

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018



A Mini Project Report on

## HOSPITAL MANAGEMENT SYSTEM

*Submitted in partial fulfillment of the requirements as a part of the DBMS Lab for the V Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi*

Submitted by

**MANIKANTA Y A**  
**1RN17IS052**

**KALPAJ PISE**  
**1RN17IS039**

Under the Guidance of

**Faculty Incharge**  
**Mr. Ramesh S Chakrasali**  
Assistant Professor  
Dept. of ISE, RNSIT

**Lab Incharge**  
**Mr. R Rajkumar**  
Assistant Professor  
Dept. of ISE, RNSIT



ESTD:2001  
*An Institute with a Difference*

**Department of Information Science and Engineering**  
**RNS Institute of Technology**  
Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,  
Bengaluru – 560 098

**2019 – 2020**

# RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, RR Nagar Post,  
Bengaluru – 560 098

## DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING



### CERTIFICATE

This is to certify that the mini project report entitled ***THE HOSPITAL MANAGEMENT*** has been successfully completed by **MANIKANTA Y A** bearing USN **1RN17IS052** and **KALPAJ PISE** bearing USN **1RN17IS039**, presently V semester student of **RNS Institute of Technology** in partial fulfillment of the requirements as a part of the DBMS Laboratory for the award of the degree ***Bachelor of Engineering in Information Science and Engineering*** under **Visvesvaraya Technological University, Belagavi** during academic year 2019 – 2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements as a part of DBMS Laboratory for the said degree.

\_\_\_\_\_  
**Mr. Ramesh S Chakrasali**  
Faculty Incharge

\_\_\_\_\_  
**Mr. R Rajkumar**  
Lab Incharge

\_\_\_\_\_  
**Dr. M V Sudhamani**  
Professor and HOD

#### External Viva

Name of the Examiners

Signature with date

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_

# DECLARATION

We, **MANIKANTA Y A [USN: 1RN17IS052]** and **KALPAJ PISE [USN: 1RN17IS039]** student of V Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Mini Project work entitled “**HOSPITAL MANAGEMENT SYSTEM**” has been carried out by us and submitted in partial fulfillment of the requirements for the *V Semester degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi* during academic year 2019-2020.

Place : Bengaluru

Date :

**MANIKANTA Y A (1RN17IS052)**

**KALPAJ PISE (1RN17IS039)**

# **ABSTRACT**

The purpose of the project entitled as “HOSPITAL MANAGEMENT SYSTEM” is to computerize the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details, while system output is to get these details on to the screen. 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. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

# ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management of RNS Institute of Technology** for providing such a healthy environment to carry out this Project work.

We would like to thank our beloved Director **Dr. H N Shivashankar** for his confidence feeling words and support for providing facilities throughout the course.

We would like to express my thanks to our Principal **Dr. M K Venkatesha** for his support and inspired me towards the attainment of knowledge.

We wish to place on record my words of gratitude to **Dr. M V Sudhamani**, Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind my Project work.

We would like to express my profound and cordial gratitude to my Faculty incharge **Mr. R Rajkumar**, Assistant Professor, Department of Information Science and Engineering for his valuable guidance, constructive comments and continuous encouragement throughout the Project work.

We would like to express my profound and cordial gratitude to my guide **Mr. Ramesh S Chakrasali**, Assistant Professor, Department of Information Science and Engineering for his valuable guidance in preparing Project report.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped me to carry out the project work.

And lastly, we would hereby acknowledge and thank my parents who have been a source of inspiration and also instrumental in carrying out this Project work.

**MANIKANTA Y A**  
**USN: 1RN17IS052**

**KALPAJ PISE**  
**USN:1RN17IS039**

# TABLE OF CONTENTS

<b>CERTIFICATE</b>	
<b>DECLARATION</b>	<b>i</b>
<b>ABSTRACT</b>	<b>ii</b>
<b>ACKNOWLEDGMENT</b>	<b>iii</b>
<b>TABLE OF CONTENTS</b>	<b>iv</b>
<b>LIST OF FIGURES</b>	<b>v</b>
<b>ABBREVIATIONS</b>	<b>vi</b>
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Background	1
1.2 Introduction to Hospital Management	1
<b>2. E R DIAGRAM AND RELATIONAL SCHEMA DIAGRAM</b>	<b>3</b>
2.1 Description of ER Diagram	3
2.2 Description of Relational Schema Diagram	5
<b>3. SYSTEM DESIGN</b>	<b>8</b>
3.1 Table Description	8
3.2 Stored Procedure	11
3.3 Triggers	11
<b>4. IMPLEMENTATION</b>	<b>12</b>
4.1 Front-end Development	12
4.2 Back-end Development	14
4.3 Discussion of code Segment	16
4.4 System Requirement	17
4.5 Result	18
<b>5. APPLICATION</b>	<b>22</b>
<b>6. CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>23</b>
<b>REFERENCES</b>	<b>24</b>

