

100000010001100101010111

0

10100001111100101011010

010110001000000 01011001

011 Computer Science(083) 010

001 PROJECT FILE 1110

000 Hospital Management System 1010

SESSION: 2020 – 21

101 0101000011100001

0100010101000011100001

11000101000011

010101000001100

1010001001001010

1010010101001010

0100101001001010

1000000101111100100101101

Submitted by:

Name: Arihant Debnath

Class: XII – C

Board Roll no: 21668593

School: DPS Indirapuram

# CERTIFICATE



This is to certify that ARIHANT DEBNATH of class XII-C has prepared the project on "Hospital management system". The project is result of his efforts and endeavors. This project is found worthy of acceptance as the final project report for the subject computer science of class XII. He has prepared this project under my guidance.

Ms. Rinkoo Gupta  
(Computer Science Teacher)  
(DPS Indirapuram)

## **ACKNOWLEDGEMENT**

I would like to express a deep sense of gratitude towards my computer science teacher **Ms. Rinkoo Gupta** ma'am for guiding me through the course of my project. She always evinced keen interest in my work and her constructive advice and constant motivation have been responsible for the successful completion of this project.

My sincere thanks goes to **Ms. Sangeeta Hajela**, our school principal for her coordination in extending every possible support possible in the success of this project.

I would like to thank all those who have helped directly or indirectly in the completion of the project.

**Arihant Debnath**

**XII -C**

Computer Science(083) PROJECT FILE

SNO.	<u>TOPIC</u>	SGN.
1.	INTRODUCTION TO THE PROJECT	
2.	DATABASE/Backend  Mysql tables used and their structure	
3.	HARDWARE AND SOFTWARE  REQUIREMENTS	
4	CODE	
5.	OUTPUTS	
6	CONCLUSION	
7	FUTURE ENHANCEMENTS	
	BIBLIOGRAPHY	

(I.)

# INTRODUCTION TO THE PROJECT

**Hospital management system** is a computer system that helps manage the information related to health care and aids in the job completion of health care providers effectively.

HMS came into the picture of hospital management as early as 1960 and have ever since been evolving and synchronizing with the technologies while modernizing healthcare facilities. In today's world, the management of healthcare starts from the hands of the patients through their mobile phones and facilitates the needs of the patient.

- Maintain the medical records of the patient
- Maintain the contact details of the patient
- Keep track of the appointment dates

The advantages of HMS can be pinpointed to the following:

- Time-saving Technology
- Improved Efficiency by avoiding human errors
- Reduces scope for Error
- Data security and correct data retrieval made possible
- Cost effective and easily manageable
- Easy access to patient data with correct patient history
- Improved patient care made possible
- Easy monitoring of supplies in inventory
- Reduces the work of documentation
- Better Audit controls and policy compliance.

# Hospital Management System in Python

The hospital management system in python which we are going to build will look something like this:



## Project Prerequisites

**tkinter** – Please run below command to install tkinter  
 pip install tkinter

**pillow** – Please run below command to install tkinter  
 pip install pillow

**pymysql** – Please run below command to install tkinter  
 pip install pymysql

**Note:** You are required to have MySQL server installed on your system in order to make pymysql work. If you do not have it ready, please download from [MySQL Official website](#)

## Description of Project Files

Below are the project files you will get once you download and extract the HMS project:

- **patient.py**- which does function call to all other python files
- Hospital images – Hospital images for the UI

## Description of Tables

*Create Tables and the Database*

```
c:\wamp64\bin\mysql\mysql5.7.31\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 85
Server version: 5.7.31 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| college |
| hospital |
| mysql |
| performance_schema |
| store |
| sys |
| test |
| try7 |
| university |
+-----+
10 rows in set (0.00 sec)

mysql> use hospital;
Database changed
mysql> show tables;
+-----+
| Tables_in_hospital |
+-----+
| patient |
+-----+
1 row in set (0.00 sec)
```

# Computer Science(083) PROJECT FILE

```
c:\wamp64\bin\mysql\mysql5.7.31\bin\mysql.exe
mysql> use hospital;
Database changed
mysql> show tables;
+-----+
| Tables_in_hospital |
+-----+
| patient |
+-----+
1 row in set (0.00 sec)

mysql> describe patient;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Dep | varchar(45) | YES | | NULL |
| Past | varchar(45) | YES | | NULL |
| Room | varchar(45) | YES | | NULL |
| Floor | varchar(45) | YES | | NULL |
| patient_id | int(11) | YES | | NULL |
| Name | varchar(45) | YES | | NULL |
| Division | varchar(45) | YES | | NULL |
| Roll | varchar(45) | YES | | NULL |
| Gender | varchar(45) | YES | | NULL |
| Dob | varchar(45) | YES | | NULL |
| Email | varchar(45) | YES | | NULL |
| Phone | varchar(45) | YES | | NULL |
| Address | varchar(45) | YES | | NULL |
| Doctor | varchar(45) | YES | | NULL |
+-----+-----+-----+-----+-----+-----+
14 rows in set (0.04 sec)

mysql>
```

Activate Window  
Go to Settings to act

## Working Description

This program consists of 6 options:

- Add record of the patients
- Update records
- Reset the Reacords
- Delete the records
- Display all the records
- Exit

This is CRUD(Create, Read, Update, Delete) application.

*CRUD* is an acronym that comes from the world of computer programming and refers to the four functions that are considered necessary to implement a persistent storage application: *create*, *read*, *update* and *delete*. The *CRUD* acronym identifies all of the major functions that are inherent to *relational databases* and the applications used to manage them, which include Oracle Database, Microsoft SQL Server, MySQL, and others.



