

AISSCE 2020-21

Project Report of Informatics Practices

HOSPITAL
MANAGEMENT SYSTEM

Submitted to:
Mr. Abhishek Mathur

Submitted by:
Rohan Nag
XII-A

CONTENTS

1. Certificate

2. Acknowledgement

3. Introduction

4. Languages

5. Database Structure

6. Software Code and Actual

7. Future Development

8. Bibliography

CERTIFICATE

This is to certify that the content of this project entitled, “Hospital Management Software” by **Rohan Nag**, student of Class XII-A, is the bona fide work of him submitted for the consideration of Informatics Practices Report for the **Academic Session 2020-21**.

This is also certified that the work of student is as per CBSE syllabus prescribed for the subject under my supervision.

I wish him all the success.

Ms. Karuna Yadav

PRINCIPAL

Mr. Abhishek Mathur

PGT(Informatics Practices)

External's Signature

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude to everyone who was involved in the making and improvement of this project.

I also thank our **Principal Ms. Karuna Yadav** for providing a conducive environment to us for completing this project.

I also thank my Informatics Practices mentor **Mr. Abhishek Mathur** for pushing us to explore this new python module and helping me when I encountered any error and in debugging it. We thank him for providing solutions to our queries which concluded in this software's successful development.

Lastly, we also thank our parents for pointing out various deficiencies of the software and making it more practical.

Rohan Nag

Class XII-A

INTRODUCTION

This software created in Python 3.7.4 deals with the general management of Hospitals. It basically encompasses the following domains:

- **PATIENT:** This software provides the ability to register, update and search the patient records linking each with a unique ID. It also allows to maintain surgery procedures and tests records of the patient
- **APPOINTMENTS:** The software deals with O.P.D.(Out Patient Department) separately by allowing booking of appointments.
- **EMPLOYEES:** It can also save the data of employees working in the hospital.
- **BILL:** When discharging the patient, it automatically generates the billing amount.
- **PATIENT GRAPHS:** This section encompasses the age graph of patients which can be used for research purposes if required.

This is an entire package capable of managing entire functioning of a small-scale hospital.

LANGUAGES

This software basically uses two languages:

- **Python:** The software has been developed in Python 3.4.7 using the Python IDLE. Python is an object oriented programming language developed in the 1991. It is a high level language which makes it very easy to use and programmer friendly. It uses an interpreter and has gained popularity due to various factors. It is cross-platform and thus can run on various operating systems. Some of its salient features are:

- ❧ Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- ❧ Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- ❧ Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

This project uses different libraries of python such as tkinter, pymysql, matplotlib and pil. It also uses some python modules like datetime and ttk.

- **MySQL:** This is Structured Query Language which is used to make relational database management system. MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language. The software has been developed using MySQL 5.5 to manage the table structures. It is a 4th generation non procedural language developed in IBM Laboratory in 1970s. Some of its salient features are:
 - ❧ It stores information in the form of tables. A single database can contain thousands of tables which can further contain thousands of records.
 - ❧ SQL is extremely manageable and simple to learn the language. SQL is flexible as it works with information base frameworks from **Oracle, IBM, Microsoft**, and so forth.

DATABASE STRUCTURE

Database Name: vtreat

Tables_in_vtreat
appointments
bill
employee
patient
sur_test

Tables:

- Patient

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(25)	YES		NULL	
gender	char(1)	YES		NULL	
age	int(3)	YES		NULL	
date	datetime	YES		NULL	
contact	bigint(20)	YES		NULL	
address	varchar(50)	YES		NULL	
email	varchar(30)	YES		NULL	
doctor	varchar(30)	YES		NULL	
department	varchar(20)	YES		NULL	
category	varchar(20)	YES		NULL	
room_type	varchar(25)	YES		NULL	
room_no	int(5)	YES		NULL	
room_price	int(10)	YES		NULL	

- Sur_test

Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
name	varchar(25)	YES		NULL	
date_of_admission	date	YES		NULL	
date_of_discharge	date	YES		NULL	
ins_category	varchar(30)	YES		NULL	
total	float	YES		NULL	

- Employee

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(25)	YES		NULL	
gender	char(1)	YES		NULL	
age	int(3)	YES		NULL	
date_of_joining	date	YES		NULL	
contact	bigint(20)	YES		NULL	
address	varchar(50)	YES		NULL	
email	varchar(30)	YES		NULL	
salary	int(11)	YES		NULL	
department	varchar(20)	YES		NULL	
experience	int(5)	YES		NULL	

- Appointments

Field	Type	Null	Key	Default	Extra
app_id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(25)	YES		NULL	
gender	char(1)	YES		NULL	
age	int(3)	YES		NULL	
contact	bigint(20)	YES		NULL	
queue_no	int(4)	YES		NULL	
app_date	date	YES		NULL	
app_time	time	YES		NULL	
doctor	varchar(25)	YES		NULL	
department	varchar(20)	YES		NULL	
c_fee	int(5)	YES		NULL	

- Bill

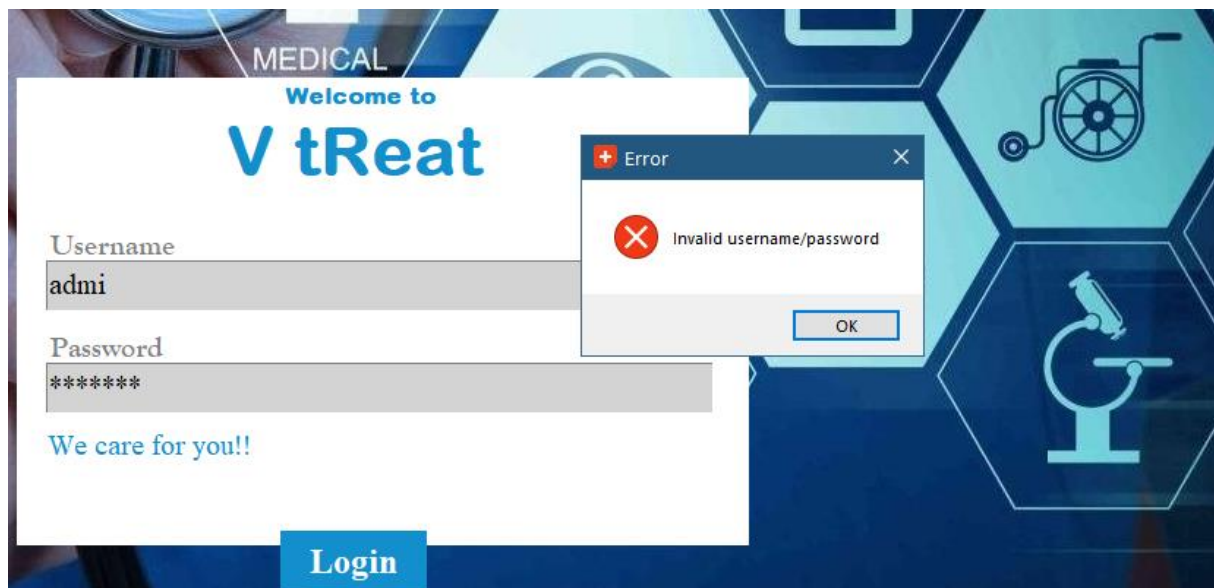
Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
name	varchar(25)	YES		NULL	
date_of_admission	date	YES		NULL	
date_of_discharge	date	YES		NULL	
ins_category	varchar(30)	YES		NULL	
total	float	YES		NULL	

SOFTWARE

AND

CODING

LOGIN PAGE



LOGIN PAGE_CODE

```
from tkinter import*
from h2_main_menu import menu
from PIL import ImageTk
from tkinter import messagebox

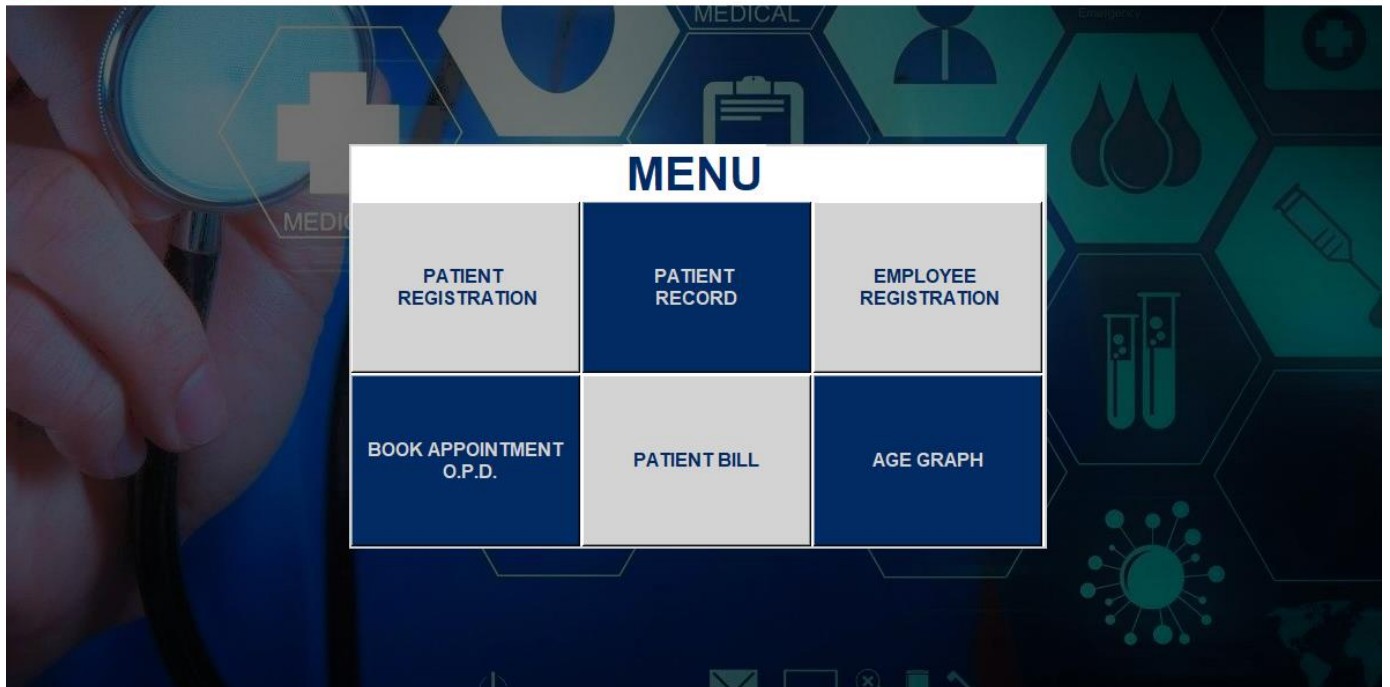
class Login:
    def __init__(self,root):
        self.root=root
        #BGImage
        self.bg=ImageTk.PhotoImage(file="hospital.jpg")
        self.bg_image=Label(root, image=self.bg).place(x=0,y=0,relwidth=1,relheight=1)
        #Frame
        frame=Frame(self.root, bg='white').place(x=80, y=200, height=320, width=500)
        title=Label(frame, text="Welcome to", font=("Arial Rounded MT Bold",12,"bold"), fg="#128fcc",
bg="white").place(x=260,y=200)
        title=Label(frame, text="V tReat", font=("Arial Rounded MT Bold",35,"bold"), fg="#128fcc",
bg="white").place(x=220,y=220)
        lbl_user=Label(frame, text="Username", font=("Goudy old style",15,"bold"), fg="gray",
bg="white").place(x=100,y=300)
        self.txt_user=Entry(frame, font=("Times new roman",15),bg="lightgray", text="Here")
        self.txt_user.place(x=100,y=325, relwidth=0.38,height=35)
        lbl_pass=Label(frame, text="Password", font=("Goudy old style",15,"bold"), fg="gray",
bg="white").place(x=100,y=370)
        self.txt_pass=Entry(frame, font=("Times new roman",15),bg="lightgray",show="*")
        self.txt_pass.place(x=100,y=395,relwidth=0.38,height=35)

        lbl=Label(frame, text="We care for you!!", bg="white", bd=0, font=("Times new roman", 15),
fg="#128fcc").place(x=100,y=440)
        login_bt=Button(frame, text="Login", bg="#128fcc", bd=0, font=("Times new roman", 18,
"bold"),command=self.loginfunc, fg="white").place(x=260,y=510, width=100)
        def loginfunc(self):
            if self.txt_pass.get()==" " or self.txt_user.get()==" ":
                messagebox.showerror("Error","All fields are required", parent=self.root)
            elif self.txt_pass.get()!="admin" or self.txt_user.get()!="admin":
                messagebox.showerror("Error","Invalid username/password", parent=self.root)
            else:
                root.destroy()
                menu()
root=Tk()
root.iconbitmap('logo.ico')
root.title("V tReat_LOGIN")
root.geometry("1199x600+100+50")
Call=Login(root)
root.resizable(False,False)
root.mainloop()
```

MAIN MENU

 MAIN MENU



MAIN MENU_CODE

```
from tkinter import*
from PIL import ImageTk
import tkinter.messagebox
import matplotlib.pyplot as plt
import pymysql
from h3_pat_form import PAT
from h4_pat_record import SUR_TEST
from h5_employee_reg import EMP
from h6_app_form import APP
from h7_bill import BILL

#EXIT for MENU
def gp():
    global root2
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    c.execute("select age from PATIENT;")
    ages=c.fetchall()
    plt.hist(ages,bins=9)
    plt.title("Patient Age Histogram")
    plt.ylabel("Number of Patients in Hospital")
    plt.xlabel("Ages")
    plt.show()
    #root2.destroy()

#MENU BUTTONS
def menu():
    global root2,button1,button2,button3,button4,button5,m,button6
    class Login:
        def __init__(self,root2):
            self.root2=root2
            #BGimage
            self.bg=ImageTk.PhotoImage(file="hospital1.jpg")
            self.bg_image=Label(root2, image=self.bg).place(x=0,y=0,relwidth=1,relheight=1)
            frame=Frame(self.root2, bg='white',highlightbackground="light gray",
            highlightthickness=2).place(x=298, y=120, height=350, width=604)
            root2=tkinter.Tk()
            Call=Login(root2)
            root2.title("MAIN MENU")
            m=tkinter.Label(root2,text="MENU",font='Goudyoldstyle 30 bold ',fg='#022B63',bg="white")
            button1=tkinter.Button(root2,text="PATIENT \n REGISTRATION",command=PAT,bg='light
gray',fg='#022B63',font='Arial 12 bold',) #bd=0
```

```
button2 = tkinter.Button(root2, text="PATIENT \nRECORD",bg='#022B63',fg='light
gray',command=SUR_TEST,font='Arial 12 bold',)#bd=0)
button3 = tkinter.Button(root2, text="EMPLOYEE \n REGISTRATION",bg='light
gray',fg='#022B63',command=EMP,font='Arial 12 bold',)#bd=0)
button4 = tkinter.Button(root2, text="BOOK APPOINTMENT\n O.P.D.",bg='#022B63',fg='light
gray',command=APP,font='Arial 12 bold',)#bd=0)
button5 = tkinter.Button(root2, text="PATIENT BILL",bg='light
gray',fg='#022B63',command=BILL,font='Arial 12 bold',)#bd=0)
button6 = tkinter.Button(root2, text="AGE GRAPH",command=gp,bg='#022B63',fg='light
gray',font='Arial 12 bold',)#bd=0)
m.place(x=535,y=120)

button1.place(x=300,y=170, width=198, height=148)

button2.place(x=500,y=170, width=198, height=148)

button3.place(x=700,y=170, width=198, height=148)

button4.place(x=300, y=320, width=198, height=148)

button5.place(x=500,y=320, width=198, height=148)

button6.place(x=700,y=320, width=198, height=148)

root2.iconbitmap('logo.ico')
root2.geometry("1199x600+100+50")
root2.resizable(False,False)

root2.mainloop()
```

PATIENT REGISTRATION FORM

1 REGISTRATION FORM

PATIENT ID: 7

PATIENT NAME: Manan Mehra

GENDER: M

AGE: 44

DATE/TIME: Add current date/time

CONTACT NUMBER: 3216549877

ADDRESS: 55, Sector-1, Mansarovar, Jaipur

EMAIL: mm@gmail.com

CONSULTING TEAM / DOCTOR: Sunil Gupta

DEPARTMENT: General Physician
Chest Physician
Paediatrician
General Surgeon
Cardiologist

PATIENT CATEGORY: General
Critical
Emergency

ROOM TYPE: Triple Room

ROOM NUMBER: 232

ROOM CHARGES: 3000

SUBMIT BACK

Patient ID	Name	Gender	
2	Ram Kapoor	M	55
3	Damini	F	45
4	Mihan	M	15
5	Abhiroop Mathur	M	85
6	Rupa MMehta	F	47

2

V tReat DATABASE SYSTEM

PATIENT REGISTERED IN DATABASE

OK

3

V tReat_pat_for

File Edit Help

New

Exit

4

V tReat_pat_form

Edit Help

Update

Search

Delete

5

SEA...