## LIST OF FIGURES

<b>Figure No.</b>	<b>Description</b>	<b>Page No.</b>
Figure 2.1	ER Diagram	03
Figure 2.2	Schema Diagram	04
Figure 4.1	Home Page	18
Figure 4.2	Login Page	18
Figure 4.3	Admin Page	19
Figure 4.4	Admin To Add Categories	19
Figure 4.5	View Category	20
Figure 4.6	Consultation	20
Figure 4.7	Admin Profile	21
Figure 4.8	Payment	21

# ABBREVIATIONS

CSS	-	Cascading style sheets
GPL	-	General Public License
HTML	-	Hypertext Markup Language
ID	-	Identification
JS	-	JavaScript
PHP	-	PHP Hypertext Preprocessor
SQL	-	Structured Query Language
TB	-	Terabyte



# 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

database. Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

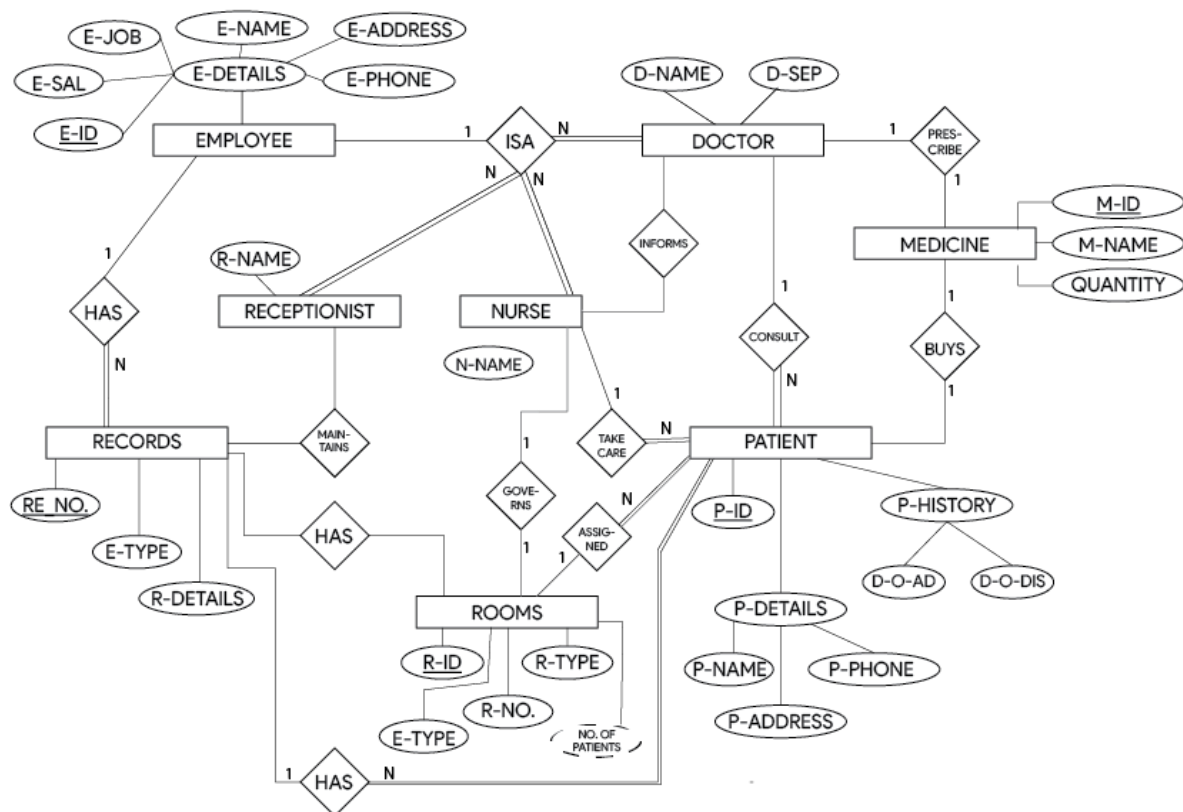
Hospital Management System is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient 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**.
6. There is a partial participation of **PATIENT** and partial participation of **MEDICINE** as:
  - All Doctor prescribe required Medicine for Patient
  - All prescribed medicine is bought by the Patient