## Computer Science(083) PROJECT FILE

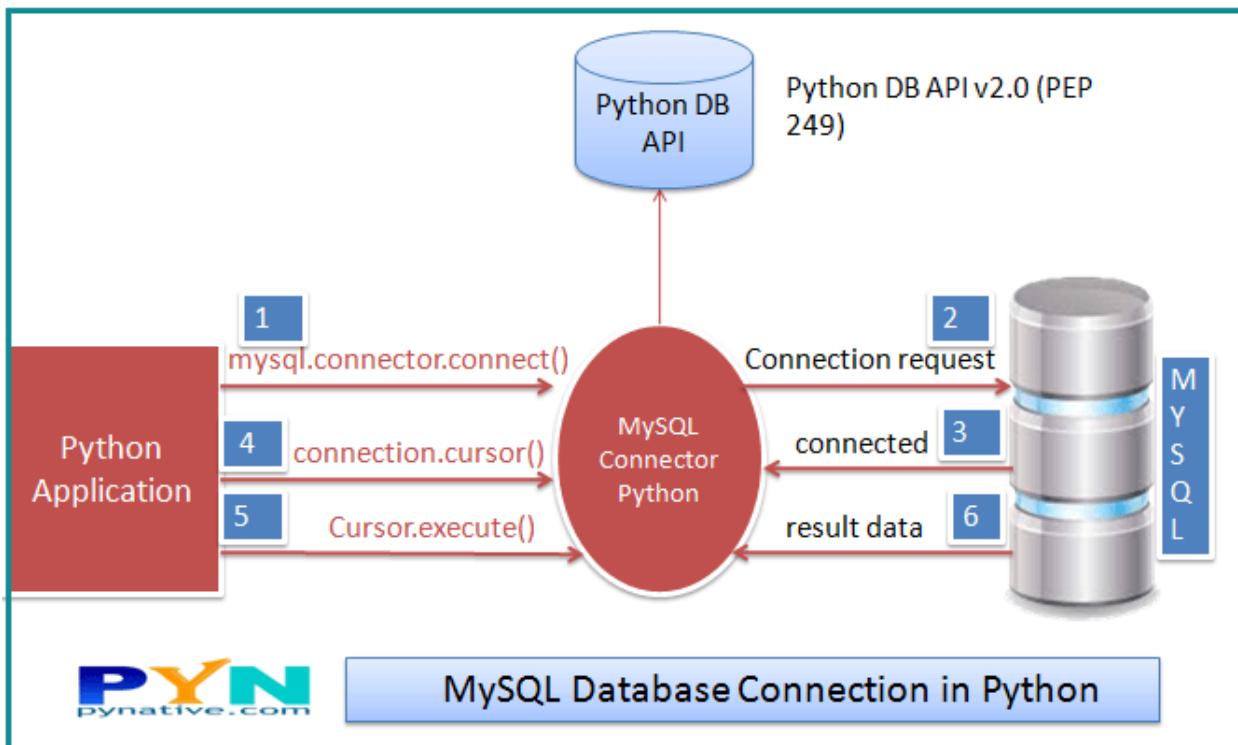
Here I am using the Mysql databases for the data management.

### (2.) Mysql tables used and their structure

## DATABASE

Here I am using the MySQL relation databases for the Data management of the Patients.

MySQL is an open-source relational database management system that works on many platforms. It provides multi-user access to support many storage engines and is backed by Oracle.



I am using MySQL with Python, for this project.

Python = Front End

MySQL = BackEnd

## (3.) HARDWARE AND SOFTWARE REQUIREMENTS

### (A) SOFTWARE

2. Python 3
3. Tkinter module
4. Pillow module
5. MySQL database

### (B) HARDWARE

**Processor (CPU):** Intel Core i3 (sixth generation or newer) or equivalent

**Operating System:** Microsoft Windows 10 x64

(free via Azure Dev Tools for Teaching. Restrictions apply.)

**Memory:** 8 GB RAM

**Storage:** 500 GB internal storage drive

**Monitor/Display:** 15" LCD monitor

**Other:** 802.11ac 2.4/5 GHz wireless adapter

## (4.) CODING

```
from tkinter import *
from tkinter import ttk
from PIL import Image, ImageTk
import mysql.connector
from tkinter import messagebox
#pip install pillow
# Setting up the database and tables
mycon = mysql.connector.connect(
    host = 'localhost',
    user = 'root'
)
cur = mycon.cursor()
cur.execute("Show databases;")
flag = False
for dbname in cur:
    if ("HOSPITAL" in dbname):
        flag = True
        break

if not flag:
    cur.execute("Create database if not exists HOSPITAL")
    print("database HOSPITAL created")
else:
    print("database HOSPITAL already exists")

con = mycon.connect(
    host = "localhost",
    user = "root",
    database = "HOSPITAL"
)

if mycon.is_connected():
    print("database opened")
else:
    print("error opening database")

# inserting tables
t_query1 = """CREATE table if not exists patient(
    Dep varchar(45),
    Past varchar(45),
    Room varchar(45),
    Floor varchar(45),
    patient_id INT,
    Name varchar(45),
    Division varchar(45),
    Roll varchar(45),
    Gender varchar(45),
    Dob varchar(45),
    Email varchar(45),
    Phone varchar(45),
    Address varchar(45),
    Doctor varchar(45))"""
cur = mycon.cursor()
cur.execute(t_query1)
class Patient:
    def __init__(self,root):
        self.root = root
        self.root.geometry("1366x695+0+0")
        self.root.title("HOSPITAL MANAGEMENT SYSTEM")
        # Program Variables
        self.var_dep = StringVar()
        self.var_past = StringVar()
        self.var_room = StringVar()
        self.var_floor = StringVar()
        self.var_patient_id = StringVar()
        self.var_patient_name = StringVar()
        self.var_div = StringVar()
        self.var_roll = StringVar()
        self.var_gender = StringVar()
        self.var_dob = StringVar()
        self.var_email = StringVar()
```