Search Using:

Name

ID

SEARCH

6

UPDATE WINDOW

UPDATE FORM

ID OF PATIENT TO UPDATE: 3 Search

PATIENT NAME: Damini

GENDER: F

AGE: 45

DATE/TIME: 2020-11-30 19:16:29 Add current date/time

CONTACT NUMBER: 654654

ADDRESS: 654

EMAIL: mn,m

CONSULTING TEAM / DOCTOR: Sunil Gupta

DEPARTMENT: General Physician
Chest Physician
Paediatrician
General Surgeon
Cardiologist

PATIENT CATEGORY: General
Critical
Emergency

ROOM TYPE: Deluxe Single Room

ROOM NUMBER:

ROOM CHARGES:

UPDATE

7

SEARCH WINDOW

PATIENT DETAILS

PATIENT ID: 3

PATIENT NAME: Damini

GENDER: F

PATIENT AGE: 45

DATE OF REGISTRATION: 2020-11-30 19:16:29

PATIENT CONTACT_NO: 654654

PATIENT ADDRESS: 654

PATIENT EMAIL-ID: mn,m

CONSULTING TEAM/DOCTOR: Sunil Gupta

DEPARTMENT: Paediatrician

CATEGORY: Emergency

ROOM TYPE: Deluxe Single Room

ROOM NO: 101

ROOM CHARGES: 8000

PATIENT REGISTRATION FORM_CODE

```
from tkinter import*
import tkinter
from PIL import ImageTk
import pymysql
from tkinter import ttk
import datetime

def PAT():
    global pat_address, pat_contact, pat_age, pat_CT, pat_date, pat_email, pat_ID, pat_name,
    pat_Gender, pat_cat, pat_dept, room_var, room_no, rate, d_sel
    global
    rootp,regform,id,name,Gender,email,ct,addr,c1,SUBMIT,menuubar,filemenu,back,SEARCH,DELETE,UPDA
    TE, r4, L1

    class Table:
        def __init__(self,rootp):
            self.rootp=rootp

            frame1=Frame(self.rootp, bg='#022B63', borderwidth=1)
            frame1.place(relx=0.5, relheight=1, relwidth=0.5)
            scroll_x=Scrollbar(frame1, orient=HORIZONTAL)
            scroll_y=Scrollbar(frame1, orient=VERTICAL)
            self.pat_table=ttk.Treeview(frame1, columns=("id", "name", "Gender",
"age", "date/time", "Phno", "address", "email", "doctor", "department", "categ", "roomt", "roomno", "roomc"
), xscrollcommand=scroll_x.set, yscrollcommand=scroll_y.set)
            scroll_x.pack(side=BOTTOM, fill=X)
            scroll_y.pack(side=RIGHT, fill=Y)
            scroll_x.config(command=self.pat_table.xview)
            scroll_y.config(command=self.pat_table.yview)
            self.pat_table.heading("id",text="Patient ID")
            self.pat_table.heading("name",text="Name")
            self.pat_table.heading("Gender",text="Gender")
            self.pat_table.heading("age",text="Age")
            self.pat_table.heading("date/time",text="Date/Time")
            self.pat_table.heading("Phno",text="Contact Number")
            self.pat_table.heading("address",text="Address")
            self.pat_table.heading("email",text="Email-ID")
            self.pat_table.heading("doctor",text="Consulting Team/Doctor")
            self.pat_table.heading("department",text="Department")
            self.pat_table.heading("categ",text="Patient Category")
            self.pat_table.heading("roomt",text="Room Type")
            self.pat_table.heading("roomno",text="Room No.")
            self.pat_table.heading("roomc",text="Room Charges")
            self.pat_table['show']='headings'
            self.pat_table.pack()
```

```

        conn=pymysql.connect(host="localhost", user="root", password="password",
database="VTREAT")
        c=conn.cursor()
        c.execute("select* from PATIENT")
        rows=c.fetchall()
        if len(rows)!=0:
            self.pat_table.delete(*self.pat_table.get_children())
            for row in rows:
                self.pat_table.insert("",END,values=row)
            print("Data inserted in treeview from vtreat database")
            conn.commit()
            conn.close()

        else:
            print("Not done")

rootp=Tk()
call=Table(rootp)
rootp.title("PATIENT FORM")
menubar=tkinter.Menu(rootp)
filemenu=tkinter.Menu(menubar, tearoff=0)
filemenu.add_command(label="New",command=PAT)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=EXO)

emenu=tkinter.Menu(menubar, tearoff=0)
emenu.add_command(label="Update",command=P_UPDATE)
emenu.add_separator()
emenu.add_command(label="Search", command=P_display)
emenu.add_separator()
emenu.add_command(label="Delete", command=D_display)

helpmenu=tkinter.Menu(menubar, tearoff=0)
helpmenu.add_command(label="Help",command=hp)
helpmenu.add_command(label="About",command=ab)

menubar.add_cascade(label="File", menu=filemenu)
menubar.add_cascade(label="Edit", menu=emenu)
menubar.add_cascade(label="Help", menu=helpmenu)
rootp.config(menu=menubar)

frame2=Frame(rootp, bg='#022B63')
frame2.place(x=0, y=0, relheight=1, relwidth=0.5)

regform=tkinter.Label(frame2,text="REGISTRATION FORM",font="Arial 24 bold",fg="light gray",
bg="#022B63")

id=tkinter.Label(frame2,text="PATIENT ID",font=("Arial",12,"bold"), fg="#022B63", bg="white")

```

```

pat_ID=tkinter.Entry(frame2,width=50)
conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
c=conn.cursor()
c.execute("select max(id) from PATIENT;")
i=c.fetchone()
i=i[0]+1
pat_ID.insert(tkinter.END, i)
pat_ID.config(state='disabled')

name=tkinter.Label(frame2,text="PATIENT NAME",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
pat_name = tkinter.Entry(frame2,width=50)

Gender=tkinter.Label(frame2,text="GENDER",font=("Arial",12,"bold"), fg="#022B63", bg="white")
pat_Gender=tkinter.Listbox(frame2,selectmode='SINGLE', exportselection=0, height=2, width=50)
pat_Gender.insert(tkinter.END, "M")
pat_Gender.insert(tkinter.END, "F")

age=tkinter.Label(frame2, text="AGE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
pat_age=tkinter.Entry(frame2,width=50)

def add_date():
    global pat_date,now
    now = datetime.datetime.now()
    pat_date=now.strftime("%Y-%m-%d %H:%M:%S")
    datet=tkinter.Label(frame2, text=pat_date,width=25)
    datet.grid(row = 5, column = 1)

date=tkinter.Label(frame2, text="DATE/TIME",font=("Arial",12,"bold"), fg="#022B63", bg="white")
dt_bt=tkinter.Button(frame2,text="Add current
date/time",command=add_date,font=("Arial",10,"bold"), fg="white", bg="#022B63",bd=0,width=23)


c1=tkinter.Label(frame2, text="CONTACT NUMBER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
pat_contact=tkinter.Entry(frame2,width=50)

addr=tkinter.Label(frame2, text="ADDRESS",font=("Arial",12,"bold"), fg="#022B63", bg="white")
pat_address=tkinter.Entry(frame2,width=50)

email=tkinter.Label(frame2, text="EMAIL",font=("Arial",12,"bold"), fg="#022B63", bg="white")
pat_email = tkinter.Entry(frame2,width=50)

ct=tkinter.Label(frame2,text="CONSULTING TEAM / DOCTOR",font=("Arial",12,"bold"), fg="#022B63",
bg="white")

```

```

conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
c=conn.cursor()
c.execute("select name from employee where department='Doctor';")
rows=c.fetchall()

L=[]
for row in rows:
    L.insert(0, row[0])

d_sel=tkinter.StringVar(rootp)
d_sel.set(L[0])
pat_CT=tkinter.OptionMenu(frame2, d_sel, *L) #app_dept is a tkinter variable and * is used in python
when unpacking is required

dept=tkinter.Label(frame2,text="DEPARTMENT",font=("Arial",12,"bold"), fg="#022B63", bg="white")
pat_dept = tkinter.Listbox(frame2, selectmode='SINGLE', exportselection=0, height=5, width=50)
pat_dept.insert(tkinter.END, "General_Physician")
pat_dept.insert(tkinter.END, "Chest_Physician")
pat_dept.insert(tkinter.END, "Paediatrician")
pat_dept.insert(tkinter.END, "General_Surgeon")
pat_dept.insert(tkinter.END, "Cardiologist")

cat=tkinter.Label(frame2, text="PATIENT CATEGORY",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
pat_cat=tkinter.Listbox(frame2, selectmode='SINGLE', exportselection=0, height=3, width=50)
pat_cat.insert(tkinter.END, "General")
pat_cat.insert(tkinter.END, "Critical")
pat_cat.insert(tkinter.END, "Emergency")

def room_sel(event):
    global r4,L1
    r2=room_var.get()
    if r2=='Deluxe_Single_Room':
        room_no.delete(0,tkinter.END)
        L1=[101,102,201,202,301,302]
        r4=8000
        for j in L1:
            room_no.insert(tkinter.END,j)
        rate.config(text=r4)
    elif r2=='Single_Room':
        room_no.delete(0,tkinter.END)
        L1=[111,112,211,212,311,312]
        r4=6000
        for j in L1:
            room_no.insert(tkinter.END,j)
        rate.config(text=r4)

```

```

elif r2=='Double_Room':
    room_no.delete(0,tkinter.END)
    L1=[121,122,221,222,321,322]
    r4=5000
    for j in L1:
        room_no.insert(tkinter.END,j)
    rate.config(text=r4)
elif r2=='Triple_Room':
    room_no.delete(0,tkinter.END)
    L1=[131,132,231,232,331,332]
    r4=3000
    for j in L1:
        room_no.insert(tkinter.END,j)
    rate.config(text=r4)
elif r2=='General_Ward':
    room_no.delete(0,tkinter.END)
    L1=[141,142,241,242,341,342]
    r4=2000
    for j in L1:
        room_no.insert(tkinter.END,j)
    rate.config(text=r4)
room_tl=tkinter.Label(frame2,text="ROOM TYPE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
L=['Deluxe_Single_Room','Single_Room','Double_Room','Triple_Room','General_Ward']
room_var=tkinter.StringVar(frame2)
room_t= tkinter.OptionMenu(frame2, room_var, *L,command=room_sel)

room_nol=tkinter.Label(frame2,text="ROOM NUMBER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
room_no = tkinter.Listbox(frame2, width=8, height=1, selectmode='SINGLE', exportselection=0)

ratel=tkinter.Label(frame2, text="ROOM CHARGES",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
rate=tkinter.Label(frame2,text="-----")

back=tkinter.Button(frame2,text="BACK",command=rootp.destroy,font=("Arial",12,"bold"),
fg="white", bg="#022B63") #,bd=0)
SUBMIT=tkinter.Button(frame2,text="SUBMIT",command=IN_PAT,font=("Arial",12,"bold"),
fg="white", bg="#022B63") #,bd=0)

regform.grid(row=0, column=0,columnspan=2)
id.grid(row = 1, column = 0, pady=5,sticky=W)
pat_ID.grid(row = 1, column = 1, pady=5)
name.grid(row = 2, column = 0, pady=5,sticky=W)
pat_name.grid(row = 2, column = 1, pady=5)
Gender.grid(row = 3, column = 0, pady=5,sticky=W)

```

```

pat_Gender.grid(row = 3, column = 1, pady=5)
age.grid(row = 4, column = 0, pady=5,sticky=W)
pat_age.grid(row = 4, column = 1, pady=5)
date.grid(row = 5, column = 0, pady=5,sticky=W)
dt_bt.grid(row = 5, column = 1, pady=5)
c1.grid(row = 6, column = 0, pady=5,sticky=W)
pat_contact.grid(row = 6, column = 1, pady=5)
addr.grid(row = 7, column = 0, pady=5,sticky=W)
pat_address.grid(row = 7, column = 1, pady=5)
email.grid(row = 8, column = 0, pady=5,sticky=W)
pat_email.grid(row = 8, column = 1, pady=5)
ct.grid(row = 9, column = 0, pady=5, padx=2, sticky=W)
pat_CT.grid(row = 9, column = 1, pady=5)
dept.grid(row = 10, column = 0, pady=5,sticky=W)
pat_dept.grid(row = 10, column = 1, pady=5)
cat.grid(row = 11, column = 0, pady=5,sticky=W)
pat_cat.grid(row = 11, column = 1, pady=5)
room_tl.grid(row = 12, column = 0, pady=5,sticky=W)
room_t.grid(row = 12, column = 1, pady=5)
room_nol.grid(row = 13, column = 0, pady=5,sticky=W)
room_no.grid(row = 13, column = 1, pady=5)
ratel.grid(row = 14, column = 0, pady=5,sticky=W)
rate.grid(row = 14, column = 1, pady=5)

```

```

SUBMIT.grid(row = 15, column = 0)
back.grid(row = 15, column = 1)

```

```

rootp.iconbitmap('logo.ico')
rootp.title("V tReat_pat_form")
rootp.geometry("1199x600+100+50")

```

```

rootp.mainloop()

```

```

#input patient form

```

```

def IN_PAT():

```

```

    global pp1, pp2, pp3, pp4, pp5, pp6, pp7, pp8, pp9, pp10, pp11, ce1,conn
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    pp1=pat_ID.get()
    pp2=pat_name.get()
    pp3=pat_Gender.get(tkinter.ACTIVE)
    pp4=pat_age.get()
    pp5=pat_date
    pp6=pat_contact.get()
    pp7=pat_address.get()
    pp8=pat_email.get()
    pp9=d_sel.get()

```

```

pp10=pat_dept.get(tkinter.ACTIVE)
pp11=pat_cat.get(tkinter.ACTIVE)
pp12=room_var.get()
pp13=room_no.get(tkinter.ACTIVE)
pp14=r4

```

```

c.execute('INSERT INTO patient(name,gender,age,date,
contact,address,email,doctor,department,category, room_type, room_no, room_price)
VALUES("{}","{}","{}","{}","{}","{}","{}","{}","{}","{}","{}","{}").format(pp2,pp3,pp4,pp5,pp6,pp7,pp8,pp9,pp10,
pp11,pp12,pp13,pp14))
rootp.destroy()
tkinter.messagebox.showinfo("V tReat DATABASE SYSTEM","PATIENT REGISTERED IN DATABASE")
conn.commit()
conn.close()# it flushes the memory

