

HOSPITAL MANAGEMENT SYSTEM

A MINI-PROJECT BY:

KAVISH RAGAV S 230701147

KIRAN HARINARAYANA 230701152

in partial fulfillment of the award of the degree

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project "HOSPITAL MANAGEMENT SYSTEM" is the bonafide work of "KAVISH RAGAV S-(230701147), KIRAN HARINARAYANA-(230701152)" who carried out the project work under my supervision.

SUBMITTED FOR PRACTICAL EXAMINATION HELD ON

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to everyone who has contributed to the successful completion of this mini project.

First and foremost, I am deeply thankful to my Professor Mrs. K. Maheshmeena my project advisor, for their invaluable guidance, insightful feedback, and continuous support throughout the duration of this mini project. Their expertise and encouragement have been instrumental in shaping my research and bringing this mini project to completion.

I would also like to express my appreciation to the faculty and staff of the Computer Science and Engineering Department at Rajalakshmi Engineering College for providing the necessary resources and a conducive learning environment. We express our sincere thanks to Dr. P. Kumar, M.E., Ph.D., Professor and Head of the Department Computer Science and Engineering for his guidance and encouragement throughout the project work.

My heartfelt thanks go to my peers and friends for their collaboration, constructive criticism, and moral support.

Thank you all for your contributions, both direct and indirect, to the success of this project. Thank you all for your contributions, both direct and indirect, to the success of this project.

HEALTHPAL

HOSPITAL MANAGEMENT SYSTEM

INDEX

<u>S.NO</u>	<u>CONTENTS</u>	<u>PG NO.</u>
<u>1.</u>	ABSTRACT TO PROJECT	2
<u>2.</u>	SYSTEM REQUIREMENTS	4
<u>3.</u>	BACKEND DETAILS	6
<u>4.</u>	SOURCE CODE	10
<u>5.</u>	OUTPUT	48
<u>6.</u>	MOTIVE	51
<u>7.</u>	BIBLIOGRAPHY	54
<u>8.</u>	LIMITATIONS OF PROJECT	55

ABSTRACT TO PROJECT

The project “Hospital Management System” includes:

- registration of patients
- storing their details into the system
- computerized billing in the pharmacy and labs.

The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room.

User can search availability of a doctor and the details of a patient using the id.

Hospital Management System can be entered using a username and password. It is accessible

either by an administrator or receptionist. Only they can add data into the database.

The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

FRONT END: Tkinter (Python)

SERVER-SIDE SCRIPT: Python

DATABASE: sqlite

SYSTEM REQUIREMENTS

Hardware Requirements:

- Printer to print the required documents of
- the project.
- Compact Drive.
- Processor: Pentium III and above.
- RAM: 256 MB (Minimum).
- Hard-disk: 20 GB (Minimum).

Software Requirements:

- Windows 7 or higher.

- MySQL server 5.5 OR file (as per requirement).
- Python IDLE 3.6 or higher/ Spyder (frontend)
- Microsoft Word 2010 or higher for documentation.

BACKEND DETAILS

Database name: healthpal

Tables in healthpal:

Tables_in_healthpal
appointment
contact_no
employee
patient
treatment

Table name: patient

Field	Type	Null	Key	Default	Extra
PATIENT_ID	int	NO	PRI	NULL	
NAME	varchar(20)	NO		NULL	
SEX	varchar(10)	NO		NULL	
BLOOD_GROUP	varchar(5)	NO		NULL	
DOB	date	NO		NULL	
ADDRESS	varchar(100)	NO		NULL	
CONSULT_TEAM	varchar(50)	NO		NULL	
EMAIL	varchar(20)	NO		NULL	

Field	Type	Null	Key	Default	Extra
EMP_ID	varchar(10)	NO	PRI	NULL	
EMP_NAME	varchar(20)	NO		NULL	
SEX	varchar(10)	NO		NULL	
AGE	int	NO		NULL	
DESIG	varchar(20)	NO		NULL	
SAL	int	NO		NULL	
EXP	varchar(100)	NO		NULL	
EMAIL	varchar(20)	NO		NULL	
PHONE	int	YES		NULL	

Table name: employee

Table name: appointment

Field	Type	Null	Key	Default	Extra
PATIENT_ID	int	NO	MUL	NULL	
EMP_ID	varchar(10)	NO	MUL	NULL	
AP_NO	varchar(10)	NO	PRI	NULL	
AP_TIME	time	YES		NULL	
AP_DATE	date	YES		NULL	
description	varchar(100)	YES		NULL	

Table name: treatment

Field	Type	Null	Key	Default	Extra
PATIENT_ID	int	NO	PRI	NULL	
TREATMENT	varchar(100)	NO		NULL	
TREATMENT_CODE	varchar(30)	NO		NULL	
T_COST	int	NO		NULL	

Table name: contact_no

Field	Type	Null	Key	Default	Extra
PATIENT_ID	int	NO	PRI	NULL	
CONTACTNO	int	NO		NULL	
ALT_CONTACT	int	YES		NULL	

SOURCE CODE

database.py

```
import sqlite3
conn=sqlite3.connect("SQL.db")
print("DATABASE CONNECTION SUCCESSFUL")
#conn.execute("Drop table if EXISTS PATIENT")
#c = conn.cursor()
#conn.execute("""Create table PATIENT
#    (PATIENT_ID int(10) primary key,
#    NAME VARCHAR(20) not null,
#    SEX varchar(10) not null,
#    BLOOD_GROUP varchar(5) not null,
#    DOB date not null,
#    ADDRESS varchar(100) not null,
#    CONSULT_TEAM varchar(50) not null,
#    EMAIL varchar(20) not null
#    )""")
```

```

print("PATIENT TABLE CREATED SUCCESSFULLY")
#conn.execute("Drop table if EXISTS CONTACT_NO")
#c = conn.cursor()
#conn.execute("""CREATE TABLE CONTACT_NO
#    (PATIENT_ID int(10) PRIMARY KEY,
#    CONTACTNO int(15) not null,
#    ALT_CONTACT int(15),
#    FOREIGN KEY(PATIENT_ID) REFERENCES
PATIENT(PATIENT_ID))
#    """)
print("CONTACT_NO TABLE CREATED SUCCESSFULLY")
#conn.execute("Drop table if EXISTS employee")
#conn.execute("""create table employee
#    (EMP_ID varchar(10) primary key,
#    EMP_NAME varchar(20)not null,
#    SEX varchar(10) not null,
#    AGE int(5) not null,
#    DESIG varchar(20) not null,
#    SAL int(10) not null,
#    EXP varchar(100) not null,
#    EMAIL varcahr(20) not null,
#    PHONE int(12))""")

print("EMPLOYEE TABLE CREATED SUCCESSFULLY")
#conn.execute("Drop table if EXISTS TREATMENT")

