**NAVRACHANA HIGHER SECONDARY SCHOOL, SAMA**



**ACADEMIC YEAR: 2022-23**

**PROJECT REPORT ON**

**HOSPITAL MANAGEMENT SYSTEM**

**ROLL NO :**

**NAME : Rupankar Majumdar**

**CLASS : 12 C**

**SUBJECT : COMPUTER SCIENCE**

**SUB CODE : 083**

**PROJECT GUIDE : Smriti Upadhyay**



**CERTIFICATE**

This is to certify that **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Roll No: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

has successfully completed the Project Work entitled **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** in the subject Computer Science (083) laid down in the regulations of CBSE for the purpose of Practical Examination in Class XII to be held in Navrachana Higher Secondary School, Sama Year 2022-23.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

**External Examiner**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature Signature

**Internal Examiner Principal**

**Smriti Upadhyay Ms. Suprabha Menon**

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS [TOC]** | | |
| **SR. NO** | **DESCRIPTION** | **PAGE NO** |
| 01 | COVER PAGE | 01 |
| 02 | CERTIFICATE | 02 |
| 03 | TABLE OF CONTENTS | 03 |
| 04 | ACKNOWLEDGEMENTS | 04 |
| 05 | INTRODUCTION | 05 |
| 06 | OBJECTIVES OF THE PROJECT | 05 |
| 07 | FLOW CHART | 06 |
| 08 | SOURCE CODE | 07 |
| 09 | OUTPUT | 13 |
| 10 | HARDWARE AND SOFTWARE REQUIREMENTS | 17 |
| 11 | BIBLIOGRAPHY | 17 |

**ACKNOWLEDGEMENT**

Apart from my efforts, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I express my heartfelt gratitude to my parents for constant encouragement while carrying out this project.

I am grateful to my friends and peers who helped me by giving me ideas whenever I got stuck somewhere in my code.

I gratefully acknowledge the contribution of the individuals who contributed in bringing this project up to this level, who continues to look after me despite my flaws.

My sincere thanks toSmriti Upadhyay project guide and computer science teacher who reviewed my project and helped in solving problems that occurred during implementation of the project.

**HOSPITAL MANAGEMENT SYSTEM**

**INTRODUCTION**

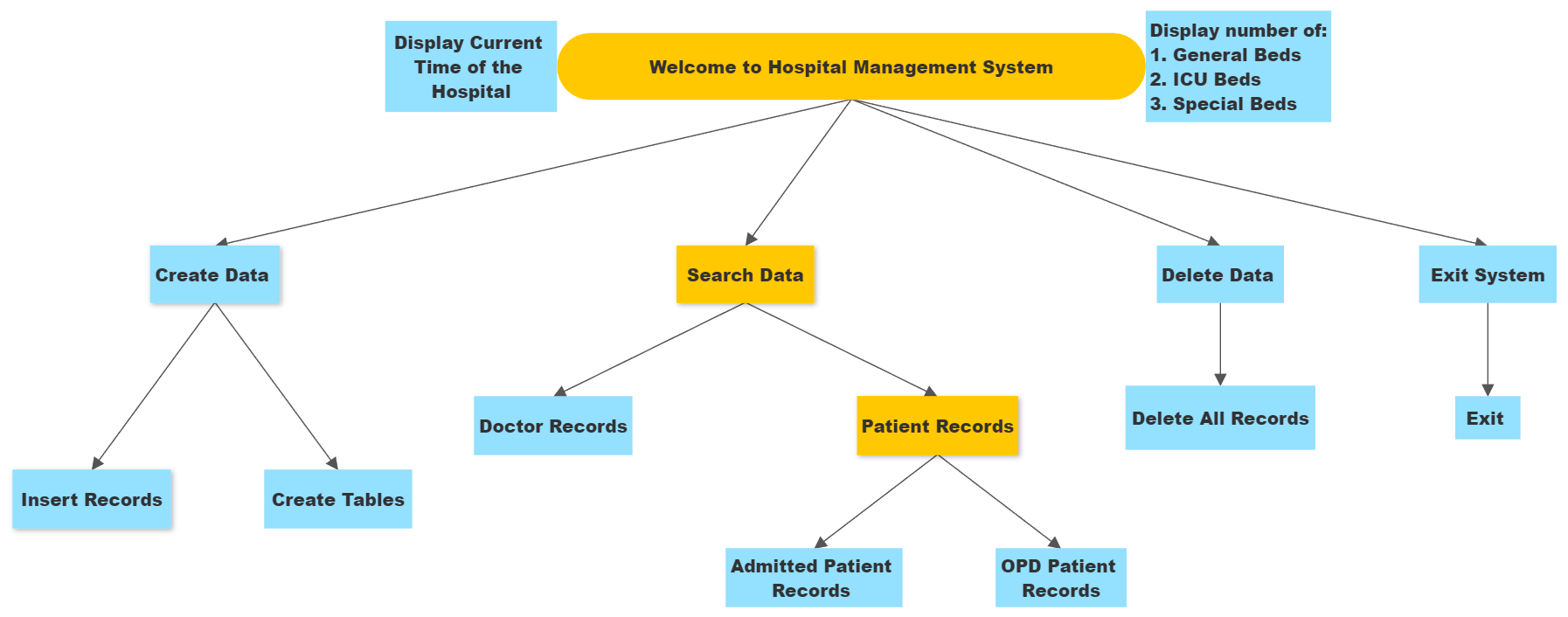
Hospital Management System is a python program which digitalizes the various documentations done in a hospital and thus reduces paperwork. This system reduces effort required to record patient data by reducing manual work as records can now be uploaded by clicking the right options and filling the required fields. This ensures that a permanent record is kept of all those who have been admitted or received services in the hospital. Since the data is stored at one place, it is not scattered and records retrieval and register maintenance have been made easy. Hospital Management System provides a user-friendly way to access and upload data and its flexibility ensures that accurate information is provided timely and without fail. The workload of the user is reduced as reports are generated automatically.