```

```

#exit from patient form
def EXO():
    rootp.destroy()

```

```

#function for patient form help
def hp():
    tkinter.messagebox.showinfo("V tReat DATABASE SYSTEM","For any help regarding the software \n
CONTACT the creators using the details given in the About Section")

```

```

def ab():
    tkinter.messagebox.showinfo("V tReat DATABASE SYSTEM"," The software is designed and created by
\n \n Vihaan S. Kumar Contact-456789xxxx\n Rohan Nag Contact-789659xxxx \n \n Email-
pydeveloper@gmail.com \n \n©2020")

```

```

'''=====PATDELSU_CODE====='''

```

```

#variables
rootU=None
rootD=None
rootS=None
head=None
inp_s=None
searchB=None
#display/search button

```

```

def Search_button():
    global inp_s,entry,errorS,t,i,q,dis1,dis2,dis3,dis4,dis5,dis6,dis7,dis8,dis9,dis10, dis11
    global l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")

```

```

c=conn.cursor()
frameS.place_forget() #forget hides the frame

if searchL.get(tkinter.ACTIVE)=="Name":
    z=str(entry.get())
    c.execute('select * from PATIENT where name="{0}"'.format(z))
    t=c.fetchall()
elif (searchL.get(tkinter.ACTIVE)=="ID"):
    inp_s=int(entry.get())
    c.execute('select * from PATIENT where ID=%d'%inp_s)
    t=c.fetchall()
if (len(t)==0):
    messagebox.showerror("Error","NO PATIENT WITH SUCH ID/NAME;", parent=rootS)
else:
    for i in t:
        pat_det=tkinter.Label(rootS,text="PATIENT DETAILS",font="Arial 24 bold",fg="light gray",
bg="#022B63")
        l1=tkinter.Label(rootS,text="PATIENT ID",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis1=tkinter.Label(rootS,text=i[0])
        l2=tkinter.Label(rootS,text="PATIENT NAME",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis2=tkinter.Label(rootS,text=i[1])
        l3=tkinter.Label(rootS,text="GENDER",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis3=tkinter.Label(rootS,text=i[2])
        l4=tkinter.Label(rootS,text="PATIENT AGE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis4=tkinter.Label(rootS,text=i[3])
        l5=tkinter.Label(rootS,text="DATE OF REGISTRATION",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        dis5=tkinter.Label(rootS,text=i[4])
        l6=tkinter.Label(rootS,text="PATIENT CONTACT_NO",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        dis6=tkinter.Label(rootS,text=i[5])
        l7=tkinter.Label(rootS,text="PATIENT ADDRESS",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        dis7=tkinter.Label(rootS,text=i[6])
        l8=tkinter.Label(rootS,text="PATIENT EMAIL-ID",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        dis8=tkinter.Label(rootS,text=i[7])
        l9=tkinter.Label(rootS,text="CONSULTING TEAM/DOCTOR ",font=("Arial",12,"bold"),
fg="#022B63", bg="white")
        dis9=tkinter.Label(rootS,text=i[8])
        l10=tkinter.Label(rootS,text="DEPARTMENT",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis10=tkinter.Label(rootS,text=i[9])
        l11=tkinter.Label(rootS,text="CATEGORY",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis11=tkinter.Label(rootS,text=i[10])
        l12=tkinter.Label(rootS,text="ROOM TYPE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis12=tkinter.Label(rootS,text=i[11])
        l13=tkinter.Label(rootS,text="ROOM NO",font=("Arial",12,"bold"), fg="#022B63", bg="white")

```



```
dis13=tkinter.Label(rootS,text=i[12])
l14=tkinter.Label(rootS,text="ROOM CHARGES",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
dis14=tkinter.Label(rootS,text=i[13])
```

```
pat_det.grid(row=0, column=0, pady=5, padx=2,columnspan=2)
l1.grid(row=1, column=0, pady=5, padx=2)
dis1.grid(row=1, column=1, pady=5, padx=2)
l2.grid(row=2, column=0, pady=5, padx=2)
dis2.grid(row=2, column=1, pady=5, padx=2)
l3.grid(row=3, column=0, pady=5, padx=2)
dis3.grid(row=3, column=1, pady=5, padx=2)
l4.grid(row=4, column=0, pady=5, padx=2)
dis4.grid(row=4, column=1, pady=5, padx=2)
l5.grid(row=5, column=0, pady=5, padx=2)
dis5.grid(row=5, column=1, pady=5, padx=2)
l6.grid(row=6, column=0, pady=5, padx=2)
dis6.grid(row=6, column=1, pady=5, padx=2)
l7.grid(row=7, column=0, pady=5, padx=2)
dis7.grid(row=7, column=1, pady=5, padx=2)
l8.grid(row=8, column=0, pady=5, padx=2)
dis8.grid(row=8, column=1, pady=5, padx=2)
l9.grid(row=9, column=0, pady=5, padx=2)
dis9.grid(row=9, column=1, pady=5, padx=2)
l10.grid(row=10, column=0, pady=5, padx=2)
dis10.grid(row=10, column=1, pady=5, padx=2)
l11.grid(row=11, column=0, pady=5, padx=2)
dis11.grid(row=11, column=1, pady=5, padx=2)
l12.grid(row=12, column=0, pady=5, padx=2)
dis12.grid(row=12, column=1, pady=5, padx=2)
l13.grid(row=13, column=0, pady=5, padx=2)
dis13.grid(row=13, column=1, pady=5, padx=2)
l14.grid(row=14, column=0, pady=5, padx=2)
dis14.grid(row=14, column=1, pady=5, padx=2)
conn.commit()
```

```
def eXO():
    rootS.destroy()
```

```
##search window
```

```
def P_display():
    global rootS,head,inp_s,entry,searchB, frameS, searchL
    rootS=tkinter.Tk()
    rootS.title("SEARCH WINDOW")
    frameS=tkinter.Frame(rootS)
```

```
    head=tkinter.Label(frameS,text="Search Using:",fg="red")
```

```
searchL=tkinter.Listbox(frameS, selectmode='SINGLE', exportselection=0, height=2, width=20)
searchL.insert(tkinter.END,"Name")
searchL.insert(tkinter.END,"ID")
```

```
entry=tkinter.Entry(frameS)
searchB=tkinter.Button(frameS,text='SEARCH',command=Search_button)
```

```
menubar= tkinter.Menu(rootS)
filemenu = tkinter.Menu(menubar, tearoff=0)
filemenu.add_command(label="NEW", command=P_display)
filemenu.add_separator()
filemenu.add_command(label="EXIT", command=eXO)
menubar.add_cascade(label="File", menu=filemenu)
rootS.config(menu=menubar)
```

```
head.grid(row=0,column=0, columnspan=2)
searchL.grid(row=1,column=0,columnspan=2)
entry.grid(row=2,column=0,columnspan=2)
searchB.grid(row=3,column=0, columnspan=2)
frameS.place(x=0,y=0)
rootS.iconbitmap('logo.ico')
rootS.mainloop()
```

#DELETE BUTTON

```
def Delete_button():
    global inp_d,entry1,errorD,disd1
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    inp_d = int(entry1.get())
    c.execute("select * from PATIENT where ID=%d;"%inp_d)
    p=c.fetchall()
    if (len(p)==0):
        errorD = tkinter.Label(rootD, text="PATIENT RECORD NOT FOUND")
        errorD.pack()
    else:
        c.execute('DELETE FROM PATIENT where ID=%d;'%inp_d)
        tkinter.messagebox.showinfo("VtReat Database System", "The Patient record has been
permanently deleted")
        conn.commit()
        rootD.destroy()
        rootP.destroy()
```

DELETE SCREEN

```
def D_display():
```

```

global rootD,headD,inp_d,entry1,DeleteB
rootD=tkinter.Tk()
rootD.title("DELETE WINDOW")
headD=tkinter.Label(rootD,text="ENTER PATIENT ID TO DELETE",fg="blue")
entry1=tkinter.Entry(rootD)
DeleteB=tkinter.Button(rootD,text="DELETE",command=Delete_button)
headD.pack()
entry1.pack()
DeleteB.pack()
rootD.iconbitmap('logo.ico')
rootD.mainloop()

##-----PATIENT UPDATE SCREEN -----##
def P_UPDATE():
    global patu_ID,patu_name, patu_Gender,
    patu_age,patu_date,patu_contact,patu_address,patu_email,patu_CT, n,patu_dept,patu_cat,
    roomu_var, r4, L1, roomu_no
    global rootU, regform, id, name, dob, Gender, email, ct, addr, c1, c2, bg, SUBMIT, menubar, filemenu,
    p1f, p2f,HEAD
    rootU = tkinter.Tk()
    rootU.title("UPDATE WINDOW")
    menubar = tkinter.Menu(rootU)
    filemenu = tkinter.Menu(menubar, tearoff=0)
    filemenu.add_command(label="Help", command=hp)
    filemenu.add_command(label="About", command=ab)
    filemenu.add_command(label="EXIT", command=EXITT)
    rootU.config(menu=menubar)
    menubar.add_cascade(label="File", menu=filemenu)

    def Pupdate_box():
        global patu_ID,patu_name, patu_Gender,
        patu_age,patu_date,patu_contact,patu_address,patu_email,patu_CT, n,patu_dept,patu_cat,
        roomu_var, roomu_no,r4
        conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
        c=conn.cursor()
        n=int(patu_ID.get())
        c.execute('Select * from PATIENT where id=%d'%n)
        p=c.fetchall()
        if ((len(p))!=0):
            name=tkinter.Label(rootU,text="PATIENT NAME",font=("Arial",12,"bold"), fg="#022B63",
            bg="white")
            patu_name = tkinter.Entry(rootU,width=50)
            Gender=tkinter.Label(rootU,text="GENDER",font=("Arial",12,"bold"), fg="#022B63", bg="white")
            patu_Gender=tkinter.Listbox(rootU,selectmode='SINGLE', exportselection=0, height=1, width=50)
            patu_Gender.insert(tkinter.END, "M")
            patu_Gender.insert(tkinter.END, "F")

            age=tkinter.Label(rootU, text="AGE",font=("Arial",12,"bold"), fg="#022B63", bg="white")

```

```

patu_age=tkinter.Entry(rootU,width=50)

date=tkinter.Label(rootU, text="DATE/TIME",font=("Arial",12,"bold"), fg="#022B63", bg="white")
patu_date=tkinter.Entry(rootU, width=50)

def add_date():
    patu_date.delete(0,'end')
    now = datetime.datetime.now()
    datet=now.strftime("%Y-%m-%d %H:%M:%S")
    patu_date.insert(0,datet)
    dt_bt=tkinter.Button(rootU,text="Add current
date/time",command=add_date,font=("Arial",10,"bold"), fg="white", bg="#022B63",bd=0,width=23)

    c1=tkinter.Label(rootU, text="CONTACT NUMBER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
    patu_contact=tkinter.Entry(rootU,width=50)

    addr=tkinter.Label(rootU, text="ADDRESS",font=("Arial",12,"bold"), fg="#022B63", bg="white")
    patu_address=tkinter.Entry(rootU,width=50)

    email=tkinter.Label(rootU, text="EMAIL",font=("Arial",12,"bold"), fg="#022B63", bg="white")
    patu_email = tkinter.Entry(rootU,width=50)

    ct=tkinter.Label(rootU,text="CONSULTING TEAM / DOCTOR",font=("Arial",12,"bold"),
fg="#022B63", bg="white")
    patu_CT=tkinter.Entry(rootU,width=50)

    dept=tkinter.Label(rootU,text="DEPARTMENT",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
    patu_dept = tkinter.Listbox(rootU, selectmode='SINGLE', exportselection=0, height=5, width=50)
    patu_dept.insert(tkinter.END, "General_Physician")
    patu_dept.insert(tkinter.END, "Chest_Physician")
    patu_dept.insert(tkinter.END, "Paediatrician")
    patu_dept.insert(tkinter.END, "General_Surgeon")
    patu_dept.insert(tkinter.END, "Cardiologist")

    cat=tkinter.Label(rootU, text="PATIENT CATEGORY",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
    patu_cat=tkinter.Listbox(rootU, selectmode='SINGLE', exportselection=0, height=3, width=50)
    patu_cat.insert(tkinter.END, "General")
    patu_cat.insert(tkinter.END, "Critical")
    patu_cat.insert(tkinter.END, "Emergency")

def room_sel(event):
    global r4,L1
    r2=roomu_var.get()
    if r2=='Deluxe_Single_Room':
        roomu_no.delete(0,tkinter.END)

```

```

L1=[101,102,201,202,301,302]
r4=8000
for j in L1:
    roomu_no.insert(tkinter.END,j)
    rateu.config(text=r4)
elif r2=='Single_Room':
    roomu_no.delete(0,tkinter.END)
    L1=[111,112,211,212,311,312]
    r4=6000
    for j in L1:
        roomu_no.insert(tkinter.END,j)
        rateu.config(text=r4)
elif r2=='Double_Room':
    roomu_no.delete(0,tkinter.END)
    L1=[121,122,221,222,321,322]
    r4=5000
    for j in L1:
        roomu_no.insert(tkinter.END,j)
        rateu.config(text=r4)
elif r2=='Triple_Room':
    roomu_no.delete(0,tkinter.END)
    L1=[131,132,231,232,331,332]
    r4=3000
    for j in L1:
        roomu_no.insert(tkinter.END,j)
        rateu.config(text=r4)
elif r2=='General_Ward':
    roomu_no.delete(0,tkinter.END)
    L1=[141,142,241,242,341,342]
    r4=2000
    for j in L1:
        roomu_no.insert(tkinter.END,j)
        rateu.config(text=r4)
roomu_tl=tkinter.Label(rootU,text="ROOM TYPE",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
L=['Deluxe_Single_Room','Single_Room','Double_Room','Triple_Room','General_Ward']
roomu_var=tkinter.StringVar(rootU)
roomu_t= tkinter.OptionMenu(rootU, roomu_var, *L,command=room_sel)

roomu_nol=tkinter.Label(rootU,text="ROOM NUMBER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
roomu_no = tkinter.Listbox(rootU, width=8, height=1, selectmode='SINGLE', exportselection=0)

rateul=tkinter.Label(rootU, text="ROOM CHARGES",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
rateu=tkinter.Label(rootU,text="-----")

```

```

c.execute('Select * from PATIENT where id={};'.format(pat_u_ID.get()))
t=c.fetchall()
for i in t:
    patu_name.insert(0,i[1])
    patu_Gender.insert(0,i[2])
    patu_age.insert(0,i[3])
    patu_date.insert(0,i[4])
    patu_contact.insert(0,i[5])
    patu_address.insert(0,i[6])
    patu_email.insert(0,i[7])
    patu_CT.insert(0,i[8])
    roomu_var.set(i[11])

```

```

UPDATE=tkinter.Button(rootU,text="UPDATE",command=up1,font=("Arial",12,"bold"),
fg="white", bg="#022B63") #,bd=0)

