

## **DBMS Lab Assignment 4**

### **WEB APP DEVELOPMENT**

# **HOSPITAL MANAGEMENT SYSTEM**

Group Name :- **DATABASE SQUASHERS**

Group Members:- Pentamsetty Tharakeshwar 20CS10044  
Banoth Dinesh Karthik 20CS30009  
Saikat Moi 20CS10050  
Karamcheti Vamsy 20CS30025  
Dasari Giridhar Ram Chand 20CS10022

## **ABSTRACT**

### **Objective :**

Hospitals currently use manual systems for maintaining critical information. These systems require numerous paper forms, with data stores spread throughout the hospital management infrastructure. Information (on forms) is often incomplete, or does not follow management standards. Often forms are lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information in the hospitals may lead to inconsistencies in data in various data stores.

Operation of any hospital involves the acquisition, management and timely retrieval of large volumes of information. This information typically involves patient personal information and medical history, staff information, room and ward scheduling and various facilities waiting lists. All of this information must be managed efficiently so that an institution's resources may be effectively utilized. HMS will efficiently automate the management of the hospital and make it error free. It helps in ensuring data integrity and reducing inconsistencies.

### **Overview :**

The Hospital Management System (HMS) is designed to replace the manual, paper based systems in the hospitals. This new system controls the patient information, room availability, staff-patient appointments. HMS will automate the management of the hospital making it more efficient and error free. It aims at standardizing data, consolidating data, ensuring data integrity and reducing inconsistencies.

## **System Design:**

In this software we have developed some forms. The brief description about them is as follow:-

### **Operator:**

The operator module handles various functionalities for the Front Desk Operators such as registering patients, discharging patients etc. and for the Data Entry Operators such as entering patient data of tests and treatments within the hospital.

- Patient Login
- Patient Registration
- Patient Deregistration
- Patient Health Information
- Patient Discharge.

### **Admin:**

This module handles all the administrative details such as deleting or adding new users, managing appointments etc.

- Manage Users
- Manage Doctors
- Appointments
- Manage Patients
- Search

**Doctor:**

This module handles all the doctor details such as enquiring patient information, appointment scheduling etc.

- My Profile
- My Appointments
- Manage Patients
- Search Patient Info

**Patient:**

This module handles all the patient details such as booking appointments, booking rooms, managing medical history etc.

- My Profile
- My Appointments
- Book Appointment
- Book Room
- Medical History

## **Software Requirements:**

Web Technologies :

HTTP

Language :

HTML, JavaScript, CSS (Front End)

PHP (Back End)

Database :

MySQL

Server :

Apache Server

Operating System :

WINDOWS, Linux, MacOS

# **INDEX**

**S. N**

**CONTENTS**

- 1. INTRODUCTION**
- 2. ANALYSIS**
  - 2.1 SYSTEM ANALYSIS**
  - 2.2 SYSTEM SPECIFICATIONS**
- 3. DESIGN APPROACH**
  - 3.1 INTRODUCTION TO DESIGN**
  - 3.2 RELATIONAL SCHEMA**
- 4. PROJECT MODULES**
- 5. IMPLEMENTATION**
- 6. OUTPUT SCREENS**
- 7. CONCLUSION**
- 8. FUTURE ENHANCEMENTS**

## **INTRODUCTION**

Human Body is a very complex and sophisticated structure and comprises millions of functions. All these complicated functions have been understood by man himself, part-by-part through their research and experiments. As science and technology progressed, medicine became an integral part of the research. Gradually, medical science became an entirely new branch of science. As of today, the Health Sector comprises of Medical institutions i.e. Hospitals, HOSPITALS etc. research and development institutions and medical colleges. Thus the Health sector aims at providing the best medical facilities to the common man.

Hospital Management System provides the benefits of streamlined operations, enhanced administration, control, superior patient care, strict cost control and improved profitability. HMS is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. The project ‘Hospital Management System’ is based on the database and networking techniques. As there are many areas where we keep the records in the database for which we are using MYSQL software which is one of the best and the easiest software to keep our information. This project uses HTML, PHP as the front-end software which has connectivity with MYSQL.

The entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more user friendly and expandable. It covers all the required modules right from Patient Registration, Appointment Scheduling, Doctor, Admin, discharge details etc.

# **ANALYSIS**

## **System Analysis:**

### **1. Existing System:-**

Hospitals currently use manual systems for maintaining critical information. These systems require numerous paper forms, with data stores spread throughout the hospital management infrastructure. Information (on forms) is often incomplete, or does not follow management standards. Often forms are lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information in the hospitals may lead to inconsistencies in data in various data stores.

### **2. Proposed System:-**

The Hospital Management System (HMS) is designed for hospitals to replace their existing manual, paper based system. The new system is to control the following information; patient information, room availability, staff and operating room schedules, and patient invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks.

### **3. Objective of the System:-**

A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves; patient personal information and medical history, staff information, room and ward scheduling, staff scheduling, operating theater scheduling and various facilities waiting lists. All of this information must be managed in an efficient and cost wise fashion so that an institution's resources may be effectively utilized. HMS will automate the management of the hospital making it more efficient and error free. It aims at standardizing data, consolidating data, ensuring data integrity and reducing inconsistencies.

## **System Specifications:**

### Hardware Requirements:-

- Pentium-IV(Processor).
- 256 MB Ram
- 512 KB Cache Memory
- Hard disk 10 GB
- Microsoft Compatible 101 or more Keyboard

### Software Requirements:-

- |                         |                              |
|-------------------------|------------------------------|
| • Operating System      | : Windows                    |
| • Programming Languages | : HTML, PHP, CSS, Javascript |
| • Web Technologies      | : HTTP                       |
| • Web Server            | : Apache Server              |

