****

**SUBMITTED**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT**

**FOR THE**

**ALL INDIA SENIOR SCHOOL CERTIFICATE**

**EXAMINATION(AISSCE)**

**SESSION: 2021-2022**

**INVESTIGATORY PROJECT :**

**‘HOSPITAL MANAGEMENT SYSTEM’**

**GUIDED BY: SUBMITTED BY:**

**Ms. SHEELA KALE KARTAVYA BANG**

**(HOD COMPUTER)**

INDEX

1. **INTRODUCTION ABOUT PYTHON**
2. **INTRODUCTION ABOUT MYSQL**
3. **INTRODUCTION TO THE PROJECT**
4. **TYPES OF PATIENTS**
5. **COMPONENTS OF HOSPITAL MANAGEMENT SYSTEM**
6. **FLOWCHART OF PROJECT**
7. **DATABASE AND FILES USED IN THE PROJECT**

INTRODUCTION ABOUT PYTHON

Python is [dynamically-typed](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) and [garbage-collected](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US). It supports multiple [programming paradigms](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US), including [structured](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) (particularly [procedural](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US)), [object-oriented](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) and [functional programming](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US). It is often described as a "batteries included" language due to its comprehensive [standard library](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US).

[Guido van Rossum](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) began working on Python in the late 1980s as a successor to the [ABC programming language](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as [list comprehensions](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US), [cycle-detecting](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) garbage collection, [reference counting](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US), and [Unicode](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) support. Python 3.0, released in 2008, was a major revision that is not completely [ba](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US)ckwards-compatible with earlier versions. Python 2 was discontinued version 2.7.18 in 2020.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn’t catch the exception, the interpreter prints a stack trace. A source-level debugger allows inspection of local and global variables, evaluation of an arbitrary expression, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python’s introspective.

INTRODUCTION ABOUT SQL

**SQL** ( ([https://upload.wikimedia.org/wikipedia/commons/thumb/8/8a/Loudspeaker.svg/11px-Loudspeaker.svg.pnglisten](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US)) *S-Q-L*, "sequel"; **Structured Query Language**) is a [domain-specific language](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) used in programming and designed for managing data held in a [relational database management system](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) (RDBMS), or for stream processing in a [relational data stream management system](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) (RDSMS). It is particularly useful in handling [structured data](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US), i.e. data incorporating relations among entities and variables. SQL offers two main advantages over older read-write [APIs](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) such as [ISAM](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US) or [VSAM](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US). Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify *how* to reach a record, e.g. with or without an [index](https://wikipedia.firstpartyapps.oaspapps.com/wikipedia/wikipedia_dev.html?et=%2BAA%3D%3D&_host_Info=Word$Win32$16.01$en-US).

Originally based upon [relational algebra](https://en.wikipedia.org/wiki/Relational_algebra) and [tuple relational calculus](https://en.wikipedia.org/wiki/Tuple_relational_calculus), SQL consists of many types of statements,[[6]](https://en.wikipedia.org/wiki/SQL#cite_note-6) which may be informally classed as [sublanguages](https://en.wikipedia.org/wiki/Sublanguage), commonly: a [data query language](https://en.wikipedia.org/wiki/Data_query_language) (DQL),[[a]](https://en.wikipedia.org/wiki/SQL#cite_note-7) a [data definition language](https://en.wikipedia.org/wiki/Data_definition_language) (DDL),[[b]](https://en.wikipedia.org/wiki/SQL#cite_note-8) a [data control language](https://en.wikipedia.org/wiki/Data_control_language) (DCL), and a [data manipulation language](https://en.wikipedia.org/wiki/Data_manipulation_language) (DML).[[c]](https://en.wikipedia.org/wiki/SQL#cite_note-9)[[7]](https://en.wikipedia.org/wiki/SQL#cite_note-10) The scope of SQL includes data query, data manipulation (insert, update and delete), data definition ([schema](https://en.wikipedia.org/wiki/Database_schema) creation and modification), and data access control. Although SQL is essentially a [declarative language](https://en.wikipedia.org/wiki/Declarative_programming) ([4GL](https://en.wikipedia.org/wiki/4GL)), it also includes [procedural](https://en.wikipedia.org/wiki/Procedural_programming) elements.

INTRODUCTION ABOUT PROJECT HOSPITAL MANAGEMENT SYSTEM

Hospitals are significant as they treat minor and serious diseases, illnesses and disorders of the body function of varying types and severity. Moreover, they also help in promoting health, giving information on the prevention of illnesses and providing curative services.

Further, one of the most important functions of hospitals is that they offer multiple healthcare professionals. It is filled with a host of doctors, nurses and interns. When a patient goes to a hospital, many doctors do a routine check-up to ensure maximum care.

Every hospital has a variety of tasks to accomplish like Appointment Management. Patient Management, Facility Management, Staff Management, Supply Management, Financial Management, Insurance Management, Laboratory Management.

**Our Hospital Management System will deal with the patient, doctor, prescription, tests and reports.**

TYPES OF PATIENTS

Depending upon the length of stay of the patient, there are 2 types of patients;

OPD PATIENT: OPD Patient stands for Out-Patient Department, perhaps the simplest component of the hospital management system. Patients that only came for a check-up with a doctor were not admitted to the hospital. The doctor can run necessary tests, write the prescriptions, and sends the patient back. The patient takes the medicines at home.

IPD Patient: IPD Patient stands for In-Patient Department. When a patient is admitted to a hospital bed, that patient is considered an IPD patient. The patient gets all the medicines and care in the hospital itself. The patient is only discharged from the hospital if their condition is stable. If the patient doesn’t survive then the dead body is returned to the family of the patient after filling out all the formalities.

There are many ways a patient can be an IPD patient.

