CHAPTER 1

INTRODUCTION

1.1 Background

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

The database management system (DBMS) is the software that interacts with end users, applications, the database itself to capture and analyze the data and provides facilities to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified and the database schema, which defines the database’s logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.2 Introduction to Hospital Management

The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

CHAPTER 2

ER DIAGRAM AND RELATIONAL SCHEMA

2.1 Description of ER diagram

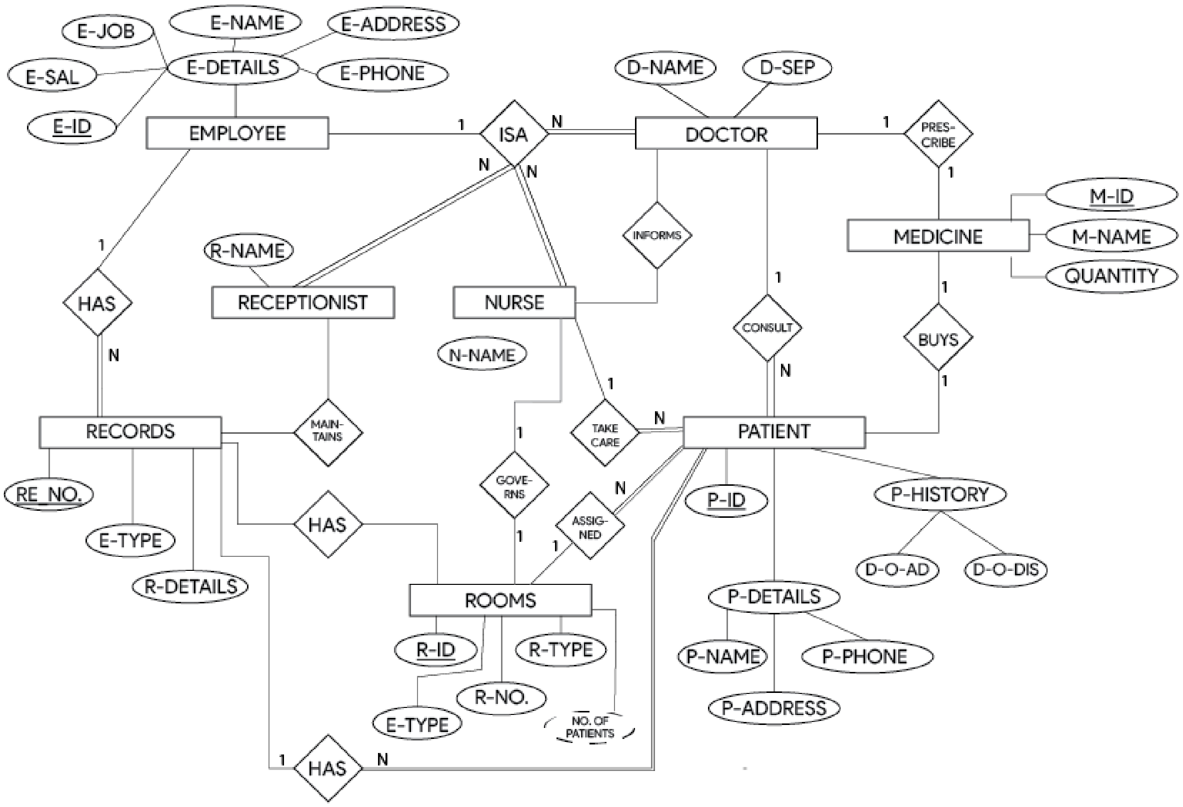


Figure 2.1: ER Diagram of Hospital DataBase

The E-R Diagram in Figure 2.1 describes entities, attributes and relationships.

- Entity types like **EMPLOYEE** and **PATIENT** are in rectangular boxes.
- Relationships like **CONSULTS** and **TAKE CARE** are in diamond boxes, attached to entity types with straight lines.
- Attributes are shown in ovals, each attached by a straight line to entity or relationship type.
- Key attributes (like ID) are underlined.
- Component attributes of a composite attribute are attached to oval representing it.

2.1.1 E-R Diagram Relationships Description

1. **EMPLOYEE**: They are people who are working for the organization, it consists of categories such as **DOCTOR, RECEPTIONIST, NURSES**. **EMPLOYEE** has cardinality ratio 1: N with company **Categories**. All **Categories Employee** are part of the **Employee**.
    - All Employees can't be a Doctor
    - All Employee Categories are Employee of the company.
2. **DOCTOR**: **PATIENTS** are treated by the **DOCTOR**, cardinality ratio of **DOCTOR** to **PATIENT** is M: N, where all doctor can treat the patient and all patients can consult the doctor hence total relationship is formed between them. Even **DOCTOR** prescribe **MEDICINE** for the **PATIENT** as a partial participation as:
    - All Doctor are Employees
    - All Medicines aren't prescribed by Doctor
    - All Doctor prescribe Medicine
3. **NURSE**: **PATIENTS** are taken care by the **NURSE**, cardinality ratio of **NURSE** to **PATIENT** is M: N. They are connected via **TAKE CARE** Relation. There is a total participation of **NURSE** and total participation of **PATIENT** as**:**
    - All Nurses take care of patient
    - All Patients are taken care by Nurse
    - Records has details of Nurses
4. **PATIENT**: Cardinality Ratio with **Employee Categories** is M: N as all the **PATIENTS** are treated and consulted by the **EMPLOYEE** hence it forms total relation with the respective relation. They are connected via **CONSULTS**, **TAKE CARE** Relation.
    - All Patients has room allotted
5. Records has Patients details. **MEDICINE**: **PATIENT** is of cardinality ratio M: N, where M **PATIENTS** can buy N **MEDICINE**. They are connected via **BUYS**. There is a partial participation of **PATIENT** and partial participation of **MEDICINE** as:
    - All Medicine are not prescribed by the Doctor
    - All Doctor prescribe required Medicine for Patient
    - All prescribed medicine is bought by the Patient

6. **ROOM**: **PATIENT** is of cardinality ratio of 1: N. Where 1 **PATIENT** can be assigned to a single room and N **ROOM** are allocated for the different Patients. They are connected via **ASSIGNED**. There is a partial participation of **PATIENT** and the **ROOMS** as:
    - Multiple Rooms aren’t allocated for a single Patient
    - Every Employee is assigned as a room in charge.
7. **Record**: **PATIENT**, **EMPLOYEE** is of cardinality ratio of 1:1. Where single Patient and Doctor has their respectively Record. They are connected via **HAS** relation. There is a partial participation of the **RECORD** and total participation of the Patient and Doctor as:
    - All Patients have record.
    - All Doctors have record.
    - Every Record are unique there is no duplication of the Record.

