INTERDISCIPLINARY PROJECT REPORT
at
Sathyabama Institute of Science and Technology
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

THARUN KUMAR TIRUPATHI
(Reg. No. 40111352)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12 B Status
by UGC | Approved by AICTE
JEPPIAR NAGAR, RAJIV GANDHISALAI,
CHENNAI – 600119

APRIL 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **THARUN KUMAR TIRUPATHI (40111352)** who carried out the project entitled "**HealthCare Management System"** under my supervision from January 2023 to March 2023.

Internal Guide
Ms. K. Karunya M.E.

Head of the Department
Dr.L.Lakshmanan M.E., Ph.D.

Submitted for Viva-voce Examination held on ____________________

Internal Examiner **External Examiner**

DECLARATION

I **THARUN KUMAR TIRUPATHI** hereby declare that the project report entitled "**HealthCare Management"** done by me under the guidance of **Ms.K.Karunya M.E.,** is submitted in partial fulfillment of the requirements for the award of at Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE: **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

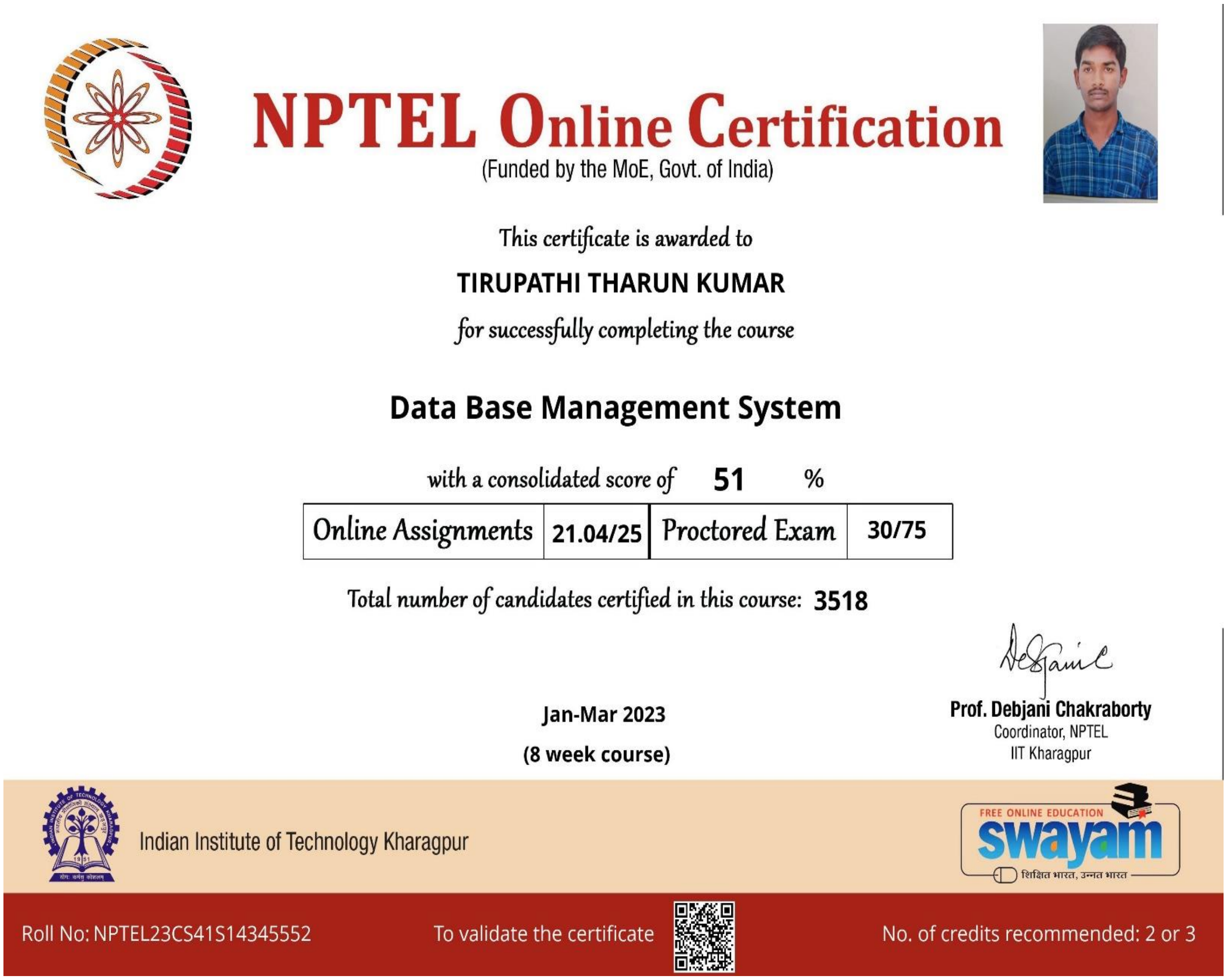
I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr.L.Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms.K.Karunya M.E,** for her valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

NPTEL CERTIFICATE



NPTEL Online Certification

(Funded by the MoE, Govt. of India)

This certificate is awarded to

TIRUPATHI THARUN KUMAR

for successfully completing the course

Data Base Management System

with a consolidated score of **51** %

Online Assignments	21.04/25	Proctored Exam	30/75

Total number of candidates certified in this course: **3518**

Jan-Mar 2023

(8 week course)

Prof. Debjani Chakraborty
Coordinator, NPTEL
IIT Kharagpur

Indian Institute of Technology Kharagpur

FREE ONLINE EDUCATION
swayam
शिक्षित भारत, उन्नत भारत

Roll No: NPTEL23CS41S14345552

To validate the certificate

No. of credits recommended: 2 or 3

ABSTRACT

Hospital Management System provides the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability. HMS is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. More importantly it is backed by reliable and dependable support. The project ‘Hospital Management System’ is based on the database and networking techniques. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses PYTHON as the front-end software and has connectivity with MY SQL Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to just fit in to your requirement. This package has been widely accepted by the clients in India and overseas. Not stopping only to this but they are highly satisfied and appreciating. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable. The package is highly customizable and can be modified as per the needs and requirements of our clients. Prolonged study of the functionalities of the hospital and its specific requirement has given it a wonderful shape both technically and usability wise. It covers all the required modules right from Patient Registration, Medicine details, Doctor, Wards, Admin, Store, Patient appointment, bill payment, record modification, discharge details etc.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Human Body is a very complex and sophisticated structure and comprises of millions of functions. All these complicated functions have been understood by man him, part-by-part their research and experiments. As science and technology progressed, medicine became an integral part of the research. Gradually, medical science became an entirely new branch of science. As of today, the Health Sector comprises of medical institutions i.e., Hospitals, HOSPITALs etc. research and development institutions and medical colleges. Thus, the Health sector aims at providing the best medical facilities to the common man.

1.1 PROBLEM STATEMENT

Since Hospital is associated with the lives of common people and their day-to-day routines so I decided to work on this project. The manual handling of the record is time consuming and highly prone to error. The purpose of this project is to automate or make online, the process of day-to-day activities like Room activities, Admission of New Patient, Discharge of Patient, assign a doctor, and finally compute the bill etc. I have tried my best to make the complicated process Hospital Management System as simple as possible using Structured & Modular technique & Menu oriented interface. I have tried to design the software in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. Even though I cannot claim that this work to be entirely exhaustive, the main purpose of my exercise is perform each Hospital’s activity in computerized way rather than manually which is time consuming. I am confident that this software package can be readily used by non-programming personal avoiding human handled chance of error.

1.2 OBJECTIVES

Hospital are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep the hospital running smoothly & successfully.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper.

Thus keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as "Administration support system for medical institutions".

The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and also provides robust & reliable storage and backup facilities.

CHAPTER 2
AIM AND SCOPE OF THE PROJECT

2.1 AIM OF THE PROJECT

To provide 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, Clinical. Financial.

2.2 SCOPE OF THE PROJECT

The proposed software product is the Hospital Management system (HMS). The system will be used in any hospital, clinic, dispensary or pathology labs. Clinic, dispensary or pathology to get the information from the patients and then storing that data for future usages. The current system in use is a paper-based system. It is too slow and cannot provide updated lists of patients within reasonable timeframe. The intention of the system is to reduce over-time pay and increase the number of patients that can be treated accurately. Requirement statements in these documents are both functional and non-functional.

1.Information about Patients is done by just writing the Patients name, age and gender. Whenever the Patient comes up his information is stored freshly.

2.Bills are generated by recording price for each facility provided to Patient on a separate sheet and at last they all are summed up.

3.Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.

4.Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.

5.Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can’t remember them at that time.

CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS USED

3.1 SOFTWARE SPECIFICATION

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

OPERATING SYSTEM:	Windows 7/ XP/8
LANGUAGE USED:	Python
SOFTWARE:	IDLE
DATABASE:	MYSQL

3.2 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

- OPERATING SYSTEM: Windows 8 or newer, 64-bit macOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 7+, and others.
- SYSTEM ARCHITECTURE: Windows- 64-bit x86, 32-bit x86; MacOS- 64-bit x86; Linux- 64-bit x86, 64-bit aarch64 (AWS Graviton2 / arm64), 64-bit Power8/Power9, s390x (Linux on IBM Z & Linux ONE).
- DISK SPACE: 5 GB to download and install.

3.3 METHODOLOGY

We have used Iterative and Incremental Development model (IID) for our project development. This development approach is also referred to as Iterative Waterfall Development approach. Iterative and Incremental Development is a software development process developed in response to the more traditional waterfall model. This model is designed to take care of such big project. The large and complicate project chiefly demand better development and testing procedure. The waterfall model is well known for its repeated testing process. Hence I choose the waterfall model for developing my software.

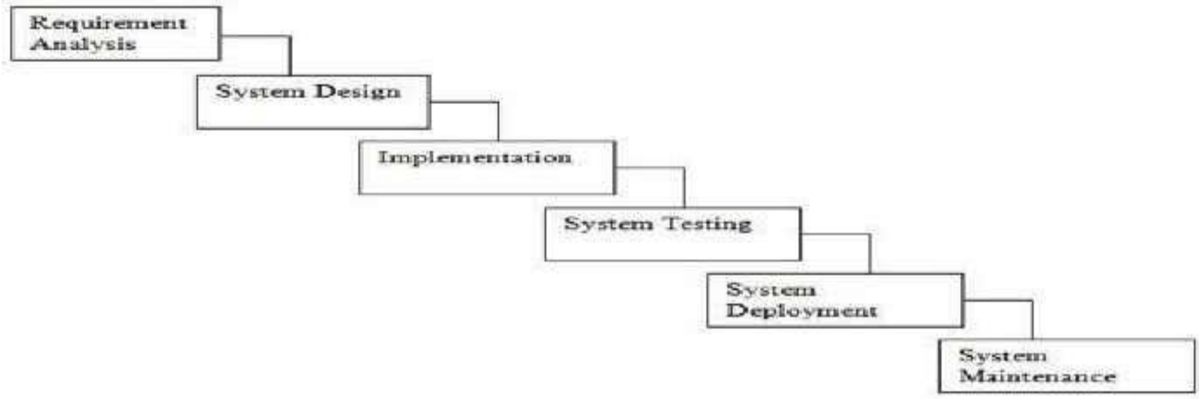


Fig:3.1 :Waterfall model

Some advantages of waterfall model:

- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.