* When the patient’s condition is very critical, for example, if a patient is met with a lethal accident, the patient is directly rushed to the hospital beds.
* When the doctor diagnoses a patient with a critical condition, such as diseases like leukaemia, malaria, etc, the patient is then admitted to the hospital.

Most hospitals specialize in a few healthcare fields, for example, Bansal Hospital in Bhopal, India specialises in,

* Cardiology
* Cardiothoracic And Vascular Surgery
* Dermatology and Cosmetology
* Dietetics
* Gastro Surgery, etc

Now that we have talked about the patients and their types, we must talk about the basic Hospital Management System.

COMPONENTS OF HOSPITAL MANAGEMENT SYSTEM

Hospital Management System has,

* **DOCTOR MANAGEMENT SYSTEM**

Every hospital has a wide variety of specialized doctors to tend to the patients, so in our system, we should be able to manage all the doctors, that includes their contact information and their consultancy time.

In this system, we must be able to add, modify and remove any data regarding the doctor and we must be able to access the data at any given time. Only the supervisors and the HR team of the hospital be able to modify this data. Regular staff should not have the privileges to modify it.

* **PATIENT MANAGEMENT SYSTEM**

Every hospital has a lot of patients coming in in the form of **OPD** and **IPD**, our system must have the data of each patient setting foot inside the hospital. The data should be separated from all the other patients, data should not be mixed so that we can give a patient a particular ID which is a unique identifier for the said patient. The amount due and the amount paid should also be there with the contact information of the patient. Any further visits should be added to the already existing data and the old data must be archived properly to stop any disputes regarding payments from happening.

* **PRESCRIPTION MANAGEMENT SYSTEM**

If the doctor has prescribed medicine to any patient, our system must store that information to make it easier for the doctor to prescribe the same medicine to other patients suffering from the same issue. It should also have the average amount prescribed in it to give other doctors ideas about the dosage of the medicine.

* **TESTS MANAGEMENT SYSTEM**

Our system must also have the collection of all the tests that a hospital can provide. It must also be able to apply to the final report of the patient with the results and the healthy value for comparison. This makes it easier for the patient to understand any abnormalities in themselves and for the doctor to prescribe effective medication.

It should also contain the price for each test to be paid by the patient. The test can be added by the test department of the hospital and must have the ability to add, modify or delete any data regarding the test, the price for each test must be accurate and the healthy data must be accurate.

* **REPORT MANAGEMENT SYSTEM**

Our system should be able to build a final report of the patient based on the tests, prescriptions and remarks by the doctors. It will contain the summary of the tests, and the final amount to be paid by the patient. The doctor should be able to hand it over to the patient via,

1. **Hard Copy (printed)**
2. **Email**
3. **WhatsApp**

To make a report, the report making must be initiated first and then only the report should be allowed to be made. All the data in the report should be double-checked and must be double-checked with the data in the database.

* **ADMIN PANEL**

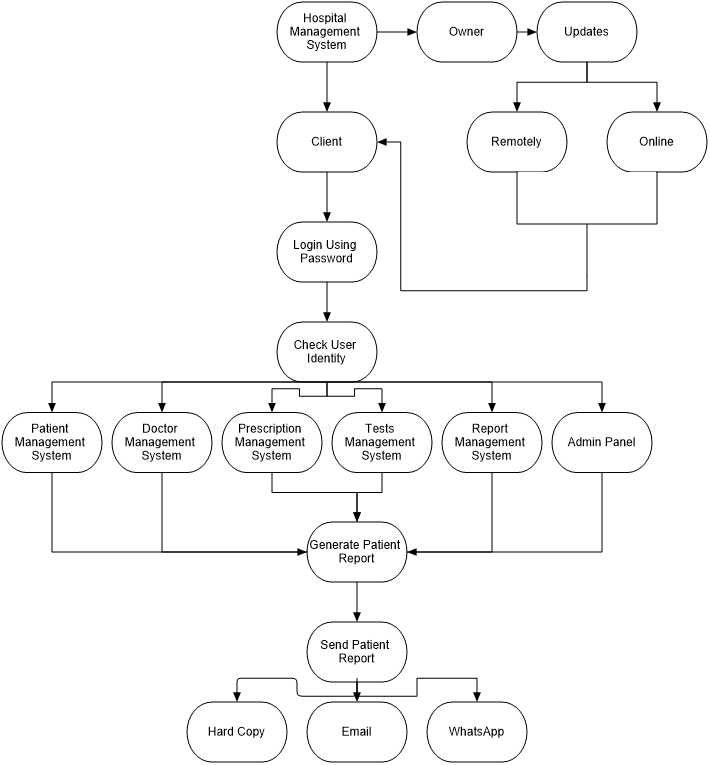
The admin panel contains the key to all of the hospital management systems. It should only be available to the highest-ranking person in the hospital, it should be password protected and the password should be stored securely to protect it from getting hacked.

The admin panel can add or modify data without any approval, that is why it should not be given to any staff, as they might delete the core components of the system.

* **OWNER PANEL**

This panel is exclusively available only to the owner of the software (hospital management system). The owner cannot look at the actual data but can add or remove new features from it, making adjustments or overall making optimisations. The updates can be made to all the clients by either remotely adding them to the client’s server or can be uploaded to a website where the client has access to it via their login ID and password.

FLOWCHART OF PROJECT



DATABASE AND FILES USED IN PROJECT

**DATABASE: HMS**

**TABLES:**

1. **PATIENTS**
2. **DOCTORS**
3. **TESTS**
4. **PRESCRIPTIONS**

**Structure of all tables:**

**Table Name:** Doctor

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Size** | **Data Type** |
| ppin | 7 | int |
| name | 20 | varchar |
| mobile | 10 | int |
| speciality | 15 | varchar |
| consultancy\_time | 10 | varchar |