# Computer Science(083) PROJECT FILE

```
self.var_phone = StringVar()
self.var_address = StringVar()
self.var_doctor = StringVar()

#Fetching the images
# 1st image
img = Image.open(r"C:\Users\thicc\Desktop\Python\Work\Learn\Hospital Mangement\Hospital images\1st.jpg")
img = img.resize((480,120), Image.ANTIALIAS)
self.photoimg = ImageTk.PhotoImage(img)

# to diaplay the images
self.btn_1 = Button(self.root, image = self.photoimg, cursor = "hand2")
self.btn_1.place(x = 0, y = 0, width = 480, height = 120)

# 2nd image
img_2 = Image.open(r"C:\Users\thicc\Desktop\Python\Work\Learn\Hospital Mangement\Hospital images\6th.jpg")
img_2 = img_2.resize((480,120), Image.ANTIALIAS)
self.photoimg_2 = ImageTk.PhotoImage(img_2)

# to diaplay the images
self.btn_2 = Button(self.root, image = self.photoimg_2, cursor = "hand2")
self.btn_2.place(x = 480, y = 0, width = 480, height = 120)

# 3rd image
img_3 = Image.open(r"C:\Users\thicc\Desktop\Python\Work\Learn\Hospital Mangement\Hospital images\5th.jpg")
img_3 = img_3.resize((480,120), Image.ANTIALIAS)
self.photoimg_3 = ImageTk.PhotoImage(img_3)

# to diaplay the images
self.btn_3 = Button(self.root, image = self.photoimg_3, cursor = "hand2")
self.btn_3.place(x = 960, y = 0, width = 480, height = 120)

#background Image for the program
img_4 = Image.open(r"C:\Users\thicc\Desktop\Python\Work\Learn\Hospital Mangement\Hospital images\hospital.jpg")
img_4 = img_4.resize((1366,695), Image.ANTIALIAS)
self.photoimg_4 = ImageTk.PhotoImage(img_4)

bg_lbl = Label(self.root, image = self.photoimg_4, bd =2, relief = RIDGE)
bg_lbl.place(x = 0, y = 100, width = 1366, height = 600)

lbl_title = Label(bg_lbl, text = "HOSPITAL MANAGEMENT SYSTEM", font = ('times new roman', 27, 'bold'), fg = "Black", bg = "white")
lbl_title.place(x = 0, y = 0, width = 1366, height = 40)

# Title Label
Management_frame = Frame(bg_lbl, bd =2, relief = RIDGE, bg = 'white')
Management_frame.place(x = 15, y = 45, width = 1330, height = 540)

# Left frame
DataLeftFrame = LabelFrame(Management_frame, bd = 4, relief = RIDGE, padx=2, text="Patient Information", font=('times new roman', 12, 'bold'), fg="red", bg="white")
DataLeftFrame.place(x = 10, y = 10, width = 650, height = 540)

# Hospital Department information/ Rules of the Hospital
img_5 = Image.open(r"C:\Users\thicc\Desktop\Python\Work\Learn\Hospital Mangement\Hospital images\rules.jpg")
img_5 = img_5.resize((650,120), Image.ANTIALIAS)
self.photoimg_5 = ImageTk.PhotoImage(img_5)

my_img = Label(DataLeftFrame, image = self.photoimg_5, bd =2, relief = RIDGE)
my_img.place(x = 0, y = 0, width = 650, height = 120)

# Hospital Department information
# Current past LabelFrame information
std_lbl_info_frame = LabelFrame(DataLeftFrame, bd=4, relief=RIDGE, padx=2, text="Hospital Department information", font=('times new roman', 12, 'bold'), fg="red", bg="white")
std_lbl_info_frame.place(x = 0, y = 120, width = 650, height = 110)

# Hospital Department
lbl_dept = Label(std_lbl_info_frame, text = "Department : ", font = ('arial', 10, 'bold'), bg = "white")
lbl_dept.grid(row = 0 , column = 0, padx = 2, sticky = W)

combo_dep = ttk.Combobox(std_lbl_info_frame, textvariable = self.var_dep, font = ('arial', 10, 'bold'), width = 17, state = "readonly")
combo_dep["value"] = ("Select Department", "Physician", "Cardiovascular", "Medicine", "Pediatrician", "Neurosurgery", "Respiratory")
combo_dep.current(0)
combo_dep.grid(row = 0, column = 1, padx = 2, pady = 5, sticky = W)

# Prior Health issues
past_std = Label(std_lbl_info_frame, font = ('arial', 10, 'bold'), text = " Prior Health issues : ", bg = "white")
past_std.grid(row = 0, column = 2, sticky = W, padx = 2, pady = 10)
```

# Computer Science(083) PROJECT FILE

```
com_txtpast_std = ttk.Combobox(std_lbl_info_frame, textvariable = self.var_past, font = ('arial', 10, 'bold'), width = 17, state = "readonly")
com_txtpast_std["value"] = ("Select Yes/ No", "YES", "NO", "Don't know")
com_txtpast_std.current(0)
com_txtpast_std.grid(row = 0, column = 3, sticky = W, padx = 2, pady =10)

# Room Type
current_room = Label(std_lbl_info_frame, font = ('arial', 10, 'bold'),text = "Room Type :", bg = "white")
current_room.grid(row = 1, column = 0, sticky = W, padx = 2, pady = 10)

com_txt_current_room = ttk.Combobox(std_lbl_info_frame, textvariable = self.var_room, font = ('arial', 10, 'bold'), width = 17, state = "readonly")
com_txt_current_room["value"] = ("Select Room Type", "Normal", "Medium", "High", "VIP")
com_txt_current_room.current(0)
com_txt_current_room.grid(row = 1, column = 1, sticky = W, padx = 2)

# Hospital Floor of operation
label_Floor = Label(std_lbl_info_frame, font = ('arial', 10, 'bold'),text = "Hospital Floor :", bg = "white")
label_Floor.grid(row = 1, column = 2, sticky = W, padx = 2, pady = 10)
comFloor = ttk.Combobox(std_lbl_info_frame, textvariable = self.var_floor, font = ('arial', 10, 'bold'), width = 17, state = "readonly")
comFloor["value"] = ("Select Floor", "Floor-1", "Floor-2", "Floor-3", "Floor-4")
comFloor.current(0)
comFloor.grid(row = 1, column = 3, sticky = W, padx = 2, pady =10)

# Patient Class Frame information
std_lbl_class_frame = LabelFrame(DataLeftFrame, bd = 4, relief = RIDGE, padx = 2, text = " Patient Frame information", font = ('times new roman', 12, 'bold'), fg = "red", bg = "white")
std_lbl_class_frame.place(x = 0, y = 235, width = 650, height = 200)

# Lables entry
# Patient ID
lbl_id = Label( std_lbl_class_frame, font = ('arial', 10, 'bold'),text = "PatientID :", bg = "white")
lbl_id.grid(row = 0, column = 0, sticky = W, padx = 2, pady = 7)

id_entry = ttk.Entry(std_lbl_class_frame, textvariable = self.var_patient_id, font = ('arial', 10, 'bold'), width = 22)
id_entry.grid(row = 0, column = 1, sticky = W, padx = 2, pady = 7)

# Patient Name
lbl_Name = Label( std_lbl_class_frame, font = ('arial', 10, 'bold'),text = "Patient Name :", bg = "white")
lbl_Name.grid(row = 0, column = 2, sticky = W, padx = 2, pady = 7)

txt_name = ttk.Entry(std_lbl_class_frame, textvariable = self.var_patient_name, font = ('arial', 10, 'bold'), width = 22)
txt_name.grid(row = 0, column = 3, sticky = W, padx = 2, pady = 7)

# Hospital Division
lbl_div = Label( std_lbl_class_frame, font = ('arial', 10, 'bold'),text = "Class Division :", bg = "white")
lbl_div.grid(row = 1, column = 0, sticky = W, padx = 2, pady = 7)

com_txt_div = ttk.Combobox(std_lbl_class_frame, textvariable = self.var_div, font = ('arial', 10, 'bold'), width = 18, state = "readonly")
com_txt_div['value'] = ("Select Division", "A", "B" , "C")
com_txt_div.current(0)
com_txt_div.grid(row = 1, column = 1, sticky = W, padx = 2, pady =7)

# Patient No/ Roll no.
lbl_roll = Label( std_lbl_class_frame, font = ('arial', 10, 'bold'),text = "Patient No : ", bg = "white")
lbl_roll.grid(row = 1, column = 2, sticky = W, padx = 2, pady = 7)

txt_roll = ttk.Entry(std_lbl_class_frame, textvariable = self.var_roll, font = ('arial', 10, 'bold'), width = 22)
txt_roll.grid(row = 1, column = 3, sticky = W, padx = 2, pady = 7)

# Gender of the Patient
lbl_gender = Label( std_lbl_class_frame, font = ('arial', 10, 'bold'),text = "Gender :", bg = "white")
lbl_gender.grid(row = 2, column = 0, sticky = W, padx = 2, pady = 7)

com_txt_gender = ttk.Combobox(std_lbl_class_frame, textvariable = self.var_gender, font = ('arial', 10, 'bold'), width = 18, state = "readonly")
com_txt_gender['value'] = ("Select gender", "Male", "Female" , "Other")
com_txt_gender.current(0)
com_txt_gender.grid(row = 2, column = 1, sticky = W, padx = 2, pady =7)

# DOB
lbl_dob = Label(std_lbl_class_frame, font = ('arial', 10, 'bold'),text = "DOB :", bg = "white")
lbl_dob.grid(row = 2, column = 2, sticky = W, padx = 2, pady = 7)

txt_dob = ttk.Entry(std_lbl_class_frame, textvariable = self.var_dob, font = ('arial', 10, 'bold'), width = 22)
txt_dob.grid(row = 2, column = 3, sticky = W, padx = 2, pady = 7)
```

