**HOSPITAL MANAGEMENT**

TABLE OF CONTENT:

1. ABSTRACT
2. SYSTEMCONFIGURATION
3. INTRODUCTION

(PYTHON MYSQL)

1. RELATION BETWEEN PYTHON ANDMYSQL
2. DATABASECONNECTIVITY
3. LIBRARIES AND FUNCTIONSUSED
4. MYSQLQUERIESUSEDINTHISPROJECT
5. SOURCECODE
6. OUTPUT(SCREENSHOT)
7. CONCLUSION
8. BIBILOGRAPHY

**ABSTRACT:**

Hospital management system project is developed using Python and MYSQL Database.This application provides an easy way for the receptionist in adding,updating,searching the details of the Patient,Doctor,Ambulance Driver.Hospital management systemin Python is a simple console application.In this project,users can perform typical management related function like adding a new doctor,patient and Ambulance Driver,displaying,modifying and editing it

**SYSTEM CONFIGURATION:**

#### HARDWARE CONFIGURATION

##### Microsoft windows 7 professional/windows 8/windows 8.2:

* **Processor :** Intel Core i3 orequivalent
* **Memory :** 2 GB (32-bit), 4 GB(64-bit)
* **Disk space :** 1.5 GB of free diskspace

#### SOFTWARE REQUIREMENTS

* 1 GB RAM (2 GB+recommended)
* 9-58 GB free hard disk space depending on edition and configuration, including space required for temporaryfiles
* DVD-ROM drive (if installing from a Media KitDVD)
* Basic GPU – Any vendor DirectX 9.0 class or better (Pixel Shader Level2)
* Intel® Pentium® or compatible, 1.6 GHz minimum (2GHz+recommended)
* 1024x768 or higher-resolutionmonitor
* MOUSE OR OTHER POINTINGDEVICE

## INTRODUCTION:

### PYTHON:

Python is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt.sources newsgroup in 1991, and version 1.0 Was released in1994.

Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been backported to Python 2. But in general, they remain not quite compatible.

Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End of life of January 1,2020has been established for Python 2, after which time it will no longer be maintained. If you are a newcomer to Python, it is recommended that you focus on Python 3, as this tutorial will do.

Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe [MontyPython’sFlyingCircus](https://en.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus),ofwhichGuidowas,andpresumablystillis,afan.Itiscommonto find references to Monty Python sketches and movies scattered throughout the Python documentation.

### Python is Popular

Python has been growing in popularity over the last few years. The 2018 Stack Overflow Developer Survey ranked Python as the 7th most popular and the number one most wanted technology of the year. World-class software development countries around the globe use Python every single day.According to research by Dice Python is also one of the hottest skills to have and the most popular programming language in the world based on the Popularity of programming Language Index.

### 

### Python is interpreted

Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer’s processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.

Thismakesforaquickerdevelopmentcyclebecauseyoujusttypeinyourcodeandrunit,without the intermediate compilationstep.

One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.

In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

### Python is Free

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes.

A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smartphones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

# 

# Python is Portable

# Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

# Python is Simple

# As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way.A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect.

# MYSQL:

##### Database Management System & Types of DBMS:

A **Database Management System**(**DBMS**) is a software application that interacts with the user, applications and the database itself to capture and analyze data.The data stored in the database can be modified, retrieved and deleted, and can be of any type like strings, numbers, imagesetc.

**Types of DBMS:**

There are mainly 4 types of DBMS, which are Hierarchical, Relational, Network, and Object-Oriented DBMS.

Hierarchical DBMS:As the name suggests, this type of DBMS has a style of predecessor- successor type of relationship. So, it has a structure similar to that of a tree, wherein the nodes represent records and the branches of the tree represent fields.

Relational DBMS (RDBMS): This type of DBMS, uses a structure that allows the users to identify and access data *in relation* to another piece of data in the database.

Network DBMS: This type of DBMS supports many to many relations wherein multiple member records can be linked.

Object-oriented DBMS: This type of DBMS uses small individual software called objects. Each object contains a piece of data, and the instructions for the actions to be done with thedata.

##### Structured Query Language (SQL)

SQL is the core of a relational database which is used for accessing and managing the database. By using SQL, you can add, update or delete rows of data, retrieve subsets of information,modify databases and perform many actions. The different subsets of SQL are as follows:

* + - **DDL** *(Data Definition Language) –*It allows you to perform various operations on the database such as CREATE, ALTER and DELETEobjects.
    - **DML** *(Data Manipulation Language)*– It allows you to access and manipulate data. It helps you to insert, update, delete and retrieve data from thedatabase.
    - **DCL** *(Data Control Language)*– It allows you to control access to the

database. Example – Grant or Revoke accesspermissions.

* + - **TCL***(Transaction Control Language)* – It allows you to deal with the transaction of the database. Example – Commit, Rollback, Savepoint, SetTransaction.

##### MySQL & its Features

**MySQL** is an open-source relational database management system that works on many platforms. It provides multi-user access to support many storage engines and is backed by Oracle. So,you can buy a commercial license version from Oracle to get premium support services.

The features of MySQL are as follows:

**Ease of Management –**The software very easily gets downloaded and also uses an event scheduler to schedule the tasks automatically.

* + - **Robust Transactional Support –**Holds the ACID (Atomicity, Consistency, Isolation, Durability) property, and also allows distributed multi-versionsupport.
    - **Comprehensive Application Development –**MySQL has plugin libraries to embed the database into any application. It also supports stored procedures, triggers, functions, views and many more for application development. You can refer to the[**RDS Tutorial**](https://www.edureka.co/blog/rds-aws-tutorial/), to understand Amazon’sRDBMS.
    - **High Performance –**Provides fast load utilities with distinct memory caches and table indexpartitioning.
    - **Low Total Cost Of Ownership –**This reduces licensing costs and hardware expenditures.
    - **Open Source & 24 \* 7 Support –**This RDBMS can be used on any platform and offers 24\*7 support for open source and enterpriseedition.
    - **Secure Data Protection –**MySQL supports powerful mechanisms to ensure that only authorized users have access to thedatabases.
    - **High Availability –**MySQL can run high-speed master/slave replication configurations and it offers clusterservers.
    - **Scalability & Flexibility –**With MySQL you can run deeply embedded applications and create data warehouses holding a humongous amount ofdata.

##### MySQL Data Types

* + - **Numeric –** This data type includes integers of various sizes, floating-point(real) of various precisions and formattednumbers.
    - **Character-string –** These data types either have a fixed, or a varying number of characters. This data type also has a variable-length string called *CHARACTER LARGE OBJECT*(*CLOB*) which is used to specify columns that have large text values.
    - **Bit-string –** These data types are either of a fixed length or varying length of bits. There is also a variable-length bit string data type called *BINARY LARGE OBJECT(BLOB),* which is available to specify columns that have large binary values, such asimages.
    - **Boolean –**This data type has TRUE or FALSE values. Since SQL, has NULL values, a three-valued logic is used, which isUNKNOWN.
    - **Date & Time –**The DATE data type has: YEAR, MONTH, and DAY in the form YYYY-MM-DD. Similarly, the TIME data type has the components HOUR, MINUTE, and SECOND in the form HH:MM: SS. These formats can change based on therequirement.
    - **Timestamp & Interval –**The TIMESTAMP data type includes a minimum of six positions, for decimal fractions of seconds and an optional WITH TIME ZONE qualifier in addition to the DATE and TIME fields. The INTERVAL data type mentions a relative value that can be used to increment or decrement an absolute value of a date, time, ortimestamp.

##### Python MySQL DatabaseConnection:

**Arguments required to connect MySQL from Python**

You need to know the following detail of the MySQL server to perform the connection from Python.

* + - **Username** – i.e., the username that you use to work with MySQL Server. The default username for the MySQL database is a**root**
    - **Password** – Password is given by the user at the time of installing the MySQL database. If you are using root then you won’t need thepassword.
    - **Host Name** – is the server name or Ip address on which MySQL is running. if you are running on localhost, then you can use localhost, or it’s IP, i.e.127.0.0.0
    - **Database Name** – Database name to which you want toconnect.

##### 

##### READ Operation

READ Operation on any database means to fetch some useful information from the database.

Once our database connection is established, you are ready to make a query into this database. You can use either **fetchone()** method to fetch single record or **fetchall()** method to fetech multiple values from a database table.

* + - * **fetchone()** − It fetches the next row of a query result set. A result set is an object that is returned when a cursor object is used to query atable.
      * **fetchall()** − It fetches all the rows in a result set. If some rows have already been extracted from the result set, then it retrieves the remaining rows from the result set.

##### DATABASE CONNECTIVITY:

**Steps to connect MySQL database in Python using MySQL Connector Python**

1. [Install MySQL Connector Python usingpip](https://pynative.com/install-mysql-connector-python/).
2. Use the **mysql.connector.connect()**method of MySQL Connector Python with required parameters to connectMySQL.
3. Use the connection object returned by a connect()method to create a **cursor** object to perform DatabaseOperations.
4. The **cursor.execute()**to execute SQL queries fromPython.
5. Close the Cursor object using a **cursor.close()** and MySQL database connection using **connection.close()** after your workcompletes.
6. Catch Exception if any that may occur during thisprocess.

**LIBRARIES AND FUNCTIONS USED:**

1. **Mysql.connector**

**FUNCTIONS USED: WRITE IN DEATAIL ABOUT PURPOSE OF THE FUNCTION.**

* 1. **TITLE()-USED TO DISPLAY ALL DETAILS OR TITLES WHAT WE DOING.**

**MYSQL QUERIES USED IN THIS PROJECT:**

* + 1. create databasestudent;
    2. use project
    3. create table patiebt\_details(Patient\_entryno int not null,Patient\_name varchar(100),Patient\_age int not null,Patient\_problem varchar(100) ,Doctor\_recommended varchar(100) not null,Patient\_ph big int,Patient\_roomno bigint);

4. create table doctor\_details(Doctor\_entryno int not null,Doctor\_name varchar(100) not null,Doctor\_age int,Doctor\_graduation varchar(100) not null,Doctor\_department varchar(100) not null,Doctor\_Salary bigint,Doctor\_ph bigint,Doctor\_available varchar(10) not null,Doctor\_roomno int);

1. create table worker\_details(Worker\_entryno int not,Worker\_name varchar(100)not null,Worker\_age int not null,Worker\_graduation varchar(100) not null,Worker\_WorkName varchar(100) not null,Worker\_Salary bigint not null,Worker\_Roomno int,Worker\_ph bigint not null);
2. create table driver details(Driver\_entryno int not null,Driver\_name varchar(100) ,Driver\_age int ,Driver\_graduation varchar(100) ,Driver\_Salary bigint,Driver\_Licensesno big int,Driver\_ph bigint,Driver\_Available varchar(10));
3. insert into patient\_details values();
4. insert into doctor\_details values();
5. insert into worker\_details values();

10.insert into driver\_dettails values();

11. select \* from patient\_details;

12. select \* from doctor\_details;

13. select \* from Worker\_details;

14. select \* from patient\_details where patient\_entryno={};

15. select \* from doctor\_details where doctor\_entryno={};

16. select \* from worker\_details where worker\_entryno={};

17. select patient\_roomno 'OCCUPIED ROOM NUMBER' from patient\_details;

18.create table billing(Dates\_of\_activity varchar(100),Item\_description varchar(100),Qty int,Charges int);

19. select \* from billing where Item\_description='{}';

20. update Doctor\_details set Doctor\_salary={} where Doctor\_entryno={};

21. update Worker\_details set Worker\_salary={} where Worker\_entryno={};

22. update Patient\_details set Patient\_roomno={} where Patient\_entryno={};

23. select count(Patient\_name) 'number of patient in the hospital' from patient\_details;

24. select count(doctor\_name) 'number of doctor in the hospital' from doctor\_details;

25. select \* from doctor\_details where doctor\_available='Y';

26. insert into Driver\_details values({},'{}',{},'{}',{},{},{},'{}');

27. select \* from driver\_details where driver\_available='Y

## SOURCE CODE:

import mysql.connector as mys

con= mys.connect(host='localhost',user='root',passwd='rk2004',database='PROJECT')

if con.is\_connected():

cur=con.cursor()

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HOSPITAL MANAGEMENT SYSTEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("STAY SAFE STAY ALIVE")

print("1:LOGIN")

print("2:EXIST")

choice=int(input("Enter 1 to login:"))

if choice==1:

user\_id=input("Enter your user id:")

password=input("Enter your password:")

if user\_id=="MIOT" and password=="MIOT62":

def title():

print("You are successfully login")

print("WELCOME MIOT INTERNATIONAL HOSPITAL")

print("1.Registering Patient details")

print("2.Registering Doctor details")

print("3.Registering worker details")

print("4.Total Patient details")

print("5.Total doctor details")

print("6.Total worker details")

print("7.Searching patient details")

print("8.Searching doctor details")

print("9.Searching worker details")

print("10.Occupied room number")

print("11.Billing for medicine and room")

print("12.Avalibility of medicine")

print("13.Update doctor details")

print("14.Update worker details")

print("15.Update patient details")

print("16.Number of Patient in the hospital")

print("17.Number of doctor in the hospital")

print("18.Available of doctor")

print("19.Ambulance driver details")

print("20.Ambulance Availability")

print("21.Exist")

choice=int(input("Enter your choice:"))

if choice==1:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("WELCOME TO MIOT PATIENT DETAILS ENTRY")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Patient\_entryno=int(input("Enter entry no of patient:"))

Patient\_name=input("Enter Patient name:")

Patient\_age=int(input("Enter Patient age:"))

Patient\_problem=input("Enter Patient Problem:")

Doctor\_recommended=input("Enter recommended doctor name:")

Patient\_ph=input("Enter Patient phone number:")

Patient\_roomno=int(input("Enter Patient room no;"))

Query="insert into patient\_details values({},'{}',{},'{}','{}',{},{})".format(Patient\_entryno,Patient\_name,Patient\_age,Patient\_problem,Doctor\_recommended,Patient\_ph,Patient\_roomno)

cur.execute(Query)

con.commit()

print("successfully registered")

A=input("Do you want to continue(y/n):")

elif choice==2:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("WELCOME TO MIOT DOCTOR DETAILS ENTRY")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Doctor\_entryno=int(input("Enter entry no of doctor:"))

Doctor\_name=input("Enter Doctor name:")

Doctor\_age=int(input("Enter Doctor age:"))

Doctor\_graduation=input("Enter Doctor graduation:")

Doctor\_department=input("Enter Doctor department:")

Doctor\_Salary=input("Enter Doctor Salary:")

Doctor\_ph=input("Enter Doctor phone number:")

Doctor\_available=input("Enter Doctor is available/not available(y/n):")

Doctor\_roomno=int(input("Enter Doctor room no:"))

Query1="insert into Doctor\_details values({},'{}',{},'{}','{}','{}',{},'{}',{})".format(Doctor\_entryno,Doctor\_name,Doctor\_age,Doctor\_graduation,Doctor\_department,Doctor\_Salary,Doctor\_ph,Doctor\_available,Doctor\_roomno)

cur.execute(Query1)

con.commit()

print("successfully registered")

A=input("Do you want to continue(y/n):")

elif choice==3:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("WELCOME TO MIOT WORKER DETAILS ENTRY")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Worker\_entryno=int(input("Enter entry no of worker:"))

Worker\_name=input("Enter worker name:")

Worker\_age=int(input("Enter worker age:"))

Worker\_graduation=input("Enter worker graduation:")

Worker\_Salary=input("Enter worker Salary:")

Worker\_WorkName=input("Enter workname:")

Worker\_Roomno=int(input("Enter worker room no:"))

Worker\_ph=int(input("Enter worker phone number:"))

Query2="insert into worker\_details values({},'{}',{},'{}','{}','{}',{},{})".format(Worker\_entryno,Worker\_name,Worker\_age,Worker\_graduation,Worker\_WorkName,Worker\_Salary,Worker\_Roomno,Worker\_ph)

cur.execute(Query2)

con.commit()

print("successfully registered")

elif choice==4:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*TOTAL PATIENT DETAILS\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Query3="select \* from patient\_details"

cur.execute(Query3)

a=cur.fetchall()

for i in a:

print(a)

elif choice==5:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*TOTAL DOCTOR DETAILS\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Query4="select \* from doctor\_details"

cur.execute(Query4)

b=cur.fetchall()

for i in b:

print(b)

elif choice==6:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*TOTAL WORKERS DETAILS\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Query6="select \* from Worker\_details"

cur.execute(Query6)

c=cur.fetchall()

for i in c:

print(c)

elif choice==7:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("WELCOME TO SEARCH PATIENT DETAILS")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

entryno=int(input("Enter patient entry number:"))

Query7="select \* from patient\_details where patient\_entryno={}".format(entryno)

cur.execute(Query7)

data=cur.fetchone()

if data!=None:

print(data)

else:

print("Record is not found!!!")

elif choice==8:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("WELCOME TO SEARCH DOCTOR DETAILS")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

entryno1=int(input("Enter doctor entry number:"))

Query8="select \* from doctor\_details where doctor\_entryno={}".format(entryno1)

cur.execute(Query8)

data=cur.fetchone()

if data!=None:

print(data)

else:

print("Record is not found!!!")

elif choice==9:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("WELCOME TO SEARCH WORKERS DETAILS")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

entryno2=int(input("Enter worker entry number:"))

Query9="select \* from worker\_details where worker\_entryno={}".format(entryno2)

cur.execute(Query9)

data=cur.fetchone()

if data!=None:

print(data)

else:

print("Record is not found!!!")

elif choice==10:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("OCCUPIED ROOM NUMBER")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Query10="select patient\_roomno 'OCCUPIED ROOM NUMBER' from patient\_details"

cur.execute(Query10)

data=cur.fetchall()

if data!=None:

print(data)

else:

print("Record is not found")

elif choice==11:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*MIOT INTERNATIONAL HOSPITAL\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Dates\_of\_activity=input("Enter the date when product is buyed:")

Item\_description=input("Enter product name:")

Qty=int(input("Enter number of product buyed:"))

Charges=int(input("Enter Charge of the product:"))

query="insert into billing values('{}','{}',{},{})".format(Dates\_of\_activity,Item\_description,Qty,Charges)

cur.execute(query)

con.commit()

print("successfully registered")

elif choice==12:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*AVAILABLE OF MEDICINE\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Name=input("Enter the name of medicine:")

query1="select \* from billing where Item\_description='{}'".format(Name)

cur.execute(query1)

m=cur.fetchone()

if m!=None:

print(m)

else:

print("Record not found!!!")

elif choice==13:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*UPDATE DOCTOR DETAILS\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

salary=int(input("Enter Doctor salary do you want to update:"))

entryno=int(input("Enter doctor\_entryno:"))

query="update Doctor\_details set Doctor\_salary={} where Doctor\_entryno={}".format(salary,entryno)

cur.execute(query)

con.commit()

elif choice==14:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*UPDATE WORKER DETAILS\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

salary=int(input("Enter Worker salary do you want to update:"))

entryno=int(input("Enter Worker\_entryno:"))

query="update Worker\_details set Worker\_salary={} where Worker\_entryno={}".format(salary,entryno)

cur.execute(query)

con.commit()

elif choice==15:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*UPDATE PATIENT DETAILS\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

no=int(input("Enter Patient room no do you want to update:"))

entryno=int(input("Enter Patient\_entryno:"))

query="update Patient\_details set Patient\_roomno={} where Patient\_entryno={}".format(no,entryno)

cur.execute(query)

con.commit()

elif choice==16:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*Number of Patient in the hospital\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

query="select count(Patient\_name) 'number of patient in the hospital' from patient\_details"

cur.execute(query)

data=cur.fetchall()

for i in data:

print(i)

elif choice==17:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*Number of Doctor in the hospital\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

query="select count(doctor\_name) 'number of doctor in the hospital' from doctor\_details"

cur.execute(query)

data=cur.fetchall()

for i in data:

print(i)

elif choice==18:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*Available of Doctor in the hospital\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

query="select \* from doctor\_details where doctor\_available='Y' "

cur.execute(query)

data=cur.fetchall()

for i in data:

print(i)

elif choice==19:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("WELCOME TO MIOT DRIVER DETAILS ENTRY")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Driver\_entryno=int(input("Enter entry no of Driver:"))

Driver\_name=input("Enter Driver name:")

Driver\_age=int(input("Enter Driver age:"))

Driver\_graduation=input("Enter Driver graduation:")

Driver\_Salary=input("Enter Driver Salary:")

Driver\_Licensesno=input("Enter Driver\_Licensesno:")

Driver\_ph=int(input("Enter worker phone number:"))

Driver\_Available=input("Enter Driver Available:")

Query2="insert into Driver\_details values({},'{}',{},'{}',{},{},{},'{}')".format(Driver\_entryno,Driver\_name,Driver\_age,Driver\_graduation,Driver\_Salary,Driver\_Licensesno,Driver\_ph,Driver\_Available)

cur.execute(Query2)

con.commit()

print("successfully registered")

elif choice==20:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("\*\*\*Available of Ambulance in the hospital\*\*\*\*\*\*\*\*\*")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

query="select \* from driver\_details where driver\_available='Y' "

cur.execute(query)

data=cur.fetchall()

for i in data:

print(i)

elif choice==21:

print("\*\*\*\*\*\*\*\*\*")

print("-EXIST-")

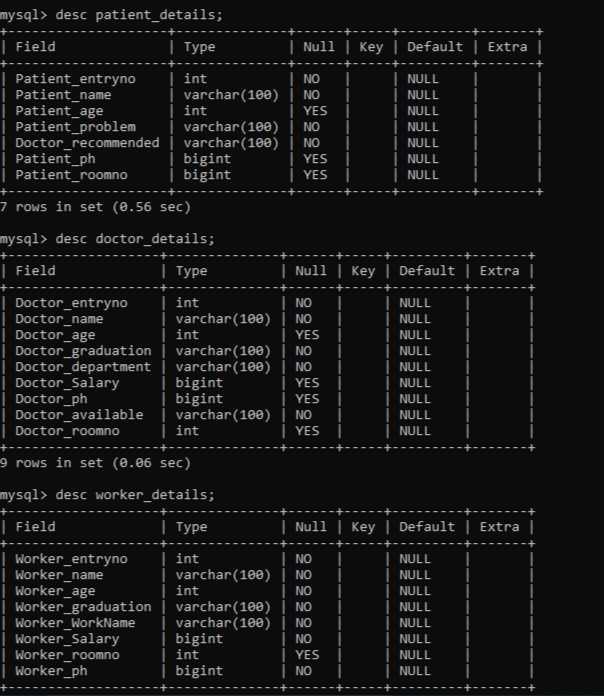
print("\*\*\*\*\*\*\*\*\*")

else:

print("Enter correct choice")

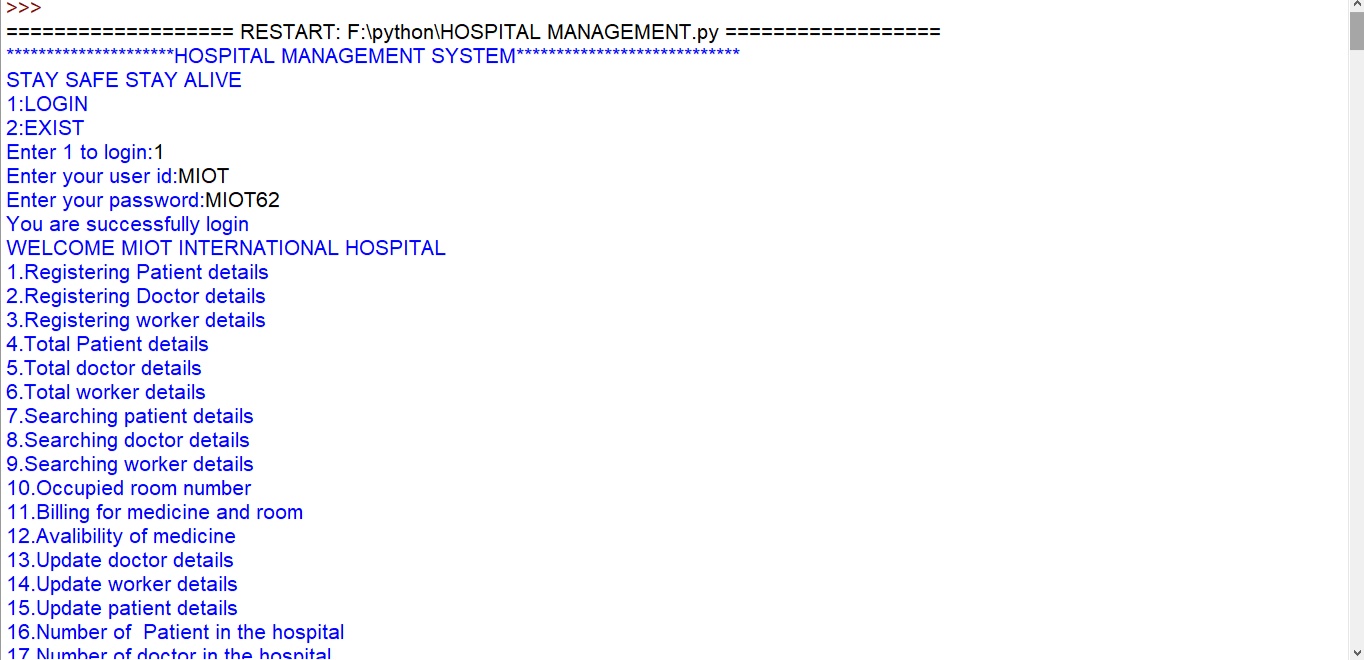
title()

Con.close()



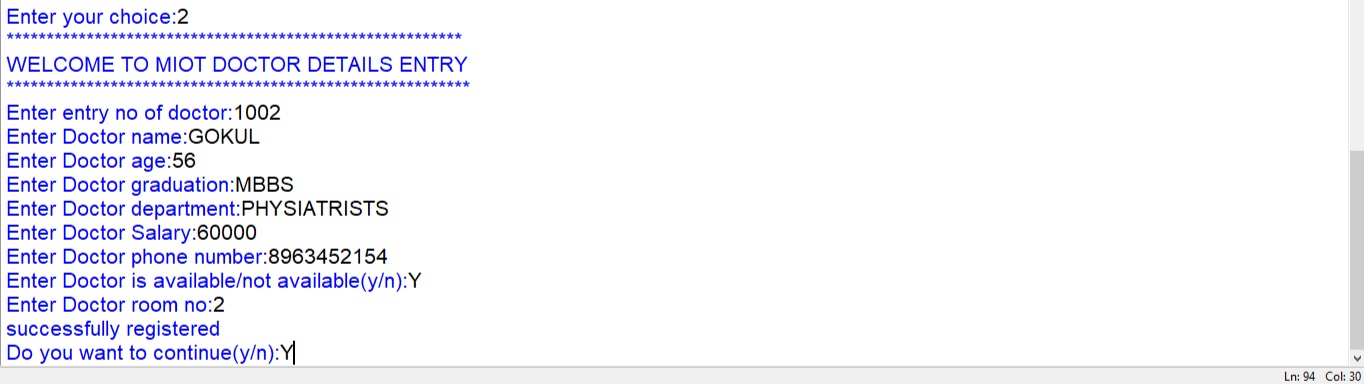
## sql2.jpg

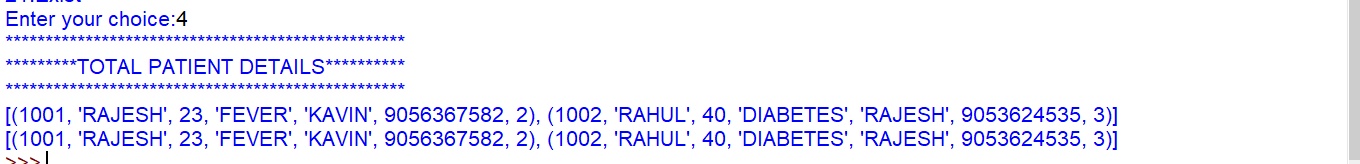
**OUTPUT:**

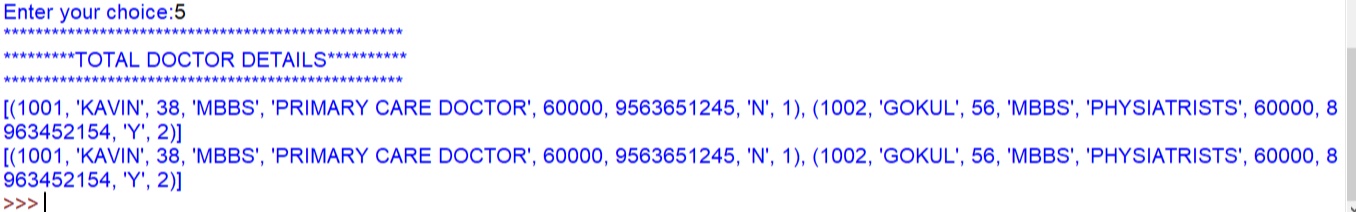
****

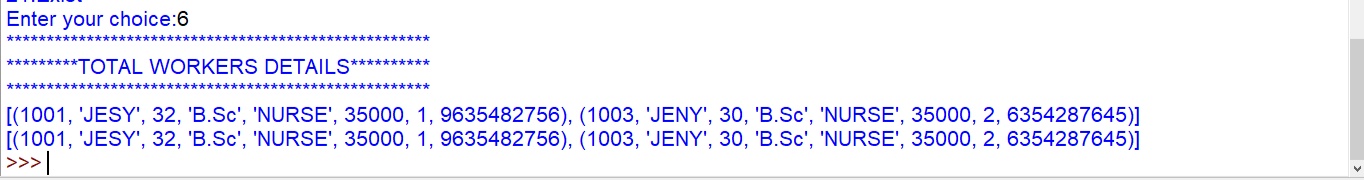




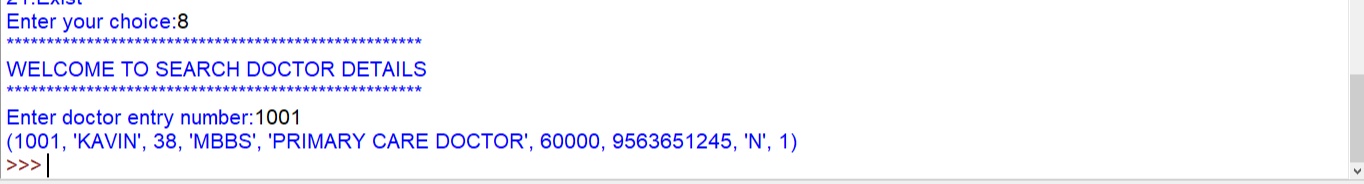








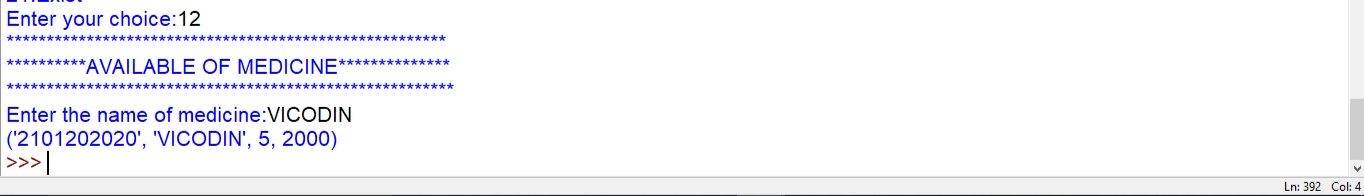












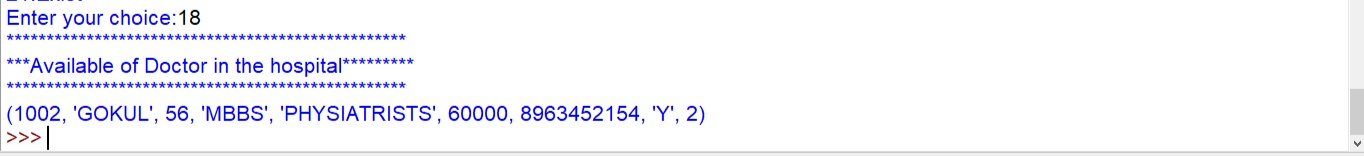




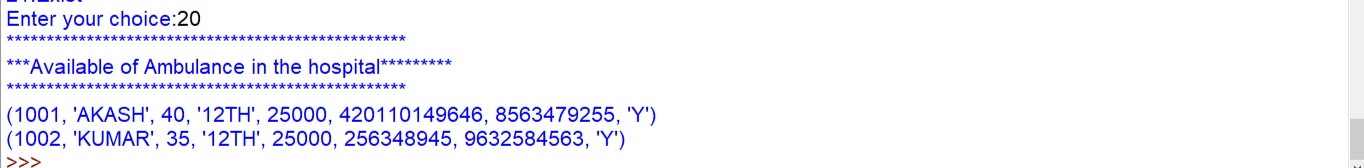














**CONCLUSION:**

This program gives the receptionist an easy\_to\_use interface where he can easily manipulate the data of users.It also easy to store all patient,doctor,workers and drivers details

# BIBLIOGRAPHY

## BOOKS AUTHORS

* + Computer sciencewithPython - Sumitha Arora

**References**

### python.org

1. CodeAcademy
2. tutorialsPoint.com
3. PythonChallenge.com
4. Google’s PythonClass
5. LearnPython.org