PYTHON

PROJECT

HOSPITAL MANAGEMENT SYSTEM USING PYTHON AND MYSQL

SHRUTI PRADHAN 12-E

BOARD ROLL NO:

INDEX

Certificate of Authenticity

Certificate of Authenticity

This is to certify that ..... bearing Roll No ..... is a student of Class XII of Sanskriti School.
He has successfully completed his project under my guidance and supervision towards the fulfillment of the practical examination in Computer Science conducted by the Central Board of Secondary Education for the academic year 2020-21.

Date of Submission:
Name of Subject Teacher:
Teacher's Signature:
School Stamp:

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude towards my Computer Science teacher Mrs Anuja Mittal for her guidance and support which allowed me to successfully complete my project on time.

I would also like to extend my gratitude towards our school's Principal Mrs Richa Sharma Agnihotri for providing me with all the facility that was required

Date

11/01/2021

Shruti Pradhan

12 E

PROJECT SYNOPSIS

The main aim of this project is to automate the process of managing (inputting data, storing data, updating data and deleting data) the details of a hospital, including tables of its 1) Patients, 2) Doctors and 3)Appointments using a Python and Mysql program, enhanced by a Graphical User Interface(GUI) built using the Tkinter library.

The process of 'Patient Registration' and 'Appointment Registration' can be done by all users but administrative tasks like 'Doctor Registration' and viewing, updating and deleting of information can only be done by a valid administrator after logging in.

This project aims to reduce the access time, improve the efficiency , automate the process and show a simple analysis of a Hospital Management System.

It tracks the details of patients and doctors (Name ,Sex, DOB, BloodGroup, Address) and appointments (PatientId, ContactNo, DoctorID, Date and time of appointment, problem)

PROJECT REQUIREMENTS

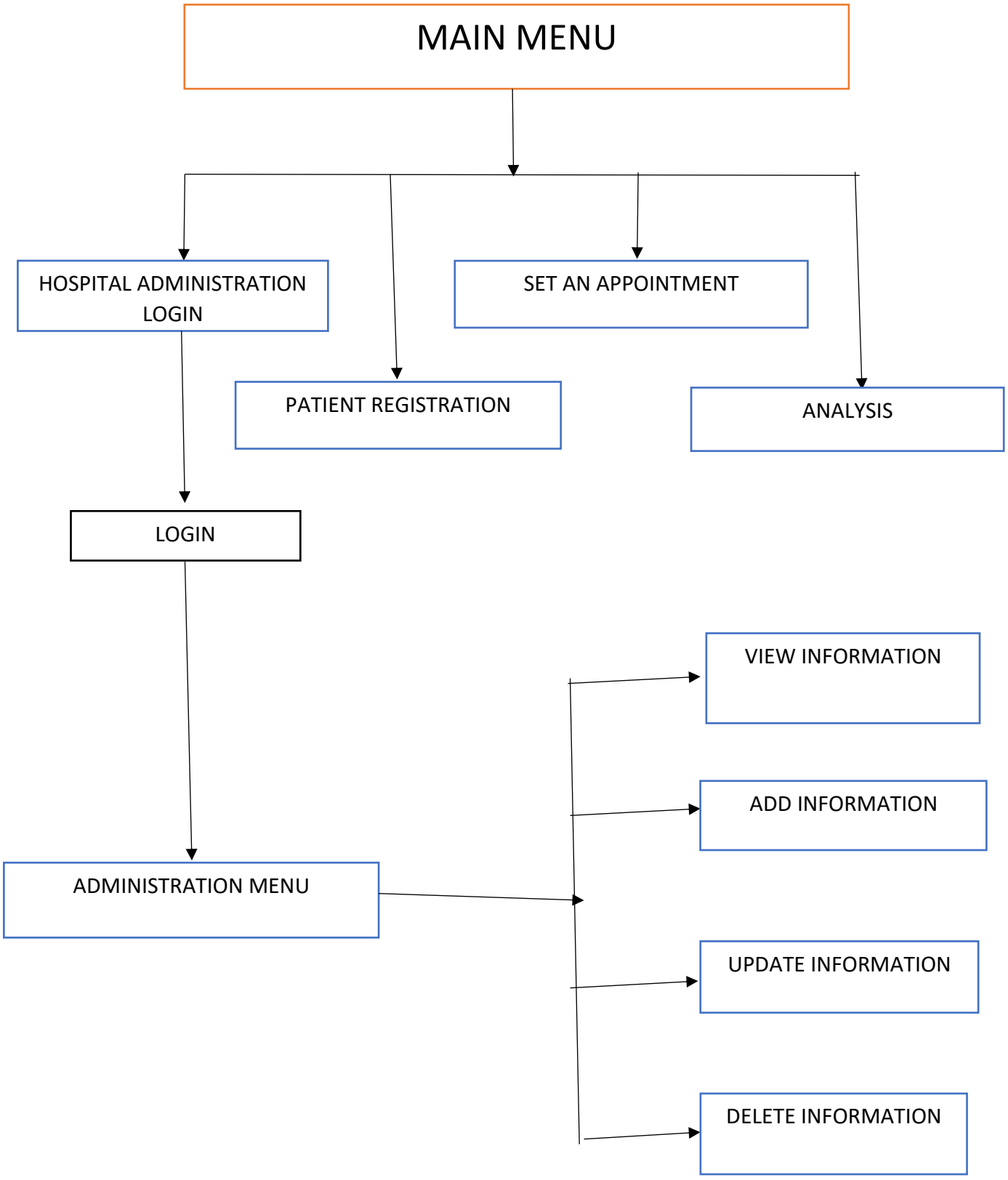
HARDWARE USED/REQUIRED:

1) Processor: Intel® Core (™) CPU @ 1.6 GHZ
2) RAM: 8.00 GB
3) System Type: 64-bit OS, x64 based processor
4) 4)HDD-500GB

SOFTWARE REQUIRED:

1) Windows 10 Operating System
2) Python Interpreter IDE 3.7.4
3) Mysql 5.7 Command line client
4) Models/Libraries used:
    - mysql. connector
    - tkinter
    - matplotlib.pyplot

