**PYTHON OVERVIEW**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is Interpreted − Python is processed at runtime by the interpreter. We do not need to compile our program before executing it. This is similar to PERL and PHP. Python is Interactive − We can actually sit at a Python prompt and interact with the interpreter directly to write our programs.

Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects. Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

## Python Features

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − We can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

**INTRODUCTION**

Health of citizen is the wealth of nation. India has contributed the most ancient Medical science “AYURVEDA” to the world besides other medical sciences. This field had witnessed a rapid metamorphosis in all of its sections. Hospital Management System is designed to improve the quality and management of hospital in the areas of clinical process analysis and activity-based costing.

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 the hospitals. More importantly it is backed by reliable and dependable support.

The project ‘Hospital Management System’ is based on the database, object oriented and networking techniques. As there are many areas where we keep records in database for which we are using MySQL software which is one of the best and the easiest software to keep our information. This project uses Python as the front-end software which is an object oriented programming and has connectivity with MySQL.

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 packages highly customizable and can be modified as per the needs and requirement has given it a wonderful shape both technically and usability wise. It covers the required modules right from Patient Registration, Medicine details, Doctor, Wards, Admin, Store, Patient appointment, bill payment, record modification, discharge details etc.

# PROBLEM DEFINITION

The current manual system has a lot of paper work. To maintain the records of sale and service manually, is a Time-consuming task. With the increase in database, it’ll become a massive task to maintain the database. Requires quite a bit of space in the office, which can be used for storing records of previous details. The retrieval of records of previously registered patients will be a tedious task. Lack of security for the records, anyone disarrange the records of your system. If someone want to check the details of the available doctors the previous system doesn’t provide any necessary detail of this type.

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

To overcome all such issues Hospital Management system is created. The project file contains a python script (setup.py) and other essential project files. This is a simple GUI based web application which is very easy to understand and use. Talking about the application, the user can easily view total patients, doctors, appointments and manage them all.

**OBJECTIVES OF THE PROJECT**

The objectives of the project are the following:

* **Computerization** – All the details regarding hospitals, whether it is small or big, will be computerised.
* **Automated inventory** – If the medicines are provided to the patients, the stock will be reduced in the inventory, and will help in to know the status the available medicines.
* **No redundancy** – For every test that is conducted of the patients, an automated report will be generated and will be available to the patients and his / her concerned doctor uniformly.
* **Keep the Records**– It will be easier task for the management to keep the record of the patients for historical purpose.
* **Appointment**– It will be easier for both the doctors and the patients to have the appointments. It is just two clicks away

## ADVANTAGES

* Low maintenance cost
* Volume of data is not an issue
* Data can be converted easily to information
* Data can’t be easily with proper backup
* It can be expanded as well as data communication is possible

**SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS**

Processor : Intel

RAM : 512 MB or more

Hard Disk :80GB or more

CD/DVD r/w multi drive combo

MONITOR :14.1 or 15 -17 inch

Keyboard and mouse

Printer

**SOFTWARE REQUIREMENTS:**

Windows OS

Python

MySQL

**IMPLEMENTATION OF THE PROJECT**

**WORKING**

Hospital Management system project is written in Python. The project file contains a python script (setup.py) and other essential project files. This is a simple GUI based web application which is very easy to understand and use. Talking about the application, the user can easily view total patients, doctors, appointments and manage them all.

To book manage patients, the user has to enter his/her name, insurance number, address and phone number. Likewise, to add doctors detail, the user has to enter his/her name, address, and phone number. In order to book an appointment, the user has to select a doctor, then the patient and book date and time. This project mainly focuses on CRUD with search functionality. The design is so simple that the user won’t find any difficulties while working on it.

Through this project a patient’s, doctor’s or a worker’s detail can be easily maintained, stored and used whenever required without wasting time on big heaps of paper

**MySQL Database**

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

**FLOWCHART**

**SOURCE CODE**

**TABLE.PY**

import mysql.connector as sql

conn=sql.connect(host='localhost',user='root',passwd='tanya',database='project')

if conn.is\_connected():

print('successfully connected')

c1=conn.cursor()

c1.execute('create table patient\_details(p\_name varchar(25) primary key,p\_age int(3),p\_problems varchar(40),p\_phono int(15))')

print('table created')

c1.execute('create table doctor\_details(d\_name varchar(25) primary key,d\_age int(3),d\_department varchar(40),d\_phono int(15))')

print('table created')

c1.execute('create table worker\_details(w\_name varchar(25) primary key,w\_age int(3),w\_workname varchar(40),w\_phono int(15))')

print('table created')

c1.close()