2.2 Relational Schema Diagram

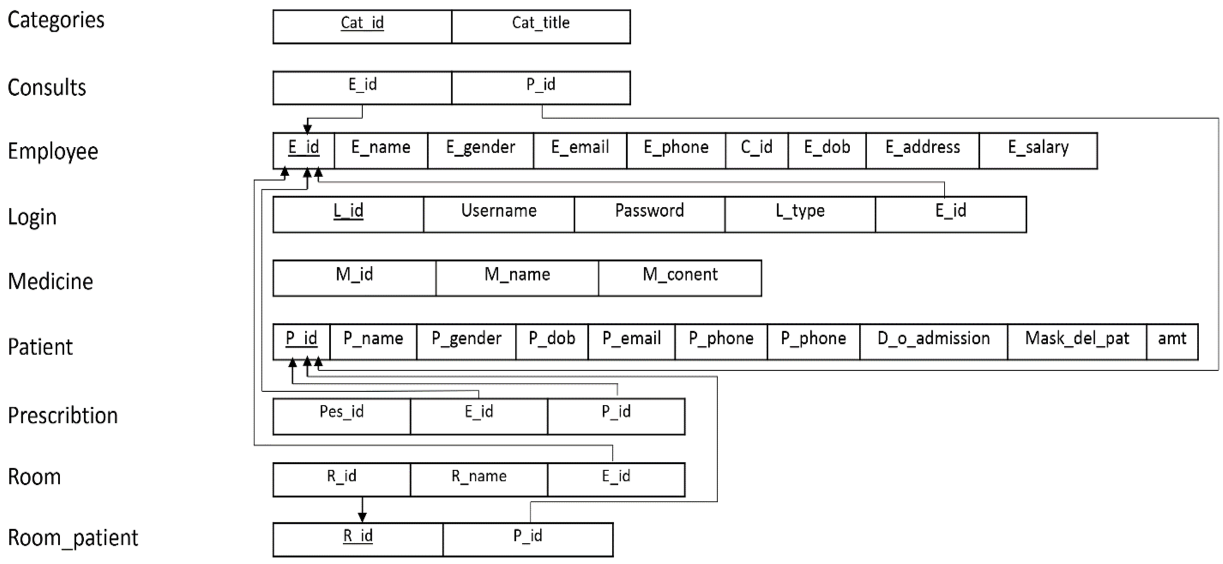


Figure 2.2: Relational schema of Hospital Database

2.2.1 General Constraints

1. **NULL Constraint**: Attributes that are under NOT NULL constraints have to be filled compulsorily. Almost all the attributes in the project are under NOT NULL constraint.
2. **Entity Integrity Constraint**: This constraint makes sure that no primary key can have a NULL value assigned to it. The primary keys involved in the project include:
   - ID
   - NAME
   - EMP ID
   - PAT ID
3. **Referential Integrity Constraints**: A table in the back end of the project may have references pointing to an attribute in another table. For example: E_ID in the PATIENT table refers to ID in EMPLOYEE table. The various tables are also linked with multiple foreign keys which are all set to cascade any update or delete operation on the attribute in the main table. The various Foreign Key attributes are:
   - E_ID
   - P_ID
   - R_ID

2.2.2 Schema Description

The above Figure .2.2 shows the relational schema of Art Gallery. It has the following entities.

1. **CATEGORIES:** It consists of categories of Employee according to their job and occupation(works).
2. **CONSULTS**: As the cardinality ratio of Employee and Patients are M: N, The **CONSULTS** table consists of details of Patients (P_id) consulting to a Doctor (E_id).
3. **EMPLOYEE:** It is a table that consists of details of Employee (Personal Details). The table consist of name, id, salary, gender, phone number, date of birth, address, etc.
4. **LOGIN:** This table consists details of all the user data. where we have attributes as username name for unique identity and password which protects the details of the Employee where the particular user can access their authority.
5. **MEDICINE:** This table consist of details of medicine which are used for the treatment of the Patient. This table has unique Medicine ID, name, content of medicine.

6. **PATIENT:** It is a table that consists of details of Patient (Personal Details). The table consist of name, id, gender, phone number, date of birth, address, etc.
7. **PRESCRIBE:** The Doctor prescribe the medicine for a particular Patient that details where for a respective prescription Id, Doctor prescribe N Medicine for that Patient.
8. **PRESCRIBTION:** It consists Id of Employee and Patient with the unique prescription Id to prescribe Medicine to that Patient.
9. **ROOM:** It consists details of the Rooms that are present in the Organization with the Room unique identification number and Room incharge.
10. **ROOM_PATIENT:** This table keeps the record of the Patient that are admitted in their respective Room as an attribute of P_Id and R_Id.

CHAPTER 3

SYSTEM DESIGN

3.1 Table Description

Categories

Table 3.1 Employee Category.

Field	Type	Null	Key	Default
Cat_id	INT(20)	No	Primary	None
Cat_title	VARCHAR(255)	No		None

The Table 3.1 contains the list of Categories. It has Id and Cat_title as the attributes where Cat_id is the primary key.

Consults

Table 3.2 Patient Consults Doctor.