```

```

#conn.execute("""CREATE TABLE TREATMENT
#    (PATIENT_ID int(10) primary key,
#    TREATMENT varchar(100) not null,
#    TREATMENT_CODE varchar(30) not null,
#    T_COST int(20) not null,
#    FOREIGN KEY(PATIENT_ID) REFERENCES
PATIENT(PATIENT_ID));
#    """)
print("TREATMENT TABLE CREATED SUCCESSFULLY")
#conn.execute("Drop table if EXISTS MEDICINE")
#conn.execute("""CREATE TABLE MEDICINE
#    (PATIENT_ID int(10) primary key,
#    MEDICINE_NAME varchar(100) not null,
#    M_COST int(20) not null,
#    M_QTY int(10) not null,
#    FOREIGN KEY(PATIENT_ID) REFERENCES
PATIENT(PATIENT_ID));
#    """)
print("MEDICINE TABLE CREATED SUCCESSFULLY")

```

```

#conn.execute("Drop table if EXISTS ROOM")
#conn.execute("""Create table ROOM
#    (PATIENT_ID int(10)not NULL ,
#    ROOM_NO varchar(20) PRIMARY KEY ,
#ROOM_TYPE varchar(10) not null,

```

```

# RATE int(10) not null,
# DATE_ADMITTED date,
# DATE_DISCHARGED date NULL,
# FOREIGN KEY(PATIENT_ID) REFERENCES
PATIENT(PATIENT_ID)

#);
# """)

print("ROOM TABLE CREATED SUCCESSFULLY")

#conn.execute("Drop table if EXISTS APPOINTMENT")

c = conn.cursor()

c.execute("""create table appointment
(
    PATIENT_ID int(20) not null,
    EMP_ID varchar(10) not null,
    AP_NO varchar(10) primary key,
    AP_TIME time,
    AP_DATE date,
    description varchar(100),
    FOREIGN KEY(PATIENT_ID) references PATIENT(PATIENT_ID),
    FOREIGN KEY(EMP_ID) references employee(EMP_ID));""")

print("APPOINTMENT TABLE CREATED SUCCESSFULLY")

conn.commit()

conn.close()

```


login.py

```
from tkinter import *
import tkinter.messagebox
from tkinter import ttk
from tkinter import font

from menu import Menu

def main():
    root = Tk()
    app= MainWindow(root)
#MAIN WINDOW FOR LOG IN
```

```

class MainWindow:
    def __init__(self, master):
        self.master = master
        self.master.title("HEALTHPAL")
        self.master.geometry("800x500+0+0")
        self.master.config(bg="powder blue")
        self.frame = Frame(self.master, bg="powder blue")
        self.frame.pack()

        self.Username = StringVar()
        self.Password = StringVar()

        self.lblTitle = Label(self.frame, text = """"HEALTHPAL\n
Your Personal HealthCare Companion""", font="Times 20
bold", bg="powder blue", fg="black")

        self.lblTitle.grid(row = 0 ,column = 0, columnspan=2, pady=40)

        #=====

        self.LoginFrame1 =
Frame(self.frame, width=400, height=80, relief="ridge", bg="cadet
blue", bd=20)

        self.LoginFrame1.grid(row=1, column=0)

        self.LoginFrame2 =
Frame(self.frame, width=400, height=80, relief="ridge", bg="cadet
blue", bd=20)

        self.LoginFrame2.grid(row=2, column=0)

        #=====LABEL AND ENTRY=====

```

```

        self.lblUsername =
Label(self.LoginFrame1,text="Username",font="Helvetica 14
bold",bg="cadet blue",bd=22)

        self.lblUsername.grid(row=0,column=0)

        self.lblUsername = Entry(self.LoginFrame1,font="Helvetica 14
bold",textvariable= self.Username,bd=2)

        self.lblUsername.grid(row=0,column=1)

        self.lblPassword = Label(self.LoginFrame1,text="Password
",font="Helvetica 14 bold",bg="cadet blue",bd=22)

        self.lblPassword .grid(row=1,column=0)

        self.lblPassword = Entry(self.LoginFrame1,font="Helvetica 14
bold",show="*",textvariable= self.Password,bd=2)

        self.lblPassword .grid(row=1,column=1)

        #=====BUTTONS=====

        self.btnLogin = Button(self.LoginFrame2,text = "Login"
,font="Helvetica 10 bold", width =10 ,bg="powder blue",command =
self.Login_system)

        self.btnLogin.grid(row=3,column=0)

        self.btnExit = Button(self.LoginFrame2,text = "Exit" ,font="Helvetica
10 bold", width =10 ,bg="powder blue",command = self.Exit)

        self.btnExit.grid(row=3,column=1)


def Login_system(self):

    S1=(self.Username.get())
    S2=(self.Password.get())
    if(S1=='admin' and S2=='1234'):

```

```

        self.newWindow = Toplevel(self.master)
        self.app = Menu(self.newWindow)
    elif(S1=='master' and S2=='2005'):
        self.newWindow = Toplevel(self.master)
        self.app = Menu(self.newWindow)
    else:
        tkinter.messagebox.askretrycancel("HOSPITAL MANAGEMENT
SYSTEM" , "PLEASE ENTER VALID USERNAME AND PASSWORD")
def Exit(self):
    self.master.destroy()
def new_window(self):
    self.newWindow = Toplevel(self.master)
    self.app = self.Menu(self.newWindow)
if __name__ == "__main__":
    main()

```

menu.py

```

from tkinter import *
import tkinter.messagebox
from tkinter import ttk
from tkinter import font
import sqlite3
from patient_form import Patient
from room_form import Room