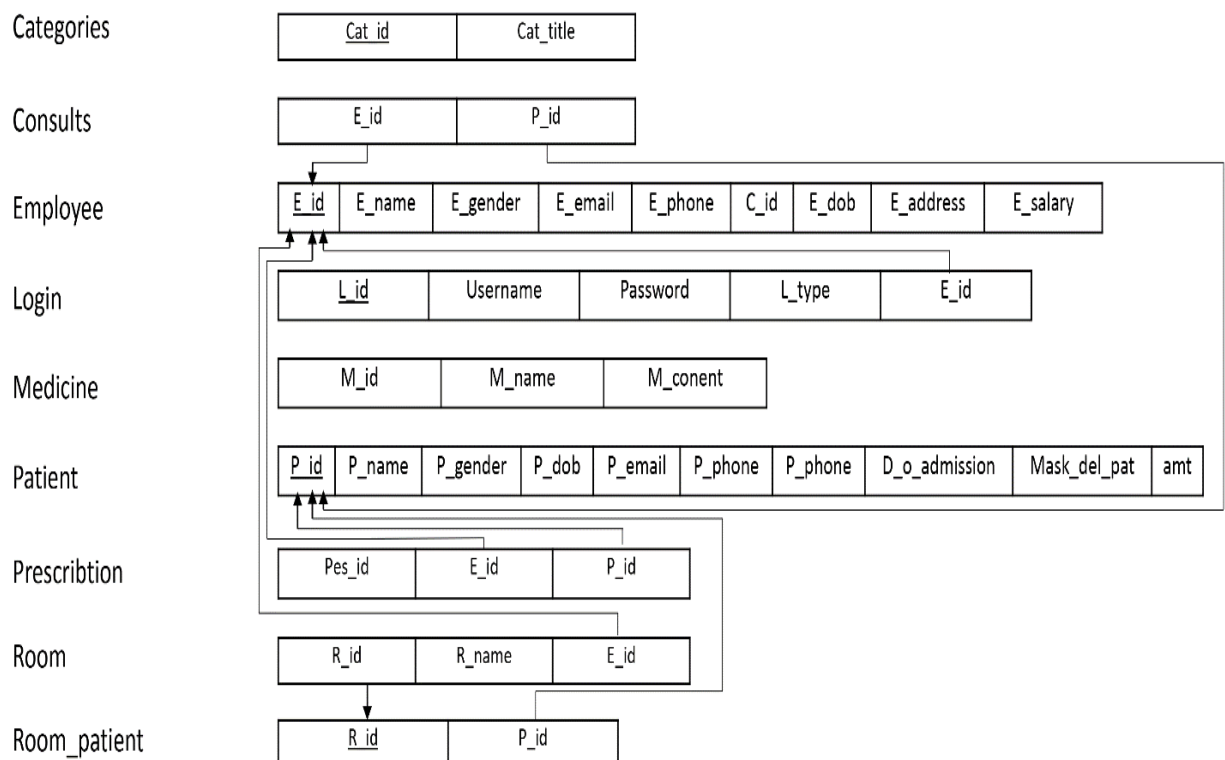
7. **ROOM: PATIENT** is of cardinality ratio of 1: N. Where 1 **PATIENT** can be assigned to a single room and N **ROOM** are allocated for the different Patients. They are connected via **ASSIGNED**. There is a partial participation of **PATIENT** and the **ROOMS** as:

- Multiple Rooms aren't allocated for a single Patient
- Every Employee is assigned as a room in charge.

8. **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. **PRESCRIPTION:** 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



The Table 3.3 contains the list of all contact details of the Employee. It has a total of 9 attributes, E\_ID, E\_name, E\_gender, E\_email, E\_phone, C\_Id, E\_dob, E\_address, E\_salary. 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. Its elements tell the browser how to display the content. Its 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. CSS 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 the servers used today. It is easy to use and runs efficiently on the server side. It can run various platforms like windows, Linux, UNIX, Mac OS-X etc.

