

CHAPTER 1

INTRODUCTION

Hospital Management System is a system enabling hospitals to manage information and data related to all aspects of healthcare – processes, providers, patients, and more, which in turn ensures that processes are completed swiftly and effectively. When one thinks of the various aspects and departments of a hospital, it becomes apparent that an HMS is critical. The hospital database management system was introduced in 1960, and has greatly evolved since then – with the ability to integrate with the existing facilities, technologies, software, and systems of a hospital. Today, patients can begin the process of healthcare in the palm of their hand – the mobile devices and apps – make this possible. This process then moves to the healthcare providers and hospitals.

In the following sections, a brief introduction about the tools, languages and the databases used to develop the project are discussed.

1.1 HTML

HTML, which stands for Hyper Text Mark-Up Language, is the language for describing structured documents as well as the language used to create web pages in the Internet. The language is based on an existing, international formatting standard SGML, Standard Generalized Mark-Up Language, which is used for text processing.

HTML documents are nothing but web pages which contains HTML tags and plain text. The purpose of a web browser is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

History

HTML, which stands for Hyper Text Mark-Up Language, is the language for describing structured documents as well as the language used to create web pages in the Internet. The language is based on an existing, international formatting standard SGML, Standard Generalized Mark-Up Language, which is used for text processing. HTML is a simplified version of SGML.

Tools in order to use HTML

Tools help us in process of creating HTML document. Some are as follows

- **TEXT EDITOR:** To create the HTML code we require a text editor or a word processor. Such as, Notepad, WordPad. We are using notepad++ in developing this project.
- **WEB BROWSER:** The code created by an editor should be executed. This operation can be performed with help of a web browser. Such as Internet Explorer, Netscape navigator, Mozilla Firefox etc.
- **GRAPHICS SOFTWARE:** To include picture we require a graphic software like Adobe Photoshop.
- **Web server:** To make the document is to be available on the internet then, we will have to host it on a web server.

1.1.1 Significant Language Features

HTML files are written in ASCII text, so the user can use any text editor to create his/her web page, though a browser of one sort or another is necessary to view the web page. HTML is case insensitive with its language commands. The characters within the document, however, are case sensitive. The language consists of various "tags" which are known as elements. These allow the browser to understand (and put into the desired/specified format) the layout, background, headings, titles, lists, text and/or graphics on the page. The elements are classified according to their function in the HTML document. There are head elements and body elements. The head elements identify properties of the entire document, while body elements actually mark text as content and show a change in the appearance in one way or another. Most elements have a beginning and an ending which encompass the text the user wishes to mark with the tag. All HTML documents must begin with the element and end with the element. Some of the other elements which may be used are tags to create lists--both ordered lists as well as unordered lists. The user may also create larger or smaller, bolder, italicized, or underlined text. Attributes may be used along with the elements. These perform functions such as placement of text, indication of the source files of images, and identification of links to the document or part of the document.

1.1.2 HTML Code

Copy and paste the following HTML code into your newly open text file. Which just displays hello world..

```
<html>
<header><title>This is title</title></header>
<body>
  This is sample text...
  <!-- We use this syntax to write comments -->
  <!-- Page content and rest of the tags here.... -->
  <!-- This is the actual area that gets shown in the browser →
Hello world
</body>
</html>
```

1.1.3 HTML TAGS

HTML tags are keywords surrounded by angle brackets like <html>. These are in pair format such that every first tag in pair is start tag where as second tag is end tag. These start and end tags are also called as opening tags and closing tags respectively.

Tags Used In Project

The HTML tags are the basis, in order to do this Project. By using some of the important and basically taught tags are used in this Project. Here are some of the tags used in making the Project called AUTOMOBILE MANAGEMENT SYSTEM.

HTML Attributes

Attributes provide additional information about HTML elements.

- HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes come in name/value pairs like: **name="value"**

Some basic text formatting HTML tags are listed:

| Tag | Description |
|--------------|------------------------------|
| <html> | Defines an HTML document |
| <body> | Defines the document's body |
| <h1> to <h6> | Defines header 1 to header 6 |
| <p> | Defines a paragraph |
| | Inserts a single line break |
| | Defines bold text |
| <!--> | Defines a comment |
| <small> | Defines small text |

Some of the HTML tags used to create a table are listed:

In an HTML file we can create tables with the Table tags, which in turn will render the browser to display the table in the web page.

| Tag | Description |
|---------|------------------------|
| <table> | Defines a table |
| <th> | Defines a table header |
| <tr> | Defines a table row |
| <td> | Defines a table cell |
| <tbody> | Defines a table body |
| <tfoot> | Defines a table footer |

A Simple Form

A form in a web page allows the users to input various data online. In an HTML document; forms can be created with the Form tags. In the following table, some basic Form tags are listed:

| Tag | Description |
|---------|-------------------------------|
| <form> | Defines a form for user input |
| <input> | Defines an input field |

| | |
|------------|-----------------------------------|
| <textarea> | Defines a text-area |
| <label> | Defines a label to a control |
| <fieldset> | Defines a fieldset |
| <legend> | Defines a caption for a fieldset |
| <select> | Defines a selectable list |
| <optgroup> | Defines an option group |
| <option> | Defines an option in the drop box |
| <button> | Defines a push button |

Image Tags

In an HTML document we can insert and display images by using the image tags.

In the following table, some basic Image tags are listed:

| Tag | Description |
|----------------------|------------------|
| | Defines an image |

The “src” attribute is used to display an image on a web page. “src” stands for “source”, and its value is the url of the image to be displayed on the page. The url indicates the location where the image is stored. Attributes may be height, width, align so on.

Background colour

Using bgcolor attribute this can be done. This is body tag attribute. Six digit hexadecimal code represent the colours.

Syntax: <body text=”text_color” bgcolor = ”background_color”>

Anchor tag

Anchor tag is used to link two or more different web pages.

Ex: click here where href stands for hyper link reference.

Areas of Application

HTML only has one area of application at this time and that is the development of web pages. However, not all browsers support all the tags in all versions of HTML. Because of this, it is wise not to design your web page for a specific browser, because what may look fantastic on your browser has no guarantee of looking great on someone else's browser.

1.2 PHP

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere.

PHP originally stood for Personal Home Page, but it now stands for the recursive backronym PHP. Hypertext Pre-processor. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks.

PHP developer

PHP developers develop programs, applications, and web sites using the dynamic scripting language PHP. PHP is known for web development and business applications. Depending on job function, PHP developers may be classified as software developers or web developers.

Tags Description

<?php to open PHP section

?> to close PHP sections

ECHO prints the lines

1.3 DATABASE

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images. **Database** software systems are programmed in SQL, and examples include Microsoft SQL Server, MySQL, Oracle SAP HANA and FoxPro.

A DBMS system is also required to protect the integrity of data and provide its security. A database management system (**DBMS**) is system software for creating and managing databases. The **DBMS** provides users and programmers with a systematic way to create, retrieve, update and manage data.

1.4 MYSQL

MySql is a powerful database. It's very good and free of charge. Many developers in the world selected mysql and php for developing their website.

The MySQL database has become the world's most popular open source database because of its consistent fast performance, high reliability and ease of use. It's used in more than 6 million installations ranging from large corporations to specialized embedded applications on every continent in the world. (Yes, even Antarctica!)

Not only is MySQL the world's most popular open source database, it's also become the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) MySQL runs on more than 20 platforms including Linux, Windows, OS/X, HP-UX, AIX, Netware, giving you the kind of flexibility that puts you in control.

Whether you're new to database technology or an experienced developer or DBA, MySQL offers a comprehensive range of certified software, support, training and consulting to make you successful.

1.5 WAMP

The acronym WAMP refers to a set of free (open source) applications, combined with Microsoft Windows, which are commonly used in Web server environments. The WAMP stack provides developers with the four key elements of a Web server: an operating system, database, Web server and Web scripting software. The combined usage of these programs is called a server stack. In this stack, Microsoft Windows is the operating system (OS), Apache is the Web server, MySQL handles the database components, while PHP, Python, or PERL represents the dynamic scripting languages.