Field	Type	Null	Key	Default
E_id	INT(50)	No	Foreign	None
P_id	INT(50)	No	Foreign	None

The Table 3.2 contains the record of Patient consulting Doctor. It has E_Id and P_Id as the attributes where E_Id and P_Id are foreign key to Employee table and Patient table.

Employee

Table 3.3 Employee Details.

Field	TYPE	Null	Key	Default
E_id	INT(50)	No	Primary	None
E_name	VARCHAR(255)	No		None
E_gender	VARCHAR(255)	No		None
E_email	VARCHAR(255)	No		None
E_phone	VARCHAR(10)	No		None
C_id	INT(10)	No	Foreign	None
E_dob	DATE	No		None
E_address	VARCHAR(255)	No		None
E_salary	VARCHAR(255)	No		None
Mask_del_emp	INT(10)	No		None

The Table 3.3 contains the list of all contact details of the Employee. It has a total of 10 attributes, E_ID, E_name, E_gender, E_email, E_phone, C_Id, E_dob, E_address, E_salary, Mask_del_emp. E_Id is the primary key and C_Id is the foreign key to the Categories table.

Login

Table 3.4 Employee Login.

Field	Type	Null	Key	Default
L_id	INT(11)	No	Primary	None
Username	VARCHAR(255)	No		None
Password	VARCHAR(255)	No		None
L_type	VARCHAR(255)	No		None
E_id	INT(11)	No	Foreign	None

The Table 3.4 contains the list of all Employee Login. It has a total of 5 attributes, L_ID, Username, Password L_type, E_Id. L_id is the primary key and E_Id is the foreign key to the Employee table.

Medicine

Table 3.5 Medicine Details.

Field	Type	Null	Key	Default
M_id	INT(11)	No	Primary	None
M_name	VARCHAR(255)	No		None
M_content	VARCHAR(255)	No		None

The Table 3.5 contains the details of Medicine. It has M_Id, M_name and M_content as the attributes where M_Id is the primary key.

Prescription

Table 3.6 Prescription Details.

Field	Type	Null	Key	Default
Pes_id	INT(11)	No	Primary	None
E_id	INT(11)	No	Foreign	None
P_id	INT(11)	No	Foreign	None

The Table 3.6 contains the details of Prescription where Doctor prescribe Patient. It has Pes_Id, E_Id and P_Id as the attributes where M_Id is the primary key. E_id and P_Id are foreign key to Employee table and Patient table.

Patient

Table 3.7 Patient Details.

Field	Type	Null	Key	Default
P_id	INT(11)	No	Primary	None
P_name	VARCHAR(255)	No		None
P_gender	VARCHAR(255)	No		None
P_dob	DATE	No		None
P_email	VARCHAR(255)	No		None
P_phone	VARCHAR(255)	No		None
P_address	VARCHAR(255)	No		None
D_o_admission	VARCHAR(255)	Yes		Null
Mask_del_pat	INT(11)	No		None
Amt	INT(11)	No		None

The Table 3.7 contains the list of all contact details of the Patient. It has a total of 10 attributes, P_ID, P_name, P_gender, P_dob, P_email, P_phone, P_address, D_o_admission, Mask_del_pat. P_Id is the primary key.

Room_patient

Table 3.8 Room allocated to patient.

Field	Type	Null	Key	Default
R_id	INT(11)	No	Foreign	1
P_id	INT(11)	No	Foreign	None

The Table 3.8 contains the record of Patient allocated Room details. It has R_Id and P_Id as the attributes where R_Id and P_Id are the foreign key to the Room table and patient table.

Room

Table 3.9 Room Details.

Field	Type	Null	Key	Default
R_id	INT(11)	No	Primary	None
R_name	VARCHAR(255)	No		None
E_id	INT(11)	No	Foreign	None

The Table 3.9 contains the details Room. It has R_Id, R_name and E_Id as the attributes where R_Id is the primary key. E_Id are the foreign key to the Employee table.

3.2 Stored Procedures

One stored is used to get the details of no of patients in the particular room when the receptionist assigns the room to the respective patient. It gives the Count of Patients that are admitted in the hospital for the unique Rooms.

```
CREATE PROCEDURE no_of_patients
AS
SELECT COUNT(DISTINCT(e_id))
FROM room_patient
GROUP BY r_id
GO;
```

To Execute procedure type **EXEC**

```
EXEC no_of_patient;
```

3.3 Trigger

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated. It Trigger the prescription whenever new patient is added.

```
CREATE TRIGGER prescription_id
BEFORE INSERT ON patient
FOR EACH ROW
SET e_id = NEW.e_id ;
```

CHAPTER 4

IMPLEMENTATION

4.1 Front-end Development

The front-end is built using a combination of technologies such as Hypertext Markup Language (HTML), JavaScript and Cascading Style Sheets (CSS). **Front-end developers design and construct the user experience elements** on the web page or app including buttons, menus, pages, links, graphics and more.

4.1.1 Hypertext Markup Language

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. HTML is the standard markup language for creating Web pages. It stands for Hyper Text Markup Language. It describes the structure of a Web page. It consists of a series of elements. It elements tell the browser how to display the content. It elements are represented by tags. HTML tags label pieces of content such as "heading", "paragraph", "table", and so on. Browsers do not display the HTML tags, but use them to render the content of the page.

4.1.2 Cascading style sheets

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications. Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. CSS solved that issue. SS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file. CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

4.1.3 JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities. Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server. JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly. Advantages are: **Less server interaction, immediate feedback to the visitors, increased interactivity** and **richer interfaces.**

4.2 Back-end Development

Backend is server side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend.

4.2.1 Backend scripting language - PHP Hypertext Preprocessor

PHP is used as the server side scripting language. PHP is an acronym for "PHP: Hypertext Preprocessor". PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. It is compatible with all servers used today. It is easy to use and runs efficiently on the server side. It can run on various platforms like windows, Linux, UNIX, Mac OS-X etc. and since it is a scripting language, it comes with predefined functions which makes it easy to implement any logic necessary.