```

```

name.grid(row = 2, column = 0, pady=5,sticky=W)
patu_name.grid(row = 2, column = 1, pady=5)
Gender.grid(row = 3, column = 0, pady=5,sticky=W)
patu_Gender.grid(row = 3, column = 1, pady=5)
age.grid(row = 4, column = 0, pady=5,sticky=W)
patu_age.grid(row = 4, column = 1, pady=5)
date.grid(row = 5, column = 0, pady=5,sticky=W)
dt_bt.grid(row = 5, column = 2, pady=5)
patu_date.grid(row = 5, column = 1, pady=5)
c1.grid(row = 6, column = 0, pady=5,sticky=W)
patu_contact.grid(row = 6, column = 1, pady=5)
addr.grid(row = 7, column = 0, pady=5,sticky=W)
patu_address.grid(row = 7, column = 1, pady=5)
email.grid(row = 8, column = 0, pady=5,sticky=W)
patu_email.grid(row = 8, column = 1, pady=5)
ct.grid(row = 9, column = 0, pady=5, padx=2, sticky=W)
patu_CT.grid(row = 9, column = 1, pady=5)
dept.grid(row = 10, column = 0, pady=5,sticky=W)
patu_dept.grid(row = 10, column = 1, pady=5)
cat.grid(row = 11, column = 0, pady=5,sticky=W)
patu_cat.grid(row = 11, column = 1, pady=5)
roomu_tl.grid(row = 12, column = 0, pady=5,sticky=W)
roomu_t.grid(row = 12, column = 1, pady=5)
roomu_nol.grid(row = 13, column = 0, pady=5,sticky=W)
roomu_no.grid(row = 13, column = 1, pady=5)
rateul.grid(row = 14, column = 0, pady=5,sticky=W)
rateu.grid(row = 14, column = 1, pady=5)

```

```

UPDATE.grid(row = 15, column = 0)

```

else:

```

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

upform=tkinter.Label(rootU,text="UPDATE FORM",font="Arial 24 bold",fg="light gray",
bg="#022B63")
id=tkinter.Label(rootU,text="ID OF PATIENT TO UPDATE",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
patu_ID=tkinter.Entry(rootU,width=50)
id_bt=tkinter.Button(rootU, text="Search",command=Pupdate_box,font=("Arial",10,"bold"),
fg="white", bg="#022B63",bd=0,width=23)

upform.grid(row=0, column=0,columnspan=2)
id.grid(row = 1, column = 0, pady=5,sticky=W)
patu_ID.grid(row = 1, column = 1, pady=5)
id_bt.grid(row = 1, column = 2, pady=5)
rootU.iconbitmap('logo.ico')
rootU.mainloop()

#Update command of button
def up1():
    global u1, u2, u3, u4, u5, u6, u7, u8, u9, u10, u11
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    u1=patu_ID.get()
    u2=patu_name.get()
    u3=patu_Gender.get(tkinter.ACTIVE)
    u4=patu_age.get()
    u5=patu_date.get()
    u6=patu_contact.get()
    u7=patu_address.get()
    u8=patu_email.get()
    u9=patu_CT.get()
    u10=patu_dept.get(tkinter.ACTIVE)
    u11=patu_cat.get(tkinter.ACTIVE)
    u12=roomu_var.get()
    u13=roomu_no.get(tkinter.ACTIVE)
    u14=r4

    c.execute('UPDATE PATIENT SET name="{0}",Gender="{0}",age={0},date="{0}",
contact={0},address="{0}",email="{0}",doctor="{0}",department="{0}",category="{0}",room_type="{0}",room_no
={0}, room_price={0} where id={0};'.format( u2, u3, u4, u5, u6,u7,u8,u9,u10,u11,u12,u13,u14,u1))

    tkinter.messagebox.showinfo("V tReat DATABSE SYSTEM", "DETAILS UPDATED INTO DATABASE")
    conn.commit()
    rootU.destroy()

def EXITT():
    rootU.destroy()

```

PATIENT RECORD

PATIENT RECORD

PATIENT ID: 4 PATIENT NAME: Mihan GENDER: M AGE: 15

Confirm

Add Surgery **Add Test**

SURGERY **LAB TESTS**

Angioplasty 2020-12-02
Insulin 2020-12-02

VtReat Database System

Test added to Database

OK

PATIENT RECORD

PATIENT ID: 4 PATIENT NAME: Mihan GENDER: M AGE: 15

Confirm

Add Surgery **Add Test**

SURGERY **LAB TESTS**

Angioplasty 2020-12-02
Insulin 2020-12-02
N/A 2020-12-03

LAB TESTS

Biochemistry_Test
Haematology_Test
X-Ray
CT_Scan
Thyroid_Test
Insulin
COVID_RTPCR
COVID_Antibody
Antigen_Test
Sonography
ECG

of Surgery 2020-12-03

INSERT **INSERT**

PATIENT RECORD

File Help

PA Help IT

PATIENT ID

V tReat DATABASE SYSTEM

The software is designed and created by
Vihaan S. Kumar Contact-456789xxxx
Rohan Nag Contact-789659xxxx

Email- pydeveloper@gmail.com

©2020

OK

V tReat DATABASE SYSTEM

For any help regarding the software
CONTACT the creators using the details given in the About
Section

OK

PATIENT RECORD_CODE

```
import pymysql
import tkinter
from tkinter import*
import tkinter.messagebox
from datetime import date
from h3_pat_form import hp,ab
```

```
P_id=None
rootR=None
```

```
#Surgery Insert Button
```

```
def sinser_t_button():
    global P_id,r1,r2,room_t,da,dd,rate,room_no,r3,r4,r5,r6,conn,cst
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    if surg_var.get()=="Knee_Replacement":
        cst=230000
    elif surg_var.get()=="Kidney_Transplant":
        cst=500000
    elif surg_var.get()=="Tonsil_Surgery":
        cst=100000
    elif surg_var.get()=="Angioplasty":
        cst=120000
    elif surg_var.get()=="Coronary_Artery_Bypass":
        cst=150000
    elif surg_var.get()=="Laparoscopic_Cholecystectomy":
        cst=50000
    r1=P_id.get()
    r2=surg_var.get()
    r3=sdat.get()
    r4=cst
    c.execute('INSERT INTO sur_test VALUES({},{},"{}",{});'.format(r1,r2,r3,r4))
    tkinter.messagebox.showinfo("VtReat Database System", "Surgery added to Database")
    conn.commit()
    conn.close()
    ins.grid_forget()
    sdat.config(state='disabled')
    surg.config(state='disabled')
```

```
#Test Insert Button
```

```

def tinsert_button():
    global cst
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    if test_var.get()=="Biochemistry_Test":
        cst=800
    elif test_var.get()=="Haematology_Test":
        cst=1000
    elif test_var.get()=="X-Ray":
        cst=500
    elif test_var.get()=="CT_Scan":
        cst=1700
    elif test_var.get()=="Thyroid_Test":
        cst=400
    elif test_var.get()=="Insulin":
        cst=200
    elif test_var.get()=="COVID_RTPCR":
        cst=1200
    elif test_var.get()=="COVID_Antibody":
        cst=1000
    elif test_var.get()=="Antigen_Test":
        cst=1200
    elif test_var.get()=="Sonography":
        cst=1700
    elif test_var.get()=="ECG":
        cst=500
    r1=P_id.get()
    r2=test_var.get()
    r3=tdat.get()
    r4=cst
    c.execute('INSERT INTO sur_test VALUES({}, "{}", "{}", {});'.format(r1,r2,r3,r4))
    tkinter.messagebox.showinfo("VtReat Database System", "Test added to Database")
    conn.commit()
    conn.close()
    ins.grid_forget()
    tdat.config(state='disabled')
    test.config(state='disabled')

```

```

def exitt():
    rootR.destroy()

```

#Surgery/Test Window

```

def SUR_TEST():
    global rootR,r_head,P_id,id,da_l,da ,dd_l,dd,Submit,Update,cr,a,surg_var, sdat,a,test_var,tdat,a,cst
    rootR=tkinter.Tk()
    rootR.title("PATIENT RECORD")
    rootR.geometry("800x800")
    rootR.iconbitmap('logo.ico')

```

```

menubar=tkinter.Menu(rootR)
filemenu=tkinter.Menu(menubar, tearoff=0)
filemenu.add_command(label="New",command=SUR_TEST)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=exitt)

helpmenu=tkinter.Menu(menubar, tearoff=0)
helpmenu.add_command(label="Help",command=hp)
helpmenu.add_command(label="About",command=ab)

menubar.add_cascade(label="File", menu=filemenu)
menubar.add_cascade(label="Help", menu=helpmenu)
rootR.config(menu=menubar)

r_head=tkinter.Label(rootR,text="PATIENT RECORD",font="Arial 24 bold",fg="light gray",
bg="#022B63")

id=tkinter.Label(rootR,text="PATIENT ID",font=("Arial",12,"bold"), fg="#022B63", bg="white")
P_id=tkinter.Entry(rootR)

surg_list=["Knee_Replacement","Kidney_Transplant","Tonsil_Surgery","Angioplasty","Coronary_Artery_
Bypass","Laparoscopic_Cholecystectomy"]
test_list=["Biochemistry_Test","Haematology_Test","X-Ray", "CT_Scan",
"Thyroid_Test","Insulin","COVID_RTPCR","COVID_Antibody","Antigen_Test","Sonography","ECG"]

a=7
def new_surg():
    global surg_var, sdat,a,cst,surg_list, ins, surg
    surgl=tkinter.Label(rootR, text="SURGERY",font=("Arial",12,"bold"), fg="#022B63", bg="white")
    surg_var=tkinter.StringVar(rootR)

surg_list=["Knee_Replacement","Kidney_Transplant","Tonsil_Surgery","Angioplasty","Coronary_Artery_
Bypass","Laparoscopic_Cholecystectomy"]
surg=tkinter.OptionMenu(rootR, surg_var, *surg_list)
surg_var.set("N/A")

s_date=tkinter.Label(rootR, text="Date of Surgery",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
d=date.today()
sdat=tkinter.Entry(rootR)
sdat.insert(tkinter.END, d)
ins=tkinter.Button(rootR, text="INSERT", command=sinsert_button,font=("Arial",12,"bold"),
fg="white", bg="#022B63")

surgl.grid(row=a , column=1, stick=W, padx=5)
surg.grid(row=a , column=2)
s_date.grid(row=a , column=3)

```

```

sdat.grid(row=a ,column=4)
ins.grid(row=a , column=5)
a=a+1

def new_test():
    global test_var,tdat,a,cst,test_list, ins,test
    testl=tkinter.Label(rootR, text="LAB TESTS",font=("Arial",12,"bold"), fg="#022B63", bg="white")
    test_var=tkinter.StringVar(rootR)
    test_var.set("N/A")
    test_list=["Biochemistry_Test","Haematology_Test","X-Ray", "CT_Scan",
"Thyroid_Test","Insulin","COVID_RTPCR","COVID_Antibody","Antigen_Test","Sonography","ECG"]
    test=tkinter.OptionMenu(rootR, test_var, *test_list)
    t_date=tkinter.Label(rootR, text="Date of Test",font=("Arial",12,"bold"), fg="#022B63", bg="white")
    d=date.today()
    tdat=tkinter.Entry(rootR)
    tdat.insert(tkinter.END, d)
    ins=tkinter.Button(rootR, text="INSERT", command=tinsert_button,font=("Arial",12,"bold"),
fg="white", bg="#022B63")
    testl.grid(row=a , column=1,padx=5)
    test.grid(row=a , column=2)
    t_date.grid(row=a , column=3)
    tdat.grid(row=a ,column=4)
    ins.grid(row=a , column=5)
    a=a+1

def extract():
    global a
    P_id.config(state='disabled')
    s_add=tkinter.Button(rootR, text="Add Surgery", command=new_surg,font=("Arial",12,"bold"),
fg="white", bg="#022B63")
    t_add=tkinter.Button(rootR, text="Add Test", command=new_test,font=("Arial",12,"bold"),
fg="white", bg="#022B63")
    s_add.grid(row=6,column=2,pady=5)
    t_add.grid(row=6,column=4,pady=5)

    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    c.execute("select name,gender,age,`test/surgery`,`date_sur/test` from PATIENT left join sur_test on
patient.id=sur_test.id where patient.id=%d;"%int(P_id.get()))
    rows=c.fetchall()

    for row in rows:
        if row[3] in surg_list:
            surgl=tkinter.Label(rootR, text="SURGERY",font=("Arial",12,"bold"), fg="#022B63", bg="white")
            surg=tkinter.Label(rootR, text=row[3])

```

```

        s_date=tkinter.Label(rootR, text="Date of Surgery",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        date=tkinter.Label(rootR, text=row[4])
        surgl.grid(row=a , column=1, stick=W, padx=5)
        surg.grid(row=a , column=2)
        s_date.grid(row=a , column=3)
        date.grid(row=a ,column=4)
        a=a+1

    elif row[3] in test_list:
        testl=tkinter.Label(rootR, text="LAB TESTS",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        test=tkinter.Label(rootR, text=row[3])
        t_date=tkinter.Label(rootR, text="Date of Test",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        date=tkinter.Label(rootR, text=row[4])
        testl.grid(row=a , column=1, stick=W, padx=5)
        test.grid(row=a , column=2)
        t_date.grid(row=a , column=3)
        date.grid(row=a ,column=4)
        a=a+1
    if len(rows)!=0:
        namel=tkinter.Label(rootR,text="PATIENT NAME:",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        name=tkinter.Label(rootR,text=row[0])
        genderl=tkinter.Label(rootR,text="GENDER:",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        gender=tkinter.Label(rootR,text=row[1])
        agel=tkinter.Label(rootR,text="AGE:",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        age=tkinter.Label(rootR,text=row[2])
        namel.grid(row=1, column=2, padx=5)
        name.grid(row=1, column=3, padx=5)
        genderl.grid(row=1, column=4, padx=5)
        gender.grid(row=1, column=5, padx=5)
        agel.grid(row=1, column=6, padx=5)
        age.grid(row=1, column=7, padx=5)
    else:
        tkinter.messagebox.showerror("VtReat Database System", "No patient with such ID")

bt=tkinter.Button(rootR, text="Confirm", command=extract)
empty=tkinter.Label(rootR, text=" ", height=2)

r_head.grid(row=0,column=0, columnspan=8)
id.grid(row=1,column=0,pady=5)
P_id.grid(row=1,column=1,pady=5)
bt.grid(row=2, column=0)
empty.grid(row=5,column=0,columnspan=8)
rootR.mainloop()

```

EMPLOYEE REGISTRATION

V tReat_emp_form

File Edit Help

EMPLOYEE REGISTRATION FORM

EMPLOYEE ID

NAME

Pawan Kumar

GENDER

M

F

AGE

30

DATE OF JOINING

2020-12-03

CONTACT NUMBER

4561237899

ADDRESS

44, Hasanpura, Jaipur

EMAIL

--

SALARY

20000

DEPARTMENT

Doctor

Nurse

Compounder

Housekeeping

Receptionist

EXPERIENCE(in years)

10

SUBMIT

BACK

EMPLOYEE ID	Name	Gender
1	Dr. L.C. Khanna	M
2	Sunil Gupta	M
3	Bharat Kumar	M

UPDATE WINDOW

File

UPDATE FORM

ID OF EMPLOYEE TO UPDATE

2

Search

EMPLOYEE NAME

Sunil Gupta

GENDER

M

AGE

35

DATE OF JOINING

2020-11-28

CONTACT NUMBER

7894561231

ADDRESS

89, Sindh Border

EMAIL

sg@gmail.com

SALARY

90000

DEPARTMENT

Doctor

Nurse

Compounder

Housekeeping

Receptionist

EXPERIENCE(in years)

