

**NEARBY FACILITIES SPOTTER APP**

**BY**

**USING SQL**

Submitted by

SAI SANJAY S.V (231501142)

SARVEESH B (231501148)

RINO CALVIN B (231501135)

**CS23332 – DATABASE MANAGEMENT SYSTEM**

Department of Artificial Intelligence and Machine Learning

Rajalakshmi Engineering College, Thandalam

BONAFIDE CERTIFICATE

This is to certify that the Mini project work titled “**NEARBY FACILITIES SPOTTER APP BY USING SQL”** done by, SAI SANJAY S.V (231501142), SARVEESH (231501148), RINO CALVIN (231501135) is a record of Bonafide work carried out by him/her under my supervision as a part of MINI PROJECT for the subject CS23332 - DATABASE MANAGEMENT SYSTEM by Department of Artificial Intelligence and Machine Learning.

SIGNATURE SIGNATURE

Dr.Sekar K M.E., Ph.D, SREESUBHA

HEAD OF THE DEPARTMENT FACULTY IN CHARGE

Department of Artificial Intelligence Department of Artificial Intelligence

And Machine Learning And Machine Learning

Rajalakshmi Engineering College Rajalakshmi Engineering College,

Thandalam ,Chennai- 602105 Thandalam ,Chennai-602105

**TABLE OF CONTENTS:**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Chapter** | **Page Number** |
| 1. | PROBLEM STATEMENT | 4 |
| 2. | NOVELTY | 4 |
| 3. | KEY FEATURES | 5 |
| 4. | IMPACTS | 6 |
| 5. | FUNDAMENTALS FOR DATABASE | 6 |
| 6. | TECH STACK | 7 |
| 7. | SOURCE CODE | 7 |
| 8. | OUTPUT | 42 |
| 9. | CONCLUSION | 44 |
| 11. | REFERENCES | 45 |

**NEARBY FACILITIES SPOTTER APP**

**PROBLEM STATEMENT:**

In a highly interconnected world, navigating nearby facilities like restaurants, temples, entertainment hubs, and hospitals remains cumbersome due to fragmented information and lack of tailored solutions. Users often face challenges such as:

* **Scattered Resources**: Information about nearby facilities is spread across multiple platforms.
* **Decision Fatigue**: Users struggle to choose the best options among a sea of choices without personalized or filtered results.
* **Time-Consuming Processes**: Manually searching and organizing facility data and locations is inefficient.
* **Limited Admin Control**: Traditional apps lack an admin system to dynamically manage facility databases.

This project addresses these gaps by providing a centralized platform that simplifies facility discovery and navigation.

**NOVELTY:**

INSIDEX stands out due to the following innovative elements:

1. **Interactive Admin and User Interfaces**: A dual-system allowing users to explore and admins to manage facility data dynamically.
2. **Top-4 Recommendations**: Automatically filtering and showcasing the top 4 facilities in selected categories for user convenience.
3. **Speech Recognition Integration**: Hands-free interaction via voice commands to search and explore facilities.
4. **Real-Time Mapping**: Visual representation of chosen locations with geocoding and customizable map styles.
5. **SMS Notifications**: Leveraging APIs to keep users informed about updates and confirmations via SMS.
6. **User-Centric Features**: Options for customization (themes, wallpapers) and user activity tracking ensure a personalized experience

**KEY FEATURES:**

1. **User Account Management**:

* New users can sign up and existing users can log in.
* Secure handling of user credentials.

1. **Admin Operations**:

* Insert, update, and delete facility records.
* Monitor user activity and profiles.

1. **Facility Search and Navigation**:

* Categorized search for facilities (e.g., hotels, hospitals, entertainment).
* Interactive map view using geocode integration.

1. **Recent Search Tracking**: Users can revisit previously searched facilities through stored activity logs.
2. **Speech Recognition and Text-to-Speech**: Simplified interaction through spoken commands and system-generated audio responses.
3. **Database and File Operations**:

* **Binary Files**: To store details about areas and user activities.
* **SQL Database**: To maintain user profiles and facility records.

1. **Customizability**:

* Changeable themes, wallpapers, and map views (OpenStreetMap, Google Maps).

**IMPACTS:**

1. **User Benefits**:
   * Simplified decision-making with categorized top-4 recommendations.
   * Hands-free navigation enhances accessibility for visually impaired or busy users.
2. **Administrative Efficiency**:
   * Easy management of facility records ensures an up-to-date database.
   * Real-time user monitoring aids in personalized support and service optimization.
3. **Technological Advancement**:
   * Incorporating speech recognition and dynamic maps into facility-based apps fosters innovation.
4. **Enhanced Engagement**:
   * Customization options and SMS alerts keep users connected to the app, driving repeated usage.

**FUNDAMENTAL FOR DATABASE:**

1. **Database Design**:

* **SQL Table**: Maintains user information, including name, username, password, age, contact number, and address.
* **Binary Files**:
  + Areas1.dat: Stores details of areas and their corresponding facilities.
  + Recent.dat: Records user search history with timestamps.

2. **Relational Model**:

* Primary keys such as usernames ensure unique identification of records.
* Relationships between user activities and facility data enable seamless access and updates.

3. **CRUD Operations**:

* **Create**: Adding new facilities and user accounts.
* **Read**: Fetching and displaying facility and user data.
* **Update**: Editing facility details.
* **Delete**: Removing outdated or irrelevant records.

4. **Efficiency and Security**:

* Data is stored securely using Python modules like pickle for binary files and mysql.connector for SQL operations.
* Proper error handling ensures data integrity and robust system performance.

**TECH STACK:**

- Frontend: Tkinter

- Backend: Python to integrate SQL database to the program and front end keys and buttons

- Database - MySQL: Maintains user information, including name, username, password, age, contact number, and address.

**SOURCE CODE:**

from tkinter import \*

from tkinter.ttk import Combobox

from tkinter import messagebox

from tkinter import filedialog

from tkinter.messagebox import askyesno

import yaml

from PIL import ImageTk,Image,ImageSequence

import time

import mysql.connector

import json

import pickle

import requests as re

import tkintermapview

import speech\_recognition as sr

import pyttsx3

import datetime

-------------------------------------#CREATING MAINWINDOW#---------------------------------------------

root=Tk()

root.title("INSIDEX")

root.geometry("1200x900")

root.state('zoomed')

def play\_gif():# DISPLAYING LOGO

img=Image.open(r'Images\gif.gif')

lbl=Label(root)

lbl.place(x=0,y=0)

def resizer(e):

bg1=Image.open(r'Images\gif.gif')

resized\_bg=bg1.resize((e.width,e.height),Image.ANTIALIAS)

new\_bg=ImageTk.PhotoImage(resized\_bg)

for img in ImageSequence.Iterator(img):

root.bind("<Configure>",resizer)

img=ImageTk.PhotoImage(img)

lbl.config(image=img)

root.update()

time.sleep(0.01)

play\_gif()

def canvas(a,file):

global my\_canvas,new\_bg

my\_canvas=Canvas(a,width=800,height=500)

my\_canvas.pack(fill="both",expand=True)

ico=ImageTk.PhotoImage(Image.open(r"Images\ins.jpeg"))

root.iconphoto(False,ico)

def resizer(e):

global bg1,resized\_bg,new\_bg

bg1=Image.open(file)

resized\_bg=bg1.resize((e.width,e.height),Image.ANTIALIAS)

new\_bg=ImageTk.PhotoImage(resized\_bg)

my\_canvas.create\_image(0,0,image=new\_bg,anchor="nw")

my\_canvas.bind("<Configure>",resizer)

root.geometry("1200x900")

status=1

-------------------------------#establishing connection to the SQL Table#--------------------------

m=mysql.connector.connect(host='localhost',user='root',passwd='Sanju@1712',database='sanjay')

c=m.cursor()

logo=Image.open("Images\logo2.jpeg")

resized=logo.resize((300,100),Image.ANTIALIAS)

new\_pic=ImageTk.PhotoImage(resized)

def admin():#for admin operations

def back1():

frame.place\_forget()

my\_canvas.destroy()

admin()

def login():

user=un.get()

password=pw.get()

userpw={'Sai Sanjay':'Sanju@1712','MR sparky':'suryaisadhonifan','Sumukesh':'Sumu!@123'}

if user in userpw and userpw[user]==password:

global status

status=0

else:

messagebox.showerror('WARNING BY INSIDEX','You have entered the wrong username/password')

status=1

if status==0:

frame1.place\_forget()

global frame

frame=Frame(my\_canvas,width=500,height=400,bg='#00ddff',highlightbackground='#ff00d4',highlightthickness=3,relief=RAISED)