## **DESIGN**

### **Introduction to Design:**

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

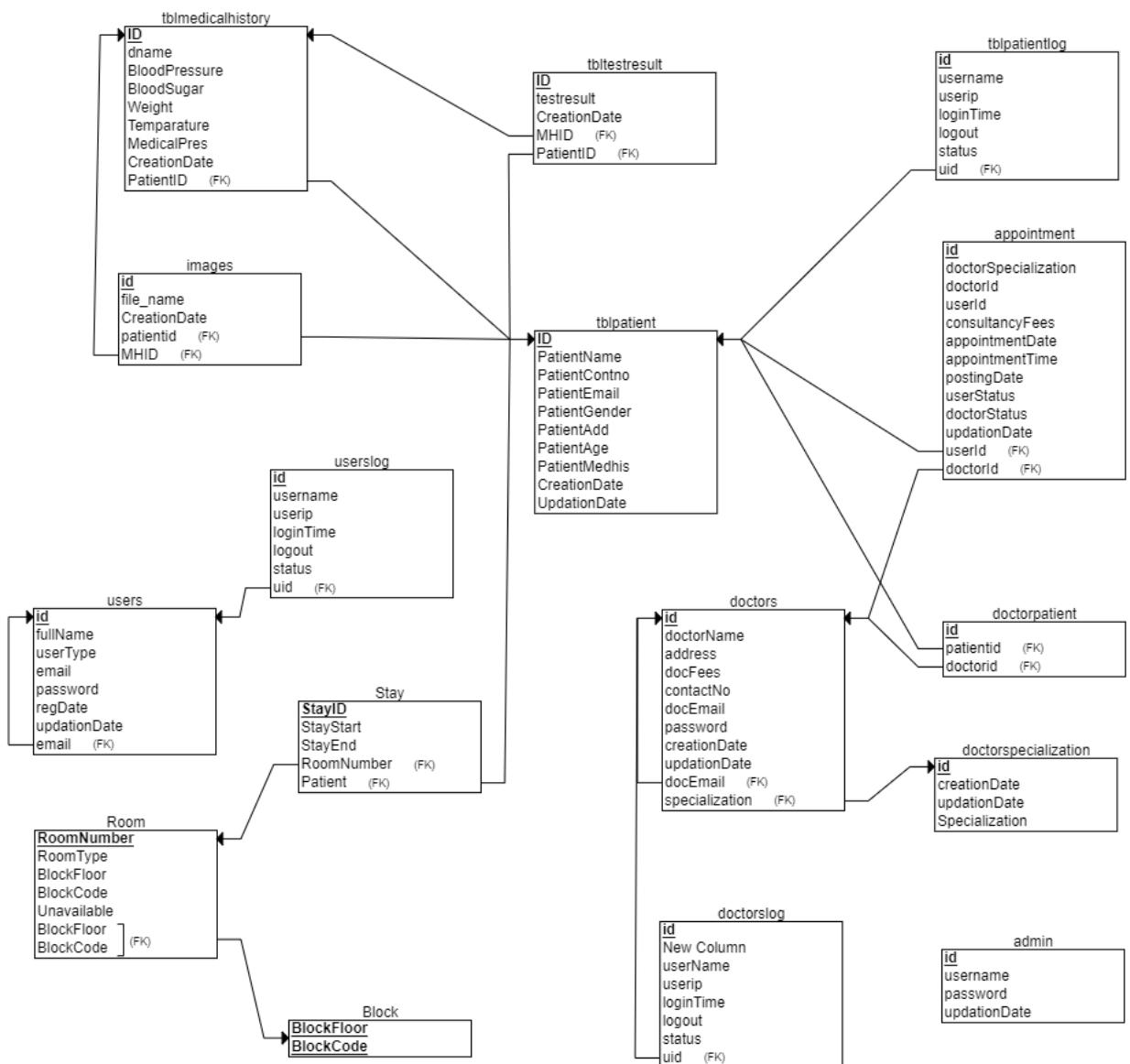
The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system. Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software.

## **Relational Schema:**

A relational schema is a collection of metadata that describes the structure of a relational database. It defines the tables or relation that make up the databases, as well as the columns or attributes within those tables, and the relationships between the tables. In other words, a relation schema provides a blueprint for the organization of the data within a database, specifying the data types, constraints and relationships between tables. It consists of a set of table definitions where each table is defined by a name and a list of attributes or columns, along with their data types and any constraints or rules that govern their values.

For example, a simple relational schema for a customer database might include tables for customers, orders and products with column names such as customer\_id, order\_id, product\_id, order\_id, order\_date and product\_price among others. The schema would also define relationships between these tables, such as the fact that each order is associated with a single customer, and each order contains one or more products. Overall, the relational schema serves as a map for how data is stored and accessed within a relational database, providing a clear and standardized way to organize and manage complex data structures.

## RELATIONAL SCHEMA FOR HMS:



## **PROJECT MODULES**

- Operators
- Admin
- Doctor
- Patient

### **Module-1:** Operators

**Description:** The Operators module handles various enquiries connected with the front-desk and data-entry operators. This module handles registration and deregistration of patients; updating patient data of tests and treatments etc.

#### **Sub modules:**

- Patient Login
- Patient Registration
- Patient Deregistration
- Patient Health Information
- Patient Discharge.

## **Module-2: Admin**

**Description:** This module handles all the administrative entry details for the hospital requirement such as deletion or addition of new users, appointment scheduling etc.

### **Sub modules:**

- Manage Users
- Manage Doctors
- Appointments
- Manage Patients
- New Queries

## **Module-3: Doctor**

**Description:** This module deals with all functionalities of doctors. This module helps in enquiring patient information, managing appointments etc.

### **Sub modules:**

- My Profile
- My Appointments
- Manage Patients
- Search Patient Info

## **Module-4: Patient**

**Description:** This module enables the maintenance of patient details. This module is integrated with the appointment booking, room booking, medical history updation etc.

### **Sub modules:**

- My Profile
- My Appointments
- Book Appointment
- Book Room
- Medical History

# **IMPLEMENTATION**

## **Technologies Used:**

### **1. SQL**

SQL (Structured Query Language) is a programming language designed to manage and manipulate relational databases. It is used to create, modify and query databases and tables within them. SQL provides a standardized syntax and commands for working with relational databases. It can be used to perform a wide range of operations, including:

1. Creating and modifying databases, tables and indexes.
2. Inserting, updating and deleting data in tables.
3. Querying data to retrieve specific information from tables
4. Joining multiple tables to combine data from different sources
5. Filtering and sorting data based on specific criteria
6. Aggregating and summarizing data using functions like COUNT, AVG, MAX, MIN and SUM.

SQL is used by a wide variety of organizations and applications, from small businesses to large enterprises. It is a powerful tool for managing and analyzing data and is often used in conjunction with other programming languages and tools to build complex applications and systems.

### **2. HTML-CSS**

HTML or HyperText Markup Language is the standard markup language for all web pages worldwide. It's not a typical programming language like Python or Java since it doesn't contain any programming logic. HTML can't perform data manipulations or calculations, for example. Instead, HTML allows you to create and format the fundamental structure and content of a web page. HTML helps in creating page layouts (header, body, footer, sidebar), paragraphs and headings, embedded media etc. Thus, HTML only allows you to determine the structure of a web page and place individual content elements within it.

CSS or Cascading Style Sheets is a style sheet language that allows you to adjust the design and feel of your HTML content. Thus, you can turn your pure-HTML pages into stunning, modern websites with CSS. CSS allows you to target individual HTML elements and apply different styling rules to them. You can use CSS to adjust backgrounds, fonts and text styling, colors etc.

### 3. PHP

PHP(Hypertext Preprocessor) is a popular server-side scripting language that is used to create dynamic web pages and web applications. It was created in 1994 by Rasmus Lerdorf, and today it is one of the most widely-used programming languages on the web, powering millions of websites and applications. One of the main strengths of PHP is its ease of use and flexibility. It can be embedded directly into HTML code, which makes it easy to create dynamic web pages that can change based on user input or other factors. PHP is also platform-independent, which means it can run on a wide variety of web servers and operating systems.

PHP can be used to perform a wide range of tasks, from simple form processing and database queries to more complex tasks like image manipulating and PDF generation. It also has extensive support for databases, including MySQL, PostgreSQL, Oracle and others. In addition to its core functionality, PHP has a large and active community of developers who have created a wide variety of libraries and frameworks that make it even easier to create powerful web applications. Some of the most popular PHP frameworks include Laravel, Symfony, CodeIgniter and CakePHP.

Overall, PHP is a powerful and versatile programming language that is well-suited for web development. Its ease of use, flexibility and wide range of functionality make it a popular choice for developers around the world.