**OBJECTIVES OF THE PROJECT**

The objectives of this project are to:

* Create a user-friendly system of uploading and accessing data.
* Help the hospital staff admit patients while checking the availability of hospital facilities.
* Retrieve and update patient data without lengthy procedures.
* Display records of hospital staff, including doctors.
* Save the environment by providing a viable alternative to paperwork.
* Reduce redundancy and cost of maintenance.
* Digitalizing records help reduce physical space required to store printed/handwritten files.

**FLOW CHART**



**SOURCE CODE**

import mysql.connector as sql

from tkinter import \*

from tkinter.ttk import \*

from tkinter import messagebox

from time import strftime

import random

mydb=sql.connect(host="localhost",user="root",passwd="admin",database="health\_hospital")

mycur=mydb.cursor()

**def create\_table():**

"""To Create Tables in Database"""

mycur.execute('create table hosp\_details(g\_bed int(2), icu\_room int(2), special\_room int(2))')

mycur.execute('create table patient\_details(p\_id char(4) primary key,p\_name char(25) ,p\_age int(3),p\_problems char(40),\

p\_phono char(10), days int(2), room char(15), room\_no int(3), Total\_Bill float(10))')

mycur.execute('create table doctor\_details(d\_name char(25) primary key,d\_age int(3),d\_department char(40),d\_phono char(10))')

mycur.execute('create table worker\_details(w\_name char(25) primary key,w\_age int(3),w\_workname char(40),w\_phono char(10))')

mycur.execute('create table patient\_details\_opd(p\_id char(4) primary key,p\_name char(25) ,p\_age int(3),p\_problems char(40),\

p\_phono char(10), Total\_Bill float(10))')

mydb.commit()

messagebox.showinfo("Alert", "Tables Created")

**def Insert\_Hosp():**

"""To insert data into Hospital Database"""

# GUIconfig

hosp = Tk()

hosp.title("Insert Hospital Records")

# Commands

**def getdata():**

g\_beds = e1.get()

icu\_room = e2.get()

special\_room = e3.get()

try:

mycur.execute("truncate table hosp\_details") #Will delete previous bed data

mydb.commit()

mycur.execute("insert into hosp\_details values({},{},{})".format(g\_beds, icu\_room, special\_room))

mydb.commit()

messagebox.showinfo("Alert", "Record Submitted")

beddata()

updatebeddata()

except:

er = "Some Error May have Occured, Please recheck"

messagebox.showinfo("Alert", er)

# Label Fields

cd1 = Label(hosp, text="Enter No.of General Beds", font="Arial")

cd2 = Label(hosp, text="Enter No.of ICU Rooms", font="Arial")

cd3 = Label(hosp, text="Enter No.of Special Rooms", font="Arial")

cd1.grid(row=0, column=0)

cd2.grid(row=1, column=0)

cd3.grid(row=2, column=0)

# Entry Fields

e1 = Entry(hosp, textvariable=IntVar)

e2 = Entry(hosp, textvariable=IntVar)

e3 = Entry(hosp, textvariable=IntVar)

e1.grid(row=0, column=1)

e2.grid(row=1, column=1)

e3.grid(row=2, column=1)

# button

Submit\_button = Button(hosp, text="Submit", command=getdata)

Submit\_button.grid(row=5, column=0)

mainloop()

**def Update\_Hosp\_S():**

"""To delete a special bed from database as it is booked by patient"""

mycur.execute("update hosp\_details set special\_room=special\_room-1")

mydb.commit()

updatebeddata()

**def Update\_Hosp\_ICU():**

"""To delete an ICU bed from database as it is booked by patient"""

mycur.execute("update hosp\_details set icu\_room=icu\_room-1")

mydb.commit()

updatebeddata()

**def Update\_Hosp\_GEN():**

"""To delete a general bed from database as it is booked by patient"""

mycur.execute("update hosp\_details set g\_bed=g\_bed-1")

mydb.commit()

updatebeddata()

**def Insert\_Doctors():**

"""To enter doctors data in database"""

# GUIconfig

doct = Tk()

doct.title("Insert Doctor Records")

# Commands

**def getdata():**

"""To get data inserted by user in text field"""

def allcheck():