Label(frame,text="ADMIN OPERATIONS",font=("Castellar",25,"bold"),bg="#ee2a7b",fg="white").place(x=75,y=10)

insertbutton=Button(frame,text="Insert",font="Castellar 15",command=insert,padx=30,pady=30,bg="#FF5349")

insertbutton.place(x=50,y=100)

deletebutton=Button(frame,text="Delete",font="Castellar 15",command=delete,padx=30,pady=30,bg="#0555c3")

deletebutton.place(x=300,y=100)

updatebutton=Button(frame,text="Update",font="Castellar 15",command=update,padx=30,pady=30,bg="#69e09b")

updatebutton.place(x=175,y=250)

Button(frame,text='Back',command=back1).place(x=50,y=300)

frame.place(x=725,y=200)

my\_canvas.destroy()

canvas(root,r'Images\admin.jpg')

frame1=Frame(my\_canvas,width=500,height=400,bg='#00ddff',highlightbackground='#ff00d4',highlightthickness=3)

def back():

frame1.place\_forget()

my\_canvas.destroy()

home()

lab=Label(frame1,text="ADMIN LOGIN PAGE",font=("Castellar",25,"bold"),bg="#ee2a7b",fg="white")

lab.place(x=70,y=5)

def on\_enter(e):

un.delete(0,'end')

def on\_leave(e):

if un.get()=='':

un.insert(0,'Username')

un=Entry(frame1,width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

un.place(x=30,y=80)

un.insert(0,'Username')

un.bind('<FocusIn>',on\_enter)

un.bind('<FocusOut>',on\_leave)

Frame(frame1,width=295,height=2,bg='black').place(x=25,y=107)

def on\_enter(e):

pw.delete(0,'end')

pw.config(show='\*')

def on\_leave(e):

if pw.get()=='':

pw.insert(0,'Password')

pw=Entry(frame1,width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

pw.place(x=30,y=130)

pw.insert(0,'Password')

pw.bind('<FocusIn>',on\_enter)

pw.bind('<FocusOut>',on\_leave)

Frame(frame1,width=295,height=2,bg='black').place(x=25,y=157)

def toggle\_password():

if pw.cget('show') == '':

pw.config(show='\*')

toggle\_btn.config(text='Show Password')

else:

pw.config(show='')

toggle\_btn.config(text='Hide Password')

toggle\_btn =Button(frame1, text='Show Password',command=toggle\_password,bg='#cc8899')

toggle\_btn.place(x=350,y=137)

Button(frame1,image=login\_btn,command=login,borderwidth=0).place(x=150,y=200)

Button(frame1, text="Back", command=back).place(x=80,y=200)

frame1.place(x=750,y=200)

-------------------------------------#INSERTING RECORDS#---------------------------------

def insert():

frame.place\_forget()

root.title("Insert")

ins=Frame(my\_canvas,width=500,height=400,bg='#00ddff',highlightbackground='#ff00d4',highlightthickness=3)

Label(ins,text='INSERTION OF RECORDS',font=("Castellar",25,"bold"),bg="#ee2a7b",fg="white").place(x=25,y=0)

f=open('Areas1.dat','rb')

options=[]

l=pickle.load(f)

for i in l:

options.append(i)

f.close()

def check(e):

typed=e.widget.get()

if typed == '':

combo\_box['values']=options

else:

data=[]

for i in options:

if typed.lower() in i.lower():

data.append(i)

combo\_box['values']=data

area=Label(ins,text="Enter Area to insert:",font="Algeria 11",bg="white",fg="red")

area.place(x=25,y=75)

combo\_box=Combobox(ins,value=options)

combo\_box.place(x=200,y=75)

combo\_box.bind("<KeyRelease>",check)

fac=Label(ins,text="Enter Facility to insert:",font="Algeria 11",bg="white",fg="red")

fac.place(x=25,y=150)

l=['Hotels and eateries','Hospitals','Shopping','Temples','Entertainment']

listbox=Listbox(ins)

listbox.place(x=200,y=150)

for i in range(len(l)):

listbox.insert(END,l[i])

global a

a=[]

for i in listbox.curselection():

a.append(listbox.get(i))

ins.place(x=750,y=200)

f2=Frame(my\_canvas,width=500,height=600,bg='#00ddff',highlightbackground='#ff00d4',highlightthickness=3)

def undo():

f2.place\_forget()

insert()

def undo1():

ins.place\_forget()

frame.place(x=750,y=200)

def show():

global g,p,q,r,a,b

a=[]

for i in listbox.curselection():

a.append(listbox.get(i))

if combo\_box.get() and a:

f2.place(x=750,y=50)

ins.place\_forget()

root.title("Insert 2")

root.configure(bg="#7018D3")

Label(f2,text=str(combo\_box.get()),font=('Arial,30')).place(x=25,y=10)

g=[]

for i in listbox.curselection():

b=listbox.get(i)

Label(f2,text=b,font=('Arial,30')).place(x=150,y=10)

Label(f2,text='Enter new '+b+' name',font="Algeria 11",bg="white",fg="red").place(x=25,y=50)

p=Entry(f2,width=25,insertwidth=2)

p.place(x=300,y=50)

Label(f2,text="Enter new "+b+' open timings',font="Algeria 11",bg="white",fg="red").place(x=25,y=100)

q=Entry(f2,width=25,insertwidth=2)

q.place(x=300,y=100)

Label(f2,text='Enter new '+b+' phone details',font="Algeria 11",bg="white",fg="red").place(x=25,y=150)

r=Entry(f2,width=25,insertwidth=2)

r.place(x=300,y=150)

back=Button(f2,text="Back",command=undo)

back.place(x=25,y=250)

else:

messagebox.showerror('WARNING BY INSIDEX','Please fill all the fields')

def clickins():

global a

f=open("Areas1.dat",'rb')

temp={}

temp=pickle.load(f)

#print(value)

if p.get() and q.get() and r.get():

g.append(p.get()+',')

g.append('Open:'+q.get()+',')

g.append('Contact:'+r.get())

for i in a:

s=''

for j in g:

s+= j

temp[combo\_box.get()][i].append(s)

l=Text(f2,height=10,width=50)

l.place(x=25,y=300)

for i in a:

l.insert(END,yaml.dump([combo\_box.get(),i,temp[combo\_box.get()][i]]

,sort\_keys=False,default\_flow\_style=False)

inse.config(state="disabled",text="inserted")

messagebox.showinfo("INFO BY INSIDEX!","You have successfully inserted "+b+" in "+combo\_box.get())

f.close()

inse=Button(f2,text="Insert the selected area and facility",font="Castellar 13",command=clickins,bg="#0555c3")

inse.place(x=25,y=200)

next=Button(ins,text='Next',padx=10,pady=10,command=show)

next.place(x=350,y=275)

back=Button(ins,text="Back",command=undo1)

back.place(x=25,y=350)