### **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_1}' 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_1 === $username && $password_1 === $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['phonenumner'];
    $emp_category_id = escape($_POST['category']);
    $emp_dob       = $_POST['dob'];
    $emp_salary    = $_POST['salary'];
    $emp_address   = escape($_POST['address1']);
    $query = " UPDATE employee SET " ;
    $query .= "e_name    = '{$emp_name}' , " ;
    $query .= "e_gender  = '{$emp_gender}' , " ;

```



```
$query .= "e_email = '{$Semp_email}'," ;  
$query .= "e_phone = '{$Semp_phone}'," ;  
$query .= "c_id = {$Semp_category_id} ," ;  
$query .= "e_dob = '{$Semp_dob}'," ;  
$query .= "e_address = '{$Semp_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)";  
  
$query.=VALUES('{$Semp_name}','{$Semp_gender}','{$Semp_email}','{$Semp_phon  
e}','{$Semp_category_id}','{$Semp_dob}','{$Semp_address}','{$Semp_salary}') ";  
$create_post_query = mysqli_query($connection, $query);
```

## 4.4 System Requirement

To run this project the system should possess with the basic requirements which are mentioned below.

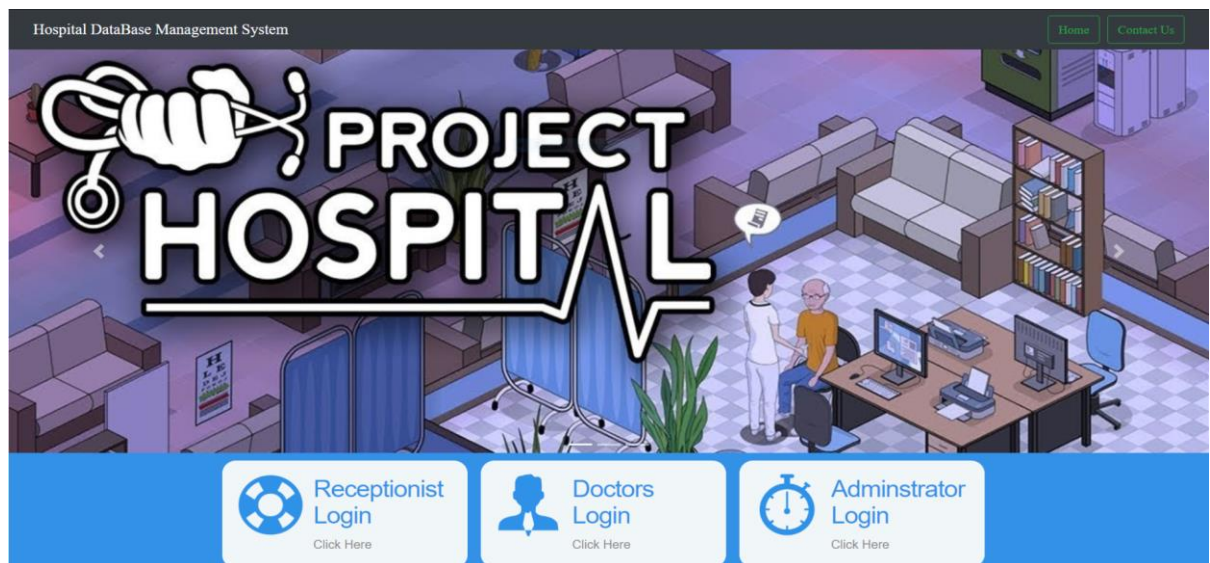
### 4.4.1 Hardware Requirements

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

### 4.4.2 Software Requirements

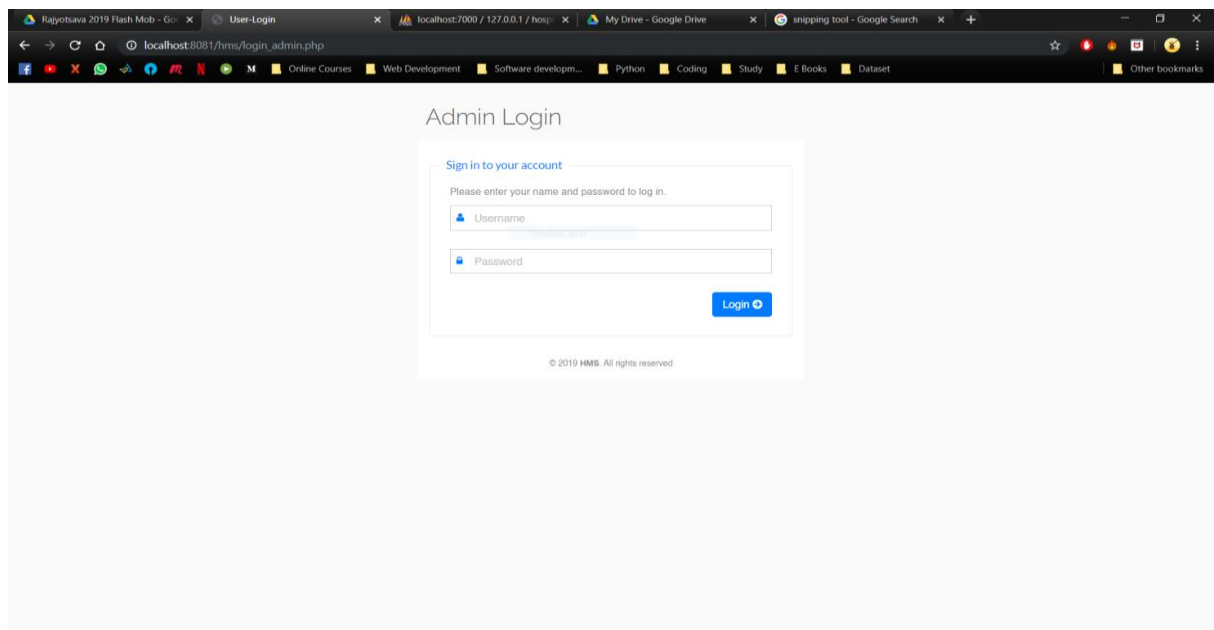
- OS: WINDOWS OS (XP / 2000 / 2003 Server/windows10)
- Server: XAMPP (Apache)
- Front End: PHP, HTML, CSS
- Back End: MySQL/phpMyAdmin)

## 4.5 Results



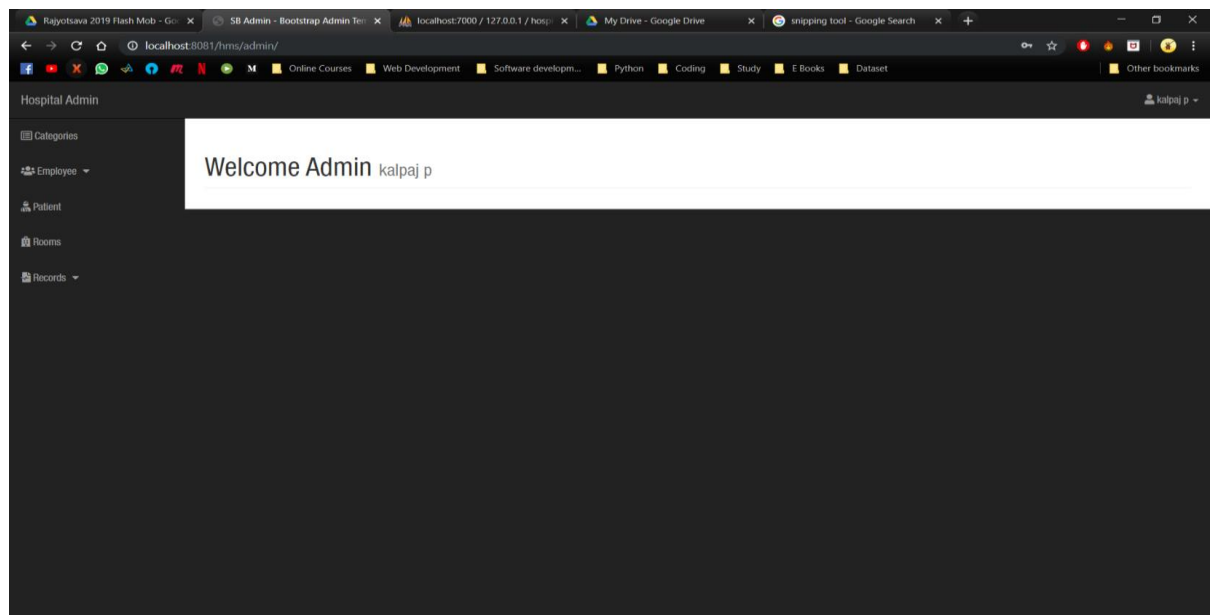
**Figure 4.1: Home page**

The Above Figure 4.1 is the snapshot of the Home page is the starting page of the Project where it runs from the localhost i.e. 192.172.0.1/homepage.html



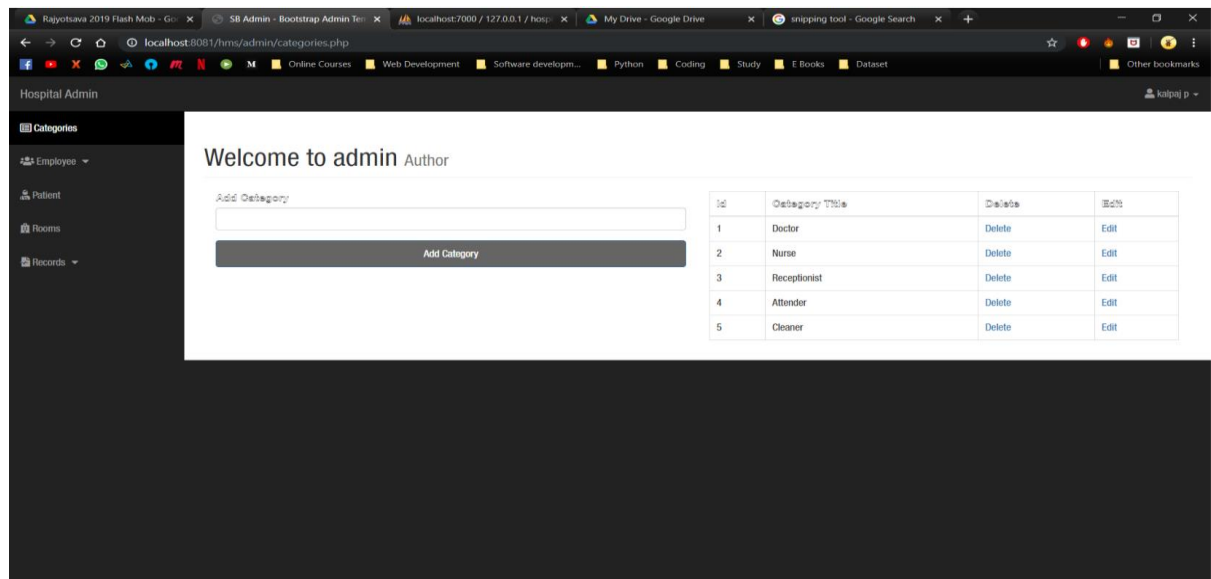
**Figure 4.2: Admin login**

The above Figure 4.2 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.



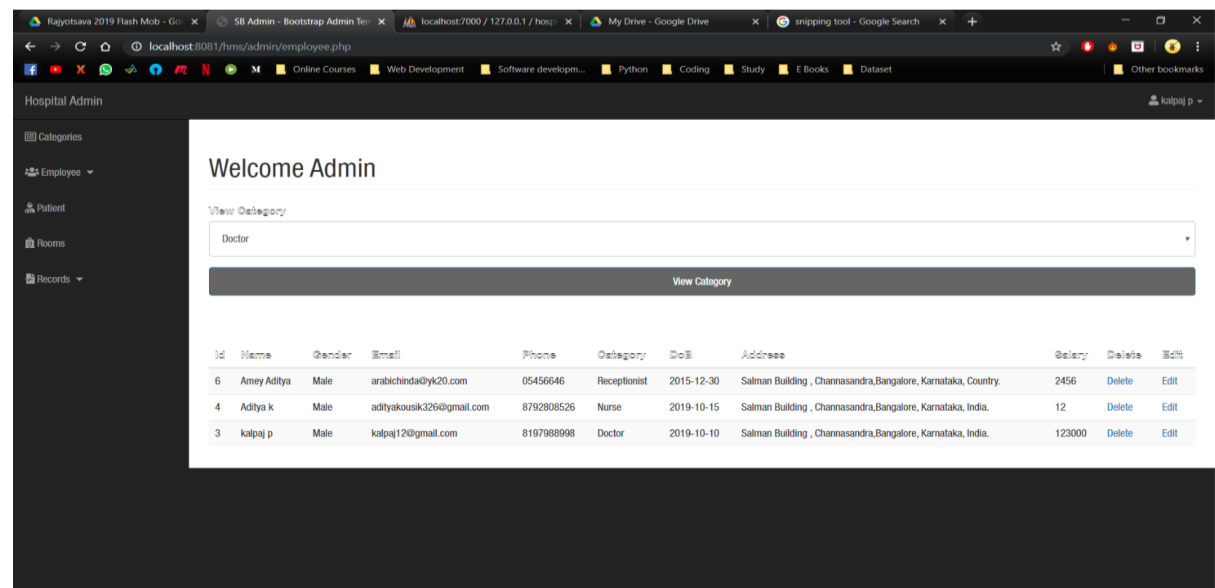