5

UPDATE

SEARCH WINDOW

File

EMPLOYEE DETAILS

EMPLOYEE ID

2

EMPLOYEE NAME

Sunil Gupta

EMPLOYEE GENDER

M

EMPLOYEE AGE

35

DATE OF JOINING

2020-11-28

EMPLOYEE CONTACT_NO

7894561231

EMPLOYEE ADDRESS

89, Sindh Border

EMPLOYEE EMAIL-ID

sg@gmail.com

SALARY

90000

DEPARTMENT

Doctor

EXPERIENCE

5

EMPLOYEE REGISTRATION_CODE

```
from tkinter import*
import tkinter
from PIL import ImageTk
import pymysql
from tkinter import ttk
from datetime import date

def EMP():
    global emp_address, emp_contact, emp_age, emp_CT, emp_date, emp_email, emp_ID, emp_name,
    emp_gender, emp_exp, emp_dept,d
    global
    rootE,regform,id,name,gender,email,ct,addr,c1,SUBMIT,menubar,filemenu,back,SEARCH,DELETE,UPDAT
    E

    class Table:
        def __init__(self,rootE):
            self.rootE=rootE

            frame1=Frame(self.rootE, bg='#022B63', borderwidth=1)
            frame1.place(relx=0.5, relheight=1, relwidth=0.5)
            scroll_x=Scrollbar(frame1, orient=HORIZONTAL)
            scroll_y=Scrollbar(frame1, orient=VERTICAL)
            self.emp_table=ttk.Treeview(frame1, columns=("emp_id", "name", "gender",
"age", "doj", "Phno", "address", "email", "salary", "department", "exper"), xscrollcommand=scroll_x.set,
yscrollcommand=scroll_y.set)
            scroll_x.pack(side=BOTTOM, fill=X)
            scroll_y.pack(side=RIGHT, fill=Y)
            scroll_x.config(command=self.emp_table.xview)
            scroll_y.config(command=self.emp_table.yview)
            self.emp_table.heading("emp_id",text="EMPLOYEE ID")
            self.emp_table.heading("name",text="Name")
            self.emp_table.heading("gender",text="Gender")
            self.emp_table.heading("age",text="Age")
            self.emp_table.heading("doj",text="Date_of_Joining")
            self.emp_table.heading("Phno",text="Contact Number")
            self.emp_table.heading("address",text="Address")
            self.emp_table.heading("email",text="Email-ID")
            self.emp_table.heading("salary",text="Salary(in Rs.)")
            self.emp_table.heading("department",text="Department")
            self.emp_table.heading("exper",text="Experience")
            self.emp_table['show']='headings'
            self.emp_table.pack()
            conn=pymysql.connect(host="localhost", user="root", password="password",
database="VTREAT")
            c=conn.cursor()
```

```

c.execute("select* from employee")
rows=c.fetchall()
if len(rows)!=0:
    self.emp_table.delete(*self.emp_table.get_children())

    for row in rows:
        self.emp_table.insert("",END,values=row)
    print("Connected to employee table of vtreat database")
    conn.commit()
    conn.close()

else:
    print("Not done")

```

```

rootE=Tk()
call=Table(rootE)
rootE.title("EMPLOYEE REGISTRATION FORM")
menubar=tkinter.Menu(rootE)
filemenu=tkinter.Menu(menubar, tearoff=0)
filemenu.add_command(label="New",command=EMP)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=EXO)

emenu=tkinter.Menu(menubar, tearoff=0)
emenu.add_command(label="Update",command=E_UPDATE)
emenu.add_separator()
emenu.add_command(label="Search", command=E_Display)
emenu.add_separator()
emenu.add_command(label="Delete", command=D_display)

helpmenu=tkinter.Menu(menubar, tearoff=0)
helpmenu.add_command(label="Help",command=hp)
helpmenu.add_command(label="About",command=ab)

menubar.add_cascade(label="File", menu=filemenu)
menubar.add_cascade(label="Edit", menu=emenu)
menubar.add_cascade(label="Help", menu=helpmenu)
rootE.config(menu=menubar)

frame2=Frame(rootE, bg='#022B63')
frame2.place(x=0, y=0, relheight=1, relwidth=0.5)

regform=tkinter.Label(frame2,text="EMPLOYEE REGISTRATION FORM",font="Arial 24 bold",fg="light
gray", bg="#022B63")

id=tkinter.Label(frame2,text="EMPLOYEE ID",font=("Arial",12,"bold"), fg="#022B63", bg="white")

```



```

emp_ID=tkinter.Entry(frame2,width=50)

name=tkinter.Label(frame2,text="NAME",font=("Arial",12,"bold"), fg="#022B63", bg="white")
emp_name = tkinter.Entry(frame2,width=50)

gender=tkinter.Label(frame2,text="GENDER",font=("Arial",12,"bold"), fg="#022B63", bg="white")
emp_gender=tkinter.Listbox(frame2,selectmode='SINGLE', exportselection=0, height=2, width=50)
emp_gender.insert(tkinter.END, "M")
emp_gender.insert(tkinter.END, "F")

age=tkinter.Label(frame2, text="AGE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
emp_age=tkinter.Entry(frame2,width=50)

datel=tkinter.Label(frame2, text="DATE OF JOINING",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
d=date.today()
emp_date=tkinter.Entry(frame2,width=50)
emp_date.insert(0,d)

c1=tkinter.Label(frame2, text="CONTACT NUMBER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
emp_contact=tkinter.Entry(frame2,width=50)

addr=tkinter.Label(frame2, text="ADDRESS",font=("Arial",12,"bold"), fg="#022B63", bg="white")
emp_address=tkinter.Entry(frame2,width=50)

email=tkinter.Label(frame2, text="EMAIL",font=("Arial",12,"bold"), fg="#022B63", bg="white")
emp_email = tkinter.Entry(frame2,width=50)

ct=tkinter.Label(frame2,text="SALARY",font=("Arial",12,"bold"), fg="#022B63", bg="white")
emp_CT=tkinter.Entry(frame2,width=50)

dept=tkinter.Label(frame2,text="DEPARTMENT",font=("Arial",12,"bold"), fg="#022B63", bg="white")
emp_dept = tkinter.Listbox(frame2, selectmode='SINGLE', exportselection=0, height=5, width=50)
emp_dept.insert(tkinter.END, "Doctor")
emp_dept.insert(tkinter.END, "Nurse")
emp_dept.insert(tkinter.END, "Compounder")
emp_dept.insert(tkinter.END, "Housekeeping")
emp_dept.insert(tkinter.END, "Receptionist")

exp=tkinter.Label(frame2,text="EXPERIENCE(in years)",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
emp_exp=tkinter.Entry(frame2, width=50)

back=tkinter.Button(frame2,text="BACK",command=rootE.destroy,font=("Arial",12,"bold"),
fg="white", bg="#022B63") #,bd=0)

```

```
SUBMIT=tkinter.Button(frame2,text="SUBMIT",command=IN_emp,font=("Arial",12,"bold"),
fg="white", bg="#022B63") #,bd=0)
```

```
regform.grid(row=0, column=0,columnspan=3)
id.grid(row = 1, column = 0, pady=5,sticky=W)
emp_ID.grid(row = 1, column = 1, pady=5)
name.grid(row = 2, column = 0, pady=5,sticky=W)
emp_name.grid(row = 2, column = 1, pady=5)
gender.grid(row = 3, column = 0, pady=5,sticky=W)
emp_gender.grid(row = 3, column = 1, pady=5)
age.grid(row = 4, column = 0, pady=5,sticky=W)
emp_age.grid(row = 4, column = 1, pady=5)
datel.grid(row = 5, column = 0, pady=5,sticky=W)
emp_date.grid(row = 5, column = 1, sticky=W)
c1.grid(row = 6, column = 0, pady=5,sticky=W)
emp_contact.grid(row = 6, column = 1, pady=5)
addr.grid(row = 7, column = 0, pady=5,sticky=W)
emp_address.grid(row = 7, column = 1, pady=5)
email.grid(row = 8, column = 0, pady=5,sticky=W)
emp_email.grid(row = 8, column = 1, pady=5)
ct.grid(row = 9, column = 0, pady=5, sticky=W)
emp_CT.grid(row = 9, column = 1, pady=5)
dept.grid(row = 10, column = 0, pady=5,sticky=W)
emp_dept.grid(row = 10, column = 1, pady=5)
exp.grid(row = 11, column = 0, pady=5,sticky=W)
emp_exp.grid(row = 11, column = 1, pady=5)
```

```
SUBMIT.grid(row = 13, column = 0)
back.grid(row = 13, column = 1)
```

```
rootE.iconbitmap('logo.ico')
rootE.title("V tReat_emp_form")
rootE.geometry("1199x600+100+50")
```

```
rootE.mainloop()
```

```
#variables
```

```
rootE=None
emp_ID=None
emp_name=None
emp_dob=None
emp_address=None
emp_gender=None
emp_BG=None
emp_email=None
```

```

emp_contact=None
emp_contactalt=None
emp_CT=None
p=None

#input EMPLOYEE form
def IN_emp():
    global ee1, ee2, ee3, ee4, ee5, ee6, ee7, ee8, ee9, ee10, ee11, ce1, conn, d
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    ee1=emp_ID.get()
    ee2=emp_name.get()
    ee3=emp_gender.get(tkinter.ACTIVE)
    ee4=emp_age.get()
    ee5=d
    ee6=emp_contact.get()
    ee7=emp_address.get()
    ee8=emp_email.get()
    ee9=emp_CT.get()
    ee10=emp_dept.get(tkinter.ACTIVE)
    ee11=emp_exp.get()

    c.execute('INSERT INTO EMPLOYEE(name,gender,age,date_of_joining,
contact,address,email,salary,department,experience)
VALUES("{}","{}",{},{},{}","{}","{}","{}","{}",{},{})'.format(ee2,ee3,ee4,ee5,ee6,ee7,ee8,ee9,ee10,ee11))
    rootE.destroy()
    tkinter.messagebox.showinfo("V tReat DATABASE SYSTEM","EMPLOYEE REGISTERED IN DATABASE")
    conn.commit()
    conn.close()

#exit from employee form
def EXO():
    rootE.destroy()

#function for employee form help
def hp():
    tkinter.messagebox.showinfo("V tReat DATABASE SYSTEM","For any help regarding the software \n
CONTACT the creators using the details given in the About Section")

def ab():
    tkinter.messagebox.showinfo("V tReat DATABASE SYSTEM"," The software is designed and created by
\n \n Vihaan S. Kumar Contact-456789xxxx\n Rohan Nag Contact-789659xxxx \n \n Email-
pydeveloper@gmail.com \n \n©2020")

```

```
'''=====EMPDELSU_CODE====='''
```

```
#variables
rootU=None
rootD=None
rootS=None
head=None
inp_s=None
searchB=None
#display/search button
```

```
def Search_button():
    global inp_s,entry,errorS,t,i,q,dis1,dis2,dis3,dis4,dis5,dis6,dis7,dis8,dis9,dis10, dis11
    global l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    frameS.place_forget() #forget hides the frame
    if searchL.get(tkinter.ACTIVE)=="ID":
        inp_s=int(entry.get())
        c.execute('select * from employee where id=%d;%in_s)
        t=c.fetchall()
    elif searchL.get(tkinter.ACTIVE)=="Name":
        inp_s=str(entry.get())
        c.execute('select * from employee where name="{}}";'.format(inp_s))
        t=c.fetchall()
    if (len(t)==0):
        messagebox.showerror("Error","NO employee WITH SUCH ID;", parent=rootS)
    else:

        for i in t:
            emp_det=tkinter.Label(rootS,text="EMPLOYEE DETAILS",font="Arial 24 bold",fg="light gray",
bg="#022B63")
            l1=tkinter.Label(rootS,text="EMPLOYEE ID",font=("Arial",12,"bold"), fg="#022B63", bg="white")
            dis1=tkinter.Label(rootS,text=i[0])
            l2=tkinter.Label(rootS,text="EMPLOYEE NAME",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
            dis2=tkinter.Label(rootS,text=i[1])
            l3=tkinter.Label(rootS,text="EMPLOYEE GENDER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
            dis3=tkinter.Label(rootS,text=i[2])
            l4=tkinter.Label(rootS,text="EMPLOYEE AGE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
            dis4=tkinter.Label(rootS,text=i[3])
            l5=tkinter.Label(rootS,text="DATE OF JOINING",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
            dis5=tkinter.Label(rootS,text=i[4])
            l6=tkinter.Label(rootS,text="EMPLOYEE CONTACT_NO",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
```

```

dis6=tkinter.Label(rootS,text=i[5])
l7=tkinter.Label(rootS,text="EMPLOYEE ADDRESS",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
dis7=tkinter.Label(rootS,text=i[6])
l8=tkinter.Label(rootS,text="EMPLOYEE EMAIL-ID",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
dis8=tkinter.Label(rootS,text=i[7])
l9=tkinter.Label(rootS,text="SALARY ",font=("Arial",12,"bold"), fg="#022B63", bg="white")
dis9=tkinter.Label(rootS,text=i[8])
l10=tkinter.Label(rootS,text="DEPARTMENT",font=("Arial",12,"bold"), fg="#022B63", bg="white")
dis10=tkinter.Label(rootS,text=i[9])
l11=tkinter.Label(rootS,text="EXPERIENCE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
dis11=tkinter.Label(rootS,text=i[10])

```

```

emp_det.grid(row=0, column=0, pady=5, padx=2,columnspan=2)

```

```

l1.grid(row=1, column=0, pady=5, padx=2)
dis1.grid(row=1, column=1, pady=5, padx=2)
l2.grid(row=2, column=0, pady=5, padx=2)
dis2.grid(row=2, column=1, pady=5, padx=2)
l3.grid(row=3, column=0, pady=5, padx=2)
dis3.grid(row=3, column=1, pady=5, padx=2)
l4.grid(row=4, column=0, pady=5, padx=2)
dis4.grid(row=4, column=1, pady=5, padx=2)
l5.grid(row=5, column=0, pady=5, padx=2)
dis5.grid(row=5, column=1, pady=5, padx=2)
l6.grid(row=6, column=0, pady=5, padx=2)
dis6.grid(row=6, column=1, pady=5, padx=2)
l7.grid(row=7, column=0, pady=5, padx=2)
dis7.grid(row=7, column=1, pady=5, padx=2)
l8.grid(row=8, column=0, pady=5, padx=2)
dis8.grid(row=8, column=1, pady=5, padx=2)
l9.grid(row=9, column=0, pady=5, padx=2)
dis9.grid(row=9, column=1, pady=5, padx=2)
l10.grid(row=10, column=0, pady=5, padx=2)
dis10.grid(row=10, column=1, pady=5, padx=2)
l11.grid(row=11, column=0, pady=5, padx=2)
dis11.grid(row=11, column=1, pady=5, padx=2)
conn.commit()

```

```

def eXO():
    rootS.destroy()

```

```

##search window

```

```

def E_Display():
    global rootS,head,inp_s,entry,searchB, frameS, searchL
    rootS=tkinter.Tk()
    rootS.title("SEARCH WINDOW")
    frameS=tkinter.Frame(rootS)

```

```
head=tkinter.Label(frameS,text="Search Using:",fg="red")
searchL=tkinter.Listbox(frameS, selectmode='SINGLE', exportselection=0, height=2, width=20)
searchL.insert(tkinter.END,"Name")
searchL.insert(tkinter.END,"ID")
```

```
entry=tkinter.Entry(frameS)
searchB=tkinter.Button(frameS,text='SEARCH',command=Search_button)
```

```
menubar= tkinter.Menu(rootS)
filemenu = tkinter.Menu(menubar, tearoff=0)
filemenu.add_command(label="NEW", command=E_Display)
filemenu.add_separator()
filemenu.add_command(label="EXIT", command=eXO)
menubar.add_cascade(label="File", menu=filemenu)
rootS.config(menu=menubar)
```

```
head.grid(row=0,column=0, columnspan=2)
searchL.grid(row=1,column=0,columnspan=2)
entry.grid(row=2,column=0,columnspan=2)
searchB.grid(row=3,column=0, columnspan=2)
frameS.place(x=0,y=0)
rootS.iconbitmap('logo.ico')
rootS.mainloop()
```