```

```
from employee_form import Employee
from appointment_form import Appointment
from billing_form import Billing
conn=sqlite3.connect("SQL.db")
print("DATABASE CONNECTION SUCCESSFUL")
```

```
class Menu:
```

```
    def __init__(self, master):
```

```
        self.master = master
```

```
        self.master.title("HEALTHPAL")
```

```
        self.master.geometry("800x600+0+0")
```

```
        self.master.config(bg="cadet blue")
```

```
        self.frame = Frame(self.master, bg="cadet blue")
```

```
        self.frame.pack()
```

```
        self.lblTitle = Label(self.frame, text = "MAIN MENU", font="Helvetica  
20 bold", bg="cadet blue")
```

```
        self.lblTitle.grid(row = 0 , column = 0, columnspan=2, pady=50)
```

```
        self.LoginFrame =
```

```
Frame(self.frame, width=400, height=80, relief="ridge", bg="cadet  
blue", bd=20)
```

```
        self.LoginFrame.grid(row=1, column=0)
```

```
self.button1 = Button(self.LoginFrame, text = "1.PATIENT  
REGISTRATION", width =30,font="Helvetica 14 bold",bg="cadet  
blue",command=self.Patient_Reg)
```

```
self.button1.grid(row=1,column=0,pady=10)
```

```
self.button2 = Button(self.LoginFrame, text="2.ROOM  
ALLOCATION",width =30,font="Helvetica 14 bold",bg="cadet  
blue",command=self.Room_Allocation)
```

```
self.button2.grid(row=3,column=0,pady=10)
```

```
self.button3 = Button(self.LoginFrame, text="3.EMPLOYEE  
REGISTRATION",width =30,font="Helvetica 14 bold",bg="cadet  
blue",command=self.Employee_Reg)
```

```
self.button3.grid(row=5,column=0,pady=10)
```

```
self.button4 = Button(self.LoginFrame, text="4.BOOK  
APPOINTMENT",width =30,font="Helvetica 14 bold",bg="cadet  
blue",command=self.Appointment_Form)
```

```
self.button4.grid(row=7,column=0,pady=10)
```

```
self.button5 = Button(self.LoginFrame, text="5.PATIENT BILL",width  
=30,font="Helvetica 14 bold",bg="cadet  
blue",command=self.Billing_Form)
```

```
self.button5.grid(row=9,column=0,pady=10)
```

```
self.button6 = Button(self.LoginFrame, text="6.EXIT",width  
=30,font="Helvetica 14 bold",bg="cadet blue",command = self.Exit)
```

```
self.button6.grid(row=11,column=0,pady=10)
```

```
#EXIT FOR MENU
def Exit(self):
    self.master.destroy()

def Patient_Reg(self):
    self.newWindow = Toplevel(self.master)
    self.app = Patient(self.newWindow)

def Room_Allocation(self):
    self.newWindow = Toplevel(self.master)
    self.app = Room(self.newWindow)
def Employee_Reg(self):
    self.newWindow = Toplevel(self.master)
    self.app = Employee(self.newWindow)
def Appointment_Form(self):
    self.newWindow = Toplevel(self.master)
    self.app = Appointment(self.newWindow)

def Billing_Form(self):
    self.newWindow = Toplevel(self.master)
    self.app = Billing(self.newWindow)
```

patient_form.py

```
from tkinter import *
import tkinter.messagebox
```

```
from tkinter import ttk
from tkinter import font
import sqlite3
```

```
conn=sqlite3.connect("SQL.db")
print("DATABASE CONNECTION SUCCESSFUL")
```

```
#PATIENT FORM
```

```
class Patient:
```

```
    def __init__(self, master):
        self.master = master
        self.master.title("HOSPITAL MANAGEMENT SYSTEM")
        self.master.geometry("1500x700+0+0")
        self.master.config(bg="cadet blue")
        self.frame = Frame(self.master, bg="cadet blue")
        self.frame.pack()
```

```
#=====ATTRIBUTES=====
```

```
self.pat_ID=IntVar()
self.pat_name=StringVar()
self.pat_dob=StringVar()
self.pat_address=StringVar()
```



```

self.pat_sex=StringVar()
self.pat_BG=StringVar()
self.pat_email=StringVar()
self.pat_contact=IntVar()
self.pat_contactalt=IntVar()
self.pat_CT=StringVar()

#=====TITLE=====

self.lblTitle = Label(self.frame,text = "PATIENT REGISTRATION
FORM", font="Helvetica 20 bold",bg="cadet blue")

self.lblTitle.grid(row =0 ,column = 0,columnspan=2,pady=50)

#=====FRAME=====

self.LoginFrame =
Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet
blue",bd=20)

self.LoginFrame.grid(row=1,column=0)

self.LoginFrame2 =
Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet
blue",bd=20)

self.LoginFrame2.grid(row=2,column=0)

#=====LABELS=====

self.lblpatid = Label(self.LoginFrame,text="PATIENT
ID",font="Helvetica 14 bold",bg="cadet blue",bd=22)

self.lblpatid.grid(row=0,column=0)