# Computer Science(083) PROJECT FILE

```
# Email
lbl_email = Label(std_lbl_class_frame, font = ('arial', 10, 'bold'),text = "Email : ", bg = "white")
lbl_email.grid(row = 3, column = 0, sticky = W, padx = 2, pady = 7)

txt_email = ttk.Entry(std_lbl_class_frame, textvariable = self.var_email, font = ('arial', 10, 'bold'), width = 22)
txt_email.grid(row = 3, column = 1, padx = 2, pady = 7)

# Phone
lbl_phone = Label(std_lbl_class_frame, font = ('arial', 10, 'bold'),text = "Phone No : ", bg = "white")
lbl_phone.grid(row = 3, column = 2, sticky = W, padx = 2, pady = 7)

txt_phone = ttk.Entry(std_lbl_class_frame, textvariable = self.var_phone, font = ('arial', 10, 'bold'), width = 22)
txt_phone.grid(row = 3, column = 3, padx = 2, pady = 7)

# Address
lbl_address = Label(std_lbl_class_frame, font = ('arial', 10, 'bold'),text = "Address : ", bg = "white")
lbl_address.grid(row = 4, column = 0, sticky = W, padx = 2, pady = 7)

txt_address = ttk.Entry(std_lbl_class_frame, textvariable = self.var_address, font = ('arial', 10, 'bold'), width = 22)
txt_address.grid(row = 4, column = 1, padx = 2, pady = 7)

# Doctor
lbl_doctor = Label(std_lbl_class_frame, font = ('arial', 10, 'bold'),text = "Doctor : ", bg = "white")
lbl_doctor.grid(row = 4, column = 2, sticky = W, padx = 2, pady = 7)

txt_doctor = ttk.Entry(std_lbl_class_frame, textvariable = self.var_doctor, font = ('arial', 10, 'bold'), width = 22)
txt_doctor.grid(row = 4, column = 3, padx = 2, pady = 7)

# Button frame
btn_frame = Frame(DataLeftFrame, bd =2, relief = RIDGE, bg = 'white')
btn_frame.place(x = 0, y = 435, width = 650, height = 38)

btn_Add = Button(btn_frame, text = "Save", command = self.add_data, font = ('arial', 10, 'bold'), width = 17, bg = 'blue', fg = 'white')
btn_Add.grid(row =0, column = 0, padx =1)

btn_update = Button(btn_frame, text = "Update", command = self.update_data, font = ('arial', 10, 'bold'), width = 17, bg = 'blue', fg = 'white')
btn_update.grid(row =0, column = 1, padx =1)

btn_delete = Button(btn_frame, text = "Delete", command = self.delete_data, font = ('arial', 10, 'bold'), width = 17, bg = 'blue', fg = 'white')
btn_delete.grid(row =0, column = 2, padx =1)

btn_reset = Button(btn_frame, text = "Reset", command = self.reset_data, font = ('arial', 10, 'bold'), width = 17, bg = 'blue', fg = 'white')
btn_reset.grid(row =0, column = 3, padx =1)

# Right frame
DataRightFrame = LabelFrame(Management_frame, bd = 4, relief = RIDGE, padx = 2, text = "Patient Information ", font = ('times new roman', 12, 'bold'), fg = "red", bg = "white")
DataRightFrame.place(x = 650, y = 10, width = 650, height = 540)

# Right frame picture
img_6= Image.open(r'C:\Users\thicc\Desktop\Python\Work\Learn\Hospital Mangement\Hospital images\6th.jpg')
img_6 = img.resize((630,200), Image.ANTIALIAS)
self.photoimg_6 = ImageTk.PhotoImage(img_6)

# to display the images
self.btn_6 = Button(DataRightFrame, image = self.photoimg_6, cursor = "hand2")
self.btn_6.place(x = 0, y = 0, width = 630, height = 120)

# Search Patient Info frame
Search_Frame = LabelFrame(DataRightFrame, bd = 4, relief = RIDGE, padx = 2, text = "Search Patient Info ", font = ('times new roman', 12, 'bold'), fg = "red", bg = "white")
Search_Frame.place(x = 0, y = 120, width = 630, height = 70)

# Search By
search_by = Label(Search_Frame, font = ('arial', 10, 'bold'),text = "Search By:", fg = 'red',bg = "black")
search_by.grid(row = 0, column = 0, sticky = W, padx = 2)

# Search
self.var_com_search = StringVar()
com_txt_search=ttk.Combobox(Search_Frame,textvariable=self.var_com_search,font=('arial',10,'bold'),width=18,state="readonly")
com_txt_search['value'] = ("Select Option", "Patient No", "Phone", "Patient ID")
com_txt_search.current(0)
com_txt_search.grid(row = 0, column = 1, sticky = W, padx = 5)
```

# Computer Science(083) PROJECT FILE

```
self.var_search = StringVar()
txt_search = ttk.Entry(Search_Frame, textvariable=self.var_search ,font = ('arial', 10, 'bold'), width = 22)
txt_search.grid(row = 0, column = 2, padx = 5)

btn_search = Button(Search_Frame,command=self.search_data,text="Search",font=('arial',10,'bold'),width=10,bg='blue',fg='white')
btn_search.grid(row =0, column = 3, padx =5)

btn_ShowAll = Button(Search_Frame,command=self.fetch_data,text="Show All",font=('arial',10,'bold'),width=10,bg='blue',fg='white')
btn_ShowAll.grid(row =0, column = 4, padx =5)

table_frame = Frame(DataRightFrame, bd = 4, relief = RIDGE)
table_frame.place(x = 0, y = 190, width = 640, height = 300)

scroll_x = ttk.Scrollbar(table_frame, orient=HORIZONTAL)
scroll_y=ttk.Scrollbar(table_frame, orient=VERTICAL)
self.patient_table=ttk.Treeview(table_frame,column=(

"dep","past","room","floor","id","name","div","roll","gender","dob","email","phone","address","doctor",), 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.patient_table.xview)
scroll_y.config(command = self.patient_table.yview)

self.patient_table.heading("dep", text = "Department")
self.patient_table.heading("past", text = "Past Health")
self.patient_table.heading("room", text = "Rooms")
self.patient_table.heading("floor", text = "Floor")
self.patient_table.heading("id", text = "patient ID")
self.patient_table.heading("name", text = "patient Name")
self.patient_table.heading("div", text = "Class Div")
self.patient_table.heading("roll", text = "Roll No")
self.patient_table.heading("gender", text = "Gender")
self.patient_table.heading("dob", text = "DOB")
self.patient_table.heading("email", text = "Email")
self.patient_table.heading("phone", text = "Phone No")
self.patient_table.heading("address", text = "Address")
self.patient_table.heading("doctor", text = "Doctor Name")

self.patient_table["show"] = "headings"

self.patient_table.column("dep", width = 100)
self.patient_table.column("past", width = 100)
self.patient_table.column("room", width = 100)
self.patient_table.column("floor", width = 100)
self.patient_table.column("id", width = 100)
self.patient_table.column("name", width = 100)
self.patient_table.column("div", width = 100)
self.patient_table.column("roll", width = 100)
self.patient_table.column("gender", width = 100)
self.patient_table.column("dob", width = 100)
self.patient_table.column("email", width = 100)
self.patient_table.column("phone", width = 100)
self.patient_table.column("address", width = 100)
self.patient_table.column("doctor", width = 100)

self.patient_table.pack(fill = BOTH, expand = 1)
self.patient_table.bind("<ButtonRelease>", self.get_cursor)
self.fetch_data()
```

