Hospital Management System

NAME:- KUMAR ARJUN

CLASS:- XII C

BOARD ROLL NO.:-

DELHI PUBLIC SCHOOL INDIRAPURAM

CERTIFICATE

This is to certify that KUMAR ARJUN of class XII-C has prepared the project on“HOSPITAL MANAGEMENT SYSTEM”. The project is result of his efforts and endeavours. 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

(PGT Computerscience)

(DPSIndirapuram)

Acknowledgement

I would like to express a deep sense of gratitude towards my computer science teacher Ms. Rinkoo Gupta for guiding me though 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.

KUMAR ARJUN

XII-C

Index

Introduction to the Project

- Header Files Used:-
    - CSV module
    - Tkinter module
    - PIL
    - mysqlconnector
    - time module

- WORKING DESCRIPTION:-

  The application has been designed in order to facilitate and bring ease in management of data in a hospital. Better inter-connectivity between different nodes in order to bring together a smoother experience.

  The application starts with checking all the data structure and creates them if not already present. Then the useer is greeted at the welcome screen. After clicking on login the user can enter the application through five different modes that are

    - Administrator
    - Cashier
    - Doctor
    - Patient
    - Pharmacist

  The user can login with the credentials provided to him

- Administrator/other employees

  The Administrator is the one who does his work and fixes the backend in case any issue aries. He can monitor most aspects however can affect only a few of them. He also has access to the pharmacy and appointment management. These features are maintained for other employees who work at the hospital. The person who sets the appointments can look through the free time availbale to a doctor on any paticular day and allot any slot of 15mins each to any patient who gets the doctors fees charged on them and stored in the transanction table while the appointment is now visible to the doctor as well as the patient on his window.

- Pharmacist

  The pharmacist as stated earlier manages the pharmacy directory and is also responsible for alloting medicines into the accounts of the patients. The transanction is stored in transanction table while the stock is reduced from the pharmacy table and the account balance is updated at the patient table simultaneously. We can also change the stock details and add new medicines to the tables using this interface.

- Patient

  The patient can view his account check different doctors available at the hospital as well as they can link minor accounts for insurance ease(not included but kept as a scope of futher development). Also one can see any prescription to his name based on date and doctor.

- Doctor

  The doctor who has given his working hours to the hospital has the flexiblity to reshedule them according to his needs as well. He can look at a days appointments and also write prescriptions to them using the application

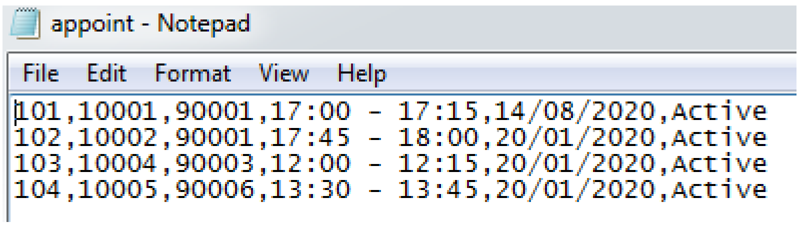
- Cashier

  The cashier is the one who takes money from the patient and makees the records in the computer. He gets the all past and pending transanction details of the patient whether it is at the pharmacy or doctor's fee all are managed using this window.

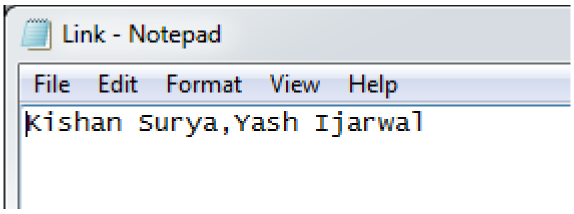
Storage Structure

The data is stored in multiple ways each depending upon on its purpose and ease of extraction. The appointment details, doctor's flexible time and account linking is based on CSV storage while the data related to transanction, patient details, employee details and pharmacy directory is stroed on MySQL linked with the help of mySQLConnector. Finally the prescription made by doctors are stored in a patient's personal directory as a text file.

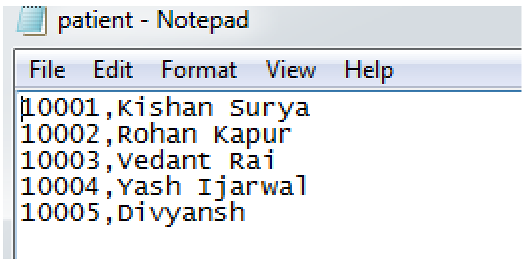
CSV FILES:-



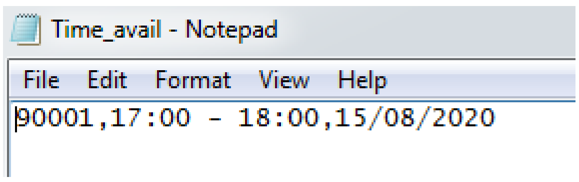
```
101,10001,90001,17:00 - 17:15,14/08/2020,Active
102,10002,90001,17:45 - 18:00,20/01/2020,Active
103,10004,90003,12:00 - 12:15,20/01/2020,Active
104,10005,90006,13:30 - 13:45,20/01/2020,Active
```



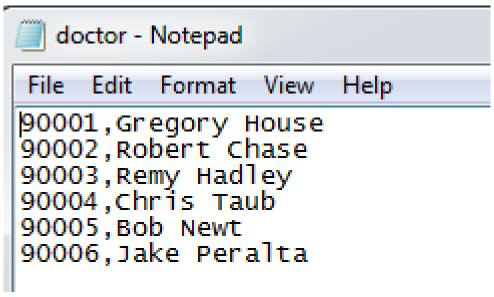
```
Kishan Surya,Yash Ijarwal
```



```
10001,Kishan Surya
10002,Rohan Kapur
10003,Vedant Rai
10004,Yash Ijarwal
10005,Divyansh
```



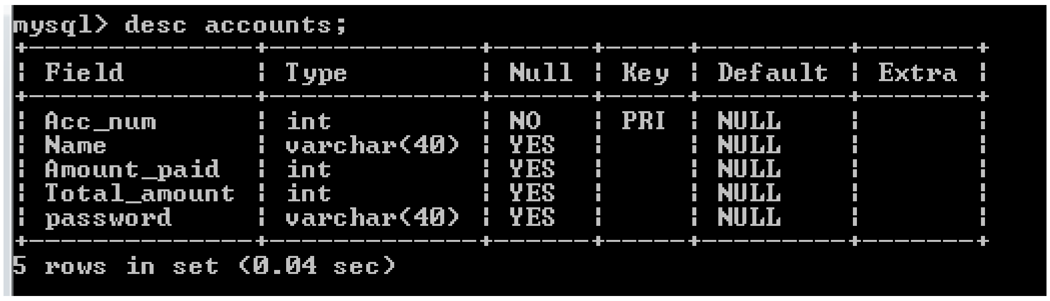
```
90001,17:00 - 18:00,15/08/2020
```



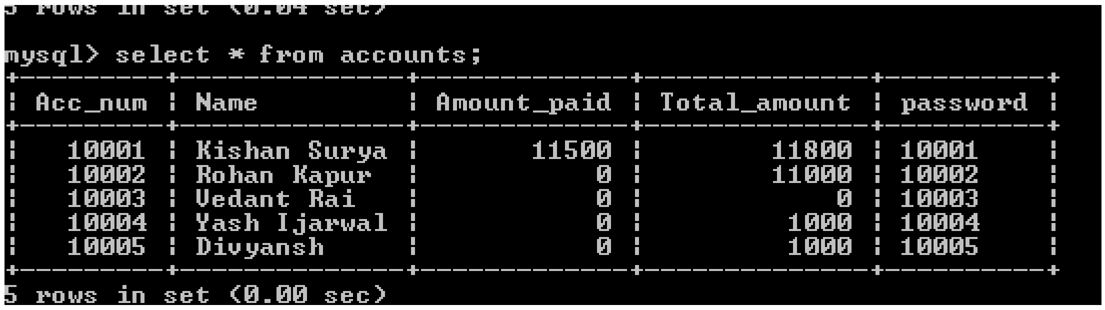
```
90001,Gregory House
90002,Robert Chase
90003,Remy Hadley
90004,Chris Taub
90005,Bob Newt
90006,Jake Peralta
```

MySQL TABLES:-

Accounts table:-

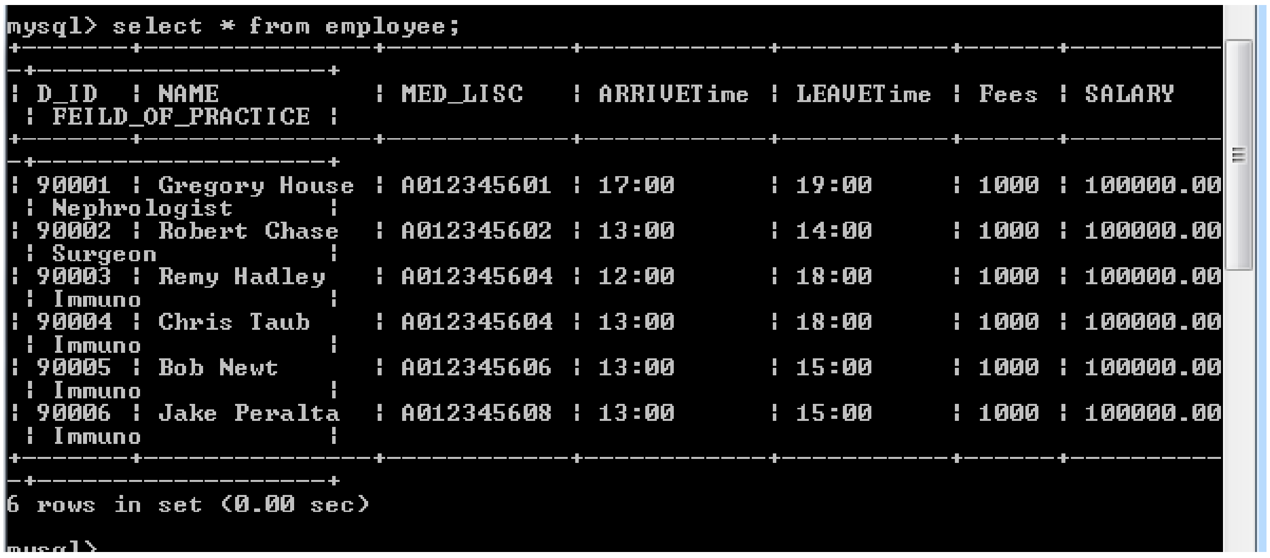


```
mysql> desc accounts;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Acc_num      | int         | NO   | PRI | NULL    |       |
| Name         | varchar(40) | YES  |     | NULL    |       |
| Amount_paid  | int         | YES  |     | NULL    |       |
| Total_amount | int         | YES  |     | NULL    |       |
| password     | varchar(40) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.04 sec)
```

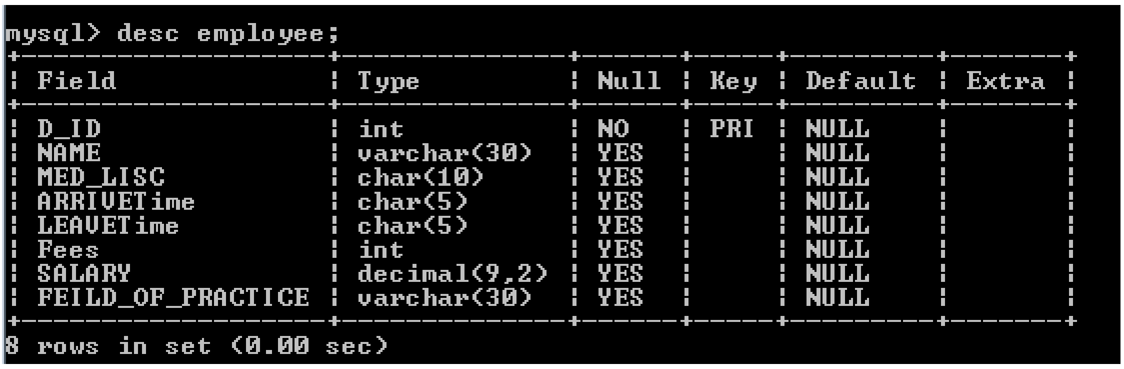


```
5 rows in set (0.04 sec)

mysql> select * from accounts;
+---------+--------------+-------------+--------------+----------+
| Acc_num | Name         | Amount_paid | Total_amount | password |
+---------+--------------+-------------+--------------+----------+
|   10001 | Kishan Surya |       11500 |        11800 | 10001    |
|   10002 | Rohan Kapur  |           0 |        11000 | 10002    |
|   10003 | Vedant Rai   |           0 |            0 | 10003    |
|   10004 | Yash Ijarwal |           0 |         1000 | 10004    |
|   10005 | Divyansh     |           0 |         1000 | 10005    |
+---------+--------------+-------------+--------------+----------+
5 rows in set (0.00 sec)
```

Employee table

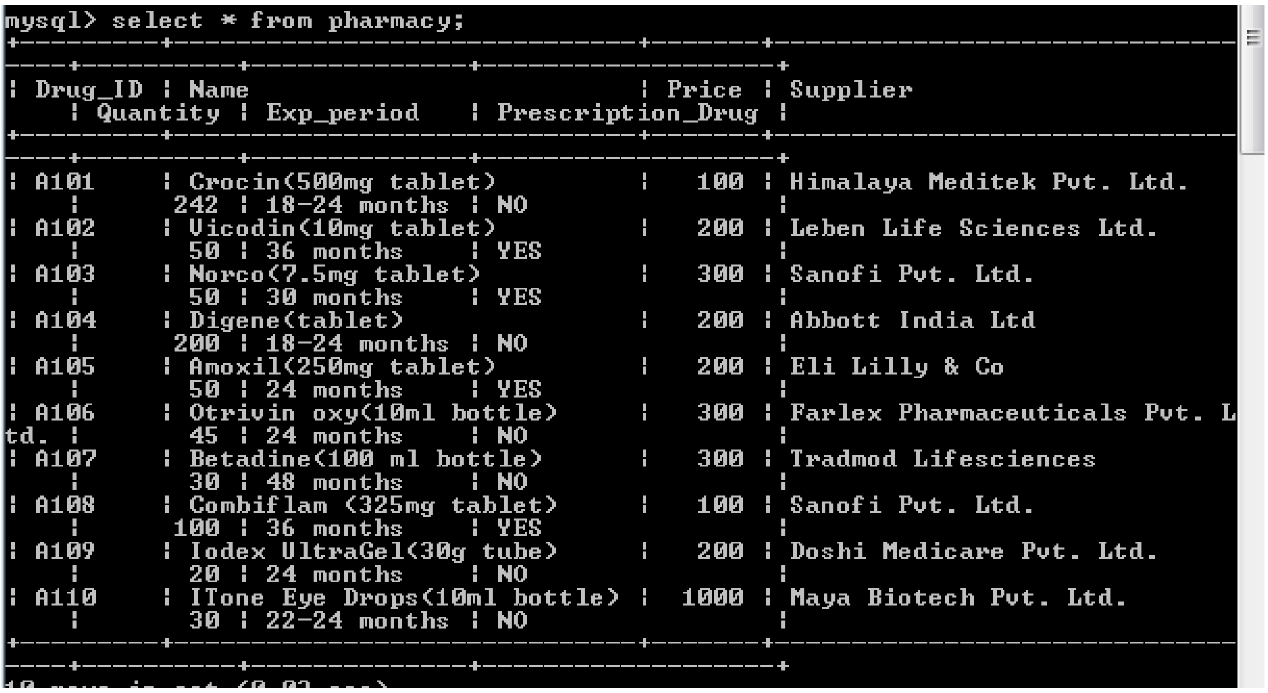


```
mysql> select * from employee;
+-------+---------------+-------------+------------+------------+------+----------
-+-------------------+
| D_ID  | NAME          | MED_LISC    | ARRIVETime | LEAVETime  | Fees | SALARY
 | FEILD_OF_PRACTICE |
+-------+---------------+-------------+------------+------------+------+----------
-+-------------------+
| 90001 | Gregory House | A012345601  | 17:00      | 19:00      | 1000 | 100000.00
 | Nephrologist      |
| 90002 | Robert Chase  | A012345602  | 13:00      | 14:00      | 1000 | 100000.00
 | Surgeon           |
| 90003 | Remy Hadley   | A012345604  | 12:00      | 18:00      | 1000 | 100000.00
 | Immuno            |
| 90004 | Chris Taub    | A012345604  | 13:00      | 18:00      | 1000 | 100000.00
 | Immuno            |
| 90005 | Bob Newt      | A012345606  | 13:00      | 15:00      | 1000 | 100000.00
 | Immuno            |
| 90006 | Jake Peralta  | A012345608  | 13:00      | 15:00      | 1000 | 100000.00
 | Immuno            |
+-------+---------------+-------------+------------+------------+------+----------
-+-------------------+
6 rows in set (0.00 sec)

mysql>
```