```

```
self.lblpatid = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable= self.pat_ID)
```

```
self.lblpatid.grid(row=0,column=1)
```

```
self.lblPatname = Label(self.LoginFrame,text="PATIENT  
NAME",font="Helvetica 14 bold",bg="cadet blue",bd=22)
```

```
self.lblPatname.grid(row=1,column=0)
```

```
self.lblPatname = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable= self.pat_name)
```

```
self.lblPatname.grid(row=1,column=1)
```

```
self.lblsex = Label(self.LoginFrame,text="SEX",font="Helvetica 14  
bold",bg="cadet blue",bd=22)
```

```
self.lblsex.grid(row=2,column=0)
```

```
self.lblsex = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable= self.pat_sex)
```

```
self.lblsex.grid(row=2,column=1)
```

```
self.lblDOB = Label(self.LoginFrame,text="DOB  
(YYYY-MM-DD)",font="Helvetica 14 bold",bg="cadet blue",bd=22)
```

```
self.lblDOB.grid(row=3,column=0)
```

```
self.lblDOB = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable= self.pat_dob)
```

```
self.lblDOB.grid(row=3,column=1)
```

```
self.lblbgrp = Label(self.LoginFrame,text="BLOOD  
GROUP",font="Helvetica 14 bold",bg="cadet blue",bd=22)
```

```
self.lblbgrp.grid(row=4,column=0)

self.lblbgrp = Entry(self.LoginFrame,font="Helvetica 14
bold",bd=2,textvariable= self.pat_BG)

self.lblbgrp.grid(row=4,column=1)


self.lblCon = Label(self.LoginFrame,text="CONTACT
NUMBER",font="Helvetica 14 bold",bg="cadet blue",bd=22)

self.lblCon.grid(row=0,column=2)

self.lblCon = Entry(self.LoginFrame,font="Helvetica 14
bold",bd=2,textvariable= self.pat_contact)

self.lblCon.grid(row=0,column=3)


self.lblAlt = Label(self.LoginFrame,text="ALTERNATE
CONTACT",font="Helvetica 14 bold",bg="cadet blue",bd=22)

self.lblAlt.grid(row=1,column=2)

self.lblAlt = Entry(self.LoginFrame,font="Helvetica 14
bold",bd=2,textvariable= self.pat_contactalt)

self.lblAlt.grid(row=1,column=3)


self.lbleid = Label(self.LoginFrame,text="EMAIL",font="Helvetica 14
bold",bg="cadet blue",bd=22)

self.lbleid.grid(row=2,column=2)

self.lbleid = Entry(self.LoginFrame,font="Helvetica 14
bold",bd=2,textvariable= self.pat_email)

self.lbleid.grid(row=2,column=3)
```

```
self.lbldoc = Label(self.LoginFrame,text="CONSULTING TEAM /  
DOCTOR",font="Helvetica 14 bold",bg="cadet blue",bd=22)
```

```
self.lbldoc.grid(row=3,column=2)
```

```
self.lbldoc = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable= self.pat_CT)
```

```
self.lbldoc.grid(row=3,column=3)
```

```
self.bladd =  
Label(self.LoginFrame,text="ADDRESS",font="Helvetica 14  
bold",bg="cadet blue",bd=22)
```

```
self.bladd.grid(row=4,column=2)
```

```
self.bladd = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable= self.pat_address)
```

```
self.bladd.grid(row=4,column=3)
```

```
self.button2 = Button(self.LoginFrame2, text="SUBMIT",width  
=10,font="Helvetica 14 bold",bg="cadet blue",command =  
self.INSERT_PAT)
```

```
self.button2.grid(row=3,column=1)
```

```
self.button3 = Button(self.LoginFrame2, text="UPDATE",width  
=10,font="Helvetica 14 bold",bg="cadet blue",command=  
self.UPDATE_PAT)
```

```
self.button3.grid(row=3,column=2)
```

```
self.button4 = Button(self.LoginFrame2, text="DELETE",width  
=10,font="Helvetica 14 bold",bg="cadet blue",command=  
self.D_DISPLAY)
```

```
self.button4.grid(row=3,column=3)
```

```
self.button5 = Button(self.LoginFrame2, text="SEARCH",width  
=10,font="Helvetica 14 bold",bg="cadet blue",command=  
self.S_DISPLAY)
```

```
self.button5.grid(row=3,column=4)
```

```
self.button6 = Button(self.LoginFrame2, text="EXIT",width  
=10,font="Helvetica 14 bold",bg="cadet blue",command = self.Exit)
```

```
self.button6.grid(row=3,column=5)
```

```
def clear(self):
```

```
self.lblpatid.delete(0,'end')
```

```
self.lblPatname.delete(0,'end')
```

```
self.lblsex.delete(0,'end')
```

```
self.lblDOB.delete(0,'end')
```

```
self.lblbgrp.delete(0,'end')
```

```
self.lblCon.delete(0,'end')
```

```
self.lblAlt.delete(0,'end')
```

```
self.lbleid.delete(0,'end')
```

```
self.lbldoc.delete(0,'end')
```

```
self.lbladd.delete(0,'end')
```

```

#INSERT DATA IN PATIENT FORM
def INSERT_PAT(self):
    global pp1, pp2, pp3, pp4, pp5, pp6, pp7, pp8, pp9, pp10, ce1, conn
    conn=sqlite3.connect("SQL.db")
    conn.cursor()
    p1=(self.pat_ID.get())
    p2=(self.pat_name.get())
    p3=(self.pat_sex.get())
    p4=(self.pat_BG.get())
    p5=(self.pat_dob.get())
    p6=(self.pat_contact.get())
    p7=(self.pat_contactalt.get())
    p8=(self.pat_address.get())
    p9=(self.pat_CT.get())
    p10=(self.pat_email.get())
    p = list(conn.execute("SELECT * FROM PATIENT WHERE
PATIENT_ID =?",(p1,)))
    x=len(p)
    if x!=0:
        tkinter.messagebox.showerror("HOSPITAL DATABASE
SYSTEM","PATIENT_ID ALREADY EXISTS")

```

```

else:

    conn.execute('INSERT INTO PATIENT
VALUES(?,?,?,?,?,?,?)',(p1,p2,p3,p4,p5,p8,p9,p10,))

    conn.execute('INSERT OR REPLACE INTO CONTACT_NO
VALUES (?,?,?)',(p1,p6,p7,))

    tkinter.messagebox.showinfo("HOSPITAL DATABASE
SYSTEM","DETAILS INSERTED INTO DATABASE")

    self.clear()

    conn.commit()

#UPDATE DATA IN PATIENT FORM

def UPDATE_PAT(self):

    global u1, u2, u3, u4, u5, u6, u7, u8, u9, u10, ue1, conn

    conn.cursor()

    u1 = (self.pat_ID.get())
    u2 = (self.pat_name.get())
    u3 = (self.pat_sex.get())
    u4 = (self.pat_dob.get())
    u5 = (self.pat_BG.get())
    u6 = (self.pat_contact.get())
    u7 = (self.pat_contactalt.get())
    u8 = (self.pat_email.get())
    u9 = (self.pat_CT.get())
    u10 = (self.pat_address.get())

    conn = sqlite3.connect("SQL.db")

```

```

    p = list(conn.execute("Select * from PATIENT where PATIENT_ID=?",
(u1,)))

    if len(p) != 0:

        conn.execute('UPDATE PATIENT SET
NAME=?,SEX=?,DOB=?,BLOOD_GROUP=?,ADDRESS=?,CONSULT_TE
AM=?,EMAIL=? where PATIENT_ID=?', ( u2, u3, u4, u5, u10, u9, u8,u1,))

        conn.execute('UPDATE CONTACT_NO set
CONTACTNO=?,ALT_CONTACT=? WHERE PATIENT_ID=?', ( u6, u7,u1,))

        tkinter.messagebox.showinfo("HOSPITAL DATABASE SYSTEM",
"DETAILS UPDATED INTO DATABASE")

        self.clear()

        conn.commit()

    else:

        tkinter.messagebox.showerror("HOSPITAL DATABSE SYSTEM",
"PATIENT IS NOT REGISTERED")

    def Exit(self):

        self.master.destroy()

    def D_DISPLAY(self):

        self.newWindow = Toplevel(self.master)

        self.app = DMenu(self.newWindow)

    def S_DISPLAY(self):

        self.newWindow= Toplevel(self.master)

        self.app = SMenu(self.newWindow)

class DMenu:

```



```

def __init__(self, master):
    global inp_d, entry1, DeleteB
    self.master = master
    self.master.title("HOSPITAL MANAGEMENT SYSTEM")
    self.master.geometry("1500x700+0+0")
    self.master.config(bg="cadet blue")
    self.frame = Frame(self.master, bg="cadet blue")
    self.frame.pack()
    self.del_pid = IntVar()

    self.lblTitle = Label(self.frame, text="DELETE WINDOW",
font="Helvetica 20 bold", bg="cadet blue")

    self.lblTitle.grid(row=0, column=0, columnspan=2, pady=50)

    #=====FRAME=====

    self.LoginFrame =
Frame(self.frame, width=400, height=80, relief="ridge", bg="cadet
blue", bd=20)

    self.LoginFrame.grid(row=1, column=0)

    self.LoginFrame2 =
Frame(self.frame, width=400, height=80, relief="ridge", bg="cadet
blue", bd=20)

    self.LoginFrame2.grid(row=2, column=0)

    #=====LABELS=====

    self.lblpatid = Label(self.LoginFrame, text="ENTER PATIENT ID TO
DELETE", font="Helvetica 14 bold", bg="cadet blue", bd=22)

    self.lblpatid.grid(row=0, column=0)

    self.lblpatid = Entry(self.LoginFrame, font="Helvetica 14
bold", bd=2, textvariable= self.del_pid)

```

```

self.lblpatid.grid(row=0,column=1)

self.DeleteB = Button(self.LoginFrame2, text="DELETE",width
=10,font="Helvetica 14 bold",bg="cadet blue",command =
self.DELETE_PAT)

self.DeleteB.grid(row=3,column=1)
def DELETE_PAT(self):
    global inp_d,del_pid
    c1= conn.cursor()
    inp_d = (self.del_pid.get())
    p=list(conn.execute("select * from PATIENT where PATIENT_ID=?",
(inp_d,)))
    if (len(p)==0):
        tkinter.messagebox.showerror("HOSPITAL DATABASE
SYSTEM","PATIENT RECORD NOT FOUND")
    else:
        conn.execute('DELETE FROM PATIENT where
PATIENT_ID=?',(inp_d,))
        tkinter.messagebox.showinfo("HOSPITAL DATABASE SYSTEM",
"DETAILS DELETED FROM DATABASE")
        conn.commit()
class SMenu:
    def __init__(self, master):
        global inp_s,s_pid,SearchB
        self.master = master
        self.master.title("HOSPITAL MANAGEMENT SYSTEM")

```

```

self.master.geometry("1500x700+0+0")
self.master.config(bg="cadet blue")
self.frame = Frame(self.master,bg="cadet blue")
self.frame.pack()
self.s_pid=IntVar()

self.lblTitle = Label(self.frame,text = "SEARCH WINDOW",
font="Helvetica 20 bold",bg="cadet blue")

self.lblTitle.grid(row =0 ,column = 0,columnspan=2,pady=25)

#=====FRAME=====

self.LoginFrame =
Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet
blue",bd=20)

self.LoginFrame.grid(row=1,column=0)

self.LoginFrame2 =
Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet
blue",bd=20)

self.LoginFrame2.grid(row=2,column=0)


#=====LABELS=====

self.lblpatid = Label(self.LoginFrame,text="ENTER PATIENT ID TO
SEARCH",font="Helvetica 14 bold",bg="cadet blue",bd=22)

self.lblpatid.grid(row=0,column=0)

self.lblpatid= Entry(self.LoginFrame,font="Helvetica 14
bold",bd=2,textvariable= self.s_pid)

self.lblpatid.grid(row=0,column=1)

```

```

        self.SearchB = Button(self.LoginFrame2, text="SEARCH",width
=10,font="Helvetica 14 bold",bg="cadet blue",command =
self.SEARCH_PAT)

        self.SearchB.grid(row=0,column=1)


    def SEARCH_PAT(self):

        global
inp_s,s_pid,errorS,t,i,q,dis1,dis2,dis3,dis4,dis5,dis6,dis7,dis8,dis9,dis10,l
1,l2,l3,l4,l5,l6,l7,l8,l9,l10

        c1=conn.cursor()

        inp_s=(self.s_pid.get())

        p=list(conn.execute('select * from PATIENT where
PATIENT_ID=?',(inp_s,)))

        if (len(p)==0):

            tkinter.messagebox.showerror("HOSPITAL DATABSE
SYSTEM","PATIENT RECORD NOT FOUND")


        else:

            t=c1.execute('SELECT * FROM PATIENT NATURAL JOIN
CONTACT_NO where PATIENT_ID=?',(inp_s,));

            for i in t:

                self.l1 = Label(self.LoginFrame,text="PATIENT
ID",font="Helvetica 14 bold",bg="cadet blue",bd=22)

                self.l1.grid(row=1,column=0)

                self.dis1= Label(self.LoginFrame,font="Helvetica 14
bold",bd=2,bg="cadet blue",text=i[0])

                self.dis1.grid(row=1,column=1)

```

```
self.l2 = Label(self.LoginFrame,text="PATIENT  
NAME",font="Helvetica 14 bold",bg="cadet blue",bd=22)
```

```
self.l2.grid(row=2,column=0)
```

```
self.dis2=Label(self.LoginFrame,font="Helvetica 14  
bold",bd=2,bg="cadet blue",text=i[1])
```

```
self.dis2.grid(row=2,column=1)
```

```
self.l3 = Label(self.LoginFrame,text="SEX",font="Helvetica  
14 bold",bg="cadet blue",bd=22)
```

```
self.l3.grid(row=3,column=0)
```

```
self.dis3 = Label(self.LoginFrame,font="Helvetica 14  
bold",bg="cadet blue",bd=2,text=i[2])
```

```
self.dis3.grid(row=3,column=1)
```

```
self.l4 = Label(self.LoginFrame,text="DOB  
(YYYY-MM-DD)",font="Helvetica 14 bold",bg="cadet blue",bd=22)
```

```
self.l4.grid(row=4,column=0)
```

```
self.dis4= Label(self.LoginFrame,font="Helvetica 14  
bold",bg="cadet blue",bd=2,text=i[3])
```

```
self.dis4.grid(row=4,column=1)
```

```
self.l5 = Label(self.LoginFrame,text="BLOOD  
GROUP",font="Helvetica 14 bold",bg="cadet blue",bd=22)
```

```
self.l5.grid(row=5,column=0)
```

```
self.dis5 = Label(self.LoginFrame,font="Helvetica 14  
bold",bg="cadet blue",bd=2,text=i[4])
```

```
self.dis5.grid(row=5,column=1)
```

```
self.l6 =  
Label(self.LoginFrame,text="ADDRESS",font="Helvetica 14  
bold",bg="cadet blue",bd=22)  
self.l6.grid(row=1,column=2)  
self.dis6 = Label(self.LoginFrame,font="Helvetica 14  
bold",bg="cadet blue",bd=2,text=i[5])  
self.dis6.grid(row=1,column=3)
```

```
self.l7 = Label(self.LoginFrame,text="CONSULTING TEAM /  
DOCTOR",font="Helvetica 14 bold",bg="cadet blue",bd=22)  
self.l7.grid(row=2,column=2)  
self.dis7 = Label(self.LoginFrame,font="Helvetica 14  
bold",bd=2,bg="cadet blue",text=i[6])  
self.dis7.grid(row=2,column=3)
```

```
self.l8 =  
Label(self.LoginFrame,text="EMAIL",font="Helvetica 14  
bold",bg="cadet blue",bd=22)  
self.l8.grid(row=3,column=2)  
self.dis8 = Label(self.LoginFrame,font="Helvetica 14  
bold",bd=2,bg="cadet blue",text=i[7])  
self.dis8.grid(row=3,column=3)
```

```
self.l9 = Label(self.LoginFrame,text="CONTACT  
NUMBER",font="Helvetica 14 bold",bg="cadet blue",bd=22)
```

```
self.l9.grid(row=4,column=2)
self.dis9 = Label(self.LoginFrame,font="Helvetica 14
bold",bd=2,bg="cadet blue",text=i[8])
self.dis9.grid(row=4,column=3)

self.l10 = Label(self.LoginFrame,text="ALTERNATE
CONTACT",font="Helvetica 14 bold",bg="cadet blue",bd=22)
self.l10.grid(row=5,column=2)
self.dis10 = Label(self.LoginFrame,font="Helvetica 14
bold",bd=2,bg="cadet blue",text=i[9])
self.dis10.grid(row=5,column=3)
```