4.2.2 Web Server – Apache

Apache is an open-source and free web server software that powers around 46% of websites around the world. The official name is Apache HTTP Server, and it's maintained and developed by the Apache Software Foundation. It allows website owners to serve content on the web — hence the name "web server". Although we call Apache a web server, it is not a physical server, but rather a software that runs on a server. Its job is to establish a connection between a server and the browsers of website visitors (Firefox, Google Chrome, Safari, etc.) while delivering files back and forth between them (client-server structure). Apache is a cross-platform software; therefore, it works on both UNIX and Windows servers.

4.2.3 Database – MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. It is developed, marketed and supported by MySQL AB, which is a Swedish company. It is released under an open-source license. So you have nothing to pay to use it. It 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. It uses a standard form of the well-known SQL data language. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It works very quickly and works well even with large data sets. It 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). It is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

4.3 Discussion of Code Segment

This section talks about the important code sections and modules that are implemented in the Airline Reservation System project. These modules add logic to the complete system, and make it function the way it is supposed to. It also talks about the integration between the front-end HTML code and the back-end MySQL database.

4.3.1 Connect to Database

```
<?php
            $connection = mysqli_connect('localhost','root',",'hospital');
            if (!$connection) {
                    die("Not Connected to DBMS");
            }
    ?>
```

4.3.2 Login Module

```
$query = "SELECT * FROM login WHERE username='{$username_l}' AND l_type = 'receptionist'";
$selet_recep_query = mysqli_query($connection,$query);

if(!$selet_recep_query){
  die("Not query " . mysqli_error($selet_recep_query));
}

while ($row = mysqli_fetch_assoc($selet_recep_query)) {
  $username = $row['username'];
  $password = $row['password'];
  $emp_id   = $row['e_id'];
}

if($username_l === $username && $password_l === $password){
    $_SESSION['r_id'] = $emp_id ;
    $_SESSION['r_username'] = $username ;
    header("Location: ../receptionist/ " );
```

```
      }


       else{


            $_SESSION['errmsg'] = "Invalid Username and Password";
            header("Location: ../login_recep.php");
      }
    }
```

4.3.3 Logout Module

```
<?php
            ob_start();
            session_start();


             $_SESSION['e_id'] = null ;
             $_SESSION['username'] = null;
             $_SESSION['errmsg'] = null;


    header("Location: ../../../index.php");


?>
```

4.3.4 Update Module

```
if(isset($_POST['update_post'])) {


            $emp_name              = escape($_POST['names']);
            $emp_gender            = escape($_POST['gender']);
            $emp_email             = escape($_POST['email']);
            $emp_phone             = $_POST['phonenumber'];
            $emp_category_id       = escape($_POST['category']);
            $emp_dob               =  $_POST['dob'];
            $emp_salary            = $_POST['salary'];
            $emp_address           = escape($_POST['addres1']);
            $query = " UPDATE employee SET " ;
```

```
        $query .= "e_name     =  '{$emp_name}' ," ;
        $query .= "e_gender   =  '{$emp_gender}' ," ;
        $query .= "e_email    =  '{$emp_email}' ," ;
        $query .= "e_phone    =  '{$emp_phone}' ," ;
        $query .= "c_id       =   {$emp_category_id} ," ;
        $query .= "e_dob      =  '{$emp_dob}' ," ;
        $query .= "e_address  =  '{$emp_address}' " ;
        $query .= "WHERE e_id = $the_employee_id " ;
        $create_post_query = mysqli_query($connection, $query);
        confirmQuery($create_post_query);
    }
```

4.3.5 Insert Module

```
$query="INSERT INTO
employee(e_name,e_gender,e_email,e_phone,c_id,e_dob,e_address,e_salary)";

$query.=VALUES('{$emp_name}','{$emp_gender}','{$emp_email}','{$emp_phone}',
{$emp_category_id},'{$emp_dob}','{$emp_address}','{$emp_salary}') ";
 $create_post_query = mysqli_query($connection, $query);
```

4.4 SYSTEM REQUIREMENTS

4.4.1 Hardware Requirements

- Processor: P-IV 2.8 GHz Processor.
- Main Memory: RAM 2GB.
- Secondary Memory: 20 GB HDD

4.4.2 Software Requirements

- OS: WINDOWS OS (XP / 2000 / 200 Server / 2003 Server/windows10)
- Server: XAMPP (Apache)
- Front End: PHP, HTML, CSS
- Back End: MySQL(phpMyAdmin)