## 4. JAVASCRIPT

JavaScript is a high-level, interpreted programming language commonly used for client-side web development. It was originally created by Netscape in 1995 and is now standardized by ECMAScript specification. JavaScript is often used to create interactive and dynamic web content, such as dropdown menus, pop-ups and form validations. It can be embedded directly into HTML and CSS documents or included as a separate file. It is also used in server-side programming, mobile app development, game development and desktop application development.

JavaScript is an object-oriented language that supports both functional and imperative programming paradigms. It has a wide range of built-in objects and functions, as well as a large and active community that creates and maintains libraries and frameworks. Some of the key features of JavaScript include its ability to manipulate the Document Object Module(DOM) of a web page, which allows developers to dynamically change the content and layout of a web page without reloading it. It also supports asynchronous programming through the use of callbacks, promises, and async-await syntax, which allows for more efficient and responsive code. Overall, JavaScript is a versatile and widely used programming language that is essential for modern web development.

### Database Tables:

#### Admin:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	int(11)	Primary
Username	Not Null	Varchar(255)	
Password	Not Null	Varchar(255)	

UpdationDate	Not Null	Varchar(255)	
--------------	----------	--------------	--

### Appointment:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Int(11)	Primary
DoctorSpecialization	Null	Varchar(255)	
DoctorID	Null	Int(11)	
UserID	Null	Int(11)	
ConsultancyFees	Null	Int(11)	
AppointmentDate	Null	Varchar(255)	
AppointmentTime	Null	Varchar(255)	
PostingDate	Null	Timestamp	
UserStatus	Null	Int(11)	
DoctorStatus	Null	Int(11)	
UpdationDate	Null	Timestamp	

### Doctors:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Int(11)	Primary
Specialization	Not Null	Varchar(2555)	
DoctorName	Null	Varchar(255)	
Address	Null	Longtext	
DocFees	Null	Varchar(255)	

Contactno	Null	Bigint(11)	
DocEmail	Null	Varchar(255)	
Password	Null	Varchar(255)	
CreationDate	Null	Timestamp	
UpdationDate	Null	Timestamp	

### Images:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Int(11)	Primary
MHID	Not Null	Int(10)	
PatientID	Not Null	Int(11)	
File_name	Not Null	Varchar(255)	
CreationDate	Null	Timestamp	

### Doctorslog:-

Name	DEFAULT	TYPE	KEY
ID	Not null	Int(11)	
UID	Null	Int(11)	
Username	Null	Varchar(255)	
Userip	Null	Binary(16)	
LoginTime	Null	Timestamp	
Logout	Null	Varchar(255)	

Status	Null	Int(11)	
--------	------	---------	--

Doctorpatient:-

Name	DEFAULT	TYPE	KEY
ID	Not null	Int(11)	Primary
Patientname	Null	Int(11)	
Age	Null	Int(11)	

DoctorSpecialization:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Int(11)	Primary
Specialization	Null	Varchar(255)	
CreationDate	Null	Timestamp	
UpdationDate	Null	Timestamp	

Tblcontactus:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Varchar(20)	Primary
Fullname	Null	Varchar(255)	
Email	Null	Varchar(255)	
Contactno	Null	Bigint(12)	
Message	Null	Mediumtext	
PostingDate	Null	Timestamp	

AdminRemark	Null	Mediumtext	
LastUpdationDate	Null	Timestamp	
IsRead	Null	Int(11)	

Tblmedicalhistory:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Int(10)	Primary
Dname	Null	Varchar(200)	
PatientID	Null	Int(10)	
BloodPressure	Null	Varchar(200)	
BloodSugar	Null	Varchar(200)	
Weight	Null	Varchar(100)	
Temperature	Null	Varchar(200)	
MedicalPres	Null	Mediumtext	
CreationDate	Not Null	Timestamp	

Tbltestresult:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Int(10)	Primary

MHID	Not Null	Int(10)	
PatientID	Null	Int(10)	
Testresult	Null	Mediumtext	
CreationDate	Not Null	Timestamp	

Tblpage:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Int(10)	Primary
PageType	Null	Varchar(200)	
PageTitle	Null	Varchar(200)	
PageDescription	Null	Mediumtext	
Email	Null	Varchar(120)	
MobileNumber	Null	Bigint(10)	
UpdationDate	Null	Timestamp	
OpeningTime	Null	Varchar(255)	

Tblpatient:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Int(10)	Primary
PatientName	Null	Varchar(200)	

PatientContno	Null	Bigint(10)	
PatientEmail	Null	Varchar(200)	
PatientGender	Null	Varchar(50)	
PatientAdd	Null	Mediumtext	
PatientAge	Null	Int(10)	
PatientMedhis	Null	Mediumtext	
CreationDate	Null	Timestamp	
UpdationDate	Null	Timestamp	

Tblpatientlog:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Int(11)	Primary
USID	Null	Int(11)	
Username	Null	Varchar(255)	
Userip	Null	Binary(16)	
LoginTime	Null	Timestamp	
Logout	Null	Varchar(255)	
Status	Null	Int(11)	

Userlog:-

Name	DEFAULT	TYPE	KEY

ID	Not Null	Int(11)	Primary
UID	Null	Int(11)	
Username	Null	Varchar(255)	
Userip	Null	Binary(16)	
LoginTime	Null	Timestamp	
Logout	Null	Varchar(255)	
Status	Null	Int(11)	

### Users:-

Name	DEFAULT	TYPE	KEY
ID	Not Null	Int(11)	Primary
Fullname	Null	Varchar(255)	
Usertype	Null	Varchar(255)	
Email	Null	Varchar(255)	
Password	Null	Varchar(255)	
Regdate	Null	Timestamp	
UpdationDate	Null	Timestamp	

### Block:-

Name	DEFAULT	TYPE	KEY
BlockFloor	Not Null	Integer	Primary
BlockCode	Not Null	Integer	Primary

**Room:-**

Name	DEFAULT	TYPE	KEY
RoomNumber	Not Null		Primary
RoomType	Null	Varchar(30)	
BlockFloor	Null		Foreign
BlockCode	Null		Foreign
Unavailable	Null		

**Stay:-**

Name	DEFAULT	TYPE	KEY
StayID	Not Null	Integer	Primary
Patient	Not Null	Integer	
Room	Not Null	Integer	Foreign
StayStart	Not Null	Datetime	
StayEnd	Not Null	Datetime	

**Rtype:-**

Name	DEFAULT	TYPE	KEY
Types	Not Null	Varchar(30)	Primary

## **Queries Used:**

### **Index.php:**

```
insert into tblcontactus(fullname,email,contactno,message)
      value(''$name'', ''$email'', ''$mobileno'', ''$dscrption'')
```

```
select * from tblpage where PageType='aboutus'
```

### **OPERATOR MODULE:**

#### **Change-password.php:**

```
SELECT password FROM users where password="'.md5($_POST['cpass']).'" &&
id=''. $_SESSION['id'].'"'
```

```
update users set password="'.md5($_POST['npass']).'", updationDate=
CURRENT_TIMESTAMP where id=''. $_SESSION['id'].'"'
```