#DELETE BUTTON

```
def Delete_button():
    global inp_d,entry1,errorD,disd1
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    inp_d = int(entry1.get())
    c.execute("select * from EMPLOYEE where id=%d;"%inp_d)
    p=c.fetchall()
    if (len(p)==0):
        errorD = tkinter.Label(rootD, text="EMPLOYEE RECORD NOT FOUND")
        errorD.pack()
    else:
        c.execute('DELETE FROM EMPLOYEE where id=%d;'%inp_d)
        tkinter.messagebox.showinfo("VtReat Database System", "The EMPLOYEE record has been
permanently deleted")
        conn.commit()
        rootD.destroy()
        rootE.destroy()
```

DELETE SCREEN

```
def D_display():
    global rootD,headD,inp_d,entry1,DeleteB
```

```

rootD=tkinter.Tk()
rootD.title("DELETE WINDOW")
headD=tkinter.Label(rootD,text="ENTER EMPLOYEE ID TO DELETE",fg="blue")
entry1=tkinter.Entry(rootD)
DeleteB=tkinter.Button(rootD,text="DELETE",command=Delete_button)
headD.pack()
entry1.pack()
DeleteB.pack()
rootD.iconbitmap('logo.ico')
rootD.mainloop()

```

##-----EMPLOYEE UPDATE SCREEN -----##

```
def E_UPDATE():
```

```

    global empu_ID,empu_name, empu_gender,
    empu_age,empu_date,empu_contact,empu_address,empu_email,empu_CT, n,empu_dept,empu_exp
    global rootU, regform, id, name, dob, gender, email, ct, addr, c1, UPDATE, menubar, filemenu
    rootU = tkinter.Tk()
    rootU.title("UPDATE WINDOW")
    menubar = tkinter.Menu(rootU)
    filemenu = tkinter.Menu(menubar, tearoff=0)
    filemenu.add_command(label="Help", command=hp)
    filemenu.add_command(label="About", command=ab)
    filemenu.add_command(label="EXIT", command=EXITT)
    rootU.config(menu=menubar)
    menubar.add_cascade(label="File", menu=filemenu)

```

```
def Pupdate_box():
```

```

    global empu_ID,empu_name, empu_gender,
    empu_age,empu_date,empu_contact,empu_address,empu_email,empu_CT, n,empu_dept,empu_exp
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    n=int(empu_ID.get())
    c.execute('Select * from employee where id=%d'%n)
    p=c.fetchall()
    if ((len(p))!=0):
        name=tkinter.Label(rootU,text="EMPLOYEE NAME",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        empu_name = tkinter.Entry(rootU,width=50)
        gender=tkinter.Label(rootU,text="GENDER",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        empu_gender=tkinter.Listbox(rootU,selectmode='SINGLE', exportselection=0, height=1,
width=50)
        empu_gender.insert(tkinter.END, "M")
        empu_gender.insert(tkinter.END, "F")

        age=tkinter.Label(rootU, text="AGE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        empu_age=tkinter.Entry(rootU,width=50)

```

```

        date=tkinter.Label(rootU, text="DATE OF JOINING",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        empu_date=tkinter.Entry(rootU,width=50)

        c1=tkinter.Label(rootU, text="CONTACT NUMBER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        empu_contact=tkinter.Entry(rootU,width=50)

        addr=tkinter.Label(rootU, text="ADDRESS",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        empu_address=tkinter.Entry(rootU,width=50)

        email=tkinter.Label(rootU, text="EMAIL",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        empu_email = tkinter.Entry(rootU,width=50)

        ct=tkinter.Label(rootU,text="SALARY",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        empu_CT=tkinter.Entry(rootU,width=50)

        dept=tkinter.Label(rootU,text="DEPARTMENT",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        empu_dept = tkinter.Listbox(rootU, selectmode='SINGLE', exportselection=0, height=5, width=50)
        empu_dept.insert(tkinter.END, "Doctor")
        empu_dept.insert(tkinter.END, "Nurse")
        empu_dept.insert(tkinter.END, "Compounder")
        empu_dept.insert(tkinter.END, "Housekeeping")
        empu_dept.insert(tkinter.END, "Receptionist")

        exp=tkinter.Label(rootU,text="EXPERIENCE(in years)",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        empu_exp=tkinter.Entry(rootU, width=50)

        c.execute('Select * from EMPLOYEE where id={}'.format(empu_ID.get()))
        t=c.fetchall()
        for i in t:
            empu_name.insert(0,i[1])
            empu_gender.insert(0,i[2])
            empu_age.insert(0,i[3])
            empu_date.insert(0,i[4])
            empu_contact.insert(0,i[5])
            empu_address.insert(0,i[6])
            empu_email.insert(0,i[7])
            empu_CT.insert(0,i[8])
            empu_exp.insert(0,i[10])

        UPDATE=tkinter.Button(rootU,text="UPDATE",command=up1,font=("Arial",12,"bold"),
fg="white", bg="#022B63") #,bd=0)

        name.grid(row = 2, column = 0, pady=5,sticky=W)

```



```

empu_name.grid(row = 2, column = 1, pady=5)
gender.grid(row = 3, column = 0, pady=5,sticky=W)
empu_gender.grid(row = 3, column = 1, pady=5)
age.grid(row = 4, column = 0, pady=5,sticky=W)
empu_age.grid(row = 4, column = 1, pady=5)
date.grid(row = 5, column = 0, pady=5,sticky=W)

empu_date.grid(row = 5, column = 1, pady=5)
c1.grid(row = 6, column = 0, pady=5,sticky=W)
empu_contact.grid(row = 6, column = 1, pady=5)
addr.grid(row = 7, column = 0, pady=5,sticky=W)
empu_address.grid(row = 7, column = 1, pady=5)
email.grid(row = 8, column = 0, pady=5,sticky=W)
empu_email.grid(row = 8, column = 1, pady=5)
ct.grid(row = 9, column = 0, pady=5, padx=2, sticky=W)
empu_CT.grid(row = 9, column = 1, pady=5)
dept.grid(row = 10, column = 0, pady=5,sticky=W)
empu_dept.grid(row = 10, column = 1, pady=5)
exp.grid(row = 11, column = 0, pady=5,sticky=W)
empu_exp.grid(row = 11, column = 1, pady=5)
UPDATE.grid(row = 13, column = 0, colspan=2)
else:
    tkinter.messagebox.showerror("V tReat DATABSE SYSTEM", "EMPLOYEE IS NOT REGISTERED")

upform=tkinter.Label(rootU,text="UPDATE FORM",font="Arial 24 bold",fg="light gray",
bg="#022B63")
id=tkinter.Label(rootU,text="ID OF EMPLOYEE TO UPDATE",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
empu_ID=tkinter.Entry(rootU,width=50)
id_bt=tkinter.Button(rootU, text="Search",command=Pupdate_box,font=("Arial",10,"bold"),
fg="white", bg="#022B63",bd=0,width=23)

upform.grid(row=0, column=0,colspan=3)
id.grid(row = 1, column = 0, pady=5,sticky=W)
empu_ID.grid(row = 1, column = 1, pady=5)
id_bt.grid(row = 1, column = 2, pady=5)
rootU.iconbitmap('logo.ico')
rootU.mainloop()

#Update command of button
def up1():
    global u1, u2, u3, u4, u5, u6, u7, u8, u9, u10, u11
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    u1=empu_ID.get()
    u2=empu_name.get()
    u3=empu_gender.get(tkinter.ACTIVE)
    u4=empu_age.get()

```

```
u5=empu_date.get()
u6=empu_contact.get()
u7=empu_address.get()
u8=empu_email.get()
u9=empu_CT.get()
u10=empu_dept.get(tkinter.ACTIVE)
u11=empu_exp.get()
```

```
c.execute('UPDATE EMPLOYEE SET name="{0}",gender="{0}",age={0},date_of_joining="{0}",
contact={0},address="{0}",email="{0}",salary={0},department="{0}",experience={0} where id={0};'.format( u2, u3,
u4, u5, u6,u7,u8,u9,u10,u11,u1))
```

```
tkinter.messagebox.showinfo("V tReat DATABSE SYSTEM", "DETAILS UPDATED INTO DATABASE")
conn.commit()
rootU.destroy()
```

```
def EXITT():
    rootU.destroy()
```

APPOINTMENT BOOKING FORM

VtReat_app_form

File Edit Help

APPOINTMENTS BOOKING FORM


QUEUE NO.	NAME	GENDER		AGE	CONTACT NUMBER
2	Bhavna Joshi	<input checked="" type="radio"/> Female	<input type="radio"/> Male	42	9638527412
DATE OF APPOINTMENT	TIME OF APPOINTMENT	CONSULTING DOCTOR		DEPARTMENT	CONSULTATION FEE(Rs.)
2020-12-03	22:05	Dr. L.C. Khanna		Cardiologist	300
<input type="button" value="BACK"/>					<input type="button" value="SUBMIT"/>

Patient ID	Name	Gender	Age	Contact Number	Queue No.	Appointment Date
1001	Rohan Nag	M	17	789456312	1	2020-11-28

ENTER PATIENT ID TO DELETE

10003

VtReat Database System

 The appointment id is not valid.

APPOINTMENT BOOKING FORM_CODE

```
from tkinter import*
import tkinter
from PIL import ImageTk
import pymysql
from tkinter import ttk
from datetime import datetime
from datetime import date

def APP():
    global app_name, app_age, app_gender, app_contact, app_qn, app_date, app_time, d_sel, app_dept,
    fee_var
    global
    roota, regform, id, name, gender, email, ct, addr, c1, SUBMIT, menubar, filemenu, back, SEARCH, DELETE, UPDAT
    class Table:
        def __init__(self, roota):
            self.roota = roota

            frame2 = Frame(self.roota, bg='#022B63', borderwidth=1)
            frame2.place(rely=0.5, relheight=0.5, relwidth=1)
            scroll_x = Scrollbar(frame2, orient=HORIZONTAL)
            scroll_y = Scrollbar(frame2, orient=VERTICAL)
            self.app_table = ttk.Treeview(frame2, columns=("id", "name", "gender",
            "age", "Phno", "Q_no", "time", "date", "doctor", "department", "c_fee"), xscrollcommand=scroll_x.set,
            yscrollcommand=scroll_y.set)
            scroll_x.pack(side=BOTTOM, fill=X)
            scroll_y.pack(side=RIGHT, fill=Y)
            scroll_x.config(command=self.app_table.xview)
            scroll_y.config(command=self.app_table.yview)
            self.app_table.heading("id", text="Patient ID")
            self.app_table.heading("name", text="Name")
            self.app_table.heading("gender", text="Gender")
            self.app_table.heading("age", text="Age")
            self.app_table.heading("Phno", text="Contact Number")
            self.app_table.heading("Q_no", text="Queue No.")
            self.app_table.heading("time", text="Appointment Date")
            self.app_table.heading("date", text="Appointment Time")
            self.app_table.heading("doctor", text="Consulting Doctor")
            self.app_table.heading("department", text="Department")
            self.app_table.heading("c_fee", text="Consultation Fee")

            self.app_table['show'] = 'headings'
            self.app_table.pack()
```

```

        conn=pymysql.connect(host="localhost", user="root", password="password",
database="VTREAT")
        c=conn.cursor()
        c.execute("select* from appointments")
        rows=c.fetchall()
        if len(rows)!=0:
            self.app_table.delete(*self.app_table.get_children())
            for row in rows:
                self.app_table.insert("",END,values=row)
            print("Connected with appointments table of vtreat database")
            conn.commit()
            conn.close()

```

```

    else:
        print("Failed to connect with appointments table of vtreat database")

```

```

roota=Tk()
call=Table(roota)
roota.title("APPOINTMENT BOOKING")
menubar=tkinter.Menu(roota)
filemenu=tkinter.Menu(menubar, tearoff=0)
filemenu.add_command(label="New",command=APP)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=EXO)
emenu=tkinter.Menu(menubar, tearoff=0)
emenu.add_command(label="Update",command=A_UPDATE)
emenu.add_separator()
emenu.add_command(label="Search", command=A_Display)
emenu.add_separator()
emenu.add_command(label="Delete", command=D_display)
helpmenu=tkinter.Menu(menubar, tearoff=0)
helpmenu.add_command(label="Help",command=hp)
helpmenu.add_command(label="About",command=ab)

```

```

menubar.add_cascade(label="File", menu=filemenu)
menubar.add_cascade(label="Edit", menu=emenu)
menubar.add_cascade(label="Help", menu=helpmenu)
roota.config(menu=menubar)

```

```

frame1=Frame(roota, bg='#022B63')
frame1.place(x=0, y=0, relheight=0.5, relwidth=1)

```

```

regform=tkinter.Label(frame1,text="APPOINTMENTS BOOKING FORM",font="Arial 24 bold",fg="light
gray", bg="#022B63")

```

```

conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
c=conn.cursor()
c.execute("select max(queue_no) from APPOINTMENTS;")
i=c.fetchone()
i=i[0]+1

qn=tkinter.Label(frame1, text="QUEUE NO.",font=("Arial",12,"bold"), fg="#022B63", bg="white")
app_qn=tkinter.Entry(frame1,width=10)
app_qn.insert(tkinter.END, i)

name=tkinter.Label(frame1,text="NAME",font=("Arial",12,"bold"), fg="#022B63", bg="white")
app_name = tkinter.Entry(frame1,width=25)

gender=tkinter.Label(frame1,text="GENDER",font=("Arial",12,"bold"), fg="#022B63", bg="white")
app_gender=StringVar(roota)
r2 = tkinter.Radiobutton(frame1, text="Female", variable=app_gender, value="F")
r1 = tkinter.Radiobutton(frame1, text="Male", variable=app_gender, value="M")

age=tkinter.Label(frame1, text="AGE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
app_age=tkinter.Entry(frame1,width=25)

c1=tkinter.Label(frame1, text="CONTACT NUMBER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
app_contact=tkinter.Entry(frame1,width=25)

d=date.today()
dat=tkinter.Label(frame1, text="DATE OF APPOINTMENT",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
app_date=tkinter.Entry(frame1,width=25)
app_date.insert(0, d)

time=tkinter.Label(frame1, text="TIME OF APPOINTMENT",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
now = datetime.now()
current_time = now.strftime("%H:%M")
app_time=tkinter.Entry(frame1,width=25)
app_time.insert(0, current_time)

ct=tkinter.Label(frame1,text="CONSULTING DOCTOR",font=("Arial",12,"bold"), fg="#022B63",
bg="white")

conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
c=conn.cursor()
c.execute("select name from employee where department='Doctor';")