# Computer Science(083) PROJECT FILE

# Computer Science(083) PROJECT FILE

```
if (self.var_dep.get() == "" or self.var_email.get() == "" or self.var_patient_id.get() == ""):  
    messagebox.showerror("Error", "All Fields are required")  
else:  
    try:  
        update = messagebox.askyesno("Update", "Are you sure update this patient data", parent = self.root)  
        if update > 0:  
            conn = mysql.connector.connect(  
                host = "localhost",  
                user = "root",  
                database = "HOSPITAL"  
            )  
            my_cursur = conn.cursor()  
            my_cursur.execute("update patient set Dep=%s,past=%s,Room=%s,floor=%s,Name=%s,Division=%s,Roll=%s,Gender=%s,Dob=%s,Email=%s,Phone=%s,Address=%s,Doctor=%s where patient_id =%s",(  
                self.var_dep.get(),  
                self.var_past.get(),  
                self.var_room.get(),  
                self.var_floor.get(),  
                self.var_patient_name.get(),  
                self.var_div.get(),  
                self.var_roll.get(),  
                self.var_gender.get(),  
                self.var_dob.get(),  
                self.var_email.get(),  
                self.var_phone.get(),  
                self.var_address.get(),  
                self.var_doctor.get(),  
                self.var_patient_id.get()  
            ))  
        else:  
            if not update:  
                return  
        conn.commit()  
        self.fetch_data()  
        conn.close()  
  
        messagebox.showinfo("Success", "Updated!", parent = self.root)  
  
    except Exception as es:  
        messagebox.showerror("Error", f"Due To: {str(es)}", parent = self.root)  
  
  
# Delete Data  
def delete_data(self):  
    if self.var_patient_id.get() == "":  
        messagebox.showerror("Error", "All Fields are required")  
    else:  
        try:  
            Delete = messagebox.askyesno("Delete", "Are you sure you wanna delete this Patient", parent = self.root)  
            if Delete > 0:  
                conn = mysql.connector.connect(  
                    host = "localhost",  
                    user = "root",  
                    database = "HOSPITAL"  
                )  
                my_cursur = conn.cursor()  
                sql = "delete from patient where patient_id = %s"  
                value = (self.var_patient_id.get(),)  
                my_cursur.execute(sql, value)  
            else:  
                if not Delete:  
                    return  
  
            conn.commit()  
            self.fetch_data()  
            conn.close()  
            messagebox.showinfo("Delete", "Your Patient has been deleted!", parent = self.root)  
        except Exception as es:  
            messagebox.showerror("Error", f"Due To: {str(es)}", parent = self.root)
```

# Computer Science(083) PROJECT FILE

```
# Reset
def reset_data(self):
    self.var_dep.set("Select Department"),
    self.var_past.set("Select Past Healt"),
    self.var_room.set("Select Room"),
    self.var_floor.set("Select Floor"),
    self.var_patient_id.set(""),
    self.var_patient_name.set(""),
    self.var_div.set("Select Division"),
    self.var_rol.set(""),
    self.var_gender.set(""),
    self.var_dob.set(""),
    self.var_email.set(""),
    self.var_phone.set(""),
    self.var_address.set(""),
    self.var_doctor.set("")

# Search Data
def search_data(self):
    if self.var_com_search.get() == "" or self.var_search.get() == "":
        messagebox.showerror("Error", "Please Select option", parent = self.root)
    else:
        try:
            conn = mysql.connector.connect(
                host = "localhost",
                user = "root",
                database = "HOSPITAL"
            )
            my_cursur = conn.cursor()
            my_cursur.execute("select * from patient where " +str(self.var_com_search.get())+" LIKE '%"+str(self.var_search.get())+"%'")
        )
        data = my_cursur.fetchall()
        if len(data) != 0:
            self.patient_table.delete(*self.patient_table.get_children())
            for i in data:
                self.patient_table.insert("", END, values = i)
            conn.commit()
            conn.close()
        except Exception as es:
            messagebox.showerror("Error", f"Due To: {str(es)}", parent = self.root)

if __name__ == "__main__":
    root = Tk()
    obj = Patient(root)
    root.mainloop()
```

## (5.) Output

On Startup of the program,

(I.) Database Hospital is created with table patient

Hospital(database). → Patient(table)

The screenshot shows a terminal window titled 'c:\wamp64\bin\mysql\mysql5.7.31\bin\mysql.exe'. It displays the MySQL monitor welcome message, server version information, copyright notice, and trademark information. The user then runs the 'show databases;' command, which lists several databases including 'information\_schema', 'college', 'hospital', 'mysql', 'performance\_schema', 'store', 'sys', 'test', 'try7', and 'university'. The output shows '10 rows in set (0.00 sec)'. Next, the user runs 'use hospital;', changing the database context. Finally, the user runs 'show tables;', which lists the single table 'patient' in the 'hospital' database. The output shows '1 row in set (0.00 sec)'.

```
c:\wamp64\bin\mysql\mysql5.7.31\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 85
Server version: 5.7.31 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| college |
| hospital |
| mysql |
| performance_schema |
| store |
| sys |
| test |
| try7 |
| university |
+-----+
10 rows in set (0.00 sec)

mysql> use hospital;
Database changed
mysql> show tables;
+-----+
| Tables_in_hospital |
+-----+
| patient |
+-----+
1 row in set (0.00 sec)
```