------------------------------#DELETING RECORDS#----------------------------

def delete():

global a

frame.place\_forget()

root.title("Delete")

k=Frame(my\_canvas,width=500,height=400,bg='#00ddff',highlightbackground='#ff00d4',highlightthickness=3)

Label(k,text='DELETION OF RECORDS',font=("Castellar",25,"bold"),bg="#ee2a7b",fg="white").place(x=25,y=0)

f=open('Areas1.dat','rb')

options=[]

l=pickle.load(f)

for i in l:

options.append(i)

def check(e):

typed=e.widget.get()

if typed == '':

combo\_box['values']=options

else:

data=[]

for i in options:

if typed.lower() in i.lower():

data.append(i)

combo\_box['values']=data

area=Label(k,text="Enter Area to delete:",font="Algeria 11",bg="white",fg="red")

area.place(x=25,y=75)

combo\_box=Combobox(k,value=options)

combo\_box.place(x=200,y=75)

combo\_box.bind("<KeyRelease>",check)

fac=Label(k,text="Enter Facility to delete:",font="Algeria 11",bg="white",fg="red")

fac.place(x=25,y=150)

l=['Hotels and eateries','Hospitals','Shopping','Temples','Entertainment']

listbox=Listbox(k)

listbox.place(x=200,y=150)

for i in range(len(l)):

listbox.insert(END,l[i])

a=[]

for i in listbox.curselection():

a.append(listbox.get(i))

k.place(x=750,y=200)

f1=Frame(my\_canvas,width=500,height=500,bg='#00ddff',highlightbackground='#ff00d4',highlightthickness=3)

def undo():

f1.place\_forget()

delete()

def undo1():

k.place\_forget()

frame.place(x=750,y=200)

def show():

global a,result,tp

a=[]

for i in listbox.curselection():

a.append(listbox.get(i))

if combo\_box.get() and a:

f1.place(x=750,y=100)

k.place\_forget()

root.title("Delete 2")

root.configure(bg="#7018D3")

Label(f1,text=combo\_box.get(),font=('Arial,30')).place(x=25,y=10)

a=[]

for i in listbox.curselection():

a.append(listbox.get(i))

tp=listbox.get(i)

Label(f1,text=tp,font=('Arial,30')).place(x=25,y=50)

Label(f1,text='Select the places u want to delete:',font=('Arial,10'),bg='#FFCCCB',fg='black').place(x=200,y=10)

fac=Listbox(f1,selectmode="multiple",width=50)

fac.place(x=200,y=50)

f=open("Areas1.dat",'rb')

d=pickle.load(f)

for i in a:

for j in d[combo\_box.get()][i]:

fac.insert(END,j)

def get(event):

global index,value,b

index,b=[],[]

sel=event.widget.curselection()

for i in range(len(sel)):

index.append(sel[i])

value=[]

for i in range(len(index)):

value.append(event.widget.get(index[i]))

result.set(str(value))

result=StringVar()

fac.bind("<<ListboxSelect>>",get)

back=Button(f1,text="Back",command=undo)

back.place(x=25,y=100)

else:

messagebox.showerror('WARNING BY INSIDEX','Please fill all the fields')

def clickdel():

f=open("Areas1.dat",'rb')

temp={}

temp=pickle.load(f)

for i in a:

for j in value:

t={'areas':combo\_box.get(),'fac':i,'name':j}

temp[t['areas']][t['fac']].remove(t['name'])

l=Text(f1,height=10,width=50)

l.place(x=25,y=300)

for i in a:

l.insert(END,yaml.dump([combo\_box.get(),i,temp[combo\_box.get()][i]]

,sort\_keys=False,default\_flow\_style=False))

dele.config(state="disabled",text="deleted")

messagebox.showinfo("INFO BY INSIDEX!","You have successfully deleted "+tp+"from "+combo\_box.get())

f.close()

dele=Button(f1,text="Delete the selected area and facility",font="Castellar 13",command=clickdel,bg="#0555c3")

dele.place(x=25,y=200)

next=Button(k,text='Next',padx=10,pady=10,command=show)

next.place(x=350,y=275)

back=Button(k,text="Back",command=undo1)

back.place(x=25,y=350)

------------------------------#UPDATING RECORDS#-----------------------

def update():

global a

frame.place\_forget()

root.title("Update")

upd=Frame(my\_canvas,width=500,height=400,bg='#00ddff',highlightbackground='#ff00d4',highlightthickness=3)

Label(upd,text='UPDATION OF RECORDS',font=("Castellar",25,"bold"),bg="#ee2a7b",fg="white").place(x=25,y=0)

f=open('Areas1.dat','rb')

options=[]

l=pickle.load(f)

for i in l:

options.append(i)

def check(e):

typed=e.widget.get()

if typed == '':

combo\_box['values']=options

else:

data=[]

for i in options:

if typed.lower() in i.lower():

data.append(i)

combo\_box['values']=data

area=Label(upd,text="Enter Area to update:",font="Algeria 11",bg="white",fg="red")

area.place(x=25,y=75)

combo\_box=Combobox(upd,value=options)

combo\_box.place(x=200,y=75)

combo\_box.bind("<KeyRelease>",check)

fac=Label(upd,text="Enter Facility to update:",font="Algeria 11",bg="white",fg="red")

fac.place(x=25,y=150)

l=['Hotels and eateries','Hospitals','Shopping','Temples','Entertainment']

listbox=Listbox(upd)

listbox.place(x=200,y=150)

for i in range(len(l)):

listbox.insert(END,l[i])

a=[]

for i in listbox.curselection():

a.append(listbox.get(i))

upd.place(x=750,y=200)

f2=Frame(my\_canvas,width=500,height=500,bg='#00ddff',highlightbackground='#ff00d4',highlightthickness=3)

def undo():

f2.place\_forget()

update()

def undo1():

upd.place\_forget()

frame.place(x=750,y=200)

def show():

global a,result,up

a=[]

for i in listbox.curselection():

a.append(listbox.get(i))

if combo\_box.get() and a:

f2.place(x=750,y=100)

upd.place\_forget()

root.title("Update 2")

root.configure(bg="#7018D3")

Label(f2,text=combo\_box.get(),font=('Arial,30')).place(x=25,y=10)

a=[]

for i in listbox.curselection():

a.append(listbox.get(i))

up=listbox.get(i)

f=open("Areas1.dat",'rb')

d=pickle.load(f)

x1,y1=150,50

for i in a:

Label(f2,text=i,font=('Arial,30')).place(x=25,y=50)

lo=d[combo\_box.get()][i]

ent={}

for j in lo:

n=Entry(f2,width=40,font=('Arial 10'))

n.insert(0,j)

ent[j]=n

n.place(x=x1,y=y1)

y1+=50

back=Button(f2,text="Back",command=undo)

back.place(x=25,y=y1+25)

else:

messagebox.showerror('WARNING BY INSIDEX','Please fill all the fields')

def clickupd():

f=open("Areas1.dat",'rb')

temp={}

temp=pickle.load(f)

o=[]

for j in lo:

o.append(ent[j].get())

for i in range(len(a)):

for j in range(len(o)):

temp[combo\_box.get()][a[i]][j]=o[j]

for i in a:

l=Text(f2,height=10,width=50)

l.place(x=25,y=300)

l.insert(END,yaml.dump([combo\_box.get(),i,temp[combo\_box.get()][i]]

,sort\_keys=False,default\_flow\_style=False))

upda.config(state="disabled",text="updated")

messagebox.showinfo("INFO BY INSIDEX!","You have successfully updated"+up+"in"+combo\_box.get())

f.close()

upda=Button(f2,text="Update the selected area and facility",font="Castellar 13",command=clickupd,bg="#0555c3")

upda.place(x=25,y=y1+50)

next=Button(upd,text='Next',padx=10,pady=10,command=show)

next.place(x=350,y=275)

back=Button(upd,text="Back",command=undo1)

back.place(x=25,y=350)

