**Mr. Unknown**

**Team members:**

**SAI NIKHIL P-181CV237(14)**

**ANANT RAGHOTHAM ARNI-181MT007(46)**

**GAUTHAMKRISHNA S-181ME125(12)**

***Python code with SQLite queries are given below*:**

**MAIN FILE: This file contains the main interface and this file will be executed.**

**Actual file name: main.py**

from tkinter import \*  
from userinfo import addurdetails  
from addcontacts import addcontacts  
from contactinfo import viewcontact  
from allcontact import allcontact  
from spam import spam  
import datetime  
date=datetime.datetime.now().date()  
date=str(date)  
class application(object):  
 def \_\_init\_\_(self,master):  
 self.master = master  
 self.top = Frame(master, height=150, bg=**'white'**)  
 self.top.pack(fill=X)  
 self.bottom = Frame(master, height=500, bg=**'black'**)  
 self.bottom.pack(fill=X)  
 *#self.top\_image = PhotoImage(file='icons/book.png')  
 #self.top\_image\_label = Label(self.top, image=self.top\_image)  
 #self.top\_image\_label.place(x=300, y=30)* self.heading = Label(self.top, text=**'Mr.Unknown'**, font=**'arial 15 bold'**, bg=**'white'**)  
 self.heading.place(x=280, y=100)  
 self.date\_lbl = Label(self.top, text=**"today's date"** + date, font=**'arial 10 bold'**, bg=**'white'**)  
 self.date\_lbl.place(x=500, y=120)  
 *# user button* self.userbutton = Button(self.bottom, text=**"add your details"**, font=**'arial 12 bold'**, command=self.userinfo)  
 self.userbutton.place(x=270, y=45)  
 *# add contacts* self.localcontactsbutton = Button(self.bottom, text=**'Local contacts'**, font=**'arial 12 bold'**,command=self.addcontacts)  
 self.localcontactsbutton.place(x=275, y=100)  
 *# allcontacts* self.allcontactsbutton = Button(self.bottom, text=**'Find Mr. Unknown'**, font=**'arial 12 bold'**,command=self.allcontact)  
 self.allcontactsbutton.place(x=285, y=155)  
 *# search contact info  
 # report spam button* self.spambutton = Button(self.bottom, text=**'report spam'**, font=**'arial 12 bold'**, command=self.spam)  
 self.spambutton.place(x=290, y=210)  
  
 def userinfo(self):  
 user = addurdetails()  
  
 def addcontacts(self):  
 contacts = addcontacts()  
  
 def contactinfo(self):  
 contactinformation = viewcontact()  
  
 def spam(self):  
 spamcontact = spam()  
  
 def allcontact(self):  
 addcontac = allcontact()  
  
def main():  
 root = Tk()  
 app=application(root)  
 root.title(**"truecaller"**)  
 root.geometry(**"650x550+350+200"**)  
 root.resizable(False,False)  
 root.mainloop()  
if \_\_name\_\_==**'\_\_main\_\_'**:  
 main()

**Code dealing with “FIND MR. UNKOWN” functionality or finding details about unknown number is given below:**

**Actual file name: allcontact.py**

from tkinter import \*  
import sqlite3  
con=sqlite3.connect(**'directory.db'**)  
cur=con.cursor()  
from tkinter import messagebox  
class allcontact(Toplevel):  
 def \_\_init\_\_(self):  
 Toplevel.\_\_init\_\_(self)  
 self.geometry(**"650x650+600+200"**)  
 self.title(**"all contacts"**)  
 self.resizable(False, False)  
 self.top = Frame(self, height=150, bg=**'white'**)  
 self.top.pack(fill=X)  
 self.bottom = Frame(self, height=500, bg=**'black'**)  
 self.bottom.pack(fill=X)  
 self.heading = Label(self.top, text=**'all contacts information'**, font=**'arial 15 bold'**, bg=**'white'**)  
 self.heading.place(x=260, y=100)  
 self.label\_num = Label(self.bottom, text=**'number'**, font=**'arial 12 bold'**)  
 self.label\_num.place(x=160, y=40)  
 self.entry\_num = Entry(self.bottom, width=30, bd=4)  
 self.entry\_num.insert(0, **"enter numner"**)  
 self.entry\_num.place(x=250, y=40)  
 *#self.listbox = Listbox(self.bottom, width=30, height=20)  
 #self.listbox.place(x=250, y=200)* button = Button(self.bottom, text=**'search in all contacts'**, font=**'arial 12 bold'**, command=self.search)  
 button.place(x=280, y=120)  
  
