HOSPITAL DATABASE MANAGEMENT SYSTEM (HDMS)

INDEX

ABSTRACT

The purpose of the project entitled, “HOSPITAL DATABASE MANAGEMENT SYSTEM” is to computerize the Front Office Management of the Hospital to develop software which is user friendly, simple, fast and cost effective. It deals with the information of Patients, Doctors and Staff. Traditionally, it was done manually. The main function of the system is to register and store the details and retrieve these details as and when required. It is also used to update and delete the existing data stored in the database. In short, it performs CRUD operations. The Hospital Database Management System can be accessed using a username and password. It is accessible either by an administrator or a receptionist.

OBJECTIVES OF THE PROJECT

The objective of this project is to let the students apply the programming knowledge into a real- world situation/problem and expose the students how programming skills help in developing good software.

- Write programs utilizing modern software tools.
- Apply object-oriented programming principles effectively when developing small to medium sized projects.
- Write effective procedural code to solve small to medium sized problems.
- Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory, and software development.
- Students will demonstrate the ability to conduct research or applied Informatics Practices Project, requiring writing and presentation skills which exemplify scholarly style in computer science.

WHAT IS A DATABASE MANAGEMENT SOFTWARE (DBMS)?

Database Management Software (DBMS) is used for storing, manipulating, and managing data, such as format, names of fields, and record and file structures in a database. Users can construct their own databases using a DBMS to satisfy their business requirements. For example, dBase was one of the first DBMS for micro-computers.

To interact with a database, a DBMS package generally uses SQL queries. It receives a command from a database administrator (DBA) and prompts the system to perform the necessary action. These instructions can be about loading, retrieving, or modifying existing data in the system.

MAIN FEATURES OF A DBMS

Some of the significant features of database management software include:

❖ LOW REPETITION AND REDUNDANCY

In a database, the chances of data duplication are quite high as several users use one database. A DBMS reduces data repetition and redundancy by creating a single data repository that can be accessed by multiple users, even allowing easy data mapping while performing ETL.

❖ EASY MAINTENANCE OF LARGE DATABASES

Most organizational data is stored in large databases. A DBMS helps maintain these databases by enforcing user-defined validation and integrity constraints, such as user-based access.

❖ ENHANCED SECURITY

When handling large amounts of data, security becomes the top-most concern for all businesses. Database management software doesn't allow full access to anyone except the database administrator or the departmental head. Only they can modify the database and control user access, making the database more secure. All other users are restricted, depending on their access level.

❖ IMPROVED FILE CONSISTENCY

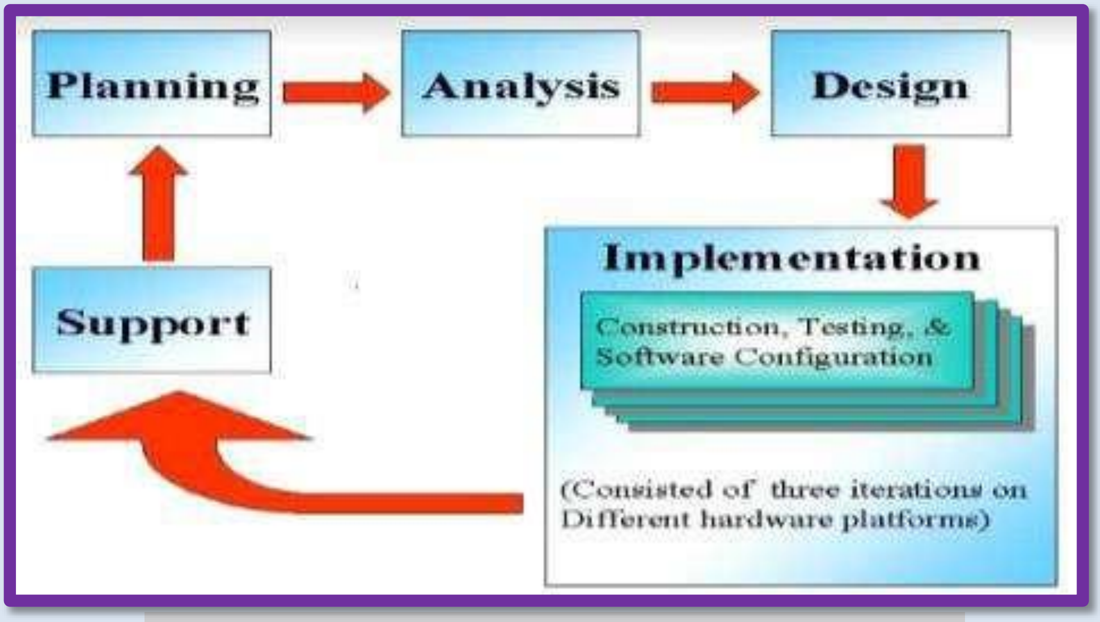
By implementing a database management system, organizations can create a standardized way to use files and ensure consistency of data with other systems and applications. This streamlines data management and manipulation because the same rules can be applied to all the data throughout the organization.

APPLICATIONS OF DATABASE MANAGEMENT SOFTWARE

Following are some of the applications of DBMS:

1. **BANKS:** Storing client information, account activities, disbursements, credits, and mortgages.
2. **AIRLINES:** Flight bookings and scheduling information.
3. **ACADEMIES:** Learner information, course registrations, grading, and result.
4. **TELECOMMUNICATION:** Keeping call archives, monthly bills, and retaining balances.
5. **ECONOMICS AND FINANCE:** Storing data about bonds, transactions, and acquisitions of fiscal instruments, such as shares and stocks.
6. **SALES AND MARKETING:** Storing data about consumers, merchandises, and sales.
7. **ENGINEERING AND MANUFACTURING:** Managing supply chain and pursuing manufacturing of items and inventory statuses in storerooms.
8. **HUMAN RESOURCES**: Keeping records about workers, remunerations, payroll and generating salaries.

SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC):


Planning
Analysis
Design
Implementation
Construction, Testing, & Software Configuration
(Consisted of three iterations on Different hardware platforms)
Support

The Systems Development Life Cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting projects allows managers to verify the successful completion of project phases before allocating resources to subsequent phases. Software development projects typically include initiation, planning, design, development, testing, implementation, and maintenance phases. However, the phases may be divided differently depending on the organization involved. End users of the system under development should be involved in reviewing the output of each phase to ensure the system is being built to deliver the needed functionality.

PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE

❖ INITIATION PHASE

The Initiation Phase begins when a business sponsor identifies a need or an opportunity.

THE PURPOSE OF THE INITIATION PHASE IS TO:

- Identify and validate an opportunity to improve business accomplishments of the organization or a deficiency related to a business need.
- Identify significant assumptions and constraints on solutions to that need.
- Recommend the exploration of alternative concepts and methods to satisfy the need including questioning the need for technology.
- Assure executive business and executive technical sponsorship.
- Infrastructure and the Strategic Plan.

“Careful oversight is required to ensure projects support strategic business objectives and resources are effectively implemented into an organizations enterprise architecture.”

❖ PLANNING PHASE

The planning phase is the most critical step in completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, is necessary to coordinate activities and manage project risks effectively. The depth and formality of project plans should be commensurated with the characteristics and risks of a given project. Project plans refine the information gathered during the initiation phase by further identifying the specific activities and resources required to complete a project.

A critical part of a project manager’s job is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and network requirements as possible. During this phase, a plan is developed that documents they approach to be used and includes a discussion of methods, tools, tasks, resources, project schedules, and user input. Personnel assignments, costs, project schedule, and target dates are established.

“A Project Management Plan is created with components related to acquisition planning, configuration management planning, quality assurance planning, concept of operations, system security, verification and validation, and systems engineering management planning.”

❖ REQUIREMENTS ANALYSIS PHASE

This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation, System Concept, and Planning phases. It also delineates the requirements in terms of data, system performance, security, and maintainability requirements for the system. The requirements are defined in this phase to a level of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the business need or opportunity identified in the Initiation Phase. The requirements that will be used to determine acceptance of the system are captured in the Test and Evaluation Master Plan.

THE PURPOSE OF THE REQUIREMENTS ANALYSIS PHASE IS TO:

- Develop detailed data and process models (system inputs, outputs, and the process.
- Develop the test and evaluation requirements that will be used to determine acceptable system performance.
- Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated, who generates it, where does the information go, and who processes it.)

❖ DESIGN PHASE

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase.

THE PURPOSE OF THE DESIGN PHASE IS TO:

- Identify potential risks and defining mitigating design features.
- Perform a security risk assessment.
- Develop a conversion plan to migrate current data to the new system.
- Determine the operating environment.
- Define major subsystems and their inputs and outputs.
- Prepare detailed logic specifications for each software module.

The result is a draft System Design Document which captures the preliminary design for the system.

❖ DEVELOPMENT PHASE

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program. Effective completion of the previous stages is a key factor in the success of the Development phase.

THE DEVELOPMENT PHASE CONSISTS OF:

- Translating the detailed requirements and design into system components.
- Testing individual elements (units) for usability.
- Preparing for integration and testing of the IT system.

❖ INTEGRATION AND TEST PHASE

Subsystem integration, system, security, and user acceptance testing are conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system.

MULTIPLE LEVELS OF TESTING ARE PERFORMED, INCLUDING:

- Testing at the development facility by the contractor and possibly supported by end users.
- Testing as a deployed system with end users working together with contract personnel.
- Operational testing by the end user alone performing all functions.

"Requirements are traced throughout testing, final Independent Verification, Validation evaluation is performed and all documentation is reviewed and accepted prior to acceptance of the system."

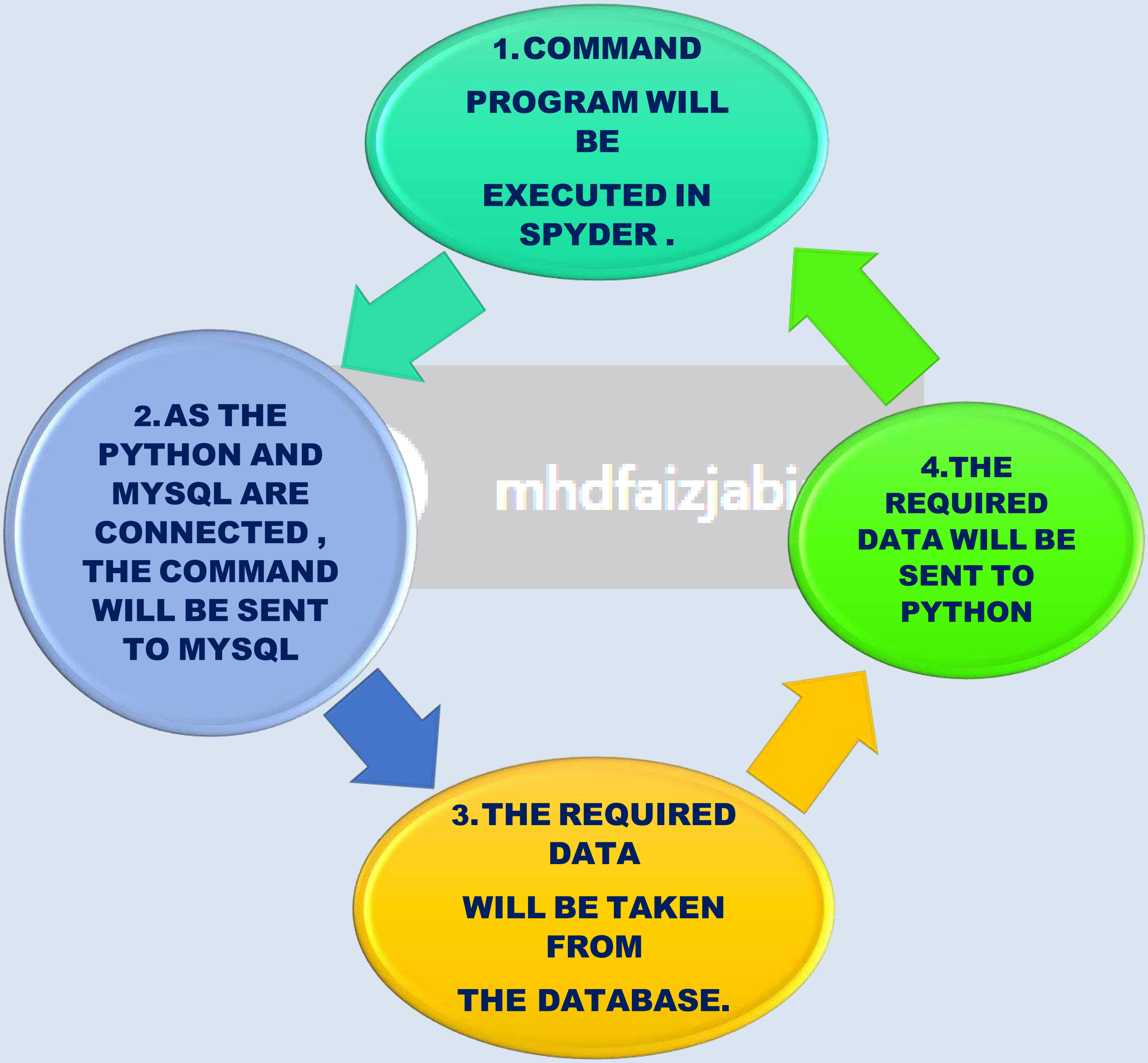
❖ IMPLEMENTATION PHASE

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

❖ OPERATIONS AND MAINTENANCE PHASE

The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to the organization’s needs. When modifications or changes are identified, the system may re-enter the planning phase.

FLOWCHART FOR HOSPITAL DATABASE MANAGEMENT SYSTEM (HDMS)



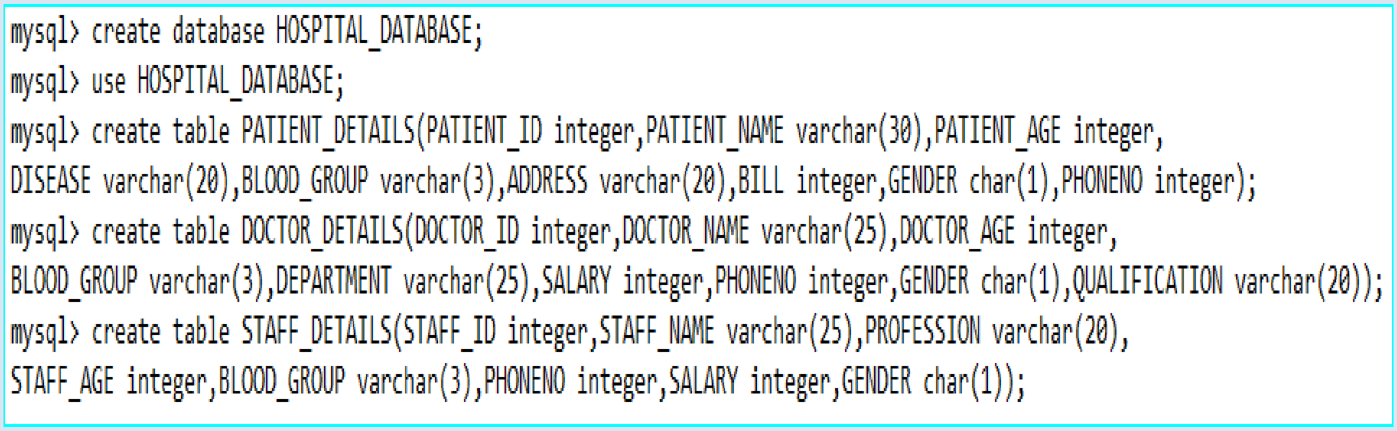
1. COMMAND PROGRAM WILL BE EXECUTED IN SPYDER .