---------------#DECLARING IMAGE VARIABLES#------------------

login\_btn=PhotoImage(file=r"Images\login.png").subsample(3,3)

signup\_btn=PhotoImage(file=r"Images\signup.png").subsample(3,3)

back\_btn=PhotoImage(file=r"Images\back.png").subsample(4,4)

wallad=r"Images\background.png"

wall=Image.open(wallad)

wall=wall.resize((400,200),Image.ANTIALIAS)

wall1=ImageTk.PhotoImage(wall)

wall2ad=r"Images\wall2.png"

wall2=PhotoImage(file=wall2ad).subsample(4,4)

wall3ad=r"Images\wall3.png"

wall3=PhotoImage(file=wall3ad).subsample(4,4)

mic\_btn=PhotoImage(file=r"Images\microphone.png").subsample(7,7)

user\_profile=PhotoImage(file=r"Images\user profile.png").subsample(4,4)

recent=PhotoImage(file=r"Images\recent.png").subsample(12,12)

op=Image.open(r"Images\empty.png")

op=op.resize((100,100),Image.ANTIALIAS)

empty=ImageTk.PhotoImage(op)

def upload\_new(fo):

global op2

op1=Image.open(fo)

op1=op1.resize((50,50),Image.ANTIALIAS)

op2=ImageTk.PhotoImage(op1)

return op2

qw=[]

----------------#USER INTERFACE#--------------------

def user():

def facili(arg): #To show facilities available in selected area

recentl=open('recent.dat','wb')

k=str(datetime.datetime.now())

qw.append(arg)

d={k[:10]:qw}

pickle.dump(d,recentl)

recentl.close()

engine=pyttsx3.init('sapi5')

voices=engine.getProperty('voices')

engine.setProperty('voice','voices[0].id')

def speak(text): #for speaking

engine.say(text)

engine.runAndWait()

faci=Toplevel(root)

faci.geometry("700x500")

faci.configure(bg="#FF00EC")

Label(faci,text=f"Welcome to:{arg.capitalize()}",font="Timesnewroman 20").grid(row=0,column=4)

speak(f"{arg} window is open now")

def sizeZ(fi):

bg=Image.open(fi)

bg=bg.resize((200,200),Image.ANTIALIAS)

bg1=ImageTk.PhotoImage(bg)

return bg1

recfac=[]

def detail(e):

recfac.append(e)

facil=Toplevel(faci)

facil.geometry("700x500")

Label(facil,text=f"{e} of {arg.capitalize()}")

t=Text(facil,font="Courier 14",selectbackground="blue",selectforeground="black")

t.pack()

Scrollbar(facil,command=t.yview).pack()

f=open('Areas1.dat','rb')

d=pickle.load(f)

t.insert(END,yaml.dump([arg,e,d[arg.capitalize()][e]],sort\_keys=False,default\_flow\_style=False))

f.close()

facil.mainloop()

hotels=sizeZ(r'Images\hotels.png')

hospital=sizeZ(r'Images\hospital.png')

temple=sizeZ(r'Images\temple.png')

shopping=sizeZ(r'Images\shopping.png')

entertain=sizeZ(r'Images\entertainment.png')

Button(faci,image=hotels,text='Hotels',compound=TOP,bg="#FF0000",command=lambda:detail('Hotels and eateries')).grid(row=1,column=3)

Button(faci,image=hospital,text='Hospitals',compound=TOP,bg="#007CFF",command=lambda:detail('Hospitals')).grid(row=1,column=4)

Button(faci,image=temple,text='Temple',compound=TOP,bg="#87FF00",command=lambda:detail('Temples')).grid(row=1,column=5)

Button(faci,image=shopping,text='Shopping',compound=TOP,bg="#00FFC5",command=lambda:detail('Shopping')).grid(row=2,column=3)

Button(faci,image=entertain,text='Entertainment',compound=TOP,bg="#C6DCD5",command=lambda:detail('Entertainment')).grid(row=2,column=4)

faci.mainloop()

def afterlogin():

global pro

c.execute("select \* from user;")

r=c.fetchall()

unn=un.get()

pww=pw.get()

for i in r:

if un.get() and pw.get() in i:

name=i[0]

#adr=i[5]

pro=i

global status

status=0

if status==1:

messagebox.showerror('WARNING BY INSIDEX','You have entered the wrong username/password')

elif status==0:

my\_canvas.destroy()

canvas(root,r'Images\background.png')

def showmap(): #for Mapview

c.execute("select \* from user;")

ro=c.fetchall()

for i in ro:

if i[1]==unn and i[2]==pww:

adr=i[5]

ol=Toplevel(root)

ol.geometry("1000x1000")

ol.configure(bg="#09D1F1")

map\_widget = tkintermapview.TkinterMapView(ol,width=800, height=600,corner\_radius=0)

map\_widget.place(x=0,y=0)

map\_widget.set\_address(adr)

map\_widget.set\_zoom(15)

def add\_marker\_event(coords):

print(f"{name}'s location:", coords)

new\_marker = map\_widget.set\_marker(coords[0], coords[1], text=f"{name}'s location")

messagebox.showinfo('INFO BY INSIDEX','Marked your location')

map\_widget.add\_right\_click\_menu\_command(label="Add Marker",command=add\_marker\_event,pass\_coords=True)

def change\_map(new\_map:str):

if new\_map=="OpenStreetMap":#Types of map display

map\_widget.set\_tile\_server("https://a.tile.openstreetmap.org/{z}/{x}/{y}.png")

elif new\_map == "Google normal":

map\_widget.set\_tile\_server("https://mt0.google.com/vt/lyrs=m&hl=en&x={x}&y={y}&z={z}&s=Ga", max\_zoom=22)

elif new\_map == "Google satellite": map\_widget.set\_tile\_server("https://mt0.google.com/vt/lyrs=s&hl=en&x={x}&y={y}&z={z}&s=Ga", max\_zoom=22)

var=StringVar()

var.set("OpenStreetMap")

mode=["OpenStreetMap", "Google normal", "Google satellite"]

mapopt=OptionMenu(ol,var,\*(mode),command=change\_map)

mapopt.place(x=800,y=0)

spl=adr.split(',')

sl=spl[0]

'''Label(ol,text=Your area is {spl[0]}

Click to search facilities in your area).place(x=800,y=50)'''

Button(ol,text=f"Search in {spl[0]}",command=lambda:facili(sl)).place(x=800,y=100)

ol.mainloop()

def change():

frame5=Frame(my\_canvas,width=500,height=400,bg="#7bd1ed")

frame5.place(x=750,y=200)

Label(frame5,text=f"WELCOME {name.title()}!",font="Helvetica 15",bg="#8C00FF").place(x=0,y=10)

Label(frame5,text="Click to mark your current location").place(x=0,y=70)

Button(frame5,text="Mark your location",command=showmap).place(x=200,y=70)

Label(frame5,text="Click to tell the area you want to search:",font="Algeria 11",bg="white",fg="red").place(x=25,y=100)

def speaking(): #Speaking function

global recentl

engine=pyttsx3.init('sapi5')

voices=engine.getProperty('voices')

engine.setProperty('voice','voices[0].id')

def speak(text):

engine.say(text)

engine.runAndWait()

def takeCommand(): #Listening function

r=sr.Recognizer()

with sr.Microphone() as source:

print("Listening...")