 def search(self):  
 num = self.entry\_num.get()  
 query = (**'SELECT username,mobilenumber,gender FROM userinfom where mobilenumber=?'**)  
 cur.execute(query, (num,))  
 c = cur.fetchone()  
 if c is None:  
 messagebox.showerror(**"error"**, **"no such number exists"**, icon=**'warning'**)  
 else:  
 print(c)

**Code dealing with “SPAM” functionality is given below**

**Actual file name: spam.py**

from tkinter import \*  
import sqlite3  
con=sqlite3.connect(**'directory.db'**)  
cur=con.cursor()  
from removespam import deletespam  
from tkinter import messagebox  
class spam(Toplevel):  
 def \_\_init\_\_(self):  
 Toplevel.\_\_init\_\_(self)  
 self.geometry(**"650x650+600+200"**)  
 self.title(**"add your details"**)  
 self.resizable(False, False)  
 self.top = Frame(self, height=150, bg=**'white'**)  
 self.top.pack(fill=X)  
 self.bottom = Frame(self, height=500, bg=**'black'**)  
 self.bottom.pack(fill=X)  
 self.label = Label(self.bottom, text=**"Enter mobile no."**, font=**'arial 12 bold'**)  
 self.label.place(x=100, y=20)  
 self.mobile\_no = Entry(self.bottom, width=30, bd=4)  
 self.mobile\_no.place(x=250, y=20)  
 self.button = Button(self.bottom, text=**'Add to spam'**, font=**'arial 12 bold'**, command=self.add)  
 self.button.place(x=250, y=50)  
 self.label = Label(self.bottom, text=**"OR"**, font=**'arial 12 bold'**)  
 self.label.place(x=250, y=90)  
 self.button1 = Button(self.bottom, text=**"VIEW ALL SPAM"**, font=**'arial 12 bold'**, command=self.view)  
 self.button1.place(x=250, y=120)  
 self.button2 = Button(self.bottom, text=**"DELETE FROM SPAM"**, font=**'arial 12 bold'**,  
 command=self.removespam)  
 self.button2.place(x=150, y=160)  
 self.listbox = Listbox(self.bottom, width=85, height=30)  
 self.listbox.place(x=180, y=200)  
 self.button3=Button(self.bottom,text=**'TOP SPAM'**,command=self.top\_spam)  
 self.button3.place(x=500,y=160)  
  
  
 cur.execute(**'CREATE TABLE IF NOT EXISTS spam(ph\_no INTEGER,count INTEGER ,FOREIGN KEY(ph\_no) REFERENCES userinfom(mobilenumber))'**)  
  
 def add(self):  
 cur.execute(**'SELECT ph\_no from spam WHERE ph\_no=?'**, (self.mobile\_no.get(),))  
 q = cur.fetchone()  
 if q is None:  
 q1 = **"INSERT INTO spam(ph\_no,count) values(?,1)"** cur.execute(q1, (self.mobile\_no.get(),))  
 con.commit()  
 else:  
 cur.execute(**'UPDATE spam SET count=count+1 WHERE ph\_no=?'**, (self.mobile\_no.get(),))  
 con.commit()  
  
 messagebox.showinfo(**"success"**, **"number reported to spam successfully"**)  
  
 def view(self):  
 q = **'SELECT username,mobilenumber,gender,emailid,city from userinfom INNER JOIN spam ON userinfom.mobilenumber =spam.ph\_no'** cur.execute(q)  
 qq = cur.fetchall()  
 count=0  
 for x in qq:  
 self.listbox.insert(count,x[0]+**" "**+x[1]+**" "**+x[2]+**" "**+x[3]+**" "**+x[4])  
 count=count+1  
 def top\_spam(self):  
 query=**'SELECT \* FROM spam ORDER BY count DESC'** pp=cur.execute(query)  
 count=0  
 for z in pp:  
 self.listbox.insert(count,str(z[0])+**" "**+str(z[1])+**" PEOPLE REPORTED AS SPAM"**)  
 count=count+1  
  