"""TO check whether data inserted have general errors or not"""

a=True

if len(e4.get()) != 10:

messagebox.showerror("Alert", "Please Enter 10 Digit Number")

a = False

if e2.get()==0:

messagebox.showerror("Alert", "Please Enter Correct Age")

a = False

return a

if allcheck()==True:

d\_name = e1.get()

d\_age = e2.get()

d\_department = e3.get()

d\_phono = e4.get()

try:

mycur.execute("insert into doctor\_details values('{}',{},'{}','{}')".format(d\_name,d\_age,d\_department,d\_phono))

mydb.commit()

messagebox.showinfo("Alert", "Record Submitted")

except:

er = "Some Error May have Occured, Please recheck"

messagebox.showinfo("Alert", er)

else:

pass

# Label Fields

cd1 = Label(doct, text="Enter Doctor Name :", font="Arial").grid(row=0, column=0)

cd2 = Label(doct, text="Enter Age :", font="Arial").grid(row=1, column=0)

cd3 = Label(doct, text="Enter Department :", font="Arial").grid(row=2, column=0)

cd4 = Label(doct, text="Enter Phone No. :", font="Arial").grid(row=3, column=0)

# Entry Fields

var1 = StringVar()

var2 = StringVar()

var3 = StringVar()

var4 = StringVar()

e1 = Entry(doct, textvariable=var1)

e2 = Entry(doct, textvariable=var2)

e3 = Entry(doct, textvariable=var3)

e4 = Entry(doct, textvariable=var4)

e1.grid(row=0, column=1)

e2.grid(row=1, column=1)

e3.grid(row=2, column=1)

e4.grid(row=3, column=1)

# button

Submit\_button = Button(doct, text="Submit", command=getdata)

Submit\_button.grid(row=5, column=0)

mainloop()

**def Insert\_Patient():**

"""To enter patient data in database"""

pat = Tk()

pat.title("Insert Patients")

**def submit():**

def select\_room():

r\_choice = 0

r\_no=0

if menu.get()==options[0]:

Update\_Hosp\_GEN()

g\_ex\_lt = []

mycur.execute("Select \* from patient\_details")

for x in mycur:

l = x[7]

g\_ex\_lt.append(l)

a=True

while a:

r\_choice = random.choice(g\_list)

if r\_choice in g\_ex\_lt:

a=True

else:

a=False

r\_no = int(r\_choice)

elif menu.get()==options[1]:

Update\_Hosp\_ICU()

g\_ex\_lt = []

mycur.execute("Select \* from patient\_details")

for x in mycur:

l = x[7]

g\_ex\_lt.append(l)

a=True

while a:

r\_choice = random.choice(i\_list)

if r\_choice in g\_ex\_lt:

a=True

else:

a=False

r\_no = int(r\_choice)

elif menu.get()==options[2]:

Update\_Hosp\_S()

g\_ex\_lt = []

mycur.execute("Select \* from patient\_details")

for x in mycur:

l = x[7]

g\_ex\_lt.append(l)

a=True

while a:

r\_choice = random.choice(s\_list)

if r\_choice in g\_ex\_lt:

a=True

else:

a=False

r\_no = int(r\_choice)

return r\_no

**def bed\_bill():**

b\_bill = 0

if menu.get()==options[0]:

b\_bill = int(sb1.get()) \* 1000

elif menu.get()==options[1]:

b\_bill = int(sb1.get()) \* 1500

elif menu.get()==options[2]:

b\_bill = int(sb1.get()) \* 2000

return b\_bill

**def allcheck():**

a=True

if menu.get()==options[0]:

if g\_beds==0:

messagebox.showerror("Error", "Sorry:( No General Beds Availaible")

a=False

elif menu.get() == options[1]:

if i\_beds==0:

messagebox.showerror("Error", "Sorry:( No ICU Beds Availaible")

a=False

elif menu.get()==options[2]:

if s\_beds==0:

messagebox.showerror("Error", "Sorry:( No Special Beds Availaible")

a=False

if len(e5.get())!=10:

messagebox.showerror("Error", "Please Enter 10 Digit Number")

a = False

if e3.get()==0:

messagebox.showerror("Error", "Please Enter Correct Age")

a = False

return a

if allcheck()==True:

r\_final = select\_room()

t\_bill = int(e8.get()) + int(e9.get()) + bed\_bill()

try:

mycur.execute("insert into patient\_details values('{}','{}',{},'{}','{}',{},'{}',{},{})".format(e1.get(), e2.get(),e3.get(), e4.get(), e5.get(), sb1.get(), menu.get(), r\_final, t\_bill))

mydb.commit()

alert\_msg="Record submitted succesfully, Total Bill is: " + str(t\_bill)

messagebox.showinfo("Alert", alert\_msg)

except Exception as e:

er = "Some Error May have Occured, Please recheck"

messagebox.showerror("Error", e)

else:

pass

# Label Fields

cd1 = Label(pat, text="Enter Patient Id :", font="Arial").grid(row=0, column=0)

cd2 = Label(pat, text="Enter Patient Name :", font="Arial").grid(row=1, column=0)

cd3 = Label(pat, text="Enter Patient Age :", font="Arial").grid(row=2, column=0)

cd4 = Label(pat, text="Enter Patient Problem:", font="Arial").grid(row=3, column=0)

cd5 = Label(pat, text="Enter Patient Contact", font="Arial").grid(row=4, column=0)

cd6 = Label(pat, text="Type of Bed :", font="Arial").grid(row=5, column=0)

cd7 = Label(pat, text="No.of Days :", font="Arial").grid(row=6, column=0)

cd8 = Label(pat, text="Medicine Charges :", font="Arial").grid(row=7, column=0)

cd9= Label(pat, text="Consultant Charges :", font="Arial").grid(row=8, column=0)