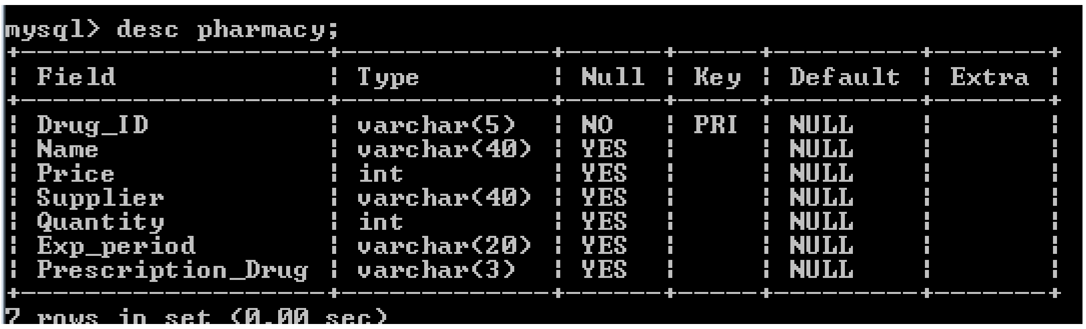


```
mysql> desc employee;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| D_ID               | int          | NO   | PRI | NULL    |       |
| NAME               | varchar(30)  | YES  |     | NULL    |       |
| MED_LISC           | char(10)     | YES  |     | NULL    |       |
| ARRIVETime         | char(5)      | YES  |     | NULL    |       |
| LEAVETime          | char(5)      | YES  |     | NULL    |       |
| Fees               | int          | YES  |     | NULL    |       |
| SALARY             | decimal(9,2) | YES  |     | NULL    |       |
| FEILD_OF_PRACTICE  | varchar(30)  | YES  |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

Pharmacy Table:-

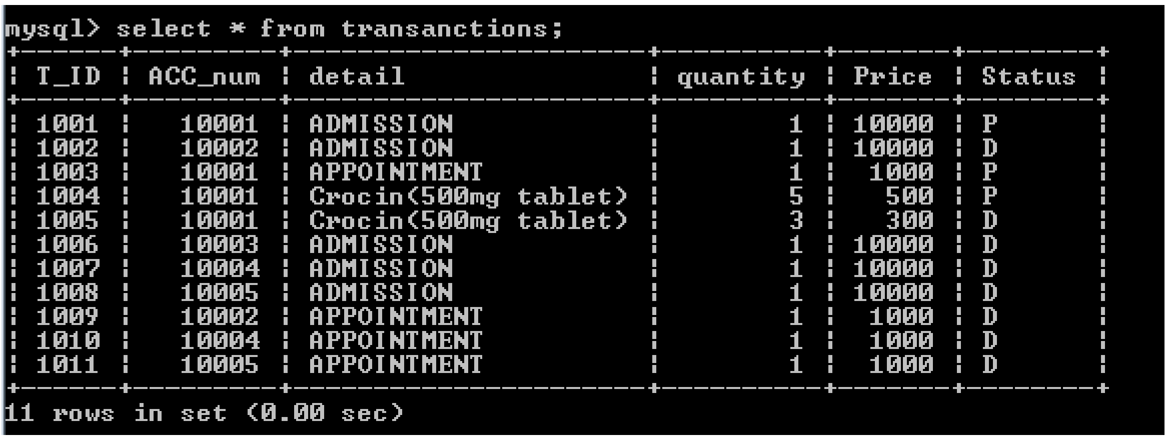


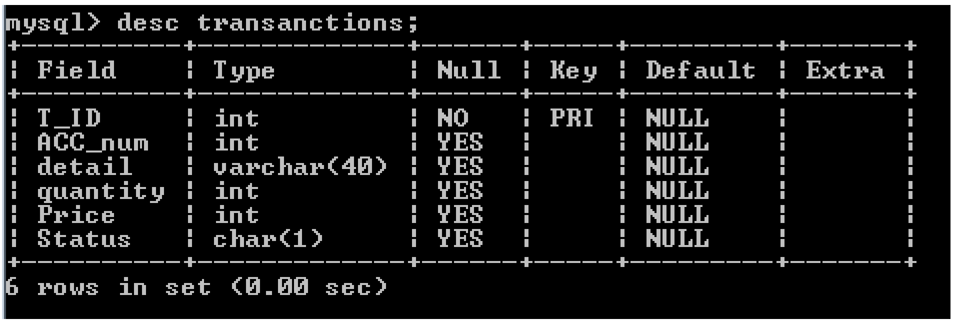
```
mysql> select * from pharmacy;
+---------+--------------------------------+--------+---------------------------
-----+----------+--------------+-------------------+
| Drug_ID | Name                           | Price  | Supplier
    | Quantity | Exp_period   | Prescription_Drug |
+---------+--------------------------------+--------+---------------------------
-----+----------+--------------+-------------------+
| A101    | Crocin(500mg tablet)           |    100 | Himalaya Meditek Pvt. Ltd.
    |      242 | 18-24 months | NO                |
| A102    | Vicodin(10mg tablet)           |    200 | Leben Life Sciences Ltd.
    |       50 | 36 months    | YES               |
| A103    | Norco(7.5mg tablet)            |    300 | Sanofi Pvt. Ltd.
    |       50 | 30 months    | YES               |
| A104    | Digene(tablet)                 |    200 | Abbott India Ltd
    |      200 | 18-24 months | NO                |
| A105    | Amoxil(250mg tablet)           |    200 | Eli Lilly & Co
    |       50 | 24 months    | YES               |
| A106    | Otrivin oxy(10ml bottle)       |    300 | Farlex Pharmaceuticals Pvt. L
td. |       45 | 24 months    | NO                |
| A107    | Betadine(100 ml bottle)        |    300 | Tradmod Lifesciences
    |       30 | 48 months    | NO                |
| A108    | Combiflam (325mg tablet)       |    100 | Sanofi Pvt. Ltd.
    |      100 | 36 months    | YES               |
| A109    | Iodex UltraGel(30g tube)       |    200 | Doshi Medicare Pvt. Ltd.
    |       20 | 24 months    | NO                |
| A110    | ITone Eye Drops(10ml bottle)   |   1000 | Maya Biotech Pvt. Ltd.
    |       30 | 22-24 months | NO                |
+---------+--------------------------------+--------+---------------------------
-----+----------+--------------+-------------------+
10 rows in set (0.02 sec)
```



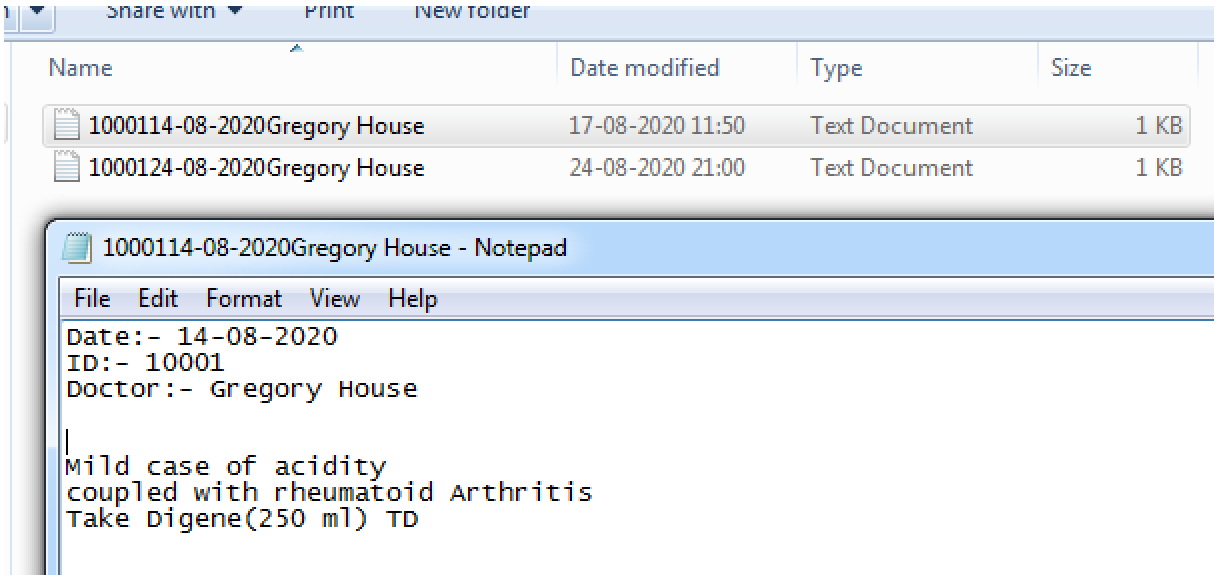
```
mysql> desc pharmacy;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| Drug_ID           | varchar(5)  | NO   | PRI | NULL    |       |
| Name              | varchar(40) | YES  |     | NULL    |       |
| Price             | int         | YES  |     | NULL    |       |
| Supplier          | varchar(40) | YES  |     | NULL    |       |
| Quantity          | int         | YES  |     | NULL    |       |
| Exp_period        | varchar(20) | YES  |     | NULL    |       |
| Prescription_Drug | varchar(3)  | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

Transancantion Table:-


```
mysql> select * from transactions;
+------+---------+-----------------------+----------+-------+--------+
| T_ID | ACC_num | detail                | quantity | Price | Status |
+------+---------+-----------------------+----------+-------+--------+
| 1001 |   10001 | ADMISSION             |        1 | 10000 | P      |
| 1002 |   10002 | ADMISSION             |        1 | 10000 | D      |
| 1003 |   10001 | APPOINTMENT           |        1 |  1000 | P      |
| 1004 |   10001 | Crocin(500mg tablet)  |        5 |   500 | P      |
| 1005 |   10001 | Crocin(500mg tablet)  |        3 |   300 | D      |
| 1006 |   10003 | ADMISSION             |        1 | 10000 | D      |
| 1007 |   10004 | ADMISSION             |        1 | 10000 | D      |
| 1008 |   10005 | ADMISSION             |        1 | 10000 | D      |
| 1009 |   10002 | APPOINTMENT           |        1 |  1000 | D      |
| 1010 |   10004 | APPOINTMENT           |        1 |  1000 | D      |
| 1011 |   10005 | APPOINTMENT           |        1 |  1000 | D      |
+------+---------+-----------------------+----------+-------+--------+
11 rows in set (0.00 sec)
```


```
mysql> desc transactions;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| T_ID     | int         | NO   | PRI | NULL    |       |
| ACC_num  | int         | YES  |     | NULL    |       |
| detail   | varchar(40) | YES  |     | NULL    |       |
| quantity | int         | YES  |     | NULL    |       |
| Price    | int         | YES  |     | NULL    |       |
| Status   | char(1)     | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

TEXT FILES:-



Name	Date modified	Type	Size
1000114-08-2020Gregory House	17-08-2020 11:50	Text Document	1 KB
1000124-08-2020Gregory House	24-08-2020 21:00	Text Document	1 KB

1000114-08-2020Gregory House - Notepad

```
Date:- 14-08-2020
ID:- 10001
Doctor:- Gregory House


Mild case of acidity
coupled with rheumatoid Arthritis
Take Digene(250 ml) TD
```

SYSTEM REQUIREMENTS

1)HARDWARE REQUIREMENTS:-

The minimum requirements needed to install Python and associated applications:

•Modern Operating System:

- Windows 7 or 10
- Mac OS X 10.11 or higher, 64-bit
- Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)

•x86 64-bit CPU (Intel / AMD architecture)
•4 GB RAM
•5 GB free disk space

2)SOFTWARE REQUIREMENTS:-

The following are the required libraies to be installed along with python:-

1) ttk along with tkinter
2) photo image library (PIL)
3) mysql connector
4) mySQL (version 8 or higher)

CODING

file #1 application

```
#1)run the installation module first
#2)change the sql password at the sqlconnection

from time import *

#The Driver Code

print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++")
print("====================Welcome to the Hospital
Management======================")
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++")
sleep(0.5)

from GUI1 import *

#checking the required packages

sleep(0.5)
print("intialising...")

sleep(0.5)
print("checking required packages")

cur.execute("show tables")
```

```
x=cur.fetchall()

for i in x:
   for k in i:
      sleep(0.5)
      print("checking structure:--"+k)
      if k in ("accounts","employee","pharmacy","transanctions"):
         sleep(0.1)
         print("sucessful!")
      else:
         sleep(0.1)
         print("error in loading \n please run the installion module")
         sleep(0.2)
         quit()

#calling the start function

sleep(0.1)
print("launching the application...")
sleep(1)
lgn_disp1()
```

File #2 GUI1

```
from tkinter import *
from GUI2 import *
from f1 import *
from PIL import ImageTk,Image

#Creating the first display screen
def lgn_disp1():
    window0=Tk()
    window0.title("XYZ HOSPITAL")


    image2=ImageTk.PhotoImage(Image.open('FINALLOGIN.png'))


    img_1=ImageTk.PhotoImage(Image.open('button.png'))


    lbc1=Canvas(window0,width=1350,height=720,bg="RoyalBlue3")
    lbc1.grid_propagate(0)
    lbc1.create_image(0,0 ,anchor=NW,image=image2)
    lbc1.grid(column=0,row=0)



    def clik():
        lgn_disp(window0)

    frame11=LabelFrame(lbc1,width=950,height=250)
    #frame11.grid(column=4,row=4,padx=175,columnspan=3)
    #frame11.grid_propagate(0)



    btn0 = Button(lbc1, image=img_1, command=clik,borderwidth=0)
    btn0.grid(column=5,row=10,pady=565,padx=780)

    btn01 = Button(lbc1, text="QUIT", command=window0.quit,width=30,height=2)
    #btn01.grid(column=5,row=11,pady=2)

    window0.geometry('1350x750')
    window0.mainloop()

#Login Screen
def lgn_disp(prev):
    window = Toplevel(prev)
    window.title("XYZ HOSPITAL")

    image1=ImageTk.PhotoImage(Image.open('dispp.png'))

    lbc=Canvas(window,width=400,height=200,bg="white")
    lbc.grid_propagate(0)
    lbc.create_image(0, 0 ,anchor=NW,image=image1)
    lbc.grid(column=0,row=0)

    lb1 = Label(lbc, text="enter credentials", font=("Arial Bold", 15),bg="White")
    lb1.grid(column=0, row=0)
```

```
txt1 = Entry(lbc, width=25)
txt1.grid(column=1, row=3,pady=2)

txt2 = Entry(lbc, width=25)
txt2.grid(column=1, row=5)

txt1.insert(0, "Administrator")
txt2.insert(0, "passwd")

x = (txt1.get(),txt2.get())

def check():
  try:
    int(txt2.get())
    if (int(txt2.get()),(txt1.get())) in select():
      nam=txt1.get()
      ID=int(txt2.get())
      doctor(nam,ID)
    elif (txt1.get(),txt2.get()) in search_all():
      patient(txt1.get(),txt2.get())
    else:
      txt1.delete(0,20)
      txt2.delete(0,20)
      txt1.insert(0, "Administrator")
      txt2.insert(0, "incorrect passwd")

  except ValueError:
        if txt1.get() == "guest":
          patient("guest")
        elif txt1.get() == "Administrator" and txt2.get() == "passwd":
          post_lg("Administrator..",prev)
        elif (txt1.get(),txt2.get()) in search_all():
          patient(txt1.get())
        else:
          txt1.delete(0,20)
          txt2.delete(0,20)
          txt1.insert(0, "Administrator")
          txt2.insert(0, "incorrect passwd")

   btn = Button(lbc, text="submit", command=check,width=10)
   btn.grid(column=1,row=8,pady=2)

   window.config(bg="White")
   window.geometry('350x150')
   window.mainloop()

#Administrator Mode Options Menu
def post_lg(x,win):
   window1 = Toplevel(win)
   window1.title("depatments")

   image2=ImageTk.PhotoImage(Image.open('dispp.png'))

   lbc1=Canvas(window1,width=400,height=200,bg="white")
```

```
  lbc1.grid_propagate(0)
  lbc1.create_image(0, 0,anchor=NW,image=image2)
  lbc1.grid(column=0,row=0)



  lb3 = Label(lbc1, text="enter your requirement", font=("Arial Bold", 10),bg="white")
  lb3.grid(column=0, row=2)

  lb4 = Label(lbc1, text="welcome "+x, font=("Segoe Print", 15),bg="white")
  lb4.grid(column=0, row=0)

  click = StringVar(window1)
  click.set("accounts")

  drop = OptionMenu(lbc1,click, "accounts", "appointments", "employee data",
"pharmacy","cashier")
  drop.config(width=11)
  drop.grid(column=1, row=2)

  def select():
    if click.get() == "accounts":
      acc(win)
    if click.get() == "appointments":
      pre_appoint()
    if click.get() == "employee data":
      empl(win)
    if click.get() == "pharmacy":
      pharm(win)
    if click.get() == "cashier":
      cashier()

  b1 = Button(lbc1, text="select", command=select,width=14)
  b1.grid(column=1, row=4)

  window1.config(bg="white")
  window1.geometry('415x150')
  window1.mainloop()
```

File#3 GUI2

```
from appointmanager1 import *
from appointmanager2 import *
from tkinter import *
from f1 import *
from f3 import *
from f2 import *
from f4 import *
from f5 import *
from PIL import ImageTk, Image
from tkinter import messagebox
from GUI3 import *
import tkinter.font as TkFont
from Acc_Link import *

def acc(win):
    window3 = Toplevel(win)
    window3.title("accounts")

    image4=ImageTk.PhotoImage(Image.open('bg.png'))

    lbc5=Canvas(window3,width=920,height=600,bg="white")
    lbc5.create_image(0, 0 ,anchor=NW,image=image4)
    lbc5.grid(column=0,row=0)
    lbc5.grid_propagate(0)

    my_font=TkFont.Font(window3,family="Monaco",size=10)
```

```
framea2=LabelFrame(lbc5,width=440,height=290)
framea2.config(bg="white")
framea2.grid(column=2,row=2,padx=20,pady=20)
framea2.grid_propagate(0)

framea3=LabelFrame(lbc5,width=350,height=290)
framea3.grid(column=3,row=2)
framea3.config(bg="white")
framea3.grid_propagate(0)

image3=ImageTk.PhotoImage(Image.open('images01.png'))

lbc3=Canvas(framea2,width=500,height=400,bg="white")
lbc3.create_image(0, 0 ,anchor=NW,image=image3)
lbc3.grid(column=0,row=0)
lbc3.grid_propagate(0)



lbc4=Canvas(framea3,width=400,height=400,bg="white")


lbc4.create_image(0, 0 ,anchor=NW,image=image3)
lbc4.grid(column=0,row=0)
lbc4.grid_propagate(0)



lb3 = Label(lbc3, text="select the account", font=("Arial Bold", 15),bg="white")
```

```
lb3.grid(column=2, row=2)

lba3 = Label(lbc4, text="updating account", font=("Arial Bold", 15),bg="white")
lba3.grid(column=2, row=2)

lb4 = Label(lbc5, text="Accounts Department", font=("Segoe Print", 25),bg="snow")
lb4.grid(column=2, row=0,columnspan=3)



txt4 = Entry(lbc3, width=30)
txt4.grid(column=2, row=4)

txt5 = Entry(lbc3, width=30)
txt5.grid(column=2, row=6,padx=3,pady=3)

txt6 = Entry(lbc4, width=30)
txt6.grid(column=2, row=12, padx=5, pady=5)

teexta = Entry(lbc4, width=30)
teexta.grid(column=2, row=4)

lb5 = Label(lbc3, text="account no.",bg="white")
lb5.grid(column=1, row=4)

lba = Label(lbc4, text="account no.",bg="white")
lba.grid(column=1, row=4,padx=5,pady=5)
```