appointment_form.py

```
from tkinter import *
import tkinter.messagebox
from tkinter import ttk
from tkinter import font
import sqlite3

conn=sqlite3.connect("SQL.db")
print("DATABASE CONNECTION SUCCESSFUL")

class Appointment:
    def __init__(self, master):
        self.master = master
        self.master.title("HOSPITAL MANAGEMENT SYSTEM")
        self.master.geometry("1500x700+0+0")
        self.master.config(bg="cadet blue")
        self.frame = Frame(self.master, bg="cadet blue")
        self.frame.pack()

        #=====ATTRIBUTES=====

        self.pat_ID=IntVar()
        self.emp_ID=StringVar()
        self.ap_no=StringVar()
```



```

self.ap_time=StringVar()
self.ap_date=StringVar()
self.des=StringVar()

#=====TITLE=====

self.lblTitle = Label(self.frame,text = "APPOINTMENT FORM",
font="Helvetica 20 bold",bg="cadet blue")

self.lblTitle.grid(row =0 ,column = 0,columnspan=2,pady=50)

#=====FRAME=====

self.LoginFrame =
Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet
blue",bd=20)

self.LoginFrame.grid(row=1,column=0)


self.LoginFrame2 =
Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet
blue",bd=20)

self.LoginFrame2.grid(row=2,column=0)

#=====LABELS=====

self.lblpid = Label(self.LoginFrame,text="PATIENT
ID",font="Helvetica 14 bold",bg="cadet blue",bd=22)

self.lblpid.grid(row=0,column=0)

self.lblpid = Entry(self.LoginFrame,font="Helvetica 14
bold",bd=2,textvariable= self.pat_ID)

self.lblpid.grid(row=0,column=1)

```

```
self.lblidid = Label(self.LoginFrame,text="DOCTOR  
ID",font="Helvetica 14 bold",bg="cadet blue",bd=22)  
self.lblidid.grid(row=1,column=0)  
self.lblidid = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable=self.emp_ID )  
self.lblidid.grid(row=1,column=1)
```

```
self.lblap = Label(self.LoginFrame,text="APPOINTMENT  
NO",font="Helvetica 14 bold",bg="cadet blue",bd=22)  
self.lblap.grid(row=2,column=0)  
self.lblap = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable=self.ap_no )  
self.lblap.grid(row=2,column=1)
```

```
self.lblapt = Label(self.LoginFrame,text="APPOINTMENT  
TIME(HH:MM:SS)",font="Helvetica 14 bold",bg="cadet blue",bd=22)  
self.lblapt.grid(row=0,column=2)  
self.lblapt = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable=self.ap_time )  
self.lblapt.grid(row=0,column=3)
```

```
self.lblapd = Label(self.LoginFrame,text="APPOINTMENT  
DATE(YYYY-MM-DD)",font="Helvetica 14 bold",bg="cadet blue",bd=22)  
self.lblapd.grid(row=1,column=2)  
self.lblapd = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable= self.ap_date)
```

```
self.lblapd.grid(row=1,column=3)
```

```
self.lbldes =  
Label(self.LoginFrame,text="DESCRIPTION",font="Helvetica 14  
bold",bg="cadet blue",bd=22)
```

```
self.lbldes.grid(row=2,column=2)
```

```
self.lbldes = Entry(self.LoginFrame,font="Helvetica 14  
bold",bd=2,textvariable=self.des)
```

```
self.lbldes.grid(row=2,column=3)
```

```
self.button2 = Button(self.LoginFrame2, text="SAVE",width  
=10,font="Helvetica 14 bold",bg="cadet blue",command =  
self.INSERT_AP)
```

```
self.button2.grid(row=3,column=1)
```

```
self.button3 = Button(self.LoginFrame2, text="DELETE",width  
=10,font="Helvetica 14 bold",bg="cadet blue",command=  
self.DE_AP_DISPLAY)
```

```
self.button3.grid(row=3,column=2)
```

```
self.button3 = Button(self.LoginFrame2, text="SEARCH  
APPOINTMENTS",width =20,font="Helvetica 14 bold",bg="cadet  
blue",command= self.S_AP_DISPLAY)
```

```
self.button3.grid(row=3,column=3)
```

```
self.button6 = Button(self.LoginFrame2, text="EXIT",width
=10,font="Helvetica 14 bold",bg="cadet blue",command = self.Exit)
self.button6.grid(row=3,column=4)
```

```
def Exit(self):
```

```
    self.master.destroy()
```

```
def INSERT_AP(self):
```

```
    global e1,e2,e3,e4,e5,e6,var
```

```
    e1=(self.pat_ID.get())
```

```
    e2=(self.emp_ID.get())
```

```
    e3=(self.ap_no.get())
```

```
    e4=(self.ap_time.get())
```

```
    e5=(self.ap_date.get())
```

```
    e6=(self.des.get())
```

```
    conn = sqlite3.connect("SQL.db")
```

```
    p = list(conn.execute("SELECT * FROM appointment WHERE
AP_NO =?",(e3,)))
```

```
    x=len(p)
```

```
    if x!=0:
```

```
        tkinter.messagebox.showerror("HOSPITAL DATABASE SYSTEM",
"APPOINTMENT ALREADY EXISTS")
```

```
    else:
```

```
        conn.execute("Insert into appointment
values(?,?,?,?,?,?)",(e1,e2,e3,e4,e5,e6,))
```

```
tkinter.messagebox.showinfo("Hospital DATABASE SYSTEM",  
"APPOINTMENT SET SUCCSESSFULLY")
```

```
conn.commit()
```

```
def DE_AP_DISPLAY(self):
```

```
    self.newWindow = Toplevel(self.master)
```

```
    self.app = DEL_AP(self.newWindow)
```

```
def S_AP_DISPLAY(self):
```

```
    self.newWindow= Toplevel(self.master)
```

```
    self.app = SEA_AP(self.newWindow)
```

```
class DEL_AP:
```

```
    def __init__(self, master):
```

```
        global de1_ap, de
```

```
        self.master = master
```

```
        self.master.title("HOSPITAL MANAGEMENT SYSTEM")
```

```
        self.master.geometry("1500x700+0+0")
```

```
        self.master.config(bg="cadet blue")
```

```
        self.frame = Frame(self.master, bg="cadet blue")
```

```
        self.frame.pack()
```

```
        self.de1_ap=StringVar()
```

```
        self.lblTitle = Label(self.frame, text = "DELETE APPOINTMENT  
WINDOW", font="Helvetica 20 bold", bg="cadet blue")
```

```
        self.lblTitle.grid(row =0 ,column = 0, columnspan=2, pady=50)
```

```

#=====FRAME=====

self.LoginFrame =
Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet
blue",bd=20)

self.LoginFrame.grid(row=1,column=0)

self.LoginFrame2 =
Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet
blue",bd=20)

self.LoginFrame2.grid(row=2,column=0)

#=====LABELS=====

self.lblpatid = Label(self.LoginFrame,text="ENTER APPOINTMENT
NO TO DELETE",font="Helvetica 14 bold",bg="cadet blue",bd=22)

self.lblpatid.grid(row=0,column=0)

self.lblpatid= Entry(self.LoginFrame,font="Helvetica 14
bold",bd=2,textvariable= self.de1_ap)

self.lblpatid.grid(row=0,column=1)


self.DeleteB = Button(self.LoginFrame2, text="DELETE",width
=10,font="Helvetica 14 bold",bg="cadet blue",command =
self.DELETE_AP)

self.DeleteB.grid(row=3,column=1)


def DELETE_AP(self):
    global inp_d
    inp_d = str(self.de1_ap.get())
    conn = sqlite3.connect("SQL.db")