**MENU.PY**

## from sys import exit

## import mysql.connector as sql

## conn=sql.connect(host='localhost',user='root',passwd='tanya',database='project')

## if conn.is\_connected():

## print('successfully connected')

## c1=conn.cursor()

## print('---------------------------------------------')

## print("HOSPITAL MANAGEMENT SYSTEM")

## print('---------------------------------------------')

## print('"GOD WISHES YOU"')

## print("1.LOGIN")

## print("2.EXIT")

## choice=int(input("ENTER YOUR CHOICE:"))

## if choice==1:

## u1=input("enter user name:")

## pwd1=input("enter the password:")

## while u1=='vasu'and pwd1=='vasu6072':

## print('connected')

## 

## print("WELCOME TO HOSPITAL")

## print("successfully connected")

## print('1.RegisteringPatient details')

## print('2.RegisteringDoctor details')

## print('3.RegisteringWorker details')

## print("4.total patient details")

## print("5.total doctor details")

## print("6.total worker details")

## print('7.Patient detail')

## print('8.Doctor detail')

## print('9.Worker detail')

## print('10.Exit')

## choice=int(input('ENTER YOUR CHOICE:'))

## if choice==1:

## p\_name=input('Enter Patient Name:')

## p\_age=int(input('Enter Age:'))

## p\_problems=input('Enter the Problem/Disease:')

## p\_phono=int(input('Enter Phone number:'))

## sql\_insert="insert into patient\_details values(""'"+p\_name+"',"+str(p\_age)+",'"+p\_problems+"',"+str(p\_phono)+")"

## c1.execute(sql\_insert)

## print('SUCCESSFULLY REGISTERED')

## conn.commit()

## 

## elif choice==2:

## d\_name=input('Enter Doctor Name:')

## d\_age=int(input('Enter Age:'))

## d\_department=input('Enter the Department:')

## d\_phono=int(input('Enter Phone number:'))

## sql\_insert="insert into doctor\_details values(""'"+d\_name+"',"+str(d\_age)+",'"+d\_department+"',"+str(d\_phono)+")"