PROJECT PLAN



SOURCE CODE

```
 from tkinter import*
import mysql.connector
import matplotlib.pyplot as plt


#establishing mysql connection
con=None
mucursor=None
def connect():
    global con,mycursor
    con = mysql.connector.connect(host="localhost", user="root",
passwd="welcome",database="project")
    mycursor = con.cursor()


#function for creating windows
def window(name,size,colour,heading):
    name.geometry(size)
    name.configure(bg=colour)
    name.title(heading)
    name.lift()
```

```
#defining variables for login
userbox=None
passbox=None

#global variables required for closing windows
scr2=None
scr4=None
scr6=None
scr9=None
scr11=None
scr13=None
scr14=None
scr16=None
scr17=None
scr18=None

#defining variables for patient registration
namebox=None
sexbox=None
dobbox=None
bloodbox=None
addressbox=None

#defining variables for doctor registration
```

```
docnamebox=None
docsexbox=None
docdepartmentbox=None
docdobbox=None
docaddressbox=None

#defining variables for appointment
patidbox=None
docidbox=None
datebox=None
timebox=None
problembox=None
phonebox=None

#defining variables for searching 1 entry
patsearchbox=None
docsearchbox=None

#defining variables for deleting
delpatbox=None
deldocbox=None

#defining variables for updating patient info
up_patidbox=None
```

```
up_namebox=None
up_sexbox=None
up_dobbox=None
up_bloodbox=None
up_addressbox=None

#defining variables for updating doctor info
up_docidbox=None
up_docnamebox=None
up_docsexbox=None
up_dbox=None
up_docdobbox=None
up_docaddressbox=None

"#################################################################
#################################################################
#####################################"
#ADMIN MENU

"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~`"
#view commands

def viewallpatient():
```

```
    global con,mycursor
    connect()
    mycursor.execute("Select * from patient")
    result=mycursor.fetchall()
    a=Toplevel()
    window(a,"900x800","grey","Patients")
    header="Id\t\t Name\t\t\t Sex\t\t DOB\t\t\t BloodGroup\t\t
Address\t\t\t"
    head=Label(a,text=header,fg="black")
    head.place(x=10,y=10)
    con.close()
    laby=50
    for row in result:
        answer='{}\t\t {}\t\t {}\t\t {}\t\t {}\t\t
{}'.format(row[0],row[1],row[2],row[3],row[4],row[5])
        Label(a,text=answer,fg="black",bg="grey").place(x=10,y=laby)
        laby+=20


 def viewonepatient():
    global patsearchbox,con,mycursor
    connect()
    identity=patsearchbox.get()


    def error1():
```

```
        d=Toplevel()
        window(d,"300x40","white","error")
        Label(d,text="No records exist for given Patient
id",fg="green",bg="grey").pack()


    try:
        mycursor.execute("Select* from patient where
id={}".format(identity) )
        result1=mycursor.fetchall()
    except:
        error1()
    else:
        if len(result1)==0:
            error1()
        else:
            c=Toplevel()
            window(c,"900x100","grey","Result")
            header1="Id\t\t Name\t\t\t Sex\t\t DOB\t\t\t BloodGroup\t\t
Address\t\t\t"
            head1=Label(c,text=header1,fg="black")
            head1.place(x=10,y=10)
            for row in result1:
                answer1='{}\t\t {}\t\t {}\t\t {}\t\t {}\t\t
{}'.format(row[0],row[1],row[2],row[3],row[4],row[5])
                Label(c,text=answer1,fg="black",bg="grey").place(x=10,y=30)
```

```
 def viewalldoctor():
    global con,mycursor
    connect()
    mycursor.execute("Select* from doctor")
    result=mycursor.fetchall()
    f=Toplevel()
    window(f,"1000x500","grey","Doctors")
    header="Id\t\t Name\t\t\t\t Sex\t\t Department\t\t\t DOB\t\t
Address\t\t\t"
    head=Label(f,text=header,fg="black")
    head.place(x=10,y=10)
    laby=50
    for row in result:
        answer='{}\t\t {}\t\t\t {}\t\t {}\t\t {}\t\t
{}'.format(row[0],row[1],row[2],row[3],row[4],row[5])
        Label(f,text=answer,fg="black",bg="grey").place(x=10,y=laby)
        laby+=20
    con.close()


 def viewonedoctor():
    global docsearchbox,con,mycursor
    connect()
    identity=docsearchbox.get()
```

```
    def error2():
        a1=Toplevel()
        window(a1,"300x50","grey","Error")
        Label(a1,text="No records exist for given Docotor
Id",fg="green",bg="grey").pack()
    try:
       mycursor.execute("Select* from doctor where id={}".format(identity)
)
       result1=mycursor.fetchall()
    except:
       error2()
    else:
       if len(result1)==0:
          error2()
       else:
          a2=Toplevel()
          window(a2,"900x100","grey","Result")
          header1="Id\t\t Name\t\t\t Sex\t\t Department\t\t\t DOB\t\t
Address\t\t\t"
          head1=Label(a2,text=header1,fg="black")
          head1.place(x=10,y=10)
          for row in result1:
             answer1='{}\t\t {}\t\t {}\t\t {}\t\t {}\t\t
{}'.format(row[0],row[1],row[2],row[3],row[4],row[5])
             Label(a2,text=answer1,fg="black",bg="grey").place(x=10,y=30)
```

```
 def viewappointment():
    global con,mycursor
    connect()
    mycursor.execute("Select* from appointment")
    result=mycursor.fetchall()
    c3=Toplevel()
    window(c3,"1000x500","grey","Appointments")
    header="AppmtNo\t Patient_ID\t\t ContactNo\t\t Doctor_ID\t
Date_Appmt\t\t Time_Appmt\t\t Problem\t\t"
    head=Label(c3,text=header,fg="black")
    head.place(x=10,y=10)
    laby=50
    for row in result:
        answer='{}\t\t {}\t\t\t {}\t\t {}\t\t {}\t\t {}\t\t\t
{}\t\t'.format(row[0],row[1],row[2],row[3],row[4],row[5],row[6])
        Label(c3,text=answer,fg="black",bg="grey").place(x=10,y=laby)
        laby+=20
    con.close()


 def view():
    global patsearchbox,scr9,docsearchbox


    scr9=Toplevel()
```

```
    window(scr9,"500x500","pink","View Information")
    Label(scr9,text="View/Search Information",bg="pink",fg="blue").pack()


    Label(scr9,text="Patient
Information",bg="pink",fg="black").place(x=5,y=30)
    Button(scr9,text="1. View all
information",bg='grey',fg='blue',command=viewallpatient).place(x=5,y=60
)
    Label(scr9,text="2. Search one record | Enter
Id",bg="white",fg="blue").place(x=5,y=100)
    patsearchbox=Entry(scr9,bg="white",width=20)
    patsearchbox.place(x=190,y=100)

Button(scr9,text="Search",bg="grey",fg="blue",command=viewonepatient
).place(x=335,y=100)


    Label(scr9,text="Doctor
Information",bg="pink",fg="black").place(x=5,y=140)
    Button(scr9,text="1. View all
informatiom",bg="grey",fg="blue",command=viewalldoctor).place(x=5,y=1
70)
    Label(scr9,text="2. Search one record | Enter
Id",bg="white",fg="blue").place(x=5,y=210)
    docsearchbox=Entry(scr9,bg="white",width=20)
    docsearchbox.place(x=190,y=210)
```

```
Button(scr9,text="Search",bg="grey",fg="blue",command=viewonedoctor)
.place(x=335,y=210)


    Label(scr9,text="Appointment
Information",bg="pink",fg="black").place(x=5,y=250)
    Button(scr9,text="View All
Information",bg='grey',fg='blue',command=viewappointment).place(x=5,y
=280)
    scr9.mainloop()


 "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
 #adding commands


 def docregistration():
    global
scr11,docnamebox,docsexbox,docdobbox,docdepartmentbox,docaddress
box


    #retrieve variables
    n=docnamebox.get()
    s=docsexbox.get()
    dep=docdepartmentbox.get()
    d=docdobbox.get()
```

```
    a=docaddressbox.get()


    #add to db
    global con,mycursor
    connect()
    mycursor.execute("Select max(id) from doctor")
    result=mycursor.fetchone()
    maximum=int(result[0])
    docid=maximum+1
    try:
        mycursor.execute("Insert into doctor values(
{},'{}','{}','{}','{}','{}')".format(docid,n,s,dep,d,a) )
        con.commit()
    except:
       wr1=Toplevel()
       window(wr1,"300x100","black","")
       Label(wr1,text="Values entered are incorrect \n Try
again",bg="black",fg="red").pack()


    else:
       scr11.destroy()
       #show succesful window
       a5=Toplevel()
       window(a5,"200x200","white","")
```

```
        Label(a5,text="Registration Succesful",fg='red',bg='white').pack()
        Label(a5,text="Your Doctor id is {}.\nPlease remember
it".format(docid),fg="red",bg="white").pack()
        Button(a5,text="Ok",bg='grey',fg="black",
command=a5.destroy).pack()


 def doctor_register():
    global
scr11,docnamebox,docsexbox,docdepartmentbox,docdobbox,docaddress
box
    scr11=Toplevel()
    window(scr11,"300x500","lavender blush","Doctor Registration Form")
    Label(scr11,text="Doctor Registration
Form",fg='blue',bg='white').pack()


    Label(scr11,text="Full Name",bg='grey',fg='black').pack()
    docnamebox=Entry(scr11,bg="white",width=20)
    docnamebox.pack()


    Label(scr11,text="Sex(M/F)",bg='grey',fg='black').pack()
    docsexbox=Entry(scr11,bg="white",width=8)
    docsexbox.pack()


    Label(scr11,text="Department",bg='grey',fg='black').pack()
    docdepartmentbox=Entry(scr11,bg="white",width=18)
```

```
    docdepartmentbox.pack()


    Label(scr11,text="DOB(YYYY/MM/DD)",bg='grey',fg='black').pack()
    docdobbox=Entry(scr11,bg="white",width=12)
    docdobbox.pack()


    Label(scr11,text="Address",bg='grey',fg='black').pack()
    docaddressbox=Entry(scr11,bg="white",width=25)
    docaddressbox.pack()



Button(scr11,text="Register",bg='white',fg='blue',command=docregistrati
on).place(x=122,y=450)


 def add():
    scr10=Toplevel()
    window(scr10,"400x300","pink","Add Information")
    Label(scr10,text="Add Patient/Doctor
Information",bg="pink",fg="blue").pack()
    Button(scr10,text="Add Patient
information",bg="grey",fg="blue",command=patient_register).place(x=20,
y=100)
    Button(scr10,text="Add Doctor
Information",bg="grey",fg="blue",command=doctor_register).place(x=200
,y=100)
```

```
 "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~``"
 #deleting commands
 def delpatient():
    global delpatbox,scr13,con,mycursor
    connect()
    def wr3():
        wr3=Toplevel()
        window(wr3,"200x50","black","")
        Label(wr3,text="Operation Unsuccesful. Try
Again",bg="black",fg="red").pack()



    delid=int(delpatbox.get())
    mycursor.execute("Select id from Patient")
    res=mycursor.fetchall()
    existing=[]
    con.close()
    for i in res:
        existing.append((i[0]))
    try:
        connect()
        query="delete from patient where id={}".format(delid)
        mycursor.execute(query)
```

```
        con.commit()
    except:
        wr3()
    else:
        if delid not in existing:
            wr3()

        else:
            a6=Toplevel()
            window(a6,"300x100","white","")
            ans='Patient Record with Patient Id={} succesfully
deleted.'.format(delid)
            Label(a6,text=ans,fg="blue",bg="white").pack()

Button(a6,text="Ok",fg="black",bg="grey",command=a6.destroy).pack()
            scr13.destroy()



 def deletepatient():
    global delpatbox,scr13
    scr13=Toplevel()
    window(scr13,"300x300","bisque2","Delete Patient record")
    Label(scr13,text="Enter Patient Id to be
Deleted:",bg="white",fg="red").place(x=10,y=40)
    delpatbox=Entry(scr13,bg="white",width=20)
```

```
    delpatbox.place(x=10,y=70)

Button(scr13,text="Delete",bg="grey",fg="blue",command=delpatient).pla
ce(x=10,y=100)


 def deldoctor():
    global deldocbox,scr14,con,mycursor
    connect()
    def wr4():
        wr3=Toplevel()
        window(wr3,"200x50","black","")
        Label(wr3,text="Operation Unsuccesful. Try
Again",bg="black",fg="red").pack()


    delid=int(deldocbox.get())
    mycursor.execute("Select id from Doctor")
    res=mycursor.fetchall()
    existing=[]
    for i in res:
        existing.append(int(i[0]))
    if delid not in existing:
        wr4()
        print("mistake here")
    else:
        try:
```

```
            connect()
            query="delete from doctor where id={}".format(delid)
            mycursor.execute(query)
            con.commit()
        except:
            wr4()
            print("no mistake here")
        else:
            a7=Toplevel()
            window(a7,"300x100","white","")
            ans='Doctor Record with Doctor Id={} succesfully
deleted.'.format(delid)
            Label(a7,text=ans,fg="blue",bg="white").pack()

Button(a7,text="Ok",fg="black",bg="grey",command=a7.destroy).pack()
            scr14.destroy()



 def deletedoctor():
    global deldocbox,scr14
    scr14=Toplevel()
    window(scr14,"300x300","pink3","Delete Doctor Record")
    Label(scr14,text="Enter Doctor Id to be
deleted:",bg="white",fg="blue").place(x=10,y=40)
    deldocbox=Entry(scr14,bg="white",width=20)
```

```
    deldocbox.place(x=10,y=70)

Button(scr14,text="Delete",bg="grey",fg="black",command=deldoctor).pla
ce(x=10,y=100)


 def delete():
    scr12=Toplevel()
    window(scr12,"400x300","tomato","Delete Information")
    Label(scr12,text='Delete
Entries',bg="tomato",fg="blue").place(x=150,y=10)
    Button(scr12,text="Delete Patient
Information",bg="white",fg="black",command=deletepatient).place(x=20,
y=100)
    Button(scr12,text="Delete Doctor
Information",bg="white",fg="black",command=deletedoctor).place(x=200,
y=100)


 "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~```"
 # Update commands


 def up_patient():
    global
scr17,up_patidbox,up_namebox,up_sexbox,up_dobbox,up_bloodbox,up_
addressbox,con,mycursor
    upid=int(up_patidbox.get())
```

```
    n=up_namebox.get()
    s=up_sexbox.get()
    d=up_dobbox.get()
    b=up_bloodbox.get()
    a=up_addressbox.get()

    connect()
    mycursor.execute("Select id from Patient")
    res=mycursor.fetchall()
    existing=[]
    con.close()
    for i in res:
        existing.append((i[0]))

    def wrong():
        wr5=Toplevel()
        window(wr5,"200x50","black","")
        Label(wr5,text="Operation Unsuccesful. Try
Again",bg="black",fg="red").pack()

    try:
        connect()
        mycursor.execute( "Update patient set
name='{}',sex='{}',dob='{}',blood_group='{}',address='{}' where id={}
".format(n,s,d,b,a,upid) )
```

```
        con.commit()
    except:
        wrong()
    else:
        if upid not in existing:
            wrong()
        else:
            a8=Toplevel()
            window(a8,"300x100","white","")
            ans='Patient Record with Patient Id={} succesfully
deleted.'.format(upid)
            Label(a8,text=ans,fg="blue",bg="white").pack()

Button(a8,text="Ok",fg="black",bg="grey",command=a8.destroy).pack()
            scr17.destroy()


 def updatepatient():
    global
scr17,up_patidbox,up_namebox,up_sexbox,up_dobbox,up_bloodbox,up_
addressbox
    scr17=Toplevel()
    window(scr17,"300x600","thistle1","Update Patient Information")
    Label(scr17,text="Update Patient
Information",bg="thistle1",fg="blue").pack()
```

```
    Label(scr17,text="Enter Patient Id to be
updated",bg="grey",fg="black").pack()
    up_patidbox=Entry(scr17,bg="white",width=10)
    up_patidbox.pack()