# Computer Science(083) PROJECT FILE

```
c:\wamp64\bin\mysql\mysql5.7.31\bin\mysql.exe
mysql> use hospital;
Database changed
mysql> show tables;
+-----+
| Tables_in_hospital |
+-----+
| patient |
+-----+
1 row in set (0.00 sec)

mysql> describe patient;
+-----+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Dep | varchar(45) | YES | NULL |
| Past | varchar(45) | YES | NULL |
| Room | varchar(45) | YES | NULL |
| Floor | varchar(45) | YES | NULL |
| patient_id | int(11) | YES | NULL |
| Name | varchar(45) | YES | NULL |
| Division | varchar(45) | YES | NULL |
| Roll | varchar(45) | YES | NULL |
| Gender | varchar(45) | YES | NULL |
| Dob | varchar(45) | YES | NULL |
| Email | varchar(45) | YES | NULL |
| Phone | varchar(45) | YES | NULL |
| Address | varchar(45) | YES | NULL |
| Doctor | varchar(45) | YES | NULL |
+-----+-----+-----+-----+-----+-----+
14 rows in set (0.04 sec)

mysql>
```

Activate Windows  
Go to Settings to activate Windows

(2.) Then, the GUI window will appear



# Computer Science(083) PROJECT FILE

## (3.) Data is added to table

```
Select c:\wamp64\bin\mysql\mysql5.7.31\bin\mysql.exe
Tables_in_hospital
+-----+
| patient |
+-----+
1 row in set (0.00 sec)

mysql> describe patient;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |          |          |          |          |          |          |          |          |          |          |          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Dep    | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |          |          |          |
| Past   | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |          |          |
| Room   | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |          |          |
| Floor  | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |          |
| patient_id | int(11) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |          |
| Name   | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |          |
| Division | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |          |
| Roll   | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |
| Gender  | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |
| Dob    | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |
| Email   | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |
| Phone   | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |
| Address | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |
| Doctor  | varchar(45) | YES  |      | NULL    |       |          |          |          |          |          |          |          |          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
14 rows in set (0.04 sec)

mysql> select * from patient;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Dep    | Past   | Room  | Floor | patient_id | Name      | Division | Roll | Gender | Dob     | Email    | Phone   | Address |
| Doctor |        |        |        |            |           |          |       |       |       |          |          |         |          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Physician | YES  | Normal | Floor-1 | 1 | Arihant Debnath | A | AD1 | Male | 26/09/2003 | arihantd2609@gmail.com | 9810725378 | 137 A
| Shipra | Dr. Henry | NO   | Medium | Floor-2 | 2 | Rohan Banerjee | Select Division | RB2 | Male | 10/10/2000 | rohan@gmail.com | 9910723456 | 136 A
| Shipra | Dr. Verma |      |        |        | 3 | Siddhant Debnath | B | SD3 | Male | 10/10/2000 | rohan@gmail.com | 9910723456 | 136 A
| Pediatricain | YES  | High  | Floor-2 | 4 | Sento Nath     | C | SN4 | Male | 10/10/2000 | sentu@gmail.com | 9910723456 | 136 A
| Shipra | Dr. Rohan |      |        |        | 5 | Tullika Nath    | C | TN5 | Female | 08/04/1977 | tullika@gmail.com | 9910723456 | 138 A
| Medicine | YES  | VIP   | Floor-3 |        |        |        |        |        |        |        |        |        |        |
| Neurosurgery | YES  | VIP   | Floor-4 |        |        |        |        |        |        |        |        |        |        |
| Shipra | Dr. Davus |      |        |        |        |        |        |        |        |        |        |        |        |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

## Data storage in MySQL Database

```
mysql> select * from patient;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Dep    | Past   | Room  | Floor | patient_id | Name      | Division | Roll | Gender | Dob     | Email    | Phone   | Address |
| Doctor |        |        |        |            |           |          |       |       |       |          |          |         |          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Physician | YES  | Normal | Floor-1 | 1 | Arihant Debnath | A | AD1 | Male | 26/09/2003 | arihantd2609@gmail.com | 9810725378 | 137 A
| Shipra | Dr. Henry | NO   | Medium | Floor-2 | 2 | Rohan Banerjee | Select Division | RB2 | Male | 10/10/2000 | rohan@gmail.com | 9910723456 | 136 A
| Shipra | Dr. Verma |      |        |        | 3 | Siddhant Debnath | B | SD3 | Male | 10/10/2000 | rohan@gmail.com | 9910723456 | 136 A
| Pediatricain | YES  | High  | Floor-2 | 4 | Sento Nath     | C | SN4 | Male | 10/10/2000 | sentu@gmail.com | 9910723456 | 136 A
| Shipra | Dr. Rohan |      |        |        | 5 | Tullika Nath    | C | TN5 | Female | 08/04/1977 | tullika@gmail.com | 9910723456 | 138 A
| Medicine | YES  | VIP   | Floor-3 |        |        |        |        |        |        |        |        |        |
| Neurosurgery | YES  | VIP   | Floor-4 |        |        |        |        |        |        |        |        |        |
| Shipra | Dr. Davus |      |        |        |        |        |        |        |        |        |        |        |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

### Patient Information



### Search Patient Info

Search By:	Select Option	Search	Show All		
Department	Past Health	Rooms	Floor	patient ID	patient Name
Physician	YES	Normal	Floor-1	1	Arihant Debnath A
Cardiovascular	NO	Medium	Floor-2	2	Rohan Banerjee S
Pediatricain	YES	High	Floor-2	3	Siddhant Debnath B
Medicine	YES	VIP	Floor-3	4	Sento Nath C

# Computer Science(083) PROJECT FILE

## Patient Information



### Search Patient Info

Search By: Select Option

patient Name	Class Div	Roll No	Gender	DOB	Email
Arihant Debnath	A	AD1	Male	26/09/2003	arihantd2609@gm
Rohan Banerjee	Select Division	RB2	Male	10/10/2000	rohan@gmail.co
Siddhant Debnath	B	SD3	Male	10/10/2000	rohan@gmail.co
Sentu Nath	C	SN4	Male	10/10/2000	sentu@gmail.con

## (4.) Adding Data(Create -->CURD)

HOSPITAL MANAGEMENT SYSTEM

**Patient Information**

**Visiting hours**  
11am - 1.30pm and 4pm - 8pm daily  
Some wards may have specific visiting rules - please obey any notices or instructions shown on these wards or advised by our staff.