audio=r.listen(source)

try:

statement=r.recognize\_google(audio,language='en')

print(f"user said:{statement}\n")

except Exception :

speak("Pardon me, please say that again")

return "None"

return statement

f=open('Areas1.dat','rb')

opt=[]

l=pickle.load(f)

for i in l:

opt.append(i.lower())

f.close()

if \_\_name\_\_=='\_\_main\_\_':

while True:

speak("Speak now")

statement = takeCommand().lower()

if statement in opt:

facili(statement)

time.sleep(5)

time.sleep(3)

def recentfn():#Recently searched

recentl=open('recent.dat','rb')

d=pickle.load(recentl)

recen=Text(frame5)

recen.place(x=25,y=250)

for i in d:

recen.insert(END,yaml.dump([i,d[i]],sort\_keys=False,default\_flow\_style=False))

recentl.close()

Button(frame5,image=mic\_btn,command=speaking).place(x=300,y=100)

Label(frame5,text="Try saying 'Padi' to search in Padi",bg="#09A8F1",padx=10,pady=10).place(x=200,y=150)

Label(frame5,text="Recently searched",font='Timesnewroman 15',bg="#FF0092",fg="black",padx=10,pady=10).place(x=25,y=200)

Button(frame5,image=recent,command=recentfn).place(x=250,y=200)

def wallpaper():#Change wallpaper

w=Toplevel(root)

w.geometry("1000x400")

w.configure(bg="#B111E9")

def wall\_change(file):

my\_canvas.destroy()

canvas(root,file)

change()

w.destroy()

btn1=Button(w,image=wall1,text="Default wallpaper",command=lambda:wall\_change(wallad),compound=TOP)

btn1.grid(row=0,column=1)

btn2=Button(w,image=wall2,text="Light fall",command=lambda:wall\_change(wall2ad),compound=TOP)

btn2.grid(row=0,column=2)

btn3=Button(w,image=wall3,text="Blue Texture",command=lambda:wall\_change(wall3ad),compound=TOP)

btn3.grid(row=0,column=3)

def open\_file():

openfile=filedialog.askopenfilename(initialdir="D:\Class 11\CSC\PYTHON\Images",title="Open file",filetypes=(("Image files","\*.png"),("all files","\*.\*")))

my\_canvas.destroy()

canvas(root,openfile)

change()

w.destroy()

btn4=Button(w,text="Choose a file from your computer",command=open\_file,padx=20,pady=20)

btn4.grid(row=1,column=2)

def dark\_mode():

my\_canvas.destroy()

canvas(root,r"Images\dark\_mode.jpg")

change()

def profile():#Edir profile

p=Toplevel(root)

p.geometry("375x500")

p.configure(bg="#00F7FF")

p.title("Profile")

def pic():

new=Toplevel(p)

def upload():

openfile1=filedialog.askopenfilename(initialdir=r"D:\Class 11\CSC\PYTHON\Images",title="Open file",filetypes=(("Image files","\*.png"),("all files","\*.\*")))

qw=upload\_new(openfile1)

us.config(image=qw)

new.destroy()

def remove():

us.config(image=empty)

new.destroy()

Button(new,text="Upload new profile pic",command=upload,bg="#FF0000").pack()

Button(new,text="Remove profile pic",command=remove,bg="#3DFF00").pack()

def apply():

global pro

mo=[]

for i in ty:

mo.append(ty[i].get())

c.execute(f"update user set username='{mo[1]}',password='{mo[2]}',age='{mo[3]}',contact='{mo[4]}',address='{mo[5]}' where name='{mo[0]}';")

m.commit()

messagebox.showinfo('INFO BY INSIDEX','You have successfully updated your profile!')

c.execute("select \* from user;")

r=c.fetchall()

for i in r:

if i[0]==pro[0]:

pro=i

p.destroy()

global pro

Label(p,text="My Profile",font="Castellar 20",bg="#00FF00").grid(row=0,column=6)

us=Button(p,image=user\_profile,command=pic)

us.grid(row=1,column=6)

Label(p,text="Name",font="timesnewroman 15").grid(row=2,column=5,padx=10,pady=10)

def primary(e):

messagebox.showerror('ERROR BY INSIDEX!',"Name(primary key) can't be changed")

b=Entry(p)

b.insert(0,pro[0])

b.config(state="disabled")

b.grid(row=2,column=6)

Label(p,text="Username",font="timesnewroman 15").grid(row=3,column=5,padx=10,pady=10)

ci=Entry(p)

ci.insert(0,pro[1])

ci.grid(row=3,column=6)

Label(p,text="Password",font="timesnewroman 15").grid(row=4,column=5,padx=10,pady=10)

d=Entry(p)

d.insert(0,pro[2])

d.grid(row=4,column=6)

Label(p,text="Age",font="timesnewroman 15").grid(row=5,column=5,padx=10,pady=10)

e=Entry(p)

e.insert(0,pro[3])

e.grid(row=5,column=6)

Label(p,text="Phone number",font="timesnewroman 15").grid(row=6,column=5,padx=10,pady=10)

f=Entry(p)

f.insert(0,pro[4])

f.grid(row=6,column=6)

Label(p,text="Address",font="timesnewroman 15").grid(row=7,column=5,padx=10,pady=10)

g=Entry(p)

g.insert(0,pro[5])

g.grid(row=7,column=6)

Button(p,text="Apply changes",bg="#FFA500",font="Algeria 15",command=apply).grid(row=8,column=6)

po=['name','user','pass','age','phone','addr']

ko=[b,ci,d,e,f,g]

ty={}

for i in range(len(po)):

ty[po[i]]=ko[i]

my\_menu=Menu(root)

root.config(menu=my\_menu)

profile\_menu=Menu(my\_menu,tearoff=False)

my\_menu.add\_cascade(label="Profile",menu=profile\_menu)

profile\_menu.add\_command(label="Edit your account",command=profile)

edit\_menu=Menu(my\_menu,tearoff=False)

my\_menu.add\_cascade(label="Edit",menu=edit\_menu)

edit\_menu.add\_command(label="Change wallpaper",command=wallpaper)

edit\_menu.add\_command(label="Dark mode",command=dark\_mode)

edit\_menu.add\_command(label="Language")

help\_menu=Menu(my\_menu,tearoff=False)

my\_menu.add\_cascade(label="Help",menu=help\_menu)

help\_menu.add\_cascade(label="Instructions to use the app")

change()

def back3(k):

if k=="login":

my\_canvas.destroy()

home()

elif k=="signup":

frame4.place\_forget()

login()

def signup():

global frame4,adrin,ko,userin,passwordin

frame2.place\_forget()

Button(my\_canvas,image=back\_btn,bg='#00ddff',command=lambda:back3("signup")).place(x=0,y=0)

frame4=Frame(my\_canvas,width=600,height=500,bg='#00ddff',highlightbackground='#ff00d4',highlightthickness=3)

frame4.place(x=650,y=100)

Label(frame4,text="USER SIGN-UP PAGE",font=("Castellar",30,"bold"),bg="#ee2a7b",fg="white").place(x=100,y=5)

typ=["Name:","Username:","Password:","Confirm Password:","Age:","Phone number:","Address:"]

y2=100

for i in typ:

Label(frame4,text=i).place(x=50,y=y2)

y2+=40

def on\_enter(e):

namein.delete(0,'end')

def on\_leave(e):

if namein.get()=='':

namein.insert(0,'Enter your name')

namein=Entry(frame4,width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

namein.place(x=130,y=100)

namein.insert(0,'Enter your name')

namein.bind('<FocusIn>',on\_enter)

namein.bind('<FocusOut>',on\_leave)

Frame(frame4,width=200,height=2,bg='black').place(x=120,y=125)

def on\_enter(e):

userin.delete(0,'end')

def on\_leave(e):

if userin.get()=='':

userin.insert(0,'Enter your username')

userin=Entry(frame4,width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

userin.place(x=130,y=140)

userin.insert(0,'Enter your username')

userin.bind('<FocusIn>',on\_enter)

userin.bind('<FocusOut>',on\_leave)

Frame(frame4,width=200,height=2,bg='black').place(x=120,y=165)

passwordin = Entry(frame4, show='\*',width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

passwordin.place(x=130,y=180)