```

```

        v=list(conn.execute("select * from appointment where AP_NO=?",
(inp_d ,)))
        if (len(v)==0):
            tkinter.messagebox.showerror("HOSPITAL DATABSE SYSTEM",
"PATIENT APPOINTMENT NOT FIXED")
        else:
            conn.execute('DELETE FROM APPOINTMENT where
AP_NO=?',(inp_d ,))
            tkinter.messagebox.showinfo("Hospital DATABASE SYSTEM",
"PATIENT APPOINTMENT DELETED")
            conn.commit()

```

```

class SEA_AP:

```

```

    def __init__(self, master):
        global inp_s, entry, SearchB
        self.master = master
        self.master.title("HOSPITAL MANAGEMENT SYSTEM")
        self.master.geometry("1500x700+0+0")
        self.master.config(bg="cadet blue")
        self.frame = Frame(self.master, bg="cadet blue")
        self.frame.pack()
        self.entry=StringVar()
        self.lblTitle = Label(self.frame, text = "SEARCH APPOINTMENT
WINDOW", font="Helvetica 20 bold", bg="cadet blue")
        self.lblTitle.grid(row =0 ,column = 0, columnspan=2, pady=25)
        #=====FRAME=====

```

```

        self.LoginFrame =
Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet
blue",bd=20)

        self.LoginFrame.grid(row=1,column=0)

        self.LoginFrame2 =
Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet
blue",bd=20)

        self.LoginFrame2.grid(row=2,column=0)


#=====LABELS=====

        self.lblpatid = Label(self.LoginFrame,text="ENTER DATE TO VIEW
APPOINTMENTS(YYYY-MM-DD)",font="Helvetica 14 bold",bg="cadet
blue",bd=22)

        self.lblpatid.grid(row=0,column=0)

        self.lblpatid= Entry(self.LoginFrame,font="Helvetica 14
bold",bd=2,textvariable= self.entry)

        self.lblpatid.grid(row=0,column=1)


        self.SearchB = Button(self.LoginFrame2, text="SEARCH",width
=10,font="Helvetica 14 bold",bg="cadet blue",command =
self.SEARCH_AP)

        self.SearchB.grid(row=0,column=1)


def SEARCH_AP(self):

    global
inp_s,entry,errorS,t,i,q,dis1,dis2,dis3,dis4,dis5,dis6,dis7,dis8,dis9,dis10,l
1,l2,l3,l4,l5,l6,l7,l8,l9,l10

    c1=conn.cursor()

```



```

ap=(self.entry.get())

p = list(c1.execute("select * from appointment where AP_DATE=?",
(ap,)))

if (len(p) == 0):

    tkinter.messagebox.showerror("HOSPITAL DATABASE
SYSTEM","NO APPOINTMENT FOR TODAY")

else:

    t=c1.execute('SELECT
PATIENT_ID,NAME,AP_NO,EMP_ID,AP_DATE,AP_TIME FROM
PATIENT NATURAL JOIN appointment where AP_DATE=?',(ap,))

    for i in t:

        self.l1 = Label(self.LoginFrame,text="PATIENT
ID",font="Helvetica 14 bold",bg="cadet blue",bd=22)

        self.l1.grid(row=1,column=0)

        self.dis1= Label(self.LoginFrame,font="Helvetica 14
bold",bd=2,bg="cadet blue",text=i[0])

        self.dis1.grid(row=1,column=1)

        self.l2 = Label(self.LoginFrame,text="PATIENT
NAME",font="Helvetica 14 bold",bg="cadet blue",bd=22)

        self.l2.grid(row=2,column=0)

        self.dis2=Label(self.LoginFrame,font="Helvetica 14
bold",bd=2,bg="cadet blue",text=i[1])

        self.dis2.grid(row=2,column=1)


        self.l3 = Label(self.LoginFrame,text="APPOINTMENT
NO",font="Helvetica 14 bold",bg="cadet blue",bd=22)

        self.l3.grid(row=3,column=0)

```

```
self.dis3 = Label(self.LoginFrame,font="Helvetica 14  
bold",bg="cadet blue",bd=2,text=i[2])
```

```
self.dis3.grid(row=3,column=1)
```

```
self.l4 = Label(self.LoginFrame,text="DOCTOR  
ID",font="Helvetica 14 bold",bg="cadet blue",bd=22)
```

```
self.l4.grid(row=4,column=0)
```

```
self.dis4= Label(self.LoginFrame,font="Helvetica 14  
bold",bg="cadet blue",bd=2,text=i[3])
```

```
self.dis4.grid(row=4,column=1)
```

```
self.l5 = Label(self.LoginFrame,text="APPOINTMENT  
TIME(HH:MM:SS)",font="Helvetica 14 bold",bg="cadet blue",bd=22)
```