```
click1 = StringVar(window3)
click1.set("col. value")

drop1 = OptionMenu(lbc4, click1,"acc_due", "acc_paid")
drop1.config(width=10)
drop1.grid(column=1, row=12)

lbx = Label(lbc3, text="new account ??", font=("Arial Bold", 10),bg="white")
lbx.grid(column=1, row=22)

lb6 = Label(lbc3, text="name search",bg="white")
lb6.grid(column=1, row=6)

lb7 = Label(lbc4, text="enter the value to update", font=("Arial Bold", 10),bg="white")
lb7.grid(column=2, row=10)

lbf1 = Label(lbc3, text="search results:-", font=("Arial Bold", 10),bg="white")
lbf1.grid(column=1, row=18)

frame=LabelFrame(lbc3,width=350,height=50)
frame.grid(column=2,row=18)

scroll=Scrollbar(frame,orient=VERTICAL)
listb=Listbox(frame,height=5,width=38,font=my_font,yscrollcommand=scroll.set)

scroll.config(command=listb.yview)
scroll.pack(side=RIGHT,fill=Y)
```

```
listb.pack()


gap="|"
lo=["acno.","name","Paid","Due"]
listb.insert(END,f"{lo[0]:5}{gap}{lo[1]:>12}{gap}{lo[2]:>8}{gap}{lo[3]:>8}")
listb.insert(END,'----------------------------------------------')


def search1():
    nonlocal listb
    if txt4.get()!="":
        l=search01(txt4.get())
        listb.insert(END,l)
    elif txt5.get()!="":
        l=search02(txt5.get())
        for i in l:
            listb.insert(END,i)


def update01():
    if click1.get()=="acc_due":
        up_total(teexta.get(),txt6.get())
    if click1.get()=="acc_paid":
        up_paid(teexta.get(),txt6.get())
def clear():
    listb.delete(2,END)


b1 = Button(lbc3, text="select",command=search1, width=15)
```

```
    b1.grid(column=2, row=8,padx=10,pady=10)

    bu1 = Button(lbc4, text="update(acc_no. only)",command=update01, width=15)
    bu1.grid(column=2, row=16)

    by = Button(lbc3, text="sign in", command=new_ac, width=15)
    by.grid(column=2, row=22)

    bcl = Button(lbc3, text="Clear", command=clear, width=15)
    bcl.grid(column=2, row=20,pady=5)

    window3.config(bg="white")
    window3.geometry('845x390')
    window3.mainloop()


def pharm(win):
    window4 = Toplevel(win)
    window4.title("pharmacy")

    my_font=TkFont.Font(window4,family="Monaco",size=10)

    image4=ImageTk.PhotoImage(Image.open('newph.png'))

    lbc6=Canvas(window4,width=600,height=600,bg="white")
    lbc6.create_image(0, 0 ,anchor=NW,image=image4)
    lbc6.grid(column=0,row=0)
```

```
lbc6.grid_propagate(0)



lba8 = Label(lbc6, text="Pharmacy database", font=("Arial Bold", 15),bg="white")
lba8.grid(column=2, row=2)

lb8 = Label(lbc6, text="Pharmacy Logs directory", font=("Segoe Print", 20),bg="white")
lb8.grid(column=2, row=0)

txt7 = Entry(lbc6, width=30)
txt7.grid(column=2, row=4)

txt8 = Entry(lbc6, width=30)
txt8.grid(column=2, row=6)

txt9 = Entry(lbc6, width=30)
txt9.grid(column=2, row=10, padx=5, pady=5)

lb9 = Label(lbc6, text="item identity no.",bg="white")
lb9.grid(column=1, row=4)

click2 = StringVar(lbc6)
click2.set("Quantity")

drop2 = OptionMenu(lbc6, click2,"1","2","3","4","5","6","7","8")
drop2.config(width=7)
drop2.grid(column=1, row=10)
```

```
lb10 = Label(lbc6, text="name search",bg="white")
lb10.grid(column=1, row=6)

lb11 = Label(lbc6, text="Enter the Acc. no.", font=("Arial Bold", 10),bg="white")
lb11.grid(column=2, row=9)

lbf1 = Label(lbc6, text="search results:-", font=("Arial Bold", 10),bg="white")
lbf1.grid(column=1, row=7)

lbf = Label(lbc6)
lbf.grid(column=2, row=13)

frameph=LabelFrame(lbc6,width=350,height=50)
frameph.grid(column=2,row=7)

scrollph=Scrollbar(frameph,orient=VERTICAL)
listbph=Listbox(frameph,height=5,width=50,font=my_font,yscrollcommand=scrollph.set)

scrollph.config(command=listbph.yview)
scrollph.pack(side=RIGHT,fill=Y)
listbph.pack()

gap="|"
lo=["ID","Name","Price"]
listbph.insert(END,f"{lo[0]:>5}{gap}{lo[1]:>30}{gap}{lo[2]:>10}")
listbph.insert(END,'------------------------------------------------------------')
```

```
def searchp():
    nonlocal listbph
    listbph.delete(2,END)
    if txt7.get()!="":
        l=searchp1(txt7.get())
        for i in l:
            listbph.insert(END,i)

    elif txt8.get()!="":
        l=searchp2(txt8.get())
        for i in l:
            listbph.insert(END,i)
def BILLP():
    try:
        l=search_alltr()
        flag=0
        for i in l:
            if int(txt9.get())==i[1]:
                flag=1
                break
        if flag==0:
            txt9.delete(first=0,last=25)
            txt9.insert(0,"ACCOUNT NOT PRESENT")
            return None
    except:
            txt9.delete(first=0,last=25)
            txt9.insert(0,"IMPROPER ACCOUNT")
            return None

        l=listbph.get(ANCHOR)
        newl=l.split("|")

        if check_qua(newl[0].lstrip(),int(click2.get())):
            up_totalIn2(int(txt9.get()),int(newl[2].lstrip())*int(click2.get()))
            red_qua(newl[0].lstrip(),click2.get())

insert_TR(int(txt9.get()),newl[1].lstrip(),click2.get(),int(newl[2].lstrip())*int(click2.get()),"D"
)
        else:
            txt9.delete(first=0,last=20)
            txt9.insert(0,"QTY UNAVILABLE")

    def newmed():
        up_med()

    b2 = Button(lbc6, text="Search",command=searchp,width=15)
    b2.grid(column=2, row=8)

    b3 = Button(lbc6, text="Add to Acc",command=BILLP,width=15)
    b3.grid(column=2, row=13,pady=7)

    b4 = Button(lbc6, text="New Med",command=newmed,width=15)
    b4.grid(column=2, row=16,pady=7)

    window4.geometry('600x400')
    window4.mainloop()


def empl(win):
    window5 = Toplevel(win)
    window5.title("employee data")

    my_font=TkFont.Font(window5,family="Monaco",size=10)

    framee2=LabelFrame(window5,width=550,height=310)
    framee2.grid(column=2,row=2,padx=20,pady=20)
    framee2.grid_propagate(0)

    framee3=LabelFrame(window5,width=400,height=310)
    framee3.grid(column=3,row=2)
    framee3.grid_propagate(0)

    lb12 = Label(framee2, text="select the employee", font=("Arial Bold", 15))
    lb12.grid(column=2, row=2)

    lb13 = Label(window5, text="Employees Data..", font=("Segoe Print", 25))
    lb13.grid(column=2, row=0,columnspan=3)

    lbe13 = Label(framee3, text="Update Record", font=("Arial Bold", 15))
```

```
lbe13.grid(column=2, row=4)

txt10 = Entry(framee2, width=30)
txt10.grid(column=2, row=4)

txt11 = Entry(framee2, width=30)
txt11.grid(column=2, row=6)

txt12 = Entry(framee3, width=30)
txt12.grid(column=2, row=12)

txte12 = Entry(framee3, width=30)
txte12.grid(column=2, row=8)

lb14 = Label(framee2, text="doctor ID")
lb14.grid(column=0, row=4)

click3 = StringVar(window5)
click3.set("column")

drop3 = OptionMenu(framee3, click3,"salary","department")
drop3.config(width=10)
drop3.grid(column=0, row=12)

lb15 = Label(framee2, text="Name Search")
lb15.grid(column=0, row=6)

lbe15 = Label(framee3, text="Doctor Id")
```

```
lbe15.grid(column=0, row=8)

lb16 = Label(framee3, text="enter the value", font=("Arial Bold", 10))
lb16.grid(column=2, row=10)

lb17 = Label(framee2, text="new employee ??", font=("Arial Bold", 10))
lb17.grid(column=2, row=20)

lbf1 = Label(framee2, text="search results:-", font=("Arial Bold", 10))
lbf1.grid(column=0, row=18)

lbf = Label(framee2)
lbf.grid(column=2, row=21)

framee=LabelFrame(framee2,width=300,height=50)
framee.grid(column=2,row=18)

scrolle=Scrollbar(framee,orient=VERTICAL)
listbe=Listbox(framee,height=5,width=52,font=my_font,yscrollcommand=scrolle.set)

scrolle.config(command=listbe.yview)
scrolle.pack(side=RIGHT,fill=Y)
listbe.pack()

gap="|"
le=["Dc_Id","Name","Salary","Department"]
listbe.insert(END,f"{le[0]:5}{gap}{le[1]:>15}{gap}{le[2]:>10}{gap}{le[3]:>15}")
listbe.insert(END,'-----------------------------------------------')


def searche():
    nonlocal listbe
    if txt10.get()!="":
        l=searche1(txt10.get())
        listbe.insert(END,l)
    elif txt11.get()!="":
        l=searche2(txt11.get())
        for i in l:
            listbe.insert(END,i)


def update02():
    if click3.get()=="department":
        up_feild(int(txte12.get()),txt12.get())
    if click3.get()=="salary":
        up_salary(int(txte12.get()),txt12.get())


def cleare():
    listbe.delete(2,END)




b3 = Button(framee2, text="select",command=searche,width=15)
b3.grid(column=2, row=8,pady=10)


bu2 = Button(framee3, text="update(acc no.)",command=update02, width=15)
bu2.grid(column=2, row=14)
```

```
    b4 = Button(framee2, text="sign in",command=new_empl,width=15)
    b4.grid(column=2, row=21,pady=10)

    bcle = Button(framee2, text="Clear", command=cleare, width=15)
    bcle.grid(column=2, row=19,pady=5)

    window5.geometry('1000x400')
    window5.mainloop()

def pre_appoint():
    windowx1 = Toplevel()
    windowx1.title("appointment")

    image4=ImageTk.PhotoImage(Image.open('dispp.png'))

    lbo5=Canvas(windowx1,width=920,height=600,bg="white")
    lbo5.create_image(0, 0 ,anchor=NW,image=image4)
    lbo5.grid(column=0,row=0)
    lbo5.grid_propagate(0)

    lbx3 = Label(lbo5, text="enter your requirement", font=("Arial Bold", 10))
    lbx3.grid(column=0, row=2)

    lbx4 = Label(lbo5, text="The Appointment Manager", font=("Segoe Print", 15))
    lbx4.grid(column=0, row=0)

    clickx1 = StringVar(windowx1)
    clickx1.set("Create")

    dropx = OptionMenu(lbo5, clickx1, "Create", "Search", "Delete", "Update")
    dropx.config(width=11)
    dropx.grid(column=1, row=2)

    def selectx1():
        if clickx1.get() == "Create":
            appointcr()
        if clickx1.get() == "Delete":
            appointde()
        if clickx1.get() == "Search":
            appointse()
        if clickx1.get() == "Update":
            appointup()

    bx1 = Button(lbo5, text="Select", command=selectx1,width=14)
    bx1.grid(column=1, row=4)

    windowx1.geometry('420x150')
    windowx1.mainloop()

def patient(nam,ID):
    window7 = Toplevel()
    window7.title("patient")

    my_font=TkFont.Font(window7,family="Monaco",size=10)
```

```
framep1=LabelFrame(window7,width=500,height=250,padx=5,pady=5,bg='white')
framep1.grid(column=0,row=2,padx=10,pady=10)
framep1.grid_propagate(0)

lbp1 = Label(window7, text=" ",bg='white')
lbp1.grid(column=1, row=2)

framep2=LabelFrame(window7,width=500,height=250,bg='white')
framep2.grid(column=2,row=2)

framep3=LabelFrame(window7,width=500,height=250,bg='white')
framep3.grid(column=0,row=4)

framep4=LabelFrame(window7,width=500,height=250,bg='white')
framep4.grid(column=2,row=4)

framep2.grid_propagate(0)
framep3.grid_propagate(0)
framep4.grid_propagate(0)

lbpnam = Label(window7,bg='white', text=("welcome to the portal, "+nam),font=("Arial
Bold",13))
lbpnam.grid(column=0, row=1)

lbp2 = Label(window7,bg='white',text="Patients....",font=("Segoe Print",25))
lbp2.grid(column=0, row=0)
```

```
   lbp3 = Label(window7, text="Management....",font=("Segoe Print",25 ),bg='white')
   lbp3.grid(column=2, row=0)

   lbp4 = Label(framep1, text="Appointments",font=("Arial Bold",15),bg='white')
   lbp4.grid(column=0, row=0)

   lbp5 = Label(framep2, text="Accounts",font=("Arial Bold",15),bg='white')
   lbp5.grid(column=0, row=0)

   lbp6 = Label(framep3, text="Prescriptions",font=("Arial Bold",15),bg='white')
   lbp6.grid(column=0, row=0)

   lbp7 = Label(framep4, text="Doctors",font=("Arial Bold",15),bg='white')
   lbp7.grid(column=0, row=0)

# Accounts Frame

   framep_acc41=LabelFrame(framep2,width=250,height=50)
   framep_acc41.grid(column=2,row=4)
   framep_acc41.grid_propagate(0)

   scrollp_acc=Scrollbar(framep_acc41,orient=VERTICAL)

listbp_acc=Listbox(framep_acc41,height=5,width=40,font=my_font,yscrollcommand=scrollp
_acc.set)

   scrollp_acc.config(command=listbp_acc.yview)
   scrollp_acc.pack(side=RIGHT,fill=Y)
```

```
  listbp_acc.pack()
  gap="|"
  le=["PC_ID","NAME","DUE","PAID"]

  listbp_acc.insert(END,f"{le[0]:>5}{gap}{le[1]:>12}{gap}{le[2]:>8}{gap}{le[3]:>8}")
  listbp_acc.insert(END,'-----------------------------------------------')

  if searchLink(nam):
    for i in searchLink(nam):
      l=search02(i)
      listbp_acc.insert(END,l)
  else:
    l=search02(nam)
    listbp_acc.insert(END,l)



  def LINK(nam):
    linkL(nam)

  bp2_acc = Button(framep2, text="Link acc",width=15,command=lambda: LINK(nam))
  bp2_acc.grid(column=2,row=6,pady=10)


# Doctors Frame

  lbpa4 = Label(framep4, text="name search",bg='white')
```

```
lbpa4.grid(column=0, row=2)

txta4 = Entry(framep4, width=30)
txta4.grid(column=2, row=2,pady=10)

lbf4 = Label(framep4, text="search results:-", font=("Arial Bold", 10),bg='white')
lbf4.grid(column=0, row=4)

framep41=LabelFrame(framep4,width=250,height=50)
framep41.grid(column=2,row=4)
framep41.grid_propagate(0)

scrollp=Scrollbar(framep41,orient=VERTICAL)
listbp=Listbox(framep41,height=5,width=40,font=my_font,yscrollcommand=scrollp.set)

scrollp.config(command=listbp.yview)
scrollp.pack(side=RIGHT,fill=Y)

listbp.pack()
gap="|"
le=["Dc_Id","Name","Department"]

listbp.insert(END,f"{le[0]:>5}{gap}{le[1]:>15}{gap}{le[2]:>15}")
listbp.insert(END,'------------------------------------------------')

def searchpat2():
   nonlocal listbp
```

```
    if txta4.get()!="":
      l=searche4(txta4.get())
      listbp.insert(END,l)

  bp2 = Button(framep4, text="Search",width=15,command=searchpat2)
  bp2.grid(column=2,row=6,pady=10)

# Appointments Frame

  txtp13 = Entry(framep1, width=30)
  txtp13.grid(column=1, row=4,pady=10)

  lbxp10 = Label(framep1, text="patient id",bg='white')
  lbxp10.grid(column=0, row=4)



  framep11=LabelFrame(framep1,width=250,height=50)
  framep11.grid(column=0,row=6,columnspan=3)
  framep11.grid_propagate(0)

  scrollp01=Scrollbar(framep11,orient=VERTICAL)

listbp01=Listbox(framep11,height=5,width=55,font=my_font,yscrollcommand=scrollp01.set)

  scrollp01.config(command=listbp01.yview)
  scrollp01.pack(side=RIGHT,fill=Y)
  listbp01.pack()
```

```
gap=" | "
lo=["Name","Doctor","Date","Time"]
listbp01.insert(END,f"{lo[0]:>10}{gap}{lo[1]:>10}{gap}{lo[2]:>10}{gap}{lo[3]:>13}")
listbp01.insert(END,'------------------------------------------------------')

def choosep01():
    l=searcha(txtp13.get())
    listbp01.insert(END,l)
def clearap01():
    listbp01.delete(2,END)



bp11 = Button(framep1, text="search",command=choosep01,width=10)
bp11.grid(column=1,row=8,pady=20)