1.6 Notepad++

Notepad++ is a text editor and source code editor for use with Microsoft Windows. Unlike Microsoft Notepad, the built-in Windows text editor, it supports tabbed editing, which allows working with multiple open files in a single window. The project's name comes from the C increment operator.

Notepad++ is distributed as free software. At first the project was hosted on SourceForge.net, from where it has been downloaded over 28 million times.

1.7 Web Browser

Google Chrome is a free web browser from Google which we are using here. With its clean design and advanced features, Chrome has quickly become one of the most popular web browsers worldwide. In this lesson, we'll talk about the features of Google Chrome, how to download and install Chrome to your computer, and how to sign in to Chrome using a Google account.

CHAPTER 2

System Analysis and Design

In this chapter, a complete description of the project development is discussed. The requirements of the project identified are showcased. The database design is done Using High-Level Conceptual Data Models

2.1 Requirement Analysis

Following requirements were identified during the requirement collection and analysis.

Hardware requirements:-

- i. A device (Computer/laptop)
- ii. Memory (RAM): Minimum 2GB RAM
- iii. Processor: Minimum 1GHZ; Recommended 2GHZ or more.
- iv. Hard disk – 40 GB; Recommended 64 GB or more.
- v. Ethernet connection (LAN) or, a wireless adapter (Wi-Fi)

Software Requirements :-

- i. A database like DBMS to store the list of authors and the articles. ii.
- A web browser like Chrome, Mozilla Firefox etc.
- iii. Operating System – Windows, Linux, macOS 32 bit and 64 bit

Feasibility Study

The feasibility study carried out showed that the requirements that were to be included could be provided by the use of RDBMS software such as MySQL which is available as an open source and for the front end HTML pages with processing capability provided by the Scripting language such as PHP and Javascript.

2.2 ER-Diagram

Following is the conceptual representation of the requirements identified as an ER-Diagram

