**B.Tech Minor Project Report**

**on**

**HOSPITAL MANAGEMENT SYSTEM USING PYTHON**

By

**HEMANT (11481)**

**HEMANT YADAV (11482)**

**MINTU (11497)**

**Under the Supervision of**

**Sangeeta Lamba (Assistant Professor, CUH)**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF ENGINEERING & TECHNOLOGY**

**CENTRAL UNIVERSITY OF HARYANA**

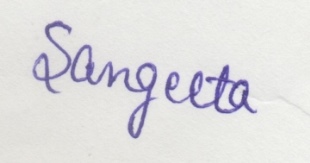
**MAHENDRAGARH – 123029**

**September , 2020 – December , 2020**

**CERTIFICATE**

This is to be certified that the project entitled “Hospital management system” has been submitted for the Bachelor of Technology in Computer Science & Engineering at Central University of Haryana during the period from September 2020 to December 2020 is a bonafide piece of project work carried out by “Hemant , Hemant Yadav & Mintu” towards the partial fulfillment for the award of the Degree (B.Tech) under the guidance of “Sangeeta Lamba” and supervision and no part of thereof has been submitted by them for any degree or diploma.

**Sign. of HOD**  **Sign. Of Supervisor** :

Prof. Rakesh Dhiman 

(H.O.D CSE , CUH) Ms.Sangeeta Lamba

(Assistant Professor , CUH)

**ACKNOWLEDGEMENT**

Every Industrial Training big or small is a successful largely due to effort of a number of people who have always given their valuable advice or lent a helping hand. I sincerely appreciate the inspiration; support and guidance of all those people who have been instrumented in making my training a success.

The present project work is the several days study of the various aspects of the project development. During this the effort in the present study, we have received a great amount of help from **Dr.Rakesh Dhiman** , which we wish to acknowledge and thank from depth of our hearts.

My Parents have put myself ahead of themselves. Because of their hard work and dedication, I have had opportunities beyond my wildest dreams. My heartfelt thanks to them for giving me all I ever needed to be successful student and individual.

Finally, I express my thanks to all my other professors, classmates, friends, neighbours and my family members who helped me for the completion of my project and without infinite love and patience this would never have been possible.

**HEMANT(11481)**

**HEMANT YADAV(11482)**

**MINTU(11497)**

**Computer Science & Engineering**

**DECLARATION**

We hereby certify that this work which is being presented in this B.Tech. Minor Project report entitled “ **SOFTWARE USING PYTHON ”,** in requirements for the award of the **Bachelor of Technology in Computer Science & Engineering** is an authentic record work performed during the period from September 2020 to December 2020 under the supervision of **Sangeeta Lamba (**Assistant Professor , CUH)Computer Science & Engineering Department.

The matter presented in this project report has not been submitted for the award of any other degree elsewhere.

*Signature of Candidates*

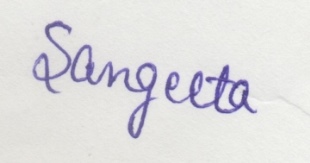
**HEMANT (11481)**

**HEMANT YADAV (11482)**

**MINTU (11497)**

This is to certify that the above statements made by the mentioned candidates are correct as per my knowledge.

Date : *Signature of Supervisor*

**

**Ms.SANGEETA LAMBA**

**(Assistant Professor , CUH)**

**Table of Content**

|  |  |
| --- | --- |
| Title | Page No. |
| Certificate | 2 |
| Acknowledgement | 3 |
| Declaration | 4 |
| Table of Content | 5 |
| List of Figures | 7 |
| 1: Introduction To Python | 9 |
| 1.1 : History Of Python | 9 |
| 1.2 : Python Versions | 9 |
| 1.3 : Installing Python | 10 |
| 1.4 : Executing Python From The Command Line | 10 |
| 1.5 : Dynamic Types | 13 |
| 1.6 : Reserved Keywords | 13 |
| 1.7 : Naming Conventions | 13 |
| 2 : What is Hospital Management System | 15 |
| 3 : SRS Diagram | 16 |
| 4 : Hospital Management System | 17 |
| 5 : Advantages | 26 |
| 6 : Limitations | 26 |
| 7 : References | 27 |

|  |  |  |
| --- | --- | --- |
| Appendix : | |  |
| A | Contributory Source Code | 28 |

**List of Figures**

|  |  |
| --- | --- |
| **Figures** | **Page No.** |
| Fig1.1 Basic interactive mode in cmd | 11 |
| Fig1.2 Basic interactive mode in Python Shell | 12 |
| Fig3.1 SRS Diagram | 14 |
| Fig4.1 Login Page with Password1,1 | 15 |
| Fig4.2 Home Page of Software | 16 |
| Fig4.3 Patient Registration Frame | 16 |
| Fig4.4 Frame with Data | 17 |
| Fig4.5 Data Saved | 17 |
| Fig4.6 Receipt Generated | 18 |
| Fig4.7 Patient Admit Frame | 18 |
| Fig4.8 Frame with Data | 19 |
| Fig4.9 Patient Registration Data Search | 19 |
| Fig4.10 Information fine out by searching | 20 |
| Fig4.11 Data Saved | 20 |
| Fig4.12 Data Search | 21 |
| Fig4.13 Data Search by Name , ID | 21 |
| Fig4.14 Patient Data Found | 22 |
| Fig4.15 Staff Record Found | 22 |
| Fig4.16 At Billing no need to enter again Patient Data | 23 |
| Fig4.17 Automatic Addition & Bill Generated | 23 |
| Fig4.18 Appointment | 24 |

**Introduction To Python**

**1.1 History of Python**

[Python](https://www.geeksforgeeks.org/python-programming-language/) is a widely used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

**Let’s dig deeper**

In the late 1980s, history was about to be written. It was that time when working on Python started. Soon after that, Guido Van Rossum began doing its application based work in December of 1989 by at Centrum Wiskunde & Informatica (CWI) which is situated in Netherland. It was started firstly as a hobby project because he was looking for an interesting project to keep him occupied during Christmas. The programming language which Python is said to have succeeded is ABC Programming Language, which had the interfacing with the Amoeba Operating System and had the feature of exception handling. He had already helped to create ABC earlier in his career and he had seen some issues with ABC but liked most of the features. After that what he did as really very clever. He had taken the syntax of ABC, and some of its good features. It came with a lot of complaints too, so he fixed those issues completely and had created a good scripting language which had removed all the flaws. The inspiration for the name came from BBC’s TV Show – ‘Monty Python’s Flying Circus’, as he was a big fan of the TV show and also he wanted a short, unique and slightly mysterious name for his invention and hence he named it Python! He was the “Benevolent dictator for life” (BDFL) until he stepped down from the position as the leader on 12th July 2018. For quite some time he used to work for Google, but currently, he is working at Dropbox.  
The language was finally released in 1991. When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & C. Its design philosophy was quite good too. Its main objective is to provide code readability and advanced developer productivity. When it was released it had more than enough capability to provide classes with inheritance, several core data types exception handling and functions.

* 1. **Python 3.9.1 is the latest version**

The two of the most used versions has to Python 2.x & 3.x. There is a lot of competition between the two and both of them seem to have quite a number of different fanbase.

For various purposes such as developing, scripting, generation and software testing, this language is utilized. Due to its elegance and simplicity, top technology organizations like Dropbox, Google, Quora, Mozilla, Hewlett-Packard, Qualcomm, IBM, and Cisco have implemented Python.

Python has come a long way to become the most popular coding language in the world. Python has just turned 30, but it still has that unknown charm & X factor which can be clearly seen from the fact that Google users have consistently searched for Python much more than they have searched for Kim Kardashian, Donald Trump, Tom Cruise etc.

Python has been an inspiration for many other coding languages such as Ruby, Cobra, Boo, Coffee Script ECMA Script, Groovy, Swift Go , Julia etc.

* 1. **Installing Python**

Installing Python is a simple matter, regardless of which platform you’re using.

The first step is to obtain a recent distribution for your machine; the most recent

One can always be found at www.python.org.

Having more than one version of Python

You may already have an earlier version of Python installed on your machine. Many Linux distributions and Mac OS X come with Python 2.x as part of the operating system.

Because Python 3 isn’t completely compatible with Python 2, it’s reasonable to wonder if installing both versions on the same computer will cause a conflict.

There’s no need to worry; you can have multiple versions of Python on the same computer.

In the case of UNIX-based systems like OS X and Linux, Python 3 installs alongside the older version and doesn’t replace it. When your system looks for “python ,”it still finds the one it expects; and when you want to access Python 3, you can runpython3.1 or idle3.1. In Windows, the different versions are installed in separate locations and have separate menu entries.

Some basic platform-specific descriptions for the Python installation are given next .The specifics can vary quite a bit depending on your platform, so be sure to read the instructions on the download pages and with the various versions. You’re probably familiar with installing software on your particular machine, so I’ll keep these descriptions

short:

􀂃 Microsoft Windows—Python can be installed in most versions of Windows by using the Python installer program, currently called python-3.1.msi. Just download it, execute it, and follow the installer’s prompts. You may need to be logged in as administrator to run the install. If you’re on a network and don’t have the administrator password, ask your system administrator to do the installation

for you.

􀂃 Macintosh—You need to get a version of Python 3 that matches your OS X version and your processor. After you determine the correct version, download the disk image file, double-click to mount it, and run the installer inside. The OSX installer sets up everything automatically, and Python 3 will be in a subfolder inside the Applications folder, labeled with the version number. Mac OS X ships with various versions of Python as part of the system, but you don’t need to worry about that—Python 3 will be installed in addition to the system version.You can find more information about using Python on OS X by following the

links on the Python home page.

􀂃 Linux/UNIX—Most Linux distributions come with Python installed. But the versions of Python vary, and the version of Python installed may not be version 3; for this book, you need to be sure you have the Python 3 packages installed. It’s also possible that IDLE isn’t installed by default, and you’ll need to install that package separately.

* 1. **Executing Python from the command Line**

**Running Python on your OS**

## Run the Python command-line interpreter, under your OS of choice,

Windows

* Open Command line:   Start menu -> Run  and type cmd
* Type:   C:\python27\python.exe
* Note: This is the default path for Python 2.7. If you are using a computer where Python is not installed in this path, change the path accordingly.

**IDLE and the basic interactive mode**

You have two built-in options for obtaining interactive access to the Python interpreter : the original basic (command-line) mode and IDLE. IDLE is available on many platforms, including Windows, Mac, and Linux, but it may not be available on others.

You may need to do more work and install additional software packages to get IDLE running, but it will be worth it because it’s a large step up from the basic interactive mode. On the other hand, even if you normally use IDLE, at times you’ll likely want to fire up the basic mode. You should be familiar enough to start and use either one.

The basic interactive mode is a rather primitive environment. But the interactive examples in this book are generally small; and later in this book, you’ll learn how to easily bring code you’ve placed in a file into your session (using the module mechanism).

Let’s look at how to start a basic session on Windows, Mac OS X, and UNIX:

􀂃 Starting a basic session on Windows—For version 3.x of Python, you navigate to the Python (command-line) entry on the Python 3.x submenu of the Programs folder on the Start menu, and click it. Alternatively, you can directly find the Python.exe executable (for example, in C:\Python31) and double-click it.Doing so brings up the window

􀂃 Starting a basic session on Mac OS X—Open a terminal window, and type python3.If you get a “Command not found” error, run the Update Shell Profile. Command script found in the Python3 subfolder in the Applications folder.

􀂃 Starting a basic session on UNIX—Type python3.1 at a command prompt. A version message similar to the one followed by the Python prompt >>> appears in the current window.

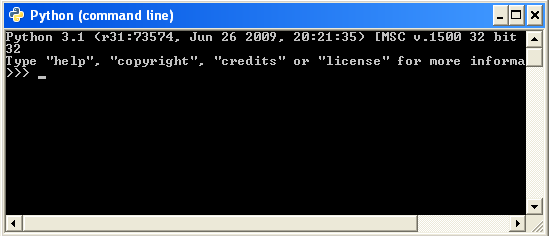


Fig1.1 Basic interactive mode in Command Prompt

􀂃 Starting IDLE on Mac OS X—Navigate to the Python 3.x subfolder in the Applications folder, and run IDLE from there.

􀂃 Starting IDLE on Linux or UNIX—Type idle3.1 at a command prompt. This

brings up a window similar to the one If you installed IDLE through your distribution’s package manager, there should also be a menu entry for IDLE under the Programming submenu or something similar

Choosing between basic interactive mode and IDLE

Which should you use, IDLE or the basic shell window? To begin, use either IDLE or the Python Shell window. Both have all you need to work through the code examplesin this book until you reach chapter 10. From there, we’ll cover writing your own modules, and IDLE will be a convenient way to create and edit files. But if you have a strongpreference for another editor, you may find that a basic shell window and your favoriteeditor serve you just as well. If you don’t have any strong editor preferences, I suggestusing IDLE from the beginning.

**Using IDLE’s Python Shell window**

The Python Shell window opens when you fire up IDLE. It provides automatic

indentation and colors your code as you type it in, based on Python syntax types.

You can move around the buffer using the mouse, the arrow keys, the Page Up and Page Down keys, and/or a number of the standard Emacs key bindings. Check the Help menu for the details.

Everything in your session is buffered. You can scroll or search up, place the cursor on any line, and press Enter (creating a hard return), and that line will be copied to the bottom of the screen, where you can edit it and then send it to the interpreter by pressing the Enter key again. Or, leaving the cursor at the bottom, you can toggle up and down through the previously entered commands using Alt-P and Alt-N. This will successively bring copies of the lines to the bottom. When you have the one you want, you can again edit it and then send it to the interpreter by pressing the Enter key. You can complete Python keywords or user-defined values by pressing Alt-/.

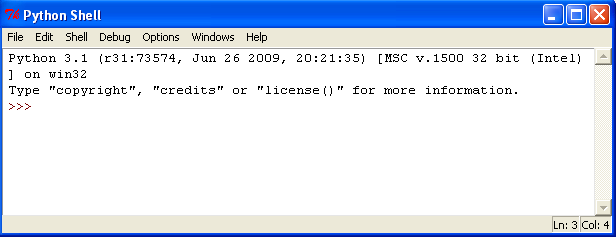


Fig1.2 Basic interactive mode in Python Shell

* 1. **Dynamic Types**

Python don't have any problem even if we don't declare the type of variable. It states the kind of variable in the runtime of the program. So, Python is a dynamically typed language.

* 1. **Reserved Keywords**

Keywords are the reserved words in Python.

We cannot use a keyword as a [variable](https://www.programiz.com/python-programming/variables-datatypes) name, [function](https://www.programiz.com/python-programming/function) name or any other identifier. They are used to define the syntax and structure of the Python language.

In Python, keywords are case sensitive.

There are 33 keywords in Python 3.7. This number can vary slightly over the course of time.

All the keywords except True, False and None are in lowercase and they must be written as they are. The list of all the keywords is given below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| False | await | else | import | pass |
| None | break | except | in | raise |
| True | class | finally | is | return |
| and | continue | for | lambda | try |
| as | def | from | nonlocal | while |
| assert | del | global | not | with |
| async | elif | if | or | Yield |

* 1. **Naming Conventions**

1. General

Avoid using names that are too general or too wordy. Strike a good balance between the two.

Bad: data\_structure, my\_list, info\_map, dictionary\_for\_the\_purpose\_of\_storing\_data\_representing\_word\_definitions

Good: user\_profile, menu\_options, word\_definitions

Don’t be a jackass and name things “O”, “l”, or “I”

When using Camel Case names, capitalize all letters of an abbreviation (e.g. HTTPServer)

2. Packages

Package names should be all lower case

When multiple words are needed, an underscore should separate them

It is usually preferable to stick to 1 word names

3. Modules

Module names should be all lower case

When multiple words are needed, an underscore should separate them

It is usually preferable to stick to 1 word names

4. Classes

Class names should follow the UpperCaseCamelCase convention

Python’s built-in classes, however are typically lowercase words

Exception classes should end in “Error”

5. Global (module-level) Variables

Global variables should be all lowercase

Words in a global variable name should be separated by an underscore

6. Instance Variables

Instance variable names should be all lower case

Words in an instance variable name should be separated by an underscore

Non-public instance variables should begin with a single underscore

If an instance name needs to be mangled, two underscores may begin its name

7. Methods

Method names should be all lower case

Words in an method name should be separated by an underscore

Non-public method should begin with a single underscore

If a method name needs to be mangled, two underscores may begin its name

8. Method Arguments

Instance methods should have their first argument named ‘self’.

Class methods should have their first argument named ‘cls’

9. Functions

Function names should be all lower case

Words in a function name should be separated by an underscore

10. Constants

Constant names must be fully capitalized

Words in a constant name should be separated by an underscore

**2. What is Hospital Management System (HMS)**

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. They manage the data related to all departments of healthcare such as,

* + 1. **Login Page**
    2. **Home Page**
    3. **Patient Registration**
    4. **Patient Admit**
    5. **Doctor Registration**
    6. **Doctor information**
    7. **Registration information**
    8. **Patient Records**
    9. **Generate Bill**
    10. **Appointment**

**3. SRS - Diagram**

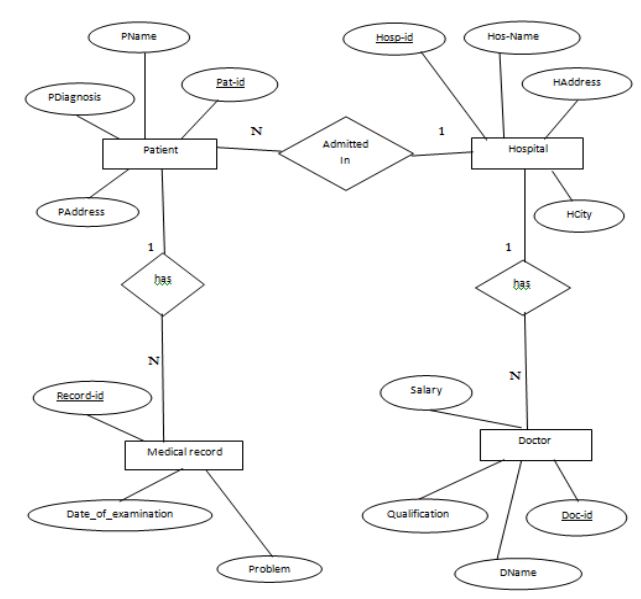


Fig3.1 SRS Diagram

4. **Hospital Management System**

**Modules :**

1. **Login Page**
2. **Home Page**
3. **Patient Registration**
4. **Patient Admit**
5. **Doctor Registration**
6. **Doctor information**
7. **Registration information**
8. **Patient Records**
9. **Generate Bill**
10. **Appointment**