#Prescripts

labell=Label(framep3,text="Choose prescripts",font=("Arial Bold",10),bg='white')
labell.grid(column=0,row=1)

txtpr13 = Entry(framep3, width=30)
txtpr13.grid(column=2, row=2,pady=5)

txtpr14 = Entry(framep3, width=30)
txtpr14.grid(column=2, row=3,pady=5)

labell2=Label(framep3,text="Date:-",font=("Arial Bold",10),bg='white')
```

```
    labell2.grid(column=0,row=2)

    labell1=Label(framep3,text="Doctor:-",font=("Arial Bold",10),bg='white')
    labell1.grid(column=0,row=3)

    def prescript():
        prescrip_open_pat(txtpr14.get(),ID,txtpr13.get())

    bpr11 = Button(framep3, text="Search",command=prescript)
    bpr11.grid(column=2,row=8,pady=10)

    window7.config(bg='white')
    window7.geometry('1000x600')
    window7.mainloop()

def doctor(nam,ID):
    windowd=Toplevel()
    windowd.title("Doctors")

    imagec4=ImageTk.PhotoImage(Image.open('docc.png'))
    lbld5=Canvas(windowd,width=1350,height=800,bg="springgreen1")
    lbld5.grid_propagate(0)

    lbld5.create_image(0, 0 ,anchor=NW,image=imagec4)
    lbld5.grid(column=0,row=0)

    lbld=Label(lbld5,text="",font=("Segoe Print",20),bg="springgreen1",fg="white")
```

```
lbld.grid(column=10,row=0,columnspan=4,pady=20)

framed1=LabelFrame(lbld5,width=500,height=150)
framed1.grid(column=4,row=1,padx=10,columnspan=2,pady=10)
framed1.grid_propagate(0)

framed2=LabelFrame(lbld5,width=500,height=305)
framed2.grid(column=4,row=2,padx=10,columnspan=2,pady=10)
framed2.grid_propagate(0)

lbld1=Label(framed1,text="PRESCRIPTION WRITER",font=("Arial Bold",10))
lbld1.grid(column=3,row=1,columnspan=2)

lblc1=Label(framed1,text="ACCOUNT NO.    ",font=("Arial Bold",10))
lblc1.grid(column=2,row=2,padx=10)

lblc01=Label(framed1,text="Date(open)    ",font=("Arial Bold",10))
lblc01.grid(column=2,row=3,padx=10)

lblc5=Label(framed2,text="CHANGE TIME AVAIL",font=("Arial Bold",10))
lblc5.grid(column=4,row=3,columnspan=3)

lblc2=Label(framed2,text="LEAVE TIME:-      ",font=("Arial Bold",10))
lblc2.grid(column=3,row=4)

lblc3=Label(framed2,text="ARRIVE TIME:-     ",font=("Arial Bold",10))
lblc3.grid(column=3,row=5)
```

```
lblc6=Label(framed2,text="DATE:-     ",font=("Arial Bold",10))
lblc6.grid(column=3,row=6)

lblc4=Label(lbld5,text="THANK YOU",font=("Arial Bold",15))
lblc4.grid(column=4,row=12)

txtd1 = Entry(framed1, width=40)
txtd1.grid(column=3, row=2,pady=10,columnspan=2)
txtd1.insert(0,"100xx")

txtd01 = Entry(framed1, width=40)
txtd01.grid(column=3, row=3,pady=10,columnspan=2)
txtd01.insert(0,"DD-MM-YYYY")

txtd2 = Entry(framed2, width=30)
txtd2.grid(column=4, row=4,pady=10,columnspan=3)

txtd3 = Entry(framed2, width=30)
txtd3.grid(column=4, row=5,pady=10,columnspan=3)

txtd4 = Entry(framed2, width=30)
txtd4.grid(column=4, row=6,pady=10,columnspan=3)



my_font=TkFont.Font(lbld5,family="Monaco",size=10)
```

```
  framed=LabelFrame(lbld5,width=600,height=720)
  framed.grid(column=0,row=1,rowspan=20,columnspan=2,padx=10)
  framed.grid_propagate(0)

  scrolld=Scrollbar(framed,orient=VERTICAL)
  listbd=Listbox(framed,height=32,width=80,font=my_font,yscrollcommand=scrolld.set)

  scrolld.config(command=listbd.yview)
  scrolld.pack(side=RIGHT,fill=Y)

  listbd.insert(END,"---------------------------------TODAY'S
APPOINTMENTS--------------------------------")
  listbd.pack()



  def presc_win(x,y):
    try:
      lm=search_alltr()
      flag=0
      for i in lm:
        if int(txtd1.get())==i[1]:
          flag=1
          break
      if flag==0:
        txtd1.delete(first=0,last=25)
        txtd1.insert(0,"ACCOUNT NOT PRESENT")
        return None
    except:
        txtd1.delete(first=0,last=25)
        txtd1.insert(0,"IMPROPER ACCOUNT")
        return None

    prescripwr(y,x)

def appoint_disp(k):
    n=searchappointments(ID)
    for i in n:
        listbd.insert(END,i)

def appoint_change(k,l,n):
    write_dcc(k,l,n)

def presc_op(ID,nam):

    try:
        lm=search_alltr()
        flag=0
        for i in lm:
            if int(txtd1.get())==i[1]:
                flag=1
                break
        if flag==0:
            txtd1.delete(first=0,last=25)
            txtd1.insert(0,"ACCOUNT NOT PRESENT")
            return None
        except:
          txtd1.delete(first=0,last=25)
          txtd1.insert(0,"IMPROPER ACCOUNT")
          return None

        try:
          split=txtd01.get().split("-")
          split[2]
          if len(split)!=3:
            return None
        except:
          txtd01.delete(0,30)
          txtd01.insert(0,"WRONG FORMAT")
          return None

        prescripop(nam,ID,txtd01.get())

    bpc1 = Button(framed1, text="Create Prescript",command=lambda:
presc_win(txtd1.get(),nam),width=15)
    bpc1.grid(column=3,row=4,pady=10)

    bpc4 = Button(framed1, text="Open Prescript",command=lambda:
presc_op(txtd1.get(),nam),width=15)
    bpc4.grid(column=4,row=4,pady=10)
    bpc2 = Button(lbld5, text="Appointments",command=lambda:
appoint_disp(ID),width=15)
    bpc2.grid(column=1,row=22,pady=10,columnspan=1)

    bpc3 = Button(framed2, text="CHANGE",command=lambda:
appoint_change(ID,txtd3.get()+" - "+txtd2.get(),txtd4.get()),width=15)
    bpc3.grid(column=4,row=7,pady=10,columnspan=3)




    windowd.geometry("1350x750")
    windowd.mainloop()

def cashier():
    windowc=Toplevel()
    windowc.title("Cashier")

    imagec4=ImageTk.PhotoImage(Image.open('cashie.png'))
    lblc5=Canvas(windowc,width=920,height=600,bg="violetred1")

    lblc5.create_image(0, 0 ,anchor=NW,image=imagec4)
    lblc5.grid(column=0,row=0)



    lblc=Label(lblc5,font=("Segoe Print",20),bg="orange",fg="white")
    lblc.grid(column=10,row=0,columnspan=6,pady=20)

    framec=LabelFrame(lblc5,width=600,height=720)
```

```
framec.grid(column=0,row=2,rowspan=20,padx=10)
framec.grid_propagate(0)

lblc1=Label(lblc5,text="ACCOUNT No.",font=("Arial Bold",10))
lblc1.grid(column=3,row=2,pady=20)

lblc2=Label(lblc5,text="PAYMENT Method",font=("Arial Bold",10))
lblc2.grid(column=3,row=4)

lblc3=Label(lblc5,text="AMOUNT RECIEVED",font=("Arial Bold",10))
lblc3.grid(column=3,row=5)

lblc4=Label(lblc5,text="THANK YOU",font=("Arial Bold",15))
lblc4.grid(column=4,row=12)

txtc1 = Entry(lblc5, width=30)
txtc1.grid(column=4, row=2,pady=10)

txtc2 = Entry(lblc5, width=30)
txtc2.grid(column=4, row=4,pady=10)

txtc3 = Entry(lblc5, width=30)
txtc3.grid(column=4, row=5,pady=10)

framec1=LabelFrame(lblc5,width=600,height=720)
framec1.grid(column=3,row=7,rowspan=5,columnspan=3)
framec1.grid_propagate(0)
```

```
def displayc():
    listbc.delete(2,END)
    listbc1.delete(2,END)

    try:
        lm=search_alltr()
        flag=0
        for i in lm:
            if int(txtc1.get())==i[1]:
                flag=1
                break
        if flag==0:
            txtc1.delete(first=0,last=25)
            txtc1.insert(0,"ACCOUNT NOT PRESENT")
            return None
    except:
        txtc1.delete(first=0,last=25)
        txtc1.insert(0,"IMPROPER ACCOUNT")
        return None

    l=searchTR02(txtc1.get())



    for i in l[0]:
        listbc.insert(END,i)
    for i in l[1]:
        listbc1.insert(END,i)


def payc():
    nonlocal txtc3
    try:
        int(txtc3.get())
    except:
        txtc3.delete(first=0,last=25)
        txtc3.insert(0,"IMPROPER INPUT")
        return None

    if int(txtc3.get())>=int(listbc.get(ANCHOR).split('|')[4].lstrip()):
        l=int(txtc3.get())-int(listbc.get(ANCHOR).split('|')[4].lstrip())


        txtc3.delete(first=0,last=20)
        txtc3.insert(0,str(l))


        up_paidIn(int(listbc.get(ANCHOR).split('|')[4].lstrip()),txtc1.get())
        up_paidTR(listbc.get(ANCHOR)[1:5])
        listbc.delete(ANCHOR)
        displayc()

bpc1 = Button(lblc5, text="Search",command=displayc,width=10)
bpc1.grid(column=4,row=3,pady=10)
```

```
bpc2 = Button(lblc5, text="PAY",command=payc,width=10)
bpc2.grid(column=4,row=6,pady=10)

my_font=TkFont.Font(lblc5,family="Monaco",size=10)

scrollc=Scrollbar(framec,orient=VERTICAL)
listbc=Listbox(framec,height=40,width=80,font=my_font,yscrollcommand=scrollc.set)

scrollc.config(command=listbc.yview)
scrollc.pack(side=RIGHT,fill=Y)

listbc.insert(END,"-----------------------------------INVOICE-----------------------------------")
listbc.pack()
gap="|"
i=("T_ID","ACCOUNT","DETAILS","QTY","PRICE")
listbc.insert(END,f"{i[0]:5}{gap}{i[1]:>18}{gap}{i[2]:^30}{gap}{i[3]:>5}{gap}{i[4]:>10}")

scrollc1=Scrollbar(framec1,orient=VERTICAL)
listbc1=Listbox(framec1,height=20,width=80,font=my_font,yscrollcommand=scrollc1.set)

scrollc1.config(command=listbc1.yview)
scrollc1.pack(side=RIGHT,fill=Y)

listbc1.insert(END,"-----------------------------PAYMENT HISTORY-----------------------------")
listbc1.pack()
listbc1.insert(END,f"{i[0]:5}{gap}{i[1]:>18}{gap}{i[2]:^30}{gap}{i[3]:>5}{gap}{i[4]:>10}")
```

```
windowc.geometry('1350x750')

windowc.mainloop()
```

File#4 GUI3

```
from appointmanager1 import *
from tkinter import *
from f1 import *
from f3 import *
from f2 import *
from f4 import *
from f5 import *
from PIL import ImageTk, Image
from tkinter import messagebox
import tkinter.font as TkFont
from timemanager import *
from Acc_Link import *
from datetime import *
from prescript import *

def new_ac():
    window8 = Toplevel()
    window8.title("accounts")

    lb1x0 = Label(window8, text="new account",font=("Arial Bold",15))
    lb1x0.grid(column=0, row=0)


    lb1x1 = Label(window8, text="name")
    lb1x1.grid(column=0, row=4)


    lb1x2 = Label(window8, text="ammount due")
    lb1x2.grid(column=0, row=6)
```

```
lb1x3 = Label(window8, text="amount paid")
lb1x3.grid(column=0, row=8)

txtx1 = Entry(window8, width=30)
txtx1.grid(column=2, row=4)

txtx2 = Entry(window8, width=30)
txtx2.grid(column=2, row=6)

txtx3 = Entry(window8, width=30)
txtx3.grid(column=2, row=8)

txtx1.insert(0, "name")
txtx2.insert(0, "10000")
txtx3.insert(0, "10000")

def ins_ac():
    insert_acc(txtx1.get(),int(txtx3.get()),int(txtx2.get()))
    writepat(txtx1.get())
    insert_TR(convert(txtx1.get())[0][0],"ADMISSION","1","10000","D")
bx = Button(window8, text="submit", width=15, command=ins_ac)
bx.grid(column=2, row=10)

window8.geometry('320x200')
window8.mainloop()
```

```
def up_med():
    win_las=Toplevel()
    win_las.title("Update MEDS")

    lab1win0 = Label(win_las, text="New Medicine",font=("Arial Bold",15))
    lab1win0.grid(column=0, row=0)


    lab1win1 = Label(win_las, text="Name")
    lab1win1.grid(column=0, row=4)

    lab1win2 = Label(win_las, text="Supplier")
    lab1win2.grid(column=0, row=6)

    lab1win3 = Label(win_las, text="Quantity")
    lab1win3.grid(column=0, row=8)

    lab1win4 = Label(win_las, text="Price")
    lab1win4.grid(column=0, row=9)

    textx1 = Entry(win_las, width=30)
    textx1.grid(column=2, row=4)

    textx2 = Entry(win_las, width=30)
    textx2.grid(column=2, row=6)

    textx3 = Entry(win_las, width=30)
    textx3.grid(column=2, row=8)
```

```
textx4 = Entry(win_las, width=30)
textx4.grid(column=2, row=9)

lab1win6 = Label(win_las, text="Update Medicine",font=("Arial Bold",15))
lab1win6.grid(column=3, row=0)


lab1win7 = Label(win_las, text="ID")
lab1win7.grid(column=3, row=4)

textnx1 = Entry(win_las, width=30)
textnx1.grid(column=4, row=4)

textnx2 = Entry(win_las, width=30)
textnx2.grid(column=4, row=6)

click7 = StringVar(win_las)
click7.set("column")

drop1 = OptionMenu(win_las, click7,"PRICE", "QTY")
drop1.config(width=10)
drop1.grid(column=3, row=6)

def add_med():
  ins(textx1.get(),textx2.get(),textx4.get(),textx3.get())

def change_med():
  if click7.get()=="PRICE":
```

```
            up_medicine_price(textnx1.get(), textnx2.get())
        if click7.get()=="QTY":
            up_medicine_qty(textnx1.get(), textnx2.get())
    b2 = Button(win_las, text="Add",command=add_med,width=15)
    b2.grid(column=2, row=10,pady=8)

    b3 = Button(win_las, text="Update",command=change_med,width=15)
    b3.grid(column=4, row=8,pady=7)





    win_las.geometry('700x250')
    win_las.mainloop()
def new_empl():
    window9 = Toplevel()
    window9.title("employee")

    lab1x0 = Label(window9, text="new employee",font=("Arial Bold",15))
    lab1x0.grid(column=0, row=0)

    lab1x1 = Label(window9, text="name")
    lab1x1.grid(column=0, row=4)

    lab1x2 = Label(window9, text="medical lisc")
    lab1x2.grid(column=0, row=6)
```

```
lab1x3 = Label(window9, text="Arrival at clinic")
lab1x3.grid(column=0, row=8)

lab1x4 = Label(window9, text="Leave from clinic")
lab1x4.grid(column=0, row=9)

lab1x01 = Label(window9, text="Fees consult")
lab1x01.grid(column=0, row=10)

lab1x02 = Label(window9, text="salary")
lab1x02.grid(column=0, row=12)

lab1x03 = Label(window9, text="feild of study")
lab1x03.grid(column=0, row=14)

textx1 = Entry(window9, width=30)
textx1.grid(column=2, row=4)

textx2 = Entry(window9, width=30)
textx2.grid(column=2, row=6)

textx3 = Entry(window9, width=30)
textx3.grid(column=2, row=8)

textx4 = Entry(window9, width=30)
textx4.grid(column=2, row=9)

textx01 = Entry(window9, width=30)
```

```
    textx01.grid(column=2, row=10)

    textx02 = Entry(window9, width=30)
    textx02.grid(column=2, row=12)

    textx03 = Entry(window9, width=30)
    textx03.grid(column=2, row=14)

    textx1.insert(0, "name")
    textx2.insert(0, "A0123456xx")
    textx3.insert(0, "HH:mm")
    textx4.insert(0, "HH:mm")

    textx01.insert(0, "1000")
    textx02.insert(0, "100000")
    textx03.insert(0, "feild")

    bex = Button(window9, text="submit", width=15, command=lambda:
new_employee(textx1.get(),textx2.get(),textx3.get(),textx4.get(),int(textx01.get()),float(textx0
2.get()),textx03.get()))
    bex.grid(column=2, row=16)

    window9.geometry('450x300')
    window9.mainloop()

def appointde():
  windowx2 = Toplevel()
  windowx2.title("appointments")
```

```
image4=ImageTk.PhotoImage(Image.open('dispp.png'))

lbx5=Canvas(windowx2,width=920,height=600,bg="white")
lbx5.create_image(0, 0 ,anchor=NW,image=image4)
lbx5.grid(column=0,row=0)
lbx5.grid_propagate(0)

lbx18 = Label(lbx5, text="select the appointment", font=("Arial Bold", 15))
lbx18.grid(column=2, row=2)

lbx19 = Label(lbx5, text="Appointments...", font=("Segoe Print", 20))
lbx19.grid(column=2, row=0)

txtx13 = Entry(lbx5, width=30)
txtx13.grid(column=2, row=4)

lbx20 = Label(lbx5, text="appointment id")
lbx20.grid(column=1, row=4)

def choosex():
    deletea(int(txtx13.get()))

bx5 = Button(lbx5, text="delete",command=choosex)
bx5.grid(column=2, row=6)

windowx2.geometry('400x200')
```

```
    windowx2.mainloop()



def appointse():
    windowx3 = Toplevel()
    windowx3.title("appointments")
    my_font=TkFont.Font(windowx3,family="Monaco",size=10)

    image4=ImageTk.PhotoImage(Image.open('dispp.png'))

    lbk5=Canvas(windowx3,width=920,height=600,bg="white")
    lbk5.create_image(0, 0 ,anchor=NW,image=image4)
    lbk5.grid(column=0,row=0)
    lbk5.grid_propagate(0)