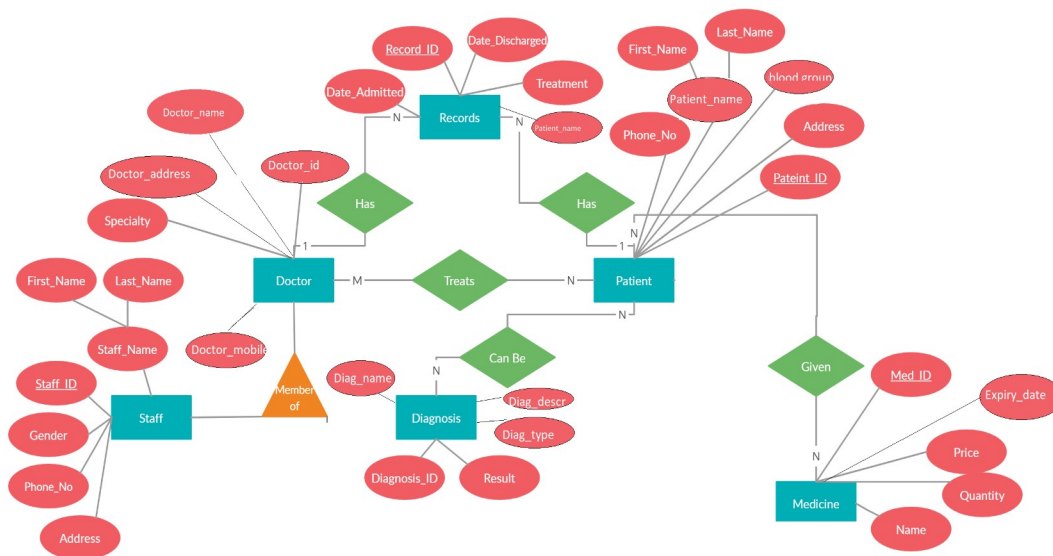
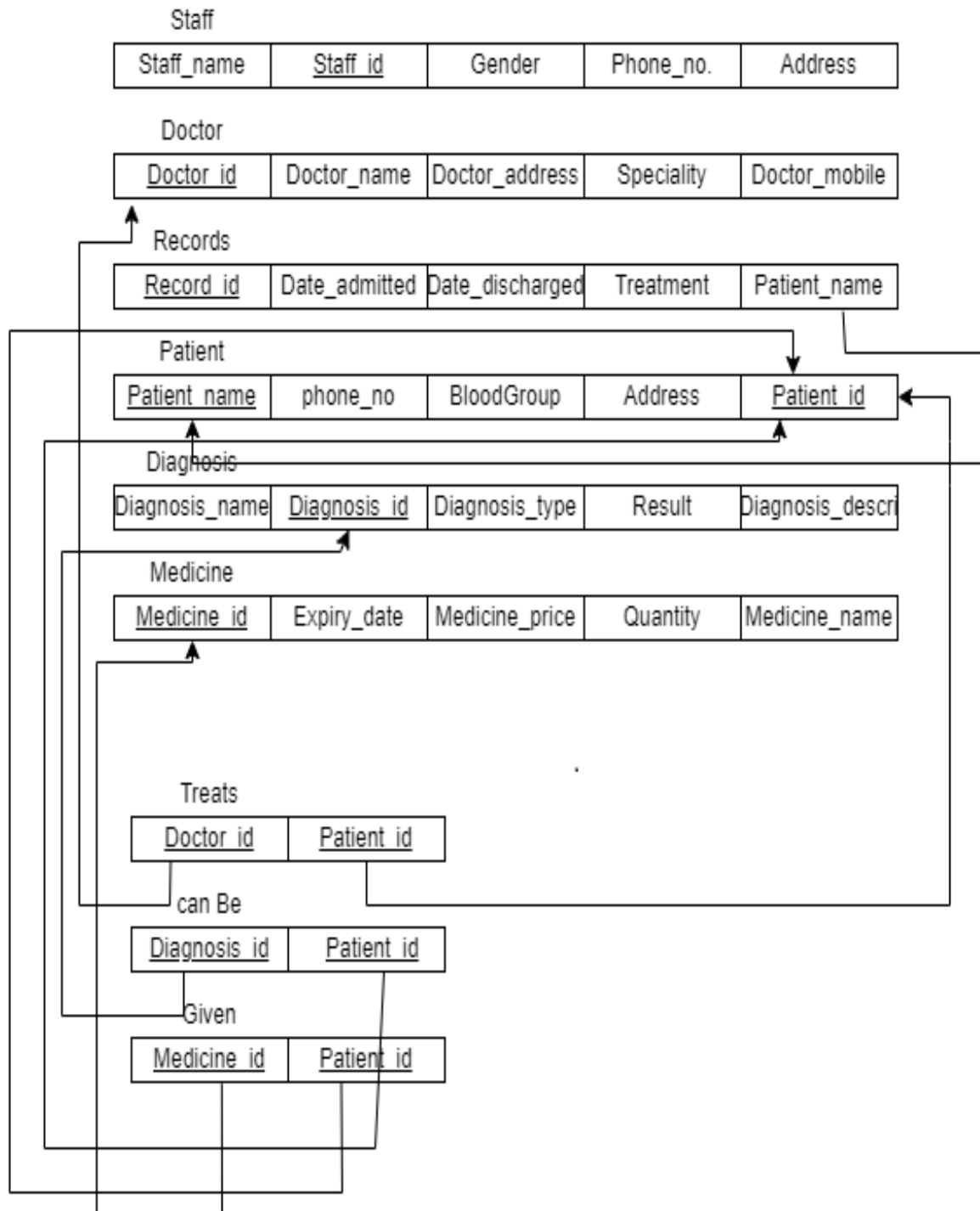


Figure 2.1 ER Diagram for Hospital management system.