    Label(scr17,text="Full Name",bg='snow',fg='black').pack()
    up_namebox=Entry(scr17,bg="white",width=20)
    up_namebox.pack()

    Label(scr17,text="Sex(M/F)",bg='snow',fg='black').pack()
    up_sexbox=Entry(scr17,bg="white",width=8)
    up_sexbox.pack()

    Label(scr17,text="DOB(YYYY/MM/DD)",bg='snow',fg='black').pack()
    up_dobbox=Entry(scr17,bg="white",width=12)
    up_dobbox.pack()

    Label(scr17,text="Blood Group",bg='snow',fg='black').pack()
    up_bloodbox=Entry(scr17,bg="white",width=8)
    up_bloodbox.pack()

    Label(scr17,text="Address",bg='snow',fg='black').pack()
    up_addressbox=Entry(scr17,bg="white",width=25)
    up_addressbox.pack()
```

```
Button(scr17,text="Update",bg='snow',fg='blue',command=up_patient).pl
ace(x=122,y=450)
 def up_doctor():
    global
scr18,up_docidbox,up_docnamebox,up_docsexbox,up_dbox,up_docdobb
ox,up_docaddressbox,con,mycursor
    upid=up_docidbox.get()
    n=up_docnamebox.get()
    s=up_docsexbox.get()
    d=up_docdobbox.get()
    dep=up_dbox.get()
    a=up_docaddressbox.get()



    connect()
    mycursor.execute("Select id from Doctor")
    res=mycursor.fetchall()
    existing=[]
    con.close()
    for i in res:
       existing.append((i[0]))


    def wrong():
```

```
        wr5=Toplevel()
        window(wr5,"200x50","black","")
        Label(wr5,text="Operation Unsuccesful. Try
Again",bg="black",fg="red").pack()


    try:
        connect()
        mycursor.execute( "Update doctor set
name='{}',sex='{}',dob='{}',department='{}',address='{}' where id={}
".format(n,s,d,dep,a,upid) )
        con.commit()
    except:
        wrong()
    else:
        if upid not in existing:
            wrong()
        else:
            a9=Toplevel()
            window(a9,"300x100","white","")
            ans='Doctor Record with Doctor Id={} succesfully
deleted.'.format(upid)
            Label(a9,text=ans,fg="blue",bg="white").pack()

Button(a9,text="Ok",fg="black",bg="grey",command=a9.destroy).pack()
            scr18.destroy()
```

```
 def updatedoctor():
    global
scr18,up_docidbox,up_docnamebox,up_docsexbox,up_departmentbox,up
_docdobbox,up_docaddressbox
    global up_dbox


    scr18=Toplevel()
    window(scr18,"300x600","thistle1","Update Doctor Information")
    Label(scr18,text="Update Doctor
Information",bg="thistle1",fg="blue").pack()


    Label(scr18,text="Enter Doctor Id to be
updated",bg="grey",fg="black").pack()
    up_docidbox=Entry(scr18,bg="white",width=10)
    up_docidbox.pack()


    Label(scr18,text="Full Name",bg='snow',fg='black').pack()
    up_docnamebox=Entry(scr18,bg="white",width=20)
    up_docnamebox.pack()


    Label(scr18,text="Sex(M/F)",bg='snow',fg='black').pack()
    up_docsexbox=Entry(scr18,bg="white",width=8)
    up_docsexbox.pack()
```

```
    Label(scr18,text="Department",bg='snow',fg='black').pack()
    up_dbox=Entry(scr18,bg="white",width=8)
    up_dbox.pack()

    Label(scr18,text="DOB(YYYY/MM/DD)",bg='snow',fg='black').pack()
    up_docdobbox=Entry(scr18,bg="white",width=12)
    up_docdobbox.pack()

    Label(scr18,text="Address",bg='snow',fg='black').pack()
    up_docaddressbox=Entry(scr18,bg="white",width=25)
    up_docaddressbox.pack()


Button(scr18,text="Update",bg='snow',fg='blue',command=up_doctor).pla
ce(x=122,y=450)

 def update():

    scr15=Toplevel()
    window(scr15,"400x300","plum1","Update Information")
    Label(scr15,text="Update
Entries",bg="plum1",fg="blue").place(x=150,y=10)
```

```
    Button(scr15,text="Update Patient information",bg="white",fg="black",command=updatepatient).place(x=20,y=100)
    Button(scr15,text="Update Doctor Information",bg="white",fg="black",command=updatedoctor).place(x=200,y=100)



 "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~`"
 #`Main admin menu
 def adminmenu():
    scr3=Toplevel()
    window(scr3,"400x400","white","Administration Menu")

    Label(scr3,text="Welcome",bg="white",fg="blue").pack()
    Button(scr3,text="View information",bg='grey',fg='black',command=view).pack()
    Button(scr3,text="Add information",bg='grey',fg='black',command=add).pack()
    Button(scr3,text="Update Information ",bg='grey',fg='black',command=update).pack()
    Button(scr3,text="Delete information",bg='grey',fg='black',command=delete).pack()
```

```
Button(scr3,text="Back",bg='grey',fg='black',command=scr3.destroy).pack(
)
 '-------------------------------------------------------------------------------------------
-----------------------------------------------------'
 "#############################################################
#############################################################
#########################################"
 #login
 def logincheck():
    global userbox,passbox,scr2
    u=userbox.get()
    p=passbox.get()



    usernames=['admin','nurse','doctor']
    passwords=['admin1','nurse1','doctor1']
    success=False

    if u in usernames:
        ind=usernames.index(u)
        password=passwords[ind]
        if password==p:
            success=True
    if success:
```

```
        scr2.destroy()
        adminmenu()
    else:
        Label(scr2,text="Incorrect login and/or password. Try
again",fg="red",bg="black").pack()

 def entry():
    global scr2
    scr2=Toplevel()
    window(scr2,"400x400","black","Login")
    global userbox,passbox

    Label(scr2,text="Enter username: ",fg="blue",bg="white").pack()
    userbox=Entry(scr2,bg='white',width=10)
    userbox.pack()

    Label(scr2,text="Enter password: ",fg="blue",bg="white").pack()
    passbox=Entry(scr2,show='*',bg='white',width=10)
    passbox.pack()


Button(scr2,text="Login",fg="blue",bg="white",command=logincheck).pac
k()
    scr2.mainloop()
```

```
 '-------------------------------------------------------------------------------------------
-------------------------------------------------------------'

 "#############################################################
#############################################################
#############################################"

 #patient registration form


 def registration():
    global scr4,namebox,sexbox,dobbox,bloodbox,addressbox


    #retrieve variables
    n=namebox.get()
    s=sexbox.get()
    d=dobbox.get()
    b=bloodbox.get()
    a=addressbox.get()


    #add to db
    global con,mycursor
    connect()
    mycursor.execute("Select max(id) from Patient")
    used=mycursor.fetchone()
    maximum=int(used[0])
    patid=maximum+1
```

```
    try:
        mycursor.execute("Insert into patient values(
{},'{}','{}','{}','{}','{}')".format(patid,n,s,d,b,a) )
        con.commit()
    except:
        wr=Toplevel()
        window(wr,"300x100","black","")
        Label(wr,text="Values entered are incorrect \n Try
again",bg="black",fg="red").pack()

    else:
        scr4.destroy()
        #show succesful window
        scr5=Toplevel()
        window(scr5,"200x200","white","")
        Label(scr5,text="Registration Succesful",fg='red',bg='white').pack()
        Label(scr5,text="Your patient id is {}.\nPlease remember
it".format(patid),fg="red",bg="white").pack()
        Button(scr5,text="Ok",bg='grey',fg="black",
command=scr5.destroy).pack()

 def patient_register():
    global scr4,namebox,sexbox,dobbox,bloodbox,addressbox
    scr4=Toplevel()
    window(scr4,"300x500","turquoise1","Patient Registration Form")
```

```
Label(scr4,text="Patient Registration Form",fg='blue',bg='white').pack()

Label(scr4,text="Full Name",bg='grey',fg='black').pack()
namebox=Entry(scr4,bg="white",width=20)
namebox.pack()

Label(scr4,text="Sex(M/F)",bg='grey',fg='black').pack()
sexbox=Entry(scr4,bg="white",width=8)
sexbox.pack()

Label(scr4,text="DOB(YYYY/MM/DD)",bg='grey',fg='black').pack()
dobbox=Entry(scr4,bg="white",width=12)
dobbox.pack()

Label(scr4,text="Blood Group",bg='grey',fg='black').pack()
bloodbox=Entry(scr4,bg="white",width=8)
bloodbox.pack()

Label(scr4,text="Address",bg='grey',fg='black').pack()
addressbox=Entry(scr4,bg="white",width=25)
addressbox.pack()
```

```
Button(scr4,text="Register",bg='white',fg='blue',command=registration).pl
ace(x=122,y=450)


 "-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------"

 ##########################################################################
##########################################################################
####################

 "##########################################################################
##########################################################################
##########################################################################
############"

 ''' SETTING AN APPOINTMENT'''


 def confirm():

    global
con,mycursor,patidbox,docidbox,datebox,timebox,problembox,phonebox,
scr6

    p=patidbox.get()

    ph=phonebox.get()

    d=docidbox.get()

    dat=datebox.get()

    time=timebox.get()

    problem=problembox.get()


    global con,mycursor
```

```
    connect()
    mycursor.execute("Select max(AppointmentNo) from Appointment")
    used=mycursor.fetchone()
    maximum=int(used[0])
    appmtid=maximum+1

    try:
        mycursor.execute("Insert into appointment values(
{},{},{},{},'{}','{}','{}')".format(appmtid,p,ph,d,dat,time,problem) )
        con.commit()
    except:
        wr=Toplevel()
        window(wr,"300x100","black","")
        Label(wr,text="Values entered are incorrect \n Try
again",bg="black",fg="red").pack()

    else:
        scr7=Toplevel()
        window(scr7,"450x100","grey","")
        Label(scr7,text="Appointment confirmed.\n Please check your phone
SMS for confirmation message",fg="blue",bg="grey").pack()

Button(scr7,text="Ok",fg="black",bg="white",command=scr7.destroy).pac
k()
        global scr6
```

```
        scr6.destroy()

 def showdocinfo():
    global con,mycursor
    connect()
    mycursor.execute("Select Id,Name,Sex,Department from doctor")
    result=mycursor.fetchall()
    a3=Toplevel()
    window(a3,"500x400","aquamarine","Doctors")
    header="Id\t\t Name\t\t\t\t Sex\t\t Department\t\t\t"
    Label(a3,text=header,fg="black").place(x=10,y=10)
    laby=50
    for row in result:
        answer='{}\t\t {}\t\t\t\t {}\t\t
{}\t\t'.format(row[0],row[1],row[2],row[3])

Label(a3,text=answer,fg="black",bg="aquamarine").place(x=10,y=laby)
        laby+=20
    con.close()

Button(a3,text="Ok",bg="white",fg="black",command=a3.destroy).place(x
=250,y=350)

 def appointment():
    global scr6,patidbox,docidbox,datebox,timebox,problembox,phonebox
```

```
    scr6=Toplevel()
    window(scr6,"700x400","light goldenrod","Set Appointment")
    warning=Label(scr6,text="Please make sure to register first and have
your Patient Id ready. \n If you haven't done so already, please register
below",fg='red',bg='light goldenrod')
    warning.pack()
    Button(scr6,text='Register',
fg='blue',bg='white',command=patient_register).pack()
    Label(scr6,text="Set an
appointment",fg="yellow",bg='grey',font=("Courier",15)).pack()

    global patidbox,docidbox,datebox,timebox,problembox,phonebox

    Label(scr6,text="Patient ID:",fg='blue',bg='light
goldenrod').place(x=90,y=120)
    patidbox=Entry(scr6,bg="white",width=25)
    patidbox.place(x=250,y=120)

    Label(scr6,text="Contact Number:",fg='blue',bg='light
goldenrod').place(x=90,y=145)
    phonebox=Entry(scr6,bg="white",width=25)
    phonebox.place(x=250,y=145)

    Label(scr6,text="Doctor ID:",fg="blue",bg="light
goldenrod").place(x=90,y=170)
```

```
    docidbox=Entry(scr6,bg="white",width=25)
    docidbox.place(x=250,y=170)
    Button(scr6,text="Click for Doctor
information",fg="blue",bg="white",command=showdocinfo).place(x=450,y
=170)

    Label(scr6,text="Date of
appointment\n(YYYY/MM/DD):",fg="blue",bg="light
goldenrod").place(x=90,y=205)
    datebox=Entry(scr6,bg="white",width=25)
    datebox.place(x=250,y=205)

    Label(scr6,text="Time of appointment\n(HH:MM):",fg="blue",bg="light
goldenrod").place(x=90,y=238)
    timebox=Entry(scr6,bg="white",width=25)
    timebox.place(x=250,y=238)

    Label(scr6,text="Brief description of probelm:",fg="blue",bg="light
goldenrod").place(x=90,y=270)
    problembox=Entry(scr6,bg="white",width=40)
    problembox.place(x=250,y=270)

Button(scr6,text="CONFIRM",fg="blue",bg="white",command=confirm).pl
ace(x=280,y=350)
```

```
 '-------------------------------------------------------------------------------------------
-------------------------------------------------'

 "#################################################################
#################################################################
#####################################"

 ''' ANALYSIS  '''



 def analyse():
    global con,mycursor
    connect()
    mycursor.execute("select * from patient")
    data=mycursor.fetchall()
    num=0
    male=0
    female=0
    for i in data:
        num+=1
        if i[2].upper()=="M":
            male+=1
        else:
            female+=1


    #mycursor.execute("create table bloodata as select count(*) as
number,blood_group from patient group by blood_group")
```

```
    con.commit()
    xaxis=["Total","Male","Female"]
    yaxis=[num,male,female]
    plt.bar(xaxis,yaxis,color="pink")
    plt.title("Analysis of Patients")
    plt.xlabel("Patients")
    plt.ylabel("Number")
    plt.show()
```

```
 ##########################################################
##########################################################
##########################################################
##########################################################
##########################################################
##########################################################
##########################################
```

```
 ##########################################################
##########################################################
##########################################################
##################################
```

```
 #main menu
 def mainmenu():
    screen1=Tk()
    screen1.title("Main Menu")
```

```
    screen1.geometry("400x400")
    screen1.configure(bg="seagreen1")
    screen1.resizable(False,False)
    image1=PhotoImage(file="C:\Python\Python3.7\project\\hospital image2.png")
    Label(screen1,image=image1).pack()

    Label(text="Welcome to MY Hospital",fg="snow",bg="dodger blue",font=("Verdana",20)).place(x=30,y=5)
    Label(text="Menu",fg="yellow",bg="blue", width=5).place(x=175,y=40)

    Button(text="Hospital Administration Login",bg="white",fg="blue",command=entry).place(x=117,y=80)
    Button(text="Patient Registration",bg="white",fg="blue",command=patient_register).place(x=143,y=130)
    Button(text="Set an appointment",bg="white",fg="blue",command=appointment).place(x=143,y=190)

Button(text="Analysis",bg="white",fg="blue",command=analyse).place(x=170,y=250)

Button(text="Exit",bg="white",fg="blue",command=screen1.destroy).place(x=170,y=310)
```