```

```

rows=c.fetchall()
L=[]
for row in rows:
    L.insert(0,row[0])

d_sel=tkinter.StringVar(roota)
d_sel.set(L[0])

app_CT=tkinter.OptionMenu(frame1, d_sel, *L) #app_dept is a tkinter variable and * is used in python
when unpacking is required

depts=["General_Physician","Chest_Physician","Paediatrician","General_Surgeon","Cardiologist"]
app_dept=tkinter.StringVar(roota)
app_dept.set(depts[0])
dept=tkinter.Label(frame1,text="DEPARTMENT",font=("Arial",12,"bold"), fg="#022B63", bg="white")
app_d=tkinter.OptionMenu(frame1, app_dept, *depts) #app_dept is variable and * is used in python
when unpacking is required
app_d.config(width=20)

fee=tkinter.Label(frame1,text="CONSULTATION FEE(Rs.)",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
fee_l=[300,500,1000]
fee_var=StringVar(roota)
fee_var.set(fee_l[0])
app_fee=tkinter.OptionMenu(frame1, fee_var, *fee_l)

SUBMIT=tkinter.Button(frame1,text="SUBMIT",command=IN_PAT,font=("Arial",12,"bold"),
fg="white", bg="#022B63") #,bd=0)
back=tkinter.Button(frame1,text="BACK",command=roota.destroy,font=("Arial",12,"bold"),
fg="white", bg="#022B63") #,bd=0)

empty=tkinter.Label(frame1, text=" ", font=("Arial",12,"bold"), bg="#022B63", height=2)

regform.grid(row=0, column=0,columnspan=5)
qn.grid(row = 1, column = 0, pady=5, padx=30,)
app_qn.grid(row = 2, column = 0, pady=5, padx=30,)
name.grid(row = 1, column = 1, pady=5, padx=30,)
app_name.grid(row = 2, column = 1, pady=5, padx=30,)
gender.grid(row = 1, column = 2, pady=5, padx=30,)
r1.grid(row = 2, column = 2, pady=5, padx=30,sticky=E)
r2.grid(row = 2, column = 2, pady=5, padx=30,sticky=W)
age.grid(row = 1, column = 3, pady=5, padx=30,)
app_age.grid(row = 2, column = 3, pady=5, padx=30,)
c1.grid(row = 1, column = 4, pady=5, padx=30,)

```

```

        c.execute('INSERT INTO appointments(name,gender,age,contact,queue_no,app_date,app_time,
doctor, department, c_fee)
VALUES("{}","{}",{},{},{},{},"{}","{}","{}","{}").format(aa1,aa2,aa3,aa4,aa5,aa6,aa7,aa8,aa9,aa10))
        roota.destroy()

```



```

tkinter.messagebox.showinfo("V tReat DATABASE SYSTEM","APPOINTMENT BOOKED IN DATABASE")
conn.commit()
conn.close()

#exit from appointments form
def EXO():
    roota.destroy()

#function for appointments form help
def hp():
    tkinter.messagebox.showinfo("V tReat DATABASE SYSTEM","For any help regarding the software \n
CONTACT the creators using the details given in the About Section")

def ab():
    tkinter.messagebox.showinfo("V tReat DATABASE SYSTEM"," The software is designed and created by
\n \n Vihaan S. Kumar Contact-456789xxxx\n \n Rohan Nag Contact-789659xxxx \n \n Email-
pydeveloper@gmail.com \n \n©2020")

'''=====APPTDELSU_CODE====='''
#variables
rootU=None
rootD=None
rootS=None
head=None
inp_s=None
searchB=None
#display/search button

def Search_button():
    global inp_s,entry,errorS,t,i,q,dis1,dis2,dis3,dis4,dis5,dis6,dis7,dis8,dis9,dis10
    global l1,l2,l3,l4,l5,l6,l7,l8,l9
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    frameS.place_forget() #forget hides the frame

    if (s_var.get()=="ID"):
        inp_s=int(entry.get())
        c.execute('select * from appointments where app_id=%d;%s'%inp_s)
        p=c.fetchall()
    elif (s_var.get()=="NAME"):
        c.execute('select * from appointments where name=%s;%s'%entry.get())
        p=c.fetchall()
    elif (s_var.get()=="DOCTOR"):
        c.execute('select * from appointments where doctor=%s;%s'%d_sel.get())

```

```

p=c.fetchall()

if (len(p)==0):
    messagebox.showerror("Error","NO APOINTMENTS WITH SUCH ID/NAME;", parent=rootS)
else:
    c.execute('SELECT * FROM APPOINTMENTS where app_id=%d;%inp_s)
    t=c.fetchall()
    for i in t:
        app_det=tkinter.Label(rootS,text="APPOINTMENTS DETAILS",font="Arial 24 bold",fg="light gray",
bg="#022B63")
        l1=tkinter.Label(rootS,text="APPOINTMENT ID",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        dis1=tkinter.Label(rootS,text=i[0])
        l2=tkinter.Label(rootS,text="PATIENT NAME",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis2=tkinter.Label(rootS,text=i[1])
        l3=tkinter.Label(rootS,text="GENDER",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis3=tkinter.Label(rootS,text=i[2])
        l4=tkinter.Label(rootS,text="AGE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis4=tkinter.Label(rootS,text=i[3])
        l5=tkinter.Label(rootS,text="CONTACT NUMBER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        dis5=tkinter.Label(rootS,text=i[4])
        l6=tkinter.Label(rootS,text="QUEUE NO.",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis6=tkinter.Label(rootS,text=i[5])
        l7=tkinter.Label(rootS,text="APPOINTMENT DATE",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        dis7=tkinter.Label(rootS,text=i[6])
        l8=tkinter.Label(rootS,text="APPOINTMENT TIME",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        dis8=tkinter.Label(rootS,text=i[7])
        l9=tkinter.Label(rootS,text="CONSULTING DOCTOR ",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        dis9=tkinter.Label(rootS,text=i[8])
        l10=tkinter.Label(rootS,text="DEPARTMENT",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        dis10=tkinter.Label(rootS,text=i[9])

    app_det.grid(row=0, column=0, pady=5, padx=2,columnspan=2)
    l1.grid(row=1, column=0, pady=5, padx=2)
    dis1.grid(row=1, column=1, pady=5, padx=2)
    l2.grid(row=2, column=0, pady=5, padx=2)
    dis2.grid(row=2, column=1, pady=5, padx=2)
    l3.grid(row=3, column=0, pady=5, padx=2)
    dis3.grid(row=3, column=1, pady=5, padx=2)

```

```

l4.grid(row=4, column=0, pady=5, padx=2)
dis4.grid(row=4, column=1, pady=5, padx=2)
l5.grid(row=5, column=0, pady=5, padx=2)
dis5.grid(row=5, column=1, pady=5, padx=2)
l6.grid(row=6, column=0, pady=5, padx=2)
dis6.grid(row=6, column=1, pady=5, padx=2)
l7.grid(row=7, column=0, pady=5, padx=2)
dis7.grid(row=7, column=1, pady=5, padx=2)
l8.grid(row=8, column=0, pady=5, padx=2)
dis8.grid(row=8, column=1, pady=5, padx=2)
l9.grid(row=9, column=0, pady=5, padx=2)
dis9.grid(row=9, column=1, pady=5, padx=2)
l10.grid(row=10, column=0, pady=5, padx=2)
dis10.grid(row=10, column=1, pady=5, padx=2)

```

```

conn.commit()

```

```

def eXO():

```

```

    rootS.destroy()

```

```

##search window

```

```

def A_Display():

```

```

    global rootS,head,entry,searchB, frameS, s_var,d_sel

```

```

    rootS=tkinter.Tk()

```

```

    rootS.title("SEARCH WINDOW")

```

```

    frameS=tkinter.Frame(rootS)

```

```

    head=tkinter.Label(frameS,text="Search Using:",font="Arial 12 bold",fg="light gray", bg="#022B63")

```

```

    list=["NAME", "ID","DOCTOR"]

```

```

    s_var=StringVar(rootS)

```

```

    searchL=tkinter.OptionMenu(frameS, s_var, *list)

```

```

    data=tkinter.Label(frameS, text="Enter the Search ID/Name")

```

```

    entry=tkinter.Entry(frameS)

```

```

    searchB=tkinter.Button(frameS,text='SEARCH',command=Search_button)

```

```

def field():

```

```

    if s_var.get()=="DOCTOR":

```

```

        conn=pymysql.connect(host="localhost", user="root", password="password",
database="VTREAT")

```

```

        c=conn.cursor()

```

```

        c.execute("select name from employee where department='Doctor';")

```

```

        rows=c.fetchall()

```

```

        d_sel=tkinter.StringVar(rootS)

```

```

        d_sel.set(rows[0])

```

```

app_CT=tkinter.OptionMenu(frameS, d_sel, *rows)
app_CT.grid(row=2,column=0)
searchB.grid(row=3,column=0,columnspan=3)

elif (s_var.get()=="ID") or (s_var.get()=="NAME"):
    data.grid(row=1, column=0)
    entry.grid(row=1,column=1)
    searchB.grid(row=3,column=0,columnspan=3)

bt=tkinter.Button(frameS, text='Confirm Field',command=field)
s_var.set("Select the desired Field")
menubar= tkinter.Menu(rootS)
filemenu = tkinter.Menu(menubar, tearoff=0)
filemenu.add_command(label="NEW", command=A_Display)
filemenu.add_separator()
filemenu.add_command(label="EXIT", command=eXO)
menubar.add_cascade(label="File", menu=filemenu)
rootS.config(menu=menubar)

head.grid(row=0,column=0)
searchL.grid(row=0,column=1)
bt.grid(row=0, column=2)
frameS.place(x=0,y=0)
rootS.mainloop()
#DELETE BUTTON
def Delete_button():
    global inp_d,entry1,errorD,disd1
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    inp_d = int(entry1.get())
    c.execute("select * from APPOINTMENTS where app_id=%d;"%inp_d)
    p=c.fetchall()
    if (len(p)==0):
        tkinter.messagebox.showerror("VtReat Database System", "The appointment id is not valid.")

    else:
        c.execute("DELETE FROM APPOINTMENTS where app_id=%d;"%inp_d)
        tkinter.messagebox.showinfo("VtReat Database System", "The appointment has been permanently
deleted")
        conn.commit()
        rootD.destroy()
        rootP.destroy()

```

```
## DELETE SCREEN
```

```
def D_display():
```

```
    global rootD,headD,inp_d,entry1,DeleteB
    rootD=tkinter.Tk()
    rootD.title("DELETE WINDOW")
    headD=tkinter.Label(rootD,text="ENTER PATIENT ID TO DELETE",fg="blue")
    entry1=tkinter.Entry(rootD)
    DeleteB=tkinter.Button(rootD,text="DELETE",command=Delete_button)
    headD.pack()
    entry1.pack()
    DeleteB.pack()
    rootD.mainloop()
```

```
##-----APPOINTMENTS UPDATE SCREEN -----##
```

```
def A_UPDATE():
```

```
    global appu_ID,appu_name, appu_gender, appu_age,appu_date,appu_contact,appu_CT,
n,appu_dept,appu_qn, appu_time, feeu_var
    global rootU, regform, id, name, dob, gender, email, ct, addr, c1, c2, bg, SUBMIT, menubar, filemenu,
p1f, p2f,HEAD
```

```
    rootU = tkinter.Tk()
    rootU.title("APPOINTMENT UPDATE WINDOW")
    menubar = tkinter.Menu(rootU)
    filemenu = tkinter.Menu(menubar, tearoff=0)
    filemenu.add_command(label="Help", command=hp)
    filemenu.add_command(label="About", command=ab)
    filemenu.add_command(label="EXIT", command=EXITT)
    rootU.config(menu=menubar)
    menubar.add_cascade(label="File", menu=filemenu)
```

```
def Pupdate_box():
```

```
    global appu_ID,appu_name, appu_gender, appu_age,appu_date,appu_contact,appu_CT,
n,appu_dept,appu_qn, appu_time, feeu_var
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    n=int(appu_ID.get())
    c.execute('Select * from APPOINTMENTS where app_id=%d'%n)
    p=c.fetchall()
    if ((len(p))!=0):
        name=tkinter.Label(rootU,text="PATIENT NAME",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
        appu_name = tkinter.Entry(rootU,width=50)

        gender=tkinter.Label(rootU,text="GENDER",font=("Arial",12,"bold"), fg="#022B63", bg="white")
        appu_gender=tkinter.Listbox(rootU,selectmode='SINGLE', exportselection=0, height=1, width=50)
```

```

appu_gender.insert(tkinter.END, "M")
appu_gender.insert(tkinter.END, "F")

age=tkinter.Label(rootU, text="AGE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
appu_age=tkinter.Entry(rootU,width=50)

c1=tkinter.Label(rootU, text="CONTACT NUMBER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
appu_contact=tkinter.Entry(rootU,width=50)

qn=tkinter.Label(rootU, text="QUEUE NO.",font=("Arial",12,"bold"), fg="#022B63", bg="white")
appu_qn=tkinter.Entry(rootU,width=50)

date=tkinter.Label(rootU, text="APPOINTMENT DATE",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
appu_date=tkinter.Entry(rootU, width=50)

time=tkinter.Label(rootU, text="APPOINTMENT TIME",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
appu_time=tkinter.Entry(rootU,width=50)

ct=tkinter.Label(rootU,text="CONSULTING DOCTOR",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
appu_CT=tkinter.Entry(rootU,width=50)

dept=tkinter.Label(rootU,text="DEPARTMENT",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
depts=["General_Physician","Chest_Physician","Paediatrician","General_Surgeon","Cardiologist"]
appu_dept=tkinter.StringVar(rootU)
appu_d=tkinter.OptionMenu(rootU, appu_dept, *depts)
appu_d.config(width=20)

fee=tkinter.Label(rootU,text="CONSULTATION FEE(Rs.)",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
fee_l=[300,500,1000]
feeu_var=StringVar(rootU)
appu_fee=tkinter.OptionMenu(rootU, feeu_var, *fee_l)
c.execute('Select * from APPOINTMENTS where app_id={}'.format(appu_ID.get()))
t=c.fetchall()
for i in t:
    appu_name.insert(0,i[1])
    appu_gender.insert(0,i[2])
    appu_age.insert(0,i[3])
    appu_contact.insert(0,i[4])

```

```
appu_qn.insert(0,i[5])
appu_date.insert(0,i[6])
appu_time.insert(0,i[7])
appu_CT.insert(0,i[8])
appu_dept.set(i[9])
feeu_var.set(i[10])
```

```
UPDATE=tkinter.Button(rootU,text="UPDATE",command=up1,font=("Arial",12,"bold"),
fg="white", bg="#022B63") #,bd=0)
```

```
name.grid(row = 2, column = 0, pady=5,sticky=W)
appu_name.grid(row = 2, column = 1, pady=5)
gender.grid(row = 3, column = 0, pady=5,sticky=W)
appu_gender.grid(row = 3, column = 1, pady=5)
age.grid(row = 4, column = 0, pady=5,sticky=W)
appu_age.grid(row = 4, column = 1, pady=5)
c1.grid(row = 5, column = 0, pady=5,sticky=W)
appu_contact.grid(row = 5, column = 1, pady=5)
qn.grid(row = 6, column = 0, pady=5,sticky=W)
appu_qn.grid(row = 6, column = 1, pady=5)
date.grid(row=7, column=0,pady=5,sticky=W)
appu_date.grid(row = 7, column = 1, pady=5)
time.grid(row=8, column=0,pady=5,sticky=W)
appu_time.grid(row = 8, column = 1, pady=5)
ct.grid(row = 9, column = 0, pady=5, padx=2, sticky=W)
appu_CT.grid(row = 9, column = 1, pady=5)
dept.grid(row = 10, column = 0, pady=5,sticky=W)
appu_d.grid(row = 10, column = 1, pady=5)
fee.grid(row = 11, column = 0, pady=5,sticky=W)
appu_fee.grid(row = 11, column = 1, pady=5)
```

```
UPDATE.grid(row = 12, column = 0)
```

```
else:
```

```
tkinter.messagebox.showerror("V tReat DATABSE SYSTEM", "APPOINTMENT NOT BOOKED")
```

```
upform=tkinter.Label(rootU,text="UPDATE FORM",font="Arial 24 bold",fg="light gray",
bg="#022B63")
```

```
id=tkinter.Label(rootU,text="ID OF PATIENT TO UPDATE",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
```

```
appu_ID=tkinter.Entry(rootU,width=50)
```

```
id_bt=tkinter.Button(rootU, text="Search",command=Pupdate_box,font=("Arial",10,"bold"),
fg="white", bg="#022B63",bd=0,width=23)
```

```
upform.grid(row=0, column=0,columnspan=2)
```

```

id.grid(row = 1, column = 0, pady=5,sticky=W)
appu_ID.grid(row = 1, column = 1, pady=5)
id_bt.grid(row = 1, column = 2, pady=5)

rootU.mainloop()

#Update command of button
def up1():
    global u1, u2, u3, u4, u5, u6, u7, u8, u9, u10, u11
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    u1=appu_ID.get()
    u2=appu_name.get()
    u3=appu_gender.get(tkinter.ACTIVE)
    u4=appu_age.get()
    u5=appu_contact.get()
    u6=appu_qn.get()
    u7=appu_date.get()
    u8=appu_time.get()
    u9=appu_CT.get()
    u10=appu_dept.get()
    u11=feeu_var.get()
    c.execute('UPDATE APPOINTMENTS SET
name="{",gender="{",age={},contact={},queue_no={},app_date="{",app_time="{", doctor="{",
department="{", c_fee={} where app_id={};'.format( u2, u3, u4, u5, u6,u7,u8,u9,u10,u11,u1))
    tkinter.messagebox.showinfo("V tReat DATABSE SYSTEM", "DETAILS UPDATED INTO DATABASE")
    conn.commit()
    rootU.destroy()
def EXITT():
    rootU.destroy()