4.5 Results

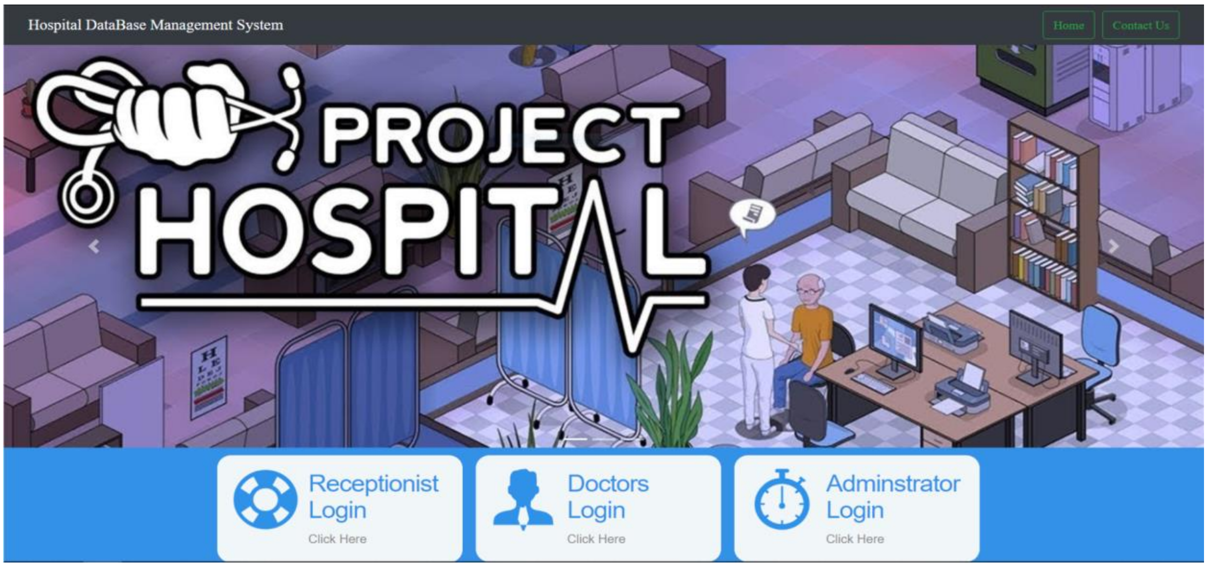
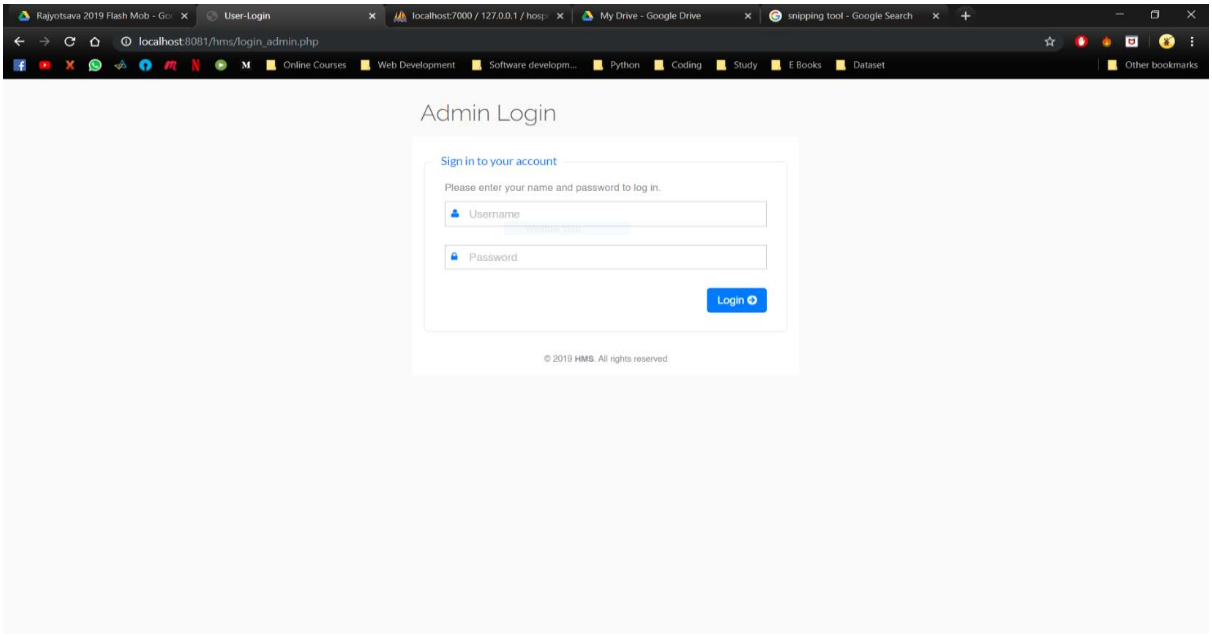


Figure 4.5.1: Home page

The Above Figure 4.5.1 is the snapshot of the Home page is the staring page of the Project where it runs from the localhost i.e. 192.172.0.1/homepage.html



Admin Login

Sign in to your account

Please enter your name and password to log in.

