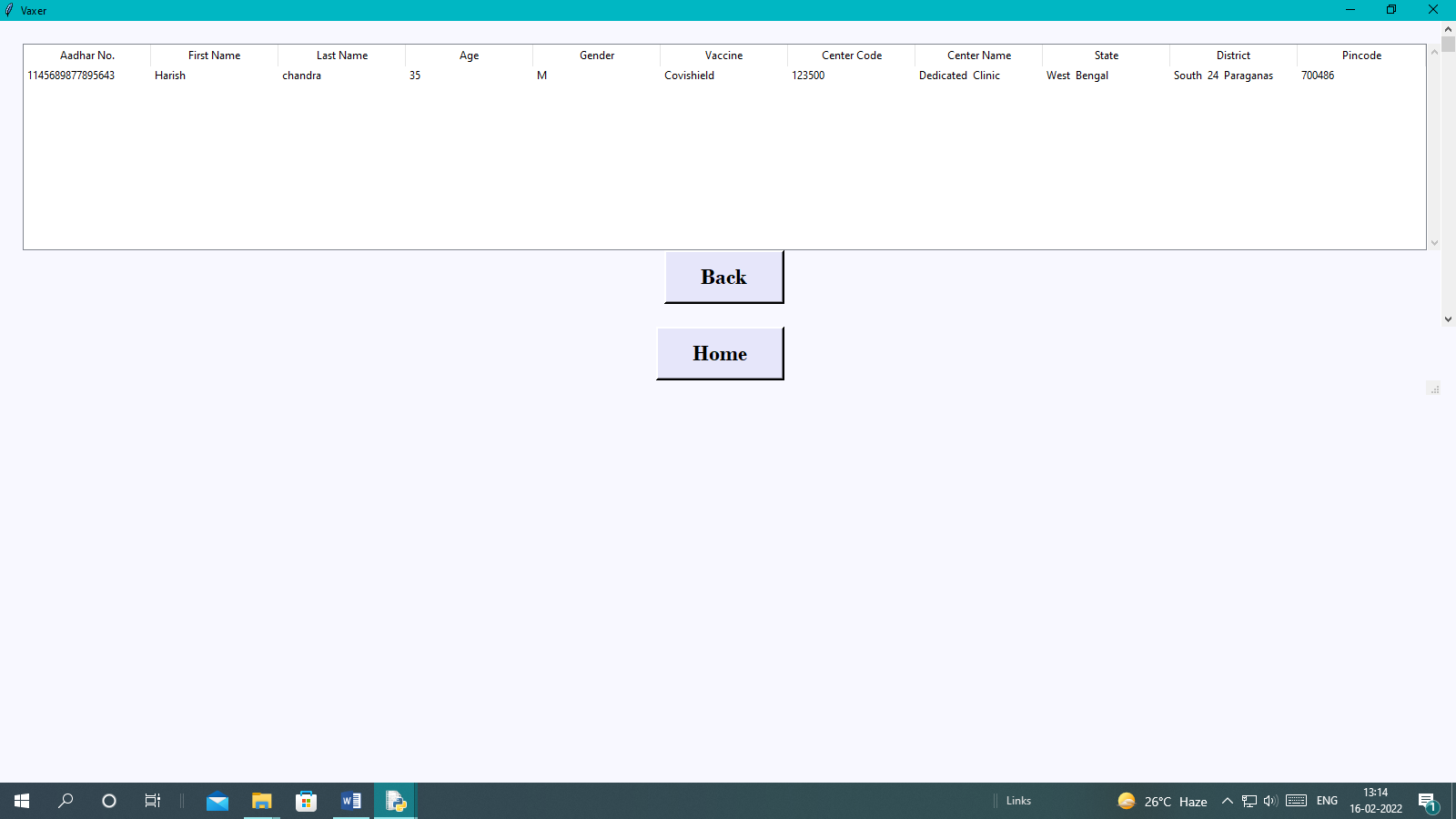
**The verification window will appear and the person will get access only if the password is correct otherwise error message will pop up**



**If the administrator does not filter the search by keywords, all the records will be shown**



**Clicking on filter will give the desired result**



**SCOPE**

* **A step towards digitalization**
* **Enhances the chances of reducing physical contact job during this pandemic**
* **Systematic approach to record management**
* **Saves time.**
* **Easy to handle.**
* **Low hardware and software requirements reduces cost of setup and running.**
* **GUI makes program convenient and intuitive to use.**

**LIMITATIONS**

* **Online facility is not available.**
* **The response might be slow as implementation of the system has been done manually.**
* **Program might start slightly slow on older hardware due to prolonged file reading duration.**

**CONCLUSION**

This program aims at providing an effective record management system for healthcare authorities, saves time and hassle, giving maximum output from minimal efforts. This program also allows users to get themselves registered for their vaccination without having to necessarily step out of their houses. This program targets at making record management effortless and efficacious. Thus we feel it would be beneficial for authorities responsible for keeping vaccine related records in this pandemic period.

**BIBLIOGRAPHY**

1. ***Computer Science With Python - Class XII By : Sumita Arora***
2. ***A Project Report On Vaccination Management System***

***Websites used*** [***https://www.geeksforgeeks.org/***](https://www.geeksforgeeks.org/)

[***https://www.w3resource.com***](https://www.w3resource.com)

***\*\*\****