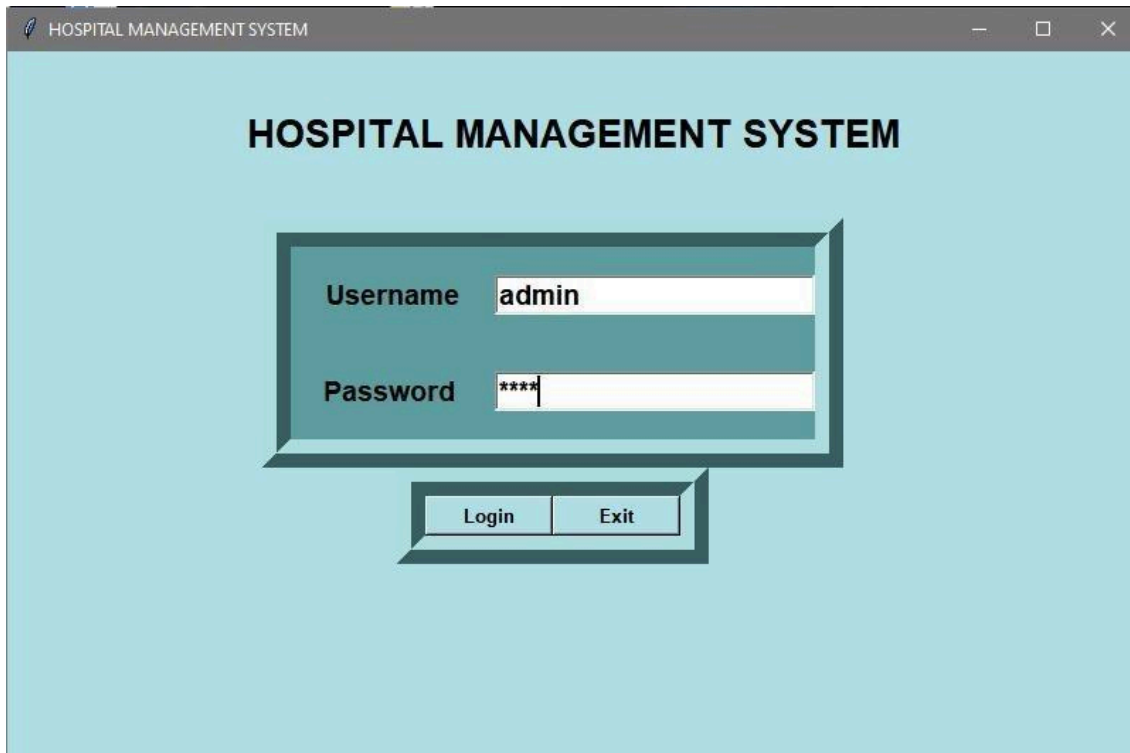
```
self.l5.grid(row=5,column=0)
```

```
self.dis5 = Label(self.LoginFrame,font="Helvetica 14  
bold",bg="cadet blue",bd=2,text=i[5])
```

```
self.dis5.grid(row=5,column=1)
```

OUTPUT

login.py



The screenshot shows a window titled "HOSPITAL MANAGEMENT SYSTEM" with a light blue background. In the center, there is a login form with a 3D effect. The form contains two input fields: "Username" with the text "admin" and "Password" with four asterisks "****". Below the input fields are two buttons, "Login" and "Exit", also with a 3D effect.

menu.py

HOSPITAL MANAGEMENT SYSTEM

PATIENT REGISTRATION FORM

PATIENT ID	111	CONTACT NUMBER	9887454520
PATIENT NAME	John	ALTERNATE CONTACT	9568567560
SEX	Male	EMAIL	johnoliver@gmail.com
DOB (YYYY-MM-DD)	1999-06-23	CONSULTING TEAM / DOCTOR	Philips
BLOOD GROUP	B+ve	ADDRESS	Bangalore

SUBMIT UPDATE DELETE SEARCH EXIT

4.BOOK APPOINTMENT

5.PATIENT BILL

6.EXIT

HOSPITAL DATABASE SYSTEM
DETAILS INSERTED INTO DATABASE
OK

Ln: 95

patient_form.py

SEARCH WINDOW

ENTER PATIENT ID TO SEARCH

111

PATIENT ID	111	ADDRESS	Bangalore
PATIENT NAME	John	CONSULTING TEAM / DOCTOR	Phillips
SEX	Male	EMAIL	johnoliver@gmail.com
DOB (YYYY-MM-DD)	O+ve	CONTACT NUMBER	9887454520
BLOOD GROUP	1999-06-23	ALTERNATE CONTACT	9568567560

SEARCH

MOTIVE

The hospital management system helps register complete patient information. It captures and stores the medical history, treatment required, details of their previous visits, upcoming appointments if any, reports, insurance details and more. It helps eliminate the need to get these details on every visit.

When patients book an appointment, Hospital Management System assigns them to the specialist available, or the one that they prefer. For assignment, it matches the patient's illness to the doctor's area of expertise.

If medical care needs to be provided at the patient's house, the system will check the doctors' availability for the remote visit. Again, the allocation happens based on availability.

Every hospital or medical institution needs to maintain records of bed availability, occupancy status of rooms with specialized care and more.

HMS collects all such information and makes it readily available whenever required. Your receptionist can easily check the information from the system and convey it to those who seek it.

To summarize, the Hospital Management System takes care of all activities and processes involved in managing the hospital. But, you should understand this technology is only for “Managing”, the aspect of patient care is lacking in it.

Patients having episodic treatment have to be nurtured properly. The one in the OutPatient Department has to be supplied with caregiving inputs so that they can be converted into in-patient for more advanced treatment. There are various other instances as well.

BIBLIOGRAPHY

- <https://www.leadsquared.com/what-is-hospital-management-system/#:~:text=The%20hospital%20management%20system%20helps,the%20details%20on%20every%20visit.>
- <https://www.tutorialspoint.com/sqlite/index.htm>
- <https://www.geeksforgeeks.org/python-tkinter-tutorial/>

LIMITATIONS OF PROJECT

1. All the IDs and information must be remembered by the admin user for future uses.
2. Multiple new windows open up instead of all the operations taking up in a single window.
3. Some bugs (in the form of glitches) might be reported when many number of records starts to get piled up.