```


BILL GENERATOR

+ PATIENT BILL

PATIENT BILL

PATIENT ID Confirm

+ PATIENT BILL

PATIENT BILL

PATIENT ID

PATIENT NAME DATE OF ADMISSION

GENDER DATE OF DISCHARGE

AGE CATEGORY

CONTACT NUMBER CONSULTING TEAM / DOCTOR

DEPARTMENT

Star Health
ICICI
MaxBupa
Bajaj Allianz
Aditya Birla

+ PATIENT BILL

PATIENT BILL

PATIENT ID

PATIENT NAME DATE OF ADMISSION

GENDER DATE OF DISCHARGE

AGE CATEGORY

CONTACT NUMBER CONSULTING TEAM / DOCTOR

DEPARTMENT

S.No.	CONTENTS	Cost per Day	TOTAL
1	{Double Room} // 121	5000	35000
2	Surgery- Angioplasty	--	120000
3	Test- CT_Scan	--	1700
4	Test- CT_Scan	--	1700
5	Test- CT_Scan	--	1700
6	Test- Biochemistry_Test	--	800
<input type="button" value="SUBTOTAL"/>			160900

BILL GENERATOR_CODE

```
import tkinter
from tkinter import*
import datetime
from datetime import date
import pymysql

def BILL():
    global pat_ID
    rootB=Tk()
    rootB.title("PATIENT BILL")

    frame1=Frame(rootB, bg='light gray')
    frame1.place(x=0, y=0, relheight=1, relwidth=1)
    b_head=tkinter.Label(frame1,text="PATIENT BILL",font="Arial 24 bold",fg="light gray", bg="#022B63")
    id=tkinter.Label(frame1,text="PATIENT ID",font=("Arial",12,"bold"), fg="#022B63", bg="white")
    pat_ID=tkinter.Entry(frame1,width=10)
    def details():
        global pat_name,date_ad, datet, cat_var

        id_bt.grid_forget()
        conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
        c=conn.cursor()
        c.execute("select* from patient where id=%d;"%int(pat_ID.get()))
        rows=c.fetchall()
        if len(rows)!=0:
            pat_ID.config(state='disabled')
            for row in rows:
                name=tkinter.Label(frame1, text=row[1])
                pat_name=row[1]
                gender=tkinter.Label(frame1, text=row[2])
                age=tkinter.Label(frame1, text=row[3])
                con=tkinter.Label(frame1, text=row[5])
                date_ad=tkinter.Label(frame1, text=row[4])
                ct=tkinter.Label(frame1, text=row[8])
                dept=tkinter.Label(frame1, text=row[9])
            else:
                rootB.destroy()
                tkinter.messagebox.showerror("VtReat Database System","No patient with this ID is registered")

        namel=tkinter.Label(frame1,text="PATIENT NAME",font=("Arial",12,"bold"), fg="#022B63",
        bg="white")
```

```

gender=tkinter.Label(frame1,text="GENDER",font=("Arial",12,"bold"), fg="#022B63", bg="white")
age=tkinter.Label(frame1, text="AGE",font=("Arial",12,"bold"), fg="#022B63", bg="white")
conl=tkinter.Label(frame1, text="CONTACT NUMBER",font=("Arial",12,"bold"), fg="#022B63",
bg="white")

date_adl=tkinter.Label(frame1, text="DATE OF ADMISSION",font=("Arial",12,"bold"), fg="#022B63",
bg="white")

def add_date():
    global now, pat_date
    dt_bt.grid_forget()
    now = datetime.datetime.now()
    pat_date=now.strftime("%Y-%m-%d %H:%M:%S")
    datet=tkinter.Label(frame1,width=20, text=pat_date)
    datet.grid(row =3, column =3)

date_disl=tkinter.Label(frame1, text="DATE OF DISCHARGE",font=("Arial",12,"bold"), fg="#022B63",
bg="white")
dt_bt=tkinter.Button(frame1,text="Add current
date/time",command=add_date,font=("Arial",10,"bold"), fg="white", bg="#022B63",bd=0,width=20)

catl=tkinter.Label(frame1, text="CATEGORY",font=("Arial",12,"bold"), fg="#022B63", bg="white")
cat_list=["General", "HDFC", "Star Health", "ICICI", "MaxBupa", "Bajaj Allianz", "Aditya Birla"]
cat_var=StringVar(rootB)
cat_var.set(cat_list[0])
cat=tkinter.OptionMenu(frame1, cat_var, *cat_list)

ctl=tkinter.Label(frame1,text="CONSULTING TEAM / DOCTOR",font=("Arial",12,"bold"),
fg="#022B63", bg="white")
deptl=tkinter.Label(frame1,text="DEPARTMENT",font=("Arial",12,"bold"), fg="#022B63",
bg="white")

def gen_bt():
    global date1
    conn=pymysql.connect(host="localhost", user="root", password="password",
database="VTREAT")
    c=conn.cursor()
    c.execute("select* from patient where id=%d;"%int(pat_ID.get()))
    rows=c.fetchall()
    for row in rows:
        date1=row[4]
        room_ty=row[11]
        room_no=row[12]

```

```

        room_d=room_ty,"/",room_no
        room_p=row[13]

        date2=now.date()
        date1=date1.date()
        print(date1)
        print(date2)
        sno=tkinter.Label(frame1, text="S.No.", font=("Arial",12,"bold"), fg="#022B63", bg="white",
width=15)
        content=tkinter.Label(frame1, text="CONTENTS", font=("Arial",12,"bold"), fg="#022B63",
bg="white", width=30)
        perday=tkinter.Label(frame1, text="Cost per Day", font=("Arial",12,"bold"), fg="#022B63",
bg="white", width=20)
        total=tkinter.Label(frame1, text="TOTAL", font=("Arial",12,"bold"), fg="#022B63", bg="white",
width=20)

        no=tkinter.Label(frame1, text=1,font=("Arial",12,"bold"),bg="light gray")
        c_rt=tkinter.Label(frame1, text=room_d,font=("Arial",12,"bold"),bg="light gray")
        price_label=tkinter.Label(frame1, text=room_p,font=("Arial",12,"bold"),bg="light gray")

        day=(date2-date1).days
        price=int(room_p*day)
        p=tkinter.Label(frame1, text=price,font=("Arial",12,"bold"),bg="light gray")
        sno.grid(row=8, column=0, padx=3)
        content.grid(row=8, column=1)
        perday.grid(row=8, column=2)
        total.grid(row=8, column=3)
        no.grid(row=9, column=0)
        c_rt.grid(row=9, column=1, stick=W)
        price_label.grid(row=9, column=2)
        p.grid(row=9, column=3)

surg_list=["Knee_Replacement","Kidney_Transplant","Tonsil_Surgery","Angioplasty","Coronary_Artery_
Bypass","Laparoscopic_Cholecystectomy"]
        test_list=["Biochemistry_Test","Haematology_Test","X-Ray", "CT_Scan",
"Thyroid_Test","Insulin","COVID_RTPCR","COVID_Antibody","Antigen_Test","Sonography","ECG"]

        c.execute("Select `test/surgery`,cost from sur_test where id=%d;"%int(pat_ID.get()))
        rows=c.fetchall()
        print(rows)
        a=10
        subt=price
        n=2
        for row in rows:

```

```

print(row)

if row[0] in surg_list:
    s="Surgery-",row[0]
    no=tkinter.Label(frame1, text=n,font=("Arial",12,"bold"), bg="light gray")
    n=n+1
    subt+=int(row[1])

    surgl=tkinter.Label(frame1, text=s,font=("Arial",12,"bold"),bg="light gray")
    hyphen=tkinter.Label(frame1, text="--",font=("Arial",12,"bold"),bg="light gray")
    cost=tkinter.Label(frame1, text=row[1],font=("Arial",12,"bold"), bg="light gray")
    no.grid(row=a, column=0)
    surgl.grid(row=a , column=1, stick=W, padx=5)
    hyphen.grid(row=a , column=2)
    cost.grid(row=a , column=3)
    a=a+1

elif row[0] in test_list:
    t="Test-",row[0]
    no=tkinter.Label(frame1, text=n,font=("Arial",12,"bold"), bg="light gray")
    n=n+1
    subt+=int(row[1])

    testl=tkinter.Label(frame1, text=t,font=("Arial",12,"bold"), bg="light gray")
    hyphen=tkinter.Label(frame1, text="--",font=("Arial",12,"bold"),bg="light gray")
    cost=tkinter.Label(frame1, text=row[1],font=("Arial",12,"bold"),bg="light gray")
    no.grid(row=a, column=0)
    testl.grid(row=a , column=1, stick=W, padx=5)
    hyphen.grid(row=a , column=2)
    cost.grid(row=a , column=3)
    a=a+1

def tot():
    final=tkinter.Label(frame1, text=subt, font=("Arial",16,"bold"), bg="light gray")
    final.grid(row=a, column=3)
    conn=pymysql.connect(host="localhost", user="root", password="password",
database="VTREAT")
    c=conn.cursor()
    bb1=pat_ID.get()
    bb2=pat_name
    bb3=date1
    bb4=pat_date
    bb5=cat_var.get()
    bb6=subt

```

```

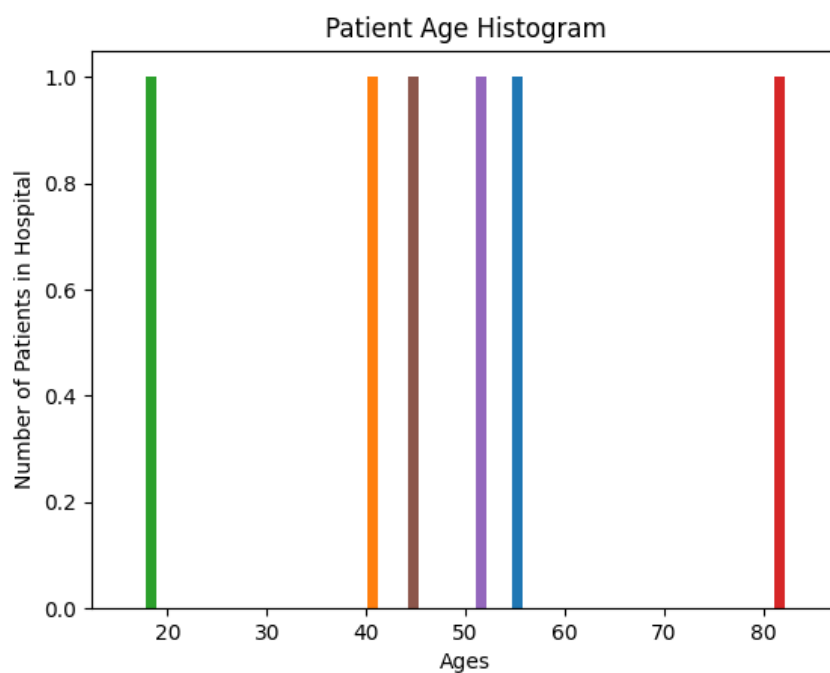
t_bt=tkinter.Button(frame1, text="SUBTOTAL", command=tot,font=("Arial",12,"bold"),
fg="white", bg="#022B63")
t_bt.grid(row=a, column=2)

genbt=tkinter.Button(frame1, text="Generate Bill", command=gen_bt,font=("Arial",12,"bold"),
fg="white", bg="#022B63")

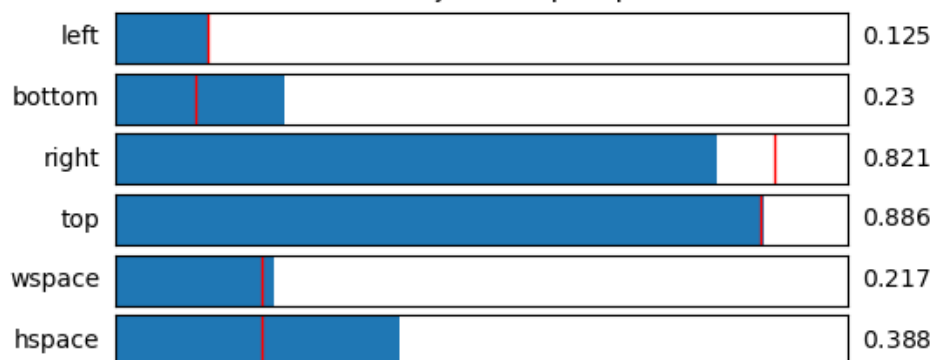
```

AGE-GRAPH

Figure 1



Click on slider to adjust subplot param



Reset

AGE-GRAPH_CODE

##Included in Main Menu Code file

```
def gp():
    global root2
    conn=pymysql.connect(host="localhost", user="root", password="password", database="VTREAT")
    c=conn.cursor()
    c.execute("select age from PATIENT;")
    ages=c.fetchall()
    plt.hist(ages,bins=9)
    plt.title("Patient Age Histogram")
    plt.ylabel("Number of Patients in Hospital")
    plt.xlabel("Ages")
    plt.show()
```


FUTURE DEVELOPMENT

Though the software is well developed and thoughtfully crafted, strives to make it better will never stop. There are still various aspects which the software can encompass to make it more effective in usage. This hospital management software may have to contain many more tables to cater to the needs of larger hospitals. The software can also be optimized for diagnostic labs other than hospitals. Though the software has been designed to be data centric, it can have better visual graphics to be more appealing to the users.

BIBLIOGRAPHY

The following resources have been used for learning various functions about tkinter and requirements of a hospital management software:

- Requirements of Basic Hospital Management Software:
www.github.com
- Matplot Lib with tkinter(Codemy):
<https://www.youtube.com/watch?v=8exB6Ly3nx0&t=405s>
- Matplot Lib charts with tkinter(Codemy):
<https://www.youtube.com/watch?v=OPUSBBD2OJw&t=354s>
- Understanding tkinter:
<https://www.youtube.com/watch?v=tUc6FMPSZDg&t=1199s>