2.3 Relational Schema

The relational schema diagram has been derived from the ER-Diagram in Figure 2.1 using the ER-Relational mapping algorithm

Figure 2.2 Relational Schema Diagram Showing the Primary key and Foreign key relationships



2.4 Functional Requirements

Functional requirements of a software project interpret the function of a part. It defines its functions, input and output. The typical functional requirements include:

Application contains 3 modules:

- Receptionist module
- Patient module
- Doctor module

Receptionist module

- Receptionist can able to register the Patient information.
- Receptionist can able to delete Patient information.
- Receptionist can able to create appointment for patient.
- Receptionist can able to update appointment for patient.
- Receptionist can able to see appointment created.

Patient module

- Patient must be to register to the application
- Patient can be able to book the appointment.
- Patient can able to see the specialities of hoapital.
- Patient can able to view the doctors available for a specific speciality.
- Patient can able to book more than one appointments.
- Patient can able to select the timings avaialable for appointment.
- Patient can be able to select the dates he wants for appointment.
- Patient can be able to view the about of hospital and can contact by the help of contact functionality.

Doctor module

- Doctor can be able to view the appointments.
- Doctor can be able to see the gender of the patient.

- Doctor can be view the problems and disease of the patient through appointment.

2.5 Non- Functional Requirements

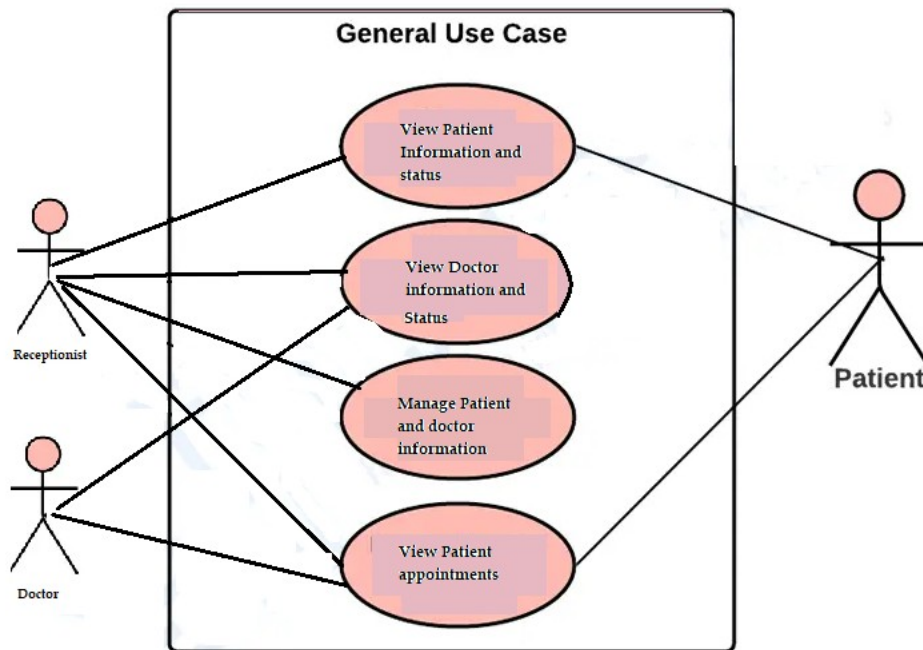
A non-functional requirement specifies the canon of the articular process not the particular judgment of the system and particular behavior of the process. Non-functional requirements define how the system works.

- This application is developed to make booking of appointment easier to patient so that it will save the time.
- This application work efficiently it works on all logical paths and independently and it should use the mobile data efficiently.
- .This application is available during all the time at booking of appointment.
- This application reduces all the complexity during appointment booking.
- To run this application efficiently mobile network is the main important factor.
- Using of application is secure, because it display appropriate information about the patient, doctor and receptionist.
- The system should capable to enhance with further technology in future to improve its features compared to the existing system such as offline booking.
- The system should be reliable and it should be related in all the condition and it should be recoverable in all the situation or condition if error occurs.