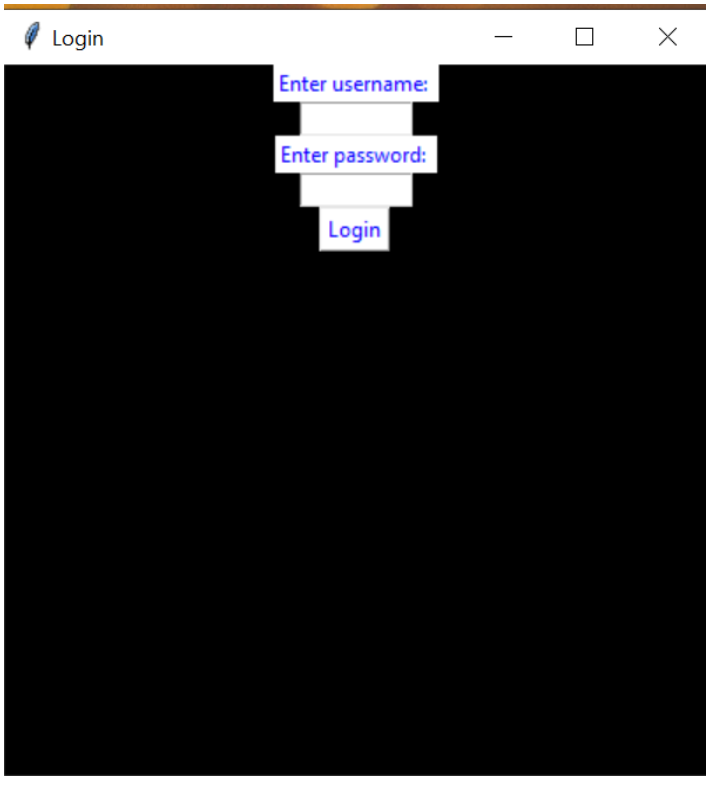
```
    screen1.mainloop()


#main loop
mainmenu()
```