# Entry Fields

e1 = Entry(pat, textvariable=StringVar)

e2 = Entry(pat, textvariable=StringVar)

e3 = Entry(pat, textvariable=IntVar)

e4 = Entry(pat, textvariable=StringVar)

e5 = Entry(pat, textvariable=StringVar)

e8 = Entry(pat, textvariable=IntVar)

e9 = Entry(pat, textvariable=IntVar)

e1.grid(row=0, column=1)

e2.grid(row=1, column=1)

e3.grid(row=2, column=1)

e4.grid(row=3, column=1)

e5.grid(row=4, column=1)

e8.grid(row=7, column=1)

e9.grid(row=8, column=1)

#MENU

menu = StringVar(pat)

options = ["General Bed", "ICU Bed", "Special Bed"]

mn1 = OptionMenu(pat, menu, options[0], \*options)

mn1.grid(row=5,column=1)

#SpinBox

sb1 = Spinbox(pat, from\_=0, to=100)

sb1.grid(row=6, column=1)

bt1 = Button(pat, text="Submit", command=submit).grid(row=9, column=0)

**def Insert\_Patient\_OPD():**

"""To enter opd patient data in database"""

pat = Tk()

pat.title("Insert OPD Patients")

**def submit():**

def allcheck():

a=True

if len(e5.get())!=10:

messagebox.showerror("Error", "Please Enter 10 Digit Number")

a = False

if e3.get()==0:

messagebox.showerror("Error", "Please Enter Correct Age")

a = False

return a

if allcheck()==True:

t\_bill = int(e8.get()) + int(e9.get())

try:

mycur.execute("insert into patient\_details\_opd values('{}','{}',{},'{}','{}',{})".format(e1.get(), e2.get(),e3.get(), e4.get(), e5.get(), t\_bill))

mydb.commit()

alert\_msg="Record submitted succesfully, Total Bill is: " + str(t\_bill)

messagebox.showinfo("Alert", alert\_msg)

except Exception as e:

er = "Some Error May have Occured, Please recheck"

messagebox.showerror("Error", e)

else:

pass

# Label Fields

cd1 = Label(pat, text="Enter Patient Id :", font="Arial").grid(row=0, column=0)

cd2 = Label(pat, text="Enter Patient Name :", font="Arial").grid(row=1, column=0)

cd3 = Label(pat, text="Enter Patient Age :", font="Arial").grid(row=2, column=0)

cd4 = Label(pat, text="Enter Patient Problem:", font="Arial").grid(row=3, column=0)

cd5 = Label(pat, text="Enter Patient Contact", font="Arial").grid(row=4, column=0)

cd8 = Label(pat, text="Medicine Charges :", font="Arial").grid(row=7, column=0)

cd9= Label(pat, text="Consultant Charges :", font="Arial").grid(row=8, column=0)

# Entry Fields

e1 = Entry(pat, textvariable=StringVar)

e2 = Entry(pat, textvariable=StringVar)

e3 = Entry(pat, textvariable=IntVar)

e4 = Entry(pat, textvariable=StringVar)

e5 = Entry(pat, textvariable=StringVar)

e8 = Entry(pat, textvariable=IntVar)

e9 = Entry(pat, textvariable=IntVar)

e1.grid(row=0, column=1)

e2.grid(row=1, column=1)

e3.grid(row=2, column=1)

e4.grid(row=3, column=1)

e5.grid(row=4, column=1)

e8.grid(row=7, column=1)

e9.grid(row=8, column=1)

# button

bt1 = Button(pat, text="Submit", command=submit).grid(row=9, column=0)

**def Insert\_Workers():**

wrks = Tk()

wrks.title("Insert Workers")

**def getdata():**

def allcheck():

a=True

if len(e3.get()) != 10:

messagebox.showerror("Error", "Please Enter 10 Digit Number")

a = False

if e2.get()==0:

messagebox.showerror("Error", "Please Enter Correct Age")

a = False

return a

w\_name = e1.get()

w\_age = e2.get()

w\_phono = e3.get()

w\_workname=""

if radio.get()==1:

w\_workname = "Nurse"

if radio.get()==2:

w\_workname = "Receptionist"

if radio.get()==3:

w\_workname = "Compounder"

if radio.get()==4:

w\_workname = "Sweeper"

if radio.get()==5:

w\_workname = "Pharamacist"

mycur.execute("insert into worker\_details values('{}',{},'{}','{}')".format(w\_name, w\_age, w\_workname, w\_phono))

mydb.commit()

messagebox.showinfo("Alert", "Record Submitted")

# Label Fields

cd1 = Label(wrks, text="Enter Worker Name : ", font="Arial").grid(row=0, column=0)

cd1 = Label(wrks, text="Enter Worker Age : ", font="Arial").grid(row=1, column=0)

cd1 = Label(wrks, text="Enter Worker Contact: ", font="Arial").grid(row=2, column=0)

# Entry Fields

e1 = Entry(wrks, textvariable=StringVar)

e2 = Entry(wrks, textvariable=StringVar)

e3 = Entry(wrks, textvariable=StringVar)

e1.grid(row=0, column=1)

e2.grid(row=1, column=1)

e3.grid(row=2, column=1)