  
  
 def removespam(self):  
 delspam = deletespam()

**Code dealing with “REMOVE SPAM” functionality is given below:**

**Actual file name: removespam.py**

from tkinter import \*  
import sqlite3  
con=sqlite3.connect(**'directory.db'**)  
cur=con.cursor()  
from tkinter import messagebox  
class deletespam(Toplevel):  
 def \_\_init\_\_(self):  
 Toplevel.\_\_init\_\_(self)  
 self.geometry(**"650x650+600+200"**)  
 self.title(**"delete from spam"**)  
 self.resizable(False, False)  
 self.top = Frame(self, height=150, bg=**'white'**)  
 self.top.pack(fill=X)  
 self.bottom = Frame(self, height=500, bg=**'black'**)  
 self.bottom.pack(fill=X)  
 self.label\_num = Label(self.bottom, text=**'number'**, font=**'arial 12 bold'**)  
 self.label\_num.place(x=160, y=40)  
 self.entry\_num = Entry(self.bottom, width=30, bd=4)  
 self.entry\_num.insert(0, **" "**)  
 self.entry\_num.place(x=250, y=40)  
 button = Button(self.bottom, text=**'delete from spam'**, font=**'arial 12 bold'**, command=self.delte)  
 button.place(x=280, y=100)  
  
 def delte(self):  
 num = self.entry\_num.get()  
 query = **"SELECT \* FROM spam WHERE ph\_no=?"** cur.execute(query, (num,))  
 q = cur.fetchone()  
 if q is None:  
 messagebox.showinfo(**"failure"**, **"no such number reported"**)  
 else:  
 b = **"DELETE FROM spam WHERE ph\_no=?"** cur.execute(b, (num,))  
 con.commit()  
 messagebox.showinfo(**"success"**, **"deleted successfully"**)

**Code dealing with “ADD CONTACT” functionality is given below :**

**Actual file name: addcontacts.py**

from tkinter import \*  
from deletecon import deletecontacts  
import sqlite3  
from tkinter import messagebox  
con=sqlite3.connect(**'directory.db'**)  
cur=con.cursor()  
class addcontacts(Toplevel):  
 def \_\_init\_\_(self):  
 Toplevel.\_\_init\_\_(self)  
 self.geometry(**"650x650+600+200"**)  
 self.title(**"add your contacts"**)  
 self.resizable(False, False)  
 self.top = Frame(self, height=150, bg=**'white'**)  
 self.top.pack(fill=X)  
 self.bottom = Frame(self, height=500, bg=**'black'**)  
 self.bottom.pack(fill=X)  
 *#self.top\_image = PhotoImage(file='icons/contact.png')  
 #self.top\_image\_label = Label(self.top, image=self.top\_image)  
 #self.top\_image\_label.place(x=300, y=30)* self.heading = Label(self.top, text=**'my contacts'**, font=**'arial 15 bold'**, bg=**'white'**)  
 self.heading.place(x=260, y=100)  
 self.label\_contactname = Label(self.bottom, text=**'contact name'**, font=**'arial 12 bold'**)  
 self.label\_contactname.place(x=160, y=80)  
 self.entry\_contactname = Entry(self.bottom, width=30, bd=4)  
 self.entry\_contactname.insert(0, **" "**)  
 self.entry\_contactname.place(x=280, y=80)  
 self.label\_contactnum = Label(self.bottom, text=**'contact number'**, font=**'arial 12 bold'**)  
 self.label\_contactnum.place(x=140, y=120)  
 self.entry\_contactnum = Entry(self.bottom, width=30, bd=4)  
 self.entry\_contactnum.insert(0, **" "**)  
 self.entry\_contactnum.place(x=280, y=120)  
 self.label\_contactprof = Label(self.bottom, text=**'contact profession'**, font=**'arial 12 bold'**)  
 self.label\_contactprof.place(x=120, y=160)  
 self.entry\_contactprof = Entry(self.bottom, width=30, bd=4)  
 self.entry\_contactprof.insert(0, **" "**)  
 self.entry\_contactprof.place(x=280, y=160)  
 self.label\_contactloc = Label(self.bottom, text=**'contact location'**, font=**'arial 12 bold'**)  
 self.label\_contactloc.place(x=140, y=200)  
 self.entry\_contactloc = Entry(self.bottom, width=30, bd=4)  
 self.entry\_contactloc.insert(0, **" "**)  
 self.entry\_contactloc.place(x=280, y=200)  
 self.button = Button(self.bottom, text=**'add contacts'**, font=**'arial 12 bold'**, command=self.add\_contacts)  
 self.button.place(x=20, y=250)  
 self.button1 = Button(self.bottom, text=**'display contacts'**, font=**'arial 12 bold'**, command=self.disp\_contacts)  
 self.button1.place(x=140, y=250)  
 self.spambutton = Button(self.bottom, text=**'delete contacts'**, font=**'arial 12 bold'**,command=self.deletecon)  
 self.spambutton.place(x=300, y=250)  
 self.listbox = Listbox(self.bottom, width=80, height=40)  
 self.listbox.place(x=120, y=300)  
  
 cur.execute(**'CREATE TABLE IF NOT EXISTS contacts(contactname TEXT NOT NULL,contactnumber TEXT PRIMARY KEY,contactprof TEXT NOT NULL,contactloc TEXT NOT NULL)'**)  
  
 def add\_contacts(self):  
 contactname = self.entry\_contactname.get()  
 contactnum = self.entry\_contactnum.get()  
 contactprof = self.entry\_contactprof.get()  
 contactloc = self.entry\_contactloc.get()  
 if contactname and contactnum and contactprof and contactloc != **""**:  
 *#try:* query = **"INSERT INTO contacts (contactname,contactnumber,contactprof,contactloc) values(?, ?, ?, ?)"** cur.execute(query, (contactname, contactnum, contactprof, contactloc))  
 con.commit()  
 *#d = cur.fetchall()  
 #print(d)* messagebox.showinfo(**"success"**, **"contact added"**)  
 *#except EXCEPTION as e:* else:  
 messagebox.showerror(**"error"**, **"fill all fields"**, icon=**'warning'**)  
  
 def disp\_contacts(self):  
 self.listbox.delete(0,END)  
 query = (**'SELECT \* FROM contacts'**)  
 cur.execute(query)  
 c = cur.fetchall()  
 count=0  
 for x in c:  
 self.listbox.insert(count,x[0]+**" "**+x[1]+**" "**+x[2]+**" "**+x[3])  
 count=count+1  
  
  
 def deletecon(self):  
 deletcontact = deletecontacts()

**Code dealing with “DELETE CONTACT” functionality is given below:**

**Actual file name: deletecon.py**

from tkinter import \*  
import sqlite3  
con=sqlite3.connect(**'directory.db'**)  
cur=con.cursor()  
from tkinter import messagebox  
class deletecontacts(Toplevel):  
 def \_\_init\_\_(self):  
 Toplevel.\_\_init\_\_(self)  
 self.geometry(**"650x650+600+200"**)  
 self.title(**"delete contact"**)  
 self.resizable(False, False)  
 self.top = Frame(self, height=150, bg=**'white'**)  
 self.top.pack(fill=X)  
 self.bottom = Frame(self, height=500, bg=**'black'**)  
 self.bottom.pack(fill=X)  
 self.label\_num = Label(self.bottom, text=**'number'**, font=**'arial 12 bold'**)  
 self.label\_num.place(x=160, y=40)  
 self.entry\_num = Entry(self.bottom, width=30, bd=4)  
 self.entry\_num.insert(0, **""**)  
 self.entry\_num.place(x=250, y=40)  
 button = Button(self.bottom, text=**'delete'**, font=**'arial 12 bold'**, command=self.delete)  
 button.place(x=280, y=100)  
  
 def delete(self):  
 num = self.entry\_num.get()  
 query = **"SELECT \* FROM contacts WHERE contactnumber=?"** cur.execute(query, (num,))  
 q = cur.fetchone()  
 if q is None:  
 messagebox.showinfo(**"failure"**, **"no contact as such exists"**)  
 else:  
 cur.execute(**'DELETE FROM contacts WHERE contactnumber=?'**, (num,))  
 con.commit()  
 messagebox.showinfo(**"success"**, **"deleted successfully"**)

**Code dealing with “USER DETAILS” functionality is given below:**

**Actual file name: userinfo.py**

from tkinter import \*  
import sqlite3  
from tkinter import messagebox  
con=sqlite3.connect(**'directory.db'**)  
cur=con.cursor()  
class addurdetails(Toplevel):  
 def \_\_init\_\_(self):  
 Toplevel.\_\_init\_\_(self)  
 self.geometry(**"650x650+600+200"**)  
 self.title(**"add your details"**)  
 self.resizable(False, False)  
 self.top = Frame(self, height=150, bg=**'white'**)  
 self.top.pack(fill=X)  
 self.bottom = Frame(self, height=500, bg=**'black'**)  
 self.bottom.pack(fill=X)  
 *self.top\_image = PhotoImage(file='icons/people.png')  
 self.top\_image\_label = Label(self.top, image=self.top\_image)  
 self.top\_image\_label.place(x=300, y=30)* self.heading = Label(self.top, text=**'add your details'**, font=**'arial 15 bold'**, bg=**'white'**)  
 self.heading.place(x=260, y=100)  
 self.label\_name = Label(self.bottom, text=**'name'**, font=**'arial 12 bold'**)  
 self.label\_name.place(x=160, y=40)  
 self.entry\_name = Entry(self.bottom, width=30, bd=4)  
 self.entry\_name.insert(0, **""**)  
 self.entry\_name.place(x=250, y=40)  
 self.label\_mobilenumber = Label(self.bottom, text=**'mobile number'**, font=**'arial 12 bold'**)  
 self.label\_mobilenumber.place(x=100, y=80)  
 self.entry\_mobilenumber = Entry(self.bottom, width=30, bd=4)  
 self.entry\_mobilenumber.insert(0, **""**)  
 self.entry\_mobilenumber.place(x=250, y=80)  
 self.label\_gender = Label(self.bottom, text=**'gender'**, font=**'arial 12 bold'**)  
 self.label\_gender.place(x=160, y=120)  
 self.entry\_gender = Entry(self.bottom, width=30, bd=4)  
 self.entry\_gender.insert(0, **""**)  
 self.entry\_gender.place(x=250, y=120)  
 self.label\_email = Label(self.bottom, text=**'emailid'**, font=**'arial 12 bold'**)  
 self.label\_email.place(x=160, y=160)  
 self.entry\_email = Entry(self.bottom, width=30, bd=4)  
 self.entry\_email.insert(0, **""**)  
 self.entry\_email.place(x=250, y=160)  
 self.label\_city = Label(self.bottom, text=**'city'**, font=**'arial 12 bold'**)  
 self.label\_city.place(x=160, y=200)  
 self.entry\_city = Entry(self.bottom, width=30, bd=4)  
 self.entry\_city.place(x=250, y=200)  
 button = Button(self.bottom, text=**'add details'**, font=**'arial 12 bold'**, command=self.add\_details)  
 button.place(x=280, y=300)  
  
 cur.execute(**'CREATE TABLE IF NOT EXISTS userinfom(acc\_id INTEGER PRIMARY KEY AUTOINCREMENT,username TEXT NOT NULL,mobilenumber TEXT NOT NULL UNIQUE,gender TEXT NOT NULL,emailid TEXT NOT NULL,city TEXT NOT NULL)'**)  
  
 def add\_details(self):  
 name = self.entry\_name.get()  
 mobile\_number = self.entry\_mobilenumber.get()  
 gender = self.entry\_gender.get()  
 email = self.entry\_email.get()  
 city = self.entry\_city.get()  
 if name and mobile\_number and gender and email and city != **""**:  
  
 query = **"INSERT INTO userinfom (username,mobilenumber,gender,emailid,city) values(?, ?, ?, ?, ?)"** cur.execute(query, (name, mobile\_number, gender, email, city))  
 con.commit()  
 messagebox.showinfo(**"success"**, **"user details added successfully"**)  
  
  
 else:  
 messagebox.showerror(**"error"**, **"fill all fields"**, icon=**'warning'**)