    lbx318 = Label(lbk5, text="select the appointment", font=("Arial Bold", 15))
    lbx318.grid(column=2, row=2)

    lbx319 = Label(lbk5, text="Appointments...", font=("Segoe Print", 20))
    lbx319.grid(column=2, row=0)

    txtx313 = Entry(lbk5, width=30)
    txtx313.grid(column=2, row=4)

    lbx320 = Label(lbk5, text="patient id")
    lbx320.grid(column=1, row=4)
    framea=LabelFrame(lbk5,width=350,height=50)
    framea.grid(column=2,row=18)
    scrolla=Scrollbar(framea,orient=VERTICAL)
    listba=Listbox(framea,height=5,width=55,font=my_font,yscrollcommand=scrolla.set)
    scrolla.config(command=listba.yview)
    scrolla.pack(side=RIGHT,fill=Y)
    listba.pack()
    gap=" | "
    lo=["Name","Doctor","Date","Time"]
    listba.insert(END,f"{lo[0]:>10}{gap}{lo[1]:>10}{gap}{lo[2]:>10}{gap}{lo[3]:>13}")
    listba.insert(END,'-----------------------------------------------------------')

    def choosex3():
        l=searcha(txtx313.get())
        listba.insert(END,l)
    def clearap():
        listba.delete(2,END)

    bx35 = Button(lbk5, text="search",command=choosex3,width=15)
    bx35.grid(column=2, row=6,pady=5)
    box35 = Button(lbk5, text="clear",command=clearap,width=15)
    box35.grid(column=2, row=19,pady=5)

    windowx3.geometry('540x300')
    windowx3.mainloop()

def appointup():
```

```
windowx4 = Toplevel()
windowx4.title("appointments")

lbx418 = Label(windowx4, text="select the appointment", font=("Arial Bold", 15))
lbx418.grid(column=2, row=2)
lbx419 = Label(windowx4, text="Appointments...", font=("Segoe Print", 20))
lbx419.grid(column=2, row=0)
lbx420 = Label(windowx4, text="ID")
lbx420.grid(column=1, row=4)

txtx414 = Entry(windowx4, width=30)
txtx414.grid(column=2, row=4)

txtx413 = Entry(windowx4, width=30)
txtx413.grid(column=2, row=6)

clickx41 = StringVar(windowx4)
clickx41.set("time")

dropx4 = OptionMenu(windowx4, clickx41, "time", "status", "date")
dropx4.config(width=7)
dropx4.grid(column=1, row=6)

def choosex4():
    updatea(int(txtx414.get()),clickx41.get(),txtx413.get())

bx45 = Button(windowx4, text="update",command=choosex4)
```

```
    bx45.grid(column=2, row=8)


    windowx4.geometry('350x250')
    windowx4.mainloop()

def invoice(a,b,c,d,e):
    res=Toplevel()
    res.title("Record")


    la=Label(res,text="Appointment Summary",font=("Arial Bold",15))
    la.grid(column=0,row=0,columnspan=2)


    la1=Label(res,text="Name:--",font=("Arial Bold",10))
    la1.grid(column=0,row=1)

    la2=Label(res,text="Doctor:--",font=("Arial Bold",10))
    la2.grid(column=0,row=2)

    la3=Label(res,text="Time:--",font=("Arial Bold",10))
    la3.grid(column=0,row=3)

    la4=Label(res,text="Date:--",font=("Arial Bold",10))
    la4.grid(column=0,row=4)

    la5=Label(res,text="Price:--",font=("Arial Bold",10))
    la5.grid(column=0,row=5)
    la6=Label(res,text=a)
    la6.grid(column=1,row=1)

    la7=Label(res,text=b)
    la7.grid(column=1,row=2)

    la8=Label(res,text=c)
    la8.grid(column=1,row=3)

    la9=Label(res,text=d)
    la9.grid(column=1,row=4)

    la10=Label(res,text=e)
    la10.grid(column=1,row=5)

    res.geometry("250x150")
    res.mainloop()

def appointcr():
    window6 = Toplevel()
    window6.title("appointments")

    image4=ImageTk.PhotoImage(Image.open('dispp.png'))

    lbn5=Canvas(window6,width=920,height=600,bg="white")
    lbn5.create_image(0, 0 ,anchor=NW,image=image4)
    lbn5.grid(column=0,row=0)
```

```
lbn5.grid_propagate(0)

my_font=TkFont.Font(window6,family="Monaco",size=10)

lb18 = Label(lbn5, text="select the appointment", font=("Arial Bold", 15))
lb18.grid(column=2, row=2)

lb19 = Label(lbn5, text="Appointments...", font=("Segoe Print", 20))
lb19.grid(column=2, row=0)

txt13 = Entry(lbn5, width=30)
txt13.grid(column=2, row=4)

txt14 = Entry(lbn5, width=30)
txt14.grid(column=2, row=6)

txt30 = Entry(lbn5, width=30)
txt30.grid(column=2, row=10)

txt31 = Entry(lbn5, width=30)
txt31.grid(column=2, row=12)

lb20 = Label(lbn5, text="Doctor ID")
lb20.grid(column=1, row=4)

lb21 = Label(lbn5, text="Patient ID")
lb21.grid(column=1, row=6)
```

```
lb31 = Label(lbn5, text="enter the date")
lb31.grid(column=1, row=10)

lb32 = Label(lbn5, text="status")
lb32.grid(column=1, row=12)

lb33 = Label(lbn5, text="time")
lb33.grid(column=1, row=16)

frama=LabelFrame(lbn5,width=350,height=50)
frama.grid(column=2,row=16)

scrollap=Scrollbar(frama,orient=VERTICAL)
listbap=Listbox(frama,height=5,width=25,font=my_font,yscrollcommand=scrollap.set)

scrollap.config(command=listbap.yview)
scrollap.pack(side=RIGHT,fill=Y)
listbap.pack()

gap="|"
loap=["acno.","name","due","paid"]
listbap.insert(END,'-------------------------------------------------')

def choose():
    listbap.delete(1,END)
    d=check_avail(txt13.get(),txt30.get())
    k=appoint_time(txt30.get(),txt13.get(),d[0],d[1])
    for i in k:
        listbap.insert(END,i)

  def time_sel():
    x=listbap.get(ANCHOR)
    listbap.delete(ANCHOR)

    writea(txt14.get(),txt13.get(),x,txt30.get(),txt31.get())
    up_totalIn(int(txt13.get()),int(txt14.get()))

    insert_TR01(txt14.get(),"APPOINTMENT",txt13.get(),"D")
    l=invoiceg(int(txt13.get()),int(txt14.get()))

    invoice(l[2],l[0],x,txt30.get(),l[1])

  b5 = Button(lbn5, text="search time",command=choose,width=10)
  b5.grid(column=2, row=14,pady=5)

  bsel = Button(lbn5, text="Create",command=time_sel,width=10)
  bsel.grid(column=2, row=18,pady=5)

  window6.geometry('400x330')
  window6.mainloop()

def linkL(a):
  wind=Toplevel()
  wind.title("LINK")
```

```
    lblink0=Label(wind,text="LINK>>>",font=("Arial Bold",15))
    lblink0.grid(column=0,row=0)

    lblink1=Label(wind,text="Enter link name")
    lblink1.grid(column=0,row=4)

    txtlin1 = Entry(wind, width=30)
    txtlin1.grid(column=2, row=4)

    def my_Link():
        writeLink(a,txtlin1.get())

    blin = Button(wind, text="Link",command=my_Link,width=10)
    blin.grid(column=2, row=6,pady=5)

    wind.geometry('290x125')
    wind.mainloop()

def prescripwr(I,ID):
    root=Toplevel()
    root.title("prescription")

    m=str(datetime.now())[:10].split("-")

    DATE=""
    DATE=m[2]+"-"+m[1]+"-"+m[0]
    txttt=Text(root,height=35,width=80)
    txttt.grid(row=0,column=0)
    txttt.insert(END,"Date:- "+DATE+"\n")
    txttt.insert(END,"ID:- "+ID+"\n")
    txttt.insert(END,"Doctor:- "+I+"\n")

    def write_prescript(ID,DATE,I):
        presc_wr(ID,DATE,I,txttt.get(1.0,END))



    bt=Button(root,text="Submit",command=lambda: write_prescript(ID,DATE,I),width=10)
    bt.grid(row=1,column=0)

    root.geometry('630x601')
    root.mainloop()

def prescripop(I,ID,date):
    root1=Toplevel()
    root1.title("prescription")

    m=str(datetime.now())[:10].split("-")

    DATE=""
    DATE=m[2]+"-"+m[1]+"-"+m[0]

    txttt=Text(root1,height=35,width=80)
    txttt.grid(row=0,column=0)
    l=presc_se(ID,date,I)
    for i in l:
        txttt.insert(END,i)
    def save_prescript(ID,date,I):
        presc_wr(ID,date,I,txttt.get(1.0,END))

    bt=Button(root1,text="Save",command=lambda: save_prescript(ID,date,I),width=10)
    bt.grid(row=1,column=0)

    root1.geometry('630x601')
    root1.mainloop()

def prescrip_open_pat(I,ID,date):
    root1=Toplevel()
    root1.title("prescription")

    m=str(datetime.now())[:10].split("-")

    DATE=""
    DATE=m[2]+"-"+m[1]+"-"+m[0]

    txttt=Text(root1,height=35,width=80)
    txttt.grid(row=0,column=0)
    l=presc_se(ID,date,I)
    for i in l:
        txttt.insert(END,i)
    def save_prescript(ID,date,I):
```

```
    presc_wr(ID,date,I,txttt.get(1.0,END))


bt=Button(root1,text="Save",command=lambda: save_prescript(ID,date,I),width=10)
#bt.grid(row=1,column=0)


root1.geometry('630x601')
root1.mainloop()
```

File #5 appointmentmanager1

```
import csv
import datetime
import tkinter as tk
#importing the required modules
#Aliases for all the storage files
fn = 'appoint.csv'
dn = 'doctor.csv'
pn = 'patient.csv'
#Appointment Record Creator
def writea(b,c,d,e,g):
    l = []
    global fn
    with open(fn, 'a+') as f:


        csw = csv.writer(f, lineterminator='\n')
        f.seek(0)
        l = list(csv.reader(f))
        if len(l) == 0:
            a = 101
        else:
            a = int(l[-1][0]) + 1
        rec = [a,b,c,d,e,g]
        csw.writerow(rec)
        print("Appointment created sucessfully")

#Searching the Files
```

```
def reada():
   global fn
   global dn
   global pn
   rec = []
   hdr = 'AppId\tPatientId\tPatient Name\tDoctor Id\tDoctor
Name\tAppointDate\tAppointTime\tStatus'
   with open(fn, 'r') as f, open(dn, 'r') as d, open(pn, 'r') as p:
      csr = csv.reader(f)
      csrd = list(csv.reader(d))
      csrp = list(csv.reader(p))
      print(hdr)
      for i in csr:
         drnm = ''
         pnm = ''
         for j in csrp:
            if j[0] == i[1]:
               pnm = j[1]
               break
         for j in csrd:
            if j[0] == i[2]:
               drnm = j[1]
               break

         res=(i[0], '\t', i[1], '\t', pnm, '\t', i[2], '\t', drnm, '\t', i[3], '\t', i[4], '\t', i[5])
         print(res)
         return res
```

```
#Basiclly a single Search
def searcha(pid):
    global fn
    global dn
    global pn
    with open(fn, 'r') as f, open(dn, 'r') as d, open(pn, 'r') as p:
        csr = csv.reader(f)
        csrd = list(csv.reader(d))
        csrp = list(csv.reader(p))
        l=[]
        flag = False
        for i in csr:
         if i[1] == pid:
           flag = True
           l=i
           drnm = ''
           pnm = ''
           for j in csrp:
              if j[0] == i[1]:
                 pnm = j[1]
                 pnm0=pnm.split()
                 break
           for j in csrd:
              if j[0] == i[2]:
                 drnm = j[1]
                 drnm0=drnm.split()
```

```
                break



        if flag == False:
            res=('No Record')
            print (res)
            return res
        else:
            gap=" | "
            t="Dr. "
            drnm0[1]=t+drnm0[1]
            res=(f"{pnm0[0]:>10}{gap}{drnm0[1]:>10}{gap}{l[4]:>10}{gap}{l[3]:>13}")
            print(res)
            return res


#Updating Values
def updatea(ID,s,val):
    global fn
    with open(fn, 'r') as f:
        flag = False
        csr = csv.reader(f)
        reclst = list(csr)
        for i in range(len(reclst)):
            if int(reclst[i][0]) == ID:
                print(reclst[i])
                flag = True
            if s == 'time':
                reclst[i][3] = val
            elif s == 'date':
                reclst[i][4] = val
            elif s == 'status':
                reclst[i][5] = val

    with open(fn, 'w') as f:
        csw = csv.writer(f, lineterminator='\n')
        csw.writerows(reclst)

    if flag:
        print('Record updated')
    else:
        print('Record not found')


def deletea(ID):
    global fn
    with open(fn, 'r') as f:
        flag = False
        csr = csv.reader(f)
        reclst = list(csr)
        for i in range(len(reclst)):
            if int(reclst[i][0]) == ID:
                flag = True
                reclst.pop(i)
            break
    with open(fn, 'w') as f:
        csw = csv.writer(f, lineterminator='\n')
        csw.writerows(reclst)

    if flag:
        print('Record deleted')
    else:
        print('Record not found')

def searchtime(did):
    global fn
    global dn
    global pn
    with open(fn, 'r') as f, open(dn, 'r') as d, open(pn, 'r') as p:
        csr = csv.reader(f)
        csrd = list(csv.reader(d))
        csrp = list(csv.reader(p))
        l=[]
        k=[]
        flag = False
        for i in csr:
         if i[2] == did:
           flag = True
           l.append([i[3],i[4],i[5]])

    return l
```

```
def searchappointments(did):
   global fn
   global dn
   global pn
   m=str(datetime.datetime.now())[:10].split("-")
   da=""
   da=m[2]+"/"+m[1]+"/"+m[0]
   with open(fn, 'r') as f, open(dn, 'r') as d, open(pn, 'r') as p:
      csr = csv.reader(f)
      csrd = list(csv.reader(d))
      csrp = list(csv.reader(p))
      l=[]
      k=[]
      flag = False
      for i in csr:
       if i[2] == str(did) and i[4]==da:
         flag = True
         gap="|"
         l.append(f"{i[0]:>3}{gap}{i[1]:>5}{gap}{i[2]:>5}{gap}{i[3]:>15}{gap}{i[4]:>15}{gap}
{i[5]:>15}")

   return l
```

File #6 Appointmentmanager2

```
import csv
import datetime
import tkinter as tk

fn = 'appoint.csv'
dn = 'doctor.csv'
pn = 'patient.csv'
Tn = 'Time_avail.csv'

def writedoc(b):
  l = []
  global dn
  with open(dn, 'a+') as f:


    csw = csv.writer(f, lineterminator='\n')
    f.seek(0)


    l = list(csv.reader(f))
    if len(l) == 0:
      a = 90001
    else:
      a = int(l[-1][0]) + 1
    rec = [a,b]
    csw.writerow(rec)
  return a
```

```
def writepat(b):
    l = []
    global pn
    with open(pn, 'a+') as f:

        csw = csv.writer(f, lineterminator='\n')
        f.seek(0)

        l = list(csv.reader(f))
        if len(l) == 0:
            a = 10001
        else:
            a = int(l[-1][0]) + 1
        rec = [a,b]
        csw.writerow(rec)
    return a

def write_dcc(k,m,n):
    global Tn
    with open(Tn,'a+') as f:
        csw = csv.writer(f, lineterminator='\n')
        f.seek(0)

        l = list(csv.reader(f))
        rec = [k,m,n]
        csw.writerow(rec)
    return l


def read_dcc(k,n):
    global Tn
    with open(Tn,'r') as f:
        x=None
        l=list(csv.reader(f))
        for i in l:
            if i[0]==k and i[2]==n:
                x=i[1].split(" - ")
        return x
```

File #7 f1

```
from appointmanager2 import *
from tkinter import *
from sqlconnection import *

def new_employee(y, z, k, o, m, l, n):
   a=writedoc(y)
   cur.execute("insert into employee values(%s,%s,%s,%s,%s,%s,%s,%s)", (a, y, z, k, o, m, l,
n))
   con.commit()
   print("record saved successfully")


def gen(x, y, z):
   s1 = []
   cur.execute("select " + z + " from employee where " + y + "='%s'" % x)
   for l in cur:
      for i in l:
         s1.append(i)
   print(s1)


def select():
   cur.execute("select D_ID,NAME from employee")
   return cur.fetchall()
```

```
def searche1(x):
   cur = con.cursor()
   cur.execute("select D_ID,Name,salary,feild_of_practice from employee where
D_ID="+x)
   d=cur.fetchall()
   st=""
   gap="|"
   for i in d:
      st=f"{i[0]:>5}{gap}{i[1]:>15}{gap}{i[2]:>10}{gap}{i[3]:>15}"
   print(st)
   print("record updated successfully")
   k=StringVar()
   k.set(st)
   return st

def searche2(x):
   cur = con.cursor()
   cur.execute("select D_ID,Name,salary,feild_of_practice from employee where name =
'%s'" %x)
   d=cur.fetchall()
   l=[]
   for i in d:
      st1=""
      gap="|"
      st1=f"{i[0]:>5}{gap}{i[1]:>15}{gap}{i[2]:>10}{gap}{i[3]:>15}"
      l.append(st1)
      print(st1)
```