#RadioButton

radio = IntVar(wrks)

rd1 = Radiobutton(wrks, text="Nurse", variable=radio, value=1)

rd2 = Radiobutton(wrks, text="Receptionist", variable=radio, value=2)

rd3 = Radiobutton(wrks, text="Compounder", variable=radio, value=3)

rd4 = Radiobutton(wrks, text="Sweeper", variable=radio, value=4)

rd5 = Radiobutton(wrks, text="Pharamacist", variable=radio, value=5)

rd1.grid(row=3, column=0)

rd2.grid(row=3, column=1)

rd3.grid(row=4, column=0)

rd4.grid(row=4, column=1)

rd5.grid(row=5, column=0)

submit = Button(wrks, text="Submit",command=getdata)

submit.grid(row=6,column=0)

mainloop()

**def Display\_Hosp():**

hosp\_d = Tk()

hosp\_d.title("Hospital Details")

# Label Fields

cd1 = Label(hosp\_d, text="Enter No.of General Beds", font="Arial").grid(row=0, column=0)

cd2 = Label(hosp\_d, text="Enter No.of ICU Rooms", font="Arial").grid(row=1, column=0)

cd3 = Label(hosp\_d, text="Enter No.of Special Rooms", font="Arial").grid(row=2, column=0)

# Entry Fields

e1 = Entry(hosp\_d, textvariable=StringVar)

e2 = Entry(hosp\_d, textvariable=StringVar)

e3 = Entry(hosp\_d, textvariable=StringVar)

e1.grid(row=0, column=1)

e2.grid(row=1, column=1)

e3.grid(row=2, column=1)

mycur.execute("Select \* from hosp\_details")

for x in mycur:

e1.insert(0, x[0])

e2.insert(0, x[1])

e3.insert(0, x[2])

mainloop()

**def Search\_Records\_Doctors():**

srch = Tk()

srch.title("Search Records")

**def getdata():**

for item in treev.get\_children():

treev.delete(item)

dname\_list = []

dage\_list = []

dd\_list = []

dp\_list = []

mycur.execute("Select \* from doctor\_details where d\_name='{}'".format(e1.get()))

for x in mycur:

dname\_list.append(x[0])

dage\_list.append(x[1])

dd\_list.append(x[2])

dp\_list.append(x[3])

for i in range(0, len(dname\_list)):

treev.insert("", 'end', text="L1",values=(dname\_list[i], dage\_list[i], dd\_list[i], dp\_list[i]))

**def showall():**

dname\_list = []

dage\_list = []

dd\_list = []

dp\_list = []

mycur.execute("Select \* from doctor\_details".format(e1.get()))

for x in mycur:

dname\_list.append(x[0])

dage\_list.append(x[1])

dd\_list.append(x[2])

dp\_list.append(x[3])

for i in range(0, len(dname\_list)):

treev.insert("", 'end', text="L1", values=(dname\_list[i], dage\_list[i], dd\_list[i], dp\_list[i]))

**def modify():**

srchm = Tk()

srchm.title("Modify Data")

where = treev.item(treev.selection()[0])['values'][0]

**def tdata():**

s = treev.selection()[0]

me1.insert(0, treev.item(s)['values'][0])

me2.insert(0, treev.item(s)['values'][1])

me3.insert(0, treev.item(s)['values'][2])

me4.insert(0, treev.item(s)['values'][3])

**def modifym():**

try:

mycur.execute("update doctor\_details set d\_name='{}', d\_age={}, d\_department='{}', d\_phono='{}' where d\_name = '{}'".format(me1.get(), me2.get(), me3.get(), me4.get(), where))

mydb.commit()

messagebox.showinfo("Alert", "Record Succesfully Updated")

for item in treev.get\_children():

treev.delete(item)

showall()

except:

messagebox.showerror("Error", "There is Some Error please recheck")

# Label Fields

ml1 = Label(srchm, text="Doctor Name: ", font="Arial").grid(row=0, column=0)

ml2 = Label(srchm, text="Doctor Age : ", font="Arial").grid(row=1, column=0)

ml3 = Label(srchm, text="Department : ", font="Arial").grid(row=2, column=0)

ml4 = Label(srchm, text="Contact No : ", font="Arial").grid(row=3, column=0)

# Entry Fields

me1 = Entry(srchm, textvariable=StringVar)

me2 = Entry(srchm, textvariable=IntVar)

me3 = Entry(srchm, textvariable=StringVar)

me4 = Entry(srchm, textvariable=StringVar)

me1.grid(row=0, column=1)

me2.grid(row=1, column=1)

me3.grid(row=2, column=1)

me4.grid(row=3, column=1)

tdata()

bt5 = Button(srchm, text="Modify", command=modifym)

bt5.grid(row=4, column=0)

mainloop()

**def delete():**

name = treev.item(treev.selection()[0])['values'][0]

answer = messagebox.askyesno(title="Confirmation", message="Are you sure you want to delete selected record")

if answer:

try:

mycur.execute("delete from doctor\_details where d\_name='{}'".format(name))

mydb.commit()

messagebox.showinfo("Alert", "Selected Record Deleted")

for item in treev.get\_children():

treev.delete(item)

showall()

except:

messagebox.showerror("Error", "Sorry we cant delete please try again")

e1 = Entry(srch, textvariable=StringVar)