**Tool :**

**Python IDLE**

**DataBase :**

**Sqlite3**

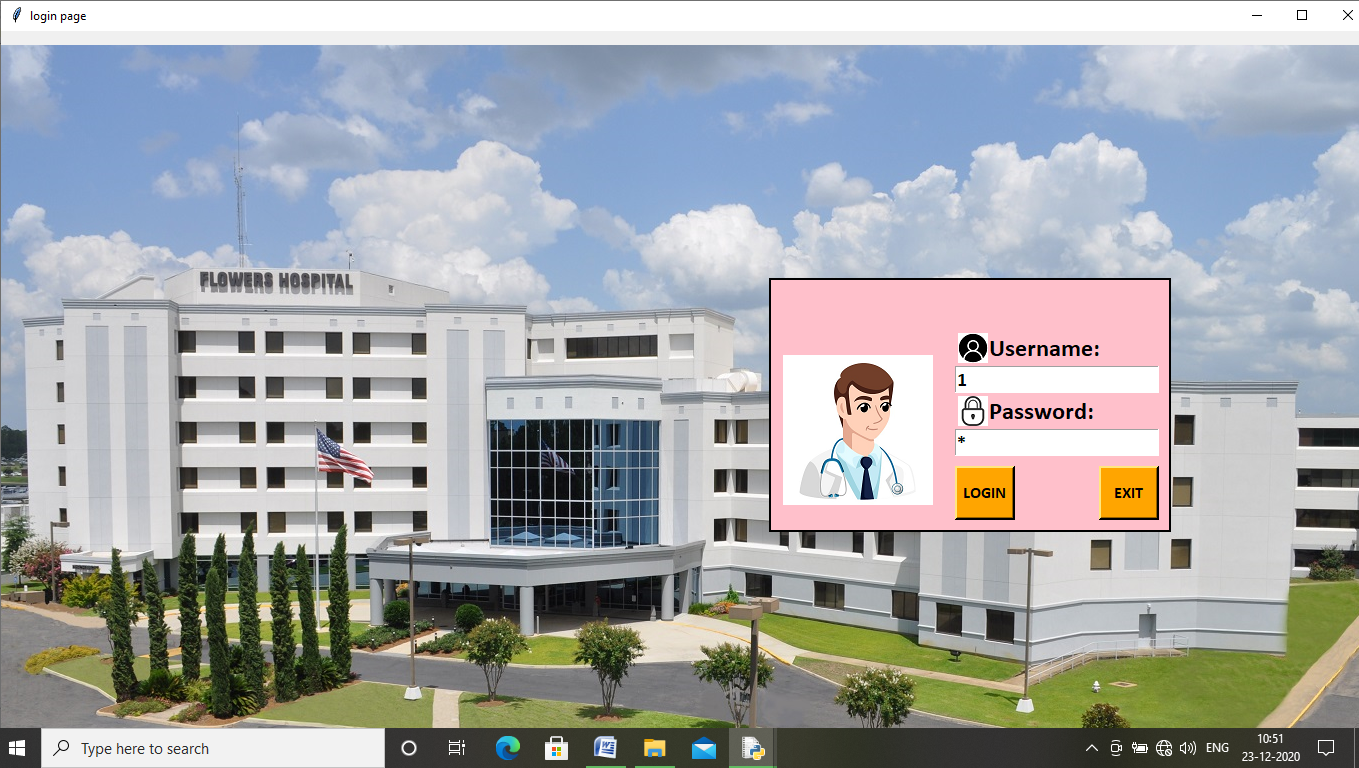


Fig4.1 Login page with password 1,1

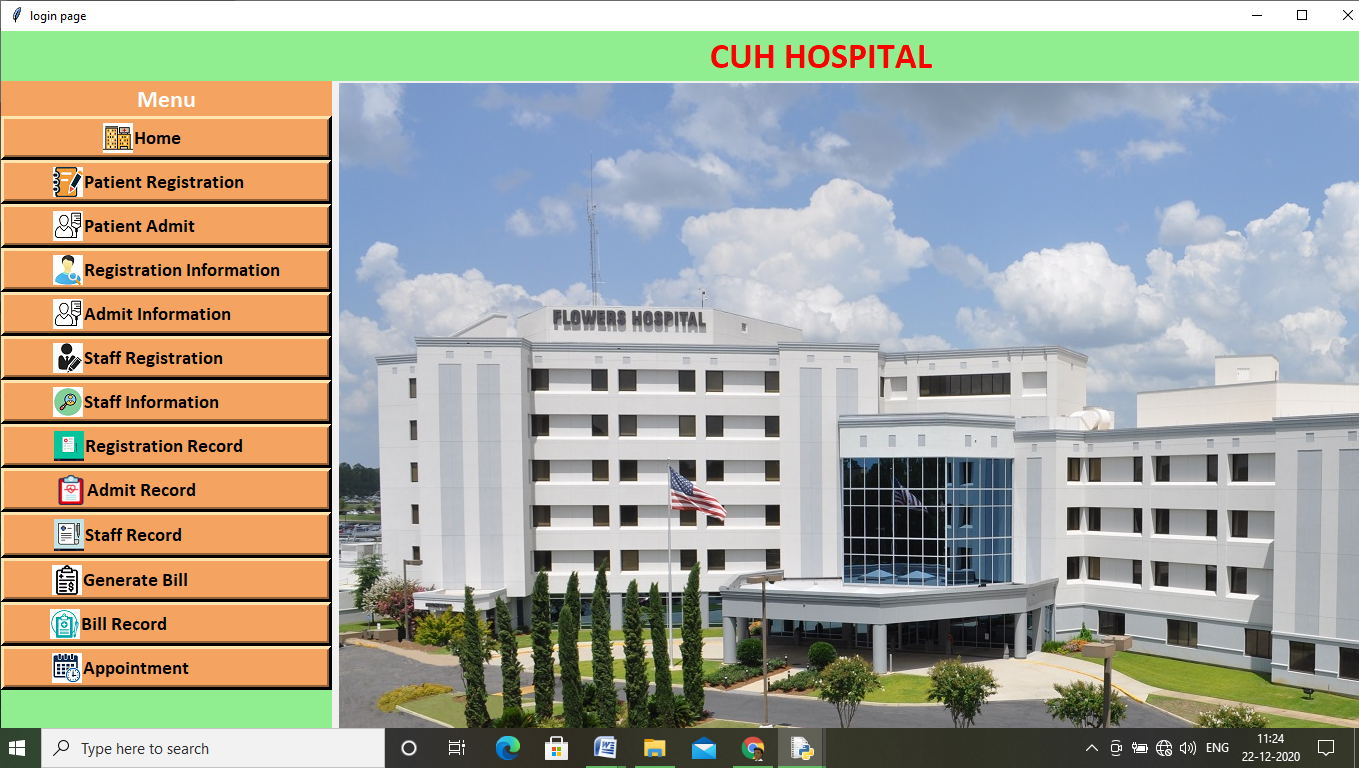


Fig4.2 Home page of software

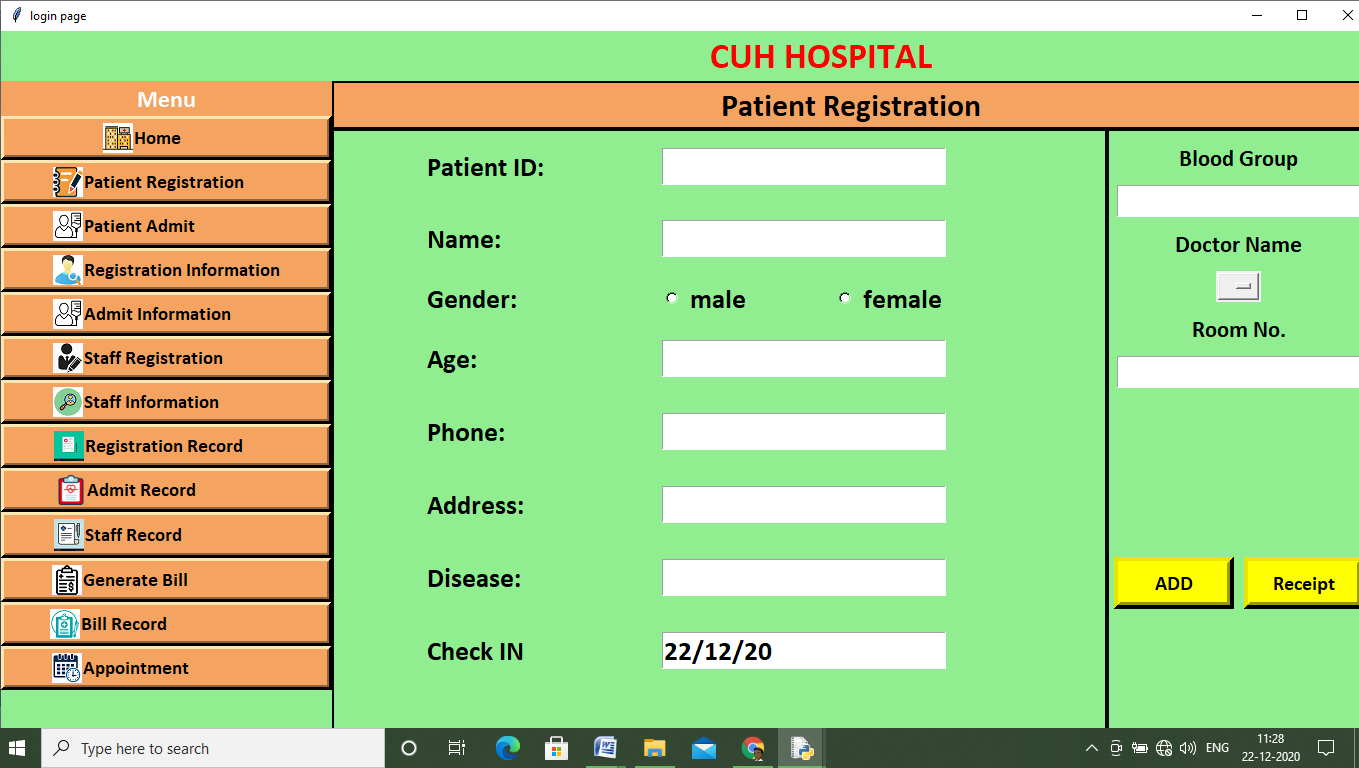


Fig4.3 Patient registration frame

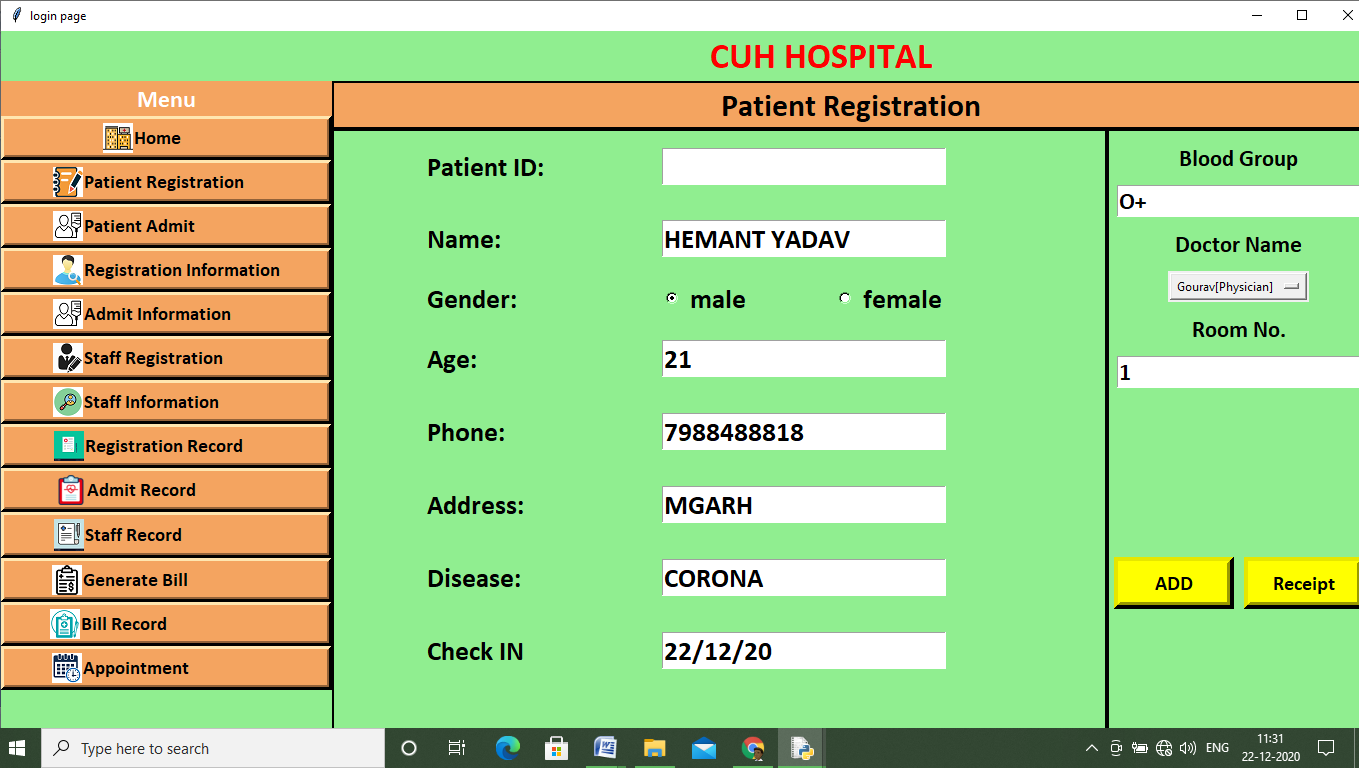


Fig4.4 Frame with data

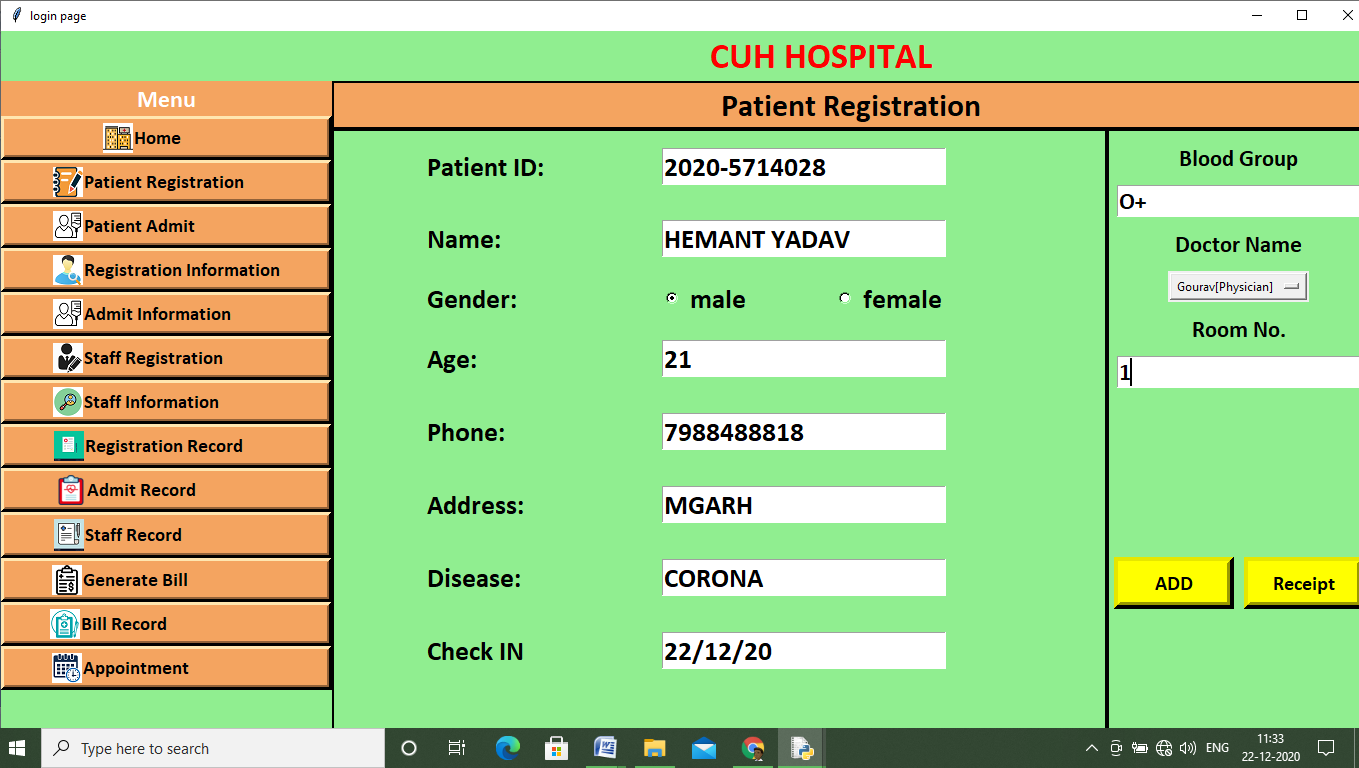


Fig4.5 Data saved

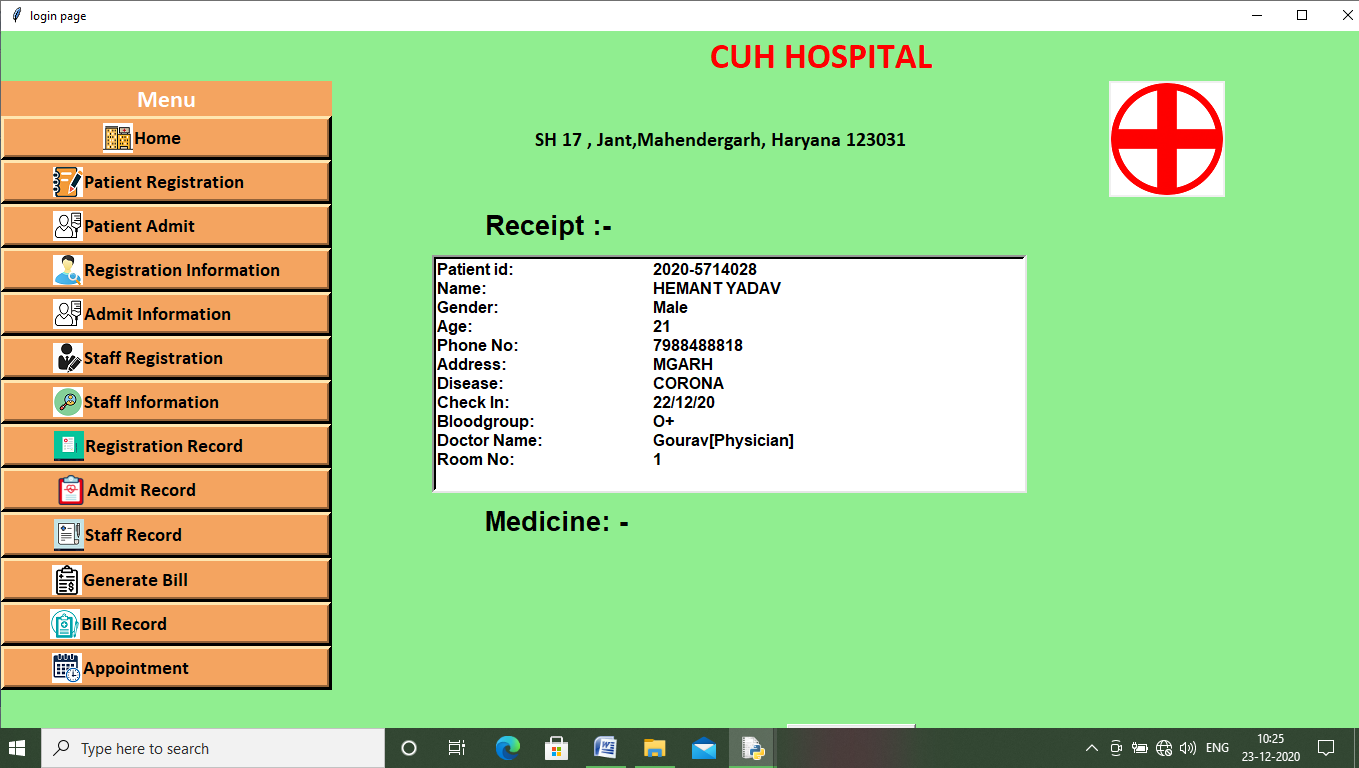


Fig4.6 Receipt generated

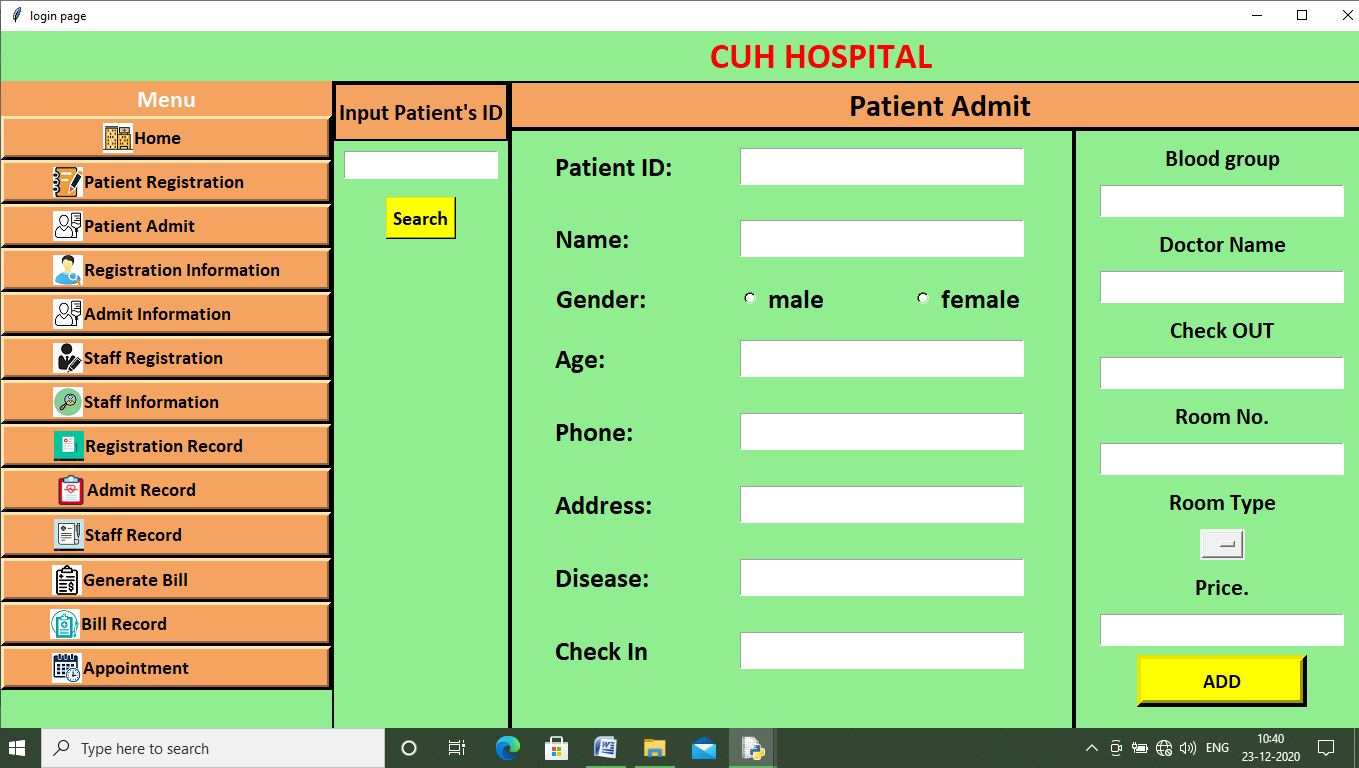


Fig4.7 Patient admit frame

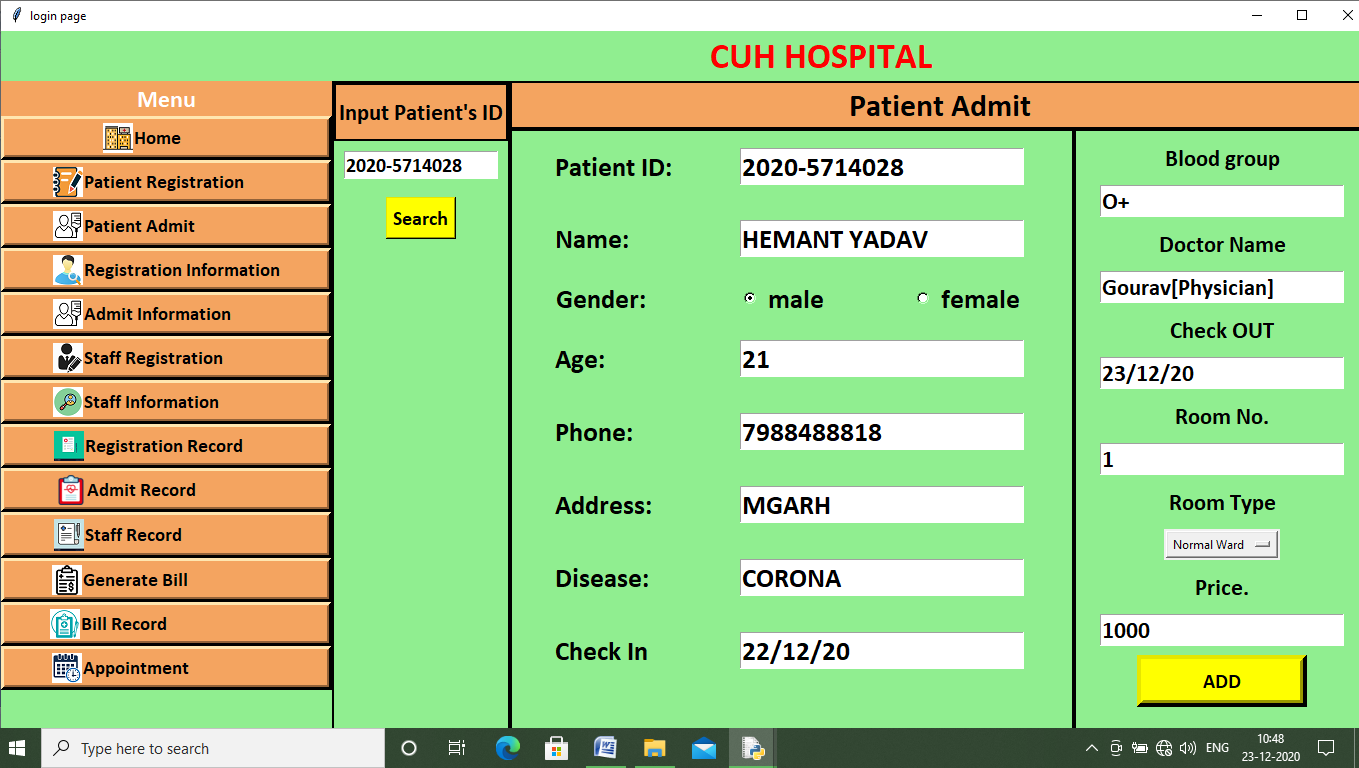


Fig4.8 Frame with data

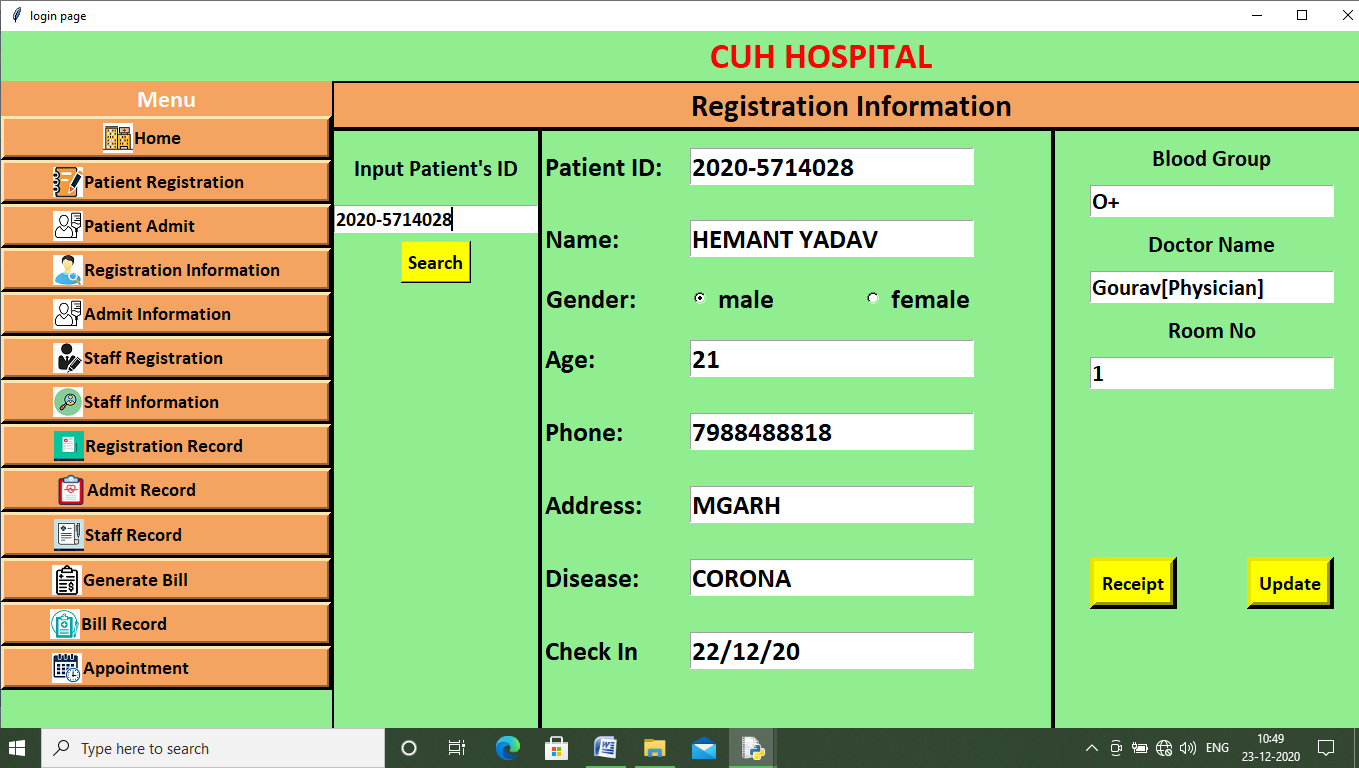


Fig4.9 Patient registration data search

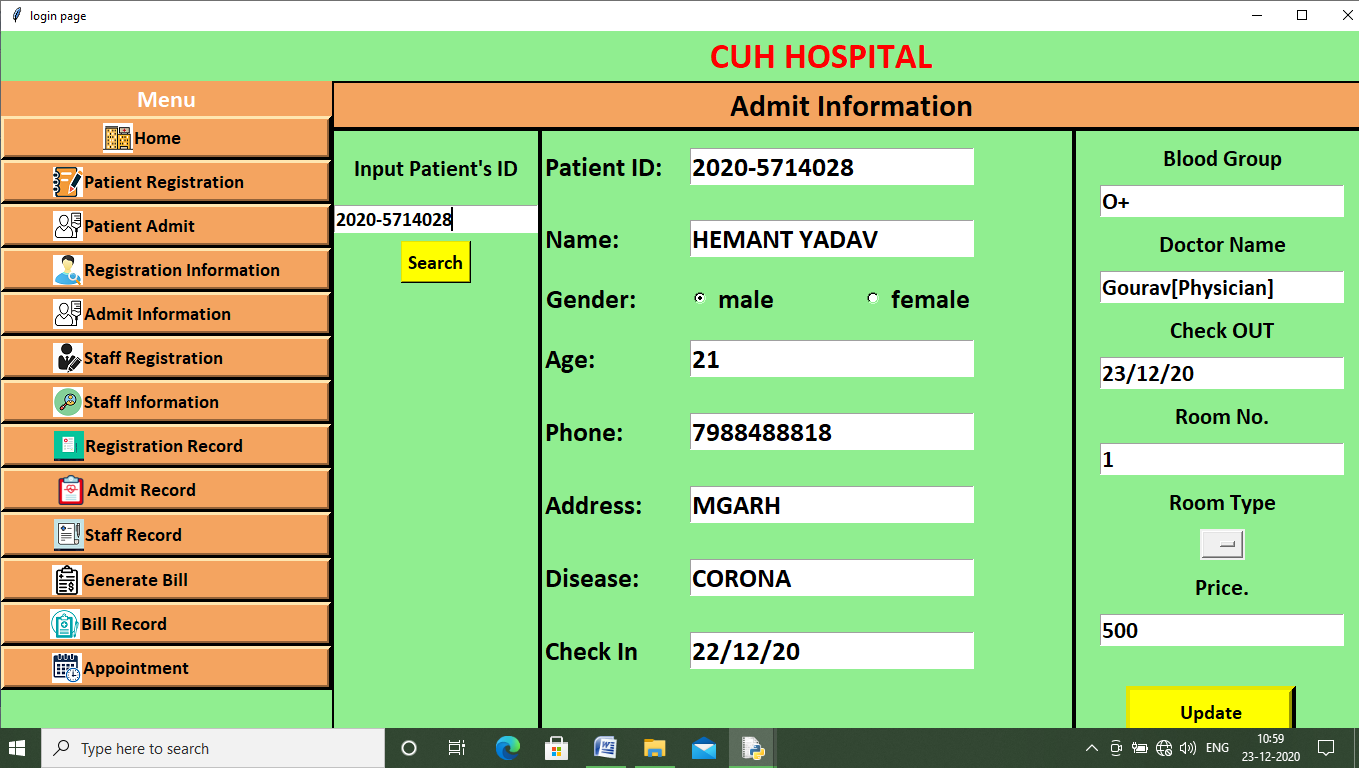


Fig4.10 Information fine out by searching

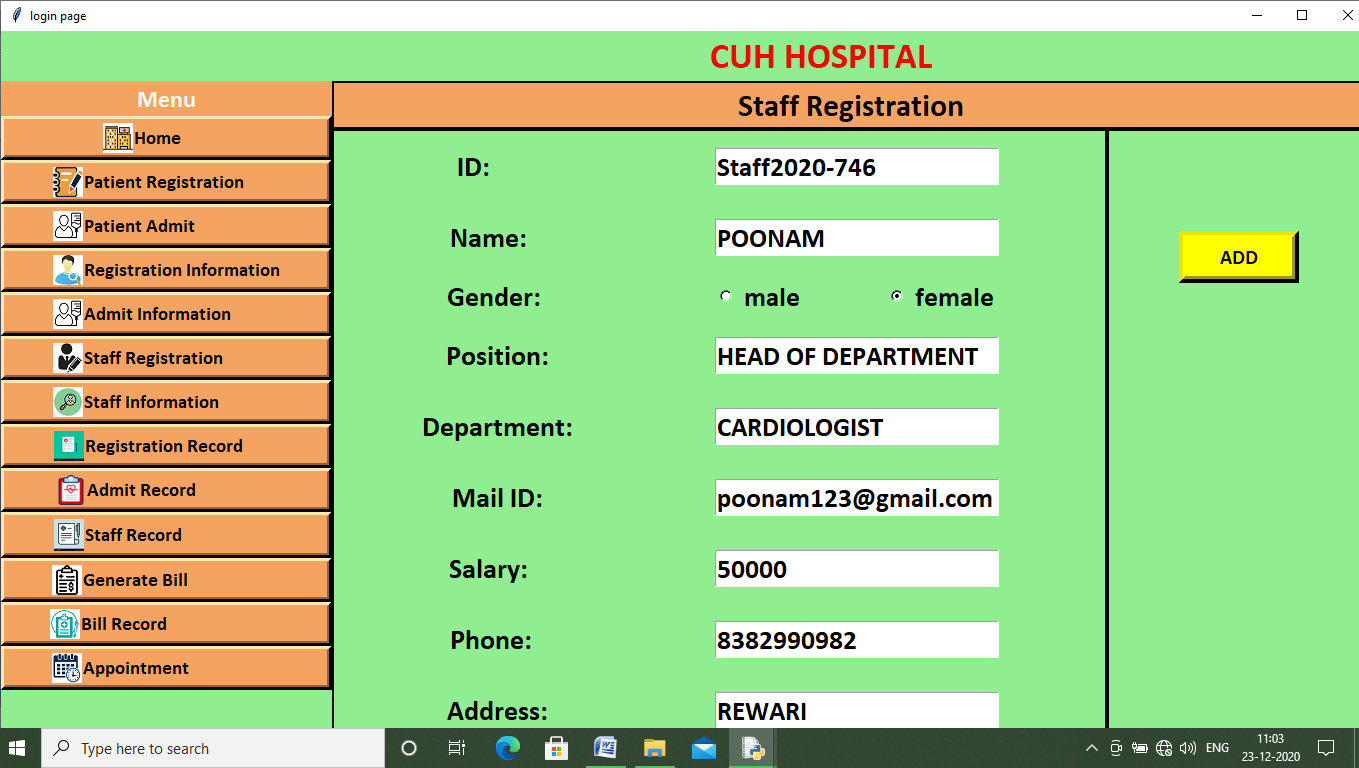


Fig4.11 Data saved

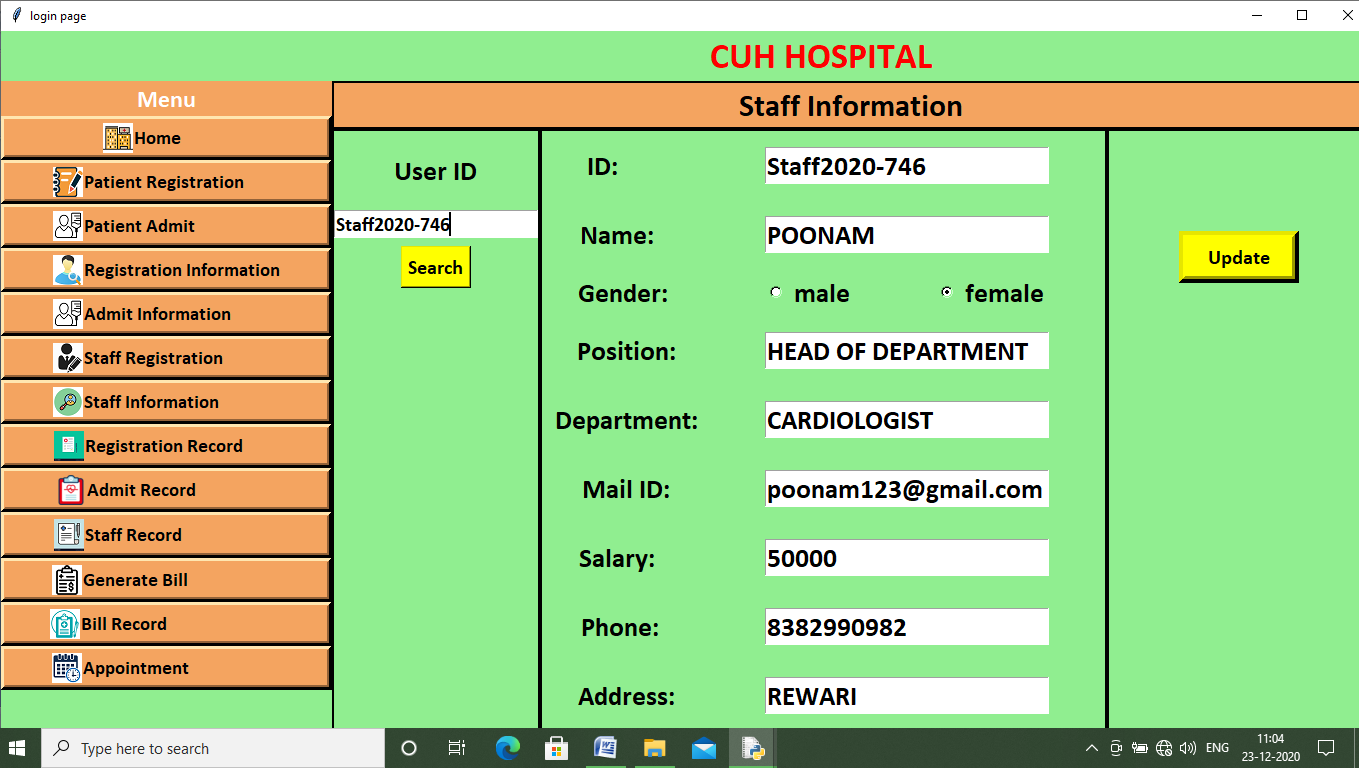


Fig4.12 Data search

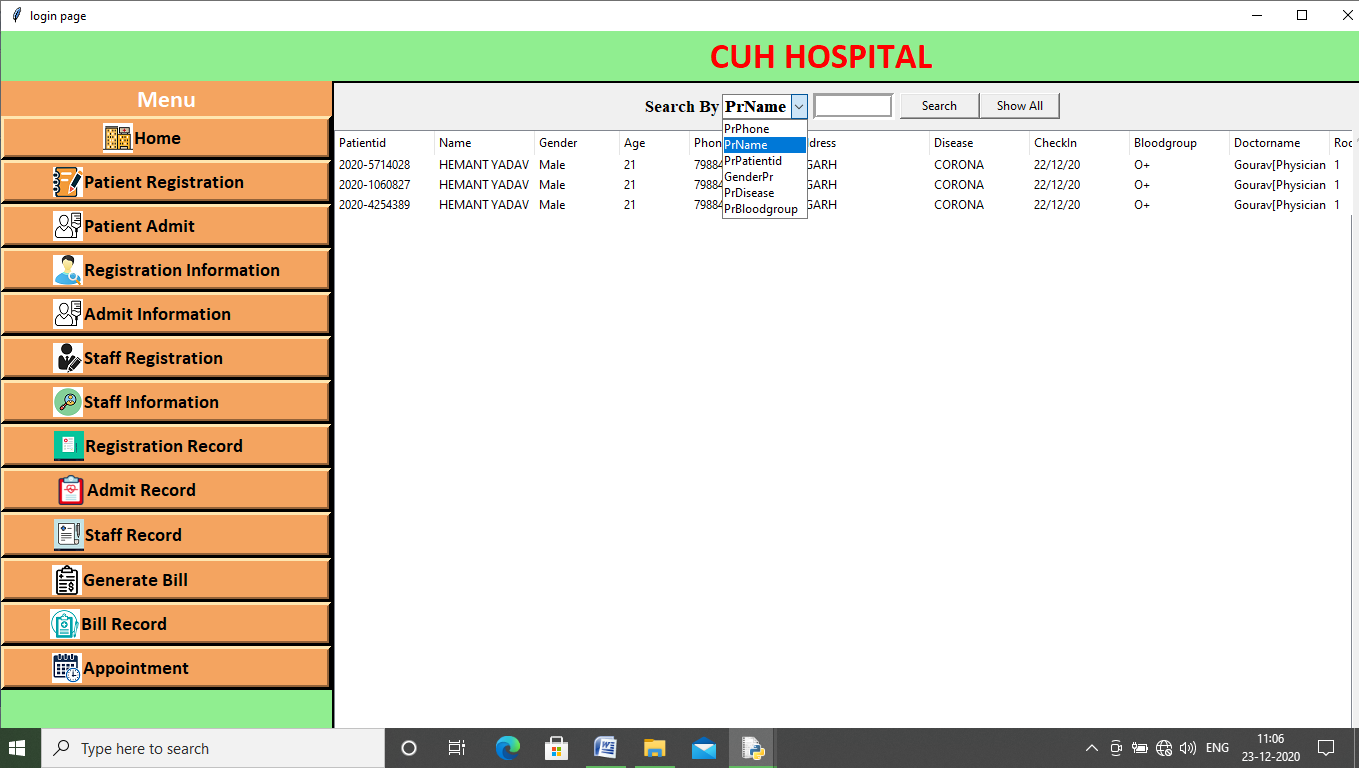


Fig4.13 Data search by name, id

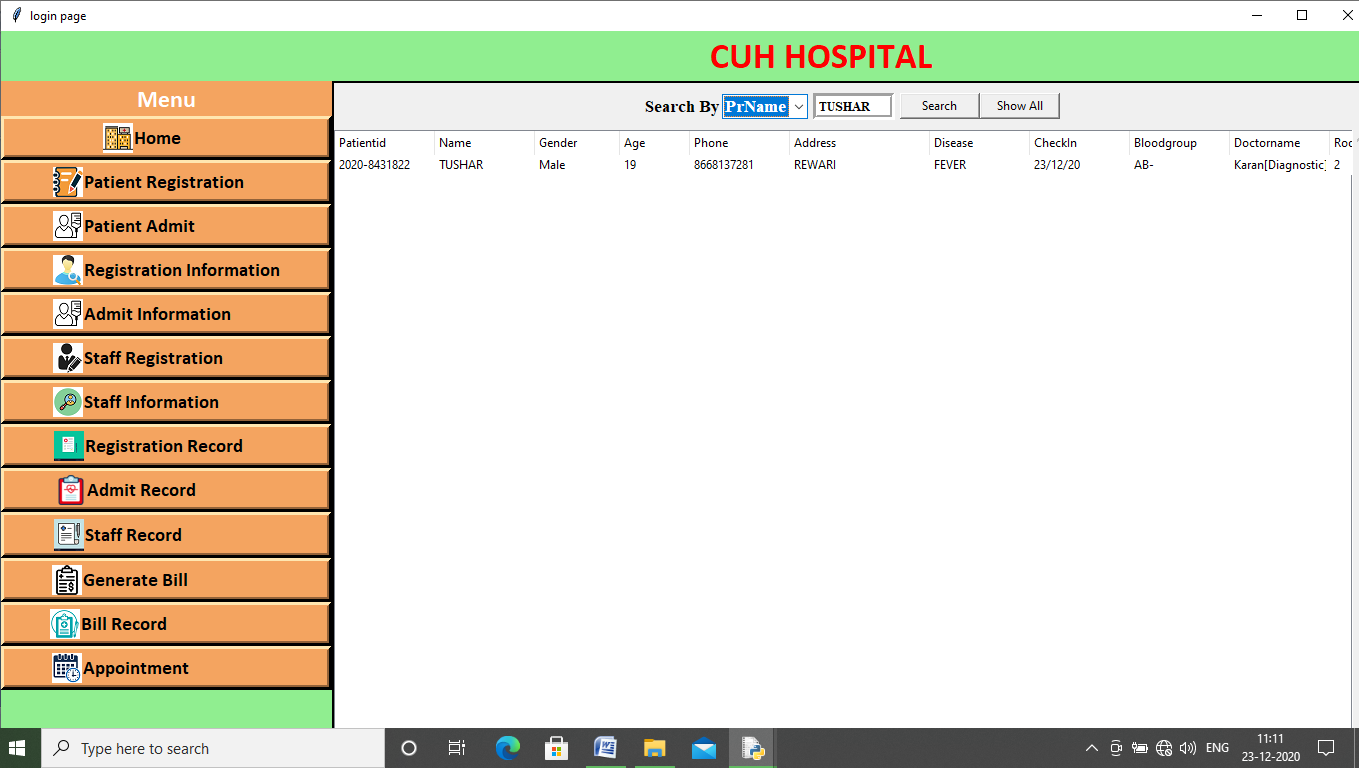


Fig4.14 Patient Data found

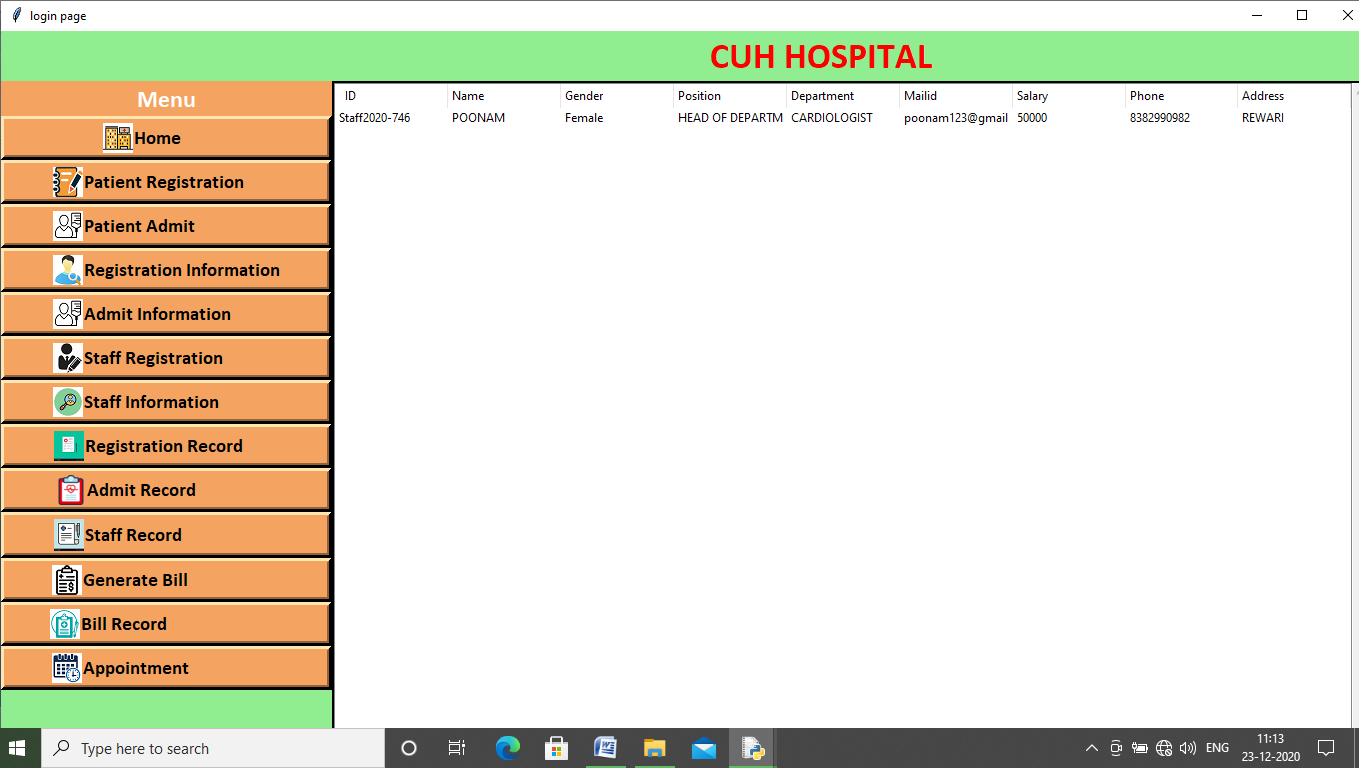


Fig4.15 Staff record found

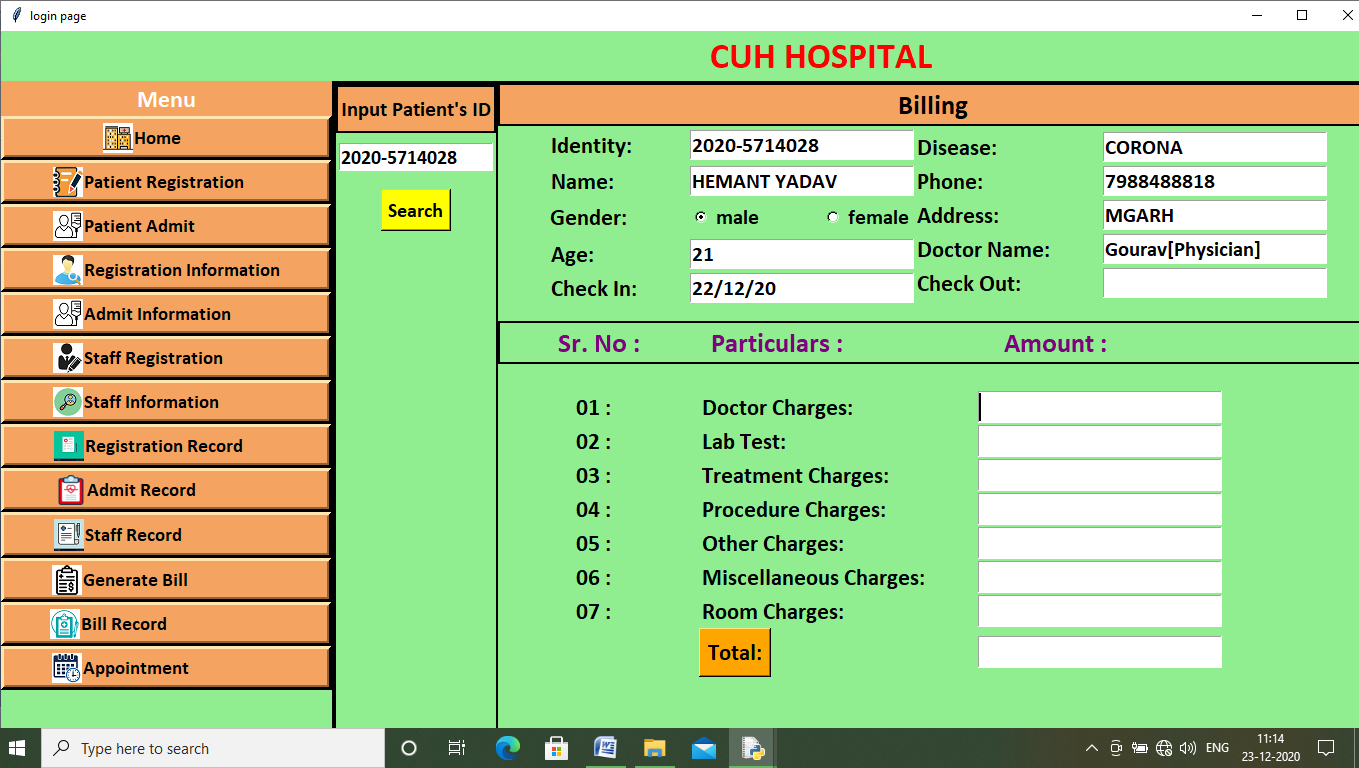


Fig4.16 At billing no need to enter again patient data

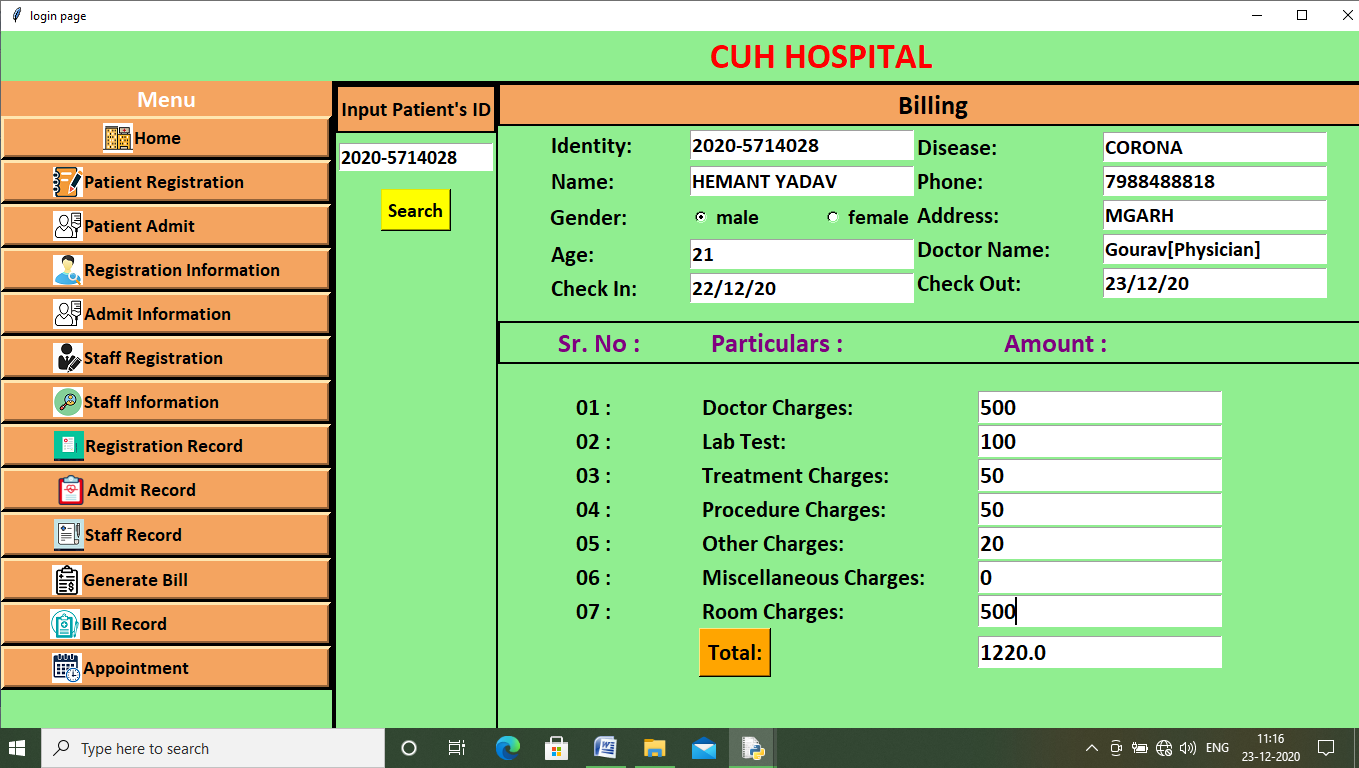


Fig4.17 Automatic addition & Bill generated

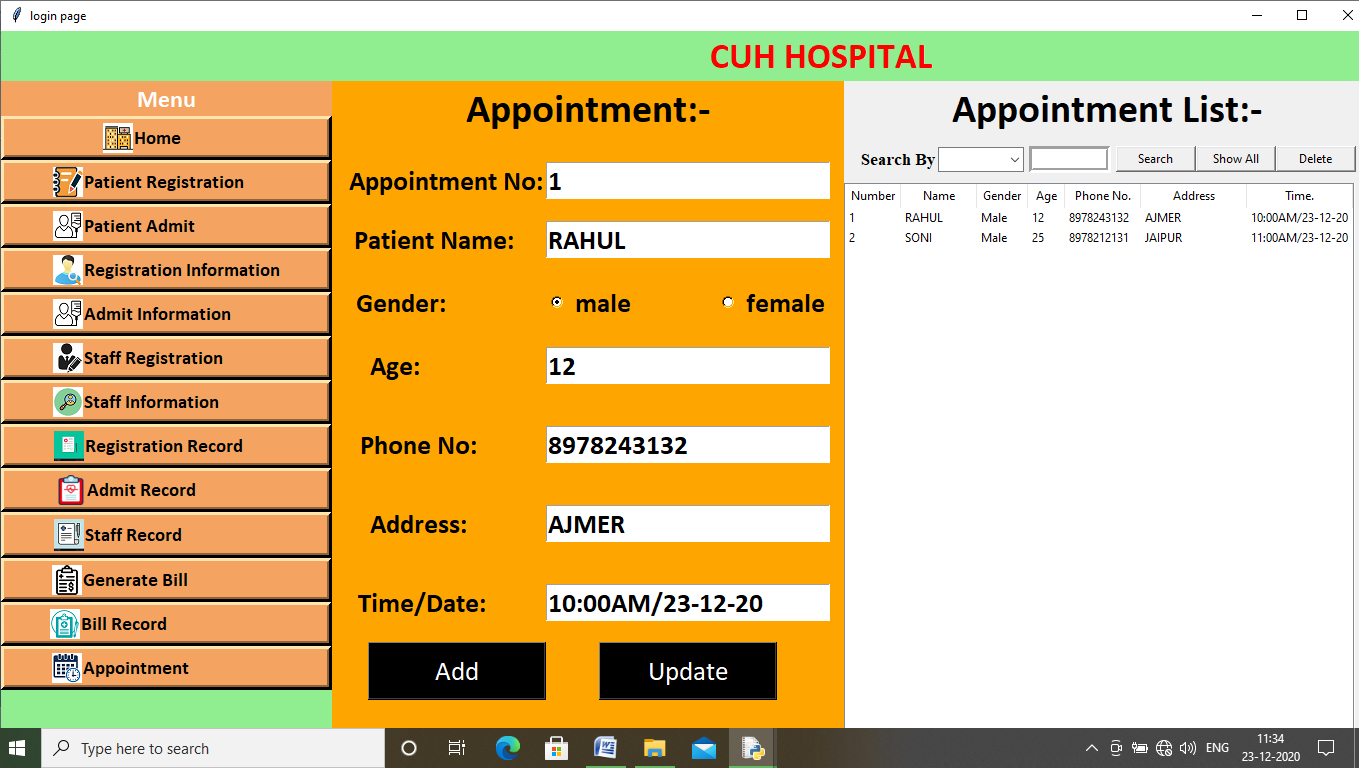


Fig4.18 Appointment

**Advantages**

1. Automatic Patient Id generated

2. Easy to search patient data

3. At search frame no data entry is required

4. Automatic Bill generated.

**Limitation**

1. Not support to all system
2. Always remember Patient Id

**Reference**

1. [www.tutorialspoint.com](http://www.tutorialspoint.com)
2. [www.W3school.com](http://www.W3school.com)

**Contributory Source Code**

from tkinter import \*

import time

import datetime

import tkinter.ttk as ttk

from tkinter import messagebox

import tkinter.messagebox as tkMessageBox

import sqlite3

import random

import tempfile

import os

global framemenuimage

global framehome

global framepatientregistrations

global framepatientadmit

global frameadmitrecord

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framemenuroot

global framerrecord

global framedoctorrecord

global framebill

global framereceipt

global framerireceiptimage

global imagecorner

global BlIdentity

global BlName

global GenderBl

global BlAge

global Blcheckin

global BlDisease

global BlPhone

global BlAddress

global BlDoctorname

global Blcheckout

global BlDoctorcharges

global BlLabtest

global TreatmentchargesBl

global BlProcedurecharges

global BlOthercharges

global BlMiscellaneouscharges

global BlRoomcharges

global BlTotal

global framereceiptimage

global framerireceiptimage

global frameprintbill

global imagemenu