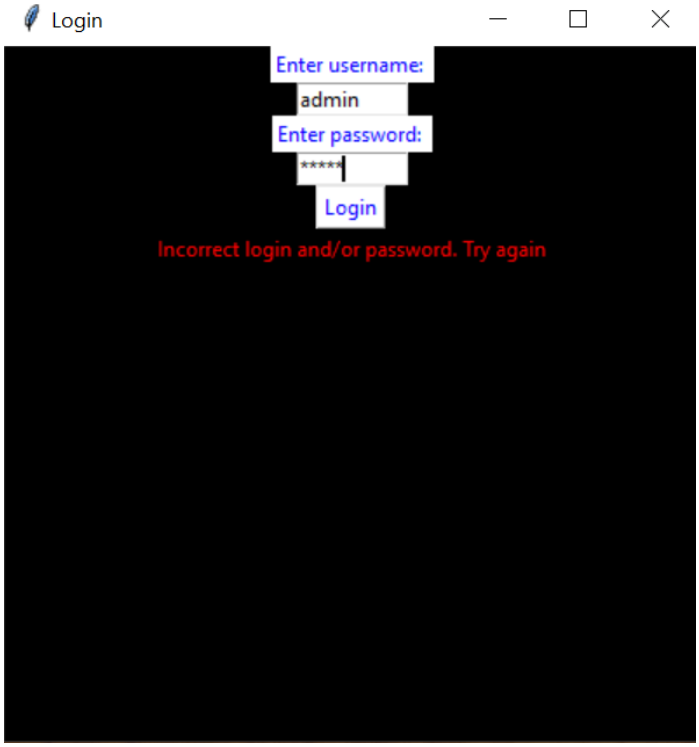
SAMPLE OUTPUT

Main menu

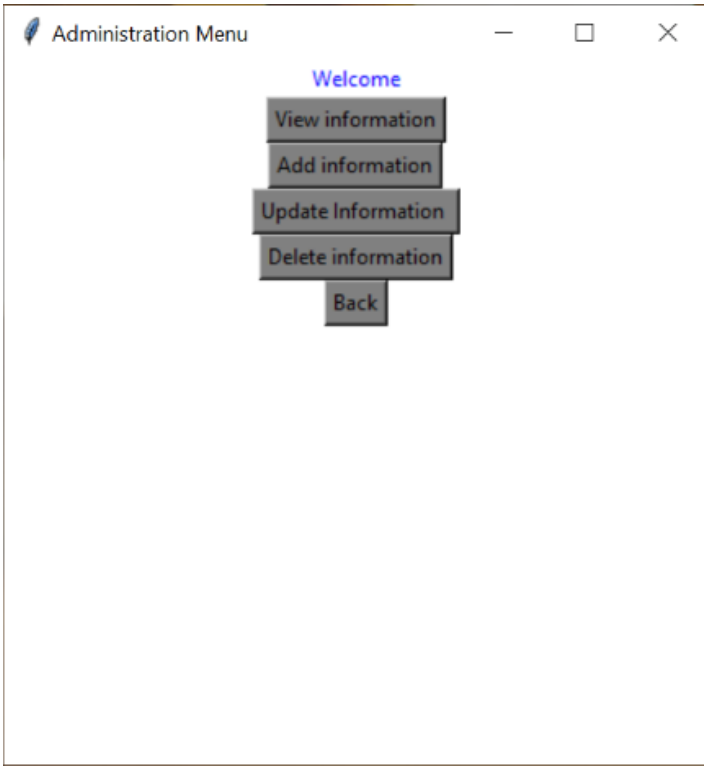
''HOSPITAL ADMINISTRATION LOGIN'' BUTTON



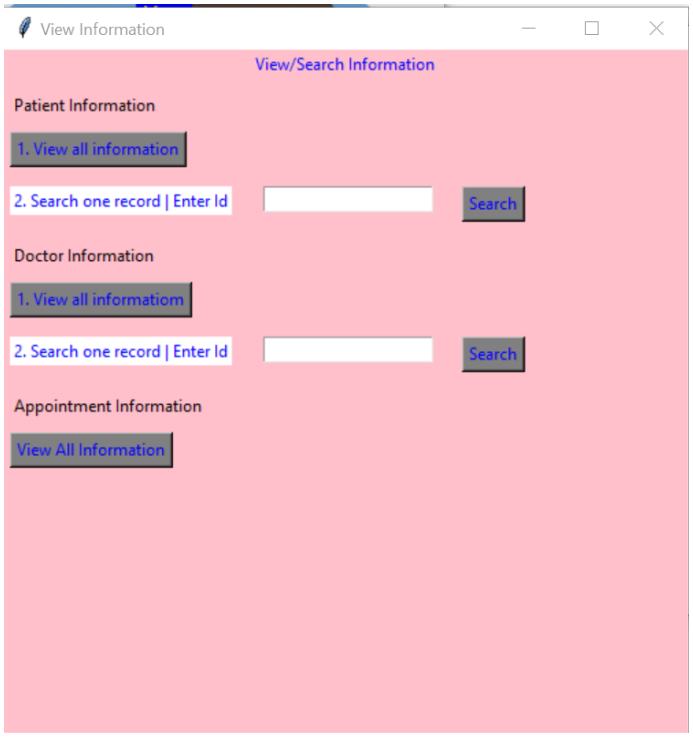
Login screen which appears when we select "Hospital administration login"



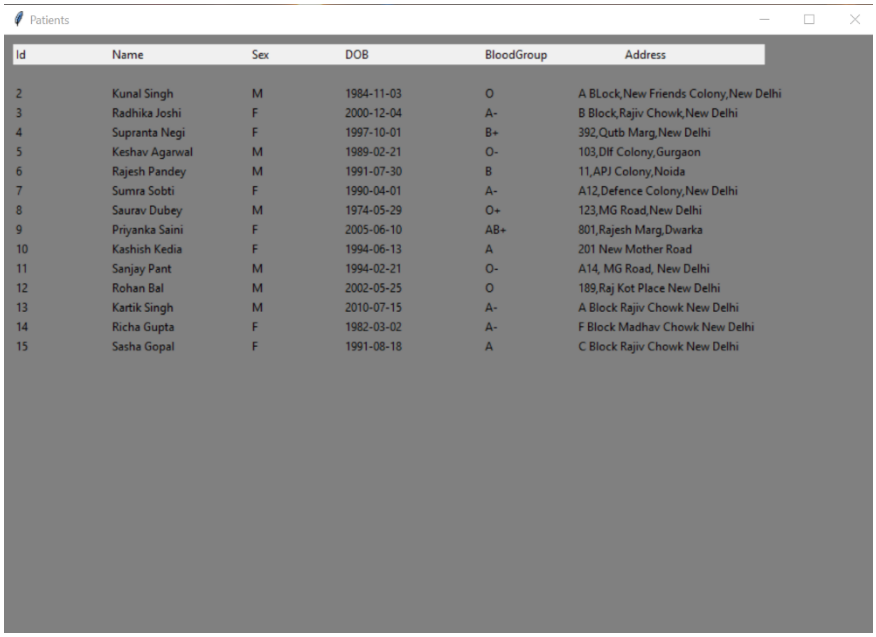
Error message which appears when we enter wrong id/ password



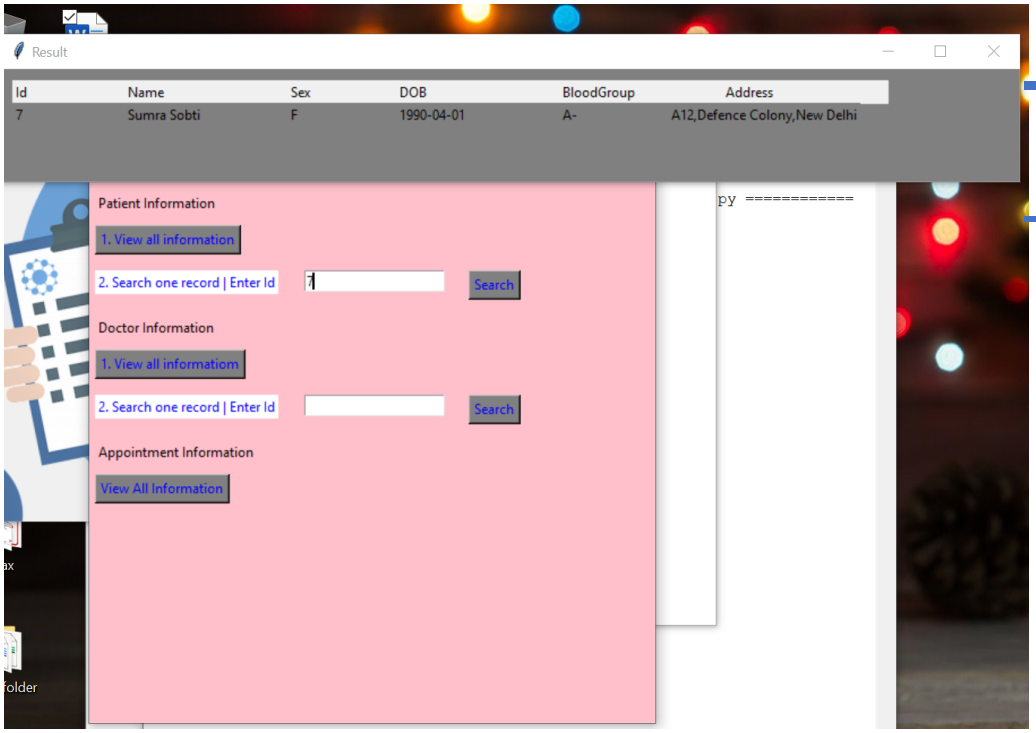
Administration Menu which appears when we enter correct id and password