**Hospital Department information**

Department :  Prior Health issues :

Room Type :  Hospital Floor :

**Patient Frame information**

PatientID :  Patient Name :   
Class Division :  Patient No :   
Gender :  DOB :   
Email :  Phone No :   
Address :  Doctor :

**Patient Information**

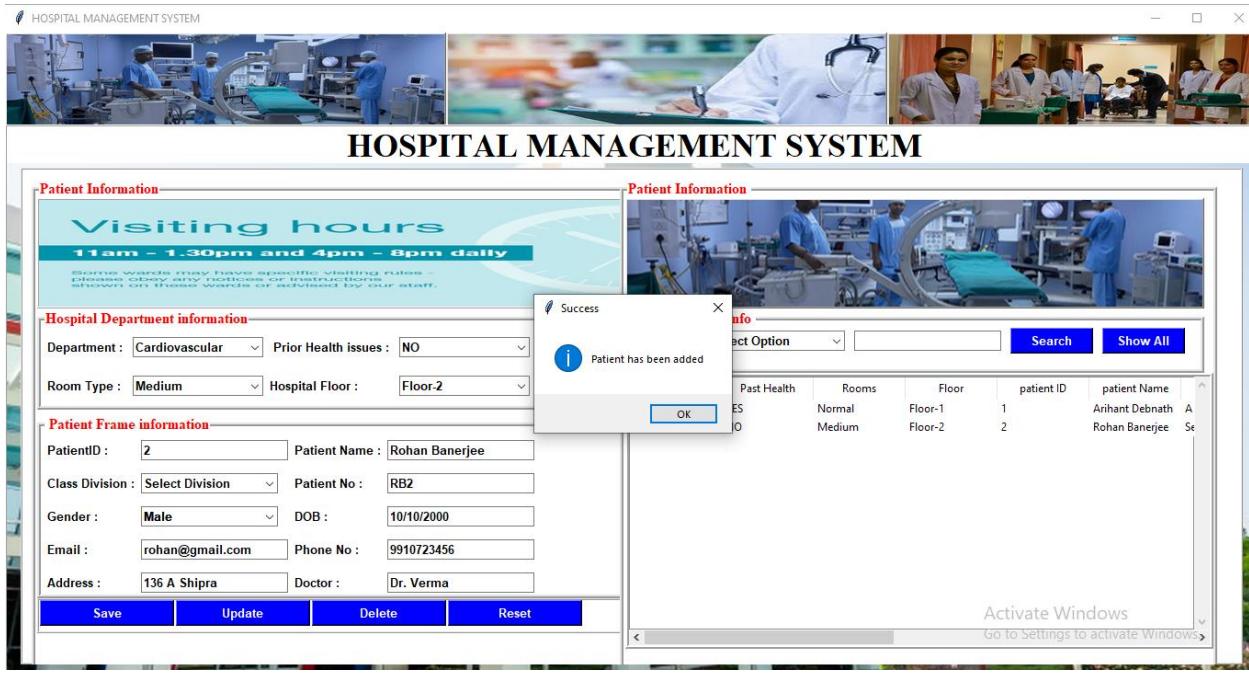
**Search Patient Info**

Search By: Select Option

Department	Past Health	Rooms	Floor	patient ID	patient Name
Physician	YES	Normal	Floor-1	1	Arihant Debnath A

Activate Windows  
Go to Settings to activate Windows

# Computer Science(083) PROJECT FILE

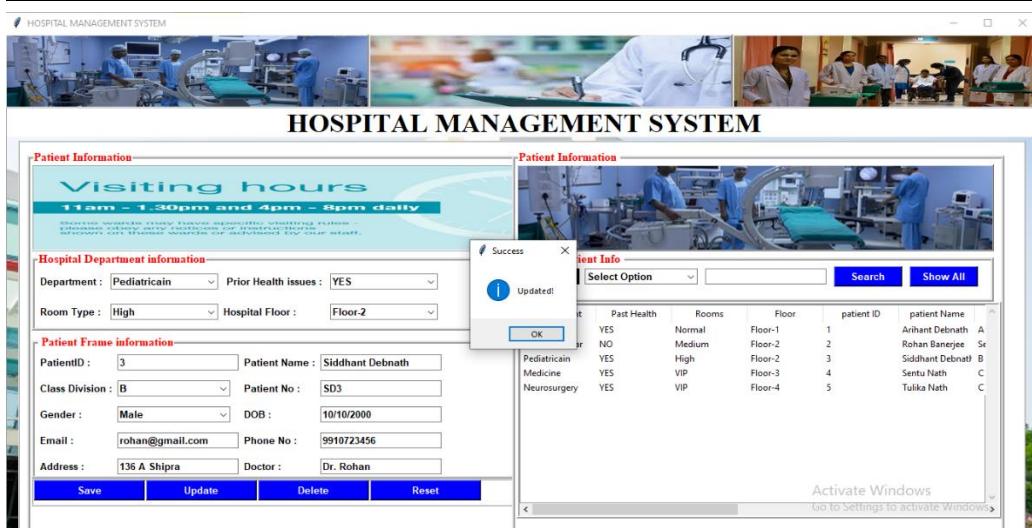
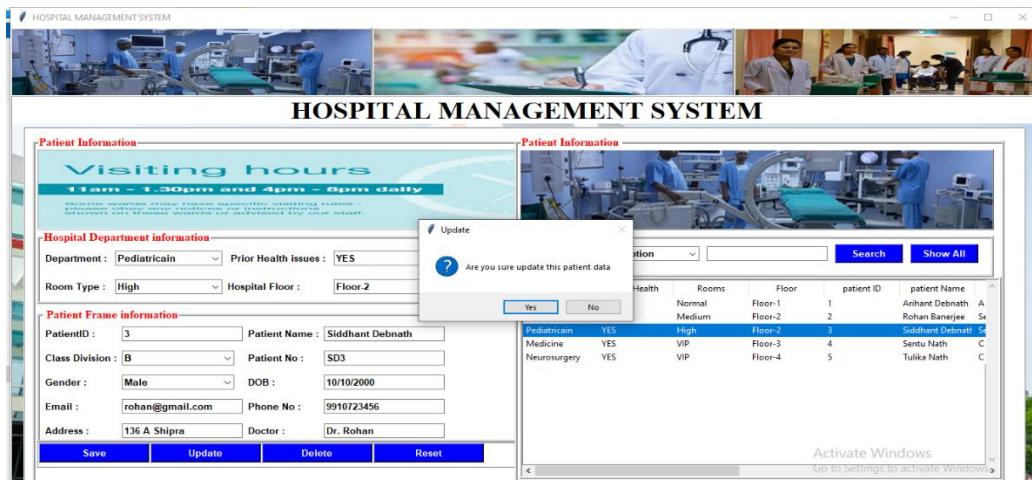


## Adding all the data



# Computer Science(083) PROJECT FILE

## (5.) Updating the Data



# Computer Science(083) PROJECT FILE

## (6.) Reseting the Data

HOSPITAL MANAGEMENT SYSTEM

The screenshot shows the Hospital Management System interface. On the left, the 'Patient Information' page displays 'Visiting hours' (11am - 1.30pm and 4pm - 8pm daily), 'Hospital Department information' (Department: Neurosurgery, Prior Health issues: YES, Room Type: VIP, Hospital Floor: Floor-4), and 'Patient Frame information' (PatientID: 5, Patient Name: Tulika Nath, Class Division: C, Patient No: TN5, Gender: Female, DOB: 08/04/1977, Email: tullika@gmail.com, Phone No: 9910723456, Address: 138 A Shipra, Doctor: Dr. Davus). Below these are buttons for Save, Update, Delete, and Reset. On the right, a 'Search Patient Info' section includes a search bar and buttons for 'Search' and 'Show All'. A table lists patient data:

Department	Past Health	Rooms	Floor	patient ID	patient Name
Physician	YES	Normal	Floor-1	1	Arihant Debnath A
Cardiovascular	NO	Medium	Floor-2	2	Rohan Banerjee S
Pediatrician	YES	High	Floor-2	3	Siddhant Debnath B
Medicine	YES	VIP	Floor-3	4	Sentu Nath C

Activate Windows  
Go to Settings to activate Windows

HOSPITAL MANAGEMENT SYSTEM

The screenshot shows the Hospital Management System interface. On the left, the 'Patient Information' page displays 'Visiting hours' (11am - 1.30pm and 4pm - 8pm daily), 'Hospital Department information' (Department: Select Department, Prior Health issues: Select Past Healt, Room Type: Select Room, Hospital Floor: Select Floor), and 'Patient Frame information' (PatientID: [empty], Patient Name: [empty], Class Division: Select Division, Patient No: [empty], Gender: [empty], DOB: [empty], Email: [empty], Phone No: [empty], Address: [empty], Doctor: [empty]). Below these are buttons for Save, Update, Delete, and Reset. On the right, a 'Search Patient Info' section includes a search bar and buttons for 'Search' and 'Show All'. A table lists patient data:

Department	Past Health	Rooms	Floor	patient ID	patient Name
Physician	YES	Normal	Floor-1	1	Arihant Debnath A
Cardiovascular	NO	Medium	Floor-2	2	Rohan Banerjee S
Pediatrician	YES	High	Floor-2	3	Siddhant Debnath B
Medicine	YES	VIP	Floor-3	4	Sentu Nath C

Activate Windows  
Go to Settings to activate Windows

# Computer Science(083) PROJECT FILE

## (7.) Searching Data

The screenshot shows the Hospital Management System interface. On the left, the 'Patient Information' page displays visiting hours (11am - 1.30pm and 4pm - 8pm daily), hospital department information (Department: Select Department, Room Type: Select Room, Hospital Floor: Select Floor), and patient frame information (PatientID: [ ] Patient Name: [ ] Class Division: Select Division, Patient No: [ ] Gender: [ ] DOB: [ ] Email: [ ] Phone No: [ ] Address: [ ] Doctor: [ ]). Below these are buttons for Save, Update, Delete, and Reset. On the right, a search results table titled 'Search Patient Info' is shown, with columns for Department, Past Health, Rooms, Floor, patient ID, and patient Name. The table contains data for five patients: Arihant Debnath (A), Rohan Banerjee (B), Siddhant Debnath (C), and Sentu Nath (D). A 'Search By' dropdown menu is set to 'Patient ID'. Buttons for 'Search' and 'Show All' are also present.

## (8.) Deleting Data

The screenshot shows the Hospital Management System interface. On the left, the 'Patient Information' page displays visiting hours (11am - 1.30pm and 4pm - 8pm daily), hospital department information (Department: Neurosurgery, Room Type: VIP, Hospital Floor: Floor-4), and patient frame information (PatientID: 5, Patient Name: Tulika Nath, Class Division: C, Patient No: TN5, Gender: Female, DOB: 08/04/1977, Email: tulika@gmail.com, Phone No: 9910723456, Address: 138 A Shipra, Doctor: Dr. Davus). Below these are buttons for Save, Update, Delete, and Reset. A 'Delete' confirmation dialog box is overlaid on the page, asking 'Are you sure you wanna delete this Patient?' with 'Yes' and 'No' buttons. On the right, a search results table titled 'Search Patient Info' is shown, with columns for Department, Past Health, Rooms, Floor, patient ID, and patient Name. The table contains data for five patients: Arihant Debnath (A), Rohan Banerjee (B), Siddhant Debnath (C), and Sentu Nath (D). A 'Search By' dropdown menu is set to 'Patient ID'. Buttons for 'Search' and 'Show All' are also present.

# Computer Science(083) PROJECT FILE

HOSPITAL MANAGEMENT SYSTEM

**Patient Information**

**Visiting hours**  
11am - 1.30pm and 4pm - 8pm daily  
Some Wards may have specific visiting rules.  
Please obey any notices or instructions shown on these Wards or advised by our staff.

**Hospital Department information**

Department :	Neurosurgery	Prior Health Issues :	YES
Room Type :	VIP	Hospital Floor :	Floor-4

**Patient Frame information**

PatientID :	5	Patient Name :	Tulika Nath
Class Division :	C	Patient No :	TN5
Gender :	Female	DOB :	08/04/1977
Email :	tulika@gmail.com	Phone No :	9910723456
Address :	138 A Shipra	Doctor :	Dr. Davus

**Delete**

Your Patient has been deleted!

**OK**

**Past Health**   **Rooms**   **Floor**   **patient ID**   **patient Name**

Normal	Floor-1	1	Arihant Debnath	A
Medium	Floor-2	2	Rohan Banerjee	B
High	Floor-2	3	Siddhant Debnath	B
VIP	Floor-3	4	Sentu Nath	C

Activate Windows  
Go to Settings to activate Windows

**Patient Information**

**Search Patient Info**

**Search By:** Select Option

**Search**   **Show All**

**Past Health**   **Rooms**   **Floor**   **patient ID**   **patient Name**

Physician	YES	Normal	Floor-1	1	Arihant Debnath	A
Cardiovascular	NO	Medium	Floor-2	2	Rohan Banerjee	B
Pediatricain	YES	High	Floor-2	3	Siddhant Debnath	B
Medicine	YES	VIP	Floor-3	4	Sentu Nath	C

Activate Windows  
Go to Settings to activate Windows

## (6.) Conclusion

- I have developed a Hospital mangemnet system based on CURD with MySQL bacend and Python frontend.
- The software stores the patient data and also displays it through the python frontend.
- It also allows the user to manipullate the data as much as they want.
- The software has some real world applications and can be used by many small clinics and hospitals for their day to day management.
- It allows the user to add, update, reset and delete the data.
- The program has interactive GUI using Tkinter which is user friendly

## (7.) Further Improvement

- Inventory management system for medicine
- Doctor's record/ Staff records etc
- Better UI
- Could be made more user-friendly
- Adding a pharmacy system for managing a medicine dispensary.
- Hospital beds Management with real time update
- Integrated system with other Health Management System
- Forming a network for management system for different hospitals with syncing of data.

## (8.) BIBLIOGRAPHY

- <https://www.google.co.in>
- <https://en.wikipedia.org>
- Geeksforgeeks.org
- CLASS 12 COMPUTER SCIENCE WITH PYTHON by SUMITA ARORA
- <https://docs.python.org/3/library/tkinter.html>
- <https://dev.mysql.com/doc/>