2. AS THE PYTHON AND MYSQL ARE CONNECTED , THE COMMAND WILL BE SENT TO MYSQL

3. THE REQUIRED DATA WILL BE TAKEN FROM THE DATABASE.

4. THE REQUIRED DATA WILL BE SENT TO PYTHON

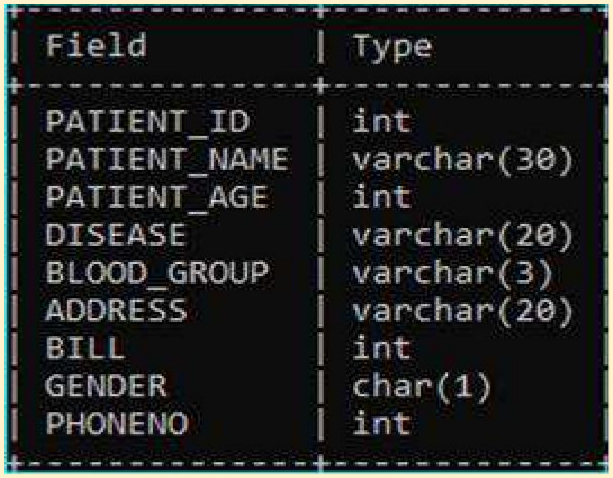
MYSQL- QUERY



```
mysql> create database HOSPITAL_DATABASE;
mysql> use HOSPITAL_DATABASE;
mysql> create table PATIENT_DETAILS(PATIENT_ID integer,PATIENT_NAME varchar(30),PATIENT_AGE integer,
DISEASE varchar(20),BLOOD_GROUP varchar(3),ADDRESS varchar(20),BILL integer,GENDER char(1),PHONENO integer);
mysql> create table DOCTOR_DETAILS(DOCTOR_ID integer,DOCTOR_NAME varchar(25),DOCTOR_AGE integer,
BLOOD_GROUP varchar(3),DEPARTMENT varchar(25),SALARY integer,PHONENO integer,GENDER char(1),QUALIFICATION varchar(20));
mysql> create table STAFF_DETAILS(STAFF_ID integer,STAFF_NAME varchar(25),PROFESSION varchar(20),
STAFF_AGE integer,BLOOD_GROUP varchar(3),PHONENO integer,SALARY integer,GENDER char(1));
```

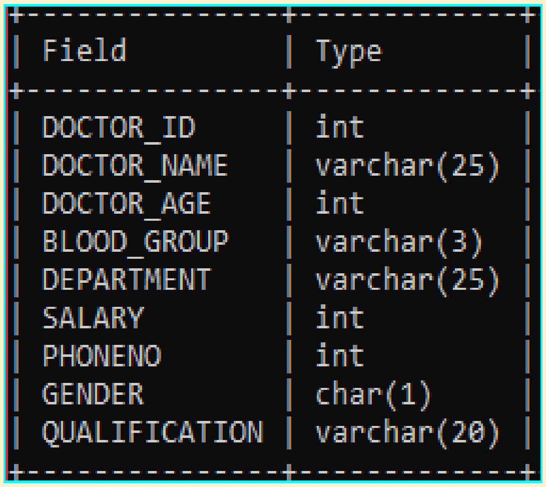
STRUCTURE OF TABLES IN MYSQL:-

- **PATIENT TABLE:**



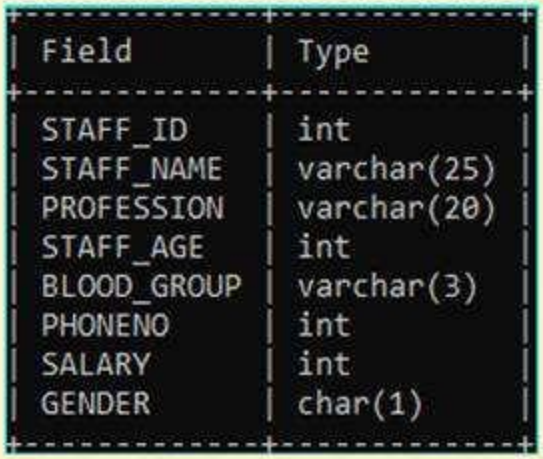
```
+--------------+-------------+
| Field        | Type        |
+--------------+-------------+
| PATIENT_ID   | int         |
| PATIENT_NAME | varchar(30) |
| PATIENT_AGE  | int         |
| DISEASE      | varchar(20) |
| BLOOD_GROUP  | varchar(3)  |
| ADDRESS      | varchar(20) |
| BILL         | int         |
| GENDER       | char(1)     |
| PHONENO      | int         |
+--------------+-------------+
```

- **DOCTOR TABLE:**



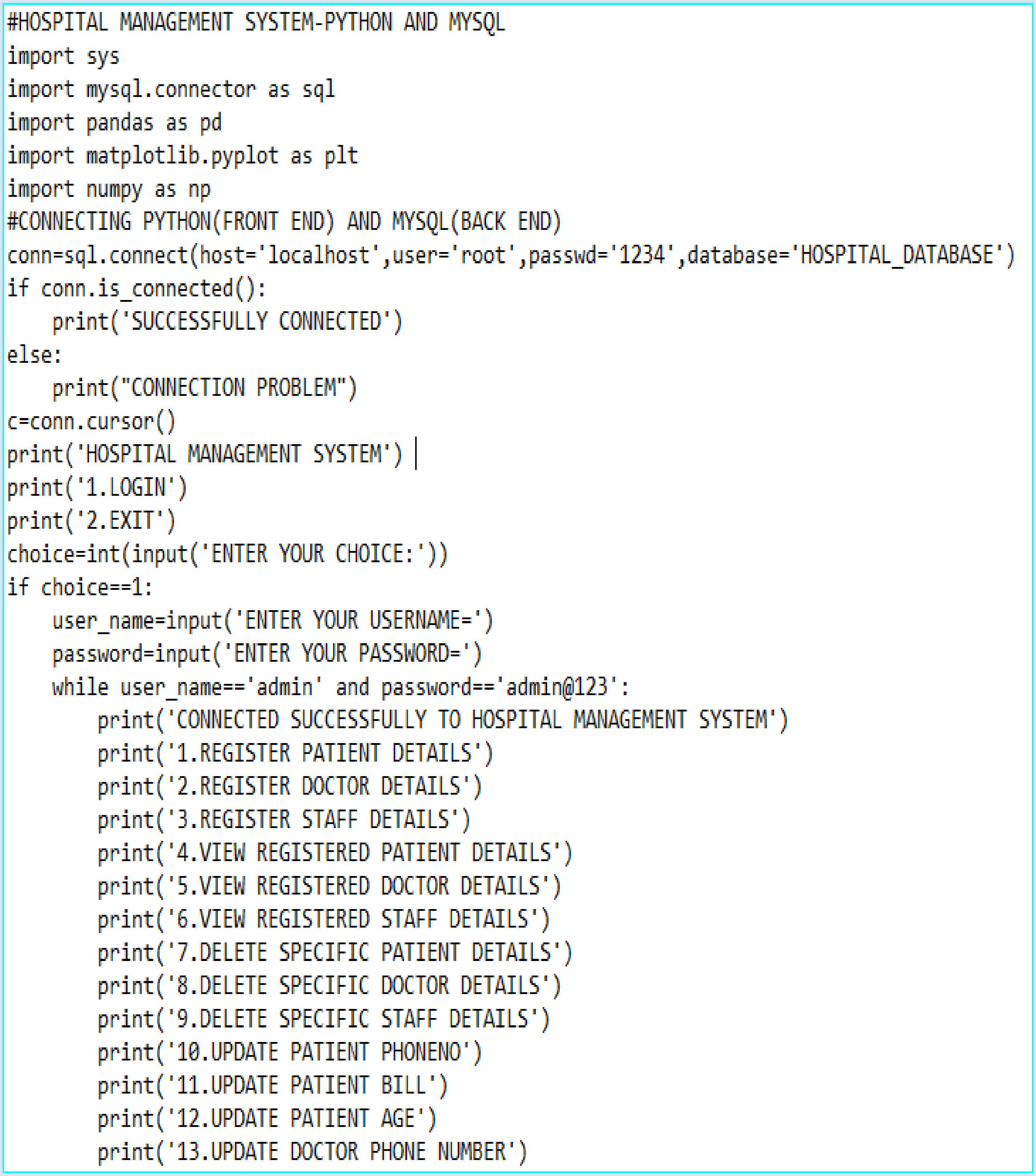
```
+---------------+-------------+
| Field         | Type        |
+---------------+-------------+
| DOCTOR_ID     | int         |
| DOCTOR_NAME   | varchar(25) |
| DOCTOR_AGE    | int         |
| BLOOD_GROUP   | varchar(3)  |
| DEPARTMENT    | varchar(25) |
| SALARY        | int         |
| PHONENO       | int         |
| GENDER        | char(1)     |
| QUALIFICATION | varchar(20) |
+---------------+-------------+
```

- **STAFF TABLE:**

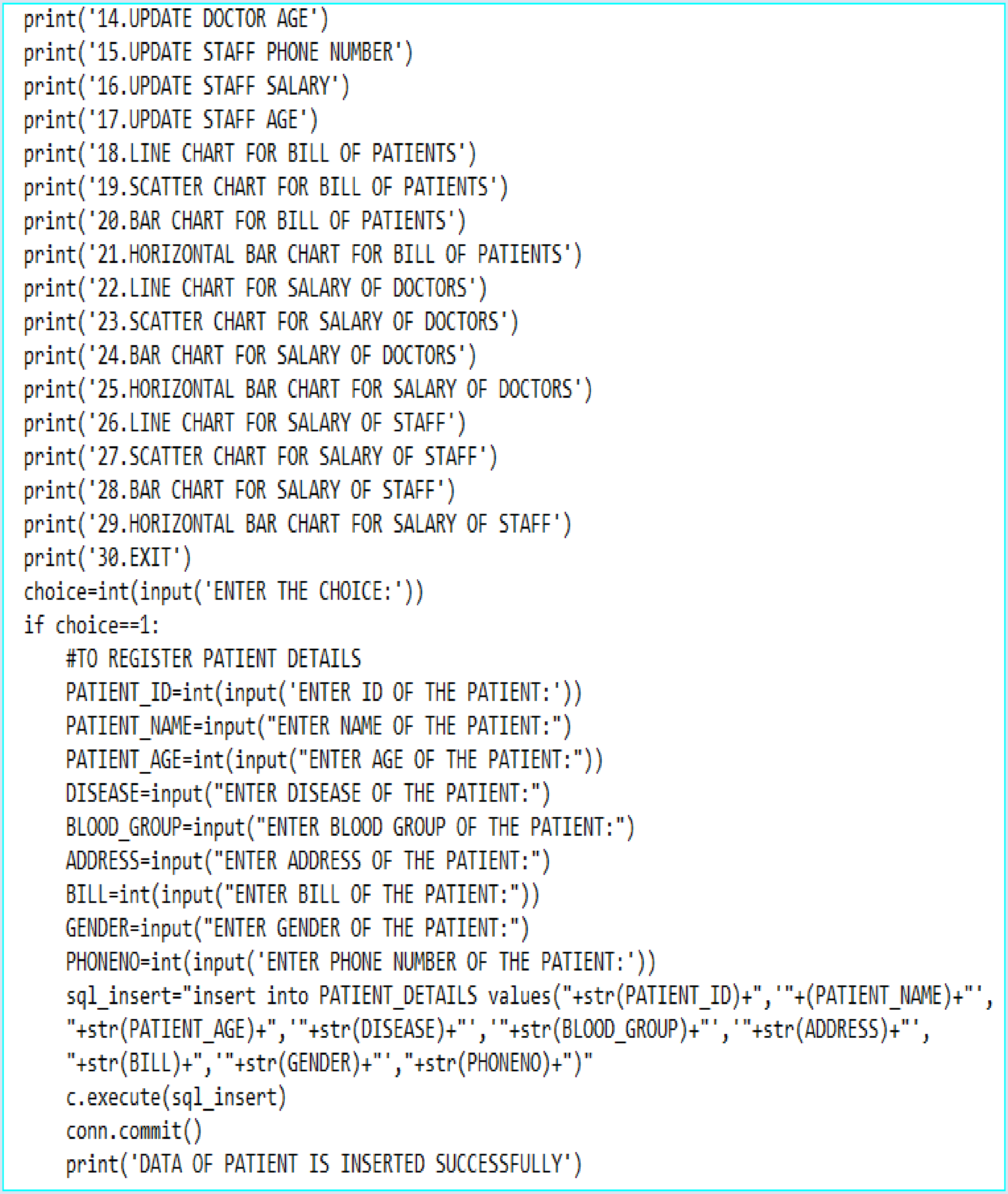


```
+--------------+-------------+
| Field        | Type        |
+--------------+-------------+
| STAFF_ID     | int         |
| STAFF_NAME   | varchar(25) |
| PROFESSION   | varchar(20) |
| STAFF_AGE    | int         |
| BLOOD_GROUP  | varchar(3)  |
| PHONENO      | int         |
| SALARY       | int         |
| GENDER       | char(1)     |
+--------------+-------------+
```

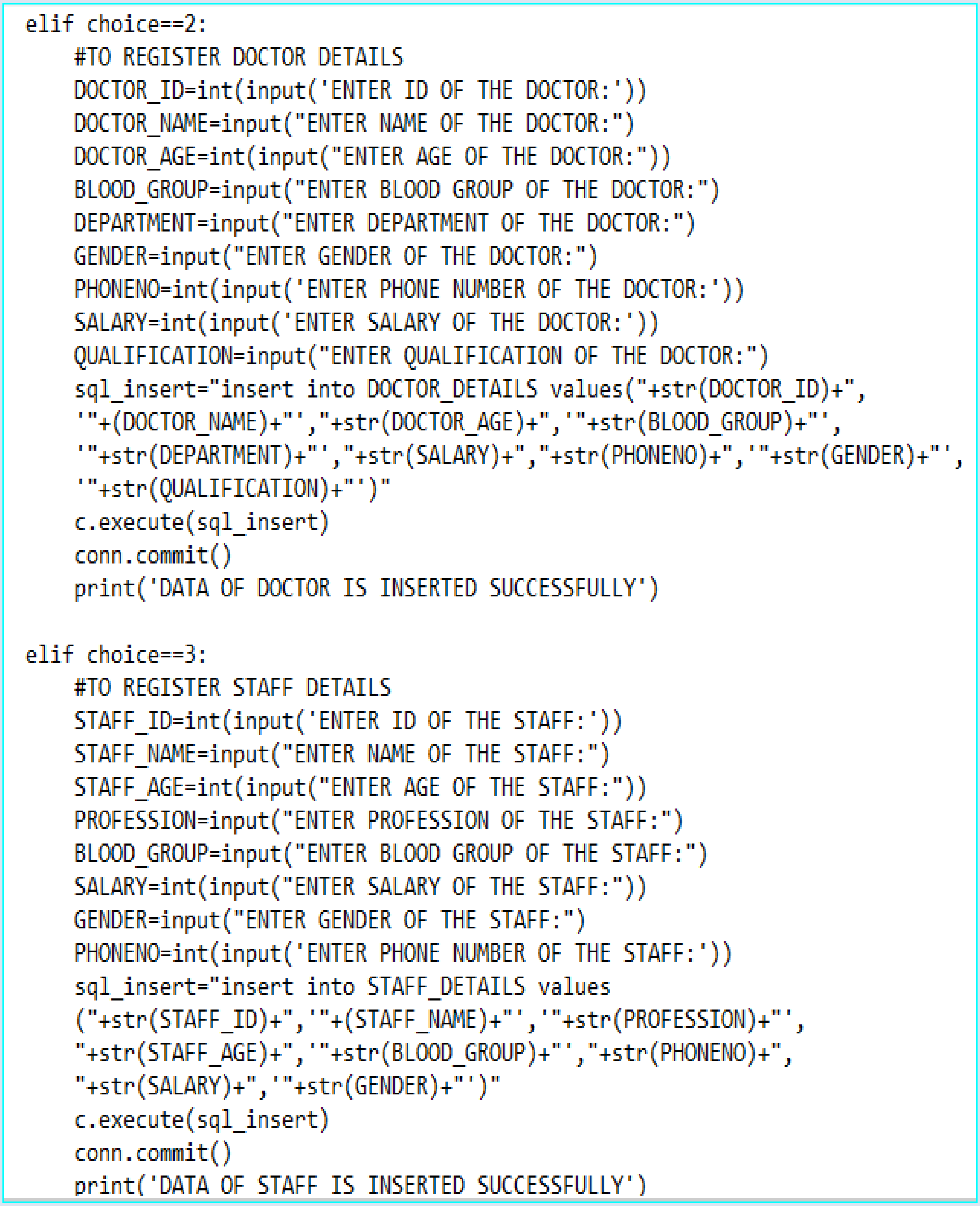
PYTHON – SOURCE CODE



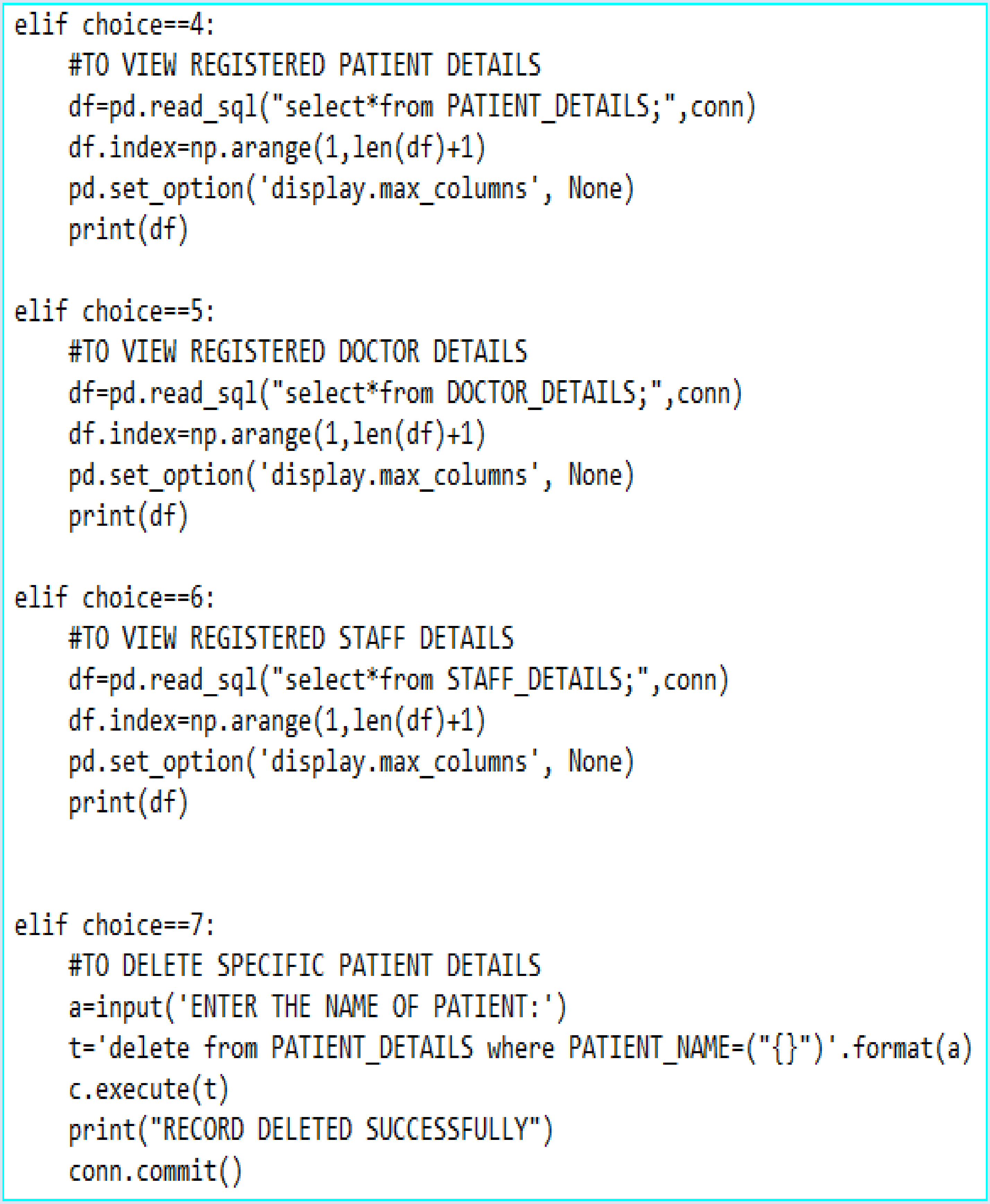
```
#HOSPITAL MANAGEMENT SYSTEM-PYTHON AND MYSQL
import sys
import mysql.connector as sql
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
#CONNECTING PYTHON(FRONT END) AND MYSQL(BACK END)
conn=sql.connect(host='localhost',user='root',passwd='1234',database='HOSPITAL_DATABASE')
if conn.is_connected():
    print('SUCCESSFULLY CONNECTED')
else:
    print("CONNECTION PROBLEM")
c=conn.cursor()
print('HOSPITAL MANAGEMENT SYSTEM')
print('1.LOGIN')
print('2.EXIT')
choice=int(input('ENTER YOUR CHOICE:'))
if choice==1:
    user_name=input('ENTER YOUR USERNAME=')
    password=input('ENTER YOUR PASSWORD=')
    while user_name=='admin' and password=='admin@123':
        print('CONNECTED SUCCESSFULLY TO HOSPITAL MANAGEMENT SYSTEM')
        print('1.REGISTER PATIENT DETAILS')
        print('2.REGISTER DOCTOR DETAILS')
        print('3.REGISTER STAFF DETAILS')
        print('4.VIEW REGISTERED PATIENT DETAILS')
        print('5.VIEW REGISTERED DOCTOR DETAILS')
        print('6.VIEW REGISTERED STAFF DETAILS')
        print('7.DELETE SPECIFIC PATIENT DETAILS')
        print('8.DELETE SPECIFIC DOCTOR DETAILS')
        print('9.DELETE SPECIFIC STAFF DETAILS')
        print('10.UPDATE PATIENT PHONENO')
        print('11.UPDATE PATIENT BILL')
        print('12.UPDATE PATIENT AGE')
        print('13.UPDATE DOCTOR PHONE NUMBER')
```



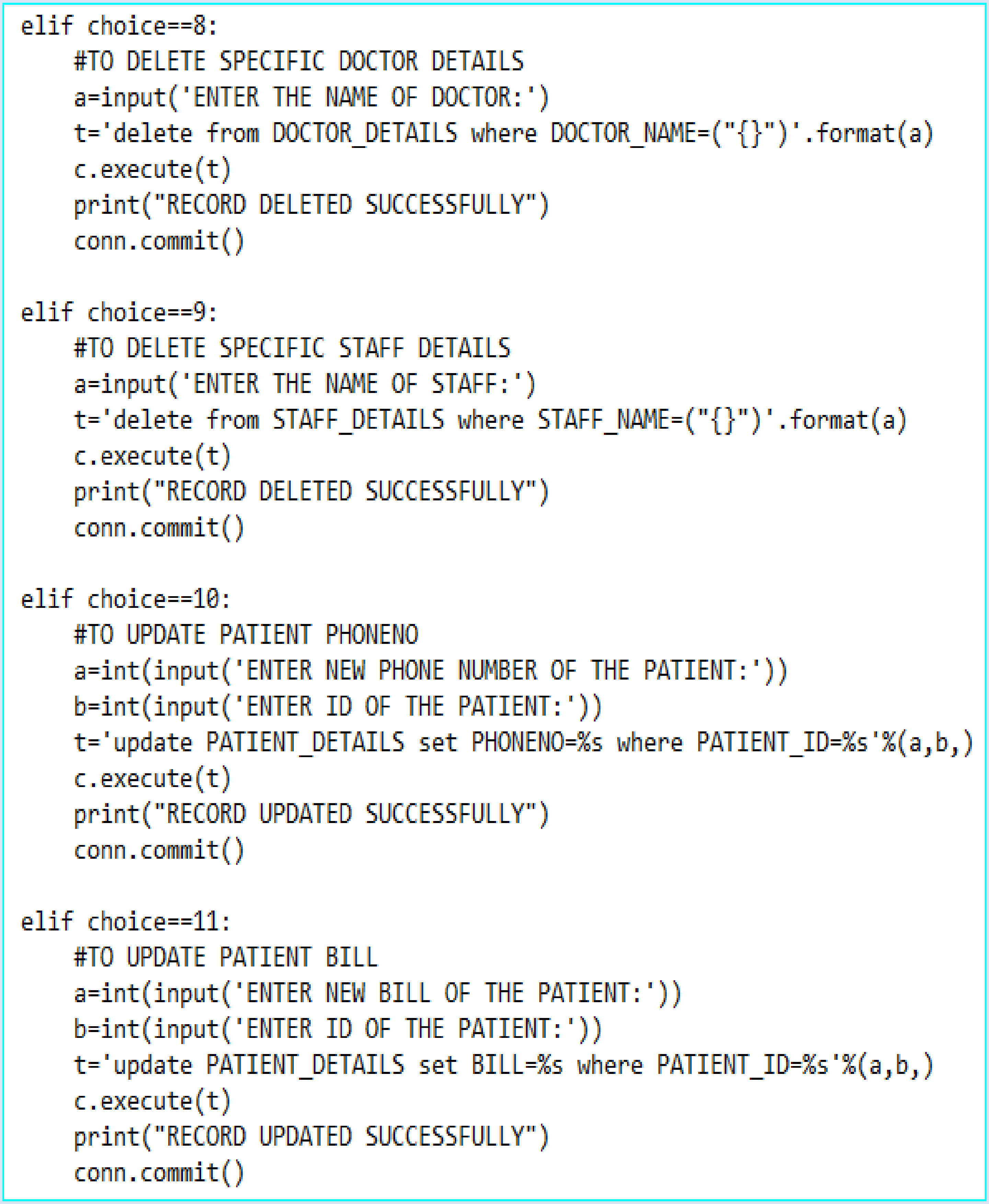
```
print('14.UPDATE DOCTOR AGE')
print('15.UPDATE STAFF PHONE NUMBER')
print('16.UPDATE STAFF SALARY')
print('17.UPDATE STAFF AGE')
print('18.LINE CHART FOR BILL OF PATIENTS')
print('19.SCATTER CHART FOR BILL OF PATIENTS')
print('20.BAR CHART FOR BILL OF PATIENTS')
print('21.HORIZONTAL BAR CHART FOR BILL OF PATIENTS')
print('22.LINE CHART FOR SALARY OF DOCTORS')
print('23.SCATTER CHART FOR SALARY OF DOCTORS')
print('24.BAR CHART FOR SALARY OF DOCTORS')
print('25.HORIZONTAL BAR CHART FOR SALARY OF DOCTORS')
print('26.LINE CHART FOR SALARY OF STAFF')
print('27.SCATTER CHART FOR SALARY OF STAFF')
print('28.BAR CHART FOR SALARY OF STAFF')
print('29.HORIZONTAL BAR CHART FOR SALARY OF STAFF')
print('30.EXIT')
choice=int(input('ENTER THE CHOICE:'))
if choice==1:
    #TO REGISTER PATIENT DETAILS
    PATIENT_ID=int(input('ENTER ID OF THE PATIENT:'))
    PATIENT_NAME=input("ENTER NAME OF THE PATIENT:")
    PATIENT_AGE=int(input("ENTER AGE OF THE PATIENT:"))
    DISEASE=input("ENTER DISEASE OF THE PATIENT:")
    BLOOD_GROUP=input("ENTER BLOOD GROUP OF THE PATIENT:")
    ADDRESS=input("ENTER ADDRESS OF THE PATIENT:")
    BILL=int(input("ENTER BILL OF THE PATIENT:"))
    GENDER=input("ENTER GENDER OF THE PATIENT:")
    PHONENO=int(input('ENTER PHONE NUMBER OF THE PATIENT:'))
    sql_insert="insert into PATIENT_DETAILS values("+str(PATIENT_ID)+",'"+(PATIENT_NAME)+"',
    "+str(PATIENT_AGE)+",'"+str(DISEASE)+"','"+str(BLOOD_GROUP)+"','"+str(ADDRESS)+"',
    "+str(BILL)+",'"+str(GENDER)+"',"+str(PHONENO)+")"
    c.execute(sql_insert)
    conn.commit()
    print('DATA OF PATIENT IS INSERTED SUCCESSFULLY')
```



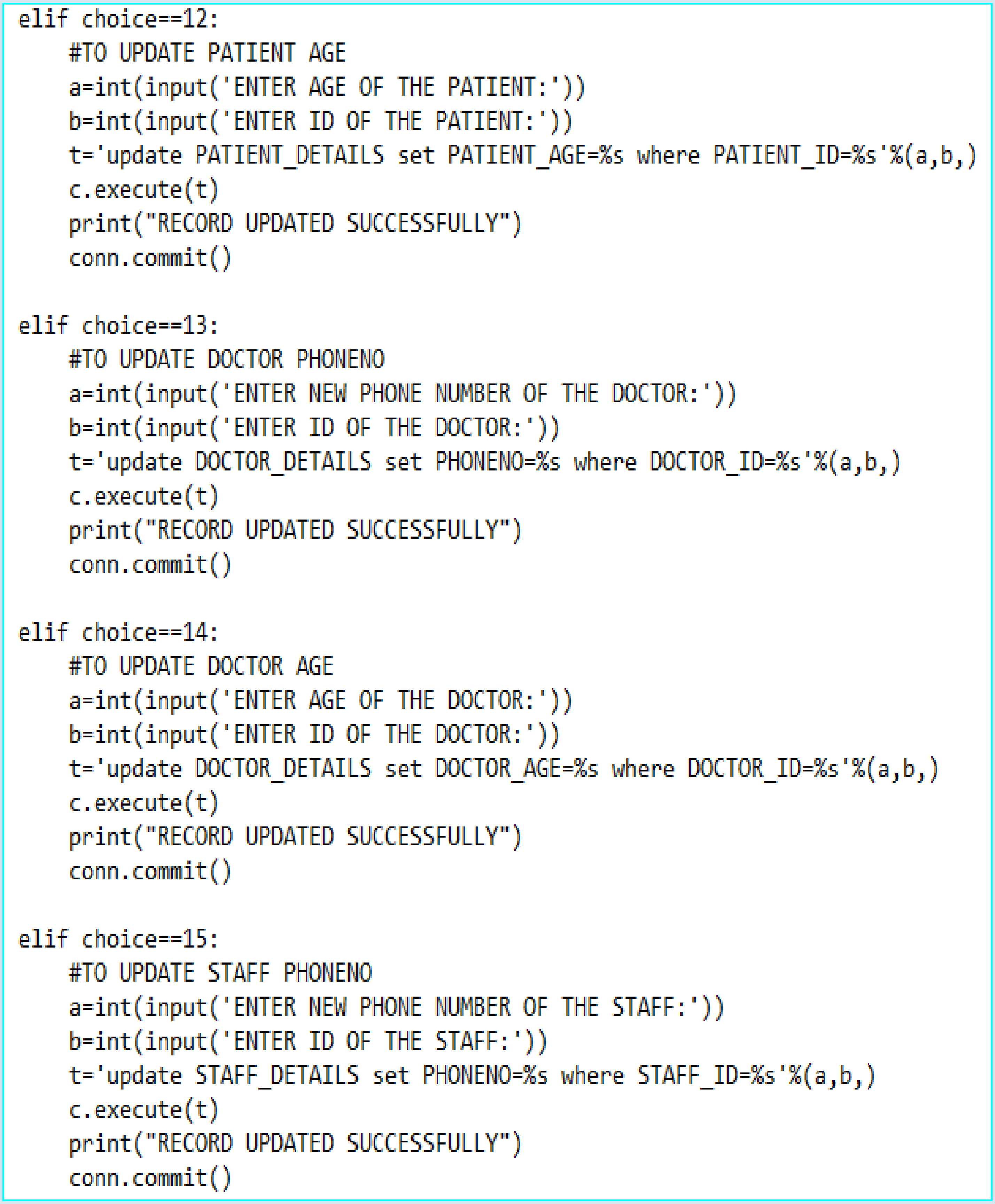
```
elif choice==2:
    #TO REGISTER DOCTOR DETAILS
    DOCTOR_ID=int(input('ENTER ID OF THE DOCTOR:'))
    DOCTOR_NAME=input("ENTER NAME OF THE DOCTOR:")
    DOCTOR_AGE=int(input("ENTER AGE OF THE DOCTOR:"))
    BLOOD_GROUP=input("ENTER BLOOD GROUP OF THE DOCTOR:")
    DEPARTMENT=input("ENTER DEPARTMENT OF THE DOCTOR:")
    GENDER=input("ENTER GENDER OF THE DOCTOR:")
    PHONENO=int(input('ENTER PHONE NUMBER OF THE DOCTOR:'))
    SALARY=int(input('ENTER SALARY OF THE DOCTOR:'))
    QUALIFICATION=input("ENTER QUALIFICATION OF THE DOCTOR:")
    sql_insert="insert into DOCTOR_DETAILS values("+str(DOCTOR_ID)+",
    '"+(DOCTOR_NAME)+"',"+str(DOCTOR_AGE)+",'"+str(BLOOD_GROUP)+"',
    '"+str(DEPARTMENT)+"',"+str(SALARY)+","+str(PHONENO)+",'"+str(GENDER)+"',
    '"+str(QUALIFICATION)+"')"
    c.execute(sql_insert)
    conn.commit()
    print('DATA OF DOCTOR IS INSERTED SUCCESSFULLY')

elif choice==3:
    #TO REGISTER STAFF DETAILS
    STAFF_ID=int(input('ENTER ID OF THE STAFF:'))
    STAFF_NAME=input("ENTER NAME OF THE STAFF:")
    STAFF_AGE=int(input("ENTER AGE OF THE STAFF:"))
    PROFESSION=input("ENTER PROFESSION OF THE STAFF:")
    BLOOD_GROUP=input("ENTER BLOOD GROUP OF THE STAFF:")
    SALARY=int(input("ENTER SALARY OF THE STAFF:"))
    GENDER=input("ENTER GENDER OF THE STAFF:")
    PHONENO=int(input('ENTER PHONE NUMBER OF THE STAFF:'))
    sql_insert="insert into STAFF_DETAILS values
    ("+str(STAFF_ID)+",'"+(STAFF_NAME)+"','"+str(PROFESSION)+"',
    "+str(STAFF_AGE)+",'"+str(BLOOD_GROUP)+"',"+str(PHONENO)+",
    "+str(SALARY)+",'"+str(GENDER)+"')"
    c.execute(sql_insert)
    conn.commit()
    print('DATA OF STAFF IS INSERTED SUCCESSFULLY')
```



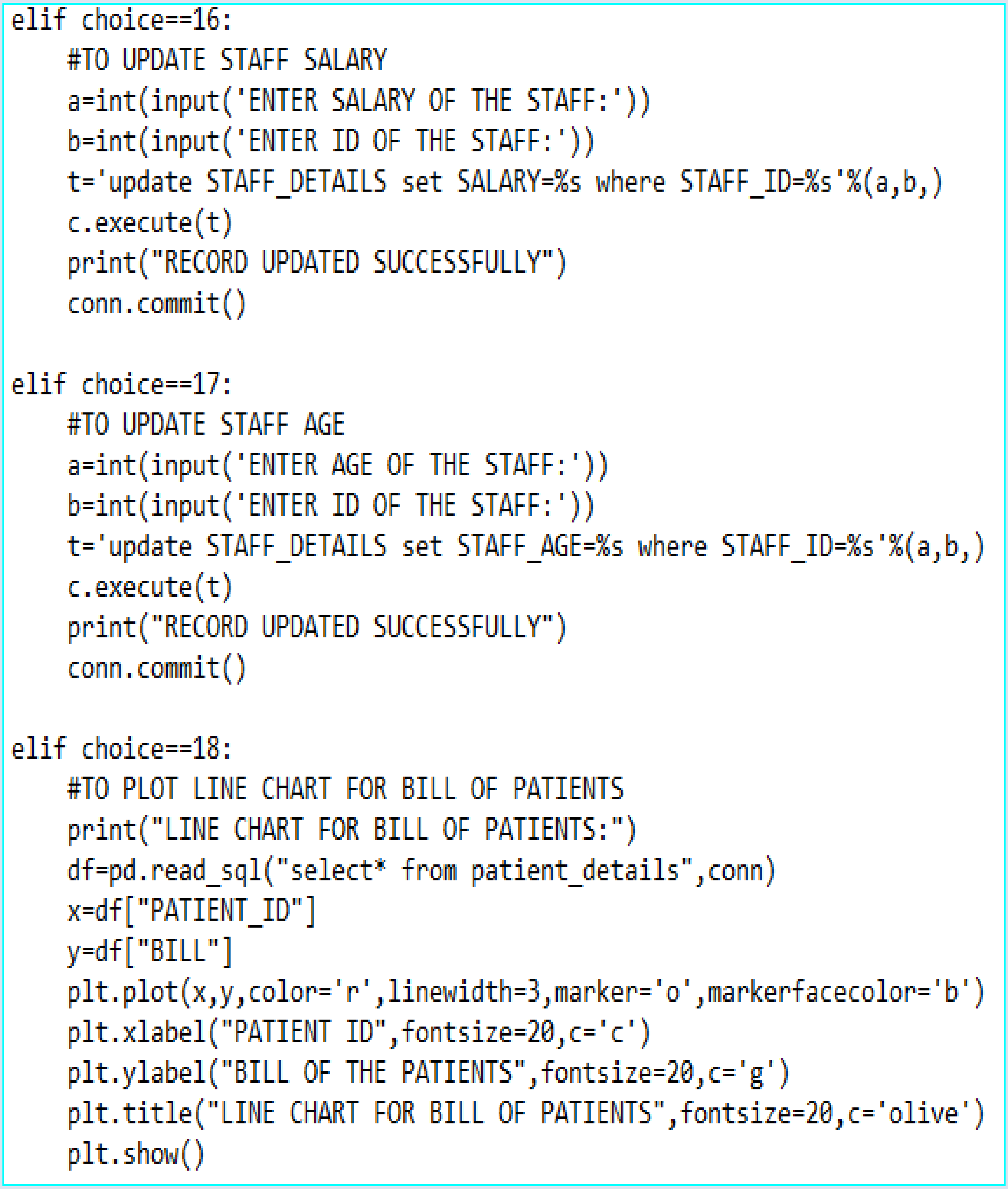
```
elif choice==4:
    #TO VIEW REGISTERED PATIENT DETAILS
    df=pd.read_sql("select*from PATIENT_DETAILS;",conn)
    df.index=np.arange(1,len(df)+1)
    pd.set_option('display.max_columns', None)
    print(df)

elif choice==5:
    #TO VIEW REGISTERED DOCTOR DETAILS
    df=pd.read_sql("select*from DOCTOR_DETAILS;",conn)
    df.index=np.arange(1,len(df)+1)
    pd.set_option('display.max_columns', None)
    print(df)

elif choice==6:
    #TO VIEW REGISTERED STAFF DETAILS
    df=pd.read_sql("select*from STAFF_DETAILS;",conn)
    df.index=np.arange(1,len(df)+1)
    pd.set_option('display.max_columns', None)
    print(df)



elif choice==7:
    #TO DELETE SPECIFIC PATIENT DETAILS
    a=input('ENTER THE NAME OF PATIENT:')
    t='delete from PATIENT_DETAILS where PATIENT_NAME=("{}")'.format(a)
    c.execute(t)
    print("RECORD DELETED SUCCESSFULLY")
    conn.commit()
```



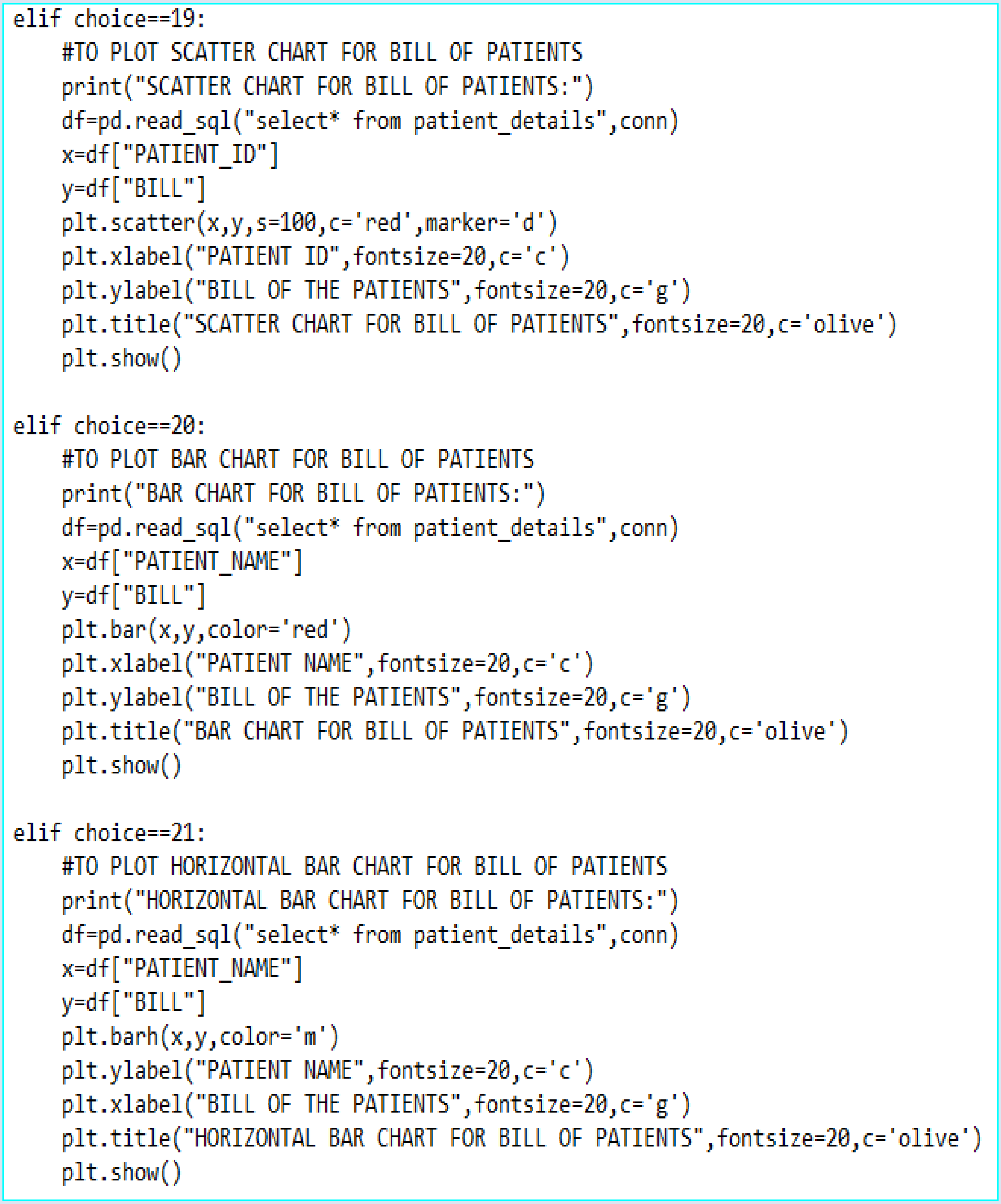
```
elif choice==8:
    #TO DELETE SPECIFIC DOCTOR DETAILS
    a=input('ENTER THE NAME OF DOCTOR:')
    t='delete from DOCTOR_DETAILS where DOCTOR_NAME=("{}")'.format(a)
    c.execute(t)
    print("RECORD DELETED SUCCESSFULLY")
    conn.commit()

elif choice==9:
    #TO DELETE SPECIFIC STAFF DETAILS
    a=input('ENTER THE NAME OF STAFF:')
    t='delete from STAFF_DETAILS where STAFF_NAME=("{}")'.format(a)
    c.execute(t)
    print("RECORD DELETED SUCCESSFULLY")
    conn.commit()

elif choice==10:
    #TO UPDATE PATIENT PHONENO
    a=int(input('ENTER NEW PHONE NUMBER OF THE PATIENT:'))
    b=int(input('ENTER ID OF THE PATIENT:'))
    t='update PATIENT_DETAILS set PHONENO=%s where PATIENT_ID=%s'%(a,b,)
    c.execute(t)
    print("RECORD UPDATED SUCCESSFULLY")
    conn.commit()

elif choice==11:
    #TO UPDATE PATIENT BILL
    a=int(input('ENTER NEW BILL OF THE PATIENT:'))
    b=int(input('ENTER ID OF THE PATIENT:'))
    t='update PATIENT_DETAILS set BILL=%s where PATIENT_ID=%s'%(a,b,)
    c.execute(t)
    print("RECORD UPDATED SUCCESSFULLY")
    conn.commit()
```



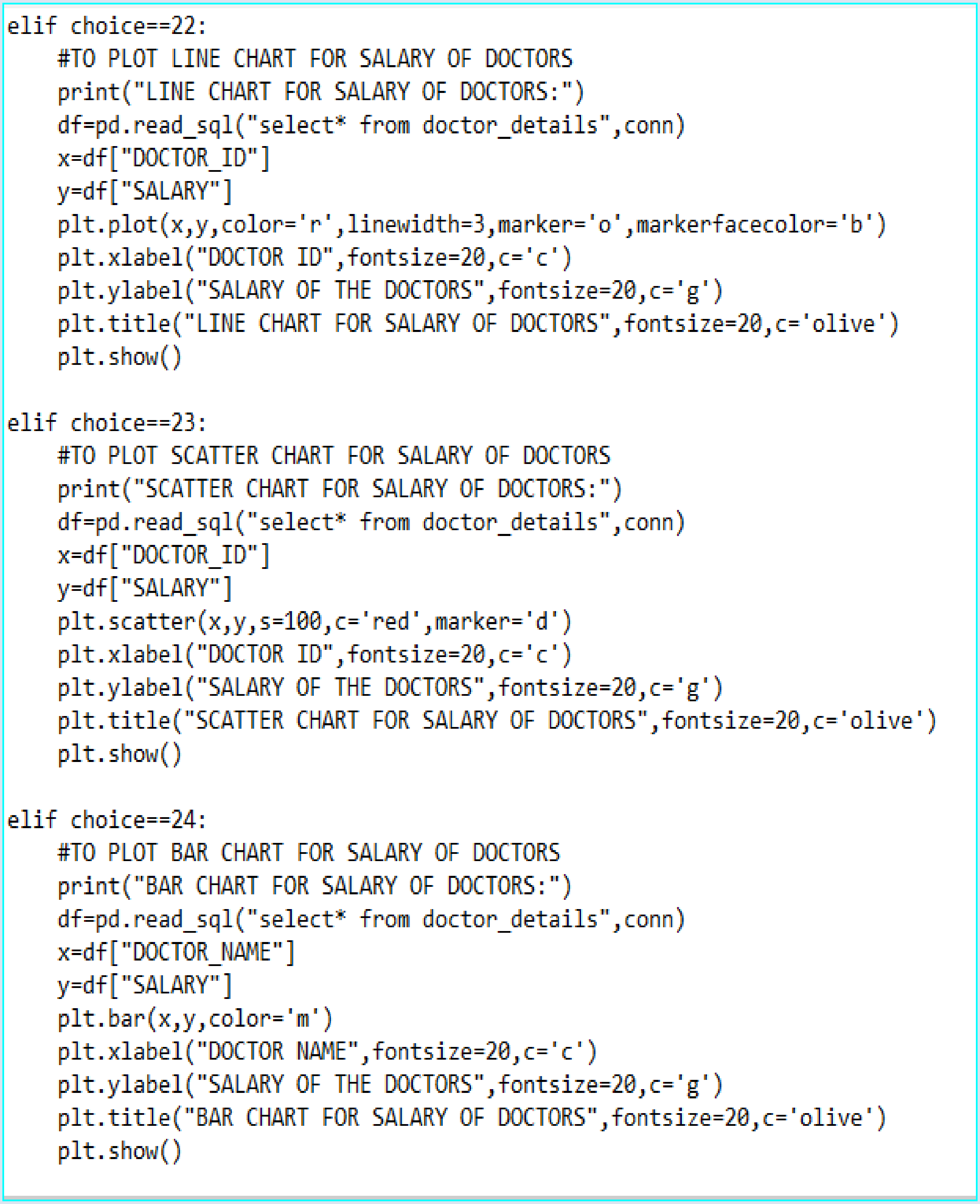
```
elif choice==12:
    #TO UPDATE PATIENT AGE
    a=int(input('ENTER AGE OF THE PATIENT:'))
    b=int(input('ENTER ID OF THE PATIENT:'))
    t='update PATIENT_DETAILS set PATIENT_AGE=%s where PATIENT_ID=%s'%(a,b,)
    c.execute(t)
    print("RECORD UPDATED SUCCESSFULLY")
    conn.commit()

elif choice==13:
    #TO UPDATE DOCTOR PHONENO
    a=int(input('ENTER NEW PHONE NUMBER OF THE DOCTOR:'))
    b=int(input('ENTER ID OF THE DOCTOR:'))
    t='update DOCTOR_DETAILS set PHONENO=%s where DOCTOR_ID=%s'%(a,b,)
    c.execute(t)
    print("RECORD UPDATED SUCCESSFULLY")
    conn.commit()

elif choice==14:
    #TO UPDATE DOCTOR AGE
    a=int(input('ENTER AGE OF THE DOCTOR:'))
    b=int(input('ENTER ID OF THE DOCTOR:'))
    t='update DOCTOR_DETAILS set DOCTOR_AGE=%s where DOCTOR_ID=%s'%(a,b,)
    c.execute(t)
    print("RECORD UPDATED SUCCESSFULLY")
    conn.commit()

elif choice==15:
    #TO UPDATE STAFF PHONENO
    a=int(input('ENTER NEW PHONE NUMBER OF THE STAFF:'))
    b=int(input('ENTER ID OF THE STAFF:'))
    t='update STAFF_DETAILS set PHONENO=%s where STAFF_ID=%s'%(a,b,)
    c.execute(t)
    print("RECORD UPDATED SUCCESSFULLY")
    conn.commit()
```



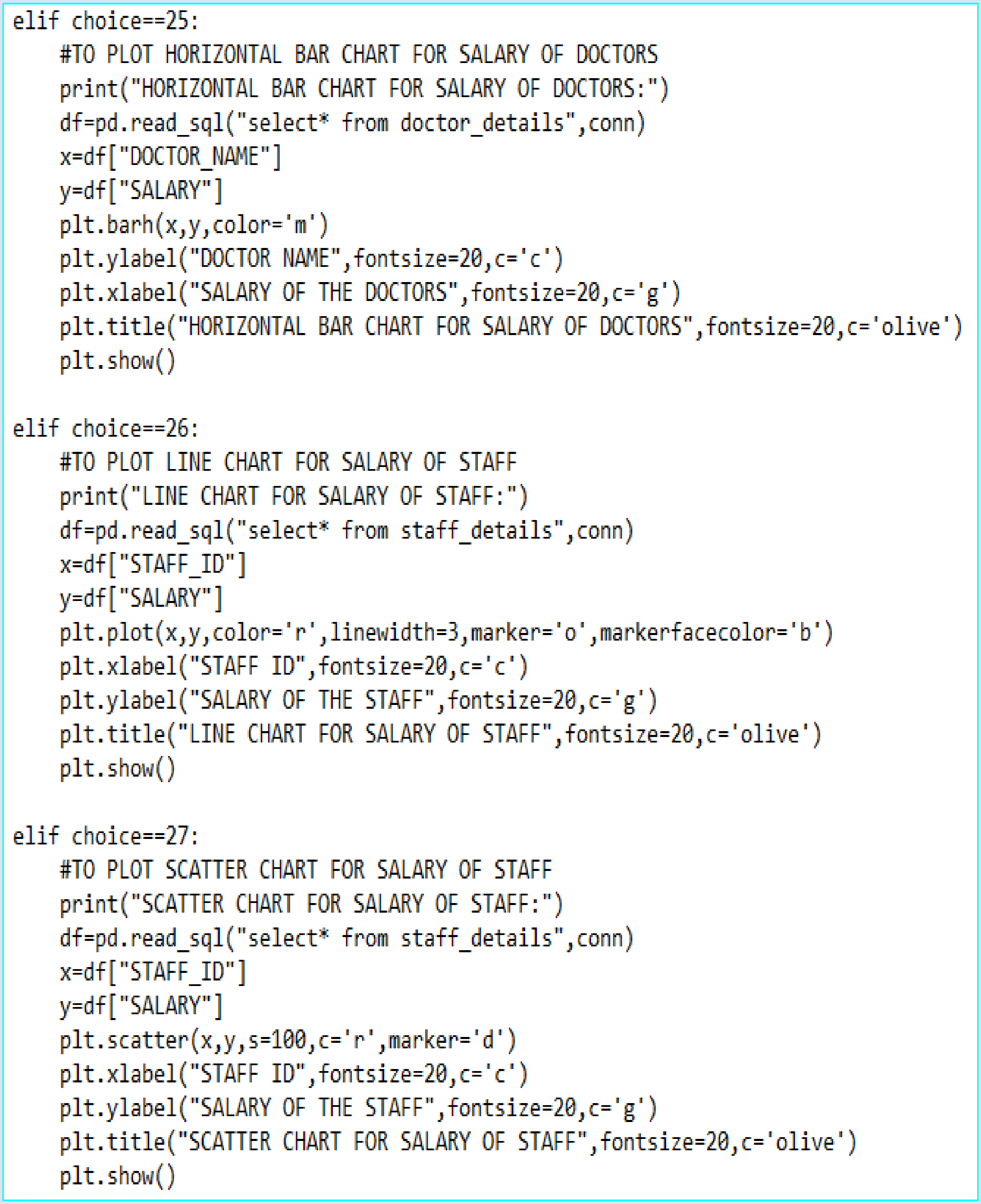
```
elif choice==16:
    #TO UPDATE STAFF SALARY
    a=int(input('ENTER SALARY OF THE STAFF:'))
    b=int(input('ENTER ID OF THE STAFF:'))
    t='update STAFF_DETAILS set SALARY=%s where STAFF_ID=%s'%(a,b,)
    c.execute(t)
    print("RECORD UPDATED SUCCESSFULLY")
    conn.commit()

elif choice==17:
    #TO UPDATE STAFF AGE
    a=int(input('ENTER AGE OF THE STAFF:'))
    b=int(input('ENTER ID OF THE STAFF:'))
    t='update STAFF_DETAILS set STAFF_AGE=%s where STAFF_ID=%s'%(a,b,)
    c.execute(t)
    print("RECORD UPDATED SUCCESSFULLY")
    conn.commit()

elif choice==18:
    #TO PLOT LINE CHART FOR BILL OF PATIENTS
    print("LINE CHART FOR BILL OF PATIENTS:")
    df=pd.read_sql("select* from patient_details",conn)
    x=df["PATIENT_ID"]
    y=df["BILL"]
    plt.plot(x,y,color='r',linewidth=3,marker='o',markerfacecolor='b')
    plt.xlabel("PATIENT ID",fontsize=20,c='c')
    plt.ylabel("BILL OF THE PATIENTS",fontsize=20,c='g')
    plt.title("LINE CHART FOR BILL OF PATIENTS",fontsize=20,c='olive')
    plt.show()
```



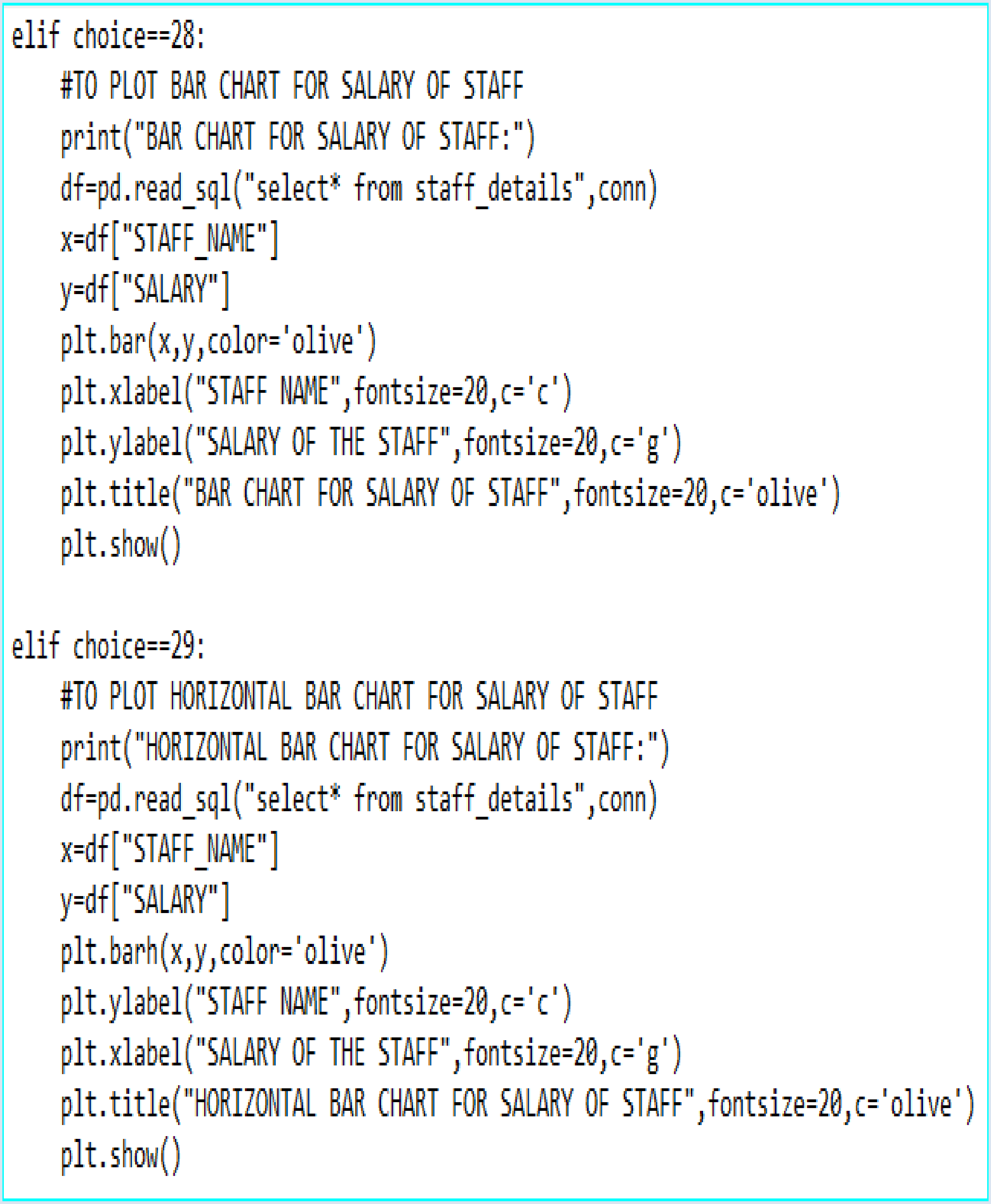
```
elif choice==19:
    #TO PLOT SCATTER CHART FOR BILL OF PATIENTS
    print("SCATTER CHART FOR BILL OF PATIENTS:")
    df=pd.read_sql("select* from patient_details",conn)
    x=df["PATIENT_ID"]
    y=df["BILL"]
    plt.scatter(x,y,s=100,c='red',marker='d')
    plt.xlabel("PATIENT ID",fontsize=20,c='c')
    plt.ylabel("BILL OF THE PATIENTS",fontsize=20,c='g')
    plt.title("SCATTER CHART FOR BILL OF PATIENTS",fontsize=20,c='olive')
    plt.show()

elif choice==20:
    #TO PLOT BAR CHART FOR BILL OF PATIENTS
    print("BAR CHART FOR BILL OF PATIENTS:")
    df=pd.read_sql("select* from patient_details",conn)
    x=df["PATIENT_NAME"]
    y=df["BILL"]
    plt.bar(x,y,color='red')
    plt.xlabel("PATIENT NAME",fontsize=20,c='c')
    plt.ylabel("BILL OF THE PATIENTS",fontsize=20,c='g')
    plt.title("BAR CHART FOR BILL OF PATIENTS",fontsize=20,c='olive')
    plt.show()

elif choice==21:
    #TO PLOT HORIZONTAL BAR CHART FOR BILL OF PATIENTS
    print("HORIZONTAL BAR CHART FOR BILL OF PATIENTS:")
    df=pd.read_sql("select* from patient_details",conn)
    x=df["PATIENT_NAME"]
    y=df["BILL"]
    plt.barh(x,y,color='m')
    plt.ylabel("PATIENT NAME",fontsize=20,c='c')
    plt.xlabel("BILL OF THE PATIENTS",fontsize=20,c='g')
    plt.title("HORIZONTAL BAR CHART FOR BILL OF PATIENTS",fontsize=20,c='olive')
    plt.show()
```



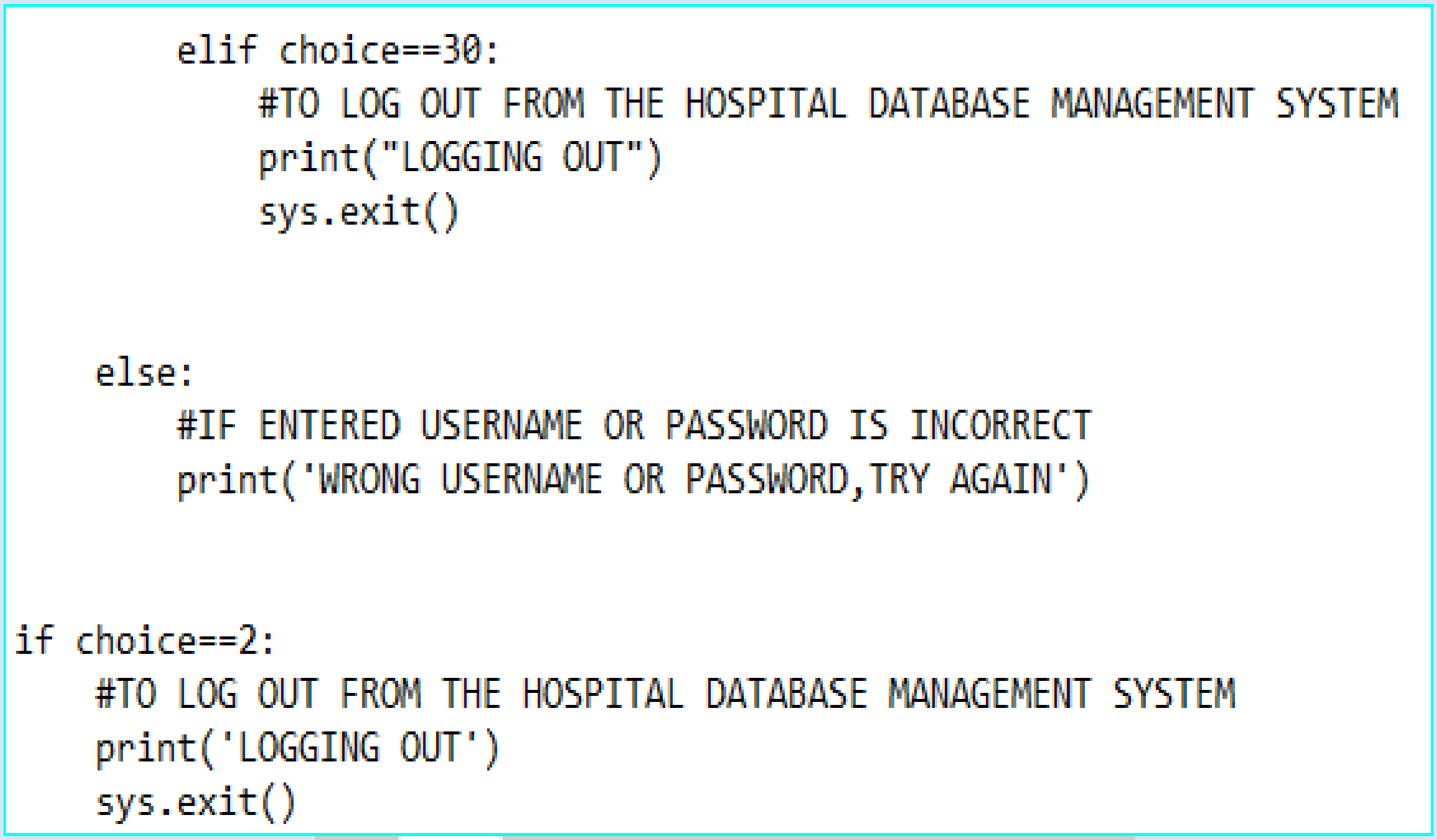
```
elif choice==22:
    #TO PLOT LINE CHART FOR SALARY OF DOCTORS
    print("LINE CHART FOR SALARY OF DOCTORS:")
    df=pd.read_sql("select* from doctor_details",conn)
    x=df["DOCTOR_ID"]
    y=df["SALARY"]
    plt.plot(x,y,color='r',linewidth=3,marker='o',markerfacecolor='b')
    plt.xlabel("DOCTOR ID",fontsize=20,c='c')
    plt.ylabel("SALARY OF THE DOCTORS",fontsize=20,c='g')
    plt.title("LINE CHART FOR SALARY OF DOCTORS",fontsize=20,c='olive')
    plt.show()

elif choice==23:
    #TO PLOT SCATTER CHART FOR SALARY OF DOCTORS
    print("SCATTER CHART FOR SALARY OF DOCTORS:")
    df=pd.read_sql("select* from doctor_details",conn)
    x=df["DOCTOR_ID"]
    y=df["SALARY"]
    plt.scatter(x,y,s=100,c='red',marker='d')
    plt.xlabel("DOCTOR ID",fontsize=20,c='c')
    plt.ylabel("SALARY OF THE DOCTORS",fontsize=20,c='g')
    plt.title("SCATTER CHART FOR SALARY OF DOCTORS",fontsize=20,c='olive')
    plt.show()

elif choice==24:
    #TO PLOT BAR CHART FOR SALARY OF DOCTORS
    print("BAR CHART FOR SALARY OF DOCTORS:")
    df=pd.read_sql("select* from doctor_details",conn)
    x=df["DOCTOR_NAME"]
    y=df["SALARY"]
    plt.bar(x,y,color='m')
    plt.xlabel("DOCTOR NAME",fontsize=20,c='c')
    plt.ylabel("SALARY OF THE DOCTORS",fontsize=20,c='g')
    plt.title("BAR CHART FOR SALARY OF DOCTORS",fontsize=20,c='olive')
    plt.show()
```



```
elif choice==25:
    #TO PLOT HORIZONTAL BAR CHART FOR SALARY OF DOCTORS
    print("HORIZONTAL BAR CHART FOR SALARY OF DOCTORS:")
    df=pd.read_sql("select* from doctor_details",conn)
    x=df["DOCTOR_NAME"]
    y=df["SALARY"]
    plt.barh(x,y,color='m')
    plt.ylabel("DOCTOR NAME",fontsize=20,c='c')
    plt.xlabel("SALARY OF THE DOCTORS",fontsize=20,c='g')
    plt.title("HORIZONTAL BAR CHART FOR SALARY OF DOCTORS",fontsize=20,c='olive')
    plt.show()

elif choice==26:
    #TO PLOT LINE CHART FOR SALARY OF STAFF
    print("LINE CHART FOR SALARY OF STAFF:")
    df=pd.read_sql("select* from staff_details",conn)
    x=df["STAFF_ID"]
    y=df["SALARY"]
    plt.plot(x,y,color='r',linewidth=3,marker='o',markerfacecolor='b')
    plt.xlabel("STAFF ID",fontsize=20,c='c')
    plt.ylabel("SALARY OF THE STAFF",fontsize=20,c='g')
    plt.title("LINE CHART FOR SALARY OF STAFF",fontsize=20,c='olive')
    plt.show()

elif choice==27:
    #TO PLOT SCATTER CHART FOR SALARY OF STAFF
    print("SCATTER CHART FOR SALARY OF STAFF:")
    df=pd.read_sql("select* from staff_details",conn)
    x=df["STAFF_ID"]
    y=df["SALARY"]
    plt.scatter(x,y,s=100,c='r',marker='d')
    plt.xlabel("STAFF ID",fontsize=20,c='c')
    plt.ylabel("SALARY OF THE STAFF",fontsize=20,c='g')
    plt.title("SCATTER CHART FOR SALARY OF STAFF",fontsize=20,c='olive')
    plt.show()
```



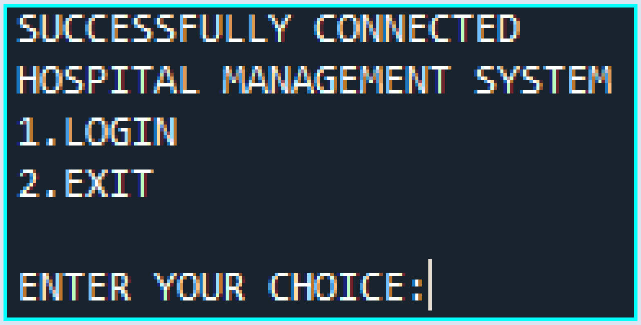
```
elif choice==28:
    #TO PLOT BAR CHART FOR SALARY OF STAFF
    print("BAR CHART FOR SALARY OF STAFF:")
    df=pd.read_sql("select* from staff_details",conn)
    x=df["STAFF_NAME"]
    y=df["SALARY"]
    plt.bar(x,y,color='olive')
    plt.xlabel("STAFF NAME",fontsize=20,c='c')
    plt.ylabel("SALARY OF THE STAFF",fontsize=20,c='g')
    plt.title("BAR CHART FOR SALARY OF STAFF",fontsize=20,c='olive')
    plt.show()

elif choice==29:
    #TO PLOT HORIZONTAL BAR CHART FOR SALARY OF STAFF
    print("HORIZONTAL BAR CHART FOR SALARY OF STAFF:")
    df=pd.read_sql("select* from staff_details",conn)
    x=df["STAFF_NAME"]
    y=df["SALARY"]
    plt.barh(x,y,color='olive')
    plt.ylabel("STAFF NAME",fontsize=20,c='c')
    plt.xlabel("SALARY OF THE STAFF",fontsize=20,c='g')
    plt.title("HORIZONTAL BAR CHART FOR SALARY OF STAFF",fontsize=20,c='olive')
    plt.show()
```



```
        elif choice==30:
            #TO LOG OUT FROM THE HOSPITAL DATABASE MANAGEMENT SYSTEM
            print("LOGGING OUT")
            sys.exit()



    else:
        #IF ENTERED USERNAME OR PASSWORD IS INCORRECT
        print('WRONG USERNAME OR PASSWORD,TRY AGAIN')



if choice==2:
    #TO LOG OUT FROM THE HOSPITAL DATABASE MANAGEMENT SYSTEM
    print('LOGGING OUT')
    sys.exit()
```

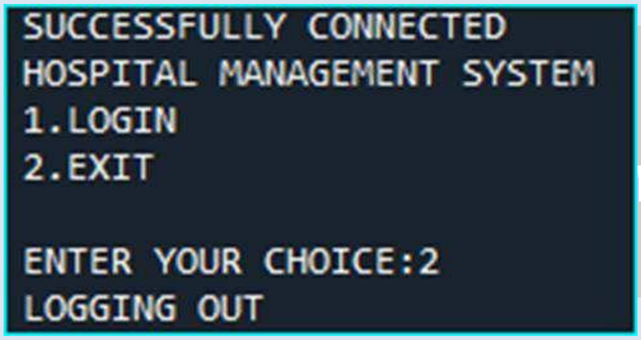
OUTPUT

1. *WHEN THE PROGRAM IS EXECUTED, IT WILL DISPLAY THAT THE PYTHON IS SUCCESSFULLY CONNECTED TO MYSQL. THE USER CAN ENTER 1 OR 2 TO PERFORM THEIR RESPECTIVE FUNCTIONS.*



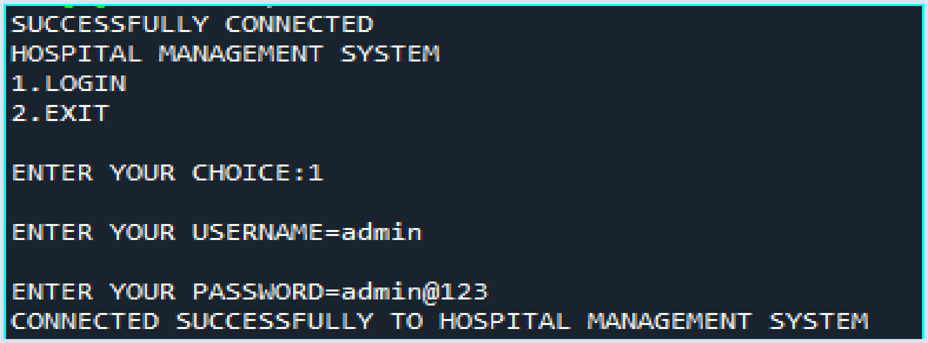
```
SUCCESSFULLY CONNECTED
HOSPITAL MANAGEMENT SYSTEM
1.LOGIN
2.EXIT

ENTER YOUR CHOICE:
```

2. *IF THE ENTERED CHOICE IS 2, THE USER WILL LOG OUT FROM THE HOSPITAL DATABASE MANAGEMENT SYSTEM.*



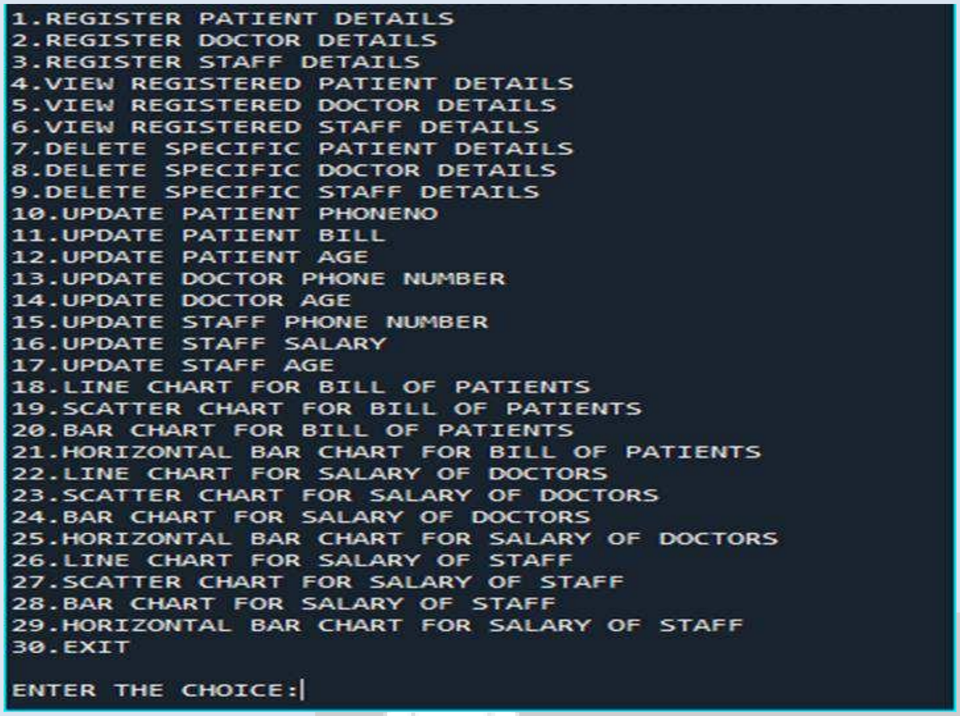
```
SUCCESSFULLY CONNECTED
HOSPITAL MANAGEMENT SYSTEM
1.LOGIN
2.EXIT

ENTER YOUR CHOICE:2
LOGGING OUT
```

3. *IF THE ENTERED CHOICE IS 1, THEN THE USER CAN ENTER THE USERNAME AND PASSWORD TO LOGIN TO THE HOSPITAL DATABASE MANAGEMENT SYSTEM.*



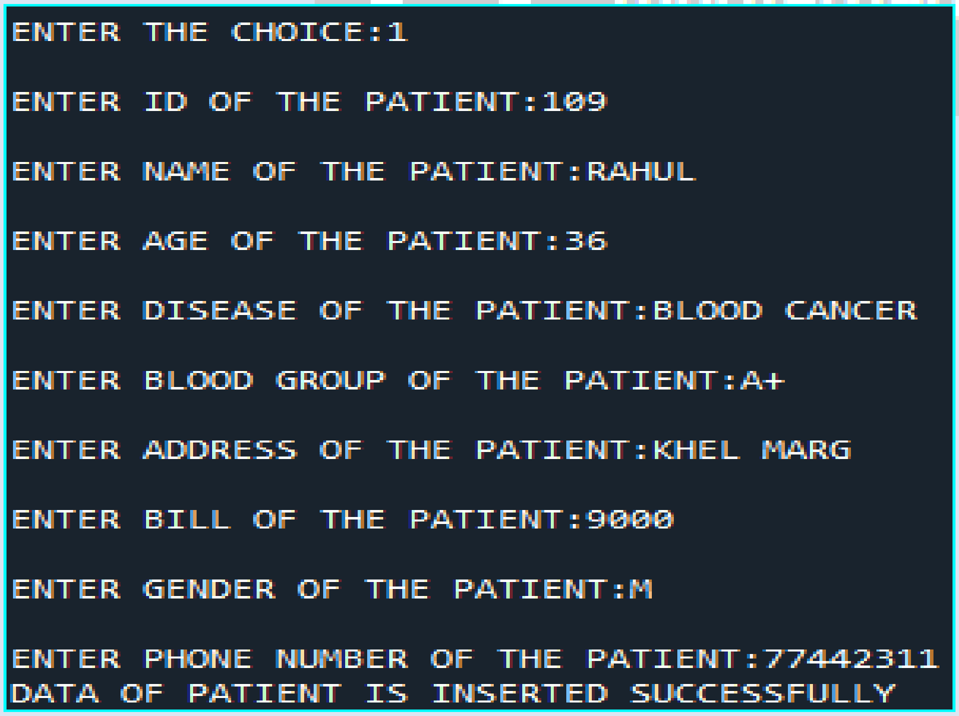
```
SUCCESSFULLY CONNECTED
HOSPITAL MANAGEMENT SYSTEM
1.LOGIN
2.EXIT

ENTER YOUR CHOICE:1

ENTER YOUR USERNAME=admin

ENTER YOUR PASSWORD=admin@123
CONNECTED SUCCESSFULLY TO HOSPITAL MANAGEMENT SYSTEM
```

4. IF THE USER IS CONNECTED SUCCESSFULLY TO HOSPITAL DATABASE MANAGEMENT SYSTEM, THEN THE USER CAN PERFORM THE FOLLOWING FUNCTIONS.



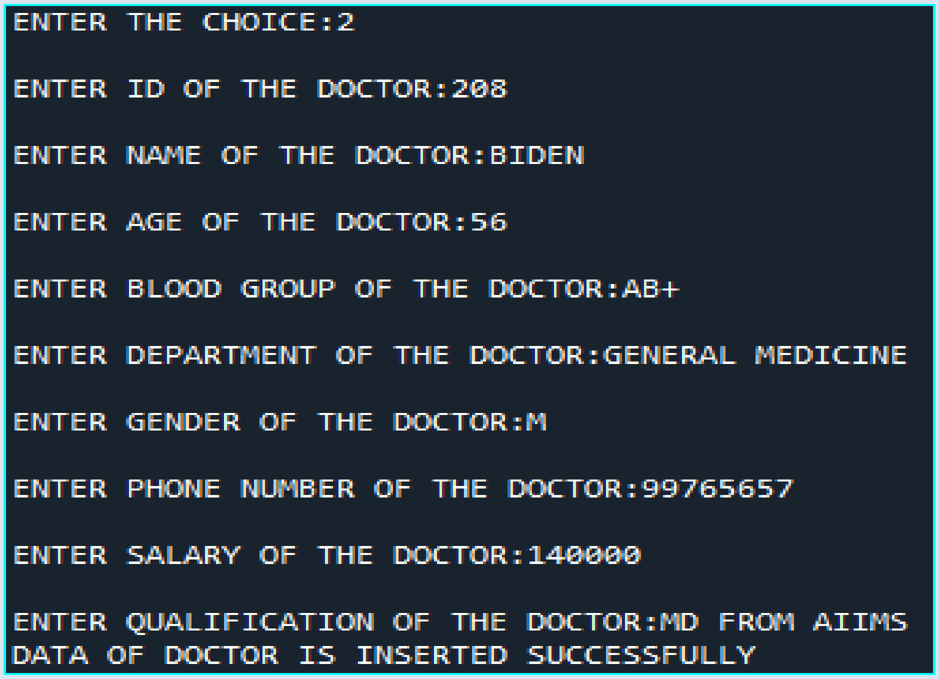
```
1.REGISTER PATIENT DETAILS
2.REGISTER DOCTOR DETAILS
3.REGISTER STAFF DETAILS
4.VIEW REGISTERED PATIENT DETAILS
5.VIEW REGISTERED DOCTOR DETAILS
6.VIEW REGISTERED STAFF DETAILS
7.DELETE SPECIFIC PATIENT DETAILS
8.DELETE SPECIFIC DOCTOR DETAILS
9.DELETE SPECIFIC STAFF DETAILS
10.UPDATE PATIENT PHONENO
11.UPDATE PATIENT BILL
12.UPDATE PATIENT AGE
13.UPDATE DOCTOR PHONE NUMBER
14.UPDATE DOCTOR AGE
15.UPDATE STAFF PHONE NUMBER
16.UPDATE STAFF SALARY
17.UPDATE STAFF AGE
18.LINE CHART FOR BILL OF PATIENTS
19.SCATTER CHART FOR BILL OF PATIENTS
20.BAR CHART FOR BILL OF PATIENTS
21.HORIZONTAL BAR CHART FOR BILL OF PATIENTS
22.LINE CHART FOR SALARY OF DOCTORS
23.SCATTER CHART FOR SALARY OF DOCTORS
24.BAR CHART FOR SALARY OF DOCTORS
25.HORIZONTAL BAR CHART FOR SALARY OF DOCTORS
26.LINE CHART FOR SALARY OF STAFF
27.SCATTER CHART FOR SALARY OF STAFF
28.BAR CHART FOR SALARY OF STAFF
29.HORIZONTAL BAR CHART FOR SALARY OF STAFF
30.EXIT

ENTER THE CHOICE:
```

- *FOR CHOICE 1, USER CAN REGISTER PATIENT DETAILS.*



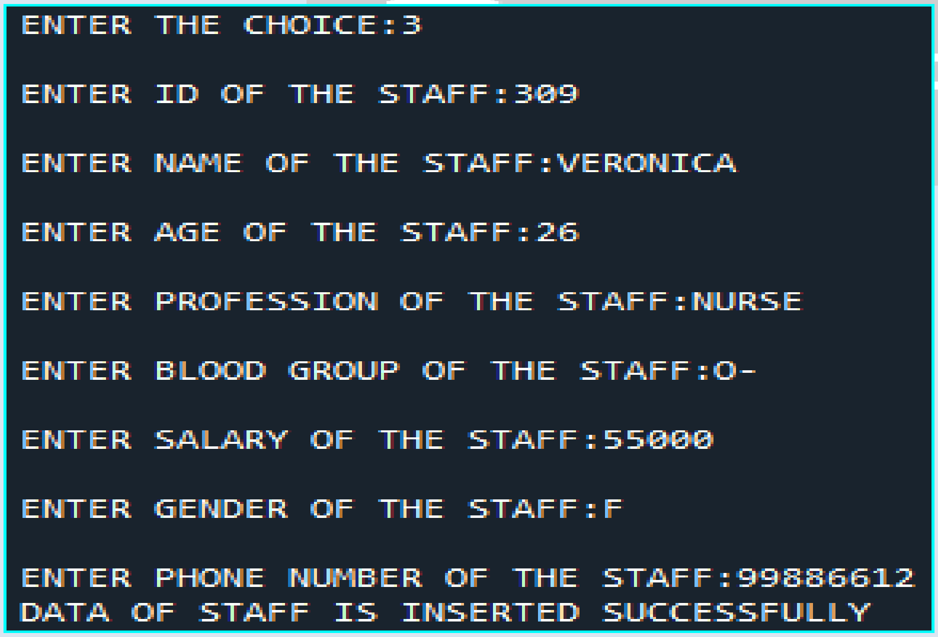
```
ENTER THE CHOICE:1

ENTER ID OF THE PATIENT:109

ENTER NAME OF THE PATIENT:RAHUL

ENTER AGE OF THE PATIENT:36

ENTER DISEASE OF THE PATIENT:BLOOD CANCER

ENTER BLOOD GROUP OF THE PATIENT:A+

ENTER ADDRESS OF THE PATIENT:KHEL MARG

ENTER BILL OF THE PATIENT:9000

ENTER GENDER OF THE PATIENT:M

ENTER PHONE NUMBER OF THE PATIENT:77442311
DATA OF PATIENT IS INSERTED SUCCESSFULLY
```

- ***FOR CHOICE 2, USER CAN REGISTER DOCTOR DETAILS.***



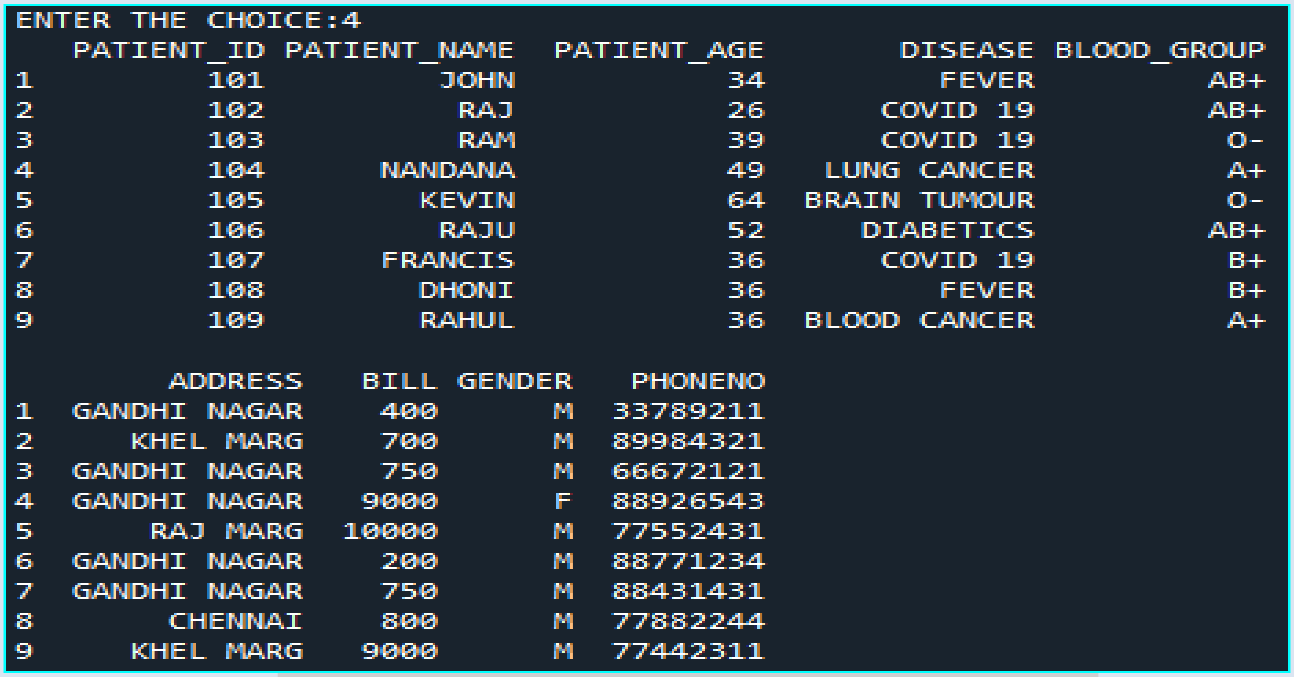
```
ENTER THE CHOICE:2

ENTER ID OF THE DOCTOR:208

ENTER NAME OF THE DOCTOR:BIDEN

ENTER AGE OF THE DOCTOR:56

ENTER BLOOD GROUP OF THE DOCTOR:AB+

ENTER DEPARTMENT OF THE DOCTOR:GENERAL MEDICINE

ENTER GENDER OF THE DOCTOR:M

ENTER PHONE NUMBER OF THE DOCTOR:99765657

ENTER SALARY OF THE DOCTOR:140000

ENTER QUALIFICATION OF THE DOCTOR:MD FROM AIIMS
DATA OF DOCTOR IS INSERTED SUCCESSFULLY
```

- ***FOR CHOICE 3, USER CAN REGISTER STAFF DETAILS.***



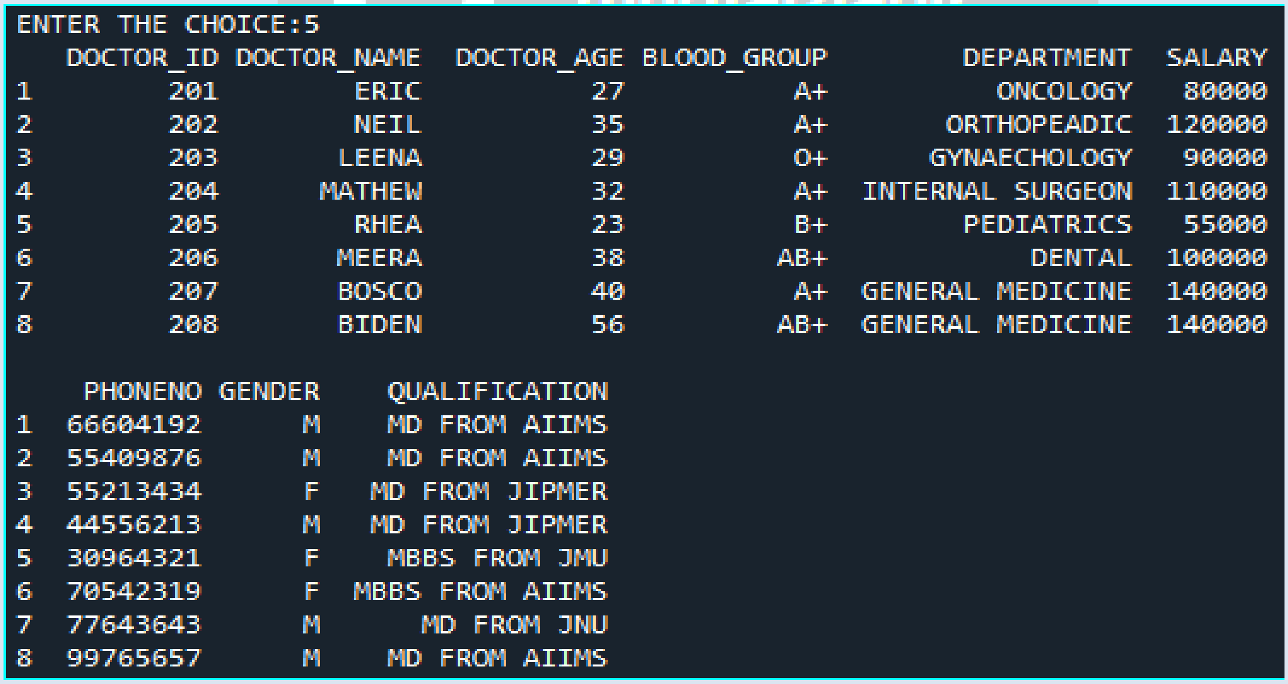
```
ENTER THE CHOICE:3

ENTER ID OF THE STAFF:309

ENTER NAME OF THE STAFF:VERONICA

ENTER AGE OF THE STAFF:26

ENTER PROFESSION OF THE STAFF:NURSE

ENTER BLOOD GROUP OF THE STAFF:O-

ENTER SALARY OF THE STAFF:55000

ENTER GENDER OF THE STAFF:F

ENTER PHONE NUMBER OF THE STAFF:99886612
DATA OF STAFF IS INSERTED SUCCESSFULLY
```

- ***FOR CHOICE 4, USER CAN VIEW PATIENT DETAILS.***



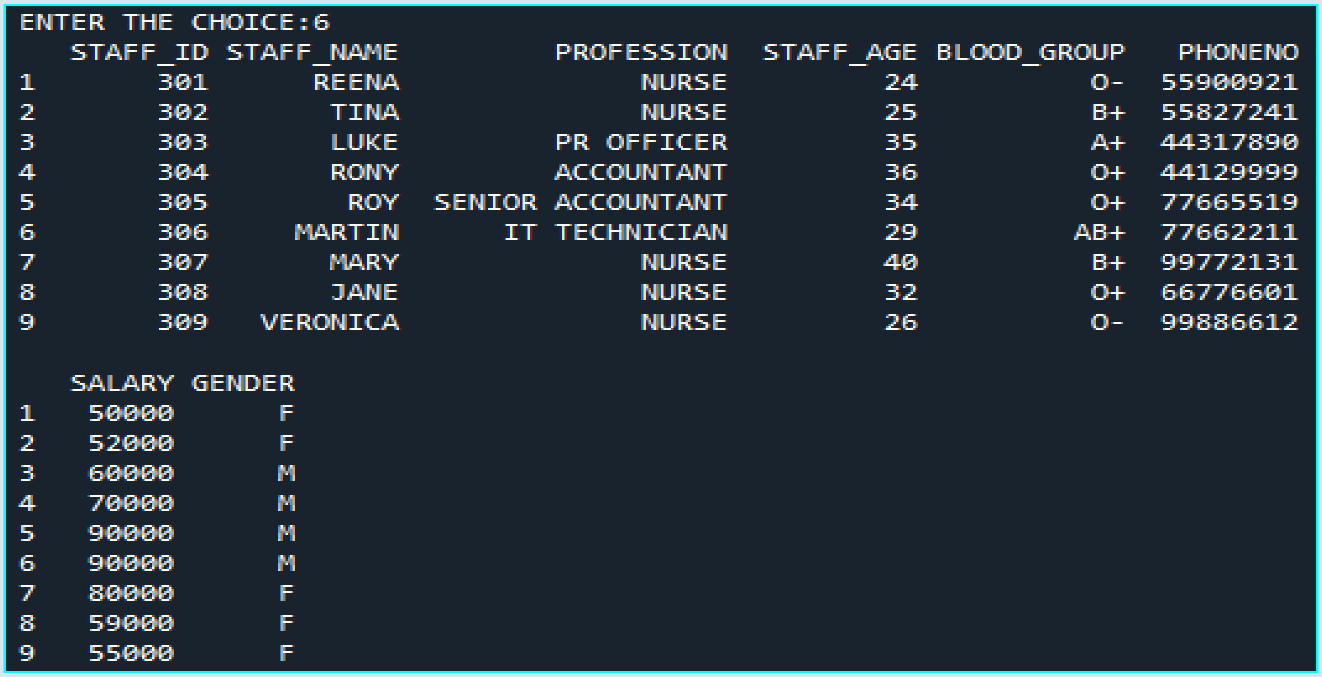
```
ENTER THE CHOICE:4
   PATIENT_ID PATIENT_NAME  PATIENT_AGE       DISEASE BLOOD_GROUP
1         101         JOHN           34         FEVER         AB+
2         102          RAJ           26      COVID 19         AB+
3         103          RAM           39      COVID 19          O-
4         104      NANDANA           49   LUNG CANCER          A+
5         105        KEVIN           64  BRAIN TUMOUR          O-
6         106         RAJU           52     DIABETICS         AB+
7         107      FRANCIS           36      COVID 19          B+
8         108        DHONI           36         FEVER          B+
9         109        RAHUL           36  BLOOD CANCER          A+

        ADDRESS   BILL GENDER   PHONENO
1  GANDHI NAGAR    400      M  33789211
2     KHEL MARG    700      M  89984321
3  GANDHI NAGAR    750      M  66672121
4  GANDHI NAGAR   9000      F  88926543
5      RAJ MARG  10000      M  77552431
6  GANDHI NAGAR    200      M  88771234
7  GANDHI NAGAR    750      M  88431431
8       CHENNAI    800      M  77882244
9     KHEL MARG   9000      M  77442311
```

- ***FOR CHOICE 5, USER CAN VIEW DOCTOR DETAILS.***



```
ENTER THE CHOICE:5
   DOCTOR_ID DOCTOR_NAME  DOCTOR_AGE BLOOD_GROUP        DEPARTMENT  SALARY
1        201        ERIC          27          A+          ONCOLOGY   80000
2        202        NEIL          35          A+       ORTHOPEADIC  120000
3        203       LEENA          29          O+      GYNAECHOLOGY   90000
4        204      MATHEW          32          A+  INTERNAL SURGEON  110000
5        205        RHEA          23          B+        PEDIATRICS   55000
6        206       MEERA          38         AB+            DENTAL  100000
7        207       BOSCO          40          A+  GENERAL MEDICINE  140000
8        208       BIDEN          56         AB+  GENERAL MEDICINE  140000

    PHONENO GENDER    QUALIFICATION
1  66604192      M    MD FROM AIIMS
2  55409876      M    MD FROM AIIMS
3  55213434      F   MD FROM JIPMER
4  44556213      M   MD FROM JIPMER
5  30964321      F    MBBS FROM JMU
6  70542319      F  MBBS FROM AIIMS
7  77643643      M      MD FROM JNU
8  99765657      M    MD FROM AIIMS
```

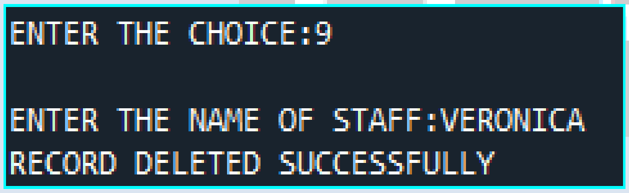
- ***FOR CHOICE 6, USER CAN VIEW STAFF DETAILS***



```
ENTER THE CHOICE:6
   STAFF_ID STAFF_NAME             PROFESSION  STAFF_AGE  BLOOD_GROUP   PHONENO
1       301      REENA                  NURSE         24           O-  55900921
2       302       TINA                  NURSE         25           B+  55827241
3       303       LUKE             PR OFFICER         35           A+  44317890
4       304       RONY             ACCOUNTANT         36           O+  44129999
5       305        ROY      SENIOR ACCOUNTANT         34           O+  77665519
6       306     MARTIN          IT TECHNICIAN         29          AB+  77662211
7       307       MARY                  NURSE         40           B+  99772131
8       308       JANE                  NURSE         32           O+  66776601
9       309   VERONICA                  NURSE         26           O-  99886612

   SALARY GENDER
1   50000      F
2   52000      F
3   60000      M
4   70000      M
5   90000      M
6   90000      M
7   80000      F
8   59000      F
9   55000      F
```

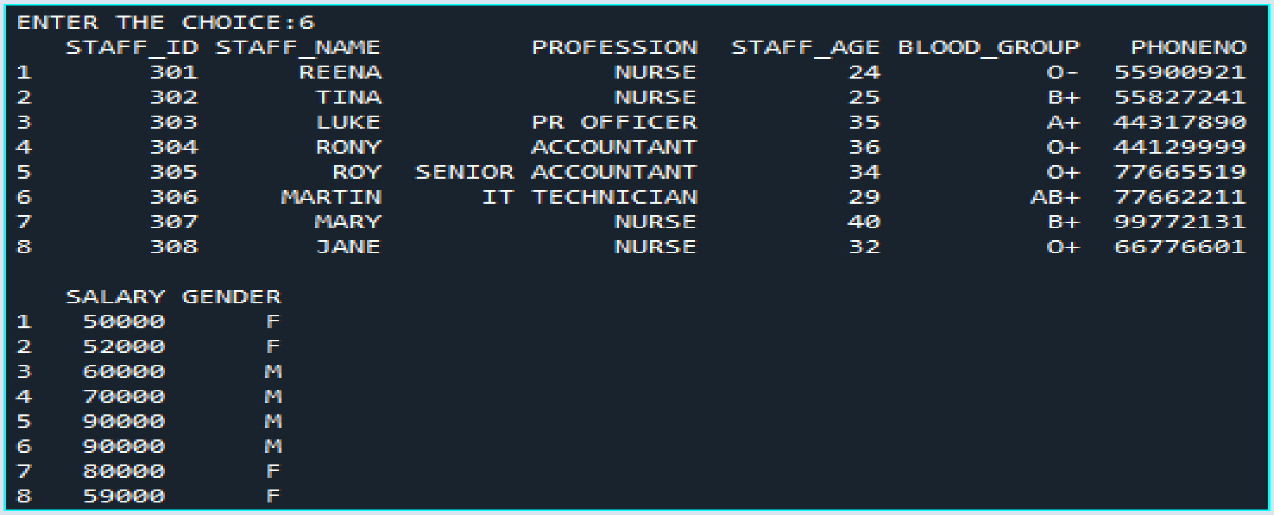
- ***TO DELETE AND UPDATE THE DATA FROM THE DATABASE, SIMILAR PROCEDURE IS FOLLOWED.***

FOR EXAMPLE, IF ENTERED CHOICE IS 9, IT WILL DELETE THE SPECIFIC STAFF DETAILS.



```
ENTER THE CHOICE:9

ENTER THE NAME OF STAFF:VERONICA
RECORD DELETED SUCCESSFULLY
```

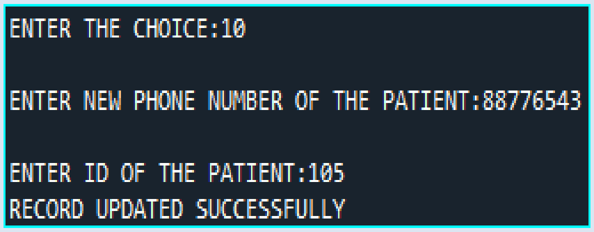
TO CHECK WHETHER THE DATA IS DELETED OR NOT, ENTER THE CHOICE 6 TO VIEW THE STAFF DETAILS.



```
ENTER THE CHOICE:6
   STAFF_ID STAFF_NAME             PROFESSION  STAFF_AGE  BLOOD_GROUP   PHONENO
1       301      REENA                  NURSE         24           O-  55900921
2       302       TINA                  NURSE         25           B+  55827241
3       303       LUKE             PR OFFICER         35           A+  44317890
4       304       RONY             ACCOUNTANT         36           O+  44129999
5       305        ROY      SENIOR ACCOUNTANT         34           O+  77665519
6       306     MARTIN          IT TECHNICIAN         29          AB+  77662211
7       307       MARY                  NURSE         40           B+  99772131
8       308       JANE                  NURSE         32           O+  66776601

   SALARY GENDER
1   50000      F
2   52000      F
3   60000      M
4   70000      M
5   90000      M
6   90000      M
7   80000      F
8   59000      F
```

DATA WITH NAME AS VERONICA IS DELETED SUCCESSFULLY.

SIMILARY, UPDATE OPERATION IS EXECUTED. FOR EXAMPLE, TO UPDATE PHONE NUMBER OF PATIENT, USER CAN ENTER CHOICE 10.

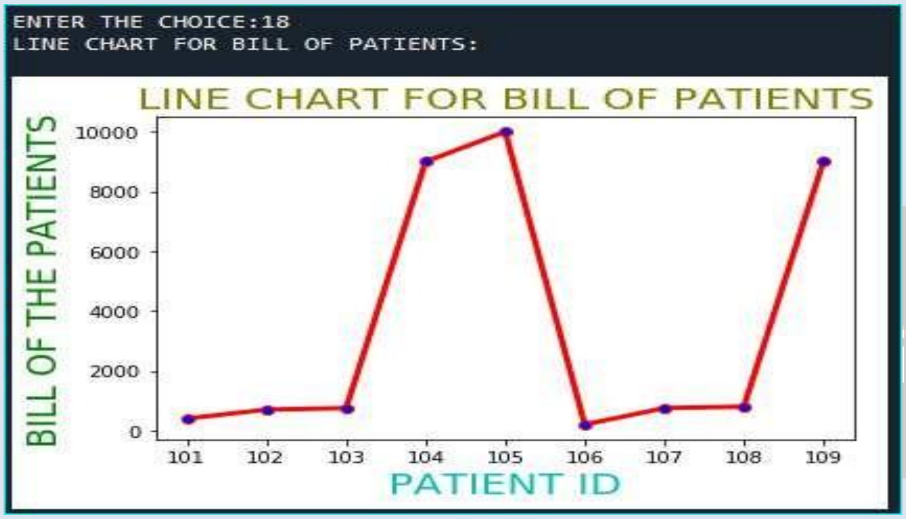


```
ENTER THE CHOICE:10

ENTER NEW PHONE NUMBER OF THE PATIENT:88776543

ENTER ID OF THE PATIENT:105
RECORD UPDATED SUCCESSFULLY
```

- ***TO PLOT DATA FROM THE DATA STORED IN DATABASE, USER CAN ENTER THE CHOICES FROM 18 TO 29 (BOTH INCLUSIVE).***

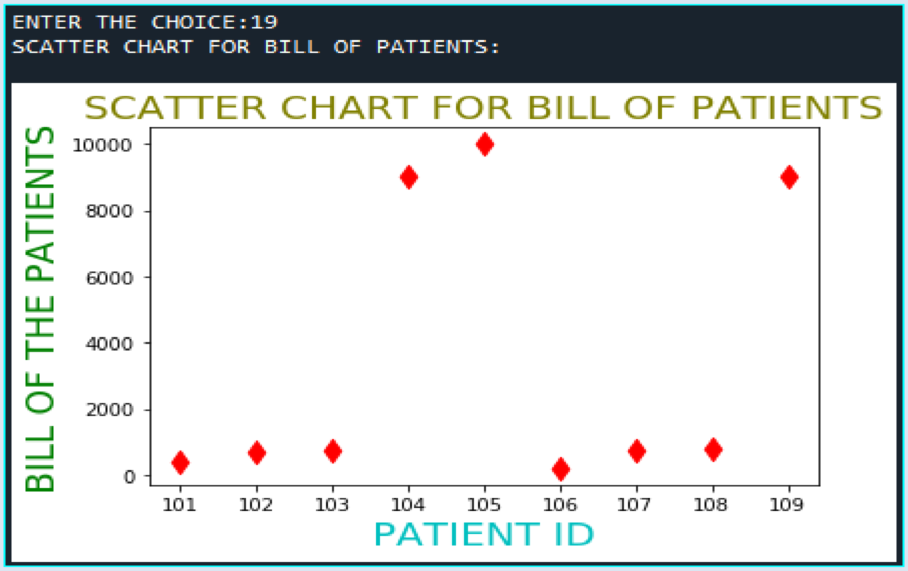
EXAMPLES:

WHEN CHOICE IS 18, LINE CHART FOR BILL OF PATIENTS IS PLOTTED.



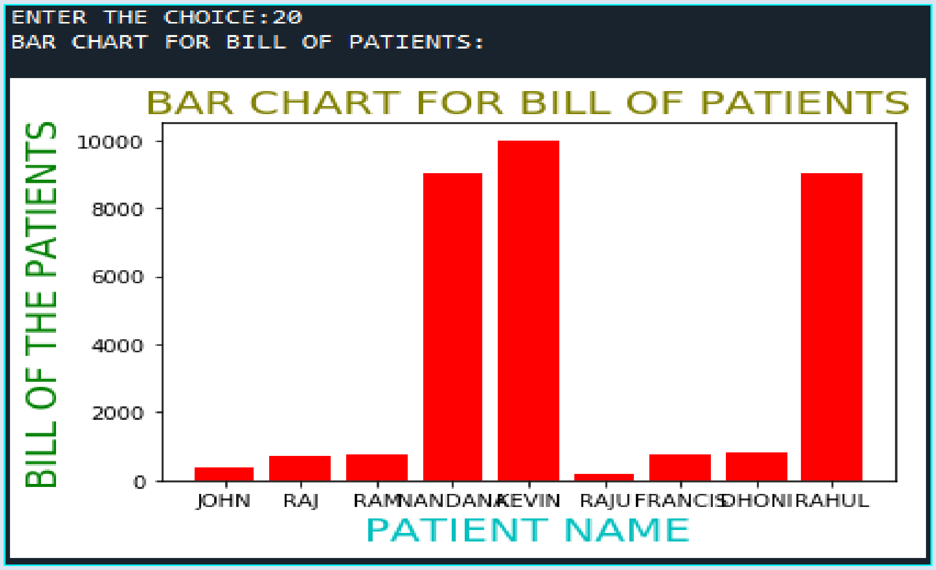
```
ENTER THE CHOICE:18
LINE CHART FOR BILL OF PATIENTS:
```

WHEN CHOICE IS 19, SCATTER CHART FOR BILL OF PATIENTS IS PLOTTED.



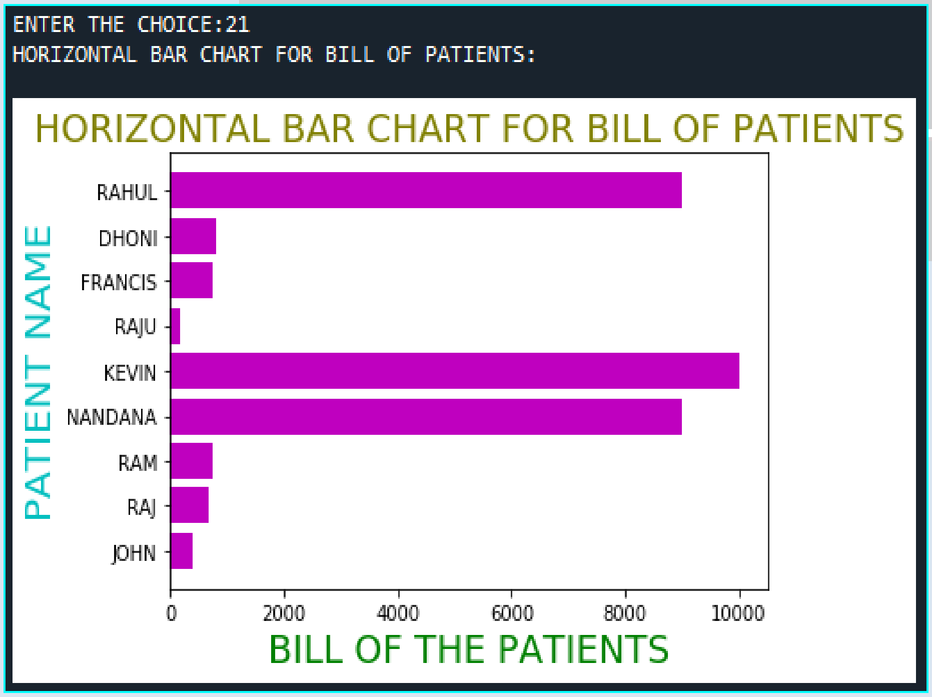
```
ENTER THE CHOICE:19
SCATTER CHART FOR BILL OF PATIENTS:
```

WHEN CHOICE IS 20, BAR CHART FOR BILL OF PATIENTS IS PLOTTED.



```
ENTER THE CHOICE:20
BAR CHART FOR BILL OF PATIENTS:
```

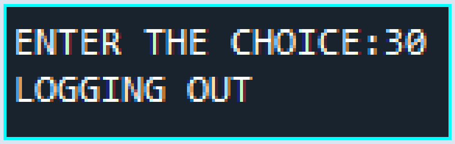
WHEN CHOICE IS 21, HORIZONTAL BAR CHART FOR BILL OF PATIENTS IS PLOTTED.



```
ENTER THE CHOICE:21
HORIZONTAL BAR CHART FOR BILL OF PATIENTS:
```

SIMILARLY, DATA OF DOCTORS AND STAFF CAN BE PLOTTED BY ENTERING THE REQUIRED CHOICES.

- ***IF THE ENTERED CHOICE IS 30, USER CAN LOG OUT FROM THE HOSPITAL DATABASE MANAGEMENT SYSTEM.***



```
ENTER THE CHOICE:30
LOGGING OUT
```

--------------------------<u>END OF PROGRAM</u>-----------------------

CONCLUSION:-

The project titled, “HOSPITAL DATABASE MANAGEMENT SYSTEM” is executed successfully.

MERITS OF THE PROJECT:-

1. *RECORDS CAN BE STORED EASILY*
2. *RECORDS CAN BE ACCESSED EASILY*
3. *RECORDS CAN BE UPDATED EASILY*
4. *RECORDS CAN BE DELETED EASILY*
5. *REDUCES MANUAL WORK*
6. *SECURED DATABASE*
7. *EASY MAINTANENACE OF DATABASE*
8. *COST EFFECTIVE*
9. *USER FRIENDLY*

****