Username

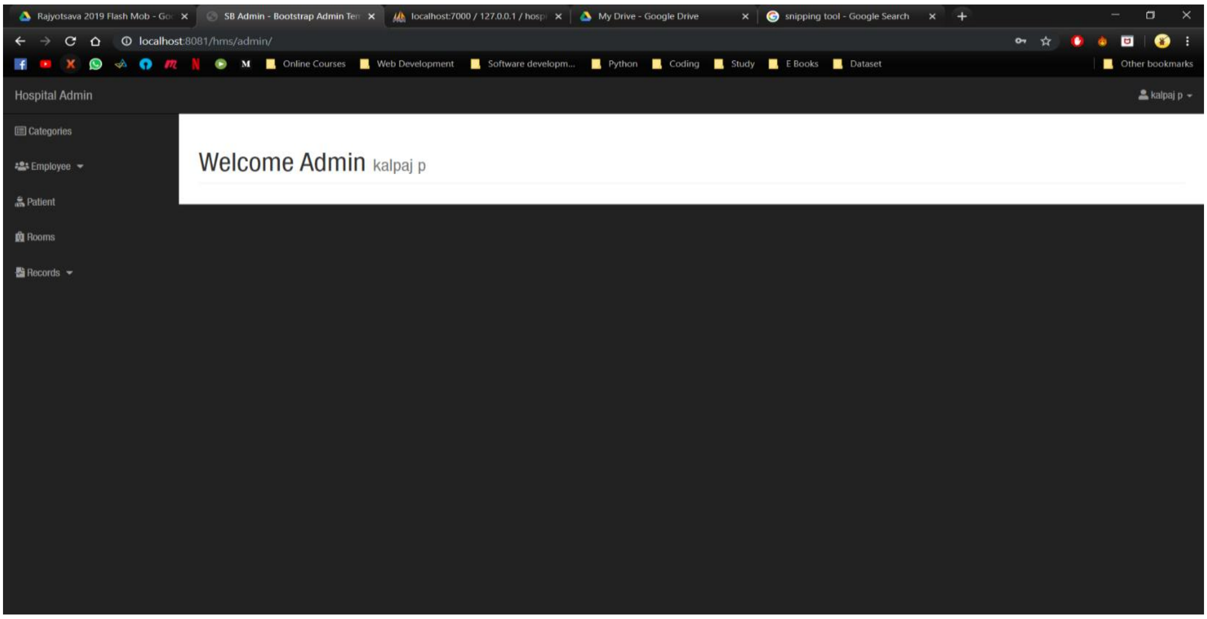
Password

Login

© 2019 **HMS**. All rights reserved

Figure 4.5.2: Admin login

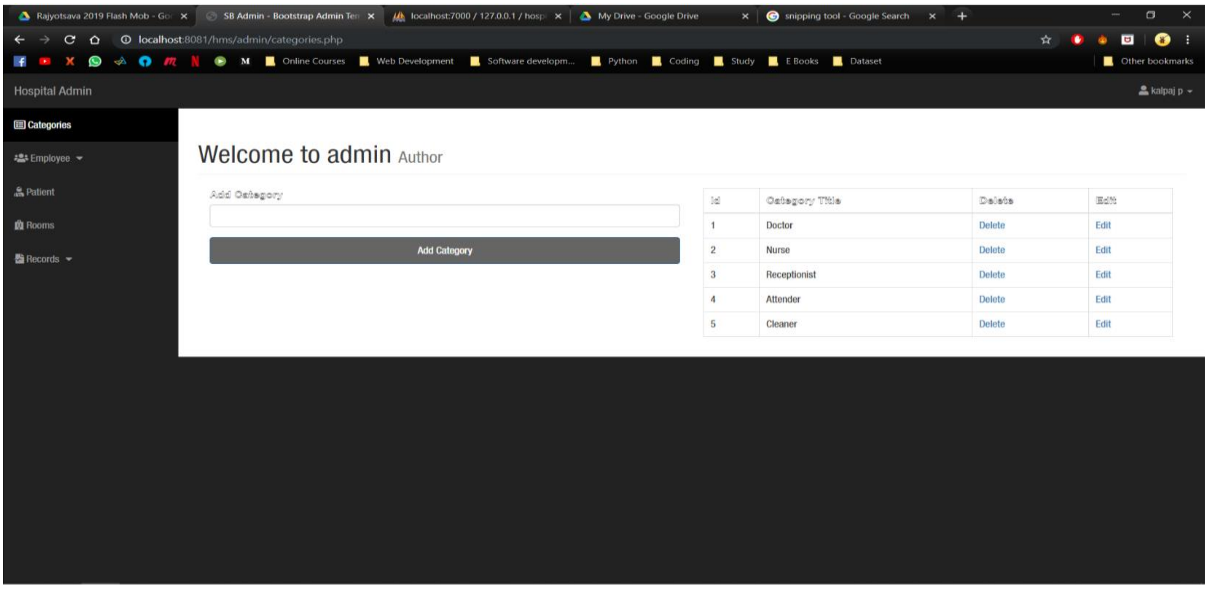
The above Figure 4.2.2 is the snapshot of the Admin Login with the login prompt. If the user has already registered with the website, he can login using his email ID or username which he used to register along with the valid password.



Welcome Admin kalpaj p

Figure 4.5.3: Admin page

The above Figure 4.5.3 is the Dashboard of the admin of the organization, here he has access of all the authority to process.

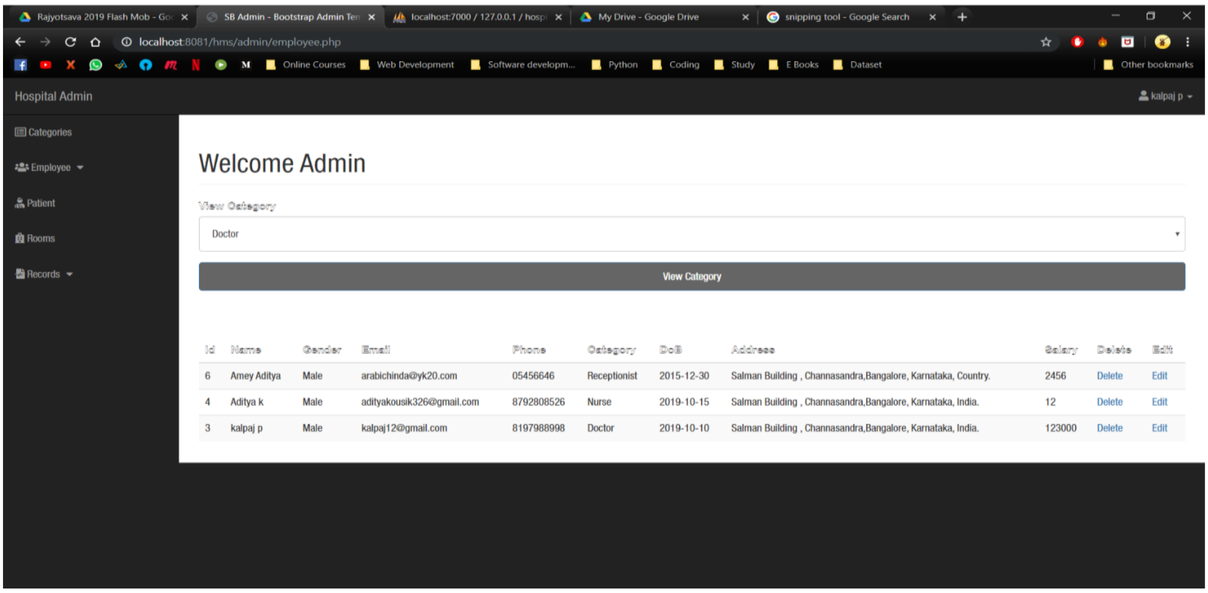


Welcome to admin Author

Id	Category Title	Delete	Edit
1	Doctor	Delete	Edit
2	Nurse	Delete	Edit
3	Receptionist	Delete	Edit
4	Attender	Delete	Edit
5	Cleaner	Delete	Edit

Figure 4.5.4: Admin to add Categories

The Figure 4.5.4 Is the list of categories that the organization has, there is a search bar for the categories as in the high-level organization there might be many categories, to come up with the problem there is a search option. It can even edit the categories and also admin can add the category.