"""TreeView Widget"""

treev = Treeview(srch, selectmode='browse')

treev.grid(row=1, column=0, columnspan=3)

verscrlbar = Scrollbar(srch,

orient="vertical",

command=treev.yview)

verscrlbar.grid(row=1, column=4)

treev.configure(xscrollcommand=verscrlbar.set)

treev["columns"] = ("1", "2", "3", "4")

treev['show'] = 'headings'

treev.column("1", width=90, anchor='c')

treev.column("2", width=90, anchor='se')

treev.column("3", width=90, anchor='se')

treev.column("4", width=90, anchor='se')

treev.heading("1", text="Name")

treev.heading("2", text="Age")

treev.heading("3", text="Department")

treev.heading("4", text="Contact")

lb1 = Label(srch, text="Doctor Name", font="Arial")

lb1.grid(row=0, column=0)

e1.grid(row=0, column=1)

#button

bt1 = Button(srch, text="Search", command=getdata)

bt2 = Button(srch, text="Show All", command=showall)

bt3 = Button(srch, text="Modify Data", command=modify)

bt4 = Button(srch, text="Delete Data", command=delete)

bt1.grid(row=0, column=2)

bt2.grid(row=2, column=0)

bt3.grid(row=2, column=1)

bt4.grid(row=2, column=2)

mainloop()

**def Search\_Records\_Patient():**

srch = Tk()

srch.title("Search Records")

**def getdata():**

for item in treev.get\_children():

treev.delete(item)

pid\_list = []

pname\_list = []

page\_list = []

pproblems\_list = []

pcontact=[]

pday = []

proomt = []

proomn = []

ptbill = []

mycur.execute("Select \* from patient\_details where p\_id='{}'".format(e1.get()))

for x in mycur:

pid\_list.append(x[0])

pname\_list.append(x[1])

page\_list.append(x[2])

pproblems\_list.append(x[3])

pcontact.append(x[4])

pday.append(x[5])

proomt.append(x[6])

proomn.append(x[7])

ptbill.append(x[8])

for i in range(0, len(pid\_list)):

treev.insert("", 'end', text="L1", values=(pid\_list[i], pname\_list[i], page\_list[i], pproblems\_list[i], pcontact[i], pday[i], proomt[i], proomn[i], ptbill[i]))

**def showall():**

for item in treev.get\_children():

treev.delete(item)

pid\_list = []

pname\_list = []

page\_list = []

pproblems\_list = []

pcontact = []

pday = []

proomt = []

proomn = []

ptbill = []

mycur.execute("Select \* from patient\_details".format(e1.get()))

for x in mycur:

pid\_list.append(x[0])

pname\_list.append(x[1])

page\_list.append(x[2])

pproblems\_list.append(x[3])

pcontact.append(x[4])

pday.append(x[5])

proomt.append(x[6])

proomn.append(x[7])

ptbill.append(x[8])

for i in range(0, len(pid\_list)):

treev.insert("", 'end', text="L1", values=(

pid\_list[i], pname\_list[i], page\_list[i], pproblems\_list[i], pcontact[i], pday[i], proomt[i], proomn[i],

ptbill[i]))

**def modify():**

srchm = Tk()

srchm.title("Modify Data")

where = treev.item(treev.selection()[0])['values'][0]

**def tdata():**

s = treev.selection()[0]

me1.insert(0, treev.item(s)['values'][0])

me2.insert(0, treev.item(s)['values'][1])

me3.insert(0, treev.item(s)['values'][2])

me4.insert(0, treev.item(s)['values'][3])

me5.insert(0, treev.item(s)['values'][4])

me6.insert(0, treev.item(s)['values'][5])

me7.insert(0, treev.item(s)['values'][6])

me8.insert(0, treev.item(s)['values'][7])

**def modifym():**

try:

mycur.execute(

"update patient\_details set p\_id='{}', p\_name='{}', p\_age={}, p\_problems='{}', p\_phono='{}', days={}, room='{}', room\_no={}, Total\_Bill={} where p\_id = '{}'".format(

me1.get(), me2.get(), me3.get(), me4.get(), me5.get(), me6.get(), me7.get(), me8.get(), me9.get(), where))

mydb.commit()

messagebox.showinfo("Alert", "Record Succesfully Updated")

for item in treev.get\_children():

treev.delete(item)

showall()

except:

messagebox.showerror("Error", "There is Some Error please recheck")

# Label Fields

ml1 = Label(srchm, text="ID : ", font="Arial").grid(row=0, column=0)

ml2 = Label(srchm, text="Name : ", font="Arial").grid(row=1, column=0)

ml3 = Label(srchm, text="Age : ", font="Arial").grid(row=2, column=0)

ml4 = Label(srchm, text="Problems : ", font="Arial").grid(row=3, column=0)

ml5 = Label(srchm, text="Contact : ", font="Arial").grid(row=4, column=0)

ml6 = Label(srchm, text="Days : ", font="Arial").grid(row=5, column=0)

ml7 = Label(srchm, text="RoomType : ", font="Arial").grid(row=6, column=0)

ml8 = Label(srchm, text="RoomNo : ", font="Arial").grid(row=7, column=0)

ml9 = Label(srchm, text="Total Bill : ", font="Arial").grid(row=8, column=0)