Frame(frame4,width=150,height=2,bg='black').place(x=120,y=200)

def toggle\_password():

if passwordin.cget('show') == '':

passwordin.config(show='\*')

toggle\_btn.config(text='Show Password')

else:

passwordin.config(show='')

toggle\_btn.config(text='Hide Password')

toggle\_btn = Button(frame4, text='Show Password', width=15, command=toggle\_password)

toggle\_btn.place(x=300,y=180)

copasswordin = Entry(frame4,show='\*',width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

copasswordin.place(x=170,y=220)

Frame(frame4,width=150,height=2,bg='black').place(x=160,y=240)

def cotoggle\_password():

if copasswordin.cget('show') == '':

copasswordin.config(show='\*')

cotoggle\_btn.config(text='Show Password')

else:

copasswordin.config(show='')

cotoggle\_btn.config(text='Hide Password')

cotoggle\_btn = Button(frame4, text='Show Password', width=15, command=cotoggle\_password)

cotoggle\_btn.place(x=350,y=220)

def on\_enter(e):

agein.delete(0,'end')

def on\_leave(e):

if agein.get()=='':

agein.insert(0,'Enter your age')

agein=Entry(frame4,width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

agein.place(x=130,y=260)

agein.insert(0,'Enter your age')

agein.bind('<FocusIn>',on\_enter)

agein.bind('<FocusOut>',on\_leave)

Frame(frame4,width=200,height=2,bg='black').place(x=120,y=280)

def on\_enter(e):

phonein.delete(0,'end')

def on\_leave(e):

if phonein.get()=='':

phonein.insert(0,'Enter your phone number')

phonein=Entry(frame4,width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

phonein.place(x=180,y=300)

phonein.insert(0,'Enter your phone number')

phonein.bind('<FocusIn>',on\_enter)

phonein.bind('<FocusOut>',on\_leave)

Frame(frame4,width=200,height=2,bg='black').place(x=170,y=320)

def on\_enter(e):

adrin.delete(0,'end')

def on\_leave(e):

if adrin.get()=='':

adrin.insert(0,'Enter your address')

adrin=Entry(frame4,width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

adrin.place(x=180,y=340)

adrin.insert(0,'Enter your address')

adrin.bind('<FocusIn>',on\_enter)

adrin.bind('<FocusOut>',on\_leave)

Frame(frame4,width=200,height=2,bg='black').place(x=170,y=360)

p=['name','user','pass','copass','age','phone','addr']

ko=[namein,userin,passwordin,copasswordin,agein,phonein,adrin]

all={}

for i in range(len(p)):

all[p[i]]=ko[i]

def aftersignup():

mo=[]

for i in all:

mo.append(all[i].get())

#print(mo)

my\_flag=False

for i in ko:

if i.get() =='':

my\_flag=True

if 'Enter your ' in i.get():

i.delete(0,"end")

if my\_flag==False:

frame4.place\_forget()

#c.execute('create table user (Name varchar(20) primary key,Username varchar(10),Password varchar(10),Age char(2),Contact varchar(20),Address varchar(30));')

c.execute('insert into user values(%s,%s,%s,%s,%s,%s)',(namein.get(),userin.get(),passwordin.get(),agein.get(),phonein.get(),adrin.get()))

m.commit()

-------------------#SMS SENDING#------------------------

service\_plan\_id="ef75d15c58274ffab6aeb4ff72273f02"

access\_token="6ee1684e2857449ca1cd4fa7e57c5ec2"

from\_="447520651115"

to="91"+str(phonein.get())

headers={

"Authorization":f"Bearer {access\_token}",

"Content-Type":"application/json"

}

payload={

"from":from\_,

"to":[to],

"body":"Hello "+str(namein.get())+' ! You have registered into INSIDEX platform.'+

' Start your beautiful journey with INSIDEX right now!!!'

}

try:

re.post(

f'https://sms.api.sinch.com/xms/v1/{service\_plan\_id}/batches',

headers=headers,

data=json.dumps(payload)

)

except:

messagebox.showerror('WARNING BY INSIDEX','Phone number given is invalid!')

frame2.place(x=700,y=100)

elif my\_flag==True:

count=0

for i in ko:

if i.get() =='':

count+=1

if count:

messagebox.showerror('WARNING BY INSIDEX','Fill all the entries to sign-up!')

c.execute('select \* from user;')

r=c.fetchall()

for i in r:

if namein.get() in i:

messagebox.showerror('WARNING BY INSIDEX','Name already exists!')

if userin.get() in i:

messagebox.showerror('WARNING BY INSIDEX','Username already exists!')

if agein.get().isdigit()== False:

messagebox.showerror('WARNING BY INSIDEX','Age must be a integer!')

if len(int(phonein.get()))!=10:

messagebox.showerror('WARNING BY INSIDEX','Given contact number is not valid')

if copasswordin.get()!=passwordin.get():

messagebox.showerror('WARNING BY INSIDEX','Password and confirm password must be given the same!')

Button(frame4,image=signup\_btn,bg="#00FFFF",command=aftersignup,borderwidth=0).place(x=300,y=400)

def login():

global frame2,un,pw

my\_canvas.destroy()

canvas(root,r'Images\userlogin.jpg')

Button(my\_canvas,image=back\_btn,bg='#00ddff',command=lambda:back3("login")).place(x=0,y=0)

frame2=Frame(my\_canvas,width=500,height=400,bg='#00ddff',highlightbackground='#ff00d4',highlightthickness=3)

lab=Label(frame2,text="USER PAGE",font=("Castellar",30,"bold"),bg="#ee2a7b",fg="white")

lab.place(x=100,y=5)

def on\_enter(e):

un.delete(0,'end')

def on\_leave(e):

if un.get()=='':

un.insert(0,'Username')

un=Entry(frame2,width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

un.place(x=30,y=80)

un.insert(0,'Username')

un.bind('<FocusIn>',on\_enter)

un.bind('<FocusOut>',on\_leave)

Frame(frame2,width=295,height=2,bg='black').place(x=25,y=107)

def on\_enter(e):

pw.delete(0,'end')

pw.config(show='\*')

def on\_leave(e):

if pw.get()=='':

pw.insert(0,'Password')

pw=Entry(frame2,width=25,fg='black',bg='#00ddff',border=0,font=('Microsoft Yahei UI Light',11))

pw.place(x=30,y=130)

pw.insert(0,'Password')

pw.bind('<FocusIn>',on\_enter)

pw.bind('<FocusOut>',on\_leave)

Frame(frame2,width=295,height=2,bg='black').place(x=25,y=157)

def toggle\_password():

if pw.cget('show') == '':

pw.config(show='\*')

toggle\_btn.config(text='Show Password')

else:

pw.config(show='')

toggle\_btn.config(text='Hide Password')

toggle\_btn =Button(frame2, text='Show Password',command=toggle\_password,bg='#00ddff')

toggle\_btn.place(x=350,y=137)

Button(frame2,image=login\_btn,command=afterlogin,borderwidth=0).place(x=50,y=200)

Label(frame2,text='(or)').place(x=100,y=250)

Label(frame2,text='Are you a new user to our platform?').place(x=50,y=280)

Button(frame2,image=signup\_btn,command=signup,borderwidth=0).place(x=50,y=320)

frame2.place(x=700,y=100)

login()

def home():#Home page

canvas(root,r"Images\background2.jpg")

f=Frame(my\_canvas)

f.configure(bg="#f01e2c")

Label(f,text="WELCOME TO INSIDEX",font=("Castellar",30,"bold"),bg="#2596be",fg="white").grid(row=0,column=1)

logolab=Label(f,image=new\_pic)

logolab.grid(row=1,column=1,padx=50,pady=50)

Button(f,text="User",bg="#2596be",font="Times 15",fg="white",height=2,border=2,command=user).grid(row=3,column=1,sticky=NSEW,padx=10,pady=10)

Button(f,text="Admin",bg="#2596be",font="Times 15",fg="white",height=2,border=2,command=admin).grid(row=4,column=1,sticky=NSEW,padx=10,pady=10)

f.place(x=730,y=250)

home()

def confirm():

ans=askyesno(title='Exit',message='Do you want to exit?')