3.3.1 Project Management life cycle

The Project Management Life Cycle has four phases. Each project life cycle phase is described along with the tasks need to complete it the four phases is

1.Initiation

2.Planning

3.Execution

4.Closure

3.4 MODULES

3.4.1 Patient Management

This module covers from the process of intake until discharge of an account of the patient’s engagement with the health-care team. Communication, empathy, examination, evaluation, diagnosis, prognosis, and intervention are all part of the process.

3.4.2 Doctor/Physician Management

The management of the physicians would be included in creating this system. Through this process, the admin will have the information and transactions made by the doctors with the patients.

3.4.3 Medicine and Prescription Management

This module will handle the process of monitoring a patient’s medications to verify that they are taken correctly and that the intended therapeutic outcome is achieved.

3.4.4 Online Appointment Management

This process is a tool that helps hospital admin manage their appointments. Internet booking is one of the tools available in an appointment management solution. Booking with a mobile app.

3.4.5 Medical and Transaction Management

Medical and transaction management modules aims to secure every transaction made by the patients and physicians in order to enhance healthcare quality and outcomes.

3.4.6 Payment and Expense Management

Payment and expense management module id meant to assist the admin in the payment management process. This will help the hospital with the full payment processing and accounts payable process.

These modules must be present in creating the Hospital Management to satisfy the needs in managing Hospital transactions. Through this, the management and monitoring of patients would be much easier for both hospital admin and physicians.

3.5 DATABASE DESIGN

Database design is the process of producing a detailed data model of database. This data model contains all the need logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different part of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structure used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structure, but also the forms and queries used as part of the overall database application within the database management system.

The Hospital management system database design is a database design used for managing hospital functions and events. It enables the admin to register a patient for the hospital, store their disease details into the database. Any of the staff members, doctor & admin can add, view, edit, update or delete data.

3.5.1 E-R Diagram of Hospital Management System

An entity-relationship diagram (ERD) is an abstract and conceptual representation of data. Entity-relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

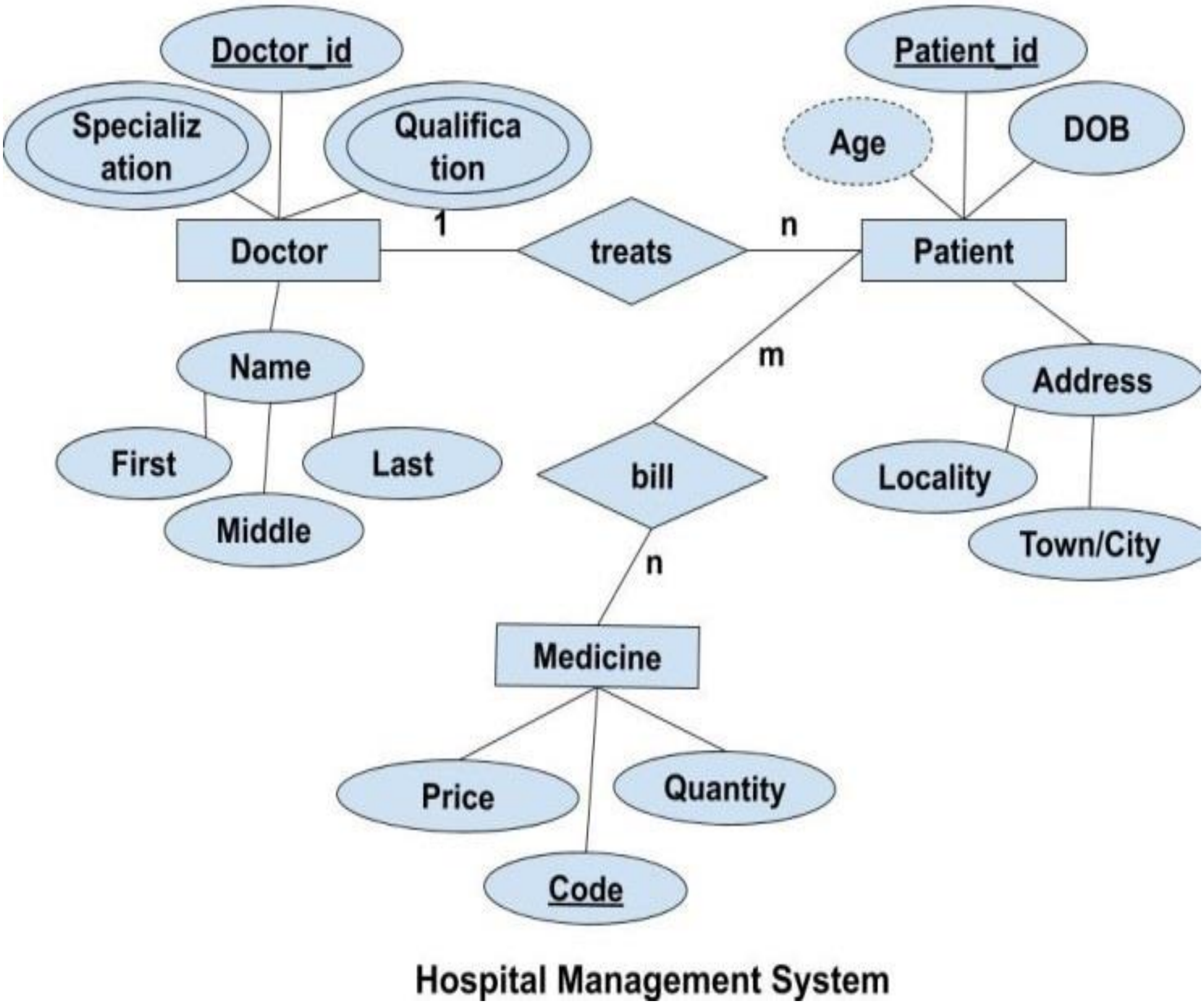


Fig:3.2 :E-R Diagram

3.6 LIBRARY USED

3.6.1 Tkinter

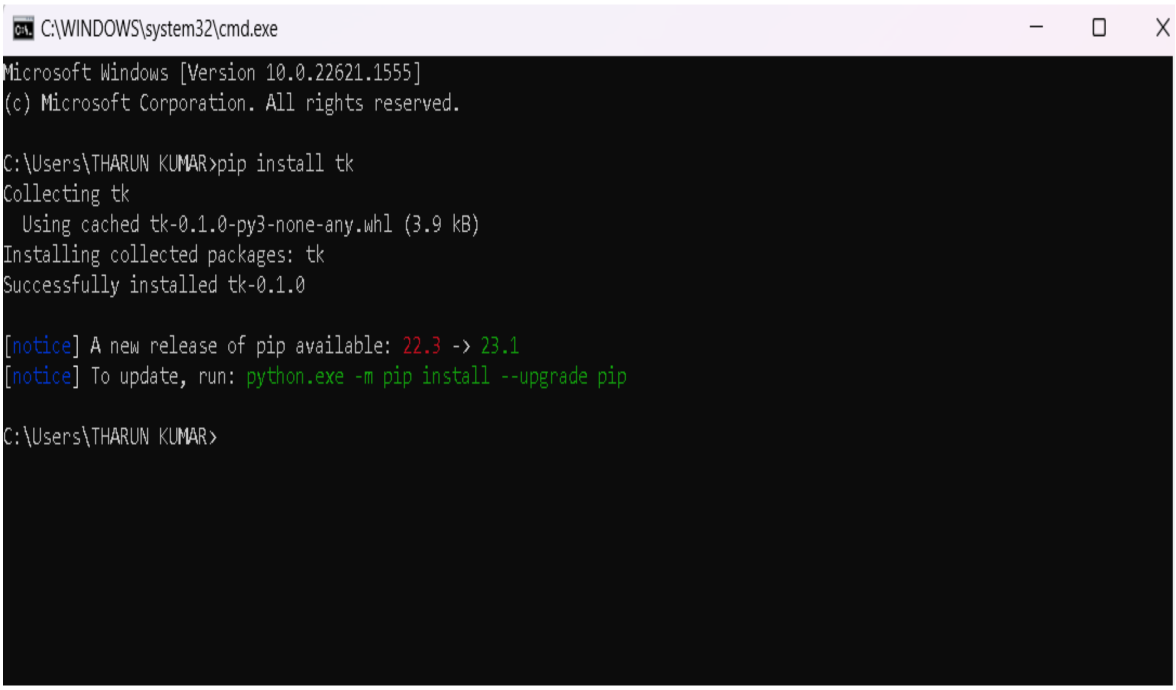
A Hospital Management System Project in Python is a basic Tkinter-based GUI-based desktop application that is user-friendly and simple to learn. This Hospital Management System In Python project just has an admin side, and the admin can only gain authorization. He or she can readily enter data into the system, as well as search.

Hospital administration enables the structured operation of healthcare and facilitates the supply of varied services. It allows large hospitals that provide a wide range of services to Improve financial tracking by better planning funds flow, improved investments, and expense control.