#### **Check\_availability.php:**

```
SELECT email FROM users WHERE email='$_email'
```

#### **Discharge.php:**

```
SELECT stay.StayID, tblpatient.PatientName, Room.RoomNumber, stay.StayStart,
stay.StayEnd
FROM stay
INNER JOIN tblpatient ON stay.Patient = tblpatient.ID
INNER JOIN Room ON stay.Room = Room.RoomNumber
where stay.StayEnd is NULL
```

```
UPDATE Room  
INNER JOIN stay ON Room.RoomNumber = stay.Room  
SET Room.Unavailable = 0  
WHERE stay.StayID = $stay_id
```

```
UPDATE stay SET StayEnd = NOW() WHERE StayID = $stay_id
```

```
select * from Stay where StayEnd is NULL
```

### Edit-profile.php:

```
Update users set fullName='".$fname',usertype='".$usertype' where  
id='".$SESSION['id']."'  
select * from users where id='".$SESSION['id']."'
```

### Forgot-password.php:

```
select id from users where fullName='".$name' and email='".$email'
```

### Health\_check\_info.php:

```
SELECT * FROM users WHERE email='".$uname' and password='".$ppwd' and usertype =  
'$usertype' and usertype = 'dataentry'
```

```
insert into userlog(uid,username,userip,status)  
values('$pid','$uname','$uiip','$status')
```

```
insert into userlog(username,userip,status) values ('$uname','$uiip','$status')
```

### Logout.php:

```
UPDATE userlog SET logout = '$ldate' WHERE uid = '". $_SESSION['id']."' ORDER BY id DESC LIMIT 1
```

### Pat\_dereg.php:

```
DELETE FROM tblpatient WHERE ID = $stay_id
```

```
SELECT * FROM tblpatient
```

### Patient\_check\_availability.php:

```
SELECT PatientEmail FROM tblpatient WHERE PatientEmail='$email'
```

### Patient\_info.php:

```
SELECT * FROM users WHERE email='$pname' and password='$ppwd' and usertype = '$usertype' and usertype = 'frontdesk'  
  
insert into userlog(uid,username,userip,status)  
values('$pid','$pname','$uiip','$status')
```

```
insert into userlog(username,userip,status) values ('$pname', '$uiip', '$status')
```

### Patient\_registration.php:

```
insert into  
tblpatient(PatientName,PatientContno,PatientEmail,PatientGender,PatientAdd,Pat  
ientAge,PatientMedhis)  
values('$patname','$patcontact','$pemail','$gender','$pataddress','$patage',  
'$medhis')
```

### Registration.php:

```
insert into users(fullname,email,password,usertype)  
values('$fname','$email','$password','$usertype')
```

### **Reset-password.php:**

```
update users set password='$newpassword' where fullName='$name' and  
email='$email'
```

### **User-login.php:**

```
SELECT * FROM users WHERE email='$pname' and password='$ppwd' and usertype =  
'$usertype'
```

```
insert into userlog(uid,username,userip,status)  
values('$pid','$pname','$uiip','$status')
```

```
insert into userlog(username,userip,status) values('$pname','$uiip','$status')
```

### **ADMIN MODULE:**

#### **Add-doctor.php:**

```
insert into  
doctors(specilization,doctorName,address,docFees,contactno,docEmail,password)  
values('$docspecialization','$docname','$docaddress','$docfees','$doccontactno'  
', '$docemail', '$password')
```

```
select * from doctorspecialization
```

#### **Add-patient.php:**

```
insert into  
tblpatient(PatientName,PatientContno,PatientEmail,PatientGender,PatientAdd,Pat  
ientAge,PatientMedhis)  
values('$patname', '$patcontact', '$patemail', '$gender', '$pataddress', '$patage',  
'$medhis')
```

#### **Add-user.php:**

```
insert into users(fullName,userType,email,password)
values('$fullName','$userType','$email','$password')
```

### Appointment-history.php:

```
select doctors.doctorName as docname,users.fullName as pname,appointment.*
from appointment join doctors on doctors.id=appointment.doctorId join users on
users.id=appointment.userId
```

### between-dates-detailsreports1.php:

```
select * from tblpatient where date(CreationDate) between '$fdate' and
'$tdate'
```

### Change-password.php:

```
SELECT password FROM admin where password='$cpass' && username='$uname'
update admin set password='$npass', updationDate='$currentTime' where
username='$uname'
```

### Check\_availability.php:

```
SELECT docEmail FROM doctors WHERE docEmail='$email'
```

### Dashboard.php:

```
SELECT * FROM doctors
```

```
SELECT * FROM appointment
```

```
SELECT * FROM tblpatient
```

### Doctor-logs.php:

```
select * from doctorslog
```

### Doctor-specialization.php:

```
insert into doctorSpecilization(specilization) values('$doctorspecialization')
```

```
delete from doctorSpecilization where id = '$sid'
```

```
select * from doctorSpecilization
```

### Edit-doctor-specialization.php:

```
update doctorSpecilization set specilization='$docspecialization' where id='$id'
```

```
select * from doctorSpecilization where id='$id'
```

### Edit-doctor.php:

```
Update doctors set specialization='$docspecialization',doctorName='$docname',address='$docaddress',docFees='$docfees',contactno='$doccontactno',docEmail='$docemail' where id='$did'
```

```
select * from doctors where id='$did'
```

```
select * from doctorspecialization
```

### Index.php:

```
SELECT * FROM admin WHERE username='$uname' and password='$upassword'
```

### Manage-doctors.php:

```
delete from doctors where id ='$docid'
```

```
select * from doctors
```

### Manage-patients.php:

```
select * from tblpatient
```

### Manage-users.php:

```
delete from users where id ='$uid'
```

```
select * from users
```

### Patient-search.php:

```
select * from tblpatient where PatientName like '%$sdata%' || PatientContno  
like '%$sdata%'
```

### Query-details1.php:

```
update tblcontactus set AdminRemark='$adminremark',IsRead='$isread' where  
id='$qid'
```

```
select * from tblcontactus where id='$qid'
```

### Read-query1.php:

```
select * from tblcontactus where IsRead is not null
```

### Users-logs.php:

```
select * from userlog
```

### View-patient.php:

```
insert  
tblmedicalhistory(PatientID,BloodPressure,BloodSugar,Weight,Temperature,Medica  
lPres) value('$vid','$bp','$bs','$weight','$temp','$pres')
```

```
select * from tblpatient where ID='$vid'
```

```
select * from tblmedicalhistory where PatientID='$vid'
```

## DOCTOR MODULE:

### Add-patient.php:

```
insert into
tblpatient(Docid,PatientName,PatientContno,PatientEmail,PatientGender,PatientA
dd,PatientAge,PatientMedhis)
values('$docid','$patname','$patcontact','$patemail','$gender','$pataddress','
$patage','$medhis')
```

### Appointment-history.php:

```
update appointment set doctorStatus='0' where id ='".$_GET['id']."'
```

```
select tblpatient.PatientName as fname,appointment.* from appointment join
tblpatient on tblpatient.ID=appointment.userId where appointment.doctorId='
```

### Change-password.php:

```
SELECT password FROM doctors where password='$cpass' && id='$did'
```

```
update doctors set password='$npassword', updationDate='$currentTime' where
id='$did'
```

### Check-availability.php:

```
SELECT PatientEmail FROM tblpatient WHERE PatientEmail='$email'
```

### Edit-patient.php:

```
update tblpatient set  
PatientName='$patname',PatientContno='$patcontact',PatientEmail='$patemail',Pa  
tientGender='$gender',PatientAdd='$pataddress',PatientAge='$patage',PatientMed  
his='$medhis' where ID='$eid'
```

```
select * from tblpatient where ID='$eid'
```