global receiptimage

global receiptimager

global imagehome

global PrPatientid

global PrName

global GenderPr

global PrAge

global PrPhone

global PrAddress

global PrDisease

global PrCheckIn

global PrBloodgroup

global PrDoctorname

global PrRoomNo

global RiPatientid

global RiName

global GenderRi

global RiAge

global RiPhone

global RiAddress

global RiDisease

global RiCheckIn

global RiBloodgroup

global RiDoctorname

global RiRoomNo

global PaPatientid

global PaName

global GenderPa

global PaAge

global PaPhone

global PaAddress

global PaDisease

global PaCheckIn

global PaBloodgroup

global PaDoctorname

global PaCheckOut

global PaRoomNo

global PaRoomtype

global PaPrice

global AiPatientid

global AiName

global GenderAi

global AiAge

global AiPhone

global AiAddress

global AiDisease

global AiCheckIn

global AiBloodgroup

global AiDoctorname

global AiCheckOut

global AiRoomNo

global AiRoomtype

global AiPrice

global SrID

global SrName

global GenderSr

global SrPosition

global SrDepartment

global SrMailid

global SrSalary

global SrPhone

global SrAddress

global SiID

global SiName

global GenderSi

global SiPosition

global SiDepartment

global SiMailid

global SiSalary

global SiPhone

global SiAddress

global Patientregistration\_table

global Patientadmit\_table

global Patient\_tables

global Item1

global Item2

global Item3

global Item4

global Item5

global Item6

global Item7

global OverAllCost

global BlTotal

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global BlrecordIdentity

global BlrecordName

global GenderBlrecord

global BlrecordAge

global Blrecordcheckin

global BlrecordDisease

global BlrecordPhone

global BlrecordAddress

global BlrecordDoctorname

global Blrecordcheckout

global BlrecordDoctorcharges

global BlrecordLabtest

global TreatmentchargesBlrecord

global BlrecordProcedurecharges

global BlrecordOthercharges

global BlrecordMiscellaneouscharges

global BlrecordRoomcharges

global BlrecordTotal

global Blsearch

global frameappointment

global frameappointmentright

global patientnoappointment

global patientnameappointment

global genderappointment

global ageappointment

global phonenoappointment

global addressappointment

global appointmenttime

global tree\_table

global appointment\_table

global PATIENTNOAPPOINTMENT

global PATIENTNAMEAPPOINTMENT

global GENDERAPPOINTMENT

global AGEAPPOINTMENT

global PHONENOAPPOINTMENT

global ADDRESSAPPOINTMENT

global APPOINTMENTTIME

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterhome=0

counterlogin=0

counterpreg=0

counterrinfo=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterbill=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

root= Tk()

root.title("login page")

root.geometry("1700x1000")