```
    k=StringVar()
    k.set(st1)
    return l

def searche3(x):
    cur = con.cursor()
    cur.execute("select D_ID,Name,feild_of_practice from employee where name = '%s'"
%x)
    d=cur.fetchall()
    st1=""
    l=[]
    for i in d:
        st1=""
        gap="|"
        st1=f"{i[0]:>5}{gap}{i[1]:>15}{gap}{i[2]:>15}"
        l.append(st1)
        print(st1)
    k=StringVar()
    k.set(st1)
    return l

def searche4(x):
    cur = con.cursor()
    cur.execute("select D_ID,Name,feild_of_practice from employee where name like
'%"+x+"%'")
    d=cur.fetchall()
    st1=""
    l=[]
    for i in d:
        st1=""
        gap="|"
        st1=f"{i[0]:>5}{gap}{i[1]:>15}{gap}{i[2]:>15}"
        l.append(st1)
        print(st1)
    k=StringVar()
    k.set(st1)
    return l


def up_salary(x, y):
    cur = con.cursor()
    cur.execute("update employee set salary=" + y + " where D_ID=" + str(x))
    con.commit()
    print("record updated successfully")


def up_feild(a, b):
    cur = con.cursor()
    cur.execute("update employee set feild_of_practice ='"+b+"' where D_ID="+str(a))
    con.commit()
    print("record updated successfully")

def check_avail(a,t):
    cur.execute("select ARRIVETime,LEAVETime from employee where D_ID="+a)
    d=cur.fetchall()


    if read_dcc(a,t):
        return read_dcc(a,t)
    else:
        return d[0]

def invoiceg(a,b):
    cur = con.cursor()
    cur.execute("select Name,Fees from employee where D_ID="+str(a))
    d=cur.fetchall()


    l=[]
    for i in d:
        for x in i:
            l.append(x)


    cur.execute("select Name from accounts where Acc_Num="+str(b))
    f=cur.fetchall()


    for j in f:
        for y in j:
            l.append(y)
    return l
```

File #8 f2

```
import mysql.connector as mycon
from tkinter import *
from sqlconnection import *

def searchp1(x):
    cur = con.cursor()
    cur.execute("select * from pharmacy where Drug_ID='%s'" %x)

    d=cur.fetchall()
    l=[]
    gap="|"
    st=""
    for i in d:
        st=f"{i[0]:>5}{gap}{i[1]:>30}{gap}{i[2]:>10}"
        l.append(st)
    print(st)
    k=StringVar()
    k.set(st)

    print("record updated successfully")
    return l

def searchp2(x):
    cur = con.cursor()
    print()
    cur.execute("select * from pharmacy where name like '"+x+"%'")
    gap="|"
    d=cur.fetchall()
    l=[]
    st=""
    for i in d:
        st=f"{i[0]:>5}{gap}{i[1]:>30}{gap}{i[2]:>10}"
        l.append(st)
    print(st)
    k=StringVar()
    k.set(st)

    print("record updated successfully")
    return l

def sel():
    cur.execute("select * from pharmacy")
    return cur.fetchall()

def ins(b, c, f, e):
    cur.execute("select Drug_Id from pharmacy")
    d=cur.fetchall()
    x=d[-1][0]
    y=int(x[1:])+1
    a="A"+str(y)
    cur.execute("insert into pharmacy values('"+a+"','"+b+"',"+f+",'"+c+"',"+e+",'12-18 months',
'NO')")
```

```
    con.commit()
    print("record inserted successfully")

def red_qua(a,b):
    cur.execute("update pharmacy set Quantity=Quantity-" + str(b) + " where Drug_Id = '" +
str(a)+"'")
    con.commit()
    print("record updated successfully")

def up_medicine_qty(a, b):
    cur = con.cursor()
    cur.execute("update pharmacy set quantity = quantity+"+b+" where Drug_ID='"+str(a)
+"'")
    con.commit()
    print("record updated successfully")

def up_medicine_price(a, b):
    cur = con.cursor()
    cur.execute("update pharmacy set price ="+b+" where Drug_ID='"+str(a)+"'")
    con.commit()
    print("record updated successfully")

def check_qua(a,b):
    cur = con.cursor()
    cur.execute("Select Quantity from pharmacy where Drug_ID='"+a+"'")
    d=cur.fetchall()
    if d[0][0]<b:
        return False
```

```
else:
    return b
```

File #9 f3

```
import mysql.connector as con
from sqlconnection import *

def write(a,b,c,d,e):
    cur.execute("insert into inventory values('"+a+"', '"+b+"', '"+c+"','"+d+"', '"+e+"')")
    con.commit()
    print("records inserted")


def read(q):
    cur.execute("select * from inventory where item_id='"+q+"'")
    print(cur.fetchall())
```

File #10 f4

```
import mysql.connector as mycon
from tkinter import *
from sqlconnection import *

def search_all():
    cur=con.cursor()
    cur.execute("select name,password from accounts")
    d=cur.fetchall()
    return d

def search_alltr():
    cur=con.cursor()
    cur.execute("select name,Acc_num from accounts")
    d=cur.fetchall()
    return d

def search01(x):
    cur = con.cursor()
    cur.execute("select * from accounts where Acc_num="+x)

    d=cur.fetchall()
    st=""
    gap="|"
    for i in d:
        st=f"{i[0]:5}{gap}{i[1]:>12}{gap}{i[2]:>8}{gap}{i[3]:>8}"
    print(st)
    k=StringVar()
    k.set(st)
    print("record updated successfully")
    return st
def convert(t):
    cur=con.cursor()
    cur.execute("select Acc_num from accounts where name='"+t+"'")
    d=cur.fetchall()
    return d

def search02(x):
    cur = con.cursor()
    cur.execute("select * from accounts where name = '%s'" %x)

    d=cur.fetchall()
    lis=[]
    for i in d:
        st1=""
        gap="|"
        st1=f"{i[0]:5}{gap}{i[1]:>12}{gap}{i[2]:>8}{gap}{i[3]:>8}"
        lis.append(st1)

    print(lis)
    print("record updated successfully")
    return lis
```

```
def up_paid(x, y):
   cur = con.cursor()
   cur.execute("update accounts set Amount_paid=" + y + " where Acc_num=" + x + "")
   con.commit()

   print("record updated successfully")


def up_total(a, b):
   cur = con.cursor()
   cur.execute("update accounts set Total_amount=" + b + " where Acc_num=" + a + "")
   con.commit()

   print("record updated successfully")

def up_totalIn2(a, b):
   cur = con.cursor()
   cur.execute("update accounts set Total_amount=Total_amount+" + str(b) + " where
Acc_num=" + str(a) + "")
   con.commit()

   print("record updated successfully")

def up_totalIn(a,b):
   cur = con.cursor()
   cur.execute("select Fees from employee where D_ID="+str(a))
   d=cur.fetchall()
   cur.execute("update accounts set Total_amount=Total_amount+" + str(d[0][0]) + " where
Acc_num=" + str(b) + "")
   con.commit()

   print("record updated successfully")

def up_paidIn(a,b):
   cur.execute("update accounts set Amount_Paid=Amount_Paid+" + str(a) + " where
Acc_num=" + str(b) + "")
   con.commit()

   print("record updated successfully")

def up_pass(a, b):
   cur = con.cursor()
   cur.execute("update accounts set password='" + b + "' where Acc_no.=" + a + "")
   con.commit()

   print("record updated successfully")


def insert_acc(y, z, k):
   cur = con.cursor()
   cur.execute("select acc_num from accounts")
   d=cur.fetchall()
   x=10001
   if d!=[]:
      for i in d:
         for j in i:
```

```
        if x<j:
            x=j
    x=x+1

cur.execute("insert into accounts values(%s,%s,%s,%s,%s)", (x, y, z, k, x))
con.commit()

print("record updated successfully")
```

File #11 f5

```
import mysql.connector as mycon
from tkinter import *
from sqlconnection import *

def searchTR_all():
   cur=con.cursor()
   cur.execute("select name,password from Transanctions")
   d=cur.fetchall()
   return d
def searchTR01(x):
   cur = con.cursor()
   cur.execute("select * from Transanctions where Acc_num="+x)

   d=cur.fetchall()
   st=""
   gap="|"
   for i in d:
      st=f"{i[0]:5}{gap}{i[1]:>12}{gap}{i[2]:>20}{gap}{i[3]:>8}{gap}{i[4]:>1}"
   print(st)
   k=StringVar()
   k.set(st)
   print("record updated successfully")
   return st

def searchTR02(x):
   cur = con.cursor()
   cur.execute("select * from Transanctions where Acc_num = %s" %x)

   d=cur.fetchall()
   lis=[]
   liss=[]
   listt=[]
   for i in d:
      st1=""
      gap="|"
      if i[5]=="D":
          st1=f"{i[0]:5}{gap}{i[1]:>18}{gap}{i[2]:^30}{gap}{i[3]:>5}{gap}{i[4]:>10}"
          listt.append(st1)
      if i[5]=="P":
          st1=f"{i[0]:5}{gap}{i[1]:>18}{gap}{i[2]:^30}{gap}{i[3]:>5}{gap}{i[4]:>10}"
          liss.append(st1)
   lis.append(listt)
   lis.append(liss)
   print(lis)
   print("record updated successfully")
   return lis

def up_paidTR(y):
   cur = con.cursor()
   cur.execute("update Transanctions set status='P' where T_ID=" + y + "")
   con.commit()
   print("record updated successfully")


def up_totalTR(a, b):
   cur = con.cursor()
   cur.execute("update Transanctions set Total_amount=" + b + " where Acc_num=" + a +
"")
   con.commit()

   print("record updated successfully")

def up_totalInTR(a,b):
   cur = con.cursor()
   cur.execute("select Fees from employee where D_ID="+str(a))
   d=cur.fetchall()
   cur.execute("update Transanctions set Total_amount=Total_amount+" + str(d[0][0]) + "
where Acc_num=" + str(b) + "")
   con.commit()

   print("record updated successfully")

def up_passTR(a, b):
   cur = con.cursor()
   cur.execute("update Transanctions set password='" + b + "' where Acc_no.=" + a + "")
   con.commit()

   print("record updated successfully")
```

```
def insert_TR(y, z, m, k, l):
    cur = con.cursor()
    cur.execute("select T_ID from Transanctions")
    d=cur.fetchall()
    x=1001
    if d!=[]:
        for i in d:
            for j in i:
                if x<j:
                    x=j
        x=x+1

    cur.execute("insert into Transanctions values(%s,%s,%s,%s,%s,%s)", (x, y, z, m, k, l))
    con.commit()

    print("record updated successfully")

def insert_TR01(y, z, k, l):
    cur = con.cursor()
    cur.execute("select Fees from employee where D_ID="+str(k))
    f=cur.fetchall()
    cur.execute("select T_ID from Transanctions")
    d=cur.fetchall()
    x=1001
    if d!=[]:
        for i in d:
```

```
        for j in i:
            if x<j:
                x=j
    x=x+1

cur.execute("insert into Transanctions values(%s,%s,%s,%s,%s,%s)", (x, y, z,'1', f[0][0],
l))
con.commit()

print("record updated successfully")
```

file #12 Acc_link

```
import csv
import datetime
import tkinter as tk

#Linking several accounts so that we can display them together on patients window

#Creating a csv file as a storage method
fn = 'Link.csv'



#writing into the csv
def writeLink(b,c):
    l = []
    global fn
    with open(fn, 'a+') as f:


        csw = csv.writer(f, lineterminator='\n')
        f.seek(0)
        l = list(csv.reader(f))
        new=[]
        for i in l:
            if i[0]==b:
                deleteLink(b)
                new=i
                new.append(c)
    if new==[]:
      new=[b,c]
    csw.writerow(new)
    print("acc linked")

#deleting a record from the csv
def deleteLink(ID):
  global fn
  with open(fn, 'r') as f:
    flag = False
    csr = csv.reader(f)
    reclst = list(csr)
    for i in range(len(reclst)):
      if reclst[i][0]==ID:
        flag = True
        reclst.pop(i)
        break
  with open(fn, 'w') as f:
    csw = csv.writer(f, lineterminator='\n')
    csw.writerows(reclst)

  if flag:
    print('Record deleted')
  else:
    print('Record not found')

#Searching a link
```

```
def searchLink(a):
   l = []
   global fn
   with open(fn, 'a+') as f:


      csw = csv.writer(f, lineterminator='\n')
      f.seek(0)
      l = list(csv.reader(f))
      for i in l:
        if i[0]==a:
           return I
```

File #13 Prescript

```
import os
def presc_se(ID,Date,I):
   try:
      f=open("Prescripts\\"+ID+"\\"+ID+Date+I+".txt",'r')
      return f.readlines()
   except:
      return []
def presc_wr(ID,Date,I,l):
  try:
      f=open("Prescripts\\"+ID+"\\"+ID+Date+I+".txt","w+")
      f.writelines(l)
      print("added to existing directory")
  except:
      try:
        os.mkdir("Prescripts\\"+ID)
      except:
        os.mkdir("Prescripts")
        os.mkdir("Prescripts\\"+ID)
      f=open("Prescripts\\"+ID+"\\"+ID+Date+I+".txt","w+")
      f.writelines(l)
      print("new directory created and saved")
```

File #14 sqlconnection

```
import mysql.connector as mycon


try:

con=mycon.connect(host="localhost",username="root",passwd="br13s8010",database="hospital_m")
   if con.is_connected():
       print("connection successful")
   cur=con.cursor()


except mycon.errors.ProgrammingError:
   print("wrong sql passwd plz change it in application file")
   quit()
```

File #15 Timemanager

```
from appointmanager1 import *
from appointmanager2 import *

def appoint_time(date,k,a,b):
  h1=int(a.split(":")[0])
  h2=int(b.split(":")[0])
  m1=int(a.split(":")[1])
  m2=int(b.split(":")[1])

  l=[]
  for i in range(h2-h1+1):
   if i+h1<h2:
      while m1<60:
        if m1!=45 and m1!=0:
          l.append(str(h1+i)+":"+str(m1)+" - "+str(h1+i)+":"+str(m1+15))
          m1+=15
        elif m1==0:
          l.append(str(h1+i)+":"+"00"+" - "+str(h1+i)+":"+str(m1+15))
          m1+=15
        else:
          l.append(str(h1+i)+":"+str(m1)+" - "+str(h1+i+1)+":"+"00")
          m1+=15
      m1=0
   elif i+h1==h2:
      while m1<m2:
        if m1!=45 and m1!=0:
          l.append(str(h1+i)+":"+str(m1)+" - "+str(h1+i)+":"+str(m1+15))
          m1+=15
        elif m1==0:
          l.append(str(h1+i)+":"+"00"+" - "+str(h1+i)+":"+str(m1+15))
          m1+=15
        else:
          l.append(str(h1+i)+":"+str(m1)+" - "+str(h1+i+1)+":"+"00")
          m1+=15
      m1=0
  poptime=searchtime(k)
  for i in poptime:
    for j in l:
      if j==str(i[0]) and str(i[1])==str(date) and i[2]=="Active":
        l.remove(j)
  return l
```

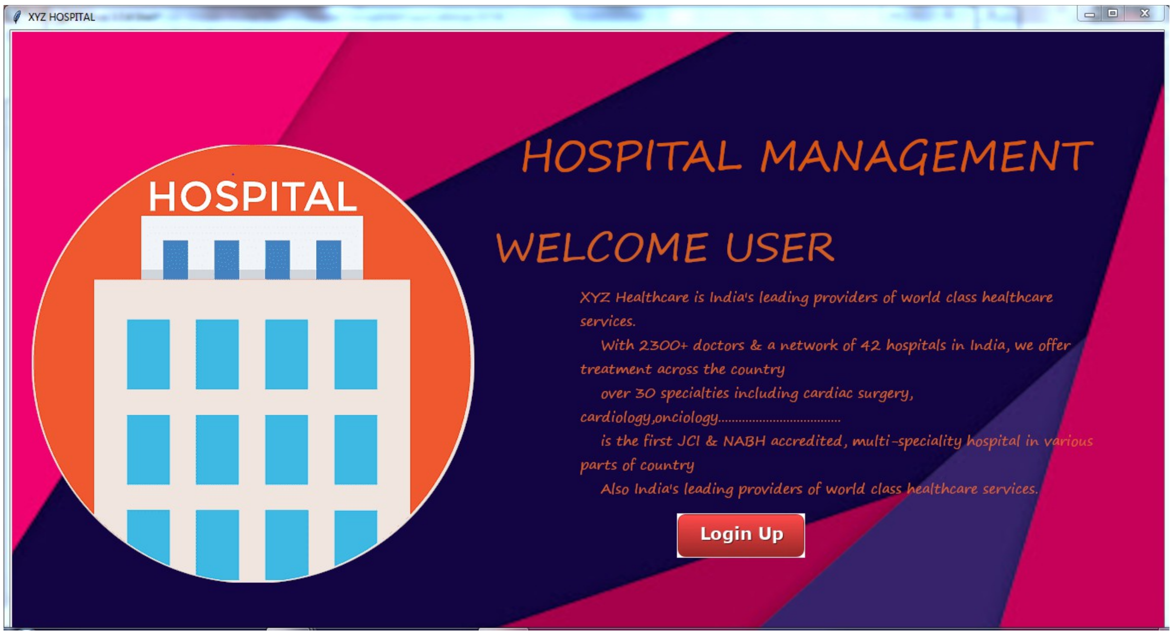
File # 16 Installation

```
#installation module
#Run This before you start working on the Application

import mysql.connector as mycon
from tkinter import *
password=input("enter your sql passwd")

con = mycon.connect(host="localhost", user="root", passwd=password)

if con.is_connected():
    print("connection successful")
cur = con.cursor()

f=open("empl.txt",'r')
x=f.readlines()

for i in x:
    cur.execute(i)
print("employee structure successfully created")

f.close()

f1=open("AccTable.txt",'r')
x1=f1.readlines()

for i in x1:
    cur.execute(i)
print("accounts structure suceessfully created")
```