### Edit-profile.php:

```
Update doctors set  
doctorName='$docname',address='$docaddress',docFees='$docfees',contactno='$doc  
contactno' where id='". $_SESSION['id'] . "'
```

```
select * from doctors where docEmail='$did'
```

### Forgot-password.php:

```
select id from doctors where contactno='$contactno' and docEmail='$email'
```

### Index.php:

```
SELECT * FROM doctors WHERE docEmail='$uname' and password='$dpassword'
```

```
insert into doctorslog(uid,username,userip,status)  
values('$uid','$uname','$uiip','$status')
```

```
insert into doctorslog(username,userip,status)  
values('$uname','$uiip','$status')
```

### Logout.php:

```
UPDATE doctorslog SET logout = '$ldate' WHERE uid = '$did' ORDER BY id DESC  
LIMIT 1
```

### Manage-patient.php:

```
select * from doctorpatient,tblpatient,doctors where doctorid='$docid' and
```

```
tblpatient.ID = doctorpatient.patientid and doctors.id=doctorpatient.doctorid
```

### Reset-password.php:

```
update doctors set password='$newpassword' where contactno='$cno' and docEmail='$email'
```

### Search.php:

```
select * from tblpatient where PatientName like '%$sdata%' || PatientContno like '%$sdata%'
```

### View-patient.php:

```
insert  
tblmedicalhistory(PatientID,dname,BloodPressure,BloodSugar,Weight,Temperature,  
MedicalPres) value('$vid','$dname','$bp','$bs','$weight','$temp','$pres')
```

```
select * from tblmedicalhistory as tmh where tmh.PatientID='$vid'
```

```
select * from tblofficeresult where PatientID='$vid'
```

### PATIENT MODULE:

#### Appointment-history.php:

```
update appointment set userStatus='0' where id = '".$_GET['id']."'
```

```
select doctors.doctorName as docname,appointment.* from appointment join  
doctors on doctors.id=appointment.doctorId where  
appointment.userId=' ".$_SESSION['id']."''
```

## Book-Appointment.php:

```
insert into  
appointment(doctorSpecialization,doctorId,userId,consultancyFees,appointmentDa  
te,appointmentTime,userStatus,doctorStatus)  
values('$specilization','$doctorid','$userid','$fees','$appdate','$time','$use  
rstatus','$docstatus')
```

```
select * from doctorspecialization
```

## Book-Room.php:

```
SELECT RoomNumber FROM Room WHERE RoomType='$specilization' AND Unavailable=0  
LIMIT 1
```

```
UPDATE Room SET Unavailable=1 WHERE RoomNumber='$roomnumber'
```

```
select count(*) from stay
```

```
INSERT INTO stay (StayID,Patient, Room, StayStart) VALUES ('$id','$userid',  
'$roomnumber', NOW())
```

```
select * from rtype
```

## Change-emailid.php:

```
Update users set email='$email' where id='". $_SESSION['id']."' . ''
```

## Edit-profile.php:

```
Update tblpatient set  
PatientName='$PatientName', PatientContno='$PatientContno', PatientEmail='$Pati  
entEmail', PatientGender='$PatientGender', PatientAdd='$PatientAdd', PatientAge='$  
PatientAge' where id='". $_SESSION['id']."' . ''
```

```
select * from tblpatient where id='". $_SESSION['id']."' . ''
```

### Edit-profile1.php:

```
Update tblpatient set  
PatientName='$PatientName',PatientContno='$PatientContno',PatientEmail='$PatientEmail',PatientGender='$PatientGender',PatientAdd='$PatientAdd',PatientAge='$PatientAge' where id='". $_SESSION['id']."''
```

```
select * from tblpatient where id='". $_SESSION['id']."''
```

### Forgot-password.php:

```
select id from users where fullName='$name' and email='$email'
```

### Get\_doctor.php:

```
select doctorName,id from doctors where  
specilization='". $_POST['specilizationid']."''
```

```
select docFees from doctors where id='". $_POST['doctor']."''
```

### Logout.php:

```
UPDATE tblpatientlog SET logout = '$ldate' WHERE uid = '". $_SESSION['id']."'"  
ORDER BY id DESC LIMIT 1
```

### Logout1.php:

```
UPDATE tblpatientlog SET logout = '$ldate' WHERE uid = '". $_SESSION['id']."'"  
ORDER BY id DESC LIMIT 1
```

### Manage-medhistory.php:

```
select tblpatient.* from tblpatient where tblpatient.ID='$uid'
```

### Manage-medhistory1.php:

```
select tblpatient.* from tblpatient where tblpatient.ID='$uid';
```

### Registration.php:

```
insert into users(fullname,address,city,gender,email,password)  
values('$fname','$address','$city','$gender','$email','$password')
```

### Reset-password.php:

```
update users set password='$newpassword' where fullName='$name' and  
email='$email'
```

### User-login.php:

```
SELECT * FROM tblpatient WHERE PatientEmail='$pname'
```

```
insert into tblpatientlog(uid,username,userip,status)  
values('$pid','$pname','$uiip','$status')
```

```
insert into tblpatientlog(username,userip,status)  
values('$pname','$uiip','$status')
```

### User-Login1.php:

```
SELECT * FROM tblpatient WHERE PatientEmail='$pname'
```

```
insert into tblpatientlog(uid,username,userip,status)  
values('$pid','$pname','$uiip','$status')
```

```
insert into tblpatientlog(username,userip,status)  
values('$pname','$uiip','$status')
```

### View-medhistory.php:

```
insert  
tblmedicalhistory(PatientID,MID,dname,BloodPressure,BloodSugar,Weight,Temperat
```

```
ure,MedicalPres) value('$vid','$mid','$dname','$bp','$bs','$weight','$temp','$pres')
```

```
select * from tblpatient where ID='$vid'
```

```
select * from tblmedicalhistory where PatientID='$vid'
```

### View-medhistory1.php:

```
insert tbltestresult(MHID,PatientID,testresult) value('$mid','$vid','$pres')
```

```
INSERT into images (MHID, patientid, file_name, CreationDate) VALUES  
('$mid','$vid','".$fileName."', NOW())
```

```
select * from tblpatient where ID='$vid'
```

```
select * from tblmedicalhistory where PatientID='$vid'
```

```
select * from tbltestresult where PatientID='$vid'
```

```
select * from images where patientid='$vid'
```