PrCheckIn = StringVar()

PrCheckIn.set(time.strftime("%d/%m/%y"))

def login():

global framemenuimage

global framereceiptimage

global framehome

global imagemenu

global imagehome

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framemenuroot

global framestaffregistration

global framestaffinformation

global framerrecord

global framedoctorrecord

global framebill

global framereceipt

global framerireceiptimage

global PrPatientid

global PrName

global GenderPr

global PrAge

global PrPhone

global PrAddress

global PrDisease

global PrCheckIn

global PrBloodgroup

global PrDoctorname

global PrRoomNo

global RiPatientid

global RiName

global GenderRi

global RiAge

global RiPhone

global RiAddress

global RiDisease

global RiCheckIn

global RiBloodgroup

global RiDoctorname

global RiRoomNo

global PaPatientid

global PaName

global GenderPa

global PaAge

global PaPhone

global PaAddress

global PaDisease

global PaCheckIn

global PaBloodgroup

global PaDoctorname

global PaCheckOut

global PaRoomNo

global PaRoomtype

global PaPrice

global AiPatientid

global AiName

global GenderAi

global AiAge

global AiPhone

global AiAddress

global AiDisease

global AiCheckIn

global AiBloodgroup

global AiDoctorname

global AiCheckOut

global AiRoomNo

global AiRoomtype

global AiPrice

global SrID

global SrName

global GenderSr

global SrPosition

global SrDepartment

global SrMailid

global SrSalary

global SrPhone

global SrAddress

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global BlrecordIdentity

global BlrecordName

global GenderBlrecord

global BlrecordAge

global Blrecordcheckin

global BlrecordDisease

global BlrecordPhone

global BlrecordAddress

global BlrecordDoctorname

global Blrecordcheckout

global BlrecordDoctorcharges

global BlrecordLabtest

global TreatmentchargesBlrecord

global BlrecordProcedurecharges

global BlrecordOthercharges

global BlrecordMiscellaneouscharges

global BlrecordRoomcharges

global BlrecordTotal

global Blsearch

global frameappointment

global frameappointmentright

global tree\_table

global appointment\_table

global Patientregistration\_table

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterlogin+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterbill=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

if usernameVar.get()=="1" and passwordVar.get()=="1":

try:

frame1.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterlogin==1:

def total():

Item1=float(BlDoctorcharges.get())

Item2=float(BlLabtest.get())

Item3=float(TreatmentchargesBl.get())

Item4=float(BlProcedurecharges.get())

Item5=float(BlOthercharges.get())

Item6=float(BlMiscellaneouscharges.get())

Item7=float(BlRoomcharges.get())

OverAllCost=str(Item1+Item2+Item3+Item4+Item5+Item6+Item7)

BlTotal.set(OverAllCost)

def Blsearch(ID):

global Blsearch

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from Bl where BlIdentity like '{0}' or BlName like '{0}' or BlPhone like '{0}'".format(ID)

cursor.execute(query)

for row in cursor:

BlrecordIdentity.set(row[0])

BlrecordName.set(row[1])

GenderBlrecord.set(row[2])

BlrecordAge.set(row[3])

Blrecordcheckin.set(row[4])

BlrecordDisease.set(row[5])

BlrecordPhone.set(row[6])

BlrecordAddress.set(row[7])

BlrecordDoctorname.set(row[8])

Blrecordcheckout.set(row[9])

BlrecordDoctorcharges.set(row[10])

BlrecordLabtest.set(row[11])

TreatmentchargesBlrecord.set(row[12])

BlrecordProcedurecharges.set(row[13])

BlrecordOthercharges.set(row[14])

BlrecordMiscellaneouscharges.set(row[15])

BlrecordRoomcharges.set(row[16])

BlrecordTotal.set(row[17])

def savebill():

global framemenuroot

global framemenuimage

global framereceiptimage

global framerireceiptimage

global frameprintbill

global framehome

global BlIdentity

global BlName

global GenderBl

global BlAge

global Blcheckin

global BlDisease

global BlPhone

global BlAddress

global BlDoctorname

global Blcheckout

global BlDoctorcharges

global BlLabtest

global TreatmentchargesBl

global BlProcedurecharges

global BlOthercharges

global BlMiscellaneouscharges

global BlRoomcharges

global BlTotal

global imagemenu

global imagehome

global framereceipt

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framebill

global framerrecord

global framedoctorrecord

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global BlrecordIdentity

global BlrecordName

global GenderBlrecord

global BlrecordAge

global Blrecordcheckin

global BlrecordDisease

global BlrecordPhone

global BlrecordAddress

global BlrecordDoctorname

global Blrecordcheckout

global BlrecordDoctorcharges

global BlrecordLabtest

global TreatmentchargesBlrecord

global BlrecordProcedurecharges

global BlrecordOthercharges

global BlrecordMiscellaneouscharges

global BlrecordRoomcharges

global BlrecordTotal

global Blsearch

global frameappointment

global frameappointmentright

global appointment\_table

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

if BlIdentity.get() == "" or BlName.get() == "" or GenderBl.get() == "" or int(BlAge.get()) == "" or Blcheckin.get() == "" or BlDisease.get() == "" or int(BlPhone.get()) == "" or BlDoctorname.get() == "" or Blcheckout.get() == "" or int(BlDoctorcharges.get()) == "" or int(BlLabtest.get()) == "" or int(TreatmentchargesBl.get()) == "" or int(BlProcedurecharges.get()) == "" or int(BlOthercharges.get()) == "" or int(BlMiscellaneouscharges.get()) == "" or int(BlRoomcharges.get()) == "":

result = tkMessageBox.showwarning('', 'Please Complete The Required Field', icon="warning")

else:

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query = "insert into bl values('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}')".format(BlIdentity.get(),BlName.get(),GenderBl.get(),BlAge.get(),Blcheckin.get(),BlDisease.get(),BlPhone.get(),BlAddress.get(),BlDoctorname.get(),Blcheckout.get(),BlDoctorcharges.get(),BlLabtest.get(),TreatmentchargesBl.get(),BlProcedurecharges.get(),BlOthercharges.get(),BlMiscellaneouscharges.get(),BlRoomcharges.get(),BlTotal.get())

cursor.execute(query)

connection.commit()

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from bl"

cursor.execute(query)

messagebox.showinfo("Congratulations","Data Added Successfully")

def SubmitAppointment():

appointment\_table.delete(\*appointment\_table.get\_children())

connection = sqlite3.connect("HEMANT.db")

cursor = connection.cursor()

cursor.execute("INSERT INTO ap (patientnoappointment,patientnameappointment,genderappointment,ageappointment,phonenoappointment,addressappointment,appointmenttime) VALUES(?, ?, ?, ?, ?, ?,?)", (str(PATIENTNOAPPOINTMENT.get()),str(PATIENTNAMEAPPOINTMENT.get()), str(GENDERAPPOINTMENT.get()), str(AGEAPPOINTMENT.get()), str(PHONENOAPPOINTMENT.get()), str(ADDRESSAPPOINTMENT.get()), str(APPOINTMENTTIME.get())))

connection.commit()

cursor.execute("SELECT \* FROM ap ORDER BY `patientnoappointment` ASC")

fetch = cursor.fetchall()

for data in fetch:

appointment\_table.insert('', 'end', values=(data))

cursor.close()

connection.close()

PATIENTNOAPPOINTMENT.set("")

PATIENTNAMEAPPOINTMENT.set("")

GENDERAPPOINTMENT.set("")

AGEAPPOINTMENT.set("")

PHONENOAPPOINTMENT.set("")

ADDRESSAPPOINTMENT.set("")

APPOINTMENTTIME.set("")

def selecteditem(ry):

cursor\_row=appointment\_table.focus()

contents=(appointment\_table.item(cursor\_row))

row=contents['values']

PATIENTNOAPPOINTMENT.set(row[0])

PATIENTNAMEAPPOINTMENT.set(row[1])

GENDERAPPOINTMENT.set(row[2])

AGEAPPOINTMENT.set(row[3])

PHONENOAPPOINTMENT.set(row[4])

ADDRESSAPPOINTMENT.set(row[5])

APPOINTMENTTIME.set(row[6])

def DeleteData():

if not appointment\_table.selection():

result = tkMessageBox.showwarning('', 'Please Select Something First!', icon="warning")

else:

result = tkMessageBox.askquestion('', 'Are you sure you want to delete this record?', icon="warning")

if result == 'yes':

curItem = appointment\_table.focus()

contents =(appointment\_table.item(curItem))

selecteditem = contents['values']

appointment\_table.delete(curItem)

connection = sqlite3.connect("HEMANT.db")

cursor = connection.cursor()

cursor.execute("DELETE FROM ap WHERE `patientnoappointment` = %d" % selecteditem[0])

connection.commit()

cursor.close()

connection.close()

def fetchappointment\_data():

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from ap"

cursor.execute(query)

rows=cursor.fetchall()

if len(rows)!=0:

appointment\_table.delete(\*appointment\_table.get\_children())

for row in rows:

appointment\_table.insert('',END,values=row)

connection.commit()

connection.close()

def search\_data():

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

cursor.execute("select \* from ap where "+str(search\_by.get())+" Like '%"+str(search\_txt.get())+"%'")

rows=cursor.fetchall()

if len(rows)!=0:

appointment\_table.delete(\*appointment\_table.get\_children())

for row in rows:

appointment\_table.insert('',END,values=row)

connection.commit()

connection.close()

def updateappointment():

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="update ap set patientnameappointment='{}', genderappointment='{}',ageappointment='{}',phonenoappointment='{}',addressappointment='{}',appointmenttime='{}' where patientnoappointment='{}'".format(PATIENTNAMEAPPOINTMENT.get(),GENDERAPPOINTMENT.get(),AGEAPPOINTMENT.get(),PHONENOAPPOINTMENT.get(),ADDRESSAPPOINTMENT.get(),APPOINTMENTTIME.get(),PATIENTNOAPPOINTMENT.get())

cursor.execute(query)

connection.commit()

fetchappointment\_data()

messagebox.showinfo("Congratulations ", "Data Updated Successfully")

def Billsearch(ID):

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pr where PrPatientid like '{0}' or PrName like '{0}' or PrPhone like '{0}'".format(ID)

cursor.execute(query)

for row in cursor:

BlIdentity.set(row[0])

BlName.set(row[1])

GenderBl.set(row[2])

BlAge.set(row[3])

BlPhone.set(row[4])

BlAddress.set(row[5])

BlDisease.set(row[6])

Blcheckin.set(row[7])

BlDoctorname.set(row[9])

def appointment():

global framemenuroot

global framemenuimage

global framereceiptimage

global framerireceiptimage

global frameprintbill

global framehome

global BlIdentity

global BlName

global GenderBl

global BlAge

global Blcheckin

global BlDisease

global BlPhone

global BlAddress

global BlDoctorname

global Blcheckout

global BlDoctorcharges

global BlLabtest

global TreatmentchargesBl

global BlProcedurecharges

global BlOthercharges

global BlMiscellaneouscharges

global BlRoomcharges

global BlTotal

global imagemenu

global imagehome

global framereceipt

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framebill

global framerrecord

global framedoctorrecord

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global framebillrecord

global BlrecordIdentity

global BlrecordName

global GenderBlrecord

global BlrecordAge

global Blrecordcheckin

global BlrecordDisease

global BlrecordPhone

global BlrecordAddress

global BlrecordDoctorname

global Blrecordcheckout

global BlrecordDoctorcharges

global BlrecordLabtest

global TreatmentchargesBlrecord

global BlrecordProcedurecharges

global BlrecordOthercharges

global BlrecordMiscellaneouscharges

global BlrecordRoomcharges

global BlrecordTotal

global Blsearch

global frameappointment

global frameappointmentright

global patientnoappointment

global patientnameappointment

global genderappointment

global ageappointment

global phonenoappointment

global addressappointment

global appointmenttime

global appointment\_table

global search\_by

global search\_txt

global PATIENTNOAPPOINTMENT

global PATIENTNAMEAPPOINTMENT

global GENDERAPPOINTMENT

global AGEAPPOINTMENT

global PHONENOAPPOINTMENT

global ADDRESSAPPOINTMENT

global APPOINTMENTTIME

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterappointment+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbill=0

counterbillrecord=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

frame1.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

if counterappointment==1:

frameappointment=Frame(root,bg="orange")

frameappointment.pack(side=LEFT,fill=BOTH,expand=True)

frameappointmentleft=Frame(frameappointment,height=100,width=500)

frameappointmentleft.pack(side=TOP)

labelappointment=Label(frameappointmentleft,text="Appointment:-",bg="orange",fg="black",font=("calibri",30,"bold"))

labelappointment.pack(anchor="nw")

frameappointmentlef=Frame(frameappointment,bg="orange",height=500,width=500)

frameappointmentlef.pack(side=TOP,pady=25)

patintnoappointment=Label(frameappointmentlef,text="Appointment No:",bg="orange",font=("calibri",20,"bold"))

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

PATIENTNOAPPOINTMENT=StringVar()

entrypatientnoappointment=Entry(frameappointmentlef,textvariable=PATIENTNOAPPOINTMENT,font=("calibri",20,"bold"))

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

patientnameappointment=Label(frameappointmentlef,text="Patient Name: ",bg="orange",font=("calibri",20,"bold"))

patientnameappointment.grid(row=1,column=0,pady=20)

PATIENTNAMEAPPOINTMENT=StringVar()

entrymenucode=Entry(frameappointmentlef,textvariable=PATIENTNAMEAPPOINTMENT,font=("calibri",20,"bold"))

entrymenucode.grid(row=1,column=1,pady=20)

GENDERAPPOINTMENT=StringVar()

GENDERAPPOINTMENT.set("")

GENDERAPPOINTMENT.set(3)

labelGender=Label(frameappointmentlef,text="Gender: ",bg="orange",font=("calibri",20,"bold"))

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

i= IntVar()

r1 = Radiobutton(frameappointmentlef,text="male",value='Male',variable=GENDERAPPOINTMENT,bg="orange",font=("calibri",20,"bold"))

r2 = Radiobutton(frameappointmentlef,text="female",value='Female',variable=GENDERAPPOINTMENT,bg="orange",font=("calibri",20,"bold"))

r1.grid(row=2,column=1,sticky="w")

r2.grid(row=2,column=1,sticky="e",padx=0)

labelItemName=Label(frameappointmentlef,text="Age: ",bg="orange",font=("calibri",20,"bold"))

labelItemName.grid(row=3,column=0,pady=20)

AGEAPPOINTMENT=StringVar()

entryitemname=Entry(frameappointmentlef,textvariable=AGEAPPOINTMENT,font=("calibri",20,"bold"))

entryitemname.grid(row=3,column=1,pady=20)

labelQuantity=Label(frameappointmentlef,text="Phone No: ",bg="orange",font=("calibri",20,"bold"))

labelQuantity.grid(row=4,column=0,pady=20)

PHONENOAPPOINTMENT=StringVar()

entryquantity=Entry(frameappointmentlef,textvariable=PHONENOAPPOINTMENT,font=("calibri",20,"bold"))

entryquantity.grid(row=4,column=1,pady=20)

labelItemName=Label(frameappointmentlef,text="Address: ",bg="orange",font=("calibri",20,"bold"))

labelItemName.grid(row=5,column=0,pady=20)

ADDRESSAPPOINTMENT=StringVar()

entryitemname=Entry(frameappointmentlef,textvariable=ADDRESSAPPOINTMENT,font=("calibri",20,"bold"))

entryitemname.grid(row=5,column=1,pady=20)

labelGender=Label(frameappointmentlef,text="Time/Date: ",bg="orange",font=("calibri",20,"bold"))

labelGender.grid(row=6,column=0,pady=20)

APPOINTMENTTIME=StringVar()

entryfname=Entry(frameappointmentlef,textvariable=APPOINTMENTTIME,font=("calibri",20,"bold"))

entryfname.grid(row=6,column=1,pady=20)

buttonADD=Button(frameappointmentlef,text="Add",command=SubmitAppointment,bg="black",fg="white",width="12",font=("calibri",20))

buttonADD.grid(row=7,column=0,sticky="e")

buttonUpdate=Button(frameappointmentlef,text="Update",command=updateappointment,bg="black",fg="white",width="12",font=("calibri",20))

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

frameappointmentright=Frame(root,width=800,height=1000)

frameappointmentright.pack(side=RIGHT,fill=BOTH,expand=True)

frameappointmentrecord=Frame(frameappointmentright,width=800,height=60)

frameappointmentrecord.pack(side=TOP)

Edititem=Label(frameappointmentrecord,text="Appointment List:-",fg="black",font=("calibri",30,"bold"))

Edititem.pack(anchor="nw",padx=90)

frameappointmentrightrecord=Frame(frameappointmentright,bg="black",width=650,height=500)

frameappointmentrightrecord.pack(side=TOP)

frameButton=Frame(frameappointmentrightrecord)

frameButton.grid()

lbl\_search=Label(frameButton,text="Search By",fg="black",font=("times new roman",13,"bold"))

lbl\_search.grid(row=0,column=0,pady=7,sticky="w")

search\_by=StringVar()

combo\_search=ttk.Combobox(frameButton,textvariable=search\_by,width=7,font=("times new roman",13,"bold"),state='readonly')

combo\_search['values']=("phonenoappointment","patientnameappointment","patientnoappointment")

combo\_search.grid(row=0,column=1,pady=10)

search\_txt=StringVar()

txt\_search=Entry(frameButton,textvariable=search\_txt,font=("times new roman",10,"bold"),bd=5,relief=GROOVE,width=10)

txt\_search.grid(row=0,column=2,padx=5,pady=10,sticky="w")

searchbtn=Button(frameButton,text="Search",command=search\_data,width=10).grid(row=0,column=3,pady=10)

showallbtn=Button(frameButton,text="Show All",width=10,command=fetchappointment\_data).grid(row=0,column=4,pady=10)

showallbtn=Button(frameButton,text="Delete",width=10,command=DeleteData).grid(row=0,column=5,pady=10)

scroll1=Scrollbar(frameappointmentright,orient=HORIZONTAL)

scroll1.pack(side=BOTTOM,fill=X)

scroll2=Scrollbar(frameappointmentright,orient=VERTICAL)

scroll2.pack(side=RIGHT,fill=Y)

appointment\_table=ttk.Treeview(frameappointmentright,columns=("Patientnoappointment","Patientnameappointment","Genderappointment","Ageappointment","Phonenoappointment","Addressappointment","Appointmenttime"),xscrollcommand=scroll1.set,yscrollcommand=scroll2.set)

scroll1.config(command=appointment\_table.xview)

scroll2.config(command=appointment\_table.yview)

appointment\_table.heading("Patientnoappointment",text="Number")

appointment\_table.heading("Patientnameappointment",text="Name")

appointment\_table.heading("Genderappointment",text="Gender")

appointment\_table.heading("Ageappointment",text="Age")

appointment\_table.heading("Phonenoappointment",text="Phone No.")

appointment\_table.heading("Addressappointment",text="Address")

appointment\_table.heading("Appointmenttime",text="Time.")

appointment\_table['show']='headings'

appointment\_table.column("Patientnoappointment",width=50)

appointment\_table.column("Patientnameappointment",width=70)

appointment\_table.column("Genderappointment",width=45)

appointment\_table.column("Ageappointment",width=30)

appointment\_table.column("Phonenoappointment",width=70)

appointment\_table.column("Addressappointment",width=100)

appointment\_table.column("Appointmenttime",width=100)

appointment\_table.pack(fill=BOTH,expand=True)

fetchappointment\_data()

appointment\_table.bind("<ButtonRelease-1>",selecteditem)

def billrecord():

global framemenuroot

global framemenuimage

global framereceiptimage

global framerireceiptimage

global frameprintbill

global framehome

global BlIdentity

global BlName

global GenderBl

global BlAge

global Blcheckin

global BlDisease

global BlPhone

global BlAddress

global BlDoctorname

global Blcheckout

global BlDoctorcharges

global BlLabtest

global TreatmentchargesBl

global BlProcedurecharges

global BlOthercharges

global BlMiscellaneouscharges

global BlRoomcharges

global BlTotal

global imagemenu

global imagehome

global framereceipt

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framebill

global framerrecord

global framedoctorrecord

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global BlrecordIdentity

global BlrecordName

global GenderBlrecord

global BlrecordAge

global Blrecordcheckin

global BlrecordDisease

global BlrecordPhone

global BlrecordAddress

global BlrecordDoctorname

global Blrecordcheckout

global BlrecordDoctorcharges

global BlrecordLabtest

global TreatmentchargesBlrecord

global BlrecordProcedurecharges

global BlrecordOthercharges

global BlrecordMiscellaneouscharges

global BlrecordRoomcharges

global BlrecordTotal

global Blsearch

global frameappointment

global frameappointmentright

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterbillrecord+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbill=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

frame1.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterbillrecord==1:

framebillrecord=Frame(root,relief="solid",borderwidth=2,bg="lightgreen")

framebillrecord.pack(side="right",fill=BOTH,expand=True)

frameybillr=Frame(framebillrecord,bg="sandybrown",relief="solid",borderwidth=2)

frameybillr.pack(side=TOP,fill=X)

labelbill=Label(frameybillr,text="Bill Record",bg="sandybrown",fg="black",font=("calibri",20,"bold"))

labelbill.pack()

frameq=Frame(framebillrecord,width=100,relief="solid",borderwidth=2,bg="lightgreen")

frameq.pack(side="left",fill=Y)

labelinputid=Label(frameq,text="Input Patient id",fg="black",bg="lightgreen",font=("calibri",20,"bold"))

labelinputid.pack(side=TOP)

searchbillrecord=StringVar()

Username=Entry(frameq,width="17",textvariable=searchbillrecord,font=("calibri",15,"bold"))

Username.pack()

Searchbutton=Button(frameq,text="Search",command=lambda:Blsearch(searchbillrecord.get()),bg="yellow",font=("calibri",15,"bold"))

Searchbutton.pack(side=TOP,pady=8)

framezbill=Frame(framebillrecord,bg="sandybrown",relief="solid",borderwidth=2)

framezbill.pack(side=TOP,fill=X)

frametbill=Frame(framebillrecord,height="200",bg="lightgreen")

frametbill.pack(side=TOP,fill=X)

framebrleft=Frame(frametbill,height="200",width="685",bg="lightgreen")

framebrleft.pack(side=LEFT)

labelPatient\_id=Label(framebrleft,text="Identity:",bg="lightgreen",font=("calibri",16,"bold"))

labelPatient\_id.grid(row=0,column=0,sticky="w",padx="30",pady=2)

BlrecordIdentity=StringVar()

Identity=Entry(framebrleft,textvariable=BlrecordIdentity,font=("calibri",16,"bold"))

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

labelName=Label(framebrleft,text="Name:",bg="lightgreen",font=("calibri",16,"bold"))

labelName.grid(row=1,column=0,sticky="w",padx="30")

BlrecordName=StringVar()

Name=Entry(framebrleft,textvariable=BlrecordName,font=("calibri",16,"bold"))

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

GenderBlrecord=StringVar()

GenderBlrecord.set("")

labelGender=Label(framebrleft,text="Gender: ",bg="lightgreen",font=("calibri",16,"bold"))

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

i= IntVar()

r1 = Radiobutton(framebrleft,text="male",value='Male',variable=GenderBlrecord,bg="lightgreen",font=("calibri",16,"bold"))

r2 = Radiobutton(framebrleft,text="female",value='Female',variable=GenderBlrecord,bg="lightgreen",font=("calibri",16,"bold"))

r1.grid(row=2,column=1,sticky="w")

r2.grid(row=2,column=1,sticky="e",padx=0)

GenderBlrecord.set(3)

labelAge=Label(framebrleft,text="Age:",bg="lightgreen",font=("calibri",16,"bold"))

labelAge.grid(row=3,column=0,sticky="w",padx="30")

BlrecordAge=StringVar()

Age=Entry(framebrleft,textvariable=BlrecordAge,font=("calibri",16,"bold"))

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

labelCheckIn=Label(framebrleft,text="Check In:",bg="lightgreen",font=("calibri",16,"bold"))

labelCheckIn.grid(row=4,column=0,sticky="w",padx="30")

Blrecordcheckin=StringVar()

checkin=Entry(framebrleft,textvariable=Blrecordcheckin,font=("calibri",16,"bold"))

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

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

framebrights=Frame(frametbill,height="200",bg="lightgreen",width="685")

framebrights.pack(side=RIGHT)

labelAge=Label(framebrights,text="Disease:",bg="lightgreen",font=("calibri",16,"bold"))

labelAge.grid(row=0,column=0,sticky="w",padx=20)

BlrecordDisease=StringVar()

Disease=Entry(framebrights,textvariable=BlrecordDisease,font=("calibri",16,"bold"))

Disease.grid(row=0,column=1,padx="0")

labelPhone=Label(framebrights,text="Phone:",bg="lightgreen",font=("calibri",16,"bold"))

labelPhone.grid(row=1,column=0,sticky="w",padx=20)

BlrecordPhone=StringVar()

Phone=Entry(framebrights,textvariable=BlrecordPhone,font=("calibri",16,"bold"))

Phone.grid(row=1,column=1,padx="30")

labelAddress=Label(framebrights,text="Address:",bg="lightgreen",font=("calibri",16,"bold"))

labelAddress.grid(row=2,column=0,sticky="w",padx=20)

BlrecordAddress=StringVar()

Address=Entry(framebrights,textvariable=BlrecordAddress,font=("calibri",16,"bold"))

Address.grid(row=2,column=1,padx="30")

labelDisease=Label(framebrights,text="Doctor Name:",bg="lightgreen",font=("calibri",16,"bold"))

labelDisease.grid(row=3,column=0,sticky="w",padx=20)

BlrecordDoctorname=StringVar()

Doctorname=Entry(framebrights,textvariable=BlrecordDoctorname,font=("calibri",16,"bold"))

Doctorname.grid(row=3,column=1,padx="30")

labelCheckOut=Label(framebrights,text="Check Out:",bg="lightgreen",font=("calibri",16,"bold"))

labelCheckOut.grid(row=4,column=0,sticky="w",padx=20)

Blrecordcheckout=StringVar()

checkout=Entry(framebrights,textvariable=Blrecordcheckout,font=("calibri",16,"bold"))

checkout.grid(row=4,column=1,padx="30")