Install Tkinter

Tkinter can be installed using pip. The following command is run in the command prompt to install Tkinter.

pip install tk



```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\THARUN KUMAR>pip install tk
Collecting tk
  Using cached tk-0.1.0-py3-none-any.whl (3.9 kB)
Installing collected packages: tk
Successfully installed tk-0.1.0

[notice] A new release of pip available: 22.3 -> 23.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\THARUN KUMAR>
```

Fig:3.3 :Tkinter

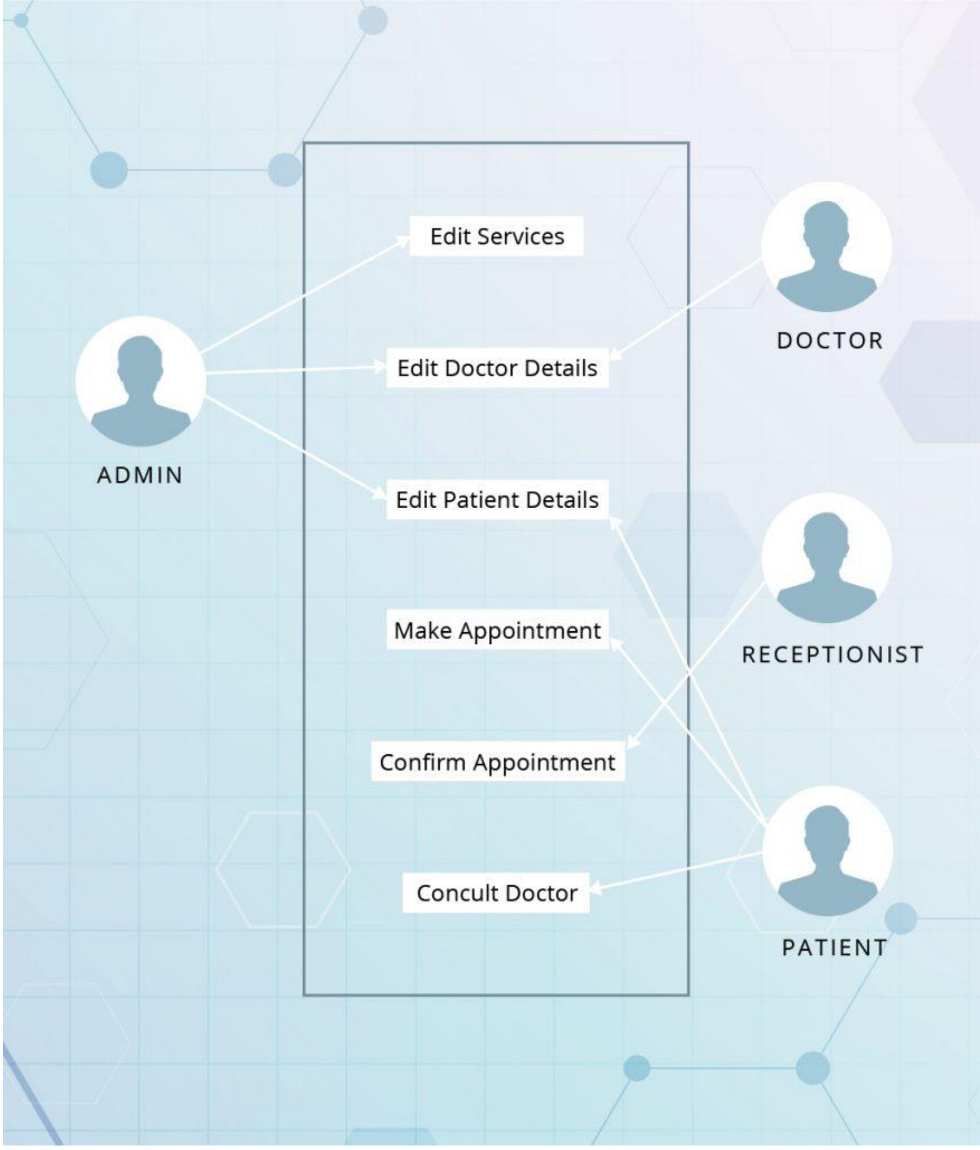


Fig:3.4 :Entity Relation Diagram

3.7 IMPLEMENTATION ENVIRONMENT

The implementation view of software requirement presents the real world manifestation of processing functions and information structures. This computerized system is specified in a manner that dictates accommodation of certain implementation details. The implementation environment of the developed system facilitates multiple users to use this system simultaneously. The user interfaces are designed keeping in mind that the users of this system are familiar to using GUI-based systems. Thus, we restricted ourselves to developing a GUI-based system so that it becomes easier for the end user to get acquainted to the developed system.

3.7.1 How to Implement HMS Successfully in Hospitals?

When compared to other industries, the health care sector is improving, with the help of advanced technologies like the Hospital Management System (HMS). The current medical care computerization is being hampered due to a lack of a perfect interface between computers and medical care providers.

The two main issues that are commonly faced during implementation of an HMS are as follows

1. HMS implementation begins but the users start facing issues by the end

This is generally due to the inflexibility of the HMS software, a weak administrative will and an inexperienced team for implementations.

2. HMS implementation gets completed, but only a single part of the functionality gets implemented.

This is due to non-user-friendly software or the unplanned and unconfigured executions.

3.7.2 Pre-Implementation Assessment

All hospital administrators who implement the HMS software in their hospitals will be apprehensive that the process may end in a partial or complete failure and measures should be taken early in order to prevent such an event from happening. The software should be well assessed before being purchased by the hospital.

3.7.3 Software Assessment

Choosing the right kind of software is one of the crucial and primary steps for getting the best results.

Earlier, all HMS Software had a single development cycle starting with a single hospital being computerized by a local developer for a single hospital. But this type of HMS development is outdated. In such cases, there is a risk of the vendor providing software that was customized for a different hospital. Implementing them may cause various issues, and it may cost more in the long run. It's better to

purchase the software from a firm, that specializes in hospital management software, than getting any software company to design software for your hospital.
It's better to choose a software that is easy to upgrade, scalable and is not hardware intensive. Cloud-based online software is better than offline software in these aspects. The providers could also easily fix any errors or bugs in case of online software.

Another aspect to be considered is to check if the customization of the software is done primarily through programming code or by using master tables during implementation. Hospitals may require some form of alteration to the purchased software, so as to better suit their needs, and in the case of master tables, such changes may easily be made by the hospital itself. In the case of online software, this may not matter much, as maintenance and support are provided by the software provider.

3.7.4 Vendor Assessment

Only a few vendors of international or national repute can be trusted easily. A drawback with a large vendor is that they do not always accommodate the requirements of small hospitals. In the case of small vendors, the hospital administrators may have to do thorough research to see if they are trustworthy.

An important aspect to be known is if the current implementation team will be available during the entire implementation cycle. Checking the attrition rate of the software company, and discussions with other hospital administrators may give you an idea about this.

Visits to Hospitals with Hospital Management Systems:
A well-planned visit to hospitals where the software has been implemented will give you insights on the performance of the software, the support provided by the vendor, and if the vendor has achieved all the goals that were initially planned. To get a good picture, it's better to visit hospitals like your own, in terms of capacity, number of specialties, and number of departments, as well as hospitals that have implemented

the software that you are considering. This will give you an idea, not only about the trustworthiness of the vendor but also on how well the software will be suited to your hospital.

Discussions with hospital administrators who have previously implemented the software in their hospitals will also give you good insights on the possible difficulties you may face during the implementation process, and on how to overcome them.

Getting your Hospital ready:

Problems may arise in HMS implementation, due to issues within the hospital. Lack of a systematic and uniform workflow within all the departments in the hospital may make the implementation process difficult. Information regarding the various equipment used, tests performed, schedules of doctors in different departments, etc must be readily available so that it could easily be imported to the system.

An important aspect is the cooperation of the various end-users, that is, the hospital employees. They have to provide enough awareness of the usefulness and importance of implementing the HMS software. The implementation may put additional stress on the employees until they get used to the system, and this may hamper the efforts of the hospital administration.

HMS implementation begins with a pre-implementation phase, in which software is chosen and the hospital prepares for the change, followed by the implementation phase and the post-implementation phase, in which the final kinks are ironed out. The whole process could be made smooth with a thorough pre-implementation phase.

Choosing software that is well suited to your hospital, having clearly defined workflows, and complete information about the working of the hospital will help you to easily implement the HMS in your hospital. Having a good team, and ensuring the cooperation of the hospital employees would help the hospital achieve its goals with the HMS.

3.8 REASON FOR THIS PROJECT

A hospital management system improves patient experiences by promoting communication between doctors and patients. The frequent interventions and connected care that are made possible after patients are discharged from the hospital to post-acute care lowers the likelihood of readmission. 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, Clinical. Financial.

3.8.1 Track Financials Better

Having a hospital management system in place will significantly benefit the Finance department of the hospital. They can track revenues, outflows, debts and receivables better with an HMS. The financial reports that an HMS can generate would give the hospital administration a very accurate picture of the health of a hospital.

3.8.2 Secure Your Data

HMS makes it impossible for unauthorized personnel from accessing sensitive and private information, especially concerning patient records. It considerably reduces the possibility of data theft.

3.8.3 Eradicate Errors

The older, manual method of hospital management is highly prone to human error. With automation of processes using an HMS, there is little or no room for error in any of the hospital workflows**.**

3.8.4 Make Diagnosis and Treatment Easier

With the useful dashboards and reports provided by an HMS, doctors and other caregivers have easy and quick access to patient records and test results, thus increasing clinical competency all around.

3.8.5 Increase Patient Satisfaction

HMS considerably reduces the time taken for processes and procedures at every stage of patient interaction, such as registration, billing, and discharge, thus making your customers happy.

3.8.6 Improve Hospital Quality Ratings

Implementation of an HMS provided by Healthcare IT solution providers has become a vital index of the quality of a hospital or healthcare facility. Thus, HMS significantly improves your standing with insurance companies and quality circles.

3.8.7 Paperless Operation

A happy outcome of the advent of HMS is the increasing use of electronic records over physical files. There might come a day when hospitals become completely paperless with further advances in hospital management systems.

3.8.8 Run Your Hospital Efficiently

The amount of time and effort you will save by setting up an HMS for your hospital can be astounding. These invaluable savings make for the efficient and smooth operation of your hospital.

Now that you know the incredible hospital management services offered by the full-scale implementation of an HMS, do not spend any more time thinking! Do your research, consult with a reputable HMS vendor and make your hospital ready for the future!

CHAPTER 4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

The hospital management system organizes the stable functioning of daily tasks and interactions. This is a special tool to support the smooth operating of the software components that are vital for the clinic administration. The hospital records management software keeps a track of all the operations, stores the users' data, performs its analysis and generates the reports. The medical institution is given the opportunity to collect its information in one place. It includes the patient and doctors' records as well as the data concerning financial affairs, supply management, etc. Furthermore, it is only processed, classified and accessible for authorized users. The hospital database management system provides users with data security due to all regulations. Implementation of different functions empowers smooth and clear functionality.

The hospital records management software tracks the number of available doctors and their working hours. This allows to have the accurate schedule of each employee, manage your facility abilities and the supply chain in order to meet all the needs of the patients. It helps to arrange the appointments for both the staff and patients' convenience.

Any clinic should store medical histories, test results, prescribed treatments, etc. The good hospital database management system will do it for you. All the details are securely stored for the access of the doctor and can be provided to the patients by their requests. They can receive the test results or medical reports by email or the user account. When the written form is required, printing will take only a few minutes for the clinic staff.

Another function is connected with managing finances. The hospital accounting software estimates the patients' payments. It might remind the bank account where you can check performed operations and the billing status of each customer.

HealthCare Management System reduces human intervention for paperwork, less paperwork, reduced staff headcount for jobs that can be easily managed within the HMS, speedier processes, reduction of errors, and data privacy and safety.

HMIS (Hospital Management Information System) is a medical informatics solution element that mainly focuses on hospital administration requirements. The HMS is a web-based or computer application that takes care of the complete hospital functionalities. The integrated system can be customized and are developed to control all hospital operations like patient details, appointment booking, billing, drug management, Electronic Medical Record, administration, Patient medical history, inventory management, bed management, revenue management, and so on.