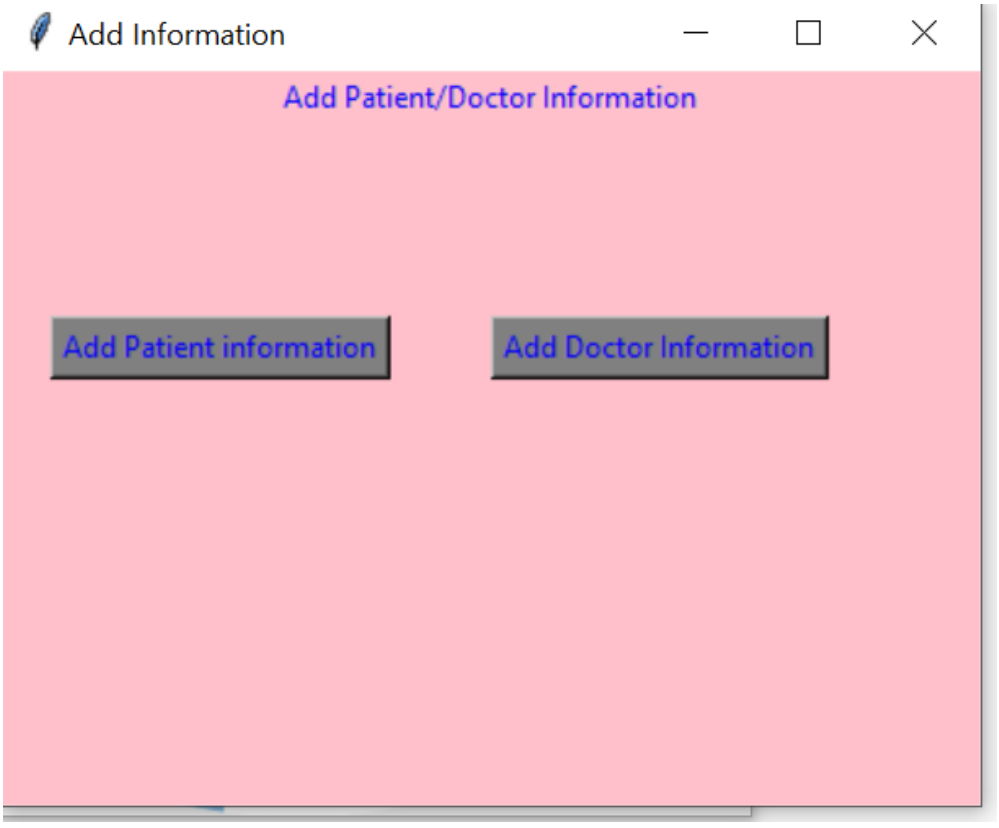
''View information'' window



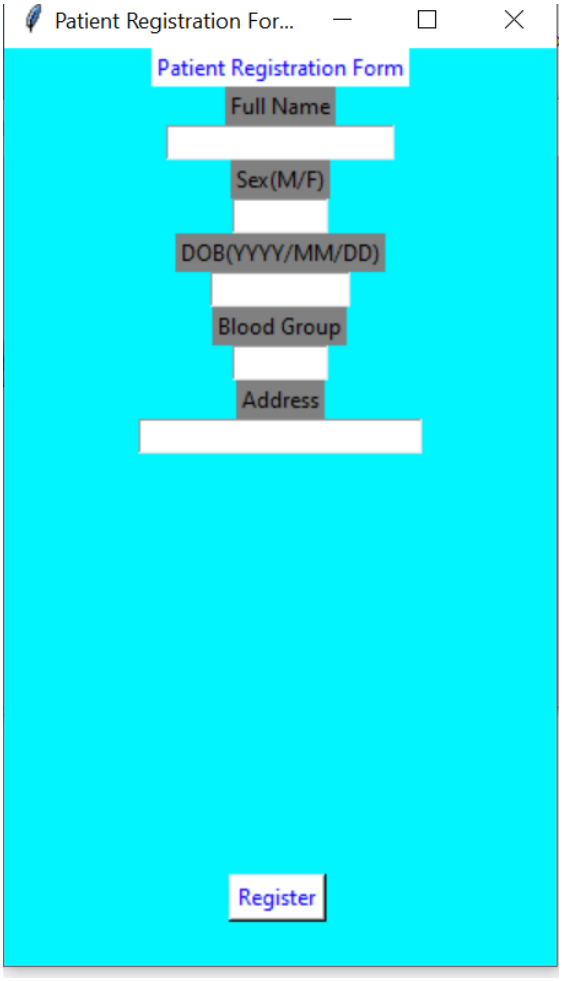
Window which appears when we press "1. View all information " under ''Patient Information''



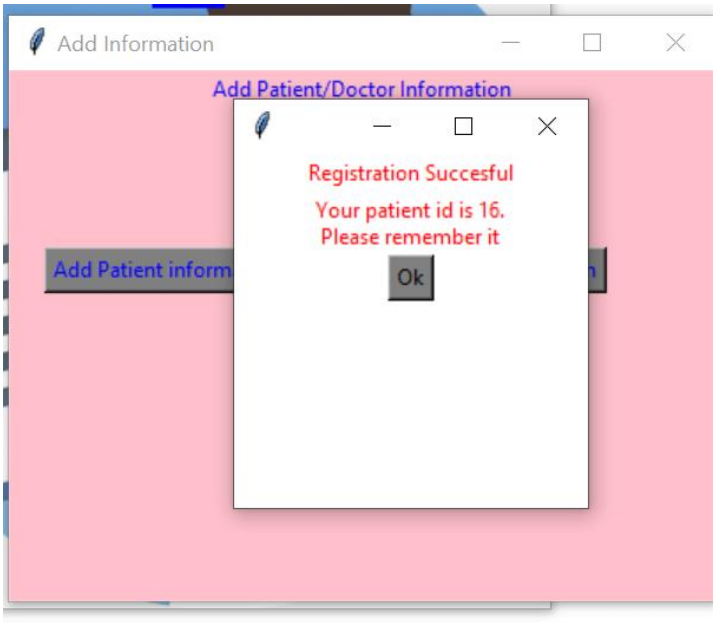
Searching information using "Search 1 record"



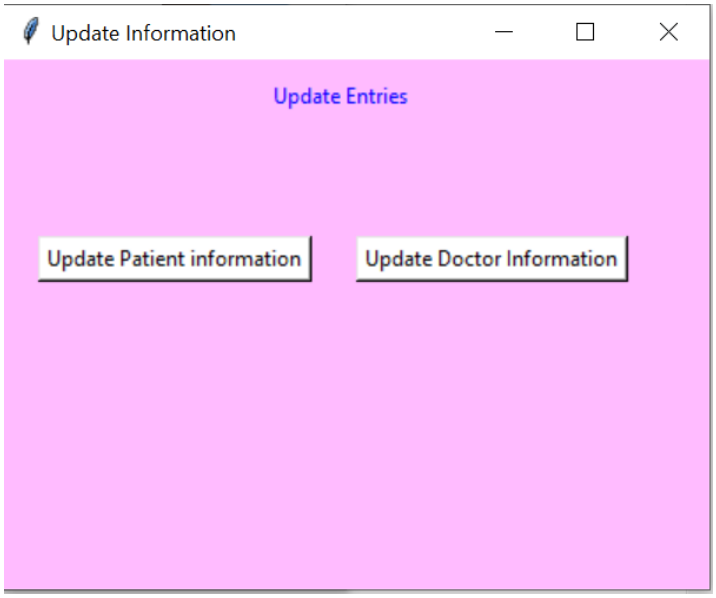
"Add Information Window"



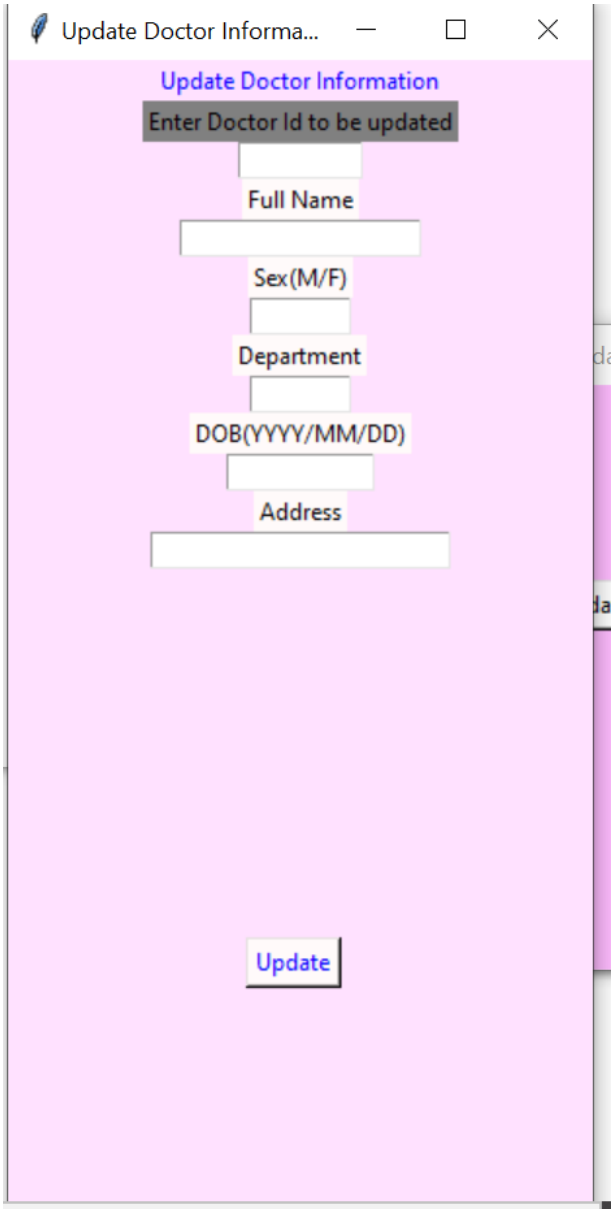
Patient Registration form which appears when we select "Add patient inforamtion".
This form can also be accessed using the "Patient Registration" Button in the main menu.



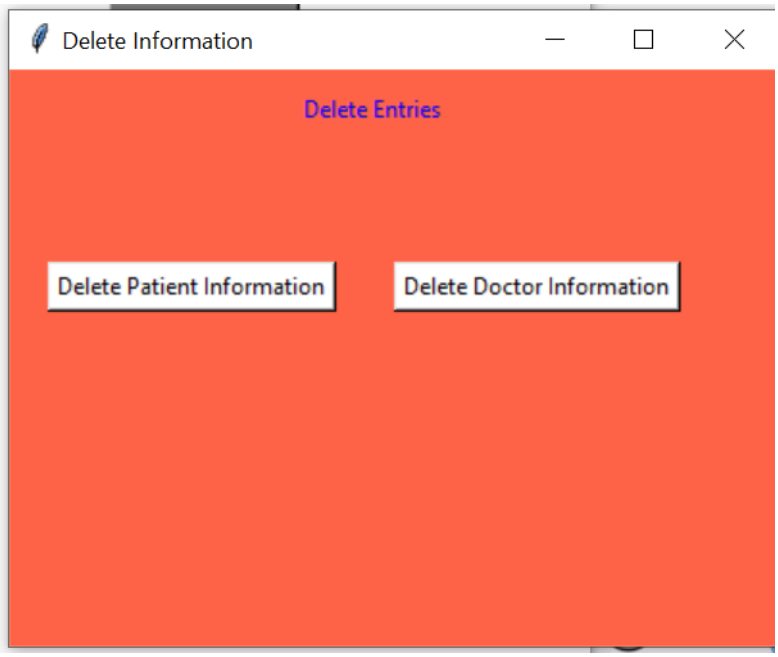
Window which confirms patient registration. It gives the patient a 'patient id', which is needed for making appointments