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

framebottomr=Frame(framebillrecord,relief="solid",bg="lightgreen",borderwidth=2)

framebottomr.pack(side=TOP,fill=X,pady=16)

labelPatient\_id=Label(framebottomr,text="Sr. No :",bg="lightgreen",fg="Purple",font=("calibri",20,"bold"))

labelPatient\_id.grid(row=0,column=0,sticky="w",padx=60)

labelPatient\_id=Label(framebottomr,text="Particulars :",bg="lightgreen",fg="Purple",font=("calibri",20,"bold"))

labelPatient\_id.grid(row=0,column=1,sticky="w",padx=60)

labelPatient\_id=Label(framebottomr,text="Amount :",bg="lightgreen",fg="Purple",font=("calibri",20,"bold"))

labelPatient\_id.grid(row=0,column=2,sticky="e",padx=130)

root.grid\_rowconfigure(0,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

root.grid\_columnconfigure(2,weight=1)

framebox=Frame(framebillrecord,bg="lightgreen")

framebox.pack(side=TOP,fill=BOTH,expand=True,pady="10")

labelPatient\_id=Label(framebox,text="01 :",bg="lightgreen",font=("calibri",17,"bold"))

labelPatient\_id.grid(row=0,column=0,sticky="w",padx=70)

labelPatient\_id=Label(framebox,text="Doctor Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelPatient\_id.grid(row=0,column=1,sticky="w",padx=45)

BlrecordDoctorcharges=StringVar()

Doctorcharges=Entry(framebox,textvariable=BlrecordDoctorcharges,font=("calibri",17,"bold"))

Doctorcharges.grid(row=0,column=2,sticky="w")

labelName=Label(framebox,text="02 :",bg="lightgreen",font=("calibri",17,"bold"))

labelName.grid(row=1,column=0,sticky="w",padx=70)

labelName=Label(framebox,text="Lab Test:",bg="lightgreen",font=("calibri",17,"bold"))

labelName.grid(row=1,column=1,sticky="w",padx=45)

BlrecordLabtest=StringVar()

Labtest=Entry(framebox,textvariable=BlrecordLabtest,font=("calibri",17,"bold"))

Labtest.grid(row=1,column=2,sticky="w")

labelGender=Label(framebox,text="03 :",bg="lightgreen",font=("calibri",17,"bold"))

labelGender.grid(row=2,column=0,sticky="w",padx=70)

labelGender=Label(framebox,text="Treatment Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelGender.grid(row=2,column=1,sticky="w",padx=45)

TreatmentchargesBlrecord=StringVar()

Treatmentcharges=Entry(framebox,textvariable=TreatmentchargesBlrecord,font=("calibri",17,"bold"))

Treatmentcharges.grid(row=2,column=2,sticky="w")

labelAge=Label(framebox,text="04 :",bg="lightgreen",font=("calibri",17,"bold"))

labelAge.grid(row=3,column=0,sticky="w",padx=70)

labelAge=Label(framebox,text="Procedure Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelAge.grid(row=3,column=1,sticky="w",padx=45)

BlrecordProcedurecharges=StringVar()

Procedurecharges=Entry(framebox,textvariable=BlrecordProcedurecharges,font=("calibri",17,"bold"))

Procedurecharges.grid(row=3,column=2,sticky="w")

labelPhone=Label(framebox,text="05 :",bg="lightgreen",font=("calibri",17,"bold"))

labelPhone.grid(row=4,column=0,sticky="w",padx=70)

labelPhone=Label(framebox,text="Other Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelPhone.grid(row=4,column=1,sticky="w",padx=45)

BlrecordOthercharges=StringVar()

Othercharges=Entry(framebox,font=("calibri",17,"bold"),textvariable=BlrecordOthercharges)

Othercharges.grid(row=4,column=2,sticky="w")

labelAddress=Label(framebox,text="06 :",bg="lightgreen",font=("calibri",17,"bold"))

labelAddress.grid(row=5,column=0,sticky="w",padx=70)

labelAddress=Label(framebox,text="Miscellaneous Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelAddress.grid(row=5,column=1,sticky="w",padx=45)

BlrecordMiscellaneouscharges=StringVar()

Miscellaneouscharges=Entry(framebox,textvariable=BlrecordMiscellaneouscharges,font=("calibri",17,"bold"))

Miscellaneouscharges.grid(row=5,column=2,sticky="w")

labelAddress=Label(framebox,text="07 :",bg="lightgreen",font=("calibri",17,"bold"))

labelAddress.grid(row=6,column=0,sticky="w",padx=70)

labelAddress=Label(framebox,text="Room Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelAddress.grid(row=6,column=1,sticky="w",padx=45)

BlrecordRoomcharges=StringVar()

Roomcharges=Entry(framebox,textvariable=BlrecordRoomcharges,font=("calibri",17,"bold"))

Roomcharges.grid(row=6,column=2,sticky="w")

labelDisease=Label(framebox,text="",bg="lightgreen",font=("calibri",17,"bold"))

labelDisease.grid(row=7,column=0,sticky="w",padx=105)

labelDisease=Label(framebox,text="Total:",fg="red",bg="lightgreen",font=("calibri",17,"bold"))

labelDisease.grid(row=7,column=1,sticky="w",padx=45)

BlrecordTotal=StringVar()

Total=Entry(framebox,textvariable=BlrecordTotal,font=("calibri",17,"bold"))

Total.grid(row=7,column=2,sticky="w")

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_rowconfigure(5,weight=1)

root.grid\_rowconfigure(6,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

root.grid\_columnconfigure(2,weight=1)

def Bill():

global framemenuimage

global framereceiptimage

global frameprintbill

global framehome

global BlIdentity

global BlName

global GenderBl

global BlAge

global Blcheckin

global BlDisease

global BlPhone

global BlAddress

global BlDoctorname

global Blcheckout

global BlDoctorcharges

global BlLabtest

global TreatmentchargesBl

global BlProcedurecharges

global BlOthercharges

global BlMiscellaneouscharges

global BlRoomcharges

global BlTotal

global imagemenu

global imagehome

global framereceipt

global framerireceiptimage

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framebill

global framerrecord

global framedoctorrecord

global framemenuroot

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global BlrecordIdentity

global BlrecordName

global GenderBlrecord

global BlrecordAge

global Blrecordcheckin

global BlrecordDisease

global BlrecordPhone

global BlrecordAddress

global BlrecordDoctorname

global Blrecordcheckout

global BlrecordDoctorcharges

global BlrecordLabtest

global TreatmentchargesBlrecord

global BlrecordProcedurecharges

global BlrecordOthercharges

global BlrecordMiscellaneouscharges

global BlrecordRoomcharges

global BlrecordTotal

global Blsearch

global frameappointment

global frameappointmentright

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterbill+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterbill==1:

framebill=Frame(root,relief="solid",borderwidth=2,bg="lightgreen")

framebill.pack(side="right",fill=BOTH,expand=True)

framerecordleft=Frame(framebill,height="60",width="200",relief="solid",borderwidth=2,bg="lightgreen")

framerecordleft.pack(side=LEFT,fill=Y)

framepatientbilltop=Frame(framerecordleft,height="100",relief="solid",borderwidth=2,bg="sandybrown")

framepatientbilltop.pack(side=TOP,fill=X)

labelinputpatientname=Label(framepatientbilltop,text="Input Patient's ID",bg="sandybrown",font=("calibri",16,"bold"))

labelinputpatientname.pack(anchor=CENTER,pady=6)

framepatientadmittopsearch=Frame(framerecordleft,bg="lightgreen")

framepatientadmittopsearch.pack(side=TOP)

searchID=StringVar()

inputpatientname=Entry(framepatientadmittopsearch,textvariable=searchID,width="15",font=("calibri",15,"bold"))

inputpatientname.pack(pady=10)

Searchbutton=Button(framepatientadmittopsearch,text="Search",command=lambda:Billsearch(searchID.get()),bg="yellow",font=("calibri",15,"bold"))

Searchbutton.pack(side=TOP,pady=8)

framexbill=Frame(framebill,bg="sandybrown",relief="solid",borderwidth=2)

framexbill.pack(side=TOP,fill=X)

labelbill=Label(framexbill,text="Billing",bg="sandybrown",fg="black",font=("calibri",20,"bold"))

labelbill.pack()

frametbill=Frame(framebill,height="200",bg="lightgreen")

frametbill.pack(side=TOP,fill=X)

framebleft=Frame(frametbill,height="200",width="685",bg="lightgreen")

framebleft.pack(side=LEFT)

labelPatient\_id=Label(framebleft,text="Identity:",bg="lightgreen",font=("calibri",17,"bold"))

labelPatient\_id.grid(row=0,column=0,sticky="w",padx="50",pady=2)

BlIdentity=StringVar()

Identity=Entry(framebleft,textvariable=BlIdentity,font=("calibri",16,"bold"))

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

labelName=Label(framebleft,text="Name:",bg="lightgreen",font=("calibri",17,"bold"))

labelName.grid(row=1,column=0,sticky="w",padx="50")

BlName=StringVar()

Name=Entry(framebleft,textvariable=BlName,font=("calibri",16,"bold"))

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

GenderBl=StringVar()

GenderBl.set("")

labelGender=Label(framebleft,text="Gender: ",bg="lightgreen",font=("calibri",17,"bold"))

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

i= IntVar()

r1 = Radiobutton(framebleft,text="male",value='Male',variable=GenderBl,bg="lightgreen",font=("calibri",16,"bold"))

r2 = Radiobutton(framebleft,text="female",value='Female',variable=GenderBl,bg="lightgreen",font=("calibri",16,"bold"))

r1.grid(row=2,column=1,sticky="w")

r2.grid(row=2,column=1,sticky="e",padx=0)

GenderBl.set(3)

labelAge=Label(framebleft,text="Age:",bg="lightgreen",font=("calibri",17,"bold"))

labelAge.grid(row=3,column=0,sticky="w",padx="50")

BlAge=StringVar()

Age=Entry(framebleft,textvariable=BlAge,font=("calibri",16,"bold"))

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

checkin=Label(framebleft,text="Check In:",bg="lightgreen",font=("calibri",17,"bold"))

checkin.grid(row=4,column=0,sticky="w",padx="50")

Blcheckin=StringVar()

checkin=Entry(framebleft,textvariable=Blcheckin,font=("calibri",16,"bold"))

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

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

framebright=Frame(frametbill,height="200",bg="lightgreen",width="685")

framebright.pack(side=RIGHT)

labelAge=Label(framebright,text="Disease:",bg="lightgreen",font=("calibri",17,"bold"))

labelAge.grid(row=0,column=0,sticky="w")

BlDisease=StringVar()

Disease=Entry(framebright,textvariable=BlDisease,font=("calibri",16,"bold"))

Disease.grid(row=0,column=1,padx="50")

labelPhone=Label(framebright,text="Phone:",bg="lightgreen",font=("calibri",17,"bold"))

labelPhone.grid(row=1,column=0,sticky="w")

BlPhone=StringVar()

Phone=Entry(framebright,textvariable=BlPhone,font=("calibri",16,"bold"))

Phone.grid(row=1,column=1,padx="50")

labelAddress=Label(framebright,text="Address:",bg="lightgreen",font=("calibri",17,"bold"))

labelAddress.grid(row=2,column=0,sticky="w")

BlAddress=StringVar()

Address=Entry(framebright,textvariable=BlAddress,font=("calibri",16,"bold"))

Address.grid(row=2,column=1,padx="50")

labelDisease=Label(framebright,text="Doctor Name:",bg="lightgreen",font=("calibri",17,"bold"))

labelDisease.grid(row=3,column=0,sticky="w")

BlDoctorname=StringVar()

Doctorname=Entry(framebright,textvariable=BlDoctorname,font=("calibri",16,"bold"))

Doctorname.grid(row=3,column=1,padx="50")

labelCheckOut=Label(framebright,text="Check Out:",bg="lightgreen",font=("calibri",17,"bold"))

labelCheckOut.grid(row=4,column=0,sticky="w")

Blcheckout=StringVar()

Doctorname=Entry(framebright,textvariable=Blcheckout,font=("calibri",16,"bold"))

Doctorname.grid(row=4,column=1,padx="50")

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

framebottom=Frame(framebill,relief="solid",bg="lightgreen",borderwidth=2)

framebottom.pack(side=TOP,fill=X,pady=16)

labelPatient\_id=Label(framebottom,text="Sr. No :",bg="lightgreen",fg="Purple",font=("calibri",20,"bold"))

labelPatient\_id.grid(row=0,column=0,sticky="w",padx=55)

labelPatient\_id=Label(framebottom,text="Particulars :",bg="lightgreen",fg="Purple",font=("calibri",20,"bold"))

labelPatient\_id.grid(row=0,column=1,sticky="w",padx=10)

labelPatient\_id=Label(framebottom,text="Amount :",bg="lightgreen",fg="Purple",font=("calibri",20,"bold"))

labelPatient\_id.grid(row=0,column=2,sticky="e",padx=145)

root.grid\_rowconfigure(0,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

root.grid\_columnconfigure(2,weight=1)

framebo=Frame(framebill,bg="lightgreen")

framebo.pack(side=TOP,fill=BOTH,expand=True,pady="10")

labelPatient\_id=Label(framebo,text="01 :",bg="lightgreen",font=("calibri",17,"bold"))

labelPatient\_id.grid(row=0,column=0,sticky="w",padx=75)

labelPatient\_id=Label(framebo,text="Doctor Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelPatient\_id.grid(row=0,column=1,sticky="w",padx=10)

BlDoctorcharges=StringVar()

Doctorcharges=Entry(framebo,textvariable=BlDoctorcharges,font=("calibri",17,"bold"))

Doctorcharges.grid(row=0,column=2,sticky="w",padx=40)

labelName=Label(framebo,text="02 :",bg="lightgreen",font=("calibri",17,"bold"))

labelName.grid(row=1,column=0,sticky="w",padx=75)

labelName=Label(framebo,text="Lab Test:",bg="lightgreen",font=("calibri",17,"bold"))

labelName.grid(row=1,column=1,sticky="w",padx=10)

BlLabtest=StringVar()

Labtest=Entry(framebo,textvariable=BlLabtest,font=("calibri",17,"bold"))

Labtest.grid(row=1,column=2,sticky="w",padx=40)

labelGender=Label(framebo,text="03 :",bg="lightgreen",font=("calibri",17,"bold"))

labelGender.grid(row=2,column=0,sticky="w",padx=75)

labelGender=Label(framebo,text="Treatment Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelGender.grid(row=2,column=1,sticky="w",padx=10)

TreatmentchargesBl=StringVar()

Treatmentcharges=Entry(framebo,textvariable=TreatmentchargesBl,font=("calibri",17,"bold"))

Treatmentcharges.grid(row=2,column=2,sticky="w",padx=40)

labelAge=Label(framebo,text="04 :",bg="lightgreen",font=("calibri",17,"bold"))

labelAge.grid(row=3,column=0,sticky="w",padx=75)

labelAge=Label(framebo,text="Procedure Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelAge.grid(row=3,column=1,sticky="w",padx=10)

BlProcedurecharges=StringVar()

Procedurecharges=Entry(framebo,textvariable=BlProcedurecharges,font=("calibri",17,"bold"))

Procedurecharges.grid(row=3,column=2,sticky="w",padx=40)

labelPhone=Label(framebo,text="05 :",bg="lightgreen",font=("calibri",17,"bold"))

labelPhone.grid(row=4,column=0,sticky="w",padx=75)

labelPhone=Label(framebo,text="Other Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelPhone.grid(row=4,column=1,sticky="w",padx=10)

BlOthercharges=StringVar()

Othercharges=Entry(framebo,font=("calibri",17,"bold"),textvariable=BlOthercharges)

Othercharges.grid(row=4,column=2,sticky="w",padx=40)

labelAddress=Label(framebo,text="06 :",bg="lightgreen",font=("calibri",17,"bold"))

labelAddress.grid(row=5,column=0,sticky="w",padx=75)

labelAddress=Label(framebo,text="Miscellaneous Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelAddress.grid(row=5,column=1,sticky="w",padx=10)

BlMiscellaneouscharges=StringVar()

Miscellaneouscharges=Entry(framebo,textvariable=BlMiscellaneouscharges,font=("calibri",17,"bold"))

Miscellaneouscharges.grid(row=5,column=2,sticky="w",padx=40)

labelAddress=Label(framebo,text="07 :",bg="lightgreen",font=("calibri",17,"bold"))

labelAddress.grid(row=6,column=0,sticky="w",padx=75)

labelAddress=Label(framebo,text="Room Charges:",bg="lightgreen",font=("calibri",17,"bold"))

labelAddress.grid(row=6,column=1,sticky="w",padx=10)

BlRoomcharges=StringVar()

Roomcharges=Entry(framebo,textvariable=BlRoomcharges,font=("calibri",17,"bold"))

Roomcharges.grid(row=6,column=2,sticky="w",padx=40)

labelDisease=Label(framebo,text="",bg="lightgreen",font=("calibri",17,"bold"))

labelDisease.grid(row=7,column=0,sticky="w",padx=75)

labelDisease=Button(framebo,text="Total:",command=total,bg="orange",font=("calibri",17,"bold"))

labelDisease.grid(row=7,column=1,sticky="w",padx=10)

BlTotal=StringVar()

Total=Entry(framebo,textvariable=BlTotal,font=("calibri",17,"bold"))

Total.grid(row=7,column=2,sticky="w",padx=40)

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_rowconfigure(5,weight=1)

root.grid\_rowconfigure(6,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

root.grid\_columnconfigure(2,weight=1)

frameprint=Frame(framebill,bg="lightgreen")

frameprint.pack(side=RIGHT)

buttonadd=Button(frameprint,text="SAVE",bg="yellow",command=savebill,bd=3,width=10,font=("calibri",15,"bold"))

buttonadd.pack(side=LEFT)

buttonPrint=Button(frameprint,text="Print",bg="yellow",command=printbill,bd=3,width=10,font=("calibri",15,"bold"))

buttonPrint.pack(side=RIGHT,padx="50")

def staffregistrationfetch\_datas():

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from sr"

cursor.execute(query)

rows=cursor.fetchall()

if len(rows)!=0:

Staffregistration\_tables.delete(\*Staffregistration\_tables.get\_children())

for row in rows:

Staffregistration\_tables.insert('',END,values=row)

connection.commit()

connection.close()

def drecord():

global framemenuimage

global framehome

global frameprintbill

global framereceiptimage

global imagemenu

global imagehome

global framereceipt

global framerireceiptimage

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framebill

global framerrecord

global framedoctorrecord

global framemenuroot

global Staffregistration\_tables

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global BlrecordIdentity

global BlrecordName

global GenderBlrecord

global BlrecordAge

global Blrecordcheckin

global BlrecordDisease

global BlrecordPhone

global BlrecordAddress

global BlrecordDoctorname

global Blrecordcheckout

global BlrecordDoctorcharges

global BlrecordLabtest

global TreatmentchargesBlrecord

global BlrecordProcedurecharges

global BlrecordOthercharges

global BlrecordMiscellaneouscharges

global BlrecordRoomcharges

global BlrecordTotal

global Blsearch

global frameappointment

global frameappointmentright

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

counterdrecord+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterbill=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

if counterdrecord==1:

framedoctorrecord=Frame(root,relief="solid",borderwidth=2)

framedoctorrecord.pack(side="right",fill=BOTH,expand=True)

scroll1=Scrollbar(framedoctorrecord,orient=HORIZONTAL)

scroll1.pack(side=BOTTOM,fill=X)

scroll2=Scrollbar(framedoctorrecord,orient=VERTICAL)

scroll2.pack(side=RIGHT,fill=Y)

Staffregistration\_tables=ttk.Treeview(framedoctorrecord,columns=("SrID", "SrName", "GenderSr", "SrPosition","SrDepartment","SrMailid","SrSalary","SrPhone","SrAddress"),xscrollcommand=scroll1.set,yscrollcommand=scroll2.set)

scroll1.config(command=Staffregistration\_tables.xview)

scroll2.config(command=Staffregistration\_tables.yview)

Staffregistration\_tables.heading('SrID', text=" ID", anchor=W)

Staffregistration\_tables.heading('SrName', text="Name", anchor=W)

Staffregistration\_tables.heading('GenderSr', text="Gender", anchor=W)

Staffregistration\_tables.heading('SrPosition', text="Position", anchor=W)

Staffregistration\_tables.heading('SrDepartment', text="Department", anchor=W)

Staffregistration\_tables.heading('SrMailid', text="Mailid", anchor=W)

Staffregistration\_tables.heading('SrSalary', text="Salary", anchor=W)

Staffregistration\_tables.heading('SrPhone', text="Phone", anchor=W)

Staffregistration\_tables.heading('SrAddress', text="Address", anchor=W)

Staffregistration\_tables['show']='headings'

Staffregistration\_tables.column("SrID",width=100)

Staffregistration\_tables.column("SrName",width=100)

Staffregistration\_tables.column("GenderSr",width=100)

Staffregistration\_tables.column("SrPosition",width=100)

Staffregistration\_tables.column("SrDepartment",width=100)

Staffregistration\_tables.column("SrMailid",width=100)

Staffregistration\_tables.column("SrSalary",width=100)

Staffregistration\_tables.column("SrPhone",width=100)

Staffregistration\_tables.column("SrAddress",width=100)

Staffregistration\_tables.pack(fill=BOTH,expand=True)

staffregistrationfetch\_datas()

def search\_dataregistration():

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

cursor.execute("select \* from pr where "+str(search\_byregistration.get())+" Like '%"+str(search\_txtregistration.get())+"%'")

rows=cursor.fetchall()

if len(rows)!=0:

Patientregistration\_table.delete(\*Patientregistration\_table.get\_children())

for row in rows:

Patientregistration\_table.insert('',END,values=row)

connection.commit()

connection.close()

def fetchappointment\_dataregistration():

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pr"

cursor.execute(query)

rows=cursor.fetchall()

if len(rows)!=0:

Patientregistration\_table.delete(\*Patientregistration\_table.get\_children())

for row in rows:

Patientregistration\_table.insert('',END,values=row)

connection.commit()

connection.close()

def patientregistrationfetch\_data():

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pr"

cursor.execute(query)

rows=cursor.fetchall()

if len(rows)!=0:

Patientregistration\_table.delete(\*Patientregistration\_table.get\_children())

for row in rows:

Patientregistration\_table.insert('',END,values=row)

connection.commit()

connection.close()

def RRecord():

global framemenuimage

global framehome

global frameprintbill

global framereceiptimage

global imagehome

global imagemenu

global framereceipt

global framerireceiptimage

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framedoctorrecord

global framebill

global framerrecord

global framemenuroot

global search\_byregistration

global search\_txtregistration

global PrPatientid

global PrName

global GenderPr

global PrAge

global PrPhone

global PrAddress

global PrDisease

global PrCheckIn

global PrBloodgroup

global PrDoctorname

global PrRoomNo

global RiPatientid

global RiName

global GenderRi

global RiAge

global RiPhone

global RiAddress

global RiDisease

global RiCheckIn

global RiBloodgroup

global RiDoctorname

global RiRoomNo

global Patientregistration\_table

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global BlrecordIdentity

global BlrecordName

global GenderBlrecord

global BlrecordAge

global Blrecordcheckin

global BlrecordDisease

global BlrecordPhone

global BlrecordAddress

global BlrecordDoctorname

global Blrecordcheckout

global BlrecordDoctorcharges

global BlrecordLabtest

global TreatmentchargesBlrecord

global BlrecordProcedurecharges

global BlrecordOthercharges

global BlrecordMiscellaneouscharges

global BlrecordRoomcharges

global BlrecordTotal

global Blsearch

global frameappointment

global frameappointmentright

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

counterrrecord+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterdrecord=0

counteradmitrecord=0

counterbill=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

if counterrrecord==1:

framerrecord=Frame(root,relief="solid",borderwidth=2)

framerrecord.pack(fill=BOTH,expand=True)

frame1=Frame(framerrecord)

frame1.pack(side=TOP)

framerrecordupper=Frame(frame1)

framerrecordupper.grid()

lbl\_search=Label(framerrecordupper,text="Search By",fg="black",font=("times new roman",13,"bold"))

lbl\_search.grid(row=0,column=0,pady=7,sticky="w")

search\_byregistration=StringVar()

combo\_search=ttk.Combobox(framerrecordupper,textvariable=search\_byregistration,width=7,font=("times new roman",13,"bold"),state='readonly')

combo\_search['values']=("PrPhone","PrName","PrPatientid","GenderPr","PrDisease","PrBloodgroup")

combo\_search.grid(row=0,column=1,pady=10)

search\_txtregistration=StringVar()

txt\_search=Entry(framerrecordupper,textvariable=search\_txtregistration,font=("times new roman",10,"bold"),bd=5,relief=GROOVE,width=10)

txt\_search.grid(row=0,column=2,padx=5,pady=10,sticky="w")

searchbtn=Button(framerrecordupper,text="Search",command=search\_dataregistration,width=10).grid(row=0,column=3,pady=10)

showallbtn=Button(framerrecordupper,text="Show All",command=fetchappointment\_dataregistration,width=10).grid(row=0,column=4,pady=10)

framerrecorddown=Frame(framerrecord)

framerrecorddown.pack(fill=BOTH,expand=True)

scroll1=Scrollbar(framerrecorddown,orient=HORIZONTAL)

scroll1.pack(side=BOTTOM,fill=X)

scroll2=Scrollbar(framerrecorddown,orient=VERTICAL)

scroll2.pack(side=RIGHT,fill=Y)

Patientregistration\_table=ttk.Treeview(framerrecorddown,columns=("PrPatientid", "PrName", "GenderPr", "PrAge","PrPhone","PrAddress",

"PrDisease","PrCheckIn","PrBloodgroup","PrDoctorname","PrRoomNo"),xscrollcommand=scroll1.set,yscrollcommand=scroll2.set)

scroll1.config(command=Patientregistration\_table.xview)

scroll2.config(command=Patientregistration\_table.yview)

Patientregistration\_table.heading('PrPatientid', text="Patientid", anchor=W)

Patientregistration\_table.heading('PrName', text="Name", anchor=W)

Patientregistration\_table.heading('GenderPr', text="Gender", anchor=W)

Patientregistration\_table.heading('PrAge', text="Age", anchor=W)

Patientregistration\_table.heading('PrPhone', text="Phone", anchor=W)

Patientregistration\_table.heading('PrAddress', text="Address", anchor=W)

Patientregistration\_table.heading('PrDisease', text="Disease", anchor=W)

Patientregistration\_table.heading('PrCheckIn', text="CheckIn", anchor=W)

Patientregistration\_table.heading('PrBloodgroup', text="Bloodgroup", anchor=W)

Patientregistration\_table.heading('PrDoctorname', text="Doctorname", anchor=W)

Patientregistration\_table.heading('PrRoomNo', text="RoomNo", anchor=W)

Patientregistration\_table['show']='headings'

Patientregistration\_table.column("PrPatientid",width=100)

Patientregistration\_table.column("PrName",width=100)

Patientregistration\_table.column("GenderPr",width=85)

Patientregistration\_table.column("PrAge",width=70)

Patientregistration\_table.column("PrPhone",width=100)

Patientregistration\_table.column("PrAddress",width=140)

Patientregistration\_table.column("PrDisease",width=100)

Patientregistration\_table.column("PrCheckIn",width=100)

Patientregistration\_table.column("PrBloodgroup",width=100)

Patientregistration\_table.column("PrDoctorname",width=100)

Patientregistration\_table.column("PrRoomNo",width=100)

Patientregistration\_table.pack(fill=BOTH,expand=True)

patientregistrationfetch\_data()

def Siupdate():