Hospital Management System is essential and mandatory for healthcare establishments like nursing homes, rehabilitation centres, clinics, hospitals, health clinics, dispensaries, and more. Some of the top benefits of implementing an HMS are role-based access control, data accuracy, revenue management, appointment booking, overall cost reduction, and data security.

CHAPTER 5

SUMMARY AND CONCLUSIONS

This project has been a rewarding experience in more than one way. The entire project work has enlightened us in the following areas. We have gained an insight into the working of the HOSPITAL. This represents a typical real world situation Our understanding of database design has been strengthened this is because in order to generate the final reports of database designing has to be properly followed. Scheduling a project and adhering to that schedule creates a strong sense of time management. Sense of teamwork has developed and confidence of handling real life project has increased to a great extent. Initially, there were problem with the validation but with discussions, we were to implement validations.

Since we are entering details of the patients electronically in the" Hospital Management System", data will be secured. Using this application we can retrieve patient's history with a single click. Thus processing information will be faster. It guarantees accurate maintenance of Patient details. It easily reduces the book keeping task and thus reduces the human effort and increases accuracy speed.

We can make the conclusion that the hospital management system is the inevitable part of the lifecycle of the modern medical institution. It automates numerous daily operations and enables smooth interactions of the users. Developing the hospital system software is a great opportunity to create the distinct, efficient and fast delivering healthcare model. Implementation of hospital management system project helps to store all the kinds of records, provide coordination and user communication, implement policies, improve day-to-day operations, arrange the supply chain, manage financial and human resources, and market hospital services. This beneficial decision covers the needs of the patients, staff and hospital authorities and simplifies their interactions. It has become the usual approach to manage the hospital. Many clinics have already experienced its advantages and continue developing new hospital management system project modules.

REFERENCES

(1) https://itsourcecode.com/fyp/hospital-management-system-project-report-documentations-pdf/

(2) https://existek.com/blog/hospital-managment-system/

(3) https://www.academia.edu/7149341/HOSPITAL_MANAGEMENT_SYSTEM_A_PROJECT_REPORT_Submitted_in_Partial_Fulfillment_of_the_requirements_for_the_Award_of_the

(4) https://mocdoc.in/blog/a-detailed-view-of-hospital-management-system-hms

(5) https://www.slideshare.net/OvercomerMichael/design-and-implementation-of-a-hospital-management-system

(6) R.S Pressman, Software Engineering: A Practitioner’s Approach, Mc-Graw-Hill, Edition-7 (2010).

(7) Python resources - https://www.python.org/

(8) Tkinter resources in python - https://docs.python.org/3/library/tkinter.html

(9) https://www.geeksforgeeks.org/create-mysql-database-login-page-in-python-using-tkinter/

(10) https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/

APPENDIX

A. SCREENSHOTS

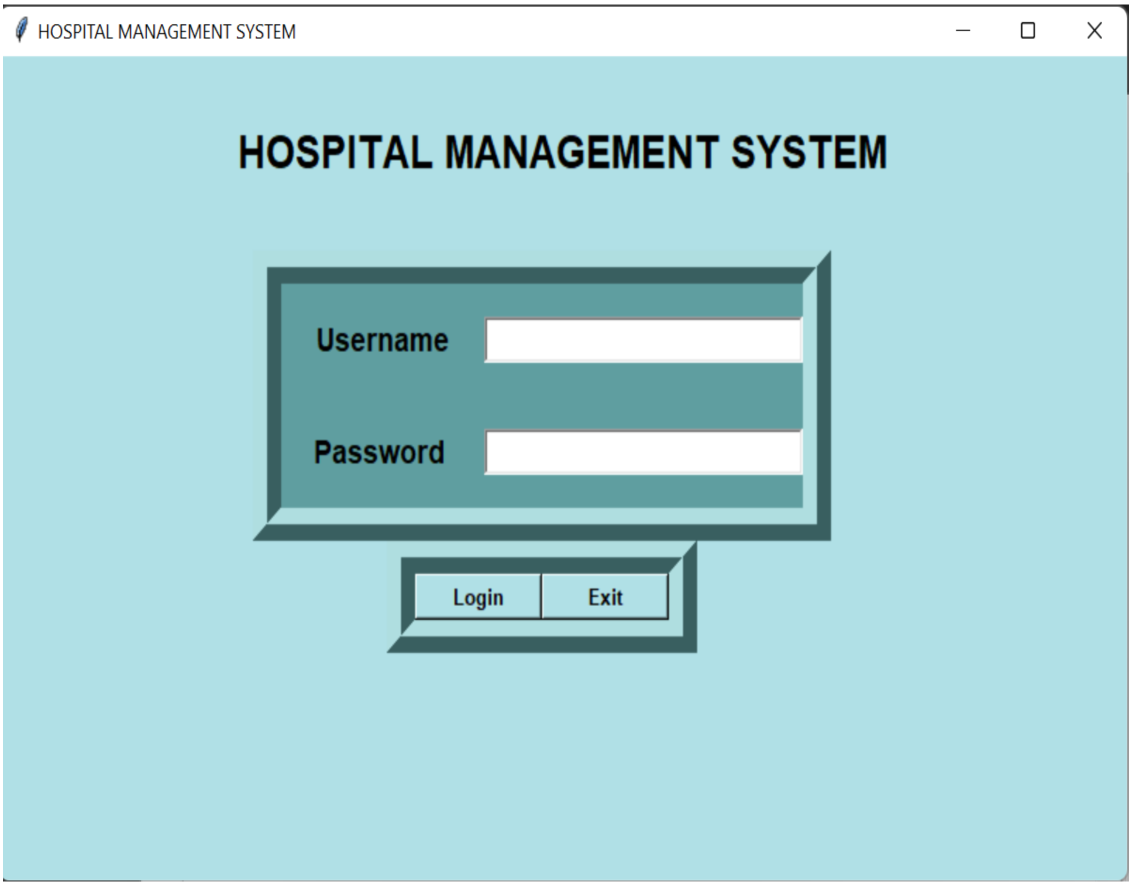


Fig:5.1 :Admin Login

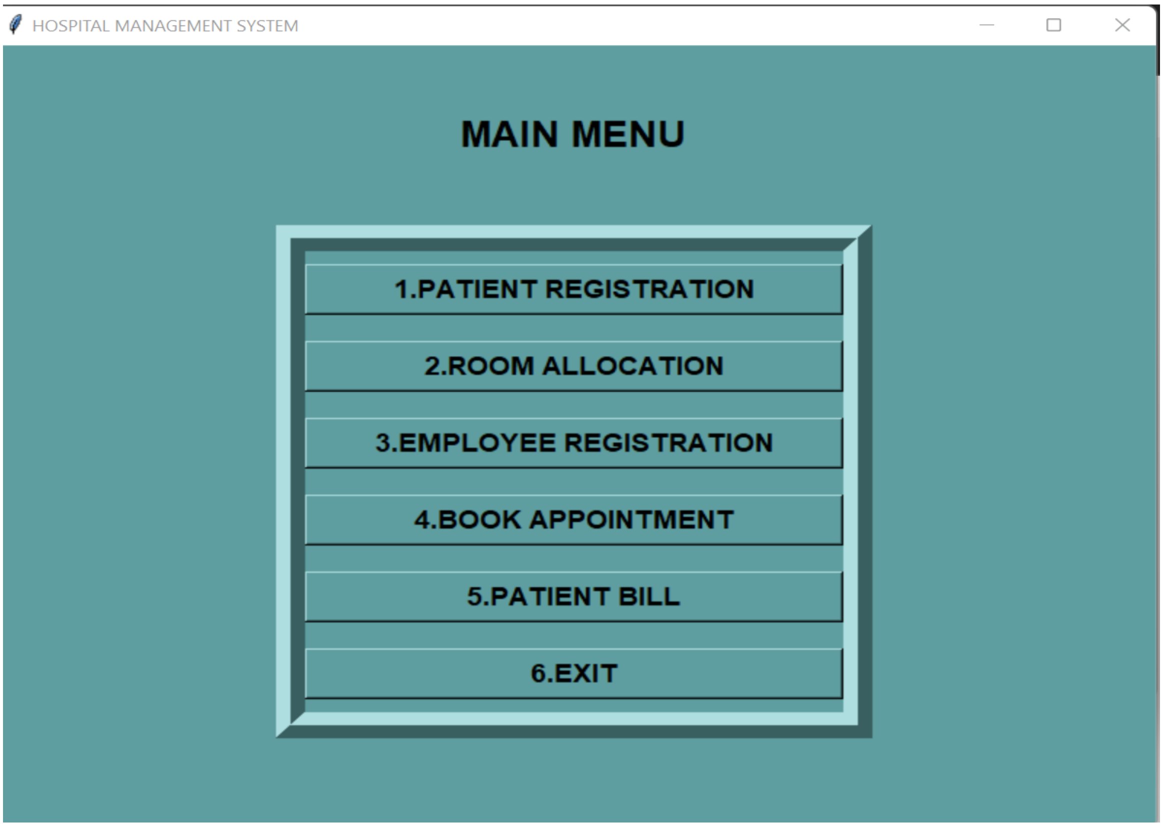
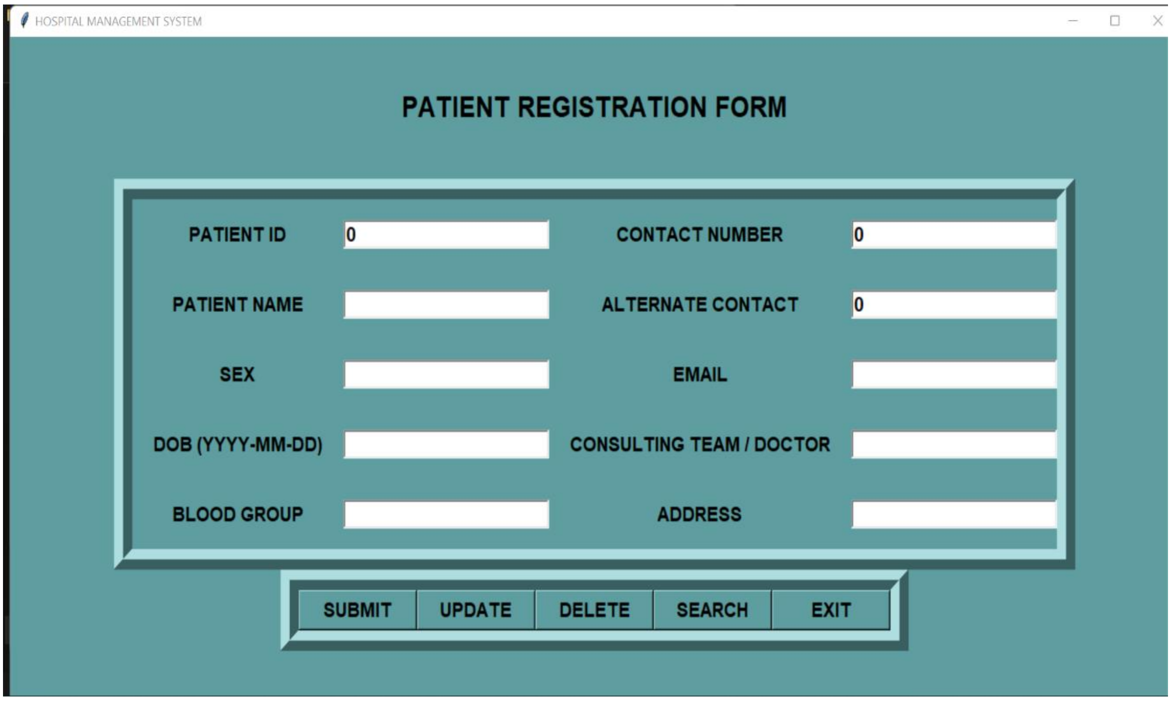