**Figure 4.3: Admin page**

The above Figure 4.3 is the Dashboard of the admin of the organization, here he has access of all the authority to process.



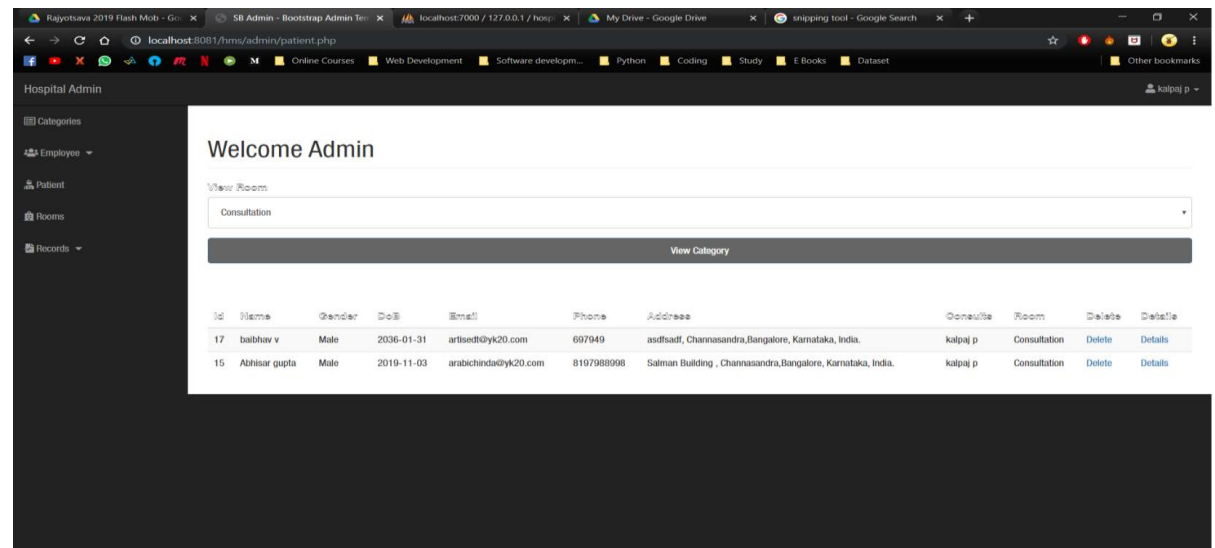
**Figure 4.4: Admin to add Categories**

The Figure 4.4 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.



**Figure 4.5: View Employee**

The above Figure 4.5 shows the list of Employee working in the organization. The enhance the list there is a search bar depending on the categories of Employee.



**Figure 4.6: Patient List**

The above Figure 4.6 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.

**Profile**

kalpaj p

Male

kalpaj12@gmail.com

8197988998

Doctor

10-Oct-2019

Salman Building , Channasandra, Bangalore, Karnataka, India.

Submit

**Figure 4.7: Admin Profile**

The above Figure 4.7 shows the Details of Employee and has access to change his details.

**Payment**

Patient Id

17

Search

Id	Name	Gender	DOB	Email	Phone	Address	Consults	Room	Amount
17	baibhav v	Male	2036-01-31	artisedi@yky20.com	607949	asdfsadf, Channasandra, Bangalore, Karnataka, India.	kalpaj p	Consultation	100

Pay

**Figure 4.8: Payment**

The above Figure 4.8 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 ENHANCEMENTS

The system stores all the 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.

The project consists of all the relation between the Patient and Doctor. There are Categories of Employee who work for the organization. They might be Doctor, Nurse, etc who treats the Patient and every record is stored into the Database. The activities of the Employee towards the Patient are store into the Database. It easily reduces the book keeping task and thus reduces the human effort and increases accuracy speed.

The Future enhancement for the project is as 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](http://www.stackoverflow.com)
- [5] [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
- [6] [www.w3schools.com](http://www.w3schools.com)