```
f1.close()

f2=open("pharmtable.txt",'r')
x2=f2.readlines()

for i in x2:
   cur.execute(i)
   con.commit()
print("pharmacy structure suceesfully created")

f2.close()

f4=open("Transanc.txt",'r')
x4=f4.readlines()

for i in x4:
   cur.execute(i)
print("transanction structure suceessfully created")

f4.close()

cur.close()
con.close()

print("installation complete")
```

OUTPUT SCREENS

WELCOME SCREEN:-

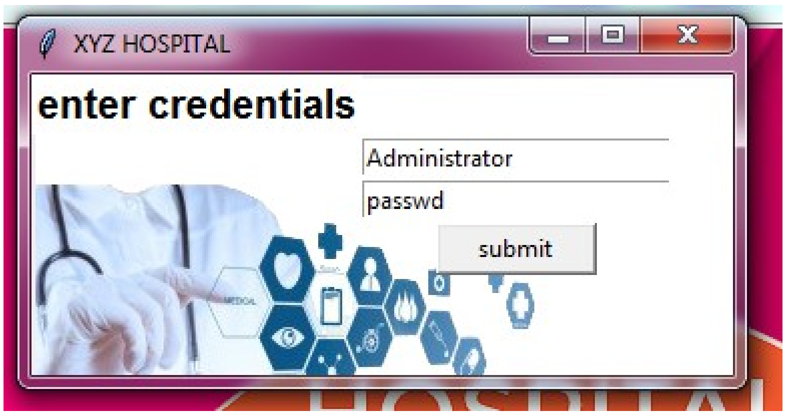


HOSPITAL MANAGEMENT

WELCOME USER

XYZ Healthcare is India's leading providers of world class healthcare services.
With 2300+ doctors & a network of 42 hospitals in India, we offer treatment across the country
over 30 specialties including cardiac surgery, cardiology,onciology..................................
is the first JCI & NABH accredited, multi-speciality hospital in various parts of country
Also India's leading providers of world class healthcare services.

LOGIN SCREEN:-

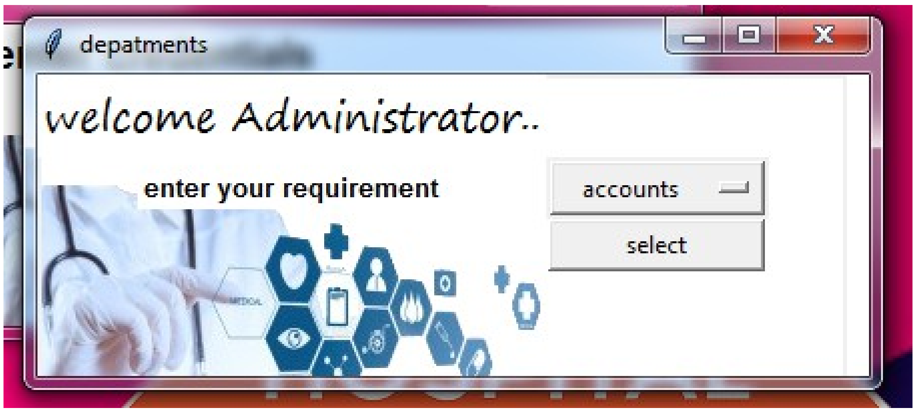


enter credentials

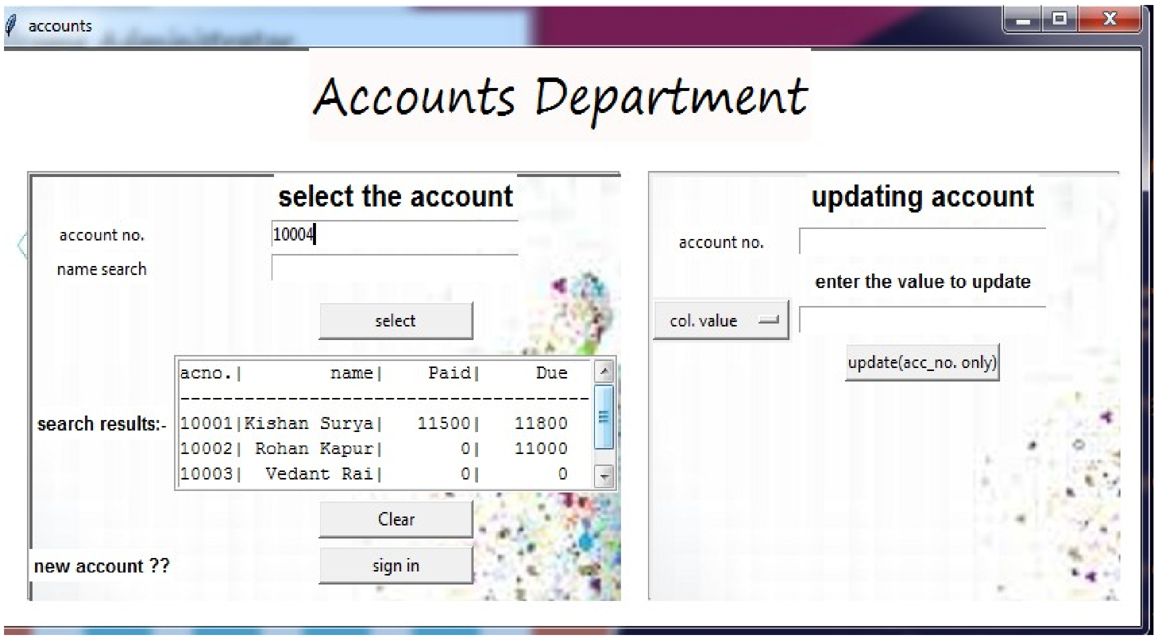
Administrator

passwd

ADMINISTRATOR SCREEN:-



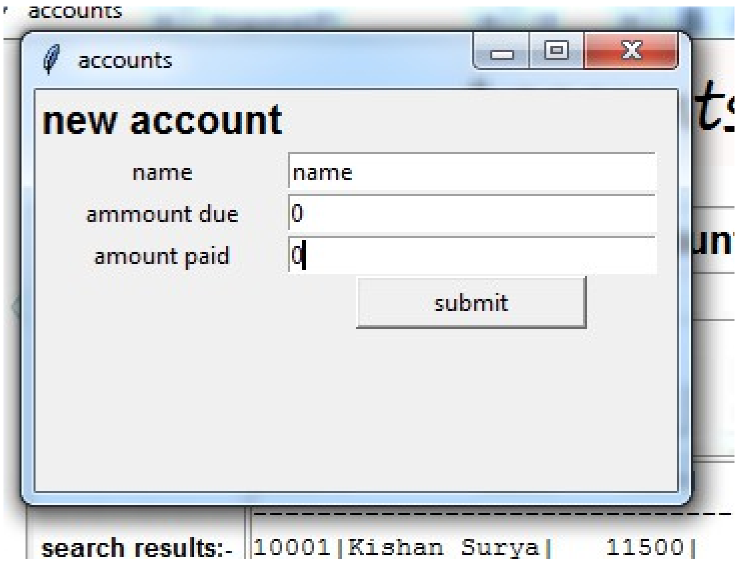
ACCOUNTS SCREEN:-



10004

```
acno.|        name|    Paid|     Due
--------------------------------------
10001|Kishan Surya|   11500|   11800
10002| Rohan Kapur|       0|   11000
10003|  Vedant Rai|       0|       0
```

NEW ACCOUNT SCREEN:-

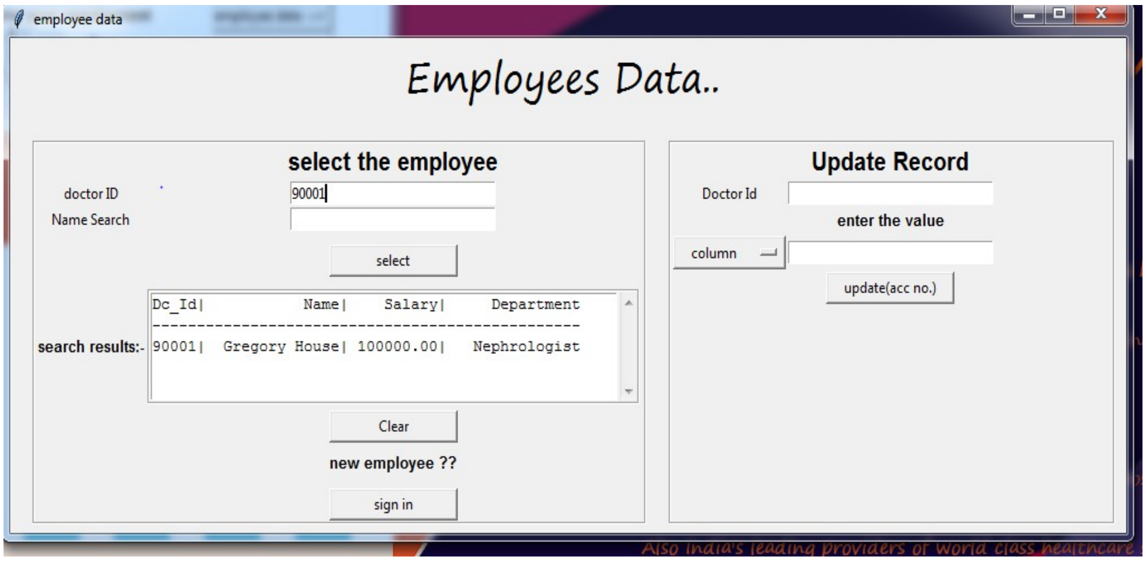


new account

name	name
ammount due	0
amount paid	0

```
search results:- 10001|Kishan Surya|    11500|
```

EMPLOYEE SCREEN:-



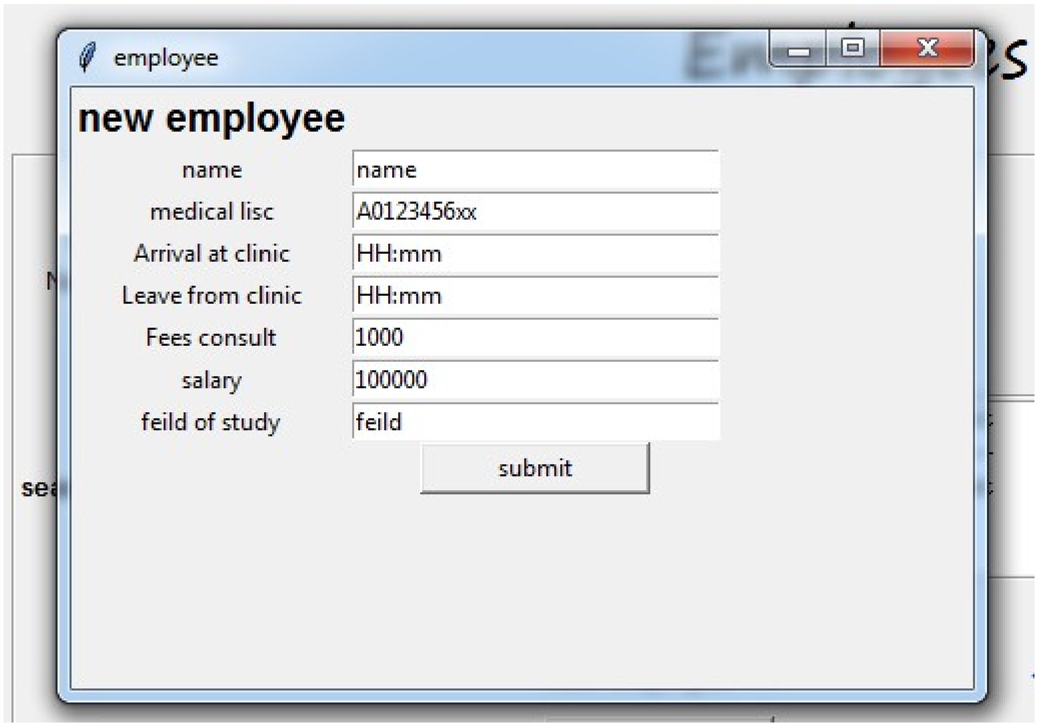
doctor ID 90001

Name Search

```
Dc_Id|          Name|     Salary|       Department
--------------------------------------------------
90001|   Gregory House| 100000.00|     Nephrologist
```

search results:-

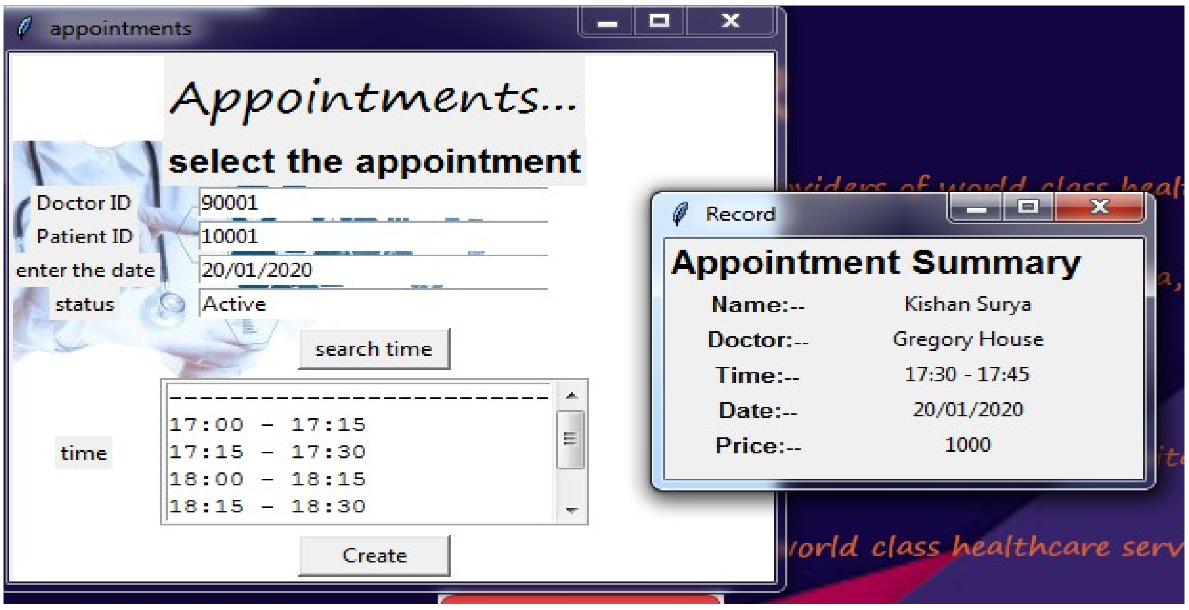
NEW EMPLOYEE SCREEN:-



new employee

name	name
medical lisc	A0123456xx
Arrival at clinic	HH:mm
Leave from clinic	HH:mm
Fees consult	1000
salary	100000
feild of study	feild

APPOINTMENT'S SCREENS:-



Appointments...

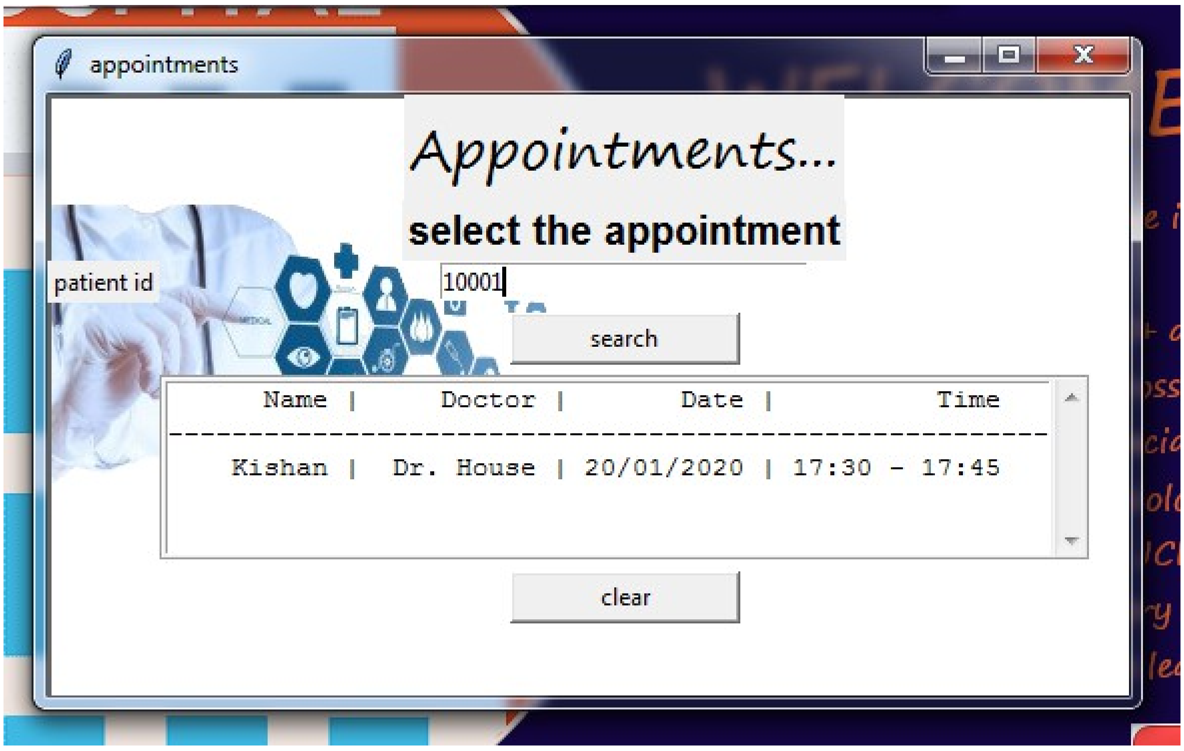
select the appointment

Doctor ID	90001
Patient ID	10001
enter the date	20/01/2020
status	Active

```
-------------------------
17:00 - 17:15
17:15 - 17:30
18:00 - 18:15
18:15 - 18:30
```

Appointment Summary

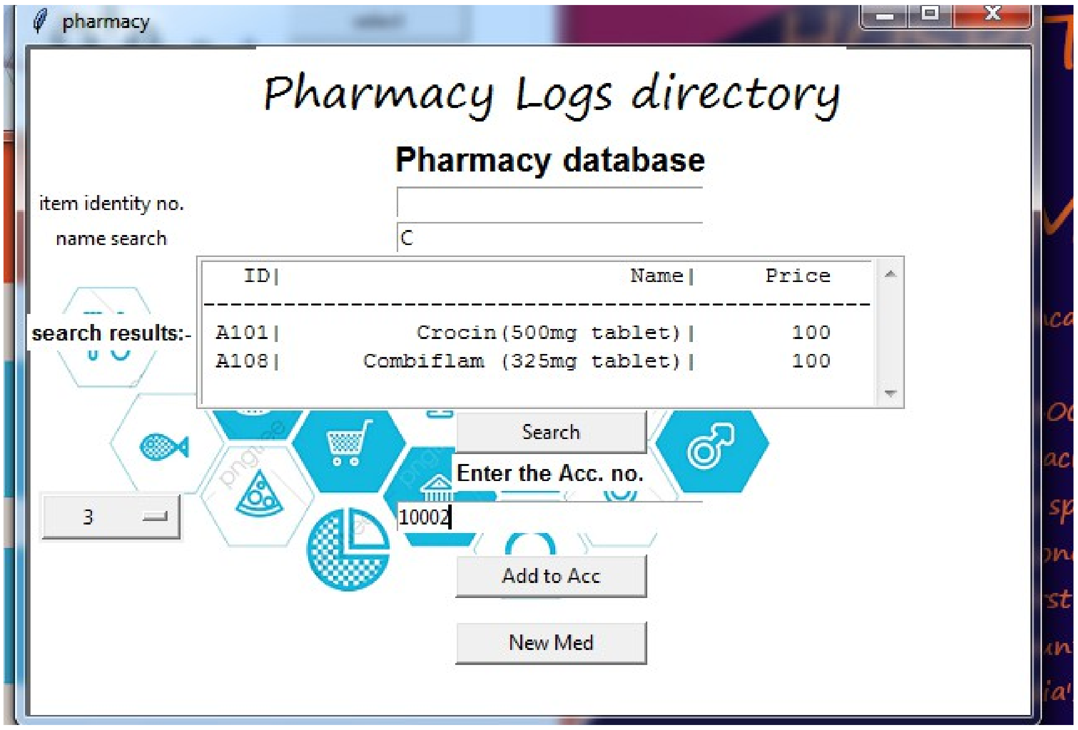
Name:--	Kishan Surya
Doctor:--	Gregory House
Time:--	17:30 - 17:45
Date:--	20/01/2020
Price:--	1000



10001

```
    Name  |      Doctor  |        Date  |          Time
---------------------------------------------------------------
  Kishan  |   Dr. House  |  20/01/2020  |  17:30 - 17:45
```

PHARMACY WINDOW:-



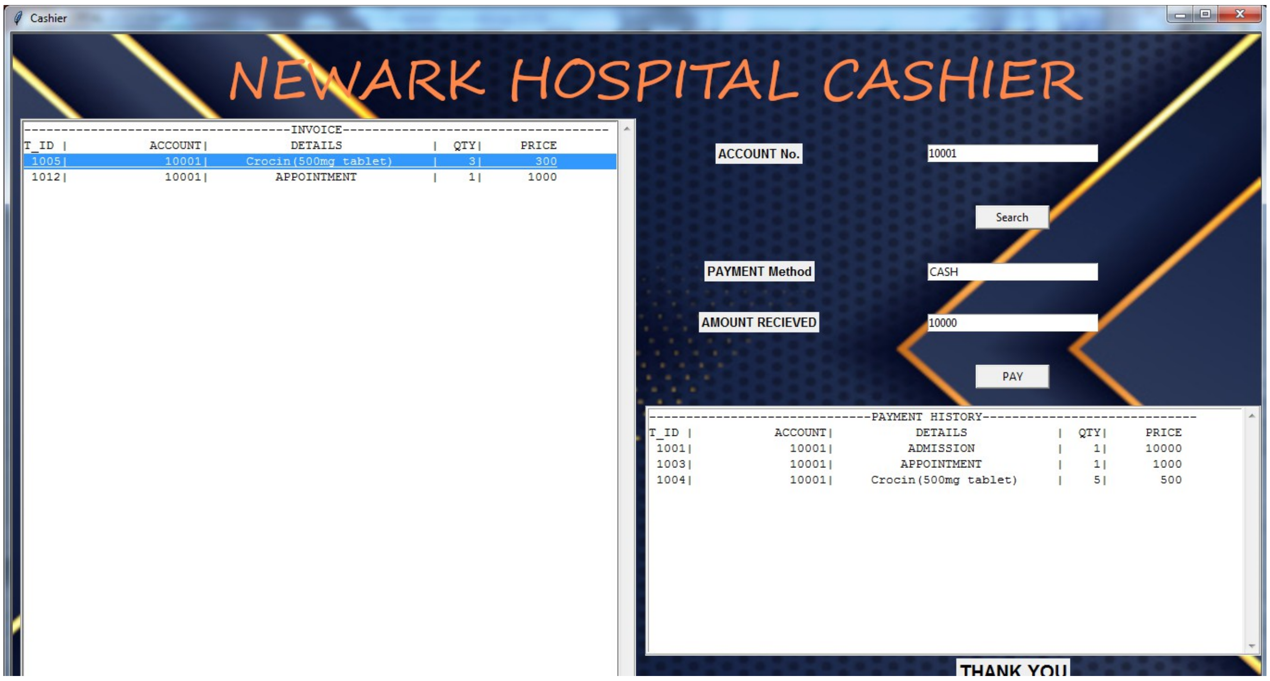
C

```
  ID|                         Name|      Price
----------------------------------------------------
A101|         Crocin(500mg tablet)|        100
A108|      Combiflam (325mg tablet)|        100
```

3

10002

CASHIER WINDOW:-



```
------------------------------------INVOICE------------------------------------
T_ID |              ACCOUNT|          DETAILS             |  QTY|     PRICE
 1005|                10001|     Crocin(500mg tablet)     |    3|       300
 1012|                10001|        APPOINTMENT           |    1|      1000
```

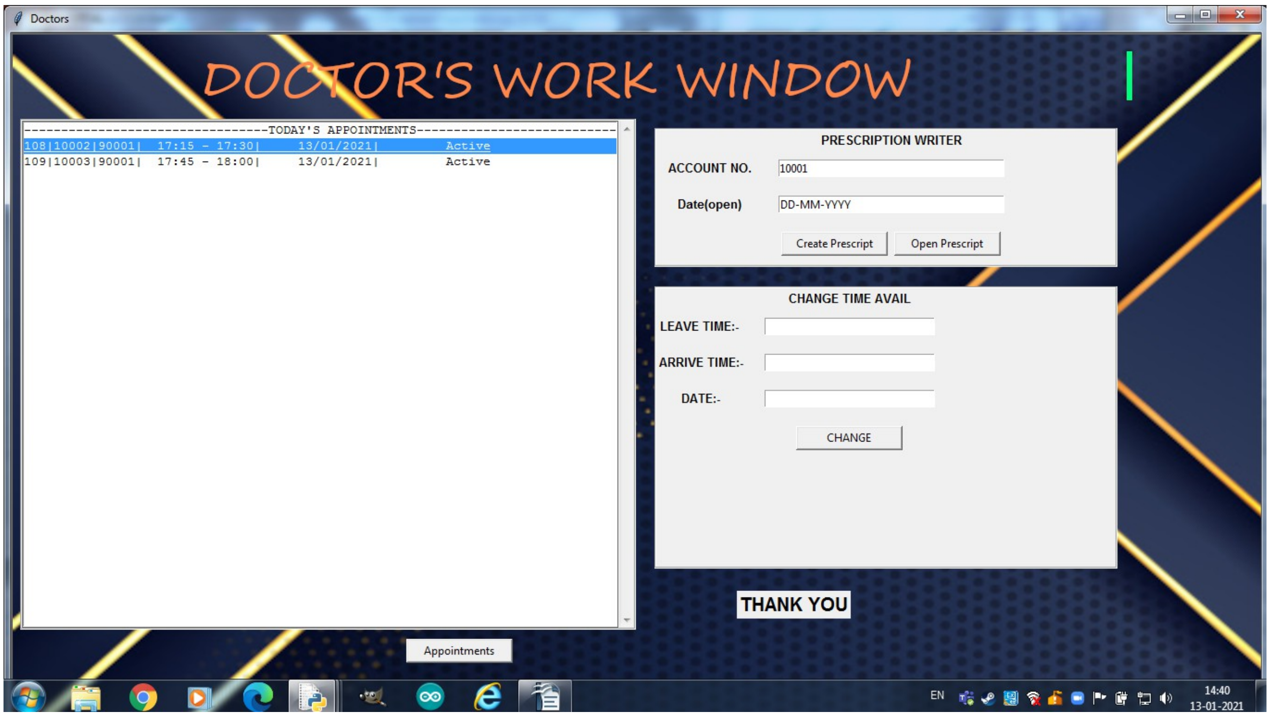
ACCOUNT No.: 10001

PAYMENT Method: CASH

AMOUNT RECIEVED: 10000

```
------------------------------PAYMENT HISTORY------------------------------
T_ID |              ACCOUNT|          DETAILS             |  QTY|     PRICE
 1001|                10001|         ADMISSION            |    1|     10000
 1003|                10001|        APPOINTMENT           |    1|      1000
 1004|                10001|     Crocin(500mg tablet)     |    5|       500
```

DOCTOR'S WINDOW:-



```
-------------------------------TODAY'S APPOINTMENTS-------------------------------
108|10002|90001|  17:15 - 17:30|      13/01/2021|          Active
109|10003|90001|  17:45 - 18:00|      13/01/2021|          Active
```

ACCOUNT NO.: 10001

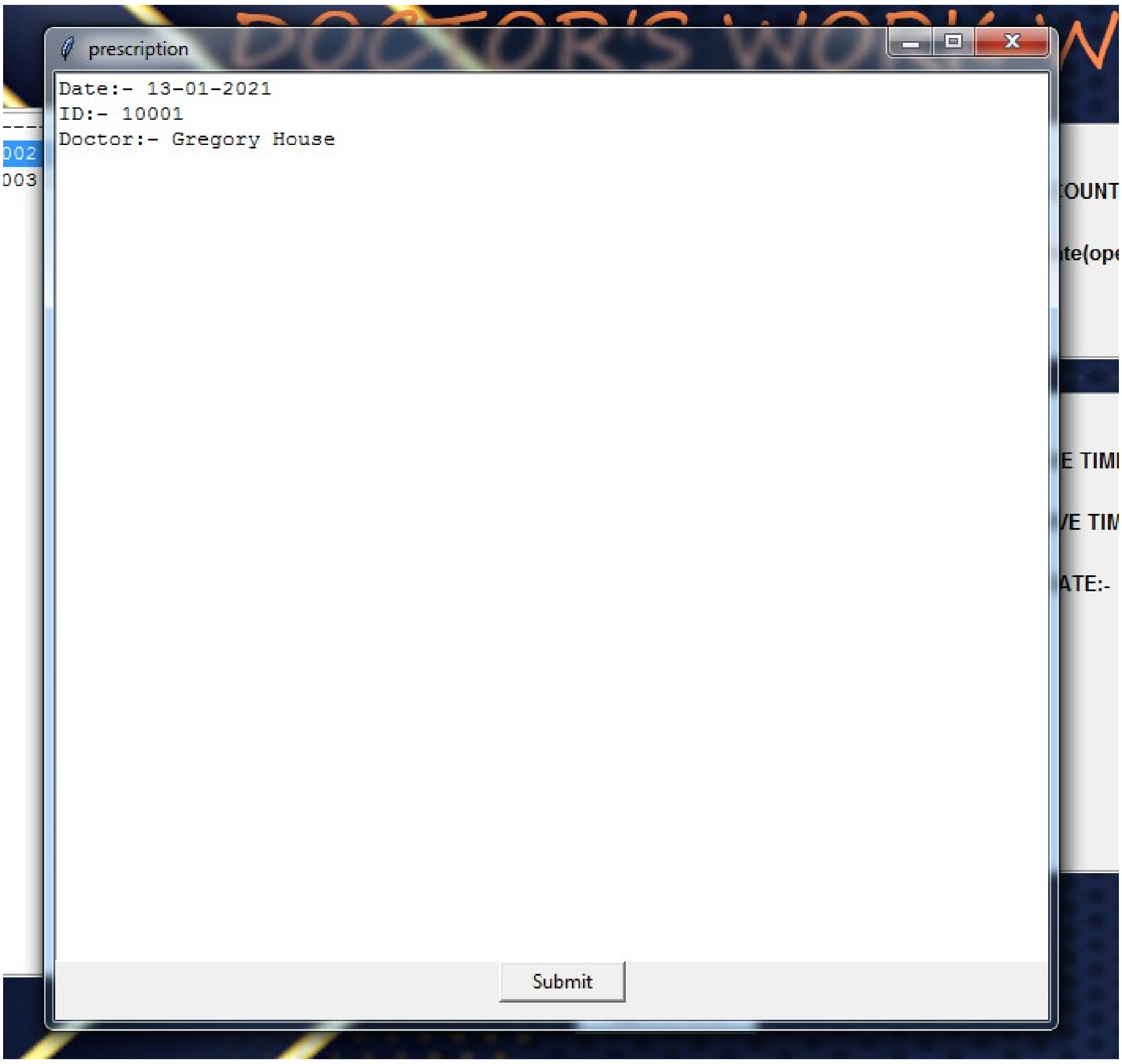
Date(open): DD-MM-YYYY

LEAVE TIME:-

ARRIVE TIME:-

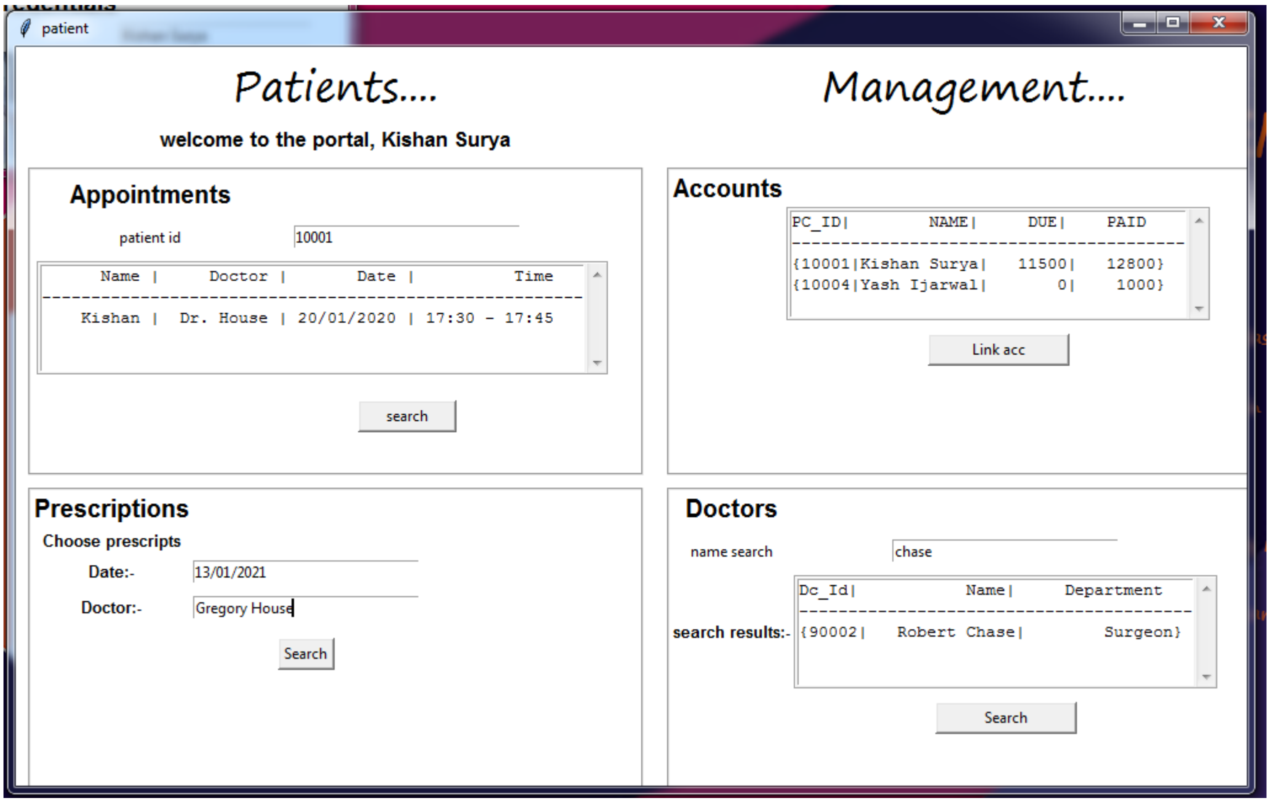
DATE:-

PRESCRIPTION WINDOW:-



```
Date:- 13-01-2021
ID:- 10001
Doctor:- Gregory House
```

PATIENT WINDOW:-



patient

Patients....

welcome to the portal, Kishan Surya

Appointments

patient id: 10001

```
   Name  |     Doctor  |        Date  |           Time
-------------------------------------------------------
   Kishan  |   Dr. House  | 20/01/2020  | 17:30 - 17:45
```

search

Prescriptions

Choose prescripts

Date:- 13/01/2021

Doctor:- Gregory House

Search

Management....

Accounts

```
PC_ID|         NAME|      DUE|     PAID
----------------------------------------
{10001|Kishan Surya|    11500|    12800}
{10004|Yash Ijarwal|        0|     1000}
```

Link acc

Doctors

name search: chase

search results:-

```
Dc_Id|            Name|      Department
---------------------------------------
{90002|    Robert Chase|         Surgeon}
```

Search

CONCLUSION

The project was made to ensure a successful interconnection between several stakeholders working under ta single hospital. The program successfully incorporates the needs of all those who come in contact with it. Using multiple modes of entrance into the hospital's connections we can see that each person can get his or her information without compromising any security.

The doctors can easily schedule their appointments and can also make sure their prescriptions are well maintained. The patients can get the required information on any doctor and see the records of past visit too.

The pharmacy is also well organized and with the help of quick searches and account no. can sell medicines efficiently too. The administrator has the records of backend and can change a few fundamental details in any record if the need arises.

In the end the project expanded well beyond any initial apprehensions. A lot of new features were added in the course of its development and turned out to be a huge success.

FUTURE ENHANCEMENTS

The program has a few inherent defects which can be improved with some more time. The features related to linking accounts and including insurance to those with linked accounts and blood relations could have been made.

Many a place code contains blocks which are never executed at any point of time but bare present in order for future enhancements. The file 'f3.py' is not used throughtout the entirety of the program but is present in order to include hospital assets such as MRI and CT scanners to a better accountablity.

With more time a little more aethetic appeal could also be incorporated into the program for a much better user experience.

Bibliograpghy

1.https://www.google.co.in

2.https://en.wikipedia.org

3.Computer science with python -Sumita Arora

4.https://codemy.com/