# Entry Fields

me1 = Entry(srchm, textvariable=StringVar)

me2 = Entry(srchm, textvariable=StringVar)

me3 = Entry(srchm, textvariable=IntVar)

me4 = Entry(srchm, textvariable=StringVar)

me5 = Entry(srchm, textvariable=StringVar)

me6 = Entry(srchm, textvariable=IntVar)

me7 = Entry(srchm, textvariable=StringVar)

me8 = Entry(srchm, textvariable=IntVar)

me9 = Entry(srchm, textvariable=IntVar)

me1.grid(row=0, column=1)

me2.grid(row=1, column=1)

me3.grid(row=2, column=1)

me4.grid(row=3, column=1)

me5.grid(row=4, column=1)

me6.grid(row=5, column=1)

me7.grid(row=6, column=1)

me8.grid(row=7, column=1)

me9.grid(row=8, column=1)

tdata()

bt5 = Button(srchm, text="Modify", command=modifym)

bt5.grid(row=9, column=0)

mainloop()

**def delete():**

name = treev.item(treev.selection()[0])['values'][0]

answer = messagebox.askyesno(title="Confirmation", message="Are you sure you want to delete selected record")

if answer:

try:

mycur.execute("delete from patient\_details where p\_id='{}'".format(name))

mydb.commit()

messagebox.showinfo("Alert", "Selected Record Deleted")

for item in treev.get\_children():

treev.delete(item)

showall()

except:

messagebox.showerror("Error", "Sorry we cant delete please try again")

e1 = Entry(srch, textvariable=StringVar)

"""TreeView Widget"""

treev = Treeview(srch, selectmode='browse')

treev.grid(row=1, column=0, columnspan=3)

verscrlbar = Scrollbar(srch,

orient="vertical",

command=treev.yview)

verscrlbar.grid(row=1, column=4)

treev.configure(xscrollcommand=verscrlbar.set)

treev["columns"] = ("1", "2", "3", "4", "5", "6", "7", "8", "9")

treev['show'] = 'headings'

treev.column("1", width=90, anchor='c')

treev.column("2", width=90, anchor='se')

treev.column("3", width=90, anchor='se')

treev.column("4", width=90, anchor='se')

treev.column("5", width=90, anchor='se')

treev.column("6", width=90, anchor='se')

treev.column("7", width=90, anchor='se')

treev.column("8", width=90, anchor='se')

treev.column("9", width=90, anchor='se')

treev.heading("1", text="ID")

treev.heading("2", text="Name")

treev.heading("3", text="Age")

treev.heading("4", text="Problems")

treev.heading("5", text="Contact")

treev.heading("6", text="Days")

treev.heading("7", text="Room Type")

treev.heading("8", text="Room No.")

treev.heading("9", text="Total Bill")

lb1 = Label(srch, text="Patient ID", font="Arial")

lb1.grid(row=0, column=0)

e1.grid(row=0, column=1)

# button

bt1 = Button(srch, text="Search", command=getdata)

bt2 = Button(srch, text="Show All", command=showall)

bt3 = Button(srch, text="Modify Data", command=modify)

bt4 = Button(srch, text="Delete Data", command=delete)

bt1.grid(row=0, column=2)

bt2.grid(row=2, column=0)

bt3.grid(row=2, column=1)

bt4.grid(row=2, column=2)

mainloop()

**def Deleteall():**

answer = messagebox.askyesno(title="Confirmation", message="Are you sure you want to delete all record?")

if answer:

try:

mycur.execute("truncate table hosp\_details")

mycur.execute("truncate table doctor\_details")

mycur.execute("truncate table patient\_details")

mycur.execute("truncate table worker\_details")

mydb.commit()

messagebox.showinfo("Alert", "All Table Record Deleted")

except:

messagebox.showerror("Error", "Sorry we cant delete please try again")

**def updatebeddata():**

for item in treev.get\_children():

treev.delete(item)

treev.insert("", 'end', text="L1", values=(g\_beds, i\_beds, s\_beds))

**def time():**

string = strftime('%H:%M:%S %p')

clk.config(text=string)

clk.after(1000, time)

**def beddata():**

global g\_beds

global s\_beds

global i\_beds

global g\_list

global s\_list

global i\_List

try:

mycur.execute("Select \* from hosp\_details")

g\_beds = 0

s\_beds=0

i\_beds=0

for x in mycur:

g\_beds=x[0]

s\_beds=x[1]

i\_beds=x[2]

g\_list = []

s\_list = []

i\_list = []

for i in range(1, g\_beds+1):

g\_list.append(100+i)

for i in range(1, s\_beds+1):

s\_list.append(200+i)

for i in range(1, i\_beds+1):

i\_list.append(300+i)

except:

g\_beds = 0

s\_beds=0

i\_beds=0

g\_list = []

s\_list = []

i\_list = []

**#---------------Menu -------------**

# importing only those functions

# which are needed for

# creating tkinter window

root = Tk()

root.title('Health Hospital')

root.geometry("510x400")

root.configure(background='#ec8833')

**# Creating Menubar**

menubar = Menu(root)

table = Menu(menubar, tearoff = 0)

table.add\_command(label ='Create Tables', command = create\_table)

sub\_menu = Menu(table, tearoff=0)

sub\_menu.add\_command(label='Hospital Records', command =Insert\_Hosp )

sub\_menu.add\_command(label='Doctors Records',command =Insert\_Doctors)

sub\_menu.add\_command(label='Workers Records',command =Insert\_Workers)

sub\_sub\_menu = Menu(sub\_menu, tearoff=0)

sub\_menu.add\_cascade(label="Patient Records", menu=sub\_sub\_menu)

sub\_sub\_menu.add\_command(label='Admit',command =Insert\_Patient)

sub\_sub\_menu.add\_command(label='OPD',command =Insert\_Patient\_OPD)

table.add\_cascade(label="Insert",menu=sub\_menu)

table.add\_separator()

table.add\_command(label ='Exit', command = root.destroy)

menubar.add\_cascade(label="Create", menu=table)