Fig:5.2 :Menu page



PATIENT ID 0

CONTACT NUMBER 0

PATIENT NAME

ALTERNATE CONTACT 0

SEX

EMAIL

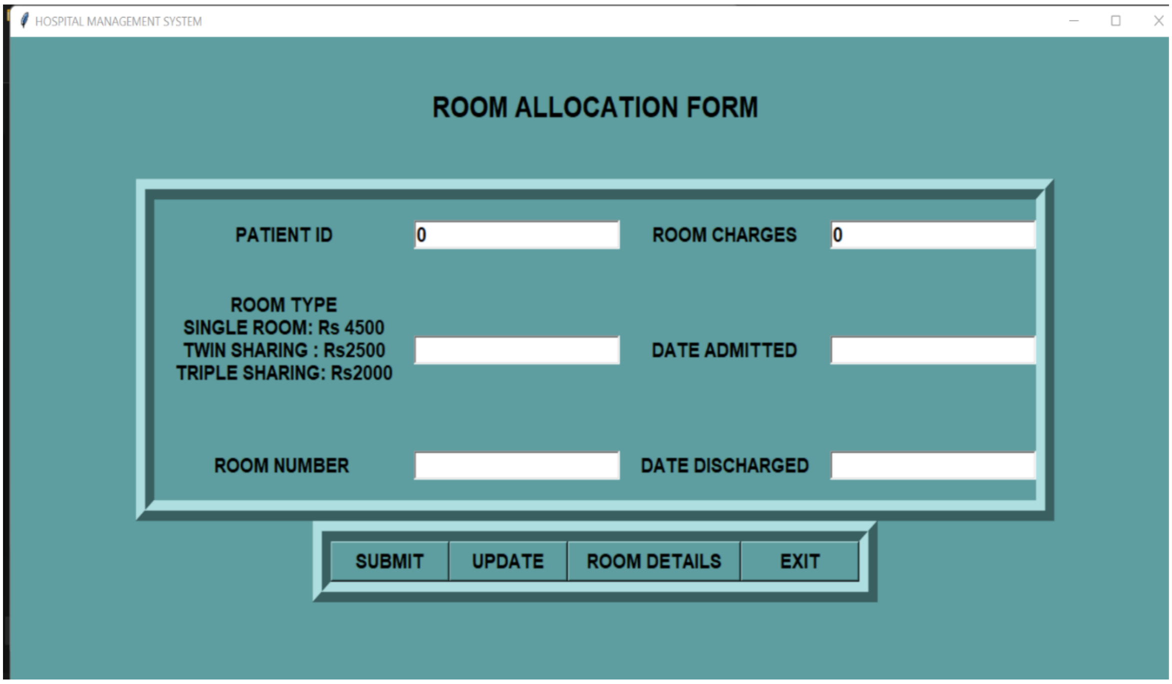
DOB (YYYY-MM-DD)

CONSULTING TEAM / DOCTOR

BLOOD GROUP

ADDRESS

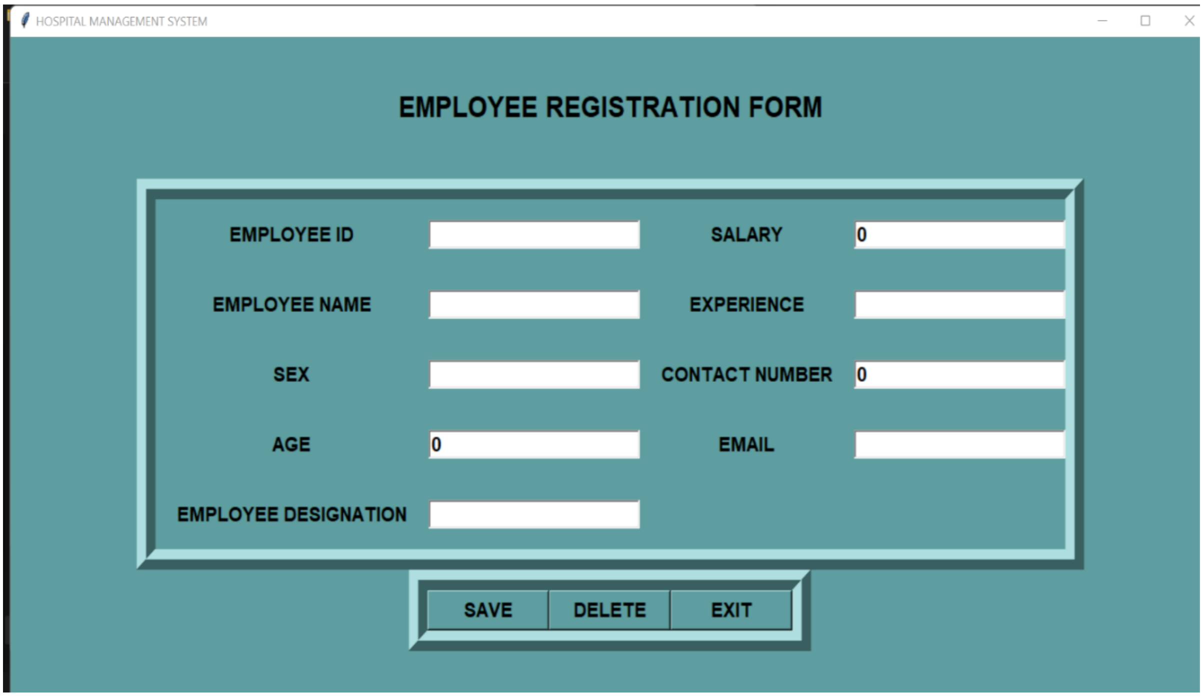
Fig:5.3 :Patient Registration Form



ROOM ALLOCATION FORM

PATIENT ID	0	ROOM CHARGES	0
ROOM TYPE SINGLE ROOM: Rs 4500 TWIN SHARING : Rs2500 TRIPLE SHARING: Rs2000		DATE ADMITTED	
ROOM NUMBER		DATE DISCHARGED	

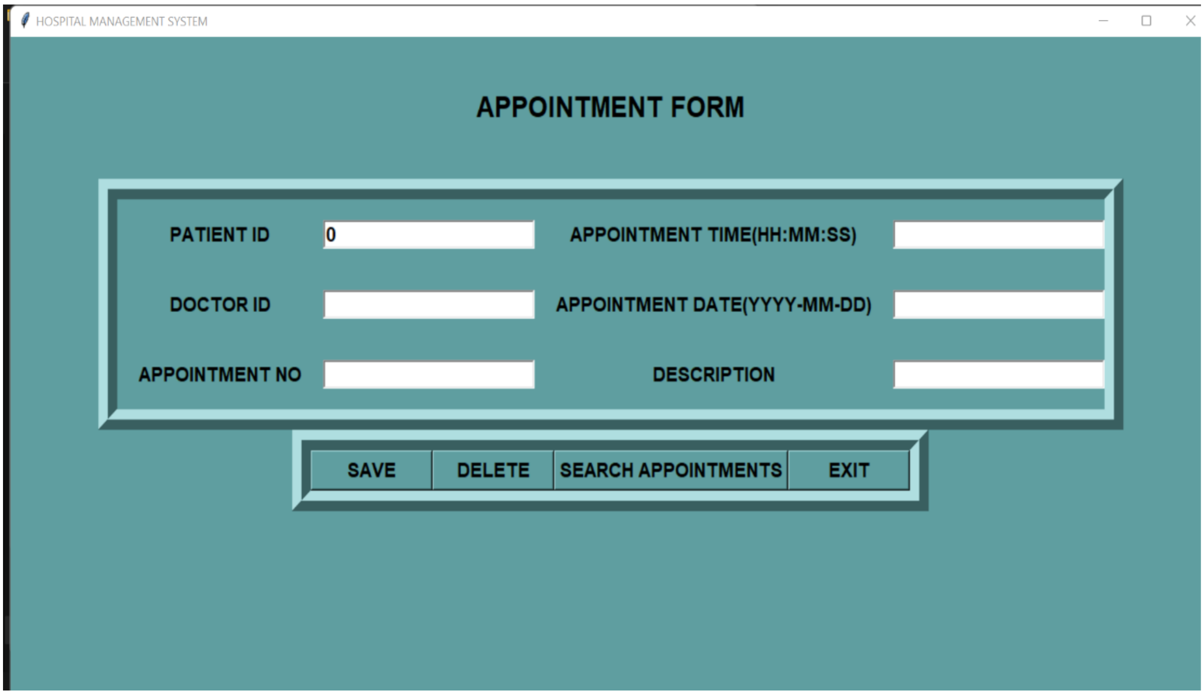
Fig:5.4 :Room Allocation



EMPLOYEE REGISTRATION FORM

EMPLOYEE ID		SALARY	0
EMPLOYEE NAME		EXPERIENCE	
SEX		CONTACT NUMBER	0
AGE	0	EMAIL	
EMPLOYEE DESIGNATION			

Fig:5.5 :Employee Registration Form

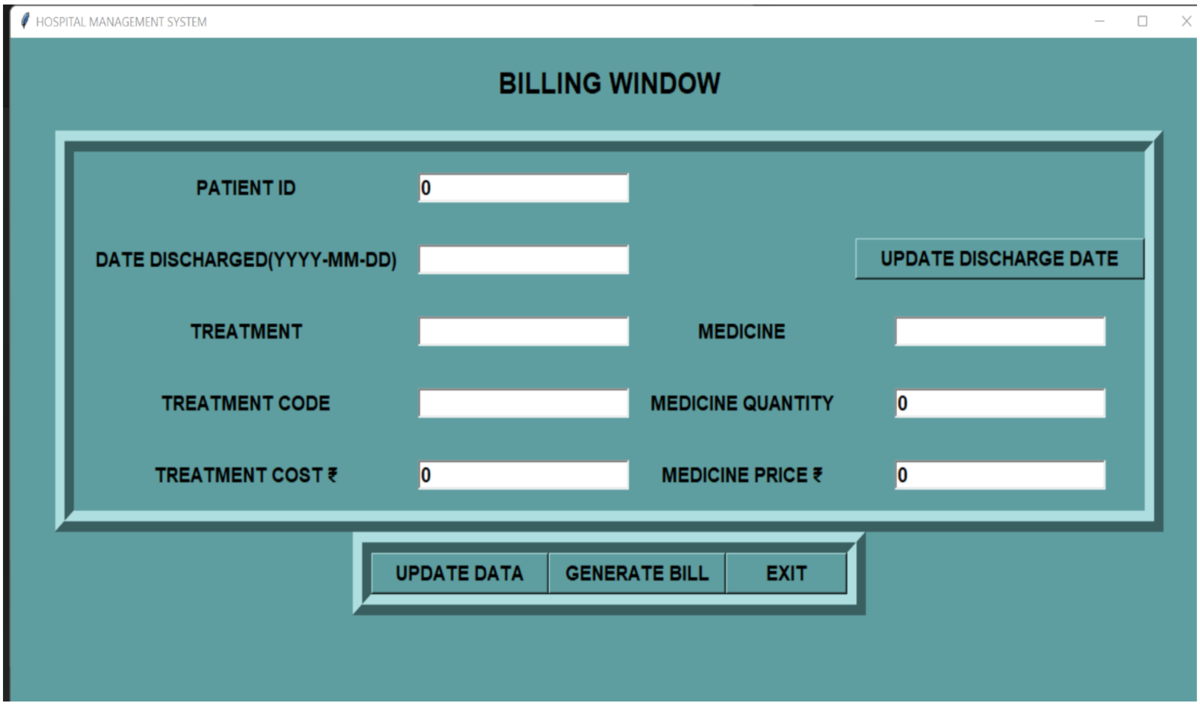


HOSPITAL MANAGEMENT SYSTEM

APPOINTMENT FORM

PATIENT ID	0	APPOINTMENT TIME(HH:MM:SS)	
DOCTOR ID		APPOINTMENT DATE(YYYY-MM-DD)	
APPOINTMENT NO		DESCRIPTION	

Fig:5.6 :Appointment Form



HOSPITAL MANAGEMENT SYSTEM

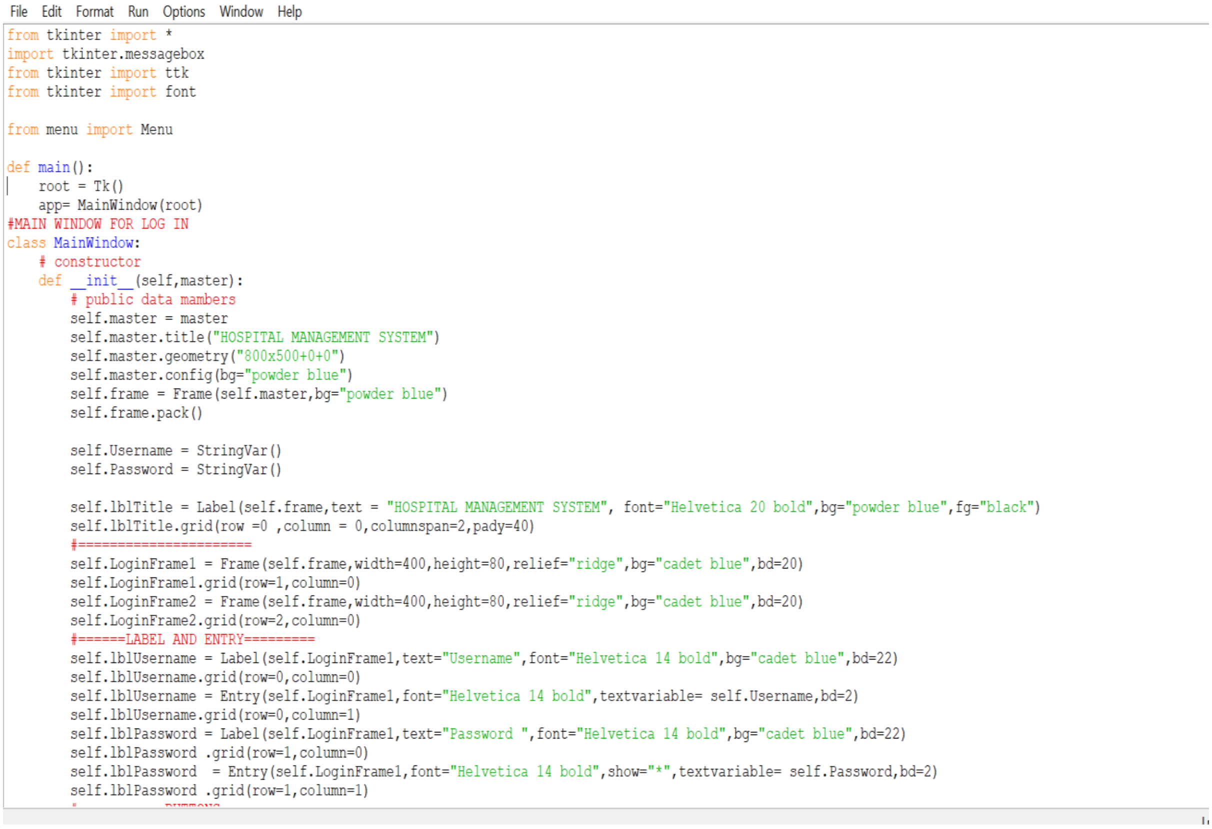
BILLING WINDOW

PATIENT ID	0		
DATE DISCHARGED(YYYY-MM-DD)			
TREATMENT		MEDICINE	
TREATMENT CODE		MEDICINE QUANTITY	0
TREATMENT COST ₹	0	MEDICINE PRICE ₹	0

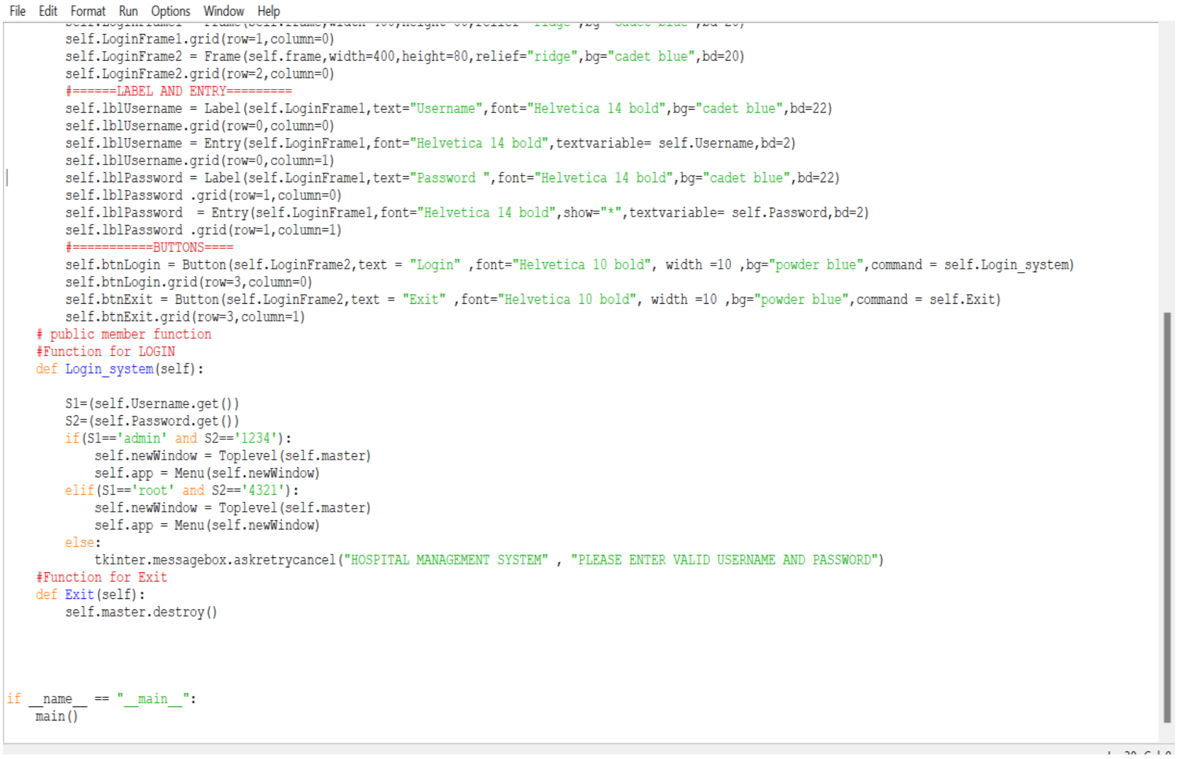
Fig:5.7 :Billing Form

B.SAMPLE CODE

Login page code

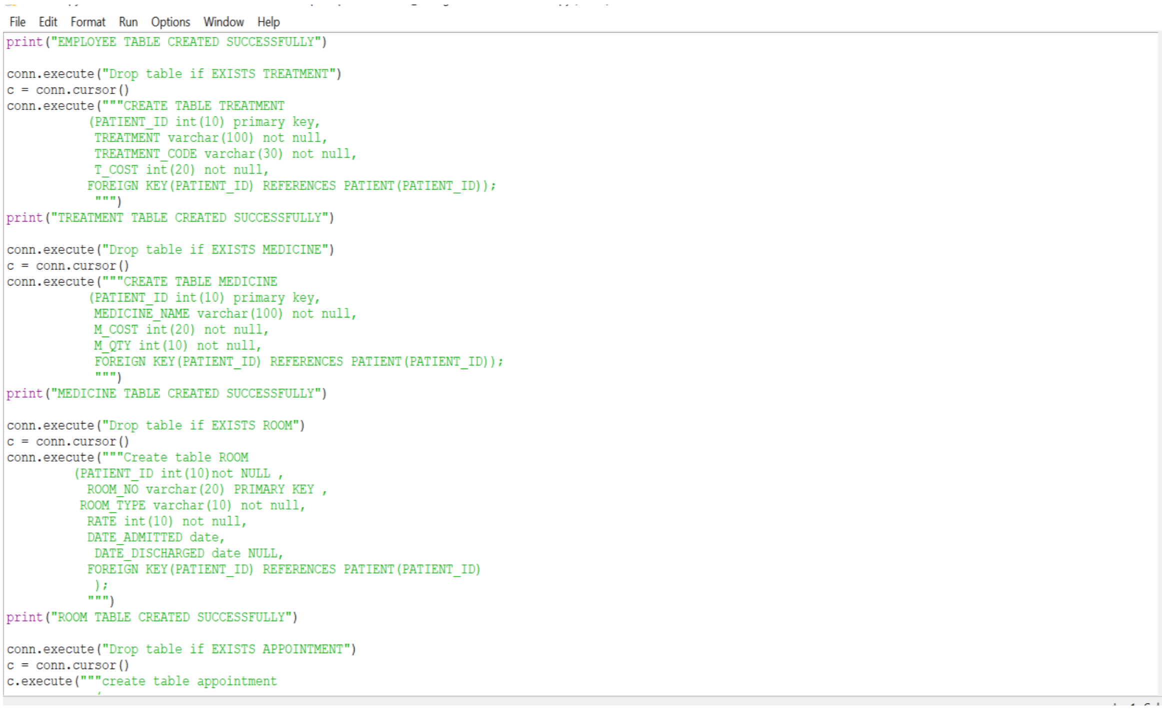


```
from tkinter import *
import tkinter.messagebox
from tkinter import ttk
from tkinter import font

from menu import Menu

def main():
    root = Tk()
    app= MainWindow(root)
#MAIN WINDOW FOR LOG IN
class MainWindow:
    # constructor
    def __init__(self,master):
        # public data mambers
        self.master = master
        self.master.title("HOSPITAL MANAGEMENT SYSTEM")
        self.master.geometry("800x500+0+0")
        self.master.config(bg="powder blue")
        self.frame = Frame(self.master,bg="powder blue")
        self.frame.pack()

        self.Username = StringVar()
        self.Password = StringVar()

        self.lblTitle = Label(self.frame,text = "HOSPITAL MANAGEMENT SYSTEM", font="Helvetica 20 bold",bg="powder blue",fg="black")
        self.lblTitle.grid(row =0 ,column = 0,columnspan=2,pady=40)
        #======================
        self.LoginFrame1 = Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet blue",bd=20)
        self.LoginFrame1.grid(row=1,column=0)
        self.LoginFrame2 = Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet blue",bd=20)
        self.LoginFrame2.grid(row=2,column=0)
        #======LABEL AND ENTRY=========
        self.lblUsername = Label(self.LoginFrame1,text="Username",font="Helvetica 14 bold",bg="cadet blue",bd=22)
        self.lblUsername.grid(row=0,column=0)
        self.lblUsername = Entry(self.LoginFrame1,font="Helvetica 14 bold",textvariable= self.Username,bd=2)
        self.lblUsername.grid(row=0,column=1)
        self.lblPassword = Label(self.LoginFrame1,text="Password ",font="Helvetica 14 bold",bg="cadet blue",bd=22)
        self.lblPassword .grid(row=1,column=0)
        self.lblPassword  = Entry(self.LoginFrame1,font="Helvetica 14 bold",show="*",textvariable= self.Password,bd=2)
        self.lblPassword .grid(row=1,column=1)
```

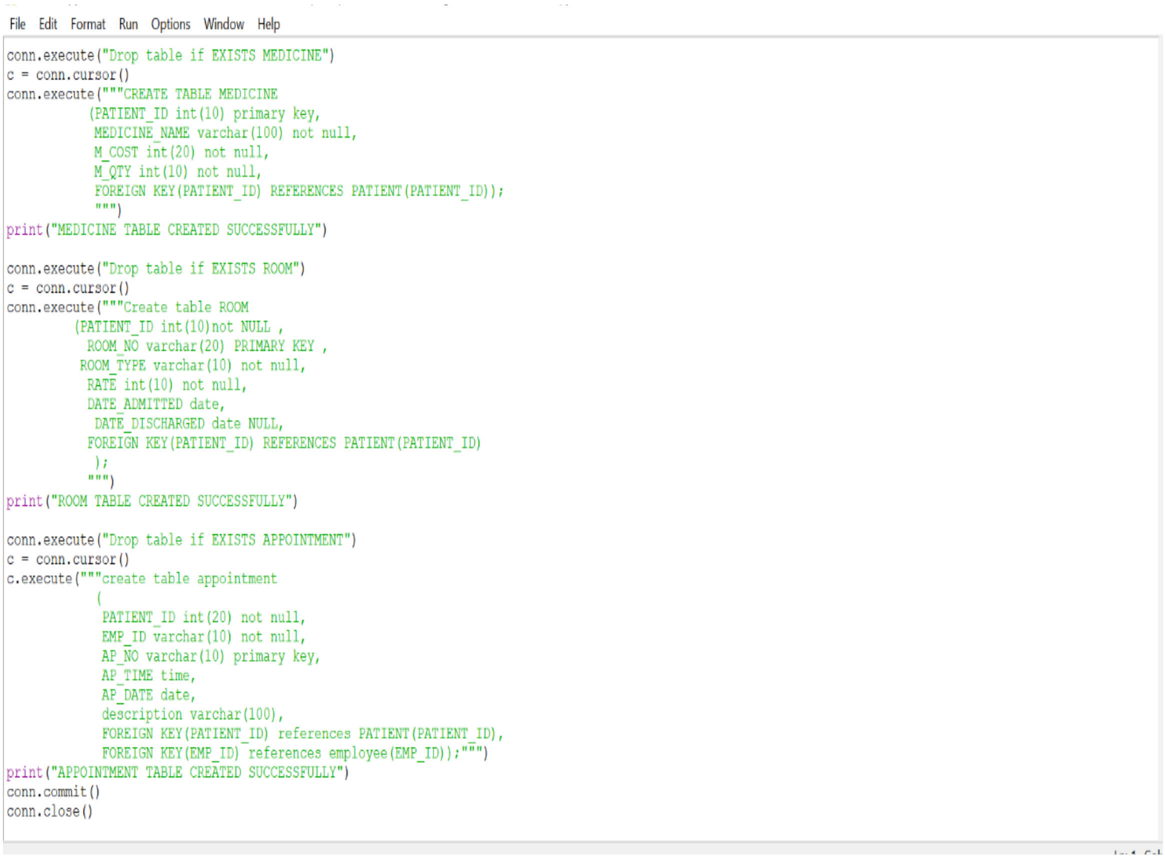


```
        self.LoginFrame1.grid(row=1,column=0)
        self.LoginFrame2 = Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet blue",bd=20)
        self.LoginFrame2.grid(row=2,column=0)
        #======LABEL AND ENTRY=========
        self.lblUsername = Label(self.LoginFrame1,text="Username",font="Helvetica 14 bold",bg="cadet blue",bd=22)
        self.lblUsername.grid(row=0,column=0)
        self.lblUsername = Entry(self.LoginFrame1,font="Helvetica 14 bold",textvariable= self.Username,bd=2)
        self.lblUsername.grid(row=0,column=1)
        self.lblPassword = Label(self.LoginFrame1,text="Password ",font="Helvetica 14 bold",bg="cadet blue",bd=22)
        self.lblPassword .grid(row=1,column=0)
        self.lblPassword  = Entry(self.LoginFrame1,font="Helvetica 14 bold",show="*",textvariable= self.Password,bd=2)
        self.lblPassword .grid(row=1,column=1)
        #===========BUTTONS====
        self.btnLogin = Button(self.LoginFrame2,text = "Login" ,font="Helvetica 10 bold", width =10 ,bg="powder blue",command = self.Login_system)
        self.btnLogin.grid(row=3,column=0)
        self.btnExit = Button(self.LoginFrame2,text = "Exit" ,font="Helvetica 10 bold", width =10 ,bg="powder blue",command = self.Exit)
        self.btnExit.grid(row=3,column=1)
    # public member function
    #Function for LOGIN
    def Login_system(self):

        S1=(self.Username.get())
        S2=(self.Password.get())
        if(S1=='admin' and S2=='1234'):
            self.newWindow = Toplevel(self.master)
            self.app = Menu(self.newWindow)
        elif(S1=='root' and S2=='4321'):
            self.newWindow = Toplevel(self.master)
            self.app = Menu(self.newWindow)
        else:
            tkinter.messagebox.askretrycancel("HOSPITAL MANAGEMENT SYSTEM" , "PLEASE ENTER VALID USERNAME AND PASSWORD")
    #Function for Exit
    def Exit(self):
        self.master.destroy()



if __name__ == "__main__":
    main()
```

Databasecode

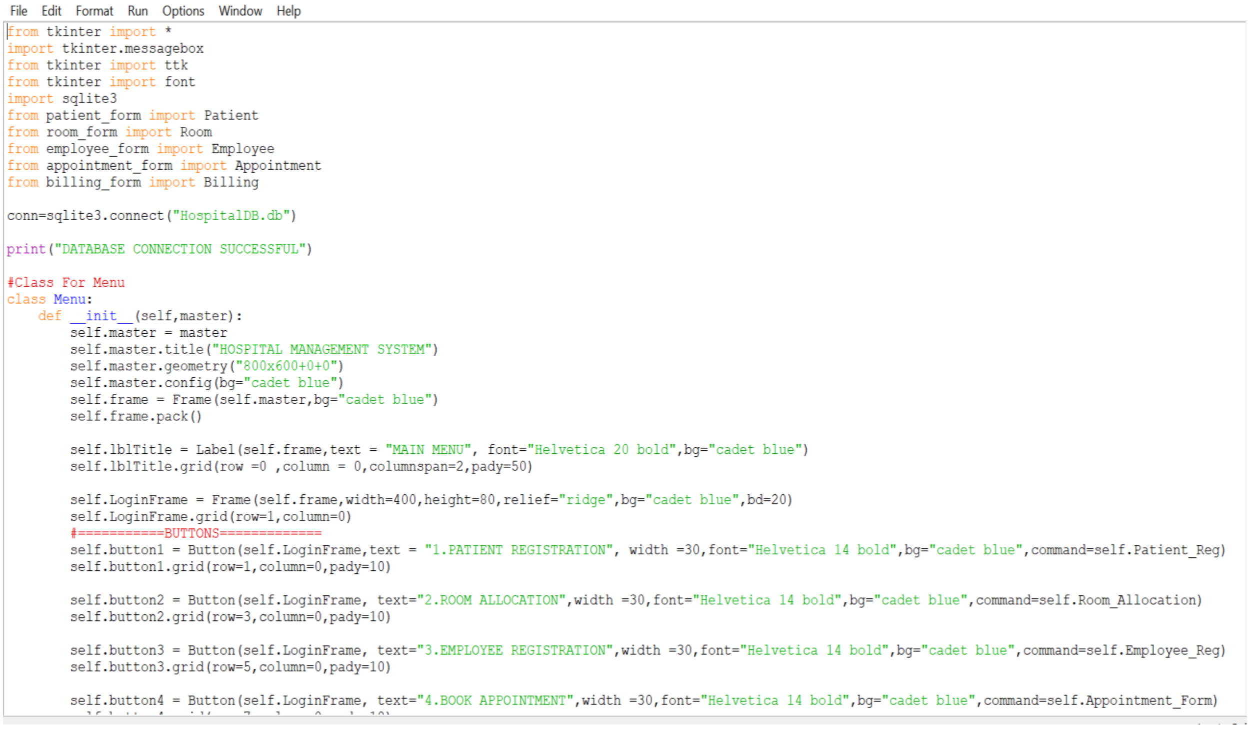


```
print("EMPLOYEE TABLE CREATED SUCCESSFULLY")

conn.execute("Drop table if EXISTS TREATMENT")
c = conn.cursor()
conn.execute("""CREATE TABLE TREATMENT
           (PATIENT_ID int(10) primary key,
            TREATMENT varchar(100) not null,
            TREATMENT_CODE varchar(30) not null,
            T_COST int(20) not null,
           FOREIGN KEY(PATIENT_ID) REFERENCES PATIENT(PATIENT_ID));
            """)
print("TREATMENT TABLE CREATED SUCCESSFULLY")

conn.execute("Drop table if EXISTS MEDICINE")
c = conn.cursor()
conn.execute("""CREATE TABLE MEDICINE
           (PATIENT_ID int(10) primary key,
            MEDICINE_NAME varchar(100) not null,
            M_COST int(20) not null,
            M_QTY int(10) not null,
            FOREIGN KEY(PATIENT_ID) REFERENCES PATIENT(PATIENT_ID));
            """)
print("MEDICINE TABLE CREATED SUCCESSFULLY")

conn.execute("Drop table if EXISTS ROOM")
c = conn.cursor()
conn.execute("""Create table ROOM
         (PATIENT_ID int(10)not NULL ,
           ROOM_NO varchar(20) PRIMARY KEY ,
          ROOM_TYPE varchar(10) not null,
           RATE int(10) not null,
           DATE_ADMITTED date,
            DATE_DISCHARGED date NULL,
           FOREIGN KEY(PATIENT_ID) REFERENCES PATIENT(PATIENT_ID)
            );
           """)
print("ROOM TABLE CREATED SUCCESSFULLY")

conn.execute("Drop table if EXISTS APPOINTMENT")
c = conn.cursor()
c.execute("""create table appointment
```

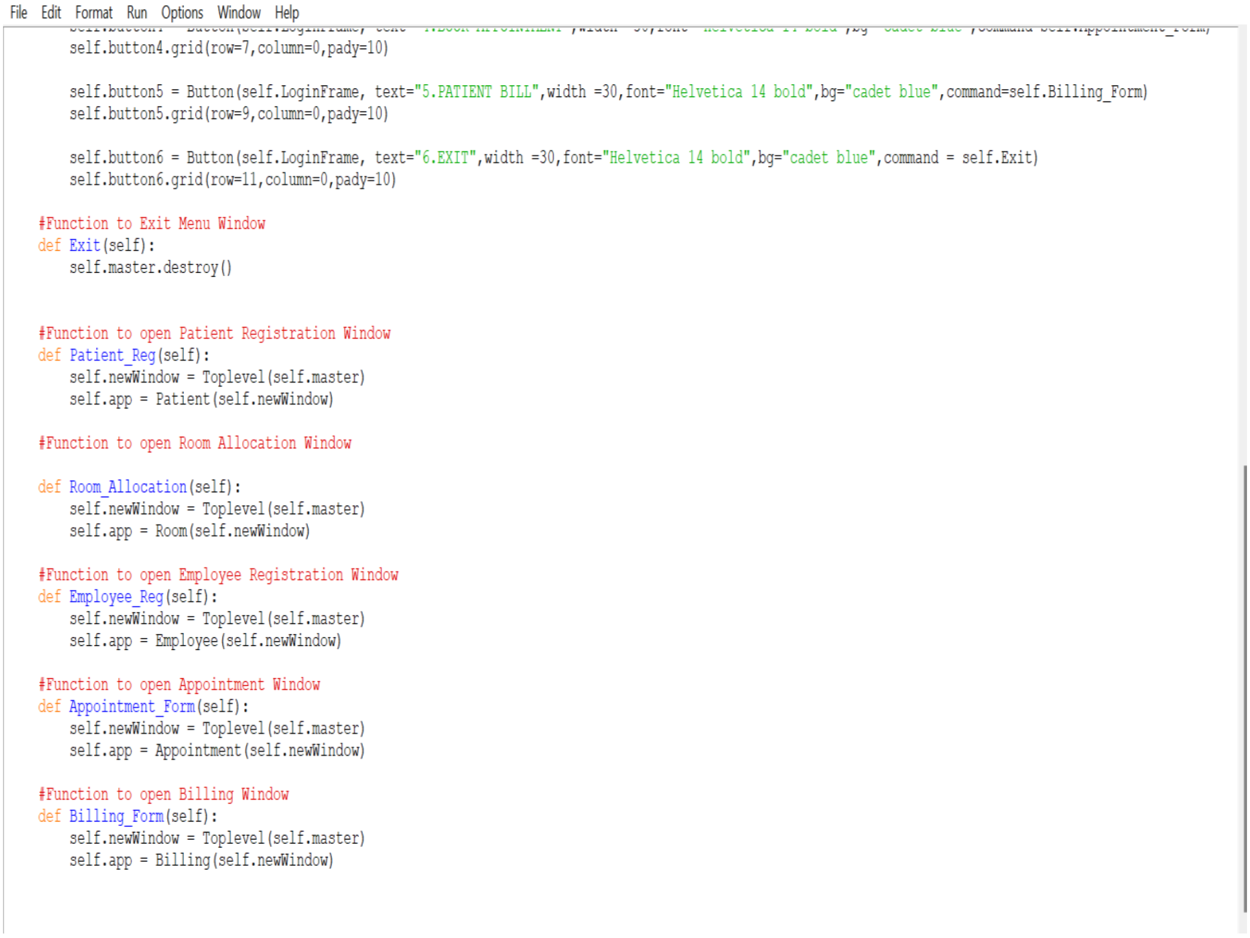


```
conn.execute("Drop table if EXISTS MEDICINE")
c = conn.cursor()
conn.execute("""CREATE TABLE MEDICINE
           (PATIENT_ID int(10) primary key,
            MEDICINE_NAME varchar(100) not null,
            M_COST int(20) not null,
            M_QTY int(10) not null,
            FOREIGN KEY(PATIENT_ID) REFERENCES PATIENT(PATIENT_ID));
            """)
print("MEDICINE TABLE CREATED SUCCESSFULLY")

conn.execute("Drop table if EXISTS ROOM")
c = conn.cursor()
conn.execute("""Create table ROOM
         (PATIENT_ID int(10)not NULL ,
           ROOM_NO varchar(20) PRIMARY KEY ,
          ROOM_TYPE varchar(10) not null,
           RATE int(10) not null,
           DATE_ADMITTED date,
            DATE_DISCHARGED date NULL,
           FOREIGN KEY(PATIENT_ID) REFERENCES PATIENT(PATIENT_ID)
            );
           """)
print("ROOM TABLE CREATED SUCCESSFULLY")

conn.execute("Drop table if EXISTS APPOINTMENT")
c = conn.cursor()
c.execute("""create table appointment
            (
             PATIENT_ID int(20) not null,
             EMP_ID varchar(10) not null,
             AP_NO varchar(10) primary key,
             AP_TIME time,
             AP_DATE date,
             description varchar(100),
             FOREIGN KEY(PATIENT_ID) references PATIENT(PATIENT_ID),
             FOREIGN KEY(EMP_ID) references employee(EMP_ID));""")
print("APPOINTMENT TABLE CREATED SUCCESSFULLY")
conn.commit()
conn.close()
```

Menu code



```python
from tkinter import *
import tkinter.messagebox
from tkinter import ttk
from tkinter import font
import sqlite3
from patient_form import Patient
from room_form import Room
from employee_form import Employee
from appointment_form import Appointment
from billing_form import Billing

conn=sqlite3.connect("HospitalDB.db")

print("DATABASE CONNECTION SUCCESSFUL")

#Class For Menu
class Menu:
    def __init__(self,master):
        self.master = master
        self.master.title("HOSPITAL MANAGEMENT SYSTEM")
        self.master.geometry("800x600+0+0")
        self.master.config(bg="cadet blue")
        self.frame = Frame(self.master,bg="cadet blue")
        self.frame.pack()

        self.lblTitle = Label(self.frame,text = "MAIN MENU", font="Helvetica 20 bold",bg="cadet blue")
        self.lblTitle.grid(row =0 ,column = 0,columnspan=2,pady=50)

        self.LoginFrame = Frame(self.frame,width=400,height=80,relief="ridge",bg="cadet blue",bd=20)
        self.LoginFrame.grid(row=1,column=0)
        #===========BUTTONS=============
        self.button1 = Button(self.LoginFrame,text = "1.PATIENT REGISTRATION", width =30,font="Helvetica 14 bold",bg="cadet blue",command=self.Patient_Reg)
        self.button1.grid(row=1,column=0,pady=10)

        self.button2 = Button(self.LoginFrame, text="2.ROOM ALLOCATION",width =30,font="Helvetica 14 bold",bg="cadet blue",command=self.Room_Allocation)
        self.button2.grid(row=3,column=0,pady=10)

        self.button3 = Button(self.LoginFrame, text="3.EMPLOYEE REGISTRATION",width =30,font="Helvetica 14 bold",bg="cadet blue",command=self.Employee_Reg)
        self.button3.grid(row=5,column=0,pady=10)

        self.button4 = Button(self.LoginFrame, text="4.BOOK APPOINTMENT",width =30,font="Helvetica 14 bold",bg="cadet blue",command=self.Appointment_Form)
```



```python
        self.button4.grid(row=7,column=0,pady=10)

        self.button5 = Button(self.LoginFrame, text="5.PATIENT BILL",width =30,font="Helvetica 14 bold",bg="cadet blue",command=self.Billing_Form)
        self.button5.grid(row=9,column=0,pady=10)

        self.button6 = Button(self.LoginFrame, text="6.EXIT",width =30,font="Helvetica 14 bold",bg="cadet blue",command = self.Exit)
        self.button6.grid(row=11,column=0,pady=10)

    #Function to Exit Menu Window
    def Exit(self):
        self.master.destroy()


    #Function to open Patient Registration Window
    def Patient_Reg(self):
        self.newWindow = Toplevel(self.master)
        self.app = Patient(self.newWindow)

    #Function to open Room Allocation Window

    def Room_Allocation(self):
        self.newWindow = Toplevel(self.master)
        self.app = Room(self.newWindow)

    #Function to open Employee Registration Window
    def Employee_Reg(self):
        self.newWindow = Toplevel(self.master)
        self.app = Employee(self.newWindow)

    #Function to open Appointment Window
    def Appointment_Form(self):
        self.newWindow = Toplevel(self.master)
        self.app = Appointment(self.newWindow)

    #Function to open Billing Window
    def Billing_Form(self):
        self.newWindow = Toplevel(self.master)
        self.app = Billing(self.newWindow)
```