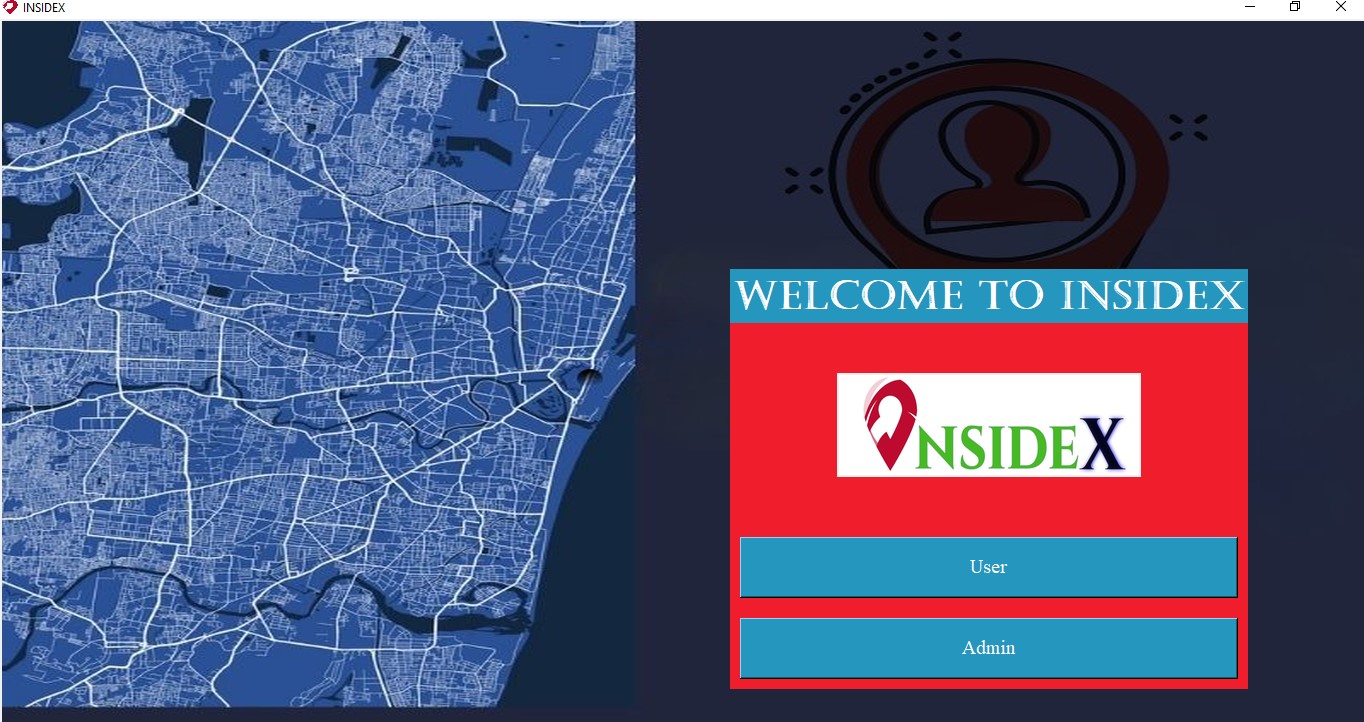
if ans:

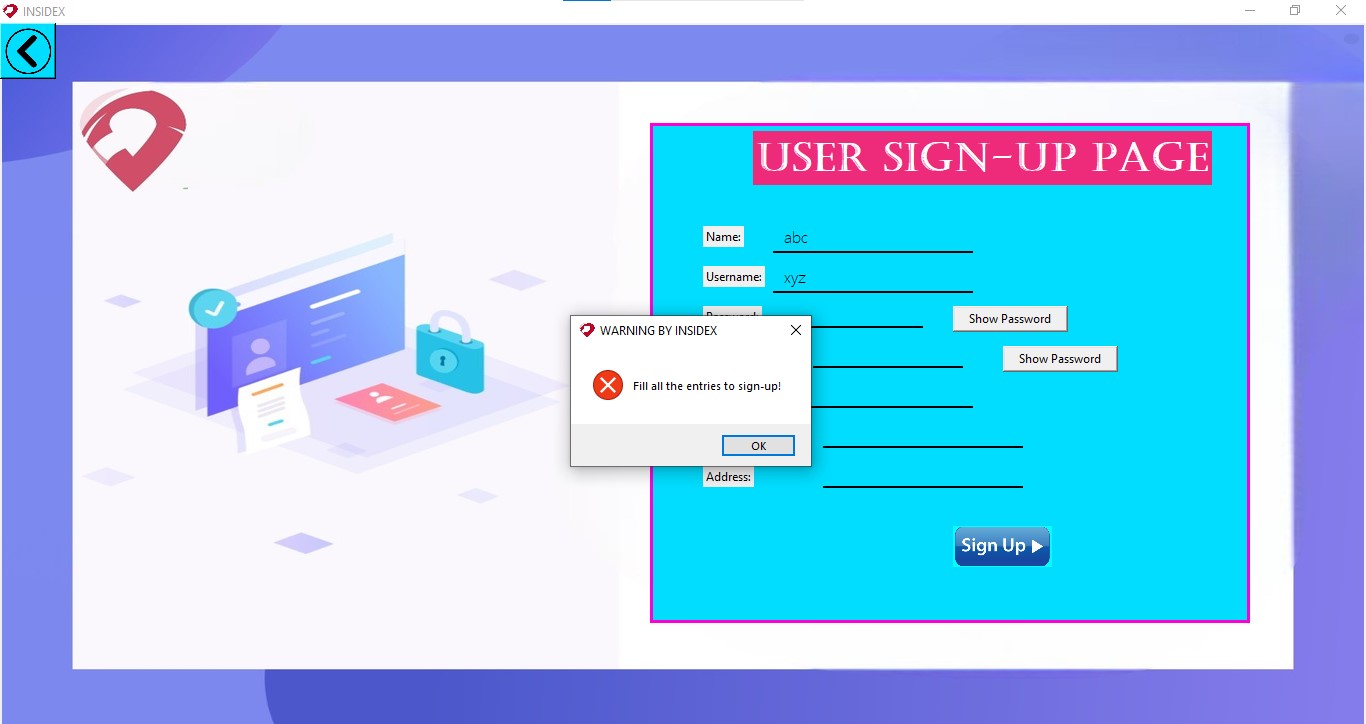
root.destroy()