Welcome Admin

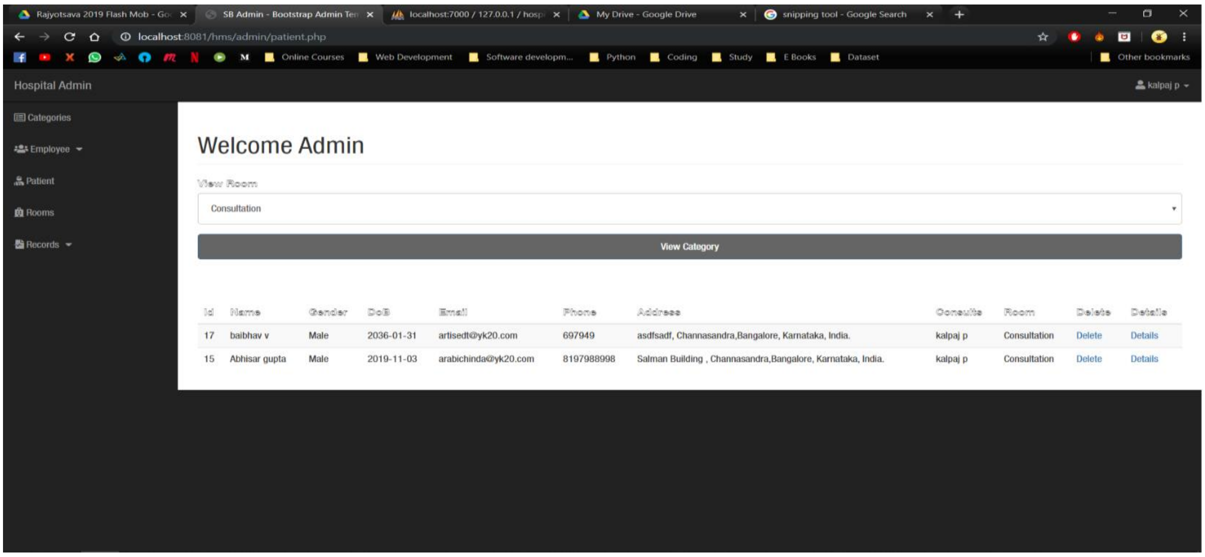
View Category

Doctor

Id	Name	Gender	Email	Phone	Category	DoB	Address	Salary	Delete	Edit
6	Amey Aditya	Male	arabichinda@yk20.com	05456646	Receptionist	2015-12-30	Salman Building , Channasandra,Bangalore, Karnataka, Country.	2456	Delete	Edit
4	Aditya k	Male	adityakousik326@gmail.com	8792808526	Nurse	2019-10-15	Salman Building , Channasandra,Bangalore, Karnataka, India.	12	Delete	Edit
3	kalpaj p	Male	kalpaj12@gmail.com	8197988998	Doctor	2019-10-10	Salman Building , Channasandra,Bangalore, Karnataka, India.	123000	Delete	Edit

Figure 4.5.5: View Employee

The above Figure 4.5.5 shows the list of Employee working in the organization. The enhance the list there is a search bar depending on the categories of Employee.



Welcome Admin

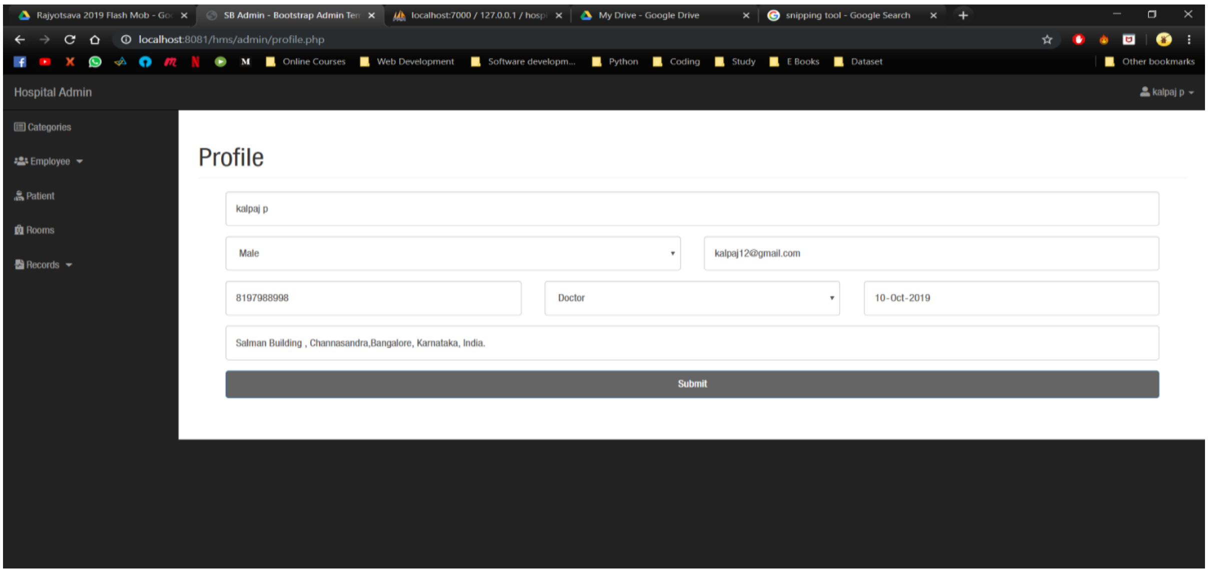
View Room

Consultation

Id	Name	Gender	DoB	Email	Phone	Address	Consults	Room	Delete	Details
17	baibhav v	Male	2036-01-31	artisedt@yk20.com	697949	asdfsadf, Channasandra,Bangalore, Karnataka, India.	kalpaj p	Consultation	Delete	Details
15	Abhisar gupta	Male	2019-11-03	arabichinda@yk20.com	8197988998	Salman Building , Channasandra,Bangalore, Karnataka, India.	kalpaj p	Consultation	Delete	Details