### Triggers Used:

```
DROP TRIGGER IF EXISTS doctorPatient;  
DELIMITER $$  
CREATE TRIGGER doctorPatient
```

```
AFTER INSERT ON appointment  
FOR EACH ROW  
  
BEGIN  
  
    insert into doctorpatient(doctorid, patientid) values (new.doctorId,  
new.userId);  
  
END $$  
  
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS discharge_on_deregister;  
  
DELIMITER $$  
  
CREATE TRIGGER discharge_on_deregister  
        BEFORE DELETE ON tblpatient  
  
FOR EACH ROW  
  
BEGIN  
  
    UPDATE Room  
  
    INNER JOIN stay ON Room.RoomNumber = stay.Room  
  
    SET Room.Unavailable = 0  
  
    WHERE stay.Patient = OLD.ID;  
  
    UPDATE stay SET StayEnd = NOW() WHERE Patient = OLD.ID;  
  
END $$  
  
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS on_patient_deregister;
```

```
DELIMITER $$

CREATE TRIGGER on_patient_deregister
    BEFORE DELETE ON tblpatient
FOR EACH ROW
BEGIN
    DELETE FROM appointment WHERE userId = OLD.ID;
    DELETE FROM images WHERE patientid = OLD.ID;
    DELETE FROM tbltestresult WHERE PatientID = OLD.ID;
    DELETE FROM tblmedicalhistory WHERE PatientID = OLD.ID;
    DELETE FROM doctorpatient WHERE patientid = OLD.ID;
END $$

DELIMITER ;
```

```
DROP TRIGGER IF EXISTS on_doctor_deregister;

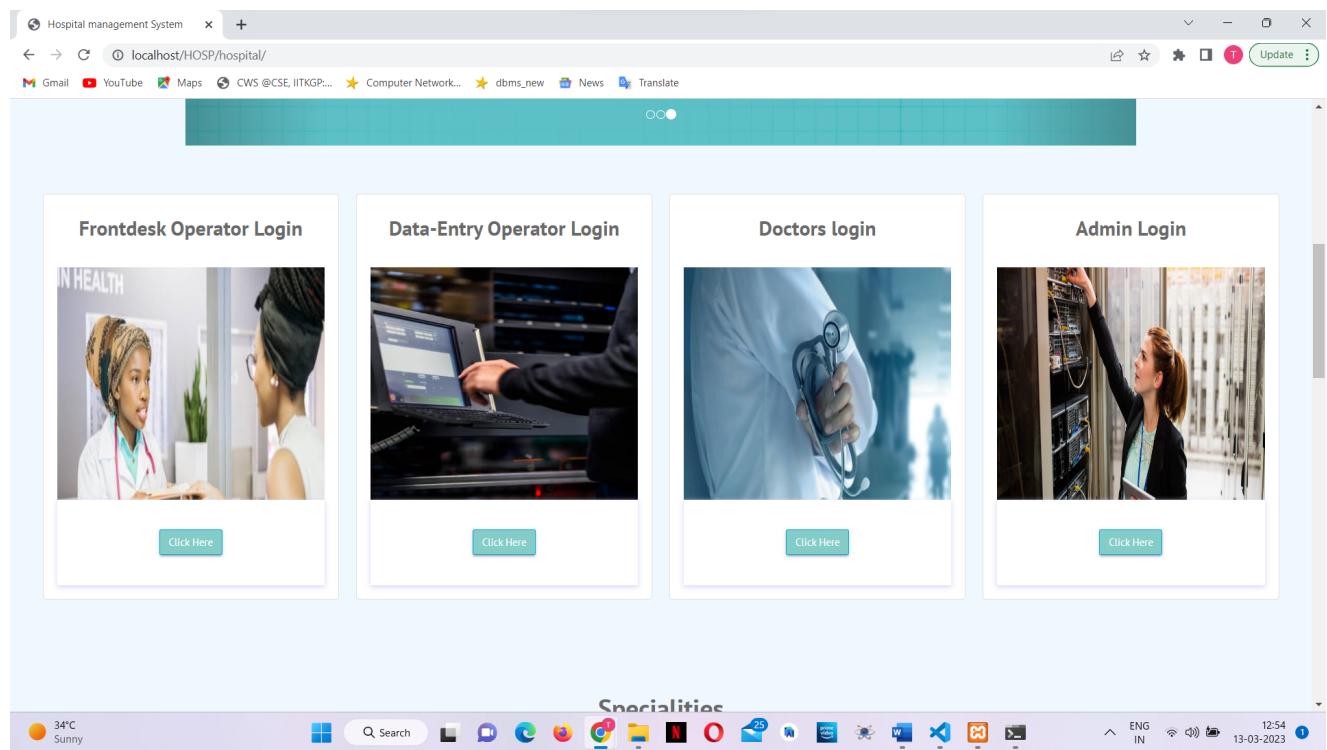
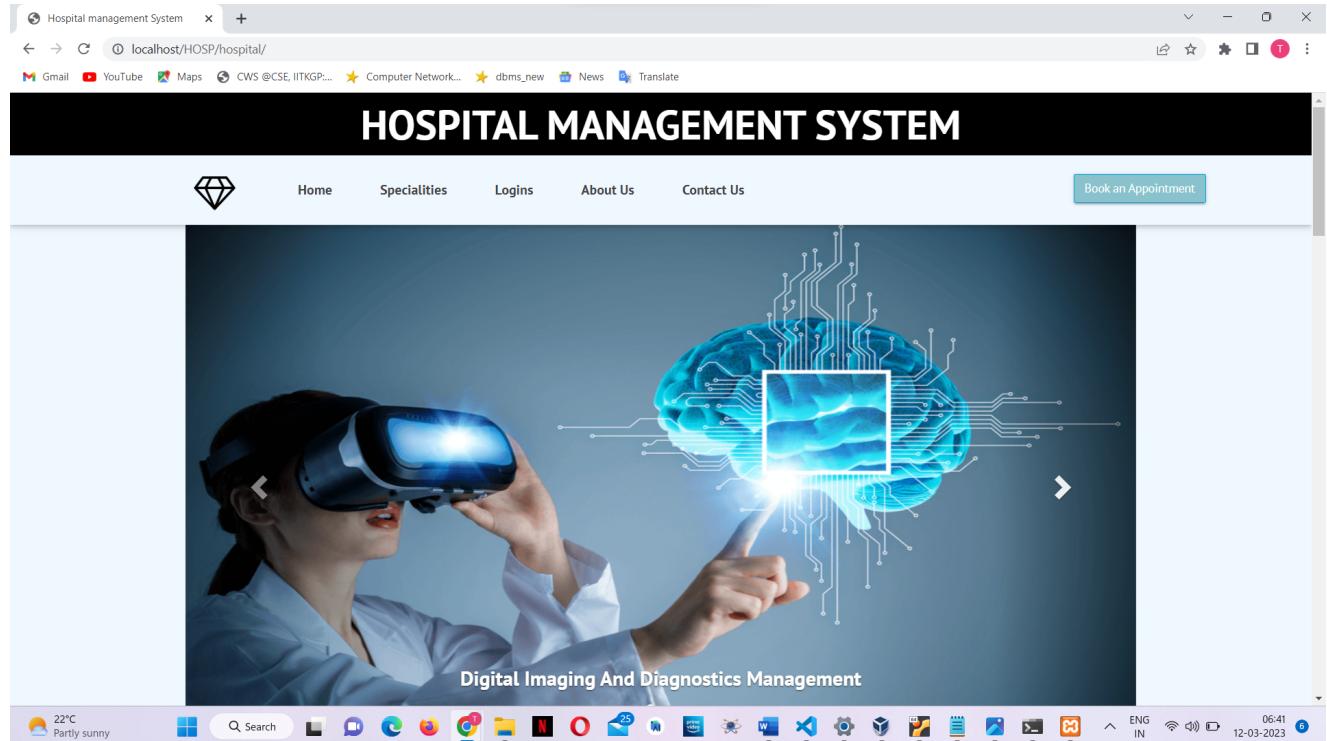
DELIMITER $$

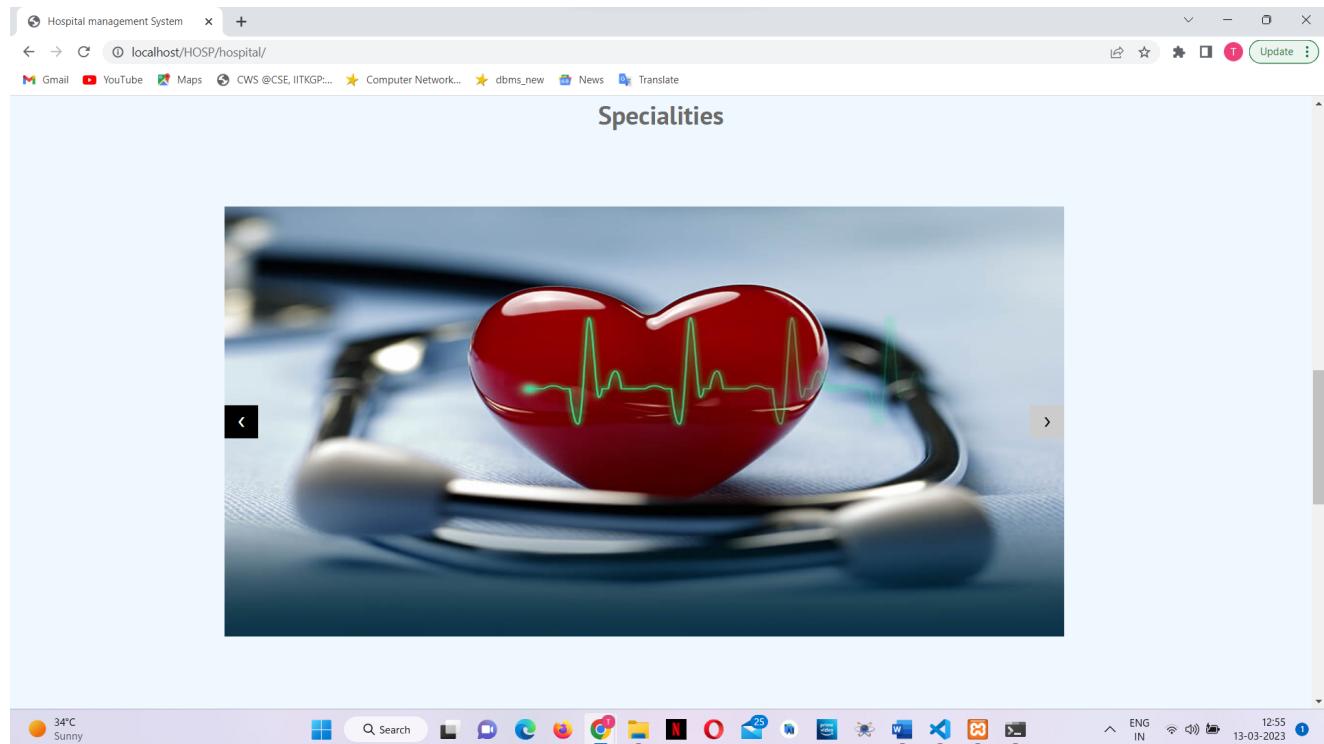
CREATE TRIGGER on_doctor_deregister
    BEFORE DELETE ON doctors
FOR EACH ROW
BEGIN
    DELETE FROM appointment WHERE doctorId = OLD.id;
    DELETE FROM doctorpatient WHERE doctorId = OLD.id;
END $$

DELIMITER ;
```

# OUTPUT SCREENS

Home Page:





A screenshot of a web browser window titled "Hospital management System". The URL in the address bar is "localhost/HOSP/hospital/". The page content is titled "About Our Hospital" and features a large image of a smiling patient in a dental chair. To the right of the image, there is a detailed description of the Hospital Management System (HMS) and its benefits. Below this, there are sections for "Useful Links" and "Contact Us".

**About Our Hospital**

The Hospital Management System (HMS) is designed for Any Hospital to replace their existing manual, paper based system. The new system is to control the following Information; patient information, room availability, staff and operating room schedules, and patient invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks.

A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves; patient personal information and medical history, staff information, room and ward scheduling, staff scheduling, operating theater scheduling and various facilities waiting lists. All of this information must be managed in an efficient and cost wise fashion so that an institutions resources may be effectively utilized HMS will automate the management of the hospital making it more efficient and error free. It aims at standardizing data, consolidating data ensuring data integrity and reducing inconsistencies.

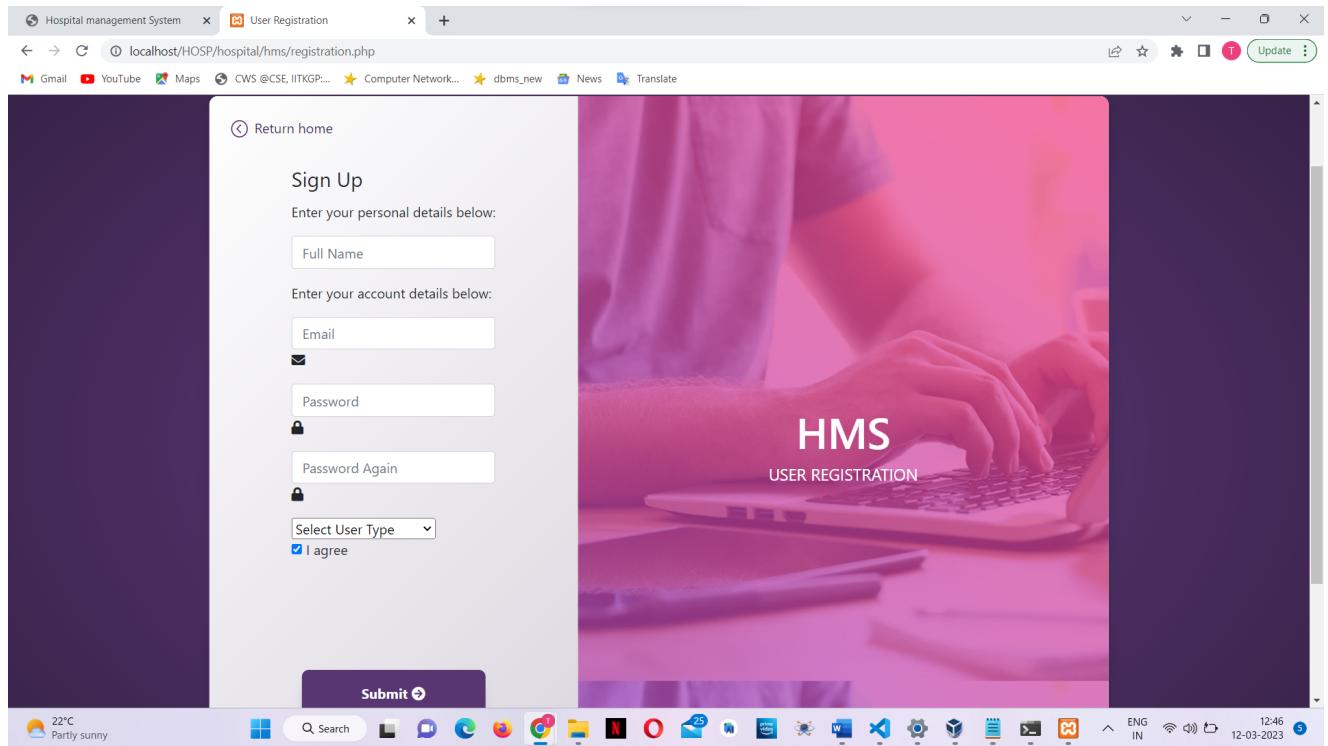
**Useful Links**

- > About us
- > Specialties
- > Logins
- > Contact us

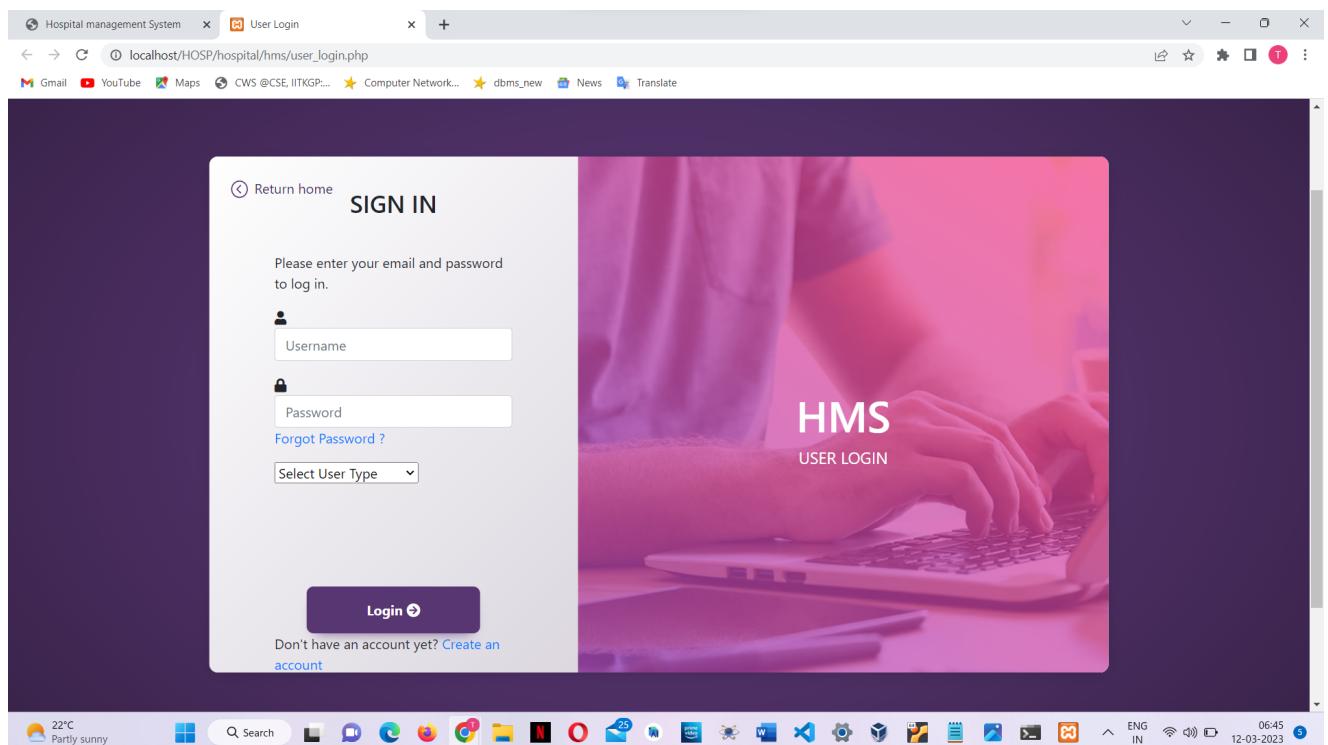
**Contact Us**

D-204, Hole Town South West, Delhi-110096, India  
Phone: 1122334455  
Email: info@gmail.com  
Timing: 9 am To 8 Pm

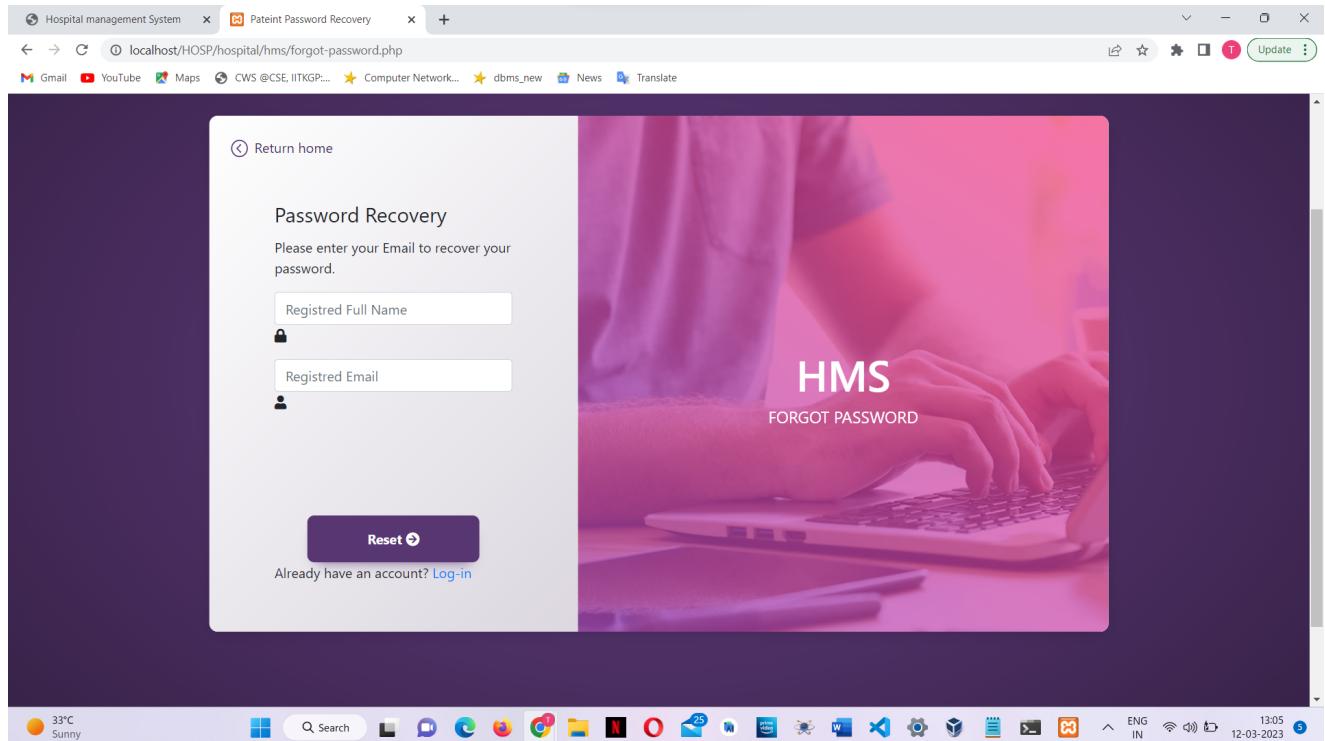
## Operator Registration:



