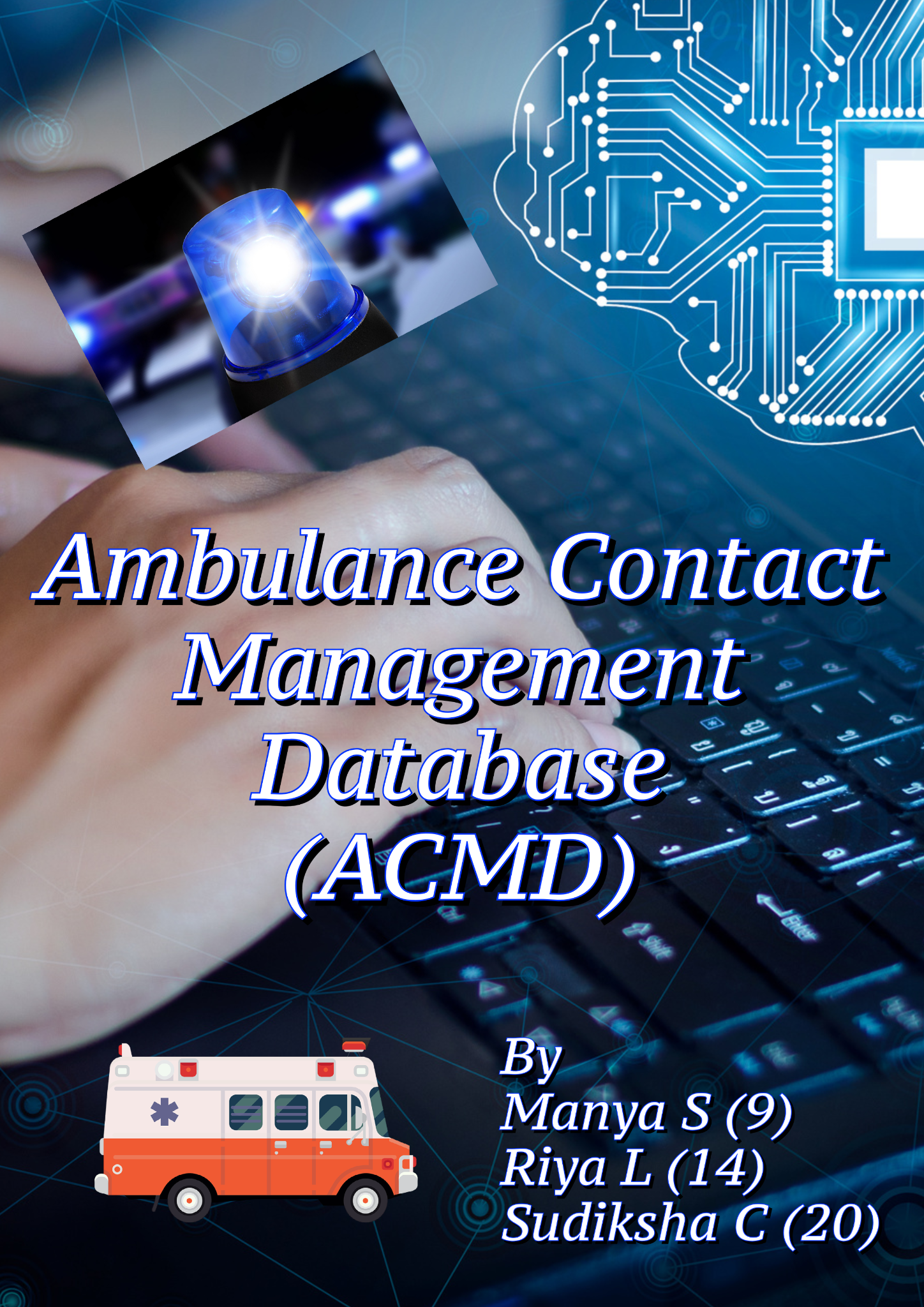
****

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Serial no.** | **Title** | **Page no.** |
| 1. | Introduction | 3 |
| 2. | Acknowledgement | 4 |
| 3. | System requirements | 5 |
| 4. | Basic Algorithm | 6-7 |
| 5. | Modules and Library functions | 8 |
| 6. | User Defined Functions | 9 |
| 7. | Variables | 10 |
| 8. | Source Code | 11-23 |
| 9. | Screen shots | 24-30 |
| 10. | Bibliography | 31 |

**INTRODUCTION**

In this project, there will be two interfaces for users and ambulance staff.

A user upon logging in can:

1. Update profile information

2. Access number of and details of ambulances and ambulance drivers in vicinity by inputting locality

3. Input information of current medical problem and required medical equipment

4. Directly contact the driver

A staff member upon logging in can:

1. View nearby hospitals for urgency

2. View a particular user’s personal information and medical requirements for checking compatibility

3. Input estimated time of commute

4. Update their ambulance information

The objective of this project is to create a symbiotic and inter dependent platform for:

1. users to contact the nearest ambulance in their locality during an emergency

2. ambulance drivers to locate nearby hospitals at a given time

As per various medical requirements.

**ACKNOWLEDGEMENT**

I would like to express a deep sense of gratitude towards my teacher Ms Revathi Krishnan for guiding me immensely, through the course of the project. She always evinced keen interest in my work. Her constructive advice and constant motivation have been responsible for the successful completion of this project.

My sincere thanks goes to Ms Shantha Chandran, our Mentor Principal and Ms Shweta Pathak Chand, our Principal for their support for the completion of this project.

I would also like to thank my family and friends for the constant support throughout the course of this project.

Finally, I would like to convey my regards to the Central Board of Secondary Education (CBSE) for giving me this opportunity.

**SYSTEM REQUIREMENTS**

● System requirements:

1. CPU

2. Monitor

● Software requirements:

1. Windows 10 or higher

2. Python 3.6 or higher

**BASIC ALGORITHM**

1. Start

2. If choice is new user input and

create username and password

3. If loginchoice is driver input name, age, medical equipment available, hospital linked and create profile

4. If loginchoice is client input name, age, medical conditions, name and contact number of family doctor and create profile

5. If choice is existing user input and verify username and password

6. Input prof

5. If prof is driver input choice

6. If choice is 1 input location

7. If choice is 2 input subchoice

8. If subchoice is 1 update profile

9. If subchoice is 2 delete value from profile

10. If subchoice is 3 delete profile

11. If choice is 3 display inbox and input

subchocie

12. If subchoice is y send message to selected client

13. If choice is 4 clear inbox

14. If choice is 5 break

15. If prof is client input choice

16. If choice is 1 input subchoice

17. If subchoice is 1 input location and display available drivers and input sub- subchoice

18. If sub- subchoice is y send message to selected driver

19. If subchoice is 2 break

20. If choice is 2 input subchoice and repeat steps 8-10

21. If choice is 3 repeat steps 11-12

22. If choice is 4 clear inbox

23. If choice is 5 break

24. Stop

**MODULES AND LIBRARY FUNCTIONS**

|  |  |  |
| --- | --- | --- |
| **Module** | **Library Functions** | **Usage** |
| os | .path.exists | To check if a file exists in the specified location |
| pickle | .load(file)  .dump(obj,file) | To load the objects from the file and to write the object into the file. |
| mysql.connector | .connect() | To connect mysql and python |
| csv | .writer(list)  .writerow(nested list)  .reader(file) | To write into a csv file and to read from the file |

**USER DEFINED FUNCTION**

|  |  |
| --- | --- |
| **Function name** | **Task** |
| conclusion | Gives conclusion for the program |

**VARIABLES**

|  |  |
| --- | --- |
| **Data Type** | **Variable name** |
| String | Username, password, nm, g, Location, dname, messagech, change |
| List | inbox,newuser, head, inp, samplelist |
| Dictionary | login |
| Integer/Floating Point | Ag, chasdr, chh, contact |

**SOURCE CODE**

#SQL PROFILE CREATION

# Location 1 - driver details

# Location 2 - client details

import mysql.connector

db=mysql.connector.connect(host='localhost', user='root', passwd='sql123', autocommit=True)

m=db.cursor()

# Checking if table exists

m.execute('select count(\*) from information\_schema.tables where table\_schema="acmd"')

z=m.fetchall()

for i in z:

for j in i:

if j==0:

m.execute('drop database if exists acmd')

m.execute('create database acmd')

m.execute('use acmd')

#Users table

m.execute('drop table if exists users')

m.execute('create table users (Name varchar(25), Age int, Gender varchar(10), Medical\_history varchar(100), Family\_doctor varchar(25), FD\_contactinfo varchar(20), current\_disease varchar(25))')

#Driver's table

m.execute('drop table if exists drivers')

m.execute("create table drivers (Name varchar(25), Age int, Gender varchar(10), Rating varchar(10), ME\_available varchar(100), LHospital varchar(25))")

else:

#print('The tables already exist')

m.execute('use acmd')

#driver interface

import csv

import pickle

#login-acmd

login={}

'''login={username:[name, password, profession, inbox]}

newuser=[name, password, profession, inbox]

inbox={sender's name: message}'''

import os

if os.path.exists('logins.dat'):

with open ("logins.dat","rb") as f:

e=pickle.load(f)

for i in e:

login[i]=e[i]

print(login)

inbox={}

newuser=[]

ct=0

#LOGIN

while True:

flag=0

print('='\*40)

print("Hello! Welcome to the ACMD Database ")

print("Enter 1 if you are a new user. ")

print("Enter 2 if you are an existing user. ")

print('='\*40)

print()

username=("Enter your username")

ch=int(input())

print()

#new user creation-done

if ch==1:

while True:

username=input("Create your username ")

if username in login:

print()

print("Sorry this username is already taken. Try again")

else:

login[username]=newuser

f1=open('logins.dat','wb')

pickle.dump(login,f1)

f1.close()

break

print()

while ct<5:

password=input("Enter your password ")

ct+=1

if len(password)<3:

print("The password must have atleast 3 characters.")

continue

elif password.isalpha()==True or password.isnumeric()==True:

print("The password must have alpahbets and numerals")

else:

print()

print("Your password has been linked successfully.")

break

print()

ctnew=0

while ct<5:

print()

print("If you are logging in as an ambulance service provider, please enter 1.")

print("If you are logging in as a client/patient, please enter 2.")

print()

loginchoice=int(input("Please enter your choice. "))

if loginchoice==1:

#'''PROFILE CREATION - DRIVER'''

nm=input("Enter your name ")

newuser=[nm, password, 'driver', {}]

login[nm]=newuser

ag=int(input('Enter your age '))

g=input('Enter your gender(Enter M for male and F for female) ')

if g=='m':

g='M'

elif g=='f':

g='F'

me=input('Enter the medical equipment available in the ambulance ')

lh=input('Enter the name of the hospital to which you are linked ')

m.execute("insert into drivers values (%s,%s,%s,NULL,%s,%s)",(username,ag,g,me,lh))

f1=open('logins.dat','wb')

pickle.dump(login,f1)

f1.close()

print('Your profile has been created successfully')

flag=1

db.commit()

break

elif loginchoice==2:

#'''PROFILE CREATION-USER''’

nm=input("Enter your name ")

newuser=[nm, password, 'client', {}]

login[nm]=newuser

ag=int(input('Enter your age '))

g=input('Enter your gender(Enter M for male and F for female) ')

if g=='m':

g='M'

elif g=='f':

g='F'

md=input('Enter your medical history ')

fd=input('Enter the name of your family doctor ')

fdc=input('Enter your family doctor\'s contact number ')

m.execute("insert into users values (%s,%s,%s,%s,%s,%s,NULL)",(nm, ag,g,md,fd,fdc))

f1=open('logins.dat','wb')

pickle.dump(login,f1)

f1.close()

print('Your profile has been created successfully')

print()

flag=1

db.commit()

break

else:

print("Sorry you have exhausted 5 trials, please begin again!")

flag=('hi')

print()

break

#EXISTING USER LOGIN

elif ch==2:

flag='hi'

username=input("Enter your username ")

while True:

if username not in login:

print("Username not found, please try again! ")

break

else:

if ct<5:

password=input("Enter your password ")

if login[username][1]!=password:

print("Password doesn't match")

ct+=1

if ct==5:

print("You have exhausted 5 trials. Please begin again!")

elif login[username][1]==password:

print("You have logged in successfully!")

flag=1

break

if flag==1:

break

if flag ==1:

print()

print("Hello welcome to ACMD - Ambulance Contact Management Database! We are here to provide you with immediate solutions in times of medical emergency.")

print("On this platform, you will be able to view the required data for contacting an ambulance for medical purposes.")

print()

#Creating csv file

locfile=open("Location1.csv",'a', newline='')

#writer=csv.writer(locfile)

dhead=['Driver Name', 'Location']

locfile.close()

locfile=open("Location2.csv",'a', newline='')

chead=['Client Name', 'Location']

#LOGIN AS DRIVER

if login[username][2]=='driver':

while True:

print('='\*40)

print("Enter 1 to input your current location to provide services.")

print("Enter 2 to update your profile on ACMD.")

print("Enter 3 to view your inbox.")

print("Enter 4 to clear your inbox.")

print("Enter 5 to exit ACMD")

print('='\*40)

print()

chasdr=int(input("Please enter your choice "))

print()

if chasdr==1:

locfile=open("Location1.csv", 'a+', newline='')

location=input("Enter your current location ")

writer=csv.writer(locfile)

name=login[username][0]

inp=[name, location]

writer.writerow(inp)

locfile.close()

continue

elif chasdr==2:

print('-'\*40)

print("Enter 1 to update any category in your profile.")

print("Enter 2 to delete a value.")

print("Enter 3 to delete your profile.")

print('-'\*40)

chh=int(input('Enter your choice '))

if chh==1:

nm=input('Enter your name ')

while True:

change=input('Enter the name of the category for which you want to change the information- age, gender,medical equipment available,linked hospital ')

if change.lower()!='name' and change.lower()!='age'and change.lower()!='gender'and change.lower()!='medical equipment available' and change.lower()!='linked hospital':

print('Invalid category name ')

else:

break

up=input("Enter the new information (If you are updating gender, enter 'M' for male and 'F' for female)")

print(f'{"Name":25s} {"Age":25s} {"Gender":25s} {"Med eqpment available":25s} {"Linked Hosp":25s}')

if change.lower()=='age':

m.execute('update drivers set age=%s where name=%s',(up,nm))

m.execute('select \* from drivers where name=%s',(nm,))

z=m.fetchall()

for i in z:

for j in i:

if type(j)==int:

print(f'{str(j):15s}',end=' ')

elif j=='4.8' or j=='4.2' or j=='4.5' or j=='4.4':

pass

elif j==None:

print(' ',end=' ')

else:

print(f'{j:15s}',end=' ')

db.commit()

elif change.lower()=='gender':

if up=='m':

up='M'

elif up=='f':

up='F'

m.execute('update drivers set gender=%s where name=%s',(up,nm))

m.execute('select \* from drivers where name=%s',(nm,))

z=m.fetchall()

for i in z:

for j in i:

if type(j)==int:

print(f'{str(j):25s}',end=' ')

elif j=='4.8' or j=='4.2' or j=='4.5' or j=='4.4':

pass

elif j==None:

print(' ',end=' ')

else:

print(f'{j:25s}',end=' ')

db.commit()

elif change.lower()=='medical history':

m.execute('update drivers set Medical\_history=%s where name=%s',(up,nm))

m.execute('select \* from drivers where name=%s',(nm,))

z=m.fetchall()

for i in z:

for j in i:

if type(j)==int:

print(f'{str(j):25s}',end=' ')

elif j=='4.8' or j=='4.2' or j=='4.5' or j=='4.4':

pass

elif j==None:

print(' ',end=' ')

else:

print(f'{j:25s}',end=' ')

db.commit()

elif change.lower()=='medical equipment':

m.execute('update drivers set ME\_available=%s where name=%s',(up,nm))

m.execute('select \* from drivers where name=%s',(nm,))

z=m.fetchall()

for i in z:

for j in i:

if type(j)==int:

print(f'{str(j):25s}',end=' ')

elif j=='4.8' or j=='4.2' or j=='4.5' or j=='4.4':

pass

elif j==None:

print(' ',end=' ')

else:

print(f'{j:25s}',end=' ')

db.commit()

elif change.lower()=='linked hospital':

m.execute('update drivers set LHospital=%s where name=%s',(up,nm))

m.execute('select \* from drivers where name=%s',(nm,))

z=m.fetchall()

for i in z:

for j in i:

if type(j)==int:

print(f'{str(j):25s}',end=' ')

elif j=='4.8' or j=='4.2' or j=='4.5' or j=='4.4':

pass

elif j==None:

print(' ',end=' ')

else:

print(f'{j:25s}',end=' ')

db.commit()

print('Your information has been updated successfully ')

print()

if chh==2:

nm=input('Enter your name ')

while True:

change=input('Enter the name of the category for which you want to delete the information- age, gender,medical equipment available,linked hospital ')

if change.lower()!='name' and change.lower()!='age'and change.lower()!='gender'and change.lower()!='medical equipment available' and change.lower()!='linked hospital':

print('Invalid category name ')

print()

else:

break

if change=='age':

m.execute('update drivers set age=NULL where name=%s',(nm,))

db.commit()

elif change=='gender':

m.execute('update drivers set gender=NULL where name=%s',(nm,))

db.commit()

elif change=='medical equipment':

m.execute('update drivers set ME\_available=NULL where name=%s',(nm,))

db.commit()

elif change=='linked hospital':

m.execute('update drivers set Lhospital=NULL where name=%s',(nm,))

db.commit()

print('The data has been deleted successfully.')

print()

if chh==3:

nm=input('Enter your name ')

m.execute('delete from drivers where name=%s',(nm,))

del login[nm]

with open ("logins.dat","wb") as f:

pickle.dump(login,f)

print('Your profile has been deleted successfully')

print()

elif chasdr==3:

readch=input("Do you wish to view your messages from your clients? Press Y or N ")

if readch.lower()=='y':

#login={username:[name, password, profession, inbox]}

f=login[username][3]

print(f)

profilech=input("Do you wish to contact any of the above clients? Press Y or N ")

if profilech.lower=='y':

cname=input("Enter the name of the client you wish to contact ")

for i in login:

if cname.lower()==login[i][0]:

m.execute('select \* from users where name=%s',(cname,))

print(f'{"Name":15s} {"Age":15s} {"Gender":15s} {"Medical history":15s} {"Family Dr":15s} {"Family Dr contact":15s} {"Current disease":15s}')

s=m.fetchall()

for i in s:

for j in i:

if type(j)==int:

print(f'{str(j):15s}',end=' ')

elif j==None:

print(' ',end=' ')

else:

print(f'{j:15s}',end=' '

messagech=input("Do you wish to enter a customised message for the client? Press Y or N ")

if messagech.lower()=='y':

message=input("Please enter your message to be sent to the client: ")

elif messagech.lower()=='n':

message="The requested driver has been assigned for emergency pick up at the earliest."

for i in login:

if i[0]==cname:

login[i][3][username]=message

f1=open('logins.dat','wb')

pickle.dump(login,f1)

f1.close()

print("You have successfully been assigned the client in your location. Ensure immediate pickup with required medical equiment.")

elif chasdr==4:

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

if username in login:

login[username][3]={}

f1=open('logins.dat','wb')

pickle.dump(login,f1)

f1.close()

print("Your inbox has been cleared.")

else:

print("Your profile does not exist")

elif chasdr==5:

break

else:

print("Invalid input. Please try again.")

continue

#LOGIN AS USER

if login[username][2]=='client':

while True:

print()

print('='\*40)

print("Enter 1 to input your current location to avail services.")

print("Enter 2 to update your profile on ACMD.")

print("Enter 3 to view your inbox.")

print("Enter 4 to clear your inbox.")

print("Enter 5 to exit ACMD")

print('='\*40)

chascl=int(input("Please enter your choice "))

print()

if chascl==1:

locfile=open("Location2.csv", 'a+', newline='')

location=input("Enter your current location ")

location=location.lower()

writer=csv.writer(locfile)

inp=[username, location]

writer.writerow(inp)

locfile.close()

while True:

print('-'\*40)

print("Enter 1 to search for drivers.")

print("Enter 2 to exit this menu.")

print('-'\*40)

print()

ccl=int(input("Enter your choice "))

if ccl==1:

print()

locfile2=open("Location1.csv", 'r')

reader=csv.reader(locfile2) #[name, location]

for i in reader:

if i[1].lower()==location:

for j in dhead:

print(j, end='\t')

print()

print(i[0], i[1], sep='\t')

print("These are the drivers in your current location.")

dname=input("Enter the name of the driver you wish to contact: ")

print("Here are the details of the driver's profile:")

m.execute('select \* from drivers where name=%s',(dname,))

print(f'{"Name":15s} {"Age":15s} {"Gender":15s} {"Med eqpment available":15s} {"Linked Hosp":15s}')

s=m.fetchall()

for i in s:

for j in i:

if type(j)==int:

print(f'{str(j):15s}',end=' ')

elif j=='4.8' or j=='4.2' or j=='4.5' or j=='4.4':

pass

elif j==None:

print(' ',end=' ')

else:

print(f'{j:15s}',end=' ')

#from sql display driver details and to print medical equiment of ambulance from sql also

contact=input("Do you wish to proceed to contact the driver? Press Y or N ")

if contact.lower()=='y':

#RIYA SOUND INPUT

## from audioplayer import AudioPlayer

# "D:\Class 12 2022-23 Riya\ACMD\phone-calling-1.mp3"

## x = AudioPlayer(r"D:/Class 12 2022-23 Riya/ACMD/phone-calling-1.mp3")

## x.play(block=True)

messagech=input("Do you wish to enter a customised message for the driver? Press Y or N ")

if messagech.lower()=='y':

message=input("Please enter your message to be sent to the driver: ")

print("You have successfully contacted the driver.")

elif messagech.lower()=='n':

message=("You have received a client request. They are in location -" +location)

for i in login:

if login[i][0]==dname:

login[i][3][username]=message # inbox={name:message}

#login={username:[name, password, profession, inbox]}

f1=open('logins.dat','wb')

pickle.dump(login,f1)

f1.close()

else:

print('Sorry, there are no drivers available in your location.')

break

if ccl==2:

break

elif chascl==2:

print('-'\*40)

print("Enter 1 to update any category in your profile.")

print("Enter 2 to delete a value.")

print("Enter 3 to delete your profile.")

print('-'\*40)

chh=int(input('Enter your choice '))

if chh==1:

nm=input('Enter your name ')

while True:

change=input('Enter the name of the category for which you want to change the information- age, gender, medical history, family doctor, family doctor contact number ')

if change.lower()!='name' and change.lower()!='age'and change.lower()!='gender'and change.lower()!='medical history'and change.lower()!='family doctor'and change.lower()!='family doctor contact number':

print('Invalid category name ')

print()

else:

break

up=input("Enter the new information (If you are updating gender, enter 'M' for male and 'F' for female)")

print(f'{"Name":15s} {"Age":15s} {"Gender":15s} {"Medical history":15s} {"Family Dr":15s} {"Family Dr contact":15s} {"Current disease":15s}')

if change.lower()=='age':

m.execute('update users set age=%s where name=%s',(up,nm))

m.execute('select \* from users where name=%s',(nm,))

z=m.fetchall()

for i in z:

for j in i:

if type(j)==int:

print(f'{str(j):15s}',end=' ')

elif j==None:

print(' ',end=' ')

else:

print(f'{j:15s}',end=' ')

if change.lower()=='gender':

if up=='m':

up='M'

elif up=='f':

up='F'

m.execute('update users set gender=%s where name=%s',(up,nm))

m.execute('select \* from users where name=%s',(nm,))

z=m.fetchall()

for i in z:

for j in i:

if type(j)==int:

print(f'{str(j):15s}',end=' ')

elif j==None:

print(' ',end=' ')

else:

print(f'{j:15s}',end=' ')

if change.lower()=='medical history':

m.execute('update users set Medical\_history=%s where name=%s',(up,nm))

m.execute('select \* from users where name=%s',(nm,))

z=m.fetchall()

for i in z:

for j in i:

if type(j)==int:

print(f'{str(j):15s}',end=' ')

elif j==None:

print(' ',end=' ')

else:

print(f'{j:15s}',end=' ')

elif change.lower()=='family doctor':

m.execute('update users set Family\_doctor=%s where name=%s',(up,nm))

m.execute('select \* from users where name=%s',(nm,))

z=m.fetchall()

for i in z:

for j in i:

if type(j)==int:

print(f'{str(j):15s}',end=' ')

elif j==None:

print(' ',end=' ')

else:

print(f'{j:15s}',end=' ')

elif change.lower()=='family doctor contact number':

m.execute('update users set FD\_contactinfo=%s where name=%s',(up,nm))

m.execute('select \* from users where name=%s',(nm,))

z=m.fetchall()

for i in z:

for j in i:

if type(j)==int:

print(f'{str(j):15s}',end=' ')

elif j==None:

print(' ',end=' ')

else:

print(f'{j:15s}',end=' ')

print('Your information has been updated successfully ')

print()

if chh==2:

nm=input('Enter your name ')

while True:

change=input('Enter the name of the category for which you want to delete the information- age, gender, medical history, family doctor, family doctor contact number ')

if change.lower()!='name' and change.lower()!='age'and change.lower()!='gender'and change.lower()!='medical history'and change.lower()!='family doctor'and change.lower()!='family doctor contact number':

print('Invalid category name ')

print()

else:

break

if change=='age':

m.execute('update users set age=NULL where name=%s',(nm,))

if change=='gender':

m.execute('update users set gender=NULL where name=%s',(nm,))

if change=='medical history':

m.execute('update users set Medical\_history=NULL where name=%s',(nm,))

elif change=='family doctor':

m.execute('update users set Family\_doctor=NULL where name=%s',(nm,))

elif change=='family doctor contact number':

m.execute('update users set FD\_contactinfo=NULL where name=%s',(nm,))

print('The data has been deleted successfully.')

print()

if chh==3:

nm=input('Enter your name ')

m.execute('delete from users where name=%s',(nm,))

del login[nm]

with open ("logins.dat","wb") as f:

pickle.dump(login,f)

print('Your profile has been deleted successfully')

print()

elif chascl==3:

readch=input("Do you wish to view your messages from driver(s)? Press Y or N ")

if readch.lower()=='y':

f=login[username][3]

print(f)

elif chascl==4:

if username in login:

login[username][3]={}

f1=open('logins.dat','wb')

pickle.dump(login,f1)

f1.close()

else:

print("Your profile does not exist.")

print("Your inbox has been cleared.")

elif chascl==5:

break

else:

print("Invalid input. Please try again.")

continue

print("Exiting ACMD.....")

print()

print()

print()

print()

print()

print()

print()

def conclusion():

print("You have exited the progam")

print("Thank you for using ACMD. We hope we were able to solve your medical emergency needs. Do consider giving us a google review when you are free and healthy.")

print("We wish to see you again!")

print("Regards, Team ACMD")

conclusion()

**SCREENSHOTS**

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated



Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**BIBLIOGRAPHY**

To make the project, help was taken from the following sources:-

1. Computer Science with Python by Sumita Arora

2. www.google.com

3. <https://www.soundjay.com/phone-sounds-2.html>

4. <https://www.youtube.com/watch?v=Wt8VRxUYDso>

5. <https://www.geeksforgeeks.org/play-sound-in-python/>

6. <https://pythonbasics.org/python-play-sound/>