Figure 4.5.6: Patient List

The above Figure 4.5.6 shows the list of Patient who are admitted in the organization. The enhance the list there is a search bar depending on the types of room in the organization.



localhost:8081/hms/admin/profile.php

Profile

kalpaj p

Male

kalpaj12@gmail.com

8197988998

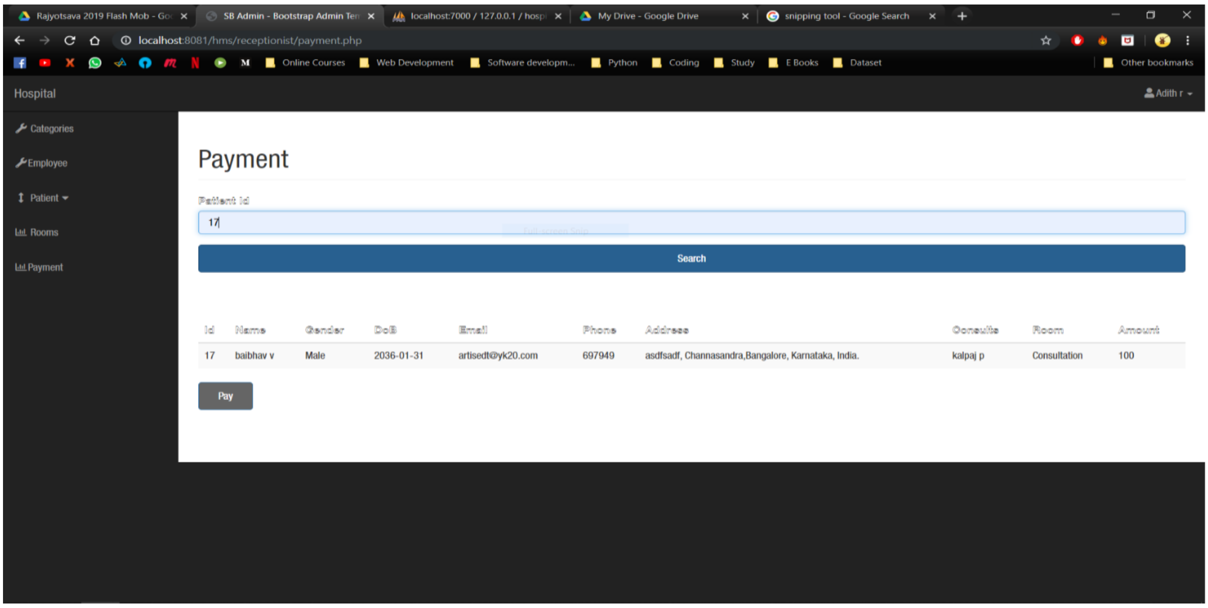
Doctor

10-Oct-2019

Salman Building , Channasandra,Bangalore, Karnataka, India.

Figure 4.5.7: Admin Profile

The above Figure 4.5.7 shows the Details of Employee and has access to change his details.



localhost:8081/hms/receptionist/payment.php

Payment

Patient Id

17

Id	Name	Gender	DoB	Email	Phone	Address	Consults	Room	Amount
17	baibhav v	Male	2036-01-31	artisedt@yk20.com	697949	asdfsadf, Channasandra,Bangalore, Karnataka, India.	kalpaj p	Consultation	100

Figure 4.2.8: Payment

The above Figure ure 4.5.6 shows the payment interface where the Patient pays the charges that are applied for the treatment for his disease. The amount is collected by the Receptionist and handled to the Account section.

CHAPTER 5

APPLICATION

The implementation of hospital management system project provides the institution with different advantages that improve the service quality and efficiency. As mentioned above it is created for three groups of users: patients, hospital staff and management, and third-parties like drug suppliers and insurance companies. The interaction between them conveys the general performance. The benefits received by a certain group of users also positively influence the work of the others. Cooperation and communication are the fundamental requirements here.

5.1 Improved Processes

Automation is one of the main benefits here. It helps to optimize the user experience. Medical specialists, patients, and hospital authorities can interact online, make the appointments and exchange information.

5.2 Digital medical records

The hospital database includes all the necessary patient data. The disease history, test results, prescribed treatment can be accessed by doctors without much delay in order to make an accurate diagnosis and monitor the patient’s health. It enables lower risks of mistakes.

5.3 Staff interaction

It is vital to engage all of your employees for improved coordination and teamwork. They do not need to make special requests and wait for a long time for an answer. Each specialist will be in charge of certain process stage and can share outcomes with colleagues just in one click.

5.4 Facility management

Hospitals authorities are able to manage their available resources, analyze staff work, reduce the equipment downtime, optimize the supply chain, etc. Another fact to mention is that hospital staff deal with the digital data instead of endless paperwork.

5.5 Patient self-service

Patients have their own system accounts where the list of various actions can be performed. They are able to make online requests or reservation, receive the test results, receive the consultation of the medical specialists and many more.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

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.

The Future enhancement for the project as are follows:

- Developing Chatbot Interface.
- Creating Virtual Interface for the communication of Doctor and Patients.
- Hosting the Web Server

REFERENCES

[1] Michael.J. Folk, Bill Zoellick, Greg Riccardi: File Structures-An Object-Oriented Approach with C++, 3rd Edition, Pearson Education, 1998.

[2] Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, McGrawHill, 2013.

[3] Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGRAW HILL, 3rd Edition.

[4] www.stackoverflow.com

[5] www.geeksfoegeeks.org

[6] www.w3schools.com