"Update information" window

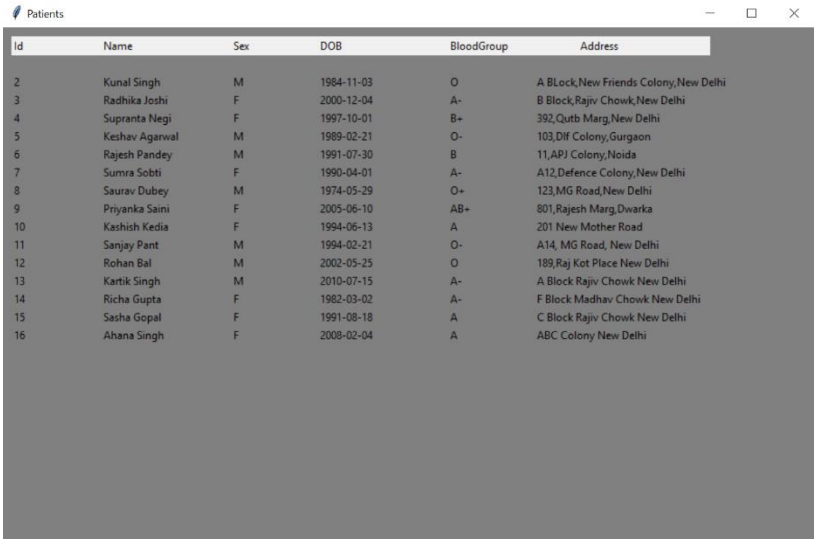


Update Doctor information window

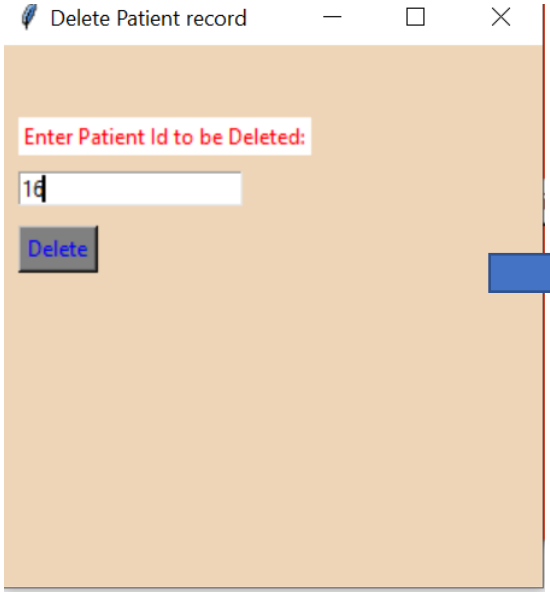


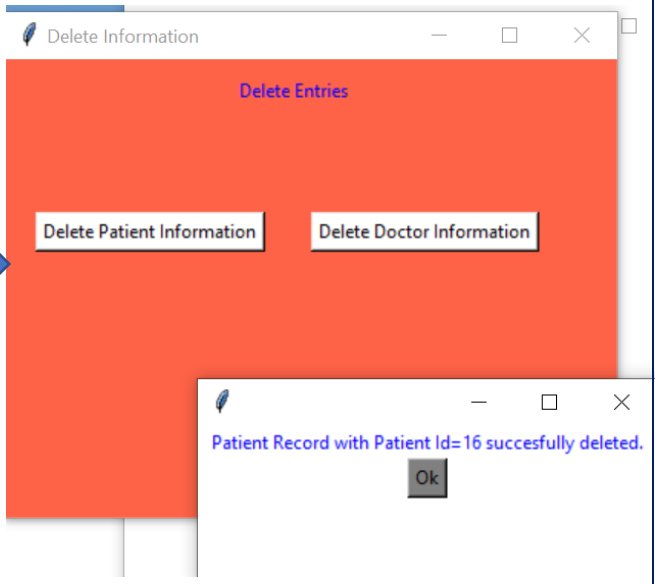
"Delete Information Window"

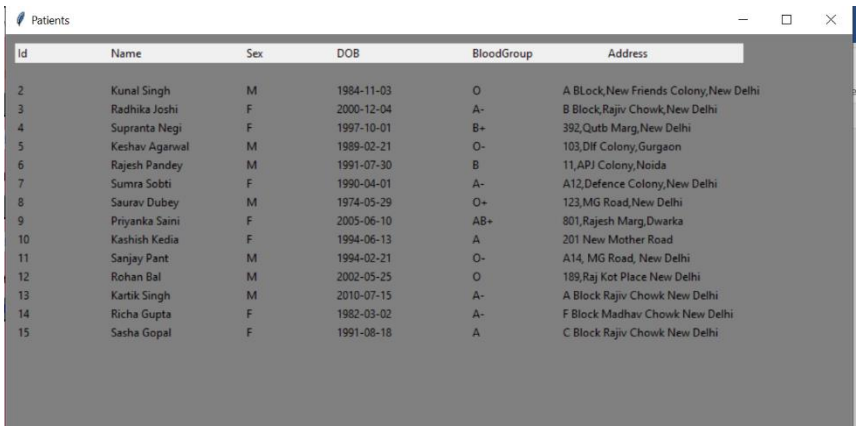
Delete patient record window which appears when we select "Delete Patient Information" button



Original patient information with patient id 16

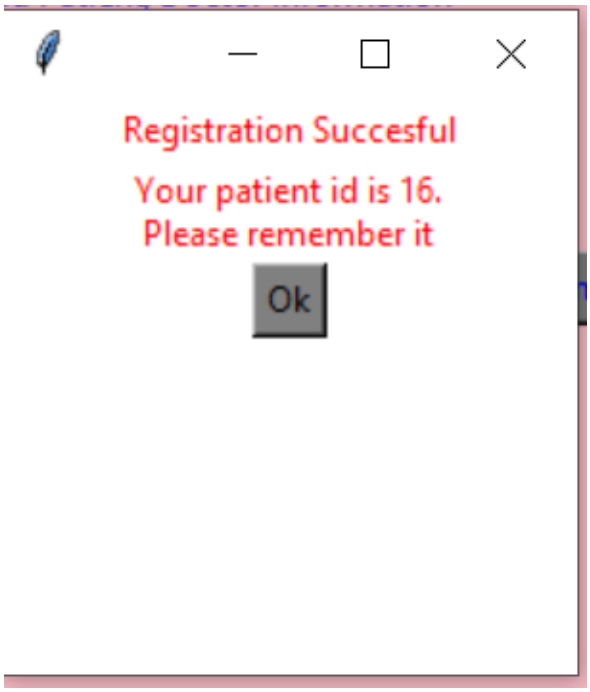




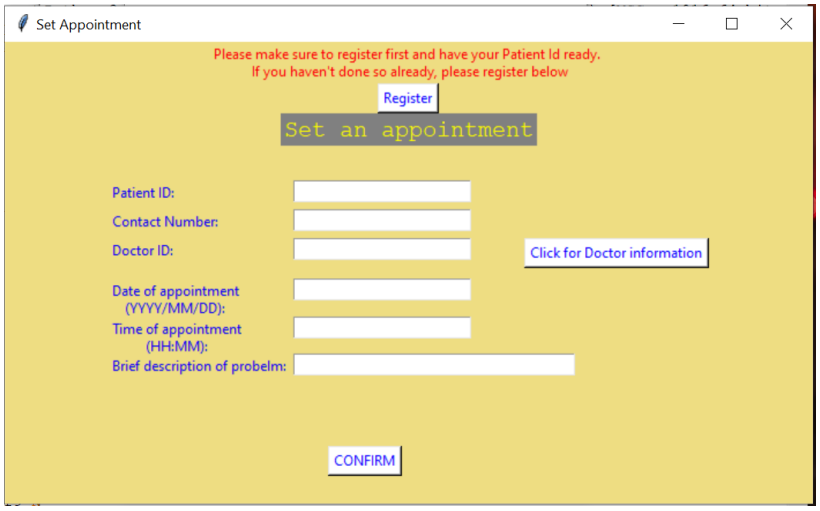


Patient information with no patient id 16 after delete operation is carried out

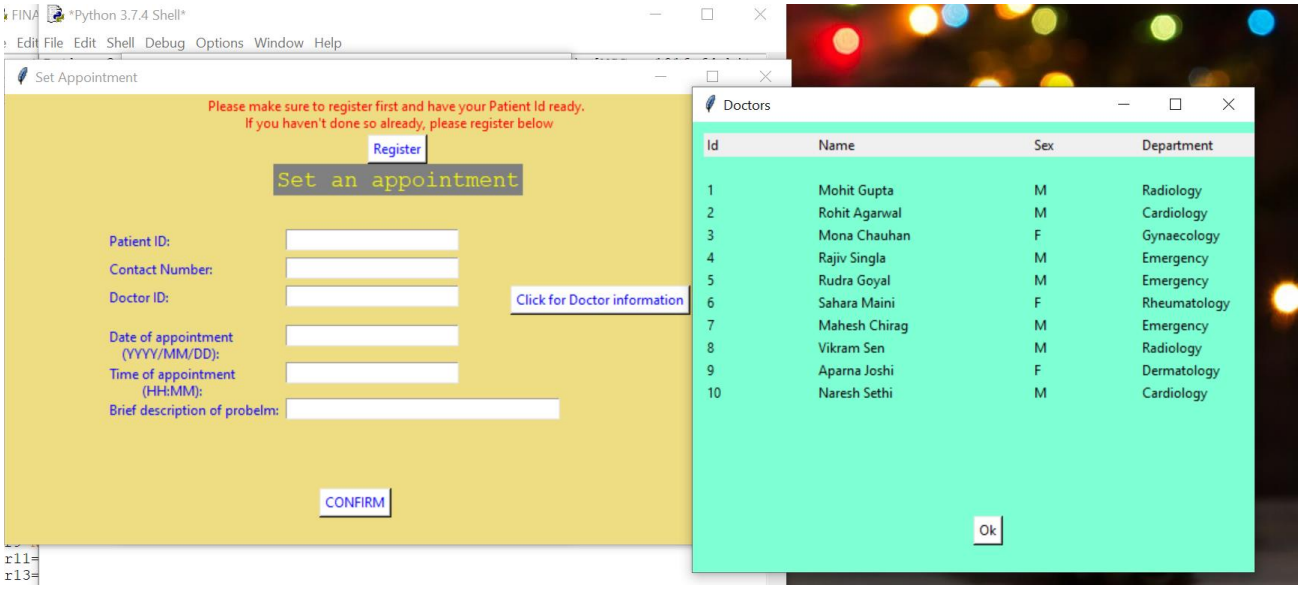
''PATIENT REGISTRATION''' BUTTON



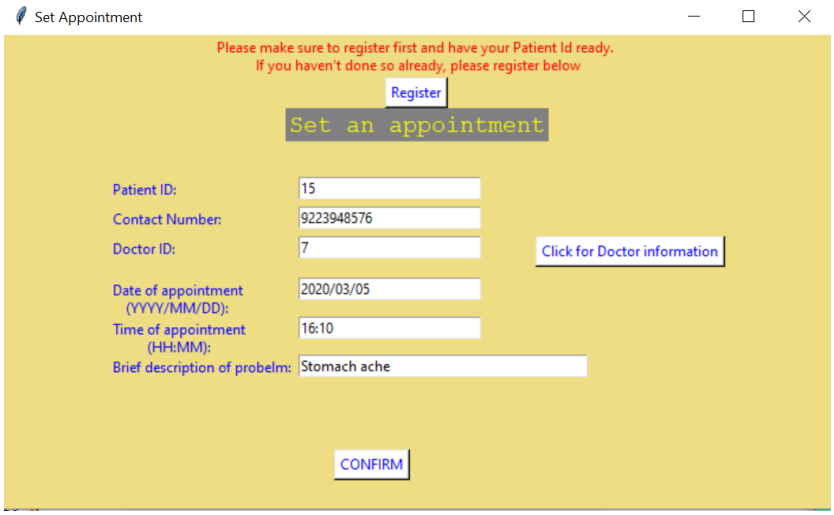
""SET AN APPOINTMENT " BUTTON



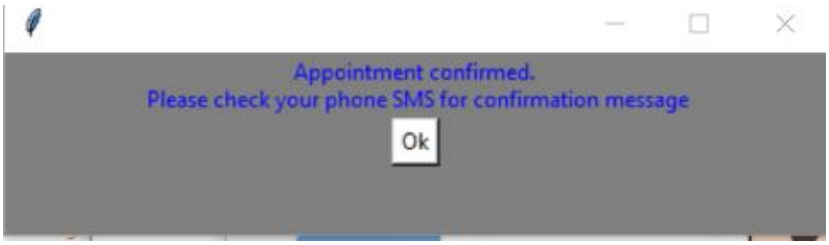
Main appointment window



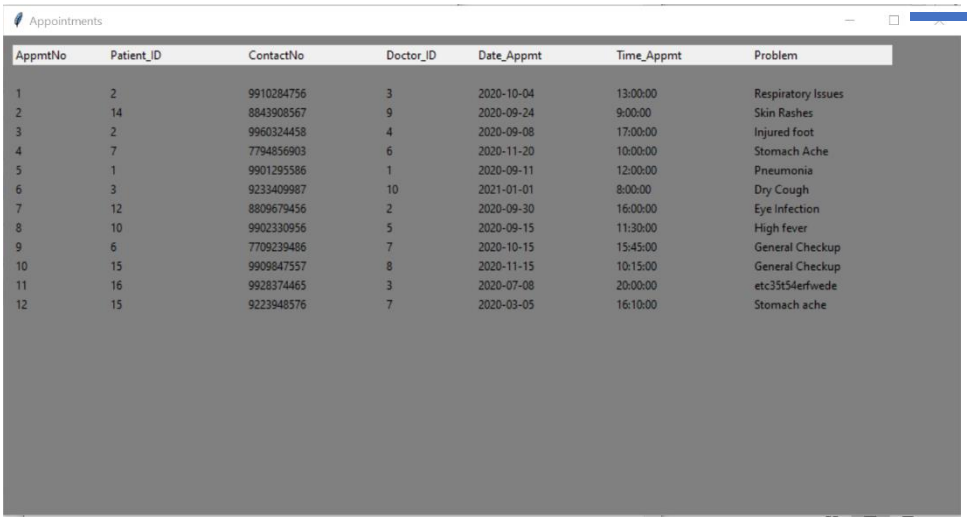
Doctor information box which appears when we press "Click for doctor information" to help patient make appointment



Making a sample appointment



Our appointment is added to the appointment table (which can be viewed through administration login button)



AppmtNo	Patient_ID	ContactNo	Doctor_ID	Date_Appmt	Time_Appmt	Problem
1	2	9910284756	3	2020-10-04	13:00:00	Respiratory Issues
2	14	8843908567	9	2020-09-24	9:00:00	Skin Rashes
3	2	9960324458	4	2020-09-08	17:00:00	Injured foot
4	7	7794856903	6	2020-11-20	10:00:00	Stomach Ache
5	1	9901295586	1	2020-09-11	12:00:00	Pneumonia
6	3	9233409987	10	2021-01-01	8:00:00	Dry Cough
7	12	8809679456	2	2020-09-30	16:00:00	Eye Infection
8	10	9902330956	5	2020-09-15	11:30:00	High fever
9	6	7709239486	7	2020-10-15	15:45:00	General Checkup
10	15	9909847557	8	2020-11-15	10:15:00	General Checkup
11	16	9928374465	3	2020-07-08	20:00:00	etc35t54erfwede
12	15	9223948576	7	2020-03-05	16:10:00	Stomach ache

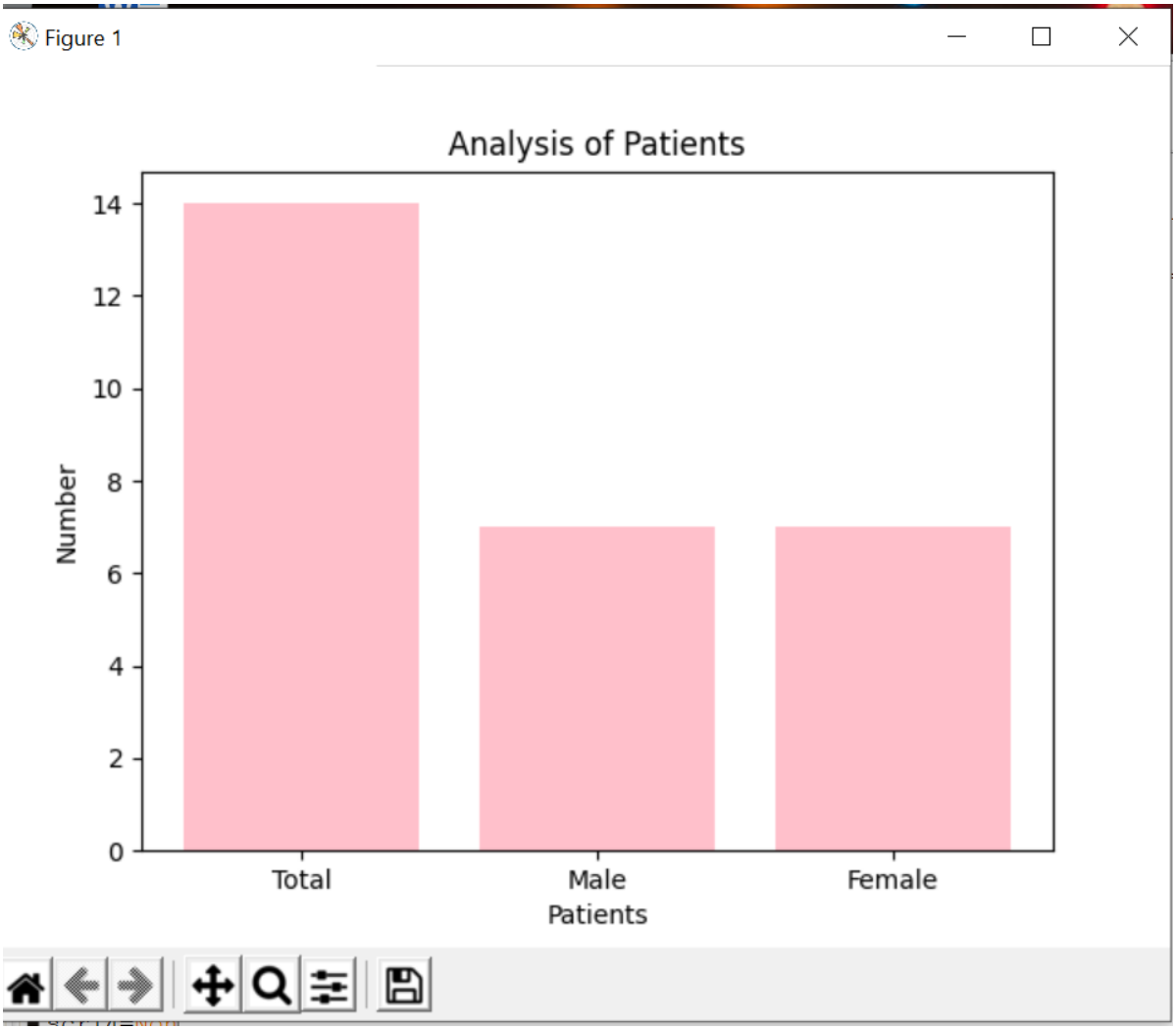


Figure 1
Analysis of Patients
Number
14
12
10
8
6
4
2
0
Total
Male
Female
Patients

LIMITATIONS

1) There is no checking to ensure that one doctor doesn't have simultaneous appointments at the same time and date as the process is very complex
2) The use of "Top levell" object in Tkinter makes it difficult to control the position of the next window.
3) If the main window is accidently closed, the accompanying "Top-level" or "Pop-up" windows also shut down.
4) There is no way for the patient to cancel his own appointment/ registration, it can only be done by the adminsistrator.

BIBLIOGRAPHY

- Computer Science with Python by Sumita Arora
- https://stackoverflow.com
- https://tutorialspoint.com/python
- https://hackerrank.com
- Tkinter tutorial by “thenewboston”