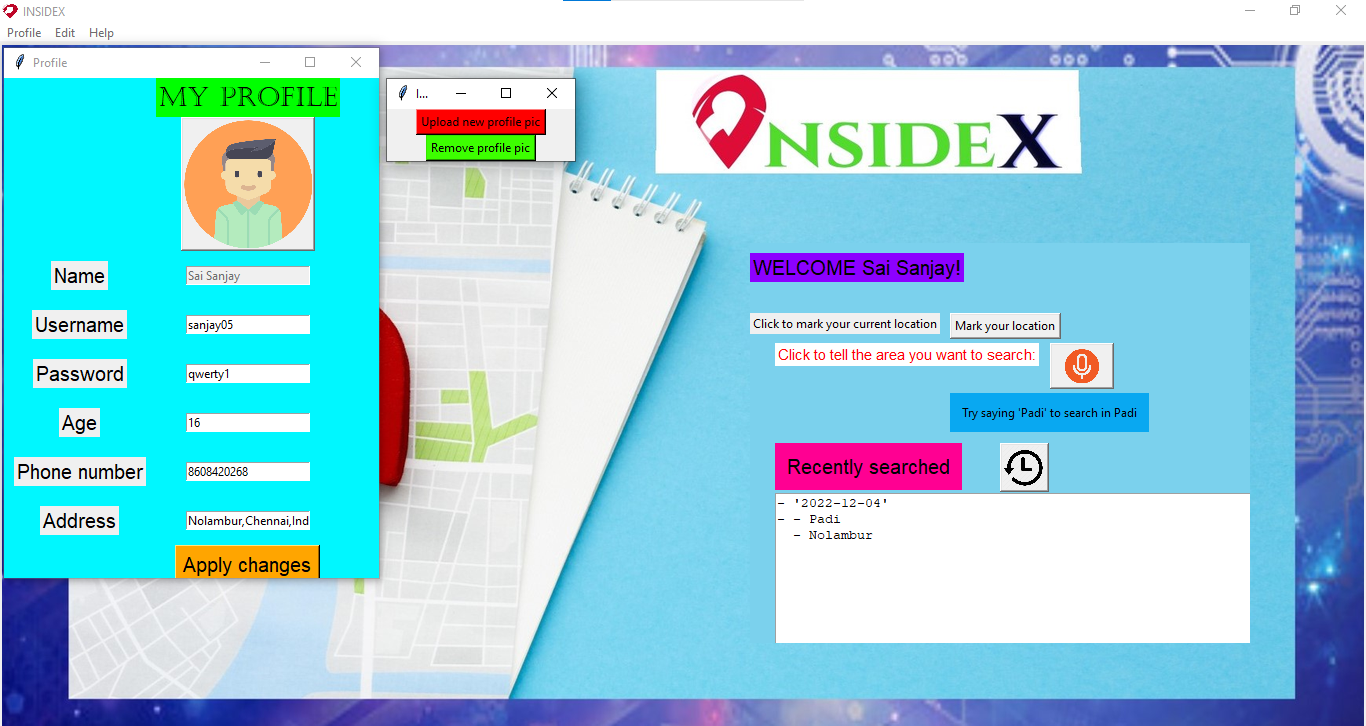
root.protocol("WM\_DELETE\_WINDOW",confirm)

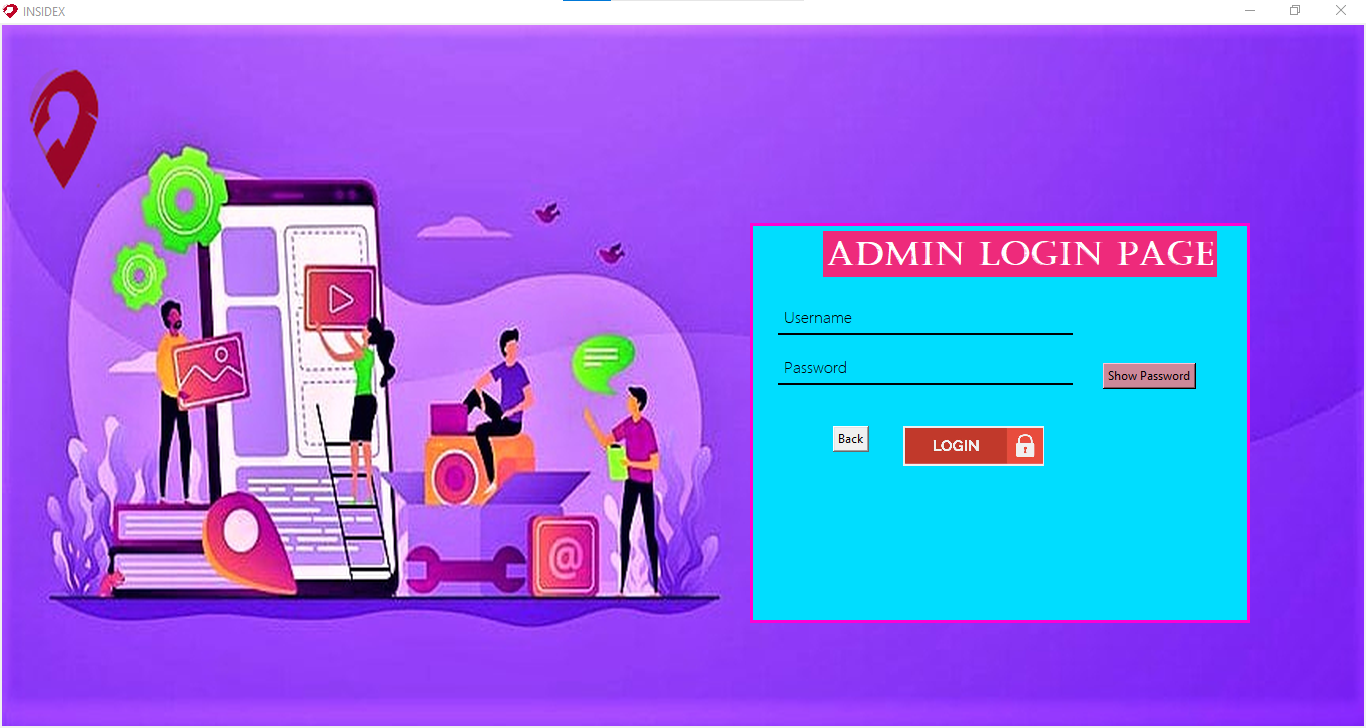
root.mainloop()

**OUTPUT:**

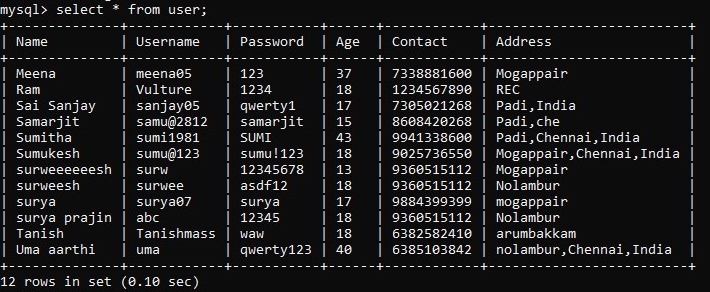
****

****

****



**DATABASE:**

****

**CONCLUSION:**

INSIDEX is a robust and innovative solution for navigating nearby facilities, offering both desktop and web-based platforms for maximum accessibility. By integrating a user-friendly interface, dynamic facility recommendations, and advanced features like speech recognition and real-time mapping, the project successfully addresses the challenges of fragmented and inefficient local navigation systems.

The use of modern technologies, including **Tkinter** for the desktop app, **Python** for backend logic, and **MySQL** for secure and efficient data storage, ensures a seamless and scalable user experience.

The project's ability to provide personalized facility recommendations, track user activity, and empower admins to manage facility data dynamically highlights its practicality and potential for real-world applications. INSIDEX not only simplifies the discovery process for users but also fosters efficient management, making it a valuable tool for both individuals and businesses.

In conclusion, INSIDEX showcases a thoughtful blend of technology and innovation, promising significant impacts in localized navigation and user engagement. It stands as a testament to the power of integrating intuitive design, structured databases, and modern backend solutions to solve real-world problems effectively.

**REFERENCES:**

1. **W3Schools**: A comprehensive platform offering tutorials on HTML, SQL, PHP, and other web technologies. It includes interactive coding exercises and examples, making it a great starting point for beginners. [Visit W3Schools](https://www.w3schools.com/)
2. **Codemy youtube channel:** A playlist in this channel taught us how to use tkinter and its modules and functions.

<https://www.youtube.com/@Codemycom>

1. Computer Science with python Textbook for Class 12 Examination 2022-2023 Textbook by Sumita Arora