2.6 Use Case Diagram

The use case diagrams usually refer to behavioral diagrams helps people to understand the interaction between user and system. Use case diagram identify different users of the system.It is used to define some set of actions, which is called as use cases. Actors are the result of some valuable use cases.Use case figures are also called as unified modeling language.

Figure 2.4
Use Case
Diagram



CHAPTER 3

SYSTEM IMPLEMENTATION

3.1 Database Design

```
create table staff(staff_name varchar(20), staff_id int, Gender
varchar(5), phone_no int, address varchar(40) ,
constraint staff_Pk primary key(staff_id));
```

```
create table doctor(doctor_id int, doctor_name
varchar(20), doctor_address varchar(20), speciality
varchar(20), doctor_mobile int,
constraint doctor_Pk primary key(doctor_id));
```

```
create table records(record_id int, date_submitted
date, date_discharged date, treatment varchar(20), patient_name
varchar(20),
constraint records_Pk primary key(record_id),
```

```
constraint record_fk foreign key(patient_name) references
patient(patient_name));
    create table patient(patient_name varchar(20), phone_no int,
blood_group varchar(5), address varchar(30), patient_id int,
constraint patient_Pk primary key(patient_id));
    create table diagnosis(diagnosis_name varchar(20), diagnosis_id int,
diagnosis_type varchar(20), result varchar(30), diagnosis_description
varchar(50),
constraint diagnosis_Pk primary key(diagnosis_id));
    create table medicine(medicine_id int, expiry_date date,
medicine_price int, quantity int, medicine_name varchar(20),
constraint medicine_Pk primary key(medicine_id));
```

3.2 Database Connectivity

```
<?php

$servername='localhost';
$username='root';
$password='';
$dbname='hms';
$conn =new mysqli($servername, $username, $password, $dbname);
if($conn -> connect_errno){
    echo "Failed to connect to MYSQL" , $mysqli-> connect_error;
    exit();
}

else {

echo "connected to Database";
}
?>
```

3.3 Implementation of Database Operations

For insertion:-

```
<?php

$servername='localhost';
$username='root';
$password='';
$dbname='hms';
$conn =new mysqli($servername, $username, $password, $dbname);
if($conn -> connect_errno){
    echo "Failed to coneect to MYSQL" , $mysqli-> connect_error;
    exit();
}

else {

echo "connected to Database";
echo "<br>";
$ss=$_POST['select_speciality'];
$sd=$_POST['select_doctor'];
$dat=$_POST['date'];
$st=$_POST['select_timing'];
$insertsql="INSERT INTO `appointment_details`(`specialization`, `doctor`,
`apt_date`, `apt_time`) VALUES ('$ss','$sd','$dat','$st')";
echo $insertsql;

if($conn->query($insertsql)==TRUE)
{
    echo "<script>
    alert('appointment booked successfully');
    window.location.href='book_appointment.html';
    </script>";
}
else
{
    echo "<script>
    alert('appointment booking failed');
    window.location.href='book_appointment.html';
    </script>";
}
}
?>
```

For retrieval:-

```
<?php

$servername='localhost';
$username='root';
$password='';
$dbname='hms';
$conn =new mysqli($servername, $username, $password, $dbname);
if($conn -> connect_errno){
    echo "Failed to coneect to MYSQL" , $mysqli-> connect_error;
```



```
        exit();
    }

    else {
        echo "<br>";
        $apt_id=$_POST['apt_id'];
        $retrievesql="SELECT * FROM `appointment_details` WHERE apt_id = $apt_id";
        $result=$conn->query($retrievesql);
        if($result->num_rows>0)
        {
            echo "retrieved data are";
            echo "<table border=1>";
            echo "<tr><th>Specialization</th><th>Doctor</th><th>Appointment
Date</th><th>Appointment Time</th><th>Appointment ID</th><tr>";
            while($row=$result->fetch_assoc())
            {
                echo "<tr><td>";
                echo $row['specialization'];echo "</td><td>";
                echo $row['doctor'];echo "</td><td>";
                echo $row['apt_date'];echo "</td><td>";
                echo $row['apt_time'];echo "</td><td>";
                echo $row['apt_id'];echo "</td><td>";
                echo "</tr>";
            }
            echo "</table>";
        }
    }
    else
    {
        echo "failed to retrieve the data ,unknow appointment id";
    }
}
?>
```