**#Search**

table = Menu(menubar, tearoff = 0) # create the file\_menu

sub\_menu1 = Menu(table, tearoff=0)

sub\_menu1.add\_command(label="Doctor", command=Search\_Records\_Doctors)

sub\_menu1.add\_command(label="Patient", command=Search\_Records\_Patient)

table.add\_cascade(label ='Search Data', menu=sub\_menu1)

table.add\_command(label ='Delete All', command = Deleteall)

menubar.add\_cascade(label="Data",menu=table)

table = Menu(menubar, tearoff = 0)

table.add\_command(label='Exit', command =root.destroy)

menubar.add\_cascade(label="Exit",menu=table)

**# MainWindow**

clk = Label(root, font=('calibri', 15, 'bold'),background='#449bfa',foreground='white')

tt = Label(root,text="Welcome to Hospital Management System", font=('calibri', 15, 'bold'),background='#ec8833',foreground='#f1f94f')

**#Treeview Beds Data**

"""TreeView Widget"""

treev = Treeview(root, selectmode='browse')

treev.place(x=0, y=50)

verscrlbar = Scrollbar(root,orient="vertical",command=treev.yview)

verscrlbar.place(x=460, y=150)

treev.configure(xscrollcommand=verscrlbar.set)

treev["columns"] = ("1", "2", "3", "4")

treev['show'] = 'headings'

treev.column("1", width=90, anchor='c')

treev.column("2", width=90, anchor='se')

treev.column("3", width=90, anchor='se')

treev.heading("1", text="General Beds")

treev.heading("2", text="ICU Beds")

treev.heading("3", text="Special Beds")

**# Beds Data**

try:

mycur.execute("Select \* from hosp\_details")

g\_beds = 0

s\_beds = 0

i\_beds = 0

for x in mycur:

g\_beds = x[0]

s\_beds = x[1]

i\_beds = x[2]

g\_list = []

s\_list = []

i\_list = []

for i in range(1, g\_beds + 1):

g\_list.append(100 + i)

for i in range(1, s\_beds + 1):

s\_list.append(200 + i)

for i in range(1, i\_beds + 1):

i\_list.append(300 + i)

updatebeddata()

except:

g\_beds = 0

s\_beds = 0

i\_beds = 0

g\_list = []

s\_list = []

i\_list = []

updatebeddata()

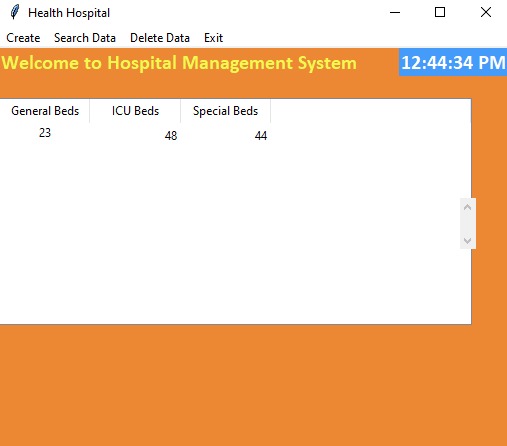
tt.place(x=0, y=0)

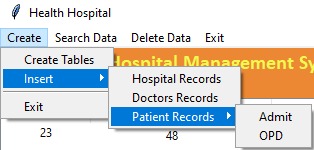
clk.place(x=400, y=0)

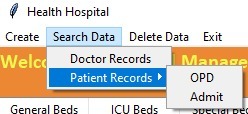
time()

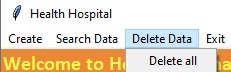
root.config(menu = menubar)

mainloop()

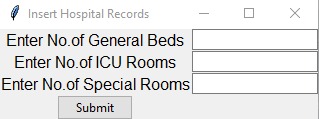
**OUTPUT**

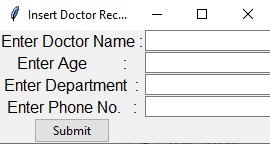


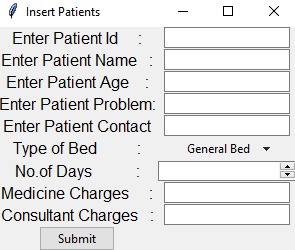


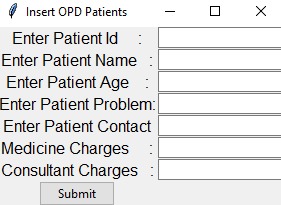






****

****

****

**HARDWARE REQUIREMENTS**

1. Operating System : Windows 10 and above
2. Processor : Intel i3 or AMD Ryzen 3 processors or better
3. RAM : 4GB and above
4. Hard Disk : 100 GB and above

**SOFTWARE REQUIREMENTS**

1. Windows 10 and above
2. Python 3.0 and above
3. MySQL 5.0 and above
4. CSV module
5. Tkinter module
6. Time module

**BIBLIOGRAPHY**

1. Computer Science with Python: Class XI and XII (Sumita Arora)
2. <https://www.geeksforgeeks.org/python-gui-tkinter/>
3. <https://www.geeksforgeeks.org/python-tkinter-treeview-scrollbar/>
4. <https://www.w3schools.com>