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="update sr set SrName = '{}',GenderSr = '{}',SrPosition = '{}',SrDepartment = '{}',SrMailid = '{}',SrSalary = '{}',SrPhone = '{}',SrAddress = '{}' where SrID = '{}'".format(SiName.get(),GenderSi.get(),SiPosition.get(),SiDepartment.get(),SiMailid.get(),SiSalary.get(),SiPhone.get(),SiAddress.get(),SiID.get())

cursor.execute(query)

connection.commit()

messagebox.showinfo("Congratulations ", "Data Updated Successfully")

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from sr"

cursor.execute(query)

for row in cursor:

SiID.set(row[0])

SiName.set(row[1])

GenderSi.set(row[2])

SiPosition.set(row[3])

SiDepartment.set(row[4])

SiMailid.set(row[5])

SiSalary.set(row[6])

SiPhone.set(row[7])

SiAddress.set(row[8])

def Sisearch(ID):

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from Sr where SrID like '{0}' or SrPhone like '{0}' or SrAddress like '{0}'".format(ID)

cursor.execute(query)

for row in cursor:

SiID.set(row[0])

SiName.set(row[1])

GenderSi.set(row[2])

SiPosition.set(row[3])

SiDepartment.set(row[4])

SiMailid.set(row[5])

SiSalary.set(row[6])

SiPhone.set(row[7])

SiAddress.set(row[8])

def Sradd():

global framemenuimage

global framepatientregistration

global imagehome

global frameprintbill

global framereceiptimage

global imagemenu

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framebill

global framedoctorrecord

global framereceipt

global framerireceiptimage

global SrID

global SrName

global GenderSr

global SrPosition

global SrDepartment

global SrMailid

global SrSalary

global SrPhone

global SrAddress

global SiID

global SiName

global GenderSi

global SiPosition

global SiDepartment

global SiMailid

global SiSalary

global SiPhone

global SiAddress

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global frameappointment

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

if SrName.get() == "" or GenderSr.get() == "" or SrPosition.get() == "" or SrDepartment.get() == "" or SrPhone.get() == "":

result = tkMessageBox.showwarning('', 'Please Complete The Required Field', icon="warning")

else:

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

if "@" and ".com" not in SrMailid.get() :

messagebox.showinfo(" INVALID DETAILS","ENTER VALID EMAIL ADDRESS")

else:

x=random.randint(0, 1000)

randomRef = str(x)

randomRef = 'Staff2020-' + randomRef

SrID.set(randomRef)

query = "insert into sr values('{}','{}','{}','{}','{}','{}','{}','{}','{}')".format(SrID.get(),SrName.get(),GenderSr.get(),SrPosition.get(),SrDepartment.get(),SrMailid.get(),int(SrSalary.get()),int(SrPhone.get()),SrAddress.get())

cursor.execute(query)

connection.commit()

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from sr"

cursor.execute(query)

messagebox.showinfo("Congratulations ", "Data Added Successfully ")

def sinfo():

global framemenuimage

global framehome

global frameprintbill

global framereceiptimage

global imagemenu

global imagehome

global framereceipt

global framerireceiptimage

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framerrecord

global framedoctorrecord

global framemenuroot

global SrID

global SrName

global GenderSr

global SrPosition

global SrDepartment

global SrMailid

global SrSalary

global SrPhone

global SrAddress

global SiID

global SiName

global GenderSi

global SiPosition

global SiDepartment

global SiMailid

global SiSalary

global SiPhone

global SiAddress

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global frameappointment

global frameappointmentright

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

countersinfo+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterpadmit=0

counterainfo=0

countersreg=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if countersinfo==1:

framestaffinformation=Frame(root,bg="lightgreen",)

framestaffinformation.pack(side="right",fill=BOTH,expand=True)

framestaffinformationtop=Frame(framestaffinformation,height="200",relief="solid",bg="sandybrown",borderwidth=2)

framestaffinformationtop.pack(side=TOP,fill=X)

labelstaffinformation=Label(framestaffinformationtop,text="Staff Information",bg="sandybrown",font=("calibri",23,"bold"))

labelstaffinformation.pack(anchor="center")

framestaffinformationleft=Frame(framestaffinformation,height="60",width="300",bg="lightgreen",relief="solid",borderwidth=2)

framestaffinformationleft.pack(side=LEFT,fill=Y)

labelUsername=Label(framestaffinformationleft,text="User ID",bg="lightgreen",font=("calibri",20,"bold"))

labelUsername.pack(anchor=CENTER,pady=20)

searchID=StringVar()

Username=Entry(framestaffinformationleft,width="20",textvariable=searchID,font=("calibri",15,"bold"))

Username.pack()

Searchbutton=Button(framestaffinformationleft,text="Search",command=lambda:Sisearch(searchID.get()),bg="yellow",font=("calibri",15,"bold"))

Searchbutton.pack(side=TOP,pady=8)

framestaffinformationright=Frame(framestaffinformation,height="60",bg="lightgreen",width="300",relief="solid",borderwidth=2)

framestaffinformationright.pack(side=RIGHT,fill=Y)

buttonUpdate=Button(framestaffinformationright,text="Update",command=Siupdate,bd=7,bg="yellow",width=10,font=("calibri",15,"bold"))

buttonUpdate.pack(side=TOP,padx="70",pady=100)

framestaffinformationcenter=Frame(framestaffinformation,height="60",width="130",bg="lightgreen",relief="solid",borderwidth=2)

framestaffinformationcenter.pack(anchor="center",fill=BOTH,expand=True)

labelid=Label(framestaffinformationcenter,text="ID: ",bg="lightgreen",font=("calibri",20,"bold"))

labelid.grid(row=0,column=0,padx=10,pady=15)

SiID=StringVar()

ID=Entry(framestaffinformationcenter,textvariable=SiID,font=("calibri",20,"bold"))

ID.grid(row=0,column=1,padx=0)

labelName=Label(framestaffinformationcenter,text="Name: ",bg="lightgreen",font=("calibri",20,"bold"))

labelName.grid(row=1,column=0,padx=10,pady=15)

SiName=StringVar()

Name=Entry(framestaffinformationcenter,textvariable=SiName,font=("calibri",20,"bold"))

Name.grid(row=1,column=1,padx=0)

GenderSi=StringVar()

GenderSi.set("")

labelGender=Label(framestaffinformationcenter,text="Gender: ",bg="lightgreen",font=("calibri",20,"bold"))

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

r1 = Radiobutton(framestaffinformationcenter,text="male",value='Male',bg="lightgreen",variable=GenderSi,font=("calibri",20,"bold"))

r1.grid(row=2,column=1,sticky="w")

r2 = Radiobutton(framestaffinformationcenter,text="female",value='Female',bg="lightgreen",variable=GenderSi,font=("calibri",20,"bold"))

r2.grid(row=2,column=1,sticky="e",padx=0)

GenderSi.set(3)

labelPosition=Label(framestaffinformationcenter,text="Position: ",bg="lightgreen",font=("calibri",20,"bold"))

labelPosition.grid(row=3,column=0,padx=10,pady=15)

SiPosition=StringVar()

Position=Entry(framestaffinformationcenter,textvariable=SiPosition,font=("calibri",20,"bold"))

Position.grid(row=3,column=1,padx=0)

labelDepartment=Label(framestaffinformationcenter,text="Department: ",bg="lightgreen",font=("calibri",20,"bold"))

labelDepartment.grid(row=4,column=0,padx=10,pady=15)

SiDepartment=StringVar()

Department=Entry(framestaffinformationcenter,textvariable=SiDepartment,font=("calibri",20,"bold"))

Department.grid(row=4,column=1,padx=0)

labelMailid=Label(framestaffinformationcenter,text="Mail ID: ",bg="lightgreen",font=("calibri",20,"bold"))

labelMailid.grid(row=5,column=0,padx=10,pady=15)

SiMailid=StringVar()

Mailid=Entry(framestaffinformationcenter,textvariable=SiMailid,font=("calibri",20,"bold"))

Mailid.grid(row=5,column=1,padx=0)

labelSalary=Label(framestaffinformationcenter,text="Salary: ",bg="lightgreen",font=("calibri",20,"bold"))

labelSalary.grid(row=6,column=0,padx=10,pady=15)

SiSalary=StringVar()

Salary=Entry(framestaffinformationcenter,textvariable=SiSalary,font=("calibri",20,"bold"))

Salary.grid(row=6,column=1,padx=0)

labelPhone=Label(framestaffinformationcenter,text="Phone: ",bg="lightgreen",font=("calibri",20,"bold"))

labelPhone.grid(row=7,column=0,padx=10,pady=15)

SiPhone=StringVar()

Phone=Entry(framestaffinformationcenter,textvariable=SiPhone,font=("calibri",20,"bold"))

Phone.grid(row=7,column=1,padx=0)

labelAddress=Label(framestaffinformationcenter,text="Address: ",bg="lightgreen",font=("calibri",20,"bold"))

labelAddress.grid(row=8,column=0,padx=10,pady=15)

SiAddress=StringVar()

Address=Entry(framestaffinformationcenter,textvariable=SiAddress,font=("calibri",20,"bold"))

Address.grid(row=8,column=1,padx=0)

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_rowconfigure(5,weight=1)

root.grid\_rowconfigure(6,weight=1)

root.grid\_rowconfigure(7,weight=1)

root.grid\_rowconfigure(8,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

def sreg():

global framemenuimage

global framehome

global frameprintbill

global framereceiptimage

global imagemenu

global imagehome

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framerrecord

global framebill

global framedoctorrecord

global framereceipt

global framerireceiptimage

global framemenuroot

global SrID

global SrName

global GenderSr

global SrPosition

global SrDepartment

global SrMailid

global SrSalary

global SrPhone

global SrAddress

global SiID

global SiName

global GenderSi

global SiPosition

global SiDepartment

global SiMailid

global SiSalary

global SiPhone

global SiAddress

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global frameappointment

global frameappointmentright

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

countersreg+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterpadmit=0

counterainfo=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if countersreg==1:

framestaffregistration=Frame(root,bg="lightgreen")

framestaffregistration.pack(side="right",fill=BOTH,expand=True)

framestaffregistrationtop=Frame(framestaffregistration,height="200",bg="sandybrown",relief="solid",borderwidth=2,)

framestaffregistrationtop.pack(side=TOP,fill=X)

labelstaffregistration=Label(framestaffregistrationtop,text="Staff Registration",bg="sandybrown",font=("calibri",23,"bold"))

labelstaffregistration.pack(anchor="center")

framestaffregistrationcenter=Frame(framestaffregistration,height="60",width="130",relief="solid",bg="lightgreen",borderwidth=2)

framestaffregistrationcenter.pack(side=LEFT,fill=BOTH,expand=True)

labelid=Label(framestaffregistrationcenter,text="ID: ",bg="lightgreen",font=("calibri",21,"bold"))

labelid.grid(row=0,column=0,padx=120,pady=15)

SrID=StringVar()

ID=Entry(framestaffregistrationcenter,textvariable=SrID,font=("calibri",20,"bold"))

ID.grid(row=0,column=1,padx=0)

labelName=Label(framestaffregistrationcenter,text="Name: ",bg="lightgreen",font=("calibri",21,"bold"))

labelName.grid(row=1,column=0,padx=10,pady=15)

SrName=StringVar()

Name=Entry(framestaffregistrationcenter,textvariable=SrName,font=("calibri",20,"bold"))

Name.grid(row=1,column=1,padx=0)

GenderSr=StringVar()

GenderSr.set("")

labelGender=Label(framestaffregistrationcenter,text="Gender: ",bg="lightgreen",font=("calibri",21,"bold"))

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

i= IntVar()

r1 = Radiobutton(framestaffregistrationcenter,text="male",value="Male",bg="lightgreen",variable=GenderSr,font=("calibri",20,"bold"))

r2 = Radiobutton(framestaffregistrationcenter,text="female",value="Female",variable=GenderSr,bg="lightgreen",font=("calibri",20,"bold"))

r1.grid(row=2,column=1,sticky="w")

r2.grid(row=2,column=1,sticky="e",padx=0)

GenderSr.set(3)

labelPosition=Label(framestaffregistrationcenter,text="Position: ",bg="lightgreen",font=("calibri",21,"bold"))

labelPosition.grid(row=3,column=0,padx=10,pady=15)

SrPosition=StringVar()

Position=Entry(framestaffregistrationcenter,textvariable=SrPosition,font=("calibri",20,"bold"))

Position.grid(row=3,column=1,padx=0)

labelDepartment=Label(framestaffregistrationcenter,text="Department: ",bg="lightgreen",font=("calibri",21,"bold"))

labelDepartment.grid(row=4,column=0,padx=10,pady=15)

SrDepartment=StringVar()

Department=Entry(framestaffregistrationcenter,textvariable=SrDepartment,font=("calibri",20,"bold"))

Department.grid(row=4,column=1,padx=0)

labelMailid=Label(framestaffregistrationcenter,text="Mail ID: ",bg="lightgreen",font=("calibri",21,"bold"))

labelMailid.grid(row=5,column=0,padx=10,pady=15)

SrMailid=StringVar()

Mailid=Entry(framestaffregistrationcenter,textvariable=SrMailid,font=("calibri",20,"bold"))

Mailid.grid(row=5,column=1,padx=0)

labelSalary=Label(framestaffregistrationcenter,text="Salary: ",bg="lightgreen",font=("calibri",21,"bold"))

labelSalary.grid(row=6,column=0,padx=10,pady=15)

SrSalary=StringVar()

Salary=Entry(framestaffregistrationcenter,textvariable=SrSalary,font=("calibri",20,"bold"))

Salary.grid(row=6,column=1,padx=0)

labelPhone=Label(framestaffregistrationcenter,text="Phone: ",bg="lightgreen",font=("calibri",21,"bold"))

labelPhone.grid(row=7,column=0,padx=10,pady=15)

SrPhone=StringVar()

Phone=Entry(framestaffregistrationcenter,textvariable=SrPhone,font=("calibri",20,"bold"))

Phone.grid(row=7,column=1,padx=0)

labelAddress=Label(framestaffregistrationcenter,text="Address: ",bg="lightgreen",font=("calibri",21,"bold"))

labelAddress.grid(row=8,column=0,padx=10,pady=15)

SrAddress=StringVar()

Address=Entry(framestaffregistrationcenter,textvariable=SrAddress,font=("calibri",20,"bold"))

Address.grid(row=8,column=1,padx=0)

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_rowconfigure(5,weight=1)

root.grid\_rowconfigure(6,weight=1)

root.grid\_rowconfigure(7,weight=1)

root.grid\_rowconfigure(8,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

framestaffregistrationright=Frame(framestaffregistration,height="60",width="300",bg="lightgreen",relief="solid",borderwidth=2)

framestaffregistrationright.pack(side=RIGHT,fill=Y)

buttonAdd=Button(framestaffregistrationright,text="ADD",command=Sradd,bd=7,bg="yellow",width=10,font=("calibri",15,"bold"))

buttonAdd.pack(side=TOP,padx="70",pady=100)

def Aisearch(ID):

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pa where PaPatientid like '{0}' or PaName like '{0}' or PaPhone like '{0}'".format(ID)

cursor.execute(query)

for row in cursor:

AiPatientid.set(row[0])

AiName.set(row[1])

GenderAi.set(row[2])

AiAge.set(row[3])

AiPhone.set(row[4])

AiAddress.set(row[5])

AiDisease.set(row[6])

AiCheckIn.set(row[7])

AiBloodgroup.set(row[8])

AiDoctorname.set(row[9])

AiCheckOut.set(row[10])

AiRoomNo.set(row[11])

AiRoomtype.set(row[12])

AiPrice.set(row[13])

def Aiupdate():

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="update pa set PaName = '{}',GenderPa = '{}',PaAge = '{}',PaPhone = '{}',PaAddress = '{}',PaDisease = '{}',PaCheckIn = '{}',PaBloodgroup = '{}',PaDoctorname = '{}',PaCheckOut = '{}',PaRoomNo = '{}',PaRoomtype = '{}',PaPrice = '{}' where PaPatientid = '{}'".format(AiName.get(),GenderAi.get(),AiAge.get(),AiPhone.get(),AiAddress.get(),AiDisease.get(),AiCheckIn.get(),AiBloodgroup.get(),AiDoctorname.get(),AiCheckOut.get(),AiRoomNo.get(),AiRoomtype.get(),AiPrice.get(),AiPatientid.get())

cursor.execute(query)

connection.commit()

messagebox.showinfo("Congratulations ", "Data Updated Successfully")

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pa"

cursor.execute(query)

for row in cursor:

AiPatientid.set(row[0])

AiName.set(row[1])

GenderAi.set(row[2])

AiAge.set(row[3])

AiPhone.set(row[4])

AiAddress.set(row[5])

AiDisease.set(row[6])

AiCheckIn.set(row[7])

AiBloodgroup.set(row[8])

AiDoctorname.set(row[9])

AiCheckOut.set(row[10])

AiRoomNo.set(row[11])

AiRoomtype.set(row[12])

AiPrice.set(row[13])

PaPatientid.set("")

PaName.set("")

GenderPa.set("")

PaAge.set("")

PaPhone.set("")

PaAddress.set("")

PaDisease.set("")

PaCheckIn.set("")

PaBloodgroup.set("")

PaDoctorname.set("")

PaCheckOut.set("")

PaRoomNo.set("")

PaRoomtype.set("")

PaPrice.set("")

def patientadmitfetch\_data():

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pa"

cursor.execute(query)

rows=cursor.fetchall()

if len(rows)!=0:

Patientadmit\_table.delete(\*Patientadmit\_table.get\_children())

for row in rows:

Patientadmit\_table.insert('',END,values=row)

connection.commit()

connection.close()

def search\_dataadmitrecord():

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

cursor.execute("select \* from pa where "+str(search\_byadmitrecord.get())+" Like '%"+str(search\_txtadmitrecord.get())+"%'")

rows=cursor.fetchall()

if len(rows)!=0:

Patientadmit\_table.delete(\*Patientadmit\_table.get\_children())

for row in rows:

Patientadmit\_table.insert('',END,values=row)

connection.commit()

connection.close()

def fetch\_dataadmitrecord():

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pa"

cursor.execute(query)

rows=cursor.fetchall()

if len(rows)!=0:

Patientadmit\_table.delete(\*Patientadmit\_table.get\_children())

for row in rows:

Patientadmit\_table.insert('',END,values=row)

connection.commit()

connection.close()

def AdmitRecord():

global framemenuimage

global framehome

global frameprintbill

global framereceiptimage

global imagehome

global imagemenu

global framereceipt

global framerireceiptimage

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framedoctorrecord

global framebill

global framerrecord

global framemenuroot

global search\_byregistration

global search\_byadmitrecord

global search\_txtadmitrecord

global search\_txtregistration

global PrPatientid

global PrName

global GenderPr

global PrAge

global PrPhone

global PrAddress

global PrDisease

global PrCheckIn

global PrBloodgroup

global PrDoctorname

global PrRoomNo

global RiPatientid

global RiName

global GenderRi

global RiAge

global RiPhone

global RiAddress

global RiDisease

global RiCheckIn

global RiBloodgroup

global RiDoctorname

global RiRoomNo

global Patientregistration\_table

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global BlrecordIdentity

global BlrecordName

global GenderBlrecord

global BlrecordAge

global Blrecordcheckin

global BlrecordDisease

global BlrecordPhone

global BlrecordAddress

global BlrecordDoctorname

global Blrecordcheckout

global BlrecordDoctorcharges

global BlrecordLabtest

global TreatmentchargesBlrecord

global BlrecordProcedurecharges

global BlrecordOthercharges

global BlrecordMiscellaneouscharges

global BlrecordRoomcharges

global BlrecordTotal

global Blsearch

global frameappointment

global frameappointmentright

global Patientadmit\_table

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

counteradmitrecord+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

if counteradmitrecord==1:

frameadmitrecord=Frame(root,relief="solid",borderwidth=2)

frameadmitrecord.pack(fill=BOTH,expand=True)

frameadmitrecordtop=Frame(frameadmitrecord)

frameadmitrecordtop.pack(side=TOP)

frameadmitrecordupper=Frame(frameadmitrecordtop)

frameadmitrecordupper.grid()

lbl\_search=Label(frameadmitrecordupper,text="Search By",fg="black",font=("times new roman",13,"bold"))

lbl\_search.grid(row=0,column=0,pady=7,sticky="w")

search\_byadmitrecord=StringVar()

combo\_search=ttk.Combobox(frameadmitrecordupper,textvariable=search\_byadmitrecord,width=7,font=("times new roman",13,"bold"),state='readonly')

combo\_search['values']=("PaPatientid","PaName","GenderPa","PaDisease","PaPhone","PaBloodgroup")

combo\_search.grid(row=0,column=1,pady=10)

search\_txtadmitrecord=StringVar()

txt\_search=Entry(frameadmitrecordupper,textvariable=search\_txtadmitrecord,font=("times new roman",10,"bold"),bd=5,relief=GROOVE,width=10)

txt\_search.grid(row=0,column=2,padx=5,pady=10,sticky="w")

searchbtn=Button(frameadmitrecordupper,text="Search",command=search\_dataadmitrecord,width=10).grid(row=0,column=3,pady=10)

showallbtn=Button(frameadmitrecordupper,text="Show All",command=fetch\_dataadmitrecord,width=10).grid(row=0,column=4,pady=10)

frameadmitrecorddown=Frame(frameadmitrecord)

frameadmitrecorddown.pack(fill=BOTH,expand=True)

scroll1=Scrollbar(frameadmitrecorddown,orient=HORIZONTAL)

scroll1.pack(side=BOTTOM,fill=X)

scroll2=Scrollbar(frameadmitrecorddown,orient=VERTICAL)

scroll2.pack(side=RIGHT,fill=Y)

Patientadmit\_table=ttk.Treeview(frameadmitrecorddown,columns=("PaPatientid", "PaName", "GenderPa", "PaAge","PaPhone","PaAddress","PaDisease","PaCheckIn","PaBloodgroup","PaDoctorname","PaCheckOut","PaRoomNo","PaRoomtype"),xscrollcommand=scroll1.set,yscrollcommand=scroll2.set)

scroll1.config(command=Patientadmit\_table.xview)

scroll2.config(command=Patientadmit\_table.yview)

Patientadmit\_table.heading('PaPatientid', text="Patientid", anchor=W)

Patientadmit\_table.heading('PaName', text="Name", anchor=W)

Patientadmit\_table.heading('GenderPa', text="Gender", anchor=W)

Patientadmit\_table.heading('PaAge', text="Age", anchor=W)

Patientadmit\_table.heading('PaPhone', text="Phone", anchor=W)

Patientadmit\_table.heading('PaAddress', text="Address", anchor=W)

Patientadmit\_table.heading('PaDisease', text="Disease", anchor=W)

Patientadmit\_table.heading('PaCheckIn', text="CheckIn", anchor=W)

Patientadmit\_table.heading('PaBloodgroup', text="Bloodgroup", anchor=W)

Patientadmit\_table.heading('PaDoctorname', text="Doctorname", anchor=W)

Patientadmit\_table.heading('PaCheckOut', text="Checkout", anchor=W)

Patientadmit\_table.heading('PaRoomNo', text="Room No", anchor=W)

Patientadmit\_table.heading('PaRoomtype', text="Room Type", anchor=W)

Patientadmit\_table['show']='headings'

Patientadmit\_table.column("PaPatientid",width=100)

Patientadmit\_table.column("PaName",width=100)

Patientadmit\_table.column("GenderPa",width=85)

Patientadmit\_table.column("PaAge",width=70)

Patientadmit\_table.column("PaPhone",width=100)

Patientadmit\_table.column("PaAddress",width=140)

Patientadmit\_table.column("PaDisease",width=100)

Patientadmit\_table.column("PaCheckIn",width=100)

Patientadmit\_table.column("PaBloodgroup",width=100)

Patientadmit\_table.column("PaDoctorname",width=100)

Patientadmit\_table.column("PaCheckOut",width=100)

Patientadmit\_table.column("PaRoomNo",width=100)

Patientadmit\_table.column("PaRoomtype",width=100)

Patientadmit\_table.pack(fill=BOTH,expand=True)

patientadmitfetch\_data()

def ainfo():

global framemenuimage

global framehome

global frameprintbill

global imagemenu

global framereceiptimage

global imagehome

global framemenuroot

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framerrecord

global framebill

global framedoctorrecord

global framereceipt

global framerireceiptimage

global AiPatientid

global AiName

global GenderAi

global AiAge

global AiPhone

global AiAddress

global AiDisease

global AiCheckIn

global AiBloodgroup

global AiDoctorname

global AiCheckOut

global AiRoomNo

global AiRoomtype

global AiPrice

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global frameappointment

global frameappointmentright

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterainfo+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterpadmit=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterainfo==1:

frameainfo=Frame(root,width=100,height=100,bg="lightgreen")

frameainfo.pack(side="right",fill=BOTH,expand=TRUE)

frameainfotop=Frame(frameainfo,height="200",relief="solid",borderwidth=2,bg="sandybrown")

frameainfotop.pack(side=TOP,fill=X)

labelpatientinfotext=Label(frameainfotop,text="Admit Information",bg="sandybrown",font=("calibri",23,"bold"))

labelpatientinfotext.pack(anchor="center")

frameainfoleft=Frame(frameainfo,height="60",width="300",relief="solid",bg="lightgreen",borderwidth=2)

frameainfoleft.pack(side=LEFT,fill=Y)

labelinputpatientname=Label(frameainfoleft,text="Input Patient's ID",bg="lightgreen",font=("calibri",17,"bold"))

labelinputpatientname.pack(anchor=CENTER,pady=20)

searchID=StringVar()

inputpatientname=Entry(frameainfoleft,textvariable=searchID,width="20",font=("calibri",15,"bold"))

inputpatientname.pack()

Search1button=Button(frameainfoleft,text="Search",command=lambda: Aisearch(searchID.get()),bg="yellow",font=("calibri",15,"bold"))

Search1button.pack(side=TOP,pady=8)

frameainfocenter=Frame(frameainfo,height="60",width="130",relief="solid",bg="lightgreen",borderwidth=2)

frameainfocenter.pack(side=LEFT,fill=BOTH,expand=True)

labelPatientid=Label(frameainfocenter,text="Patient ID: ",bg="lightgreen",font=("calibri",20,"bold"))

labelPatientid.grid(row=0,column=0,pady=10,sticky="w")

AiPatientid=StringVar()

Patientid=Entry(frameainfocenter,textvariable=AiPatientid,font=("calibri",20,"bold"))

Patientid.grid(row=0,column=1,padx=1,pady=17,sticky="w")

labelPatientid=Label(frameainfocenter,text="",bg="lightgreen",font=("calibri",20,"bold"))

labelPatientid.grid(row=0,column=2,sticky="e")

labelName=Label(frameainfocenter,text="Name: ",bg="lightgreen",font=("calibri",20,"bold"))

labelName.grid(row=1,column=0,padx=0,pady=17,sticky="w")

AiName=StringVar()

Name=Entry(frameainfocenter,textvariable=AiName,font=("calibri",20,"bold"))

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

GenderAi=StringVar()

GenderAi.set("")

labelGender=Label(frameainfocenter,text="Gender: ",bg="lightgreen",font=("calibri",20,"bold"))

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

i= IntVar()

r1 = Radiobutton(frameainfocenter,text="male",value=1,variable=GenderAi,bg="lightgreen",font=("calibri",20,"bold"))

r2 = Radiobutton(frameainfocenter,text="female",value=2,variable=GenderAi,bg="lightgreen",font=("calibri",20,"bold"))

r1.grid(row=2,column=1,sticky="w")

r2.grid(row=2,column=1,sticky="e",padx=0)

GenderAi.set(3)

labelAge=Label(frameainfocenter,text="Age: ",bg="lightgreen",font=("calibri",20,"bold"))

labelAge.grid(row=3,column=0,sticky="w",padx=0,pady=17)

AiAge=StringVar()

Age=Entry(frameainfocenter,textvariable=AiAge,font=("calibri",20,"bold"))

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

labelPhone=Label(frameainfocenter,text="Phone: ",bg="lightgreen",font=("calibri",20,"bold"))

labelPhone.grid(row=4,column=0,sticky="w",padx=0,pady=17)

AiPhone=StringVar()

Phone=Entry(frameainfocenter,textvariable=AiPhone,font=("calibri",20,"bold"))

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

labelAddress=Label(frameainfocenter,text="Address: ",bg="lightgreen",font=("calibri",20,"bold"))

labelAddress.grid(row=5,column=0,sticky="w",padx=0,pady=17)

AiAddress=StringVar()

Address=Entry(frameainfocenter,textvariable=AiAddress,font=("calibri",20,"bold"))

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

labelDisease=Label(frameainfocenter,text="Disease: ",bg="lightgreen",font=("calibri",20,"bold"))

labelDisease.grid(row=6,column=0,sticky="w",padx=0,pady=17)

AiDisease=StringVar()

Disease=Entry(frameainfocenter,textvariable=AiDisease,font=("calibri",20,"bold"))

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

labelCheckIn=Label(frameainfocenter,text="Check In",bg="lightgreen",font=("calibri",20,"bold"))

labelCheckIn.grid(row=7,column=0,sticky="w",padx=0,pady=17)

AiCheckIn=StringVar()

CheckIn=Entry(frameainfocenter,textvariable=AiCheckIn,font=("calibri",20,"bold"))

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

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_rowconfigure(5,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

frameainforight=Frame(frameainfo,height="60",width="300",relief="solid",borderwidth=2,bg="lightgreen")

frameainforight.pack(side=RIGHT,fill=Y)

labelBloodgroup=Label(frameainforight,text="Blood Group",bg="lightgreen",font=("calibri",17,"bold"))

labelBloodgroup.pack(padx=80,pady=10)

AiBloodgroup=StringVar()

Bloodgroup=Entry(frameainforight,textvariable=AiBloodgroup,width="20",font=("calibri",17,"bold"))

Bloodgroup.pack()

labelDoctorname=Label(frameainforight,text="Doctor Name",bg="lightgreen",font=("calibri",17,"bold"))

labelDoctorname.pack(padx=80,pady=10)

AiDoctorname=StringVar()

Doctorname=Entry(frameainforight,textvariable=AiDoctorname,width="20",font=("calibri",17,"bold"))

Doctorname.pack()

labelCheckOut=Label(frameainforight,text="Check OUT",bg="lightgreen",font=("calibri",17,"bold"))

labelCheckOut.pack(pady=10)

AiCheckOut=StringVar()

CheckOut=Entry(frameainforight,textvariable=AiCheckOut,width="20",font=("calibri",17,"bold"))

CheckOut.pack()

labelRoomNo=Label(frameainforight,text="Room No.",bg="lightgreen",font=("calibri",17,"bold"))

labelRoomNo.pack(pady=10)

AiRoomNo=StringVar()

RoomNo=Entry(frameainforight,textvariable=AiRoomNo,width="20",font=("calibri",17,"bold"))

RoomNo.pack()

labelRoomType=Label(frameainforight,text="Room Type",bg="lightgreen",font=("calibri",17,"bold"))

labelRoomType.pack(pady=10)

list=["Normal Ward","Personal Room"]

AiRoomtype=StringVar(frameainforight)

dropdownlist=OptionMenu(frameainforight,AiRoomtype,\*list)

dropdownlist.pack()

labelPrice=Label(frameainforight,text="Price.",bg="lightgreen",font=("calibri",17,"bold"))

labelPrice.pack(pady=10)

AiPrice=StringVar()

Price=Entry(frameainforight,textvariable=AiPrice,width="20",font=("calibri",17,"bold"))

Price.pack()

Updatebutton=Button(frameainforight,text="Update",command=Aiupdate,bg="yellow",width=15,bd=7,font=("calibri",15,"bold"))

Updatebutton.pack(side=LEFT,padx=50)

def Paadd():

global framemenuimage

global framehome

global frameprintbill

global framereceiptimage

global imagemenu

global imagehome

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framestaffregistration

global framestaffinformation

global framebill

global framedoctorrecord

global framereceipt

global framerireceiptimage

global PaPatientid

global PaName

global GenderPa

global PaAge

global PaPhone

global PaAddress

global PaDisease

global PaCheckIn

global PaBloodgroup

global PaDoctorname

global PaCheckOut

global PaRoomNo

global PaRoomtype

global PaPrice

global AiPatientid

global AiName

global GenderAi

global AiAge

global AiPhone

global AiAddress

global AiDisease

global AiCheckIn

global AiBloodgroup

global AiDoctorname

global AiCheckOut

global AiRoomNo

global AiRoomtype

global AiPrice

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global frameappointment

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

if PaPatientid.get() == "" or PaName.get() == "" or GenderPa.get() == "" or PaAge.get() == "" or PaPhone.get() == "" or PaDisease.get() == "" or PaCheckIn.get() == "" or PaCheckOut.get() == "" or PaRoomNo.get() == "" or PaPrice.get() == "":

result = tkMessageBox.showwarning('', 'Please Complete The Required Field', icon="warning")

else:

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query = "insert into pa values('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}')".format(PaPatientid.get(),PaName.get(),GenderPa.get(),PaAge.get(),PaPhone.get(),PaAddress.get(),PaDisease.get(),PaCheckIn.get(),PaBloodgroup.get(),PaDoctorname.get(),PaCheckOut.get(),PaRoomNo.get(),PaRoomtype.get(),PaPrice.get())

cursor.execute(query)

connection.commit()

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pa"

cursor.execute(query)

messagebox.showinfo("Congratulations ", "Data Added Successfully ")

def Adsearch(ID):

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pr where PrPatientid like '{0}' or PrName like '{0}' or PrPhone like '{0}'".format(ID)

cursor.execute(query)

for row in cursor:

PaPatientid.set(row[0])

PaName.set(row[1])

GenderPa.set(row[2])

PaAge.set(row[3])

PaPhone.set(row[4])

PaAddress.set(row[5])

PaDisease.set(row[6])

PaCheckIn.set(row[7])

PaBloodgroup.set(row[8])

PaDoctorname.set(row[9])

PaCheckOut.set(row[10])

def padmit():

global framemenuimage

global framehome

global frameprintbill

global framereceiptimage

global framerireceiptimage

global imagemenu

global imagehome

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framebill

global search\_byregistration

global search\_txtregistration

global search\_byadmitrecord

global search\_txtadmitrecord

global framestaffregistration

global framestaffinformation

global framerrecord

global framedoctorrecord

global framereceipt

global framemenuroot

global PaPatientid

global PaName

global GenderPa

global PaAge

global PaPhone

global PaAddress

global PaDisease

global PaCheckIn

global PaBloodgroup

global PaDoctorname

global PaCheckOut

global PaRoomNo

global PaRoomtype

global PaPrice

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global frameappointment

global frameappointmentright

global Patientadmit\_table

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterpadmit+=1

counterhome=0

counterpreg=0

counterrinfo=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterpadmit==1:

framepatientadmit=Frame(root,width=100,bg="lightgreen",height=100)

framepatientadmit.pack(side="right",fill=BOTH,expand=TRUE)

frameainfoleft=Frame(framepatientadmit,height="60",width="200",relief="solid",borderwidth=2,bg="lightgreen")

frameainfoleft.pack(side=LEFT,fill=Y)

framepatientadmittop=Frame(frameainfoleft,height="200",relief="solid",borderwidth=2,bg="sandybrown")

framepatientadmittop.pack(side=TOP,fill=X)

labelinputpatientname=Label(framepatientadmittop,text="Input Patient's ID",bg="sandybrown",font=("calibri",17,"bold"))

labelinputpatientname.pack(anchor=CENTER,pady=10)

framepatientadmittopsearch=Frame(frameainfoleft,bg="lightgreen")

framepatientadmittopsearch.pack(side=TOP)

searchID=StringVar()

inputpatientname=Entry(framepatientadmittopsearch,textvariable=searchID,width="15",font=("calibri",15,"bold"))

inputpatientname.pack(pady=10)

Searchbutton=Button(framepatientadmittopsearch,text="Search",command=lambda:Adsearch(searchID.get()),bg="yellow",font=("calibri",15,"bold"))

Searchbutton.pack(side=TOP,pady=8)

framepatientadmittop=Frame(framepatientadmit,height="200",relief="solid",borderwidth=2,bg="sandybrown")

framepatientadmittop.pack(side=TOP,fill=X)

labelpatientrinfotext=Label(framepatientadmittop,text="Patient Admit",bg="sandybrown",font=("calibri",23,"bold"))

labelpatientrinfotext.pack(anchor="center")

framepatientadmitcenter=Frame(framepatientadmit,height="60",width="130",bg="lightgreen",relief="solid",borderwidth=2)

framepatientadmitcenter.pack(side=LEFT,fill=BOTH,expand=True)

labelPatientid=Label(framepatientadmitcenter,text="Patient ID: ",bg="lightgreen",font=("calibri",20,"bold"))

labelPatientid.grid(row=0,column=0,padx=40,pady=10,sticky="w")

PaPatientid=StringVar()

Patientid=Entry(framepatientadmitcenter,textvariable=PaPatientid,font=("calibri",20,"bold"))

Patientid.grid(row=0,column=1,padx=1,pady=17,sticky="w")

labelName=Label(framepatientadmitcenter,text="Name: ",bg="lightgreen",font=("calibri",20,"bold"))

labelName.grid(row=1,column=0,padx=40,pady=17,sticky="w")

PaName=StringVar()

Name=Entry(framepatientadmitcenter,textvariable=PaName,font=("calibri",20,"bold"))

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

GenderPa=StringVar()

GenderPa.set("")

labelGender=Label(framepatientadmitcenter,text="Gender: ",bg="lightgreen",font=("calibri",20,"bold"))

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

i= IntVar()

r1 = Radiobutton(framepatientadmitcenter,text="male",value=1,variable=GenderPa,bg="lightgreen",font=("calibri",20,"bold"))

r2 = Radiobutton(framepatientadmitcenter,text="female",value=2,variable=GenderPa,bg="lightgreen",font=("calibri",20,"bold"))

r1.grid(row=2,column=1,sticky="w")

r2.grid(row=2,column=1,sticky="e",padx=0)

GenderPa.set(3)

labelAge=Label(framepatientadmitcenter,text="Age: ",bg="lightgreen",font=("calibri",20,"bold"))

labelAge.grid(row=3,column=0,sticky="w",padx=40,pady=17)

PaAge=StringVar()

Age=Entry(framepatientadmitcenter,textvariable=PaAge,font=("calibri",20,"bold"))

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

labelPhone=Label(framepatientadmitcenter,text="Phone: ",bg="lightgreen",font=("calibri",20,"bold"))

labelPhone.grid(row=4,column=0,sticky="w",padx=40,pady=17)

PaPhone=StringVar()

Phone=Entry(framepatientadmitcenter,textvariable=PaPhone,font=("calibri",20,"bold"))

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

labelAddress=Label(framepatientadmitcenter,text="Address: ",bg="lightgreen",font=("calibri",20,"bold"))

labelAddress.grid(row=5,column=0,padx=40,pady=17,sticky="w")

PaAddress=StringVar()

Address=Entry(framepatientadmitcenter,textvariable=PaAddress,font=("calibri",20,"bold"))

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

labelDisease=Label(framepatientadmitcenter,text="Disease: ",bg="lightgreen",font=("calibri",20,"bold"))

labelDisease.grid(row=6,column=0,padx=40,pady=17,sticky="w")

PaDisease=StringVar()

Disease=Entry(framepatientadmitcenter,textvariable=PaDisease,font=("calibri",20,"bold"))

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

labelCheckIn=Label(framepatientadmitcenter,text="Check In",bg="lightgreen",font=("calibri",20,"bold"))

labelCheckIn.grid(row=7,column=0,padx=40,pady=17,sticky="w")

PaCheckIn=StringVar()

CheckIn=Entry(framepatientadmitcenter,textvariable=PaCheckIn,font=("calibri",20,"bold"))

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

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_rowconfigure(5,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

framepatientadmitright=Frame(framepatientadmit,height="60",width="300",relief="solid",bg="lightgreen",borderwidth=2)

framepatientadmitright.pack(side=RIGHT,fill=Y)

labelBloodgroup=Label(framepatientadmitright,text="Blood group",bg="lightgreen",font=("calibri",17,"bold"))

labelBloodgroup.pack(padx=80,pady=10)

PaBloodgroup=StringVar()

Bloodgroup=Entry(framepatientadmitright,textvariable=PaBloodgroup,width="20",font=("calibri",17,"bold"))

Bloodgroup.pack()

labelDoctorname=Label(framepatientadmitright,text="Doctor Name",bg="lightgreen",font=("calibri",17,"bold"))

labelDoctorname.pack(padx=80,pady=10)

PaDoctorname=StringVar()

Doctorname=Entry(framepatientadmitright,textvariable=PaDoctorname,width="20",font=("calibri",17,"bold"))

Doctorname.pack()

labelCheckOut=Label(framepatientadmitright,text="Check OUT",bg="lightgreen",font=("calibri",17,"bold"))

labelCheckOut.pack(pady=10)

PaCheckOut=StringVar()

CheckOut=Entry(framepatientadmitright,textvariable=PaCheckOut,width="20",font=("calibri",17,"bold"))

CheckOut.pack()

labelRoomNo=Label(framepatientadmitright,text="Room No.",bg="lightgreen",font=("calibri",17,"bold"))

labelRoomNo.pack(pady=10)

PaRoomNo=StringVar()

RoomNo=Entry(framepatientadmitright,textvariable=PaRoomNo,width="20",font=("calibri",17,"bold"))

RoomNo.pack()

labelRoomType=Label(framepatientadmitright,text="Room Type",bg="lightgreen",font=("calibri",17,"bold"))

labelRoomType.pack(pady=10)

list=["Normal Ward","Personal Room"]

PaRoomtype=StringVar(framepatientadmitright)

dropdownlist=OptionMenu(framepatientadmitright,PaRoomtype,\*list)

dropdownlist.pack()

labelPrice=Label(framepatientadmitright,text="Price.",bg="lightgreen",font=("calibri",17,"bold"))

labelPrice.pack(pady=10)

PaPrice=StringVar()

Price=Entry(framepatientadmitright,textvariable=PaPrice,width="20",font=("calibri",17,"bold"))

Price.pack()

ADDAbutton=Button(framepatientadmitright,text="ADD",command=Paadd,bg="yellow",width=15,bd=7,font=("calibri",15,"bold"))

ADDAbutton.pack(pady=9)

def Risearch(ID):

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pr where PrPatientid like '{0}' or PrName like '{0}' or PrPhone like '{0}'".format(ID)

cursor.execute(query)

for row in cursor:

RiPatientid.set(row[0])

RiName.set(row[1])

GenderRi.set(row[2])

RiAge.set(row[3])

RiPhone.set(row[4])

RiAddress.set(row[5])

RiDisease.set(row[6])

RiCheckIn.set(row[7])

RiBloodgroup.set(row[8])

RiDoctorname.set(row[9])

RiRoomNo.set(row[10])

def Riupdate():

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="update pr set PrName = '{}',GenderPr = '{}',PrAge = '{}',PrPhone = '{}',PrAddress = '{}',PrDisease = '{}',PrCheckIn = '{}',PrBloodgroup = '{}',PrDoctorname = '{}',PrRoomNo = '{}' where PrPatientid = '{}'".format(RiName.get(),GenderRi.get(),RiAge.get(),RiPhone.get(),RiAddress.get(),RiDisease.get(),RiCheckIn.get(),RiBloodgroup.get(),RiDoctorname.get(),RiRoomNo.get(),RiPatientid.get())

cursor.execute(query)

connection.commit()

messagebox.showinfo("Congratulations ", "Data Updated Successfully")

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pr"

cursor.execute(query)

for row in cursor:

RiPatientid.set(row[0])

RiName.set(row[1])

GenderRi.set(row[2])

RiAge.set(row[3])

RiPhone.set(row[4])

RiAddress.set(row[5])

RiDisease.set(row[6])

RiCheckIn.set(row[7])

RiBloodgroup.set(row[8])

RiDoctorname.set(row[9])

RiRoomNo.set(row[10])

def Pradd():

global framemenuimage

global framepatientregistration

global frameprintbill

global framerinfo

global framereceiptimage

global imagemenu

global imagehome

global framereceipt

global framerireceiptimage

global PrPatientid

global PrName

global GenderPr

global PrAge

global PrPhone

global PrAddress

global PrDisease

global PrCheckIn

global PrBloodgroup

global PrDoctorname

global PrRoomNo

global RiPatientid

global RiName

global GenderRi

global RiAge

global RiPhone

global RiAddress

global RiDisease

global RiCheckIn

global RiBloodgroup

global RiDoctorname

global RiRoomNo

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global appointmenticon

global framebillrecord

global frameappointment

global Patientregistration\_table

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

if PrName.get() == "" or GenderPr.get() == "" or PrAge.get() == "" or PrPhone.get() == "" or PrAddress.get() == "" or PrDisease.get() == "":

result = tkMessageBox.showwarning('', 'Please Complete The Required Field', icon="warning")

else:

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

x=random.randint(0, 10000000)

randomRef = str(x)

randomRef = '2020-' + randomRef

PrPatientid.set(randomRef)

query = "insert into pr values('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}')".format(PrPatientid.get(),PrName.get(),GenderPr.get(),int(PrAge.get()),PrPhone.get(),PrAddress.get(),PrDisease.get(),PrCheckIn.get(),PrBloodgroup.get(),PrDoctorname.get(),PrRoomNo.get())

cursor.execute(query)

connection.commit()

connection = sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="select \* from pr"

cursor.execute(query)

messagebox.showinfo("Congratulations ", "Data Added Successfully ")

def rinfo():

global framemenuimage

global framehome

global frameprintbill

global framereceiptimage

global imagemenu

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framemenuroot

global framestaffregistration

global framestaffinformation

global framerrecord

global framedoctorrecord

global framebill

global framereceipt

global framerireceiptimage

global PrPatientid

global PrName

global GenderPr

global PrAge

global PrPhone

global PrAddress

global PrDisease

global PrCheckIn

global PrBloodgroup

global PrDoctorname

global PrRoomNo

global RiPatientid

global RiName

global GenderRi

global RiAge

global RiPhone

global RiAddress

global RiDisease

global RiCheckIn

global RiBloodgroup

global RiDoctorname

global RiRoomNo

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global framebillrecord

global frameappointment

global frameappointmentright

global Patientregistration\_table

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterrinfo+=1

counterhome=0

counterpreg=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterrinfo==1:

framerinfo=Frame(root,width=100,height=100)

framerinfo.pack(side="right",fill=BOTH,expand=TRUE)

framerinfotop=Frame(framerinfo,height="200",relief="solid",borderwidth=2,bg="sandybrown")

framerinfotop.pack(side=TOP,fill=X)

labelpatientrinfotext=Label(framerinfotop,text="Registration Information",bg="sandybrown",font=("calibri",23,"bold"))

labelpatientrinfotext.pack(anchor="center")

framerinfoleft=Frame(framerinfo,height="60",width="300",relief="solid",borderwidth=2,bg="lightgreen")

framerinfoleft.pack(side=LEFT,fill=Y)

labelinputpatientname=Label(framerinfoleft,text="Input Patient's ID",bg="lightgreen",font=("calibri",17,"bold"))

labelinputpatientname.pack(anchor=CENTER,pady=20)

searchID=StringVar()

inputpatientname=Entry(framerinfoleft,textvariable=searchID,width="20",font=("calibri",15,"bold"))

inputpatientname.pack()

Searchbutton=Button(framerinfoleft,text="Search",command=lambda:Risearch(searchID.get()),bg="yellow",font=("calibri",15,"bold"))

Searchbutton.pack(side=TOP,pady=8)

framerinfocenter=Frame(framerinfo,height="60",width="130",bg="lightgreen",relief="solid",borderwidth=2)

framerinfocenter.pack(side=LEFT,fill=BOTH,expand=True)

labelPatientid=Label(framerinfocenter,text="Patient ID: ",bg="lightgreen",font=("calibri",20,"bold"))

labelPatientid.grid(row=0,column=0,pady=10,sticky="w")

RiPatientid=StringVar()

Patientid=Entry(framerinfocenter,textvariable=RiPatientid,font=("calibri",20,"bold"))

Patientid.grid(row=0,column=1,padx=1,pady=17,sticky="w")

labelPatientid=Label(framerinfocenter,text="",bg="lightgreen",font=("calibri",20,"bold"))

labelPatientid.grid(row=0,column=2,sticky="e")

labelName=Label(framerinfocenter,text="Name: ",bg="lightgreen",font=("calibri",20,"bold"))

labelName.grid(row=1,column=0,padx=0,pady=17,sticky="w")

RiName=StringVar()

Name=Entry(framerinfocenter,textvariable=RiName,font=("calibri",20,"bold"))

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

GenderRi=StringVar()

GenderRi.set("")

labelGender=Label(framerinfocenter,text="Gender: ",bg="lightgreen",font=("calibri",20,"bold"))

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

r1 = Radiobutton(framerinfocenter,text="male",value='Male',variable=GenderRi,bg="lightgreen",font=("calibri",20,"bold"))

r1.grid(row=2,column=1,sticky="w")

r2 = Radiobutton(framerinfocenter,text="female",value='Female',variable=GenderRi,bg="lightgreen",font=("calibri",20,"bold"))

r2.grid(row=2,column=1,sticky="e",padx=0)

GenderRi.set(3)

labelAge=Label(framerinfocenter,text="Age: ",bg="lightgreen",font=("calibri",20,"bold"))

labelAge.grid(row=3,column=0,sticky="w",padx=0,pady=17)

RiAge=StringVar()

Age=Entry(framerinfocenter,textvariable=RiAge,font=("calibri",20,"bold"))

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

labelPhone=Label(framerinfocenter,text="Phone: ",bg="lightgreen",font=("calibri",20,"bold"))

labelPhone.grid(row=4,column=0,sticky="w",padx=0,pady=17)

RiPhone=StringVar()

Phone=Entry(framerinfocenter,textvariable=RiPhone,font=("calibri",20,"bold"))

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

labelAddress=Label(framerinfocenter,text="Address: ",bg="lightgreen",font=("calibri",20,"bold"))

labelAddress.grid(row=5,column=0,sticky="w",padx=0,pady=17)

RiAddress=StringVar()

Address=Entry(framerinfocenter,textvariable=RiAddress,font=("calibri",20,"bold"))

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

labelDisease=Label(framerinfocenter,text="Disease: ",bg="lightgreen",font=("calibri",20,"bold"))

labelDisease.grid(row=6,column=0,sticky="w",padx=0,pady=17)

RiDisease=StringVar()

Disease=Entry(framerinfocenter,textvariable=RiDisease,font=("calibri",20,"bold"))

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

labelCheckIn=Label(framerinfocenter,text="Check In",bg="lightgreen",font=("calibri",20,"bold"))

labelCheckIn.grid(row=7,column=0,sticky="w",padx=0,pady=17)

RiCheckIn=StringVar()

CheckIn=Entry(framerinfocenter,textvariable=RiCheckIn,font=("calibri",20,"bold"))

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

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_rowconfigure(5,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

framerinforight=Frame(framerinfo,height="60",width="300",bg="lightgreen",relief="solid",borderwidth=2)

framerinforight.pack(side=RIGHT,fill=Y)

labelDoctorname=Label(framerinforight,text="Blood Group",bg="lightgreen",font=("calibri",17,"bold"))

labelDoctorname.pack(padx=80,pady=10)

RiBloodgroup=StringVar()

Bloodgroup=Entry(framerinforight,textvariable=RiBloodgroup ,width="20",font=("calibri",17,"bold"))

Bloodgroup.pack()

labelDoctorname=Label(framerinforight,text="Doctor Name",bg="lightgreen",font=("calibri",17,"bold"))

labelDoctorname.pack(padx=80,pady=10)

RiDoctorname=StringVar()

Doctorname=Entry(framerinforight,textvariable=RiDoctorname ,width="20",font=("calibri",17,"bold"))

Doctorname.pack()

labelDoctorname=Label(framerinforight,text="Room No",bg="lightgreen",font=("calibri",17,"bold"))

labelDoctorname.pack(padx=80,pady=10)

RiRoomNo=StringVar()

RoomNo=Entry(framerinforight,textvariable=RiRoomNo,width="20",font=("calibri",17,"bold"))

RoomNo.pack()

Updatebutton=Button(framerinforight,text="Update",command=Riupdate,bg="yellow",bd=7,font=("calibri",15,"bold"))

Updatebutton.pack(side=RIGHT,padx=35,pady=100)

Updatebutton=Button(framerinforight,text="Receipt",command=Rireceipt,bg="yellow",bd=7,font=("calibri",15,"bold"))

Updatebutton.pack(side=RIGHT,padx=35,pady=100)

def iPrint():

global txtReceipt

q=txtReceipt.get("1.0",END)

filename=tempfile.mktemp(".txt")

open(filename,"w").write(q)

os.startfile(filename,"print")

def billPrint():

global txtReceiptbill

q=txtReceiptbill.get("1.0",END)

filename=tempfile.mktemp(".txt")

open(filename,"w").write(q)

os.startfile(filename,"print")

def home():

global framemenuimage

global framehome

global frameprintbill

global framereceiptimage

global imagehome

global imagemenu

global framereceipt

global framerireceiptimage

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framemenuroot

global framestaffregistration

global framestaffinformation

global framerrecord

global framedoctorrecord

global framebill

global PrPatientid

global PrName

global GenderPr

global PrAge

global PrPhone

global PrAddress

global PrDisease

global PrCheckIn

global PrBloodgroup

global PrDoctorname

global PrRoomNo

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global framebillrecord

global frameappointment

global frameappointmentright

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterhome+=1

counterrinfo=0

counterpreg=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterreceipt=0

counterrireceipt=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

frame1.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterhome==1:

framehome=Frame(root)

framehome.pack(side=TOP,fill=BOTH,expand=True)

imagehome=PhotoImage(file="hospital.png")

label=ttk.Label(framehome,image=imagehome)

label.pack(side=LEFT,fill=BOTH,expand=True)

def Receipt():

global framemenuimage

global framehome

global frameprintbill

global BlIdentity

global BlName

global GenderBl

global BlAge

global Blcheckin

global BlDisease

global BlPhone

global BlAddress

global BlDoctorname

global Blcheckout

global BlDoctorcharges

global BlLabtest

global TreatmentchargesBl

global BlProcedurecharges

global BlOthercharges

global BlMiscellaneouscharges

global BlRoomcharges

global BlTotal

global framereceiptimage

global receiptimage

global receiptimager

global imagemenu

global framereceipt

global framerireceiptimage

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framemenuroot

global framestaffregistration

global framestaffinformation

global framerrecord

global framedoctorrecord

global framebill

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global framebillrecord

global frameappointment

global frameappointmentright

global txtReceipt

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterreceipt+=1

counterrireceipt=0

counterrinfo=0

counterpreg=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterhome=0

counteradmitrecord=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

frame1.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterreceipt==1:

framereceiptimage=Frame(root,bg="lightgreen")

framereceiptimage.pack(side=TOP,fill=X)

#receiptimage=PhotoImage(file="VYAS.png")

#label=ttk.Label(framereceiptimage,image=receiptimage)

#label.grid(row=0,column=0)

label=Label(framereceiptimage,text="SH 17 , Jant,Mahendergarh, Haryana 123031",font=("calibri",15,"bold"),bg="lightgreen")

label.grid(row=0,column=1,padx=200)

receiptimager=PhotoImage(file="plussign.png")

label=ttk.Label(framereceiptimage,image=receiptimager)

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

framereceipt=Frame(root,bg="lightgreen")

framereceipt.pack(side="top",fill=BOTH,expand=True)

framereceipttop=Frame(framereceipt,bg="lightgreen",height=20)

framereceipttop.pack(side=TOP,fill=X)

labelreceipt=Label(framereceipttop,text="Receipt :-",bg="lightgreen",fg="black",font=('arial',20,'bold'))

labelreceipt.pack(side=LEFT,padx=150,pady=10)

txtReceipt=Text(framereceipt,width=65,height=12,bg='white',bd=4,font=('arial',12,'bold'))

txtReceipt.pack(anchor="nw",padx=100)

txtReceipt.insert(END,'Patient id:\t\t\t'+PrPatientid.get() +'\n')

txtReceipt.insert(END,'Name:\t\t\t' + PrName.get() +'\n')

txtReceipt.insert(END,'Gender:\t\t\t'+ GenderPr.get()+'\n')

txtReceipt.insert(END,'Age:\t\t\t'+ PrAge.get()+'\n')

txtReceipt.insert(END,'Phone No:\t\t\t'+ PrPhone.get()+'\n')

txtReceipt.insert(END,'Address:\t\t\t'+ PrAddress.get()+'\n')

txtReceipt.insert(END,'Disease:\t\t\t'+ PrDisease.get()+'\n')

txtReceipt.insert(END,'Check In:\t\t\t'+ PrCheckIn.get()+'\n')

txtReceipt.insert(END,'Bloodgroup:\t\t\t'+ PrBloodgroup.get()+'\n')

txtReceipt.insert(END,'Doctor Name:\t\t\t'+ PrDoctorname.get()+'\n')

txtReceipt.insert(END,'Room No:\t\t\t'+ PrRoomNo.get()+'\n')

labelreceipt=Label(framereceipt,text="Medicine: - ",fg="black",bg="lightgreen",font=('arial',20,'bold'))

labelreceipt.pack(anchor="nw",padx=150,pady=10)

buttonprint=Button(framereceipt,text="Print",width=7,fg="red",font=('arial',20,'bold'),command=iPrint)

buttonprint.pack(side=BOTTOM)

def Rireceipt():

global framemenuimage

global framehome

global frameprintbill

global BlIdentity

global BlName

global GenderBl

global BlAge

global Blcheckin

global BlDisease

global BlPhone

global BlAddress

global BlDoctorname

global Blcheckout

global BlDoctorcharges

global BlLabtest

global TreatmentchargesBl

global BlProcedurecharges

global BlOthercharges

global BlMiscellaneouscharges

global BlRoomcharges

global BlTotal

global framereceiptimage

global receiptimage

global receiptimager

global imagemenu

global framereceipt

global framerireceiptimage

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framemenuroot

global framestaffregistration

global framestaffinformation

global framerrecord

global framedoctorrecord

global framebill

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global framebillrecord

global frameappointment

global frameappointmentright

global txtReceipt

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterrireceipt+=1

counterreceipt=0

counterrinfo=0

counterpreg=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterhome=0

counteradmitrecord=0

counterprintbill=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

frame1.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterrireceipt==1:

framerireceiptimage=Frame(root,bg="lightgreen")

framerireceiptimage.pack(side=TOP,fill=X)

receiptimage=PhotoImage(file="VYAS.png")

label=ttk.Label(framerireceiptimage,image=receiptimage)

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

label=Label(framerireceiptimage,text="SH 17 , Jant,Mahendergarh, Haryana 123031",font=("calibri",15,"bold"),bg="lightgreen")

label.grid(row=0,column=1,padx=200)

receiptimager=PhotoImage(file="plussign.png")

label=ttk.Label(framerireceiptimage,image=receiptimager)

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

framerireceipt=Frame(root,bg="lightgreen")

framerireceipt.pack(side="top",fill=BOTH,expand=True)

framerireceipttop=Frame(framerireceipt,bg="lightgreen",height=20)

framerireceipttop.pack(side=TOP,fill=X)

labelreceipt=Label(framerireceipttop,text="Receipt :-",bg="lightgreen",fg="black",font=('arial',20,'bold'))

labelreceipt.pack(side=LEFT,padx=150,pady=10)

txtReceipt=Text(framerireceipt,width=65,height=12,bg='white',bd=4,font=('arial',12,'bold'))

txtReceipt.pack(anchor="nw",padx=100)

txtReceipt.insert(END,'Patient id:\t\t\t'+RiPatientid.get() +'\n')

txtReceipt.insert(END,'Name:\t\t\t' + RiName.get() +'\n')

txtReceipt.insert(END,'Gender:\t\t\t'+ GenderRi.get()+'\n')

txtReceipt.insert(END,'Age:\t\t\t'+ RiAge.get()+'\n')

txtReceipt.insert(END,'Phone No:\t\t\t'+ RiPhone.get()+'\n')

txtReceipt.insert(END,'Address:\t\t\t'+ RiAddress.get()+'\n')

txtReceipt.insert(END,'Disease:\t\t\t'+ RiDisease.get()+'\n')

txtReceipt.insert(END,'Check In:\t\t\t'+ RiCheckIn.get()+'\n')

txtReceipt.insert(END,'Bloodgroup:\t\t\t'+ RiBloodgroup.get()+'\n')

txtReceipt.insert(END,'Doctor Name:\t\t\t'+ RiDoctorname.get()+'\n')

txtReceipt.insert(END,'Room No:\t\t\t'+ RiRoomNo.get()+'\n')

labelreceipt=Label(framerireceipt,text="Medicine: - ",fg="black",bg="lightgreen",font=('arial',20,'bold'))

labelreceipt.pack(anchor="nw",padx=150,pady=10)

buttonprint=Button(framerireceipt,text="Print",width=7,fg="red",font=('arial',20,'bold'),command=iPrint)

buttonprint.pack(side=BOTTOM)

def printbill():

global framemenuimage

global framehome

global frameprintbill

global BlIdentity

global BlName

global GenderBl

global BlAge

global Blcheckin

global BlDisease

global BlPhone

global BlAddress

global BlDoctorname

global Blcheckout

global BlDoctorcharges

global BlLabtest

global TreatmentchargesBl

global BlProcedurecharges

global BlOthercharges

global BlMiscellaneouscharges

global BlRoomcharges

global BlTotal

global framereceiptimage

global receiptimage

global receiptimager

global imagemenu

global framereceipt

global framerireceiptimage

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framemenuroot

global framestaffregistration

global framestaffinformation

global framerrecord

global framedoctorrecord

global framebill

global frameprintbill

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global framebillrecord

global frameappointment

global frameappointmentright

global txtReceiptbill

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterprintbill+=1

counterrinfo=0

counterpreg=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterhome=0

counterreceipt=0

counterrireceipt=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

frame1.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framepatientregistration.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterprintbill==1:

frameprintbill=Frame(root,bg="lightgreen")

frameprintbill.pack(side="top",fill=BOTH,expand=True)

txtReceiptbill=Text(frameprintbill,width=100,height=17,bg='white',bd=4,font=('arial',12,'bold'))

txtReceiptbill.pack(anchor="nw",padx=100)

txtReceiptbill.insert(END,'Patient id:-\t\t'+BlIdentity.get() + '\t\t' +'Name:-\t\t\t' + BlName.get() +'\n')

txtReceiptbill.insert(END,'Gender:-\t\t'+ GenderBl.get()+ '\t\t' +'Age:-\t\t\t'+ BlAge.get()+'\n')

txtReceiptbill.insert(END,'CheckIn:-\t\t'+ Blcheckin.get()+ '\t\t' +'CheckOut:-\t\t\t'+ Blcheckout.get()+'\n')

txtReceiptbill.insert(END,'Disease:-\t\t'+ BlDisease.get()+ '\t\t' +'Phone No:-\t\t\t'+ BlPhone.get()+'\n')

txtReceiptbill.insert(END,'Address:-\t\t'+ BlAddress.get()+ '\t\t' +'Doctor name:-\t\t\t'+ BlDoctorname.get()+'\n'+'\n')

txtReceiptbill.insert(END,'Doctor Charges:\t\t\t'+BlDoctorcharges.get() +'\n')

txtReceiptbill.insert(END,'Lab Test:-\t\t\t' + BlLabtest.get() +'\n')

txtReceiptbill.insert(END,'Treatment Charges:-\t\t\t'+ TreatmentchargesBl.get()+'\n')

txtReceiptbill.insert(END,'Procedure Charges:-\t\t\t'+ BlProcedurecharges.get()+'\n')

txtReceiptbill.insert(END,'Other Charges:-\t\t\t'+ BlOthercharges.get()+'\n')

txtReceiptbill.insert(END,'Miscellaneous Charges:-\t\t\t'+ BlMiscellaneouscharges.get()+'\n')

txtReceiptbill.insert(END,'Room Charges:-\t\t\t'+ BlRoomcharges.get()+'\n')

txtReceiptbill.insert(END,'Total:-\t\t\t'+ BlTotal.get()+'\n')

buttonbill=Button(frameprintbill,text="Print",font=('arial',12,'bold'),fg="red",command=billPrint)

buttonbill.pack(side=BOTTOM)

def preg():

global framemenuimage

global framehome

global frameprintbill

global imagemenu

global framereceiptimage

global framereceipt

global framerireceiptimage

global framepatientregistration

global framepatientadmit

global frameadmitrecord

global framerinfo

global frameainfo

global framemenuroot

global framestaffregistration

global framestaffinformation

global framerrecord

global framedoctorrecord

global framebill

global PrPatientid

global PrName

global GenderPr

global PrAge

global PrPhone

global PrAddress

global PrDisease

global PrCheckIn

global PrBloodgroup

global PrDoctorname

global PrRoomNo

global homeicon

global patientregistrationicon

global patientadmiticon

global registrationinformationicon

global admitinformationicon

global doctorregistrationicon

global doctorinformationicon

global registrationrecordicon

global admitrecordicon

global doctorrecordicon

global generatebillicon

global billrecordicon

global framebillrecord

global frameappointment

global frameappointmentright

global Patientregistration\_table

global counterlogin

global counterhome

global counterpreg

global counterrinfo

global counterpadmit

global counterainfo

global countersreg

global countersinfo

global counterrrecord

global counteradmitrecord

global counterdrecord

global counterbill

global counterreceipt

global counterrireceipt

global counterprintbill

global counterbillrecord

global counterappointment

counterpreg+=1

counterrinfo=0

counterprintbill=0

counterpadmit=0

counterainfo=0

countersreg=0

countersinfo=0

counterrrecord=0

counteradmitrecord=0

counterdrecord=0

counterbill=0

counterlogin=0

counterhome=0

counterreceipt=0

counterrireceipt=0

counterbillrecord=0

counterappointment=0

try:

framemenuimage.destroy()

except:

pass

try:

frameprintbill.destroy()

except:

pass

try:

frame1.destroy()

except:

pass

try:

framereceiptimage.destroy()

except:

pass

try:

framereceipt.destroy()

except:

pass

try:

framerireceiptimage.destroy()

except:

pass

try:

framehome.destroy()

except:

pass

try:

framerinfo.destroy()

except:

pass

try:

framepatientadmit.destroy()

except:

pass

try:

frameadmitrecord.destroy()

except:

pass

try:

frameainfo.destroy()

except:

pass

try:

framestaffregistration.destroy()

except:

pass

try:

framestaffinformation.destroy()

except:

pass

try:

framerrecord.destroy()

except:

pass

try:

framebill.destroy()

except:

pass

try:

framedoctorrecord.destroy()

except:

pass

try:

framebillrecord.destroy()

except:

pass

try:

frameappointment.destroy()

except:

pass

try:

frameappointmentright.destroy()

except:

pass

if counterpreg==1:

framepatientregistration=Frame(root,width=100,height=100,bg="lightgreen")

framepatientregistration.pack(side="right",fill=BOTH,expand=TRUE)

framepatientregistrationtop=Frame(framepatientregistration,height="200",relief="solid",borderwidth=2,bg="sandybrown")

framepatientregistrationtop.pack(side=TOP,fill=X)

labelpatientregistrationtext=Label(framepatientregistrationtop,text="Patient Registration",bg="sandybrown",font=("calibri",23,"bold"))

labelpatientregistrationtext.pack(anchor="center")

framepatientregistrationcenter=Frame(framepatientregistration,bg="lightgreen",height="60",width="130",relief="solid",borderwidth=2)

framepatientregistrationcenter.pack(side=LEFT,fill=BOTH,expand=True)

labelPatientid=Label(framepatientregistrationcenter,bg="lightgreen",text="Patient ID: ",font=("calibri",20,"bold"))

labelPatientid.grid(row=0,column=0,padx=90,pady=10,sticky="w")

PrPatientid=StringVar()

Patientid=Entry(framepatientregistrationcenter,textvariable=PrPatientid,font=("calibri",20,"bold"))

Patientid.grid(row=0,column=1,padx=1,pady=17,sticky="w")

labelName=Label(framepatientregistrationcenter,text="Name: ",bg="lightgreen",font=("calibri",20,"bold"))

labelName.grid(row=1,column=0,padx=90,pady=17,sticky="w")

PrName=StringVar()

Name=Entry(framepatientregistrationcenter,textvariable=PrName,font=("calibri",20,"bold"))

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

GenderPr=StringVar()

GenderPr.set("")

labelGender=Label(framepatientregistrationcenter,text="Gender: ",bg="lightgreen",font=("calibri",20,"bold"))

labelGender.grid(row=2,column=0,sticky="w",padx=90)

i= IntVar()

r1 = Radiobutton(framepatientregistrationcenter,text="male",value="Male",bg="lightgreen",variable=GenderPr,font=("calibri",20,"bold"))

r2 = Radiobutton(framepatientregistrationcenter,text="female",value="Female",bg="lightgreen",variable=GenderPr,font=("calibri",20,"bold"))

r1.grid(row=2,column=1,sticky="w")

r2.grid(row=2,column=1,sticky="e",padx=0)

GenderPr.set(3)

labelAge=Label(framepatientregistrationcenter,text="Age: ",bg="lightgreen",font=("calibri",20,"bold"))

labelAge.grid(row=3,column=0,sticky="w",padx=90,pady=17)

PrAge=StringVar()

Age=Entry(framepatientregistrationcenter,textvariable=PrAge,font=("calibri",20,"bold"))

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

labelPhone=Label(framepatientregistrationcenter,text="Phone: ",bg="lightgreen",font=("calibri",20,"bold"))

labelPhone.grid(row=4,column=0,sticky="w",padx=90,pady=17)

PrPhone=StringVar()

Phone=Entry(framepatientregistrationcenter,textvariable=PrPhone,font=("calibri",20,"bold"))

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

labelAddress=Label(framepatientregistrationcenter,text="Address: ",bg="lightgreen",font=("calibri",20,"bold"))

labelAddress.grid(row=5,column=0,sticky="w",padx=90,pady=17)

PrAddress=StringVar()

Address=Entry(framepatientregistrationcenter,textvariable=PrAddress,font=("calibri",20,"bold"))

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

labelDisease=Label(framepatientregistrationcenter,text="Disease: ",bg="lightgreen",font=("calibri",20,"bold"))

labelDisease.grid(row=6,column=0,sticky="w",padx=90,pady=17)

PrDisease=StringVar()

Disease=Entry(framepatientregistrationcenter,textvariable=PrDisease,font=("calibri",20,"bold"))

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

labelCheckIn=Label(framepatientregistrationcenter,text="Check IN",bg="lightgreen",font=("calibri",20,"bold"))

labelCheckIn.grid(row=7,column=0,sticky="w",padx=90,pady=17)

CheckIn=Entry(framepatientregistrationcenter,textvariable=PrCheckIn,font=("calibri",20,"bold"))

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

root.grid\_rowconfigure(0,weight=1)

root.grid\_rowconfigure(1,weight=1)

root.grid\_rowconfigure(2,weight=1)

root.grid\_rowconfigure(3,weight=1)

root.grid\_rowconfigure(4,weight=1)

root.grid\_rowconfigure(5,weight=1)

root.grid\_rowconfigure(6,weight=1)

root.grid\_columnconfigure(0,weight=1)

root.grid\_columnconfigure(1,weight=1)

framepatientregistrationright=Frame(framepatientregistration,height="60",width="300",bg="lightgreen",relief="solid",borderwidth=2)

framepatientregistrationright.pack(side=RIGHT,fill=Y)

labelBloodgroup=Label(framepatientregistrationright,text="Blood Group",bg="lightgreen",font=("calibri",17,"bold"))

labelBloodgroup.pack(pady=10)

PrBloodgroup=StringVar()

Bloodgroup=Entry(framepatientregistrationright,textvariable=PrBloodgroup ,width="20",font=("calibri",17,"bold"))

Bloodgroup.pack()

labelDoctorname=Label(framepatientregistrationright,text="Doctor Name",bg="lightgreen",font=("calibri",17,"bold"))

labelDoctorname.pack(pady=10)

list=["Hemant[Dentist]","Hemant yadav[Cardiologist]","Mintu[Pediatricians]","Bhavesh[Physician]","Asha[Therapist]","Manish[Diagnostic]"]

PrDoctorname=StringVar(framepatientregistrationright)

dropdownlist=OptionMenu(framepatientregistrationright,PrDoctorname,\*list)

dropdownlist.pack()

labelRoomNo=Label(framepatientregistrationright,text="Room No.",bg="lightgreen",font=("calibri",17,"bold"))

labelRoomNo.pack(pady=10)

PrRoomNo=StringVar()

RoomNo=Entry(framepatientregistrationright,textvariable=PrRoomNo,width="20",font=("calibri",17,"bold"))

RoomNo.pack(padx=5)

ADDAbutton=Button(framepatientregistrationright,text="ADD",command=Pradd,bg="yellow",width=10,bd=7,font=("calibri",15,"bold"))

ADDAbutton.pack(side=LEFT,padx=5)

buttonprintreg=Button(framepatientregistrationright,text="Receipt",command=Receipt,bg="yellow",width=10,bd=7,font=("calibri",15,"bold"))

buttonprintreg.pack(side=LEFT,padx=5)

framemenuroot=Frame(root,height="190",bg="lightgreen")

framemenuroot.pack(side=TOP,fill=X)

framehospitalname=Label(framemenuroot,text=" CUH HOSPITAL",bg="lightgreen",fg="red",width="35",font=("calibri",27,"bold"))

framehospitalname.pack(anchor="center")

framemenuimage=Frame(root)

framemenuimage.pack(side=RIGHT)

imagemenu=PhotoImage(file="hospital.png",height="1200",width="1030")

label=ttk.Label(framemenuimage,image=imagemenu)

label.pack(side=RIGHT,fill=BOTH,expand=True)

framemenubackground=Frame(root,bg="lightgreen",height="60",width="50")

framemenubackground.pack(side=LEFT,fill=Y)

framemenuheading=Label(framemenubackground,text="Menu",fg="White",bg="sandybrown",width="25",font=("calibri",18,"bold"))

framemenuheading.pack(anchor="nw")

homeicon=PhotoImage(file=r"home.png")

Buttonpatienthome=Button(framemenubackground,text="Home ",compound=LEFT,image=homeicon,command=home,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

Buttonpatienthome.pack(side="top",fill=X)

patientregistrationicon=PhotoImage(file=r"patientregistration.png")

Buttonpatientregistration=Button(framemenubackground,text="Patient Registration ",compound=LEFT,image=patientregistrationicon,command=preg,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

Buttonpatientregistration.pack(side="top",fill=X)

patientadmiticon=PhotoImage(file=r"patientadmit.png")

Buttonpatientinformation=Button(framemenubackground,text="Patient Admit ",compound=LEFT,image=patientadmiticon,command=padmit,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

Buttonpatientinformation.pack(side="top",fill=X)

registrationinformationicon=PhotoImage(file=r"patientinformation.png")

Buttonregistrationinfo=Button(framemenubackground,text="Registration Information",compound=LEFT,image=registrationinformationicon,command=rinfo,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

Buttonregistrationinfo.pack(side="top",fill=X)

admitinformationicon=PhotoImage(file=r"patientadmit.png")

Buttonadmitinformation=Button(framemenubackground,text="Admit Information ",compound=LEFT,image=admitinformationicon,command=ainfo,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

Buttonadmitinformation.pack(side="top",fill=X)

doctorregistrationicon=PhotoImage(file=r"doctorregistration.png")

Buttonstaffregistration=Button(framemenubackground,text="Staff Registration ",compound=LEFT,image=doctorregistrationicon,command=sreg,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

Buttonstaffregistration.pack(side="top",fill=X)

doctorinformationicon=PhotoImage(file=r"doctorinformation.png")

Buttonstaffinformation=Button(framemenubackground,text="Staff Information ",compound=LEFT,image=doctorinformationicon,command=sinfo,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

Buttonstaffinformation.pack(side="top",fill=X)

registrationrecordicon=PhotoImage(file=r"registrationrecord.png")

ButtonRegistrationRecord=Button(framemenubackground,text="Registration Record ",compound=LEFT,image=registrationrecordicon,command=RRecord,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

ButtonRegistrationRecord.pack(side="top",fill=X)

admitrecordicon=PhotoImage(file=r"admitrecord.png")

ButtonAdmitRecord=Button(framemenubackground,text="Admit Record ",compound=LEFT,image=admitrecordicon,command=AdmitRecord,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

ButtonAdmitRecord.pack(side="top",fill=X)

doctorrecordicon=PhotoImage(file=r"doctorrecord.png")

ButtonStaffRecord=Button(framemenubackground,text="Staff Record ",compound=LEFT,image=doctorrecordicon,command=drecord,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

ButtonStaffRecord.pack(side="top",fill=X)

generatebillicon=PhotoImage(file=r"generatebill.png")

ButtonBill=Button(framemenubackground,text="Generate Bill ",compound=LEFT,image=generatebillicon,command=Bill,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

ButtonBill.pack(side="top",fill=X)

billrecordicon=PhotoImage(file=r"recordbill.png")

ButtonBill=Button(framemenubackground,text="Bill Record ",compound=LEFT,image=billrecordicon,command=billrecord,bd=5,bg="sandybrown",font=("calibri",14,"bold"))

ButtonBill.pack(side="top",fill=X)

appointmenticon=PhotoImage(file=r"appointment.png")

Buttonappointment=Button(framemenubackground,text="Appointment ",compound=LEFT,image=appointmenticon,bd=5,bg="sandybrown",font=("calibri",14,"bold"),command=appointment)

Buttonappointment.pack(side="top",fill=X)

else:

messagebox.showinfo("INVALID USERNAME OR PASSWORD ", "YOU HAVE ENTERED INVALID USERNAME OR PASSWORD ")

usernameVar.delete(0,END)

passwordVar.delete(0,END)

def exit():

if messagebox.askyesno("Confirm","Are you Sure,you want to exit"):

root.destroy()

root.protocol("WM\_DELETE\_WINDOW",exit)

frame1=Frame(root,height="500",width=500)

frame1.pack(side=TOP,fill=X)

C = Canvas(frame1,bg="blue", height=900,width=800)

filename = PhotoImage(file="hospital.png")

background\_label = Label(frame1, image=filename)

background\_label.place(x=0,y=0,relwidth=1, relheight=1)

userright=Frame(frame1,height="400",width="400",bg="pink",relief="solid",borderwidth=2)

userright.pack(side=RIGHT,padx="200")

usernullframe=Frame(userright,height="50",bg="pink")

usernullframe.pack(side=TOP)

image1=PhotoImage(file="ee.png")

label=Label(userright,image=image1,width=150,height=150,bg="pink")

label.pack(side=LEFT,padx=10)

usernameicon=PhotoImage(file=r"username.png")

labelUsername=Label(userright,text="Username:",compound=LEFT,image=usernameicon,font=("calibri",18,"bold"),bg="pink")

labelUsername.pack(anchor="nw",padx=10)

usernameVar=StringVar()

UsernameEntry=Entry(userright,font=("calibri",14,"bold"),textvariable=usernameVar)

UsernameEntry.pack(anchor="nw",padx=10)

passwordicon=PhotoImage(file=r"password.png")

labelPassword=Label(userright,text="Password:",compound=LEFT,image=passwordicon,font=("calibri",18,"bold"),bg="pink")

labelPassword.pack(anchor="nw",padx=10)

passwordVar=StringVar()

userEntry=Entry(userright)

Passwordentry=Entry(userright,show="\*",font=("calibri",14,"bold"),textvariable=passwordVar)

Passwordentry.pack(anchor="nw",padx=10)

buttonok=Button(userright,text="LOGIN",bg="orange",height=2,width=6,bd=3,font=("calibri",12,"bold"),command=login)

buttonok.pack(side=LEFT,padx="10",pady="10")

buttonexit=Button(userright,text="EXIT",bg="orange",height=2,width=6,bd=3,font=("calibri",12,"bold"),command=exit)

buttonexit.pack(side=RIGHT,padx="10",pady="10")

C.pack()

try:

connection = sqlite3.connect("HEMANT.db")

cursor = connection.cursor()

cursor.execute("CREATE TABLE IF NOT EXISTS ap (patientnoappointment TEXT NOT NULL PRIMARY KEY, patientnameappointment TEXT, genderappointment TEXT, ageappointment TEXT, phonenoappointment TEXT, addressappointment TEXT, appointmenttime TEXT)")

cursor.execute("SELECT \* FROM ap ORDER BY `patientnoappointment` ASC")

fetch = cursor.fetchall()

for data in fetch:

appointment\_table.insert('', 'end', values=(data))

cursor.close()

connection.close()

except:

pass

try:

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="CREATE TABLE bl (BlIdentity text,BlName text,GenderBl text,BlAge text,Blcheckin text,BlDisease text,BlPhone text,BlAddress text,BlDoctorname text,Blcheckout text,BlDoctorcharges text,BlLabtest text,TreatmentchargesBl text,BlProcedurecharges text,BlOthercharges text,BlMiscellaneouscharges text,BlRoomcharges text,BlTotal text)"

cursor.execute(query)

connection.commit()

except:

pass

try:

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="CREATE TABLE sr (SrID text,SrName text,GenderSr text,SrPosition text,SrDepartment text,SrMailid text,SrSalary text,SrPhone text,SrAddress text)"

cursor.execute(query)

connection.commit()

except:

pass

try:

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="CREATE TABLE pa (PaPatientid text,PaName text,GenderPa text,PaAge text,PaPhone text,PaAddress text,PaDisease text,PaCheckIn text,PaBloodgroup text,PaDoctorname text,PaCheckOut text,PaRoomNo text,PaRoomtype text,PaPrice text)"

cursor.execute(query)

connection.commit()

except:

pass

try:

connection=sqlite3.connect("HEMANT.db")

cursor=connection.cursor()

query="CREATE TABLE pr (PrPatientid text,PrName text,GenderPr text,PrAge text,PrPhone text,PrAddress text,PrDisease text,PrCheckIn text,PrBloodgroup text,PrDoctorname text,PrRoomNo text)"

cursor.execute(query)

connection.commit()

except:

pass

root.mainloop()