For Updation:-

```
<?php
$servername='localhost';
$username='root';
$password='';
$dbname='hms';
$conn = new mysqli($servername, $username, $password, $dbname);
if(!$conn){
die('Could not Connect MySql:' .mysql_error());
}
else
{
    echo "connected to db";
    $spec=$_POST['select_speciality'];
    $doct=$_POST['select_doctor'];
    $date=$_POST['date'];
    $aid=$_POST['apt_id'];
    $time=$_POST['select_timing'];
```

```
$sql = " UPDATE `appointment_details` SET  
`specialization`='$spec', `doctor`='$doct', `apt_date`='$date', `apt_time`='$tim  
e' WHERE apt_id='$aid'";
```

```
echo "<br><br>";  
echo $sql;
```

```
if ($conn->query($sql) === TRUE)  
{  
echo "<script>  
alert('APPOINTMENT DETAILS UPDATED SUCCESSFULLY');  
  
window.location.href='appointment_update.html';  
</script>";  
}  
else  
{  
echo "error";  
//die();  
echo "<script>  
alert('APPOINTMENT updation failed');  
window.location.href='employee_update_ui.php';  
</script>";  
}  
}  
?>
```

For Deletion:-

```
<?php  
  
$servername='localhost';  
$username='root';  
$password='';  
$dbname='hms';  
$conn =new mysqli($servername, $username, $password, $dbname);  
if($conn -> connect_errno){  
    echo "Failed to coneect to MYSQL" , $mysqli-> connect_error;  
    exit();  
}  
  
else {  
echo "<br>";  
$ad=$_POST['apt_id'];  
$deletesql="DELETE FROM `appointment_details` WHERE apt_id=$ad";  
echo "deleted details are";  
echo "<br>";  
echo $deletesql;  
if($conn->query($deletesql)==TRUE)  
{  
    echo "<script>  
    alert('appointment deleted successfully');  
    window.location.href='appointment_delete.html';  
    </script>";  
}  
}
```

```
else
{
    echo "<script>
    alert('login failed (invalid username and password)');
    window.location.href='login.html';
    </script>";
}
}
?>
```

3.4 Trigger and Stored Procedure

Following trigger and stored procedure have been implemented in the project.

Create a Trigger

```
CREATE TRIGGER trigger_name trigger_time trigger_event ON tbl_name FOR EACH
ROW [trigger_order] trigger_body /* where trigger_time: { BEFORE | AFTER }
trigger_event: { INSERT | UPDATE | DELETE } trigger_order: { FOLLOWS |
PRECEDES } */
```

Drop a Trigger

```
DROP TRIGGER [IF EXISTS] trigger_name;
```

Create a Stored Procedure

```
CREATE PROCEDURE sp_name(p1 datatype) BEGIN /*Stored procedure code*/ END;
```

call Stored procedure

```
CALL sp_name;
```

delete stored procedure

```
DROP PROCEDURE sp_name;
```

CHAPTER 4

Results and Discussion

In this chapter the results of the project are discussed. The snapshot of the project showing various functionalities like insert, delete, update and retrieval are showcased.



Fig. 4.1 Welcome page of project

Figure 4.1 shows the welcome page of the project. In this page the patient, doctor and receptionist can login. Also they can know about the hospital and the services and facility availables.

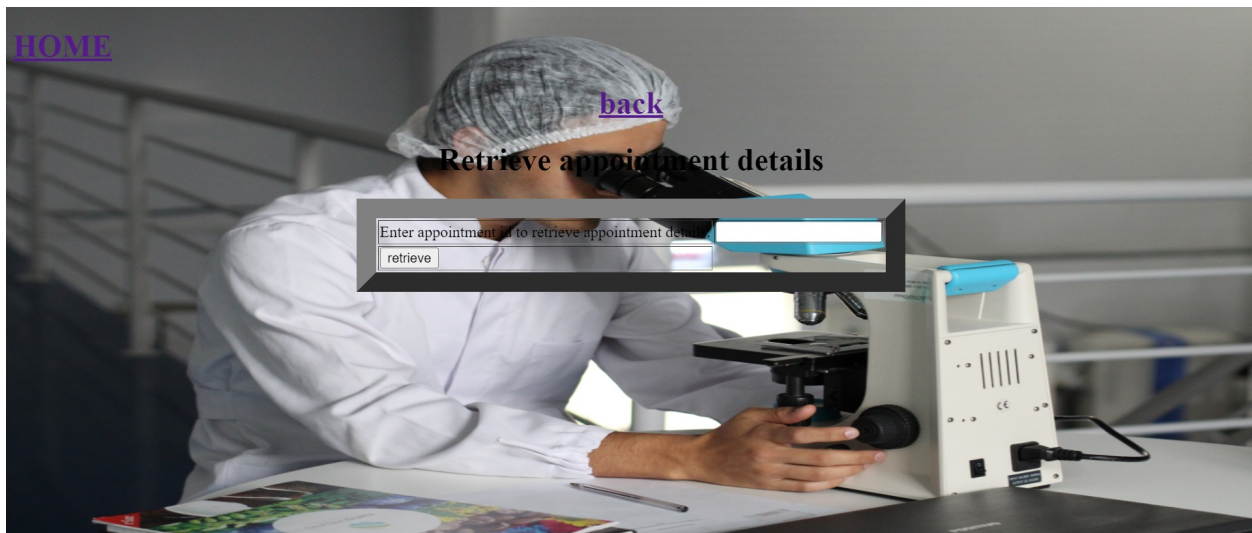


Fig. 4.2 Retrieval page of appointment

Figure 4.2 shows the retrieval page of the appointment. In this page the receptionist can retrieve the appointment details booked by the patient by giving the appointment id.

retrieved data are

| Specialization | Doctor | Appointment Date | Appointment Time | Appointment ID |
|----------------|--------------|------------------|------------------|----------------|
| neurologist | harsh mishra | 2023-10-10 | 4:00 pm | 10 |

Fig. 4.3 retrieved appointment data page

Figure 4.3 shows the retrieved appointment details. In this page receptionist can see the retrieved appointment details.



Fig. 4.4 Appointment booking page

Figure 4.4 and 4.5 shows the appointment booking page of the project. In this page the patient can book the appointment in which the different options will be available such as select speciality, select timing, select doctor, select date which is appropriate for the patient.

```
connected to Database
data inserted successfully
INSERT INTO `appointment_details`(`specialization`, `doctor`, `apt_date`, `apt_time`) VALUES ('neurologist','sumit','2023-02-20','2:00 pm')INSERT INTO `appointment_details`(`specialization`,
`doctor`, `apt_date`, `apt_time`) VALUES ('neurologist','sumit','2023-02-20','2:00 pm')
```

Fig. 4.5 Data insertion page

the entered appointment details will be viewed after successful booking of the appoinment. The different attributes given will be displayed with them.

CHAPTER 5

CONCLUSION

The project **Hospital Management System (HMS)** is for computerizing the working in a hospital. It is a great improvement over the manual system. The computerization of the system has speed up the process. In the current system, the front office managing is very slow. The hospital managing system was thoroughly checked and tested with dummy data and thus is found to be very reliable. The software takes care of all the requirements of an average hospital and is capable to provide easy and effective storage of information related to patients that come up to the hospital.

It generates test reports and also provides the facility for searching the details of the patient. It also provides billing facility on the basis of patient's status whether it is an indoor or outdoor patient. The system also provides the facility of backup as per the requirement.

REFERENCES

- [1] Database systems Models, Languages, Design and Application Programming, RamezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.