## c1.execute(sql\_insert)

## print('successfully registered')

## conn.commit()

## 

## elif choice==3:

## w\_name=input('Enter Worker Name:')

## w\_age=int(input('Enter Age:'))

## w\_workname=input('Enter type of work:')

## w\_phono=int(input('Enter Phone number:'))

## sql\_insert="insert into worker\_details values(""'"+w\_name+"',"+str(w\_age)+",'"+w\_workname+"',"+str(w\_phono)+")"

## c1.execute(sql\_insert)

## print('successfully registered')

## conn.commit()

## 

## elif choice==4:

## sql\_w='select\*from patient\_details '

## c1.execute(sql\_w)

## r = c1.fetchall()

## for i in r :

## print(i)

## 

## elif choice==5:

## sql\_x="select\*from doctor\_details"

## c1.execute(sql\_x)

## s=c1.fetchall()

## for i in s:

## print(i)

## 

## elif choice==6:

## sql\_y="select\*from worker\_details"

## c1.execute(sql\_y)

## t=c1.fetchall()

## for i in t:

## print(i)

## 

## elif choice==7:

## h=input("Enter the name:")

## sql\_w='select\*from patient\_details where p\_name=("{}")'.format(h)

## c1.execute(sql\_w)

## u = c1.fetchall()

## for i in u:

## print(i)

## 

## elif choice==8:

## d=input("Enter the name:")

## sql\_d='select\*from doctor\_details where p\_name=("{}")'.format(d)

## c1.execute(sql\_d)

## v=c1.fetchall()

## for i in v:

## print(i)

## 

## elif choice==9:

## f=input("Enter the name:")

## sql\_f='select\*from worker\_details where p\_name=("{}")'.format(f)

## c1.execute(sql\_f)

## w=c1.fetchall()

## for i in w:

## print(i)

## 

## elif choice==10:

## exit()

## break

## else:

## print('wrong username&password')

## if choice==2:

## exit()

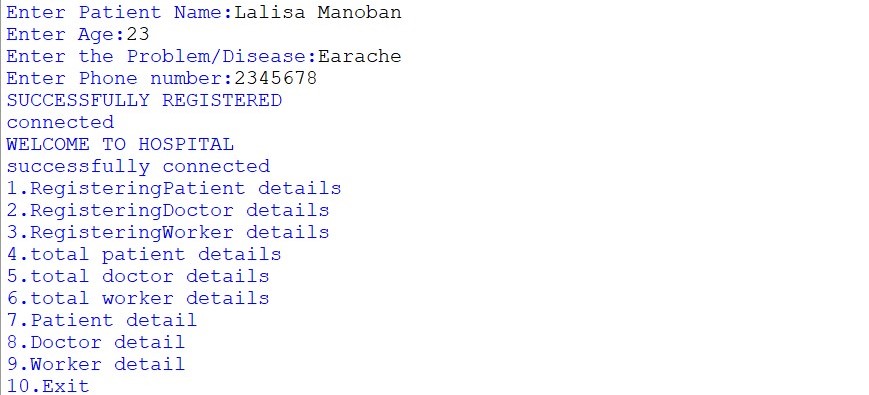
## 

## 

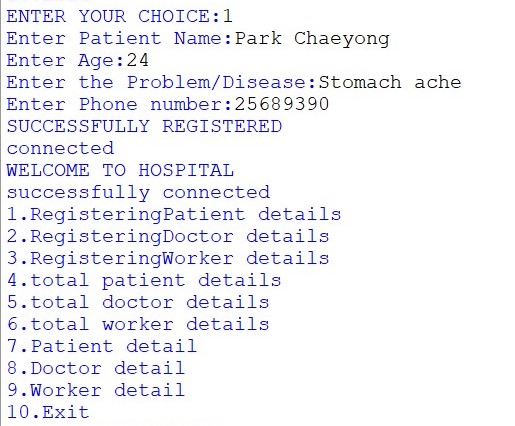
## OUTPUT



Adding first new patient:

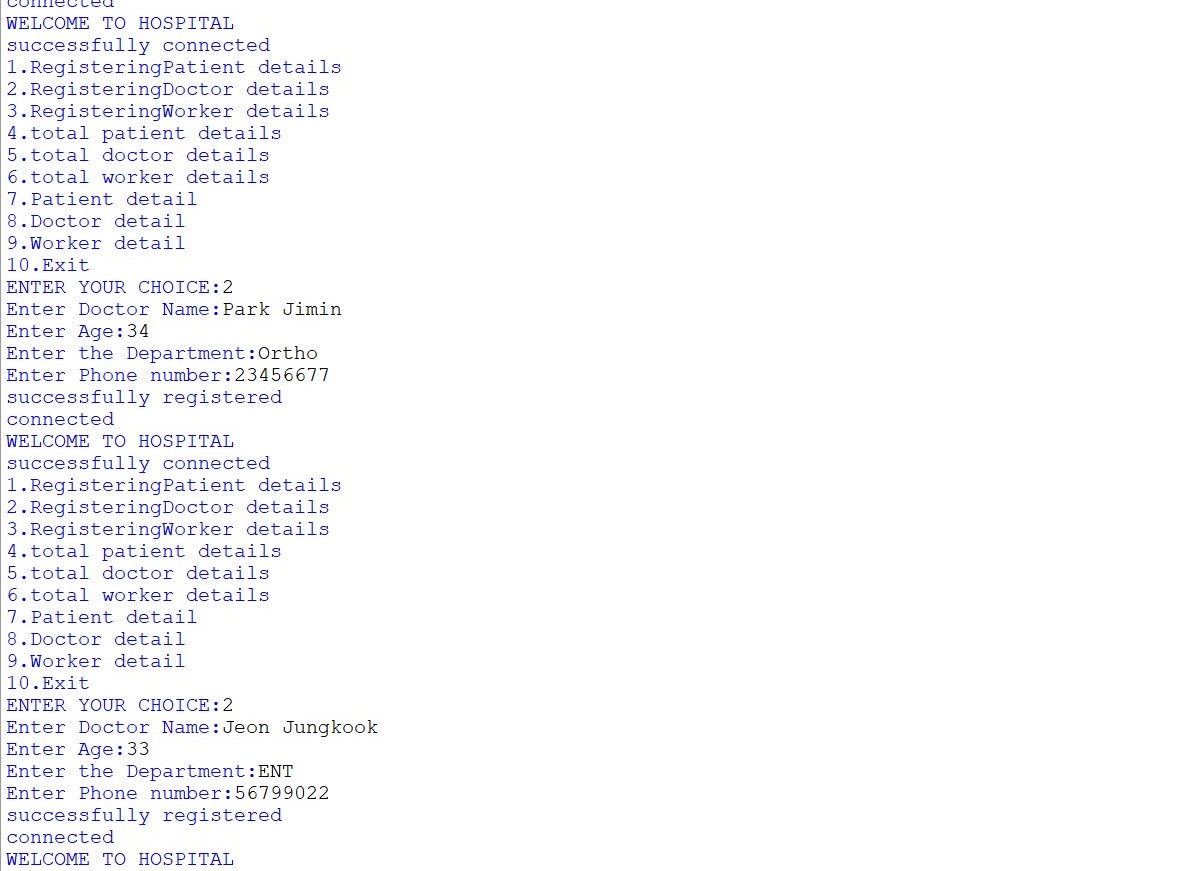


Adding second new contact:

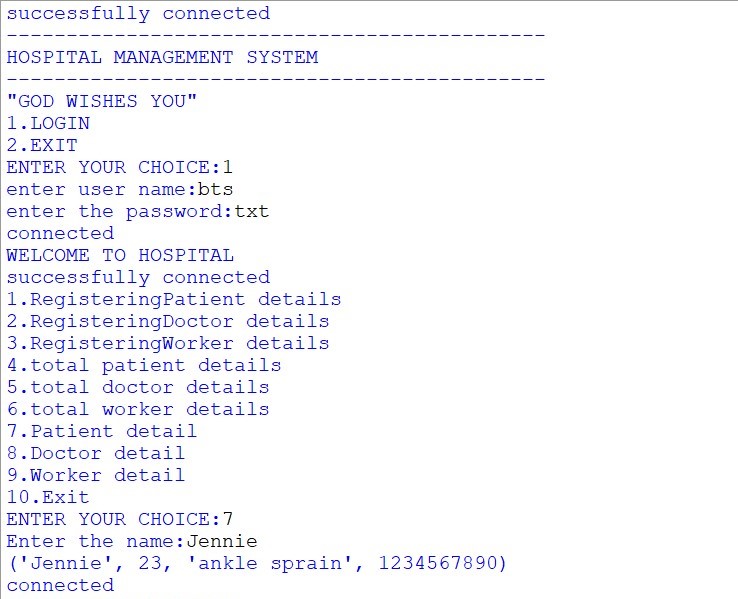


Etc…similarly more patient’s details can be added

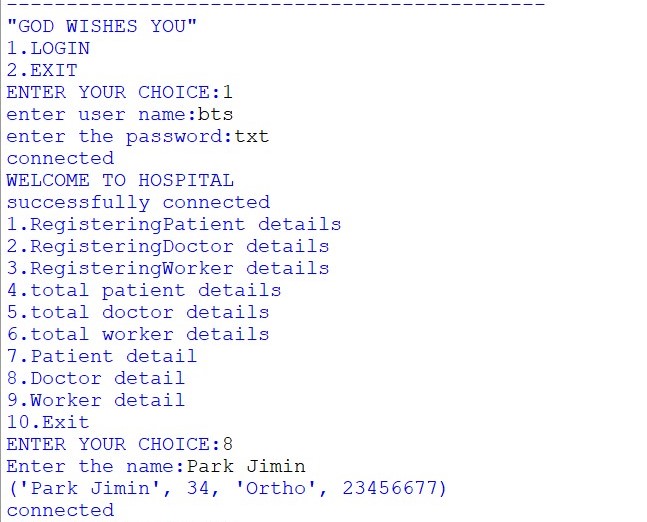
List of doctors:



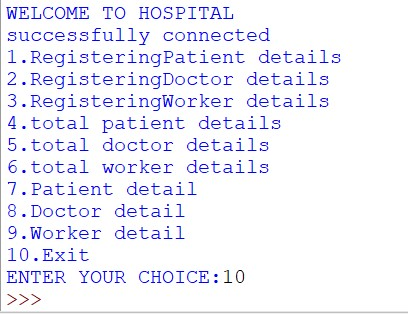
Search for patient:



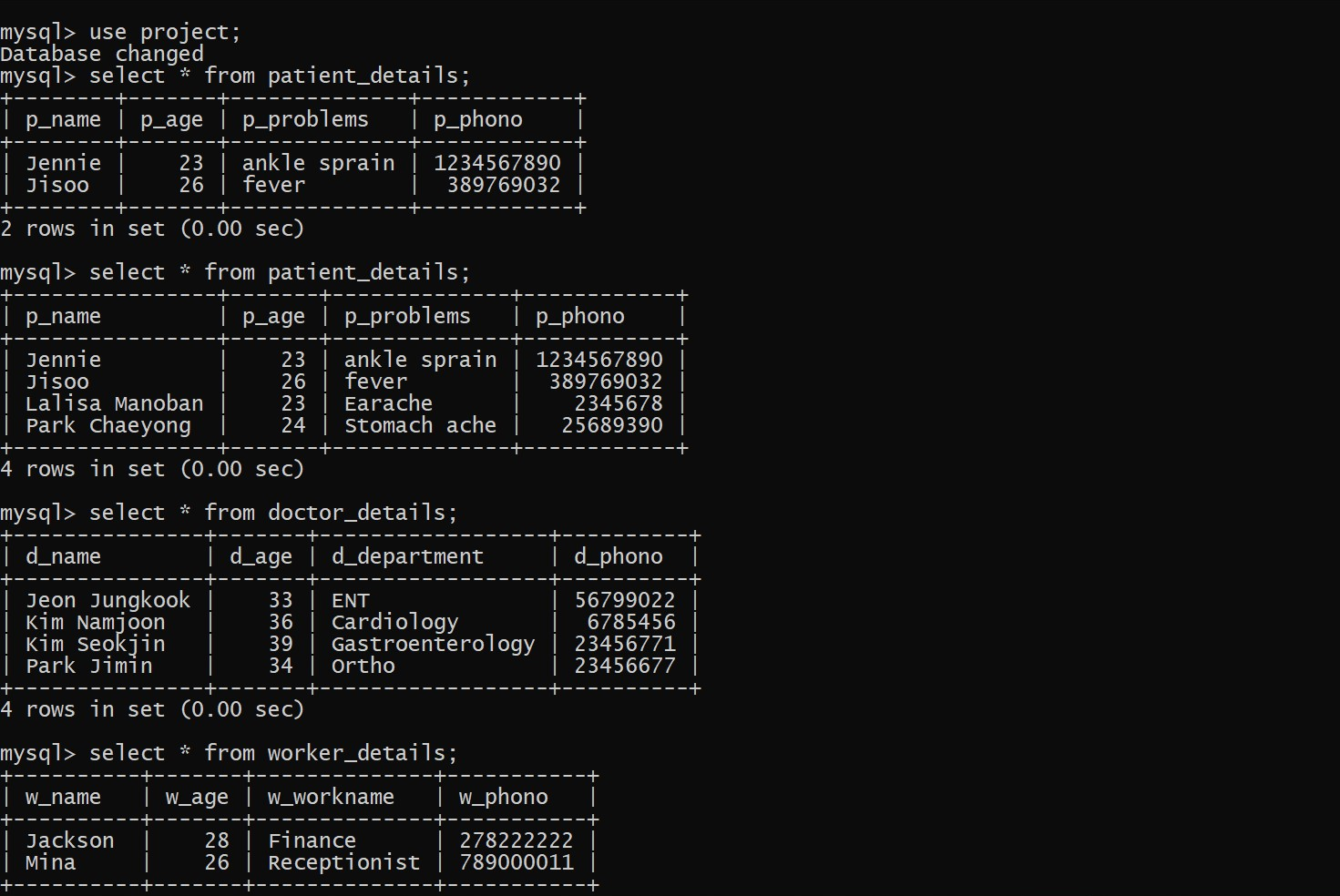
Search for Doctor:



Exit:



MySQL:

****

**INSTALLATION PROCEDURE**

* Install python and my sql from the web.
* Open mysql and create the database bank
* Then run the table.pyfile.
* Next run menu.py file.
* Enter the details in all the tables
* At last open mysql and view the details by passing queries

**CONCLUSION**

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.

This application makes our work super easy by maintaining all the patient’s records and doctor’s record. It makes the consultation and the registration process easy and burden free. There is no more any requirement of maintaining big piles of papers separately for both patient and doctor.

Database management system can ease the work of any institution in a quick move just by interfacing it with python.

**BIBLIOGRAPHY**

* Computer science With Python -  Class XII    By : Sumita Arora
* A Project Report On Hospital Management System (HMS)    By : Praveen M Jigajinni
* Website: <https://www.w3resource.com>
* https://en.wikipedia.org/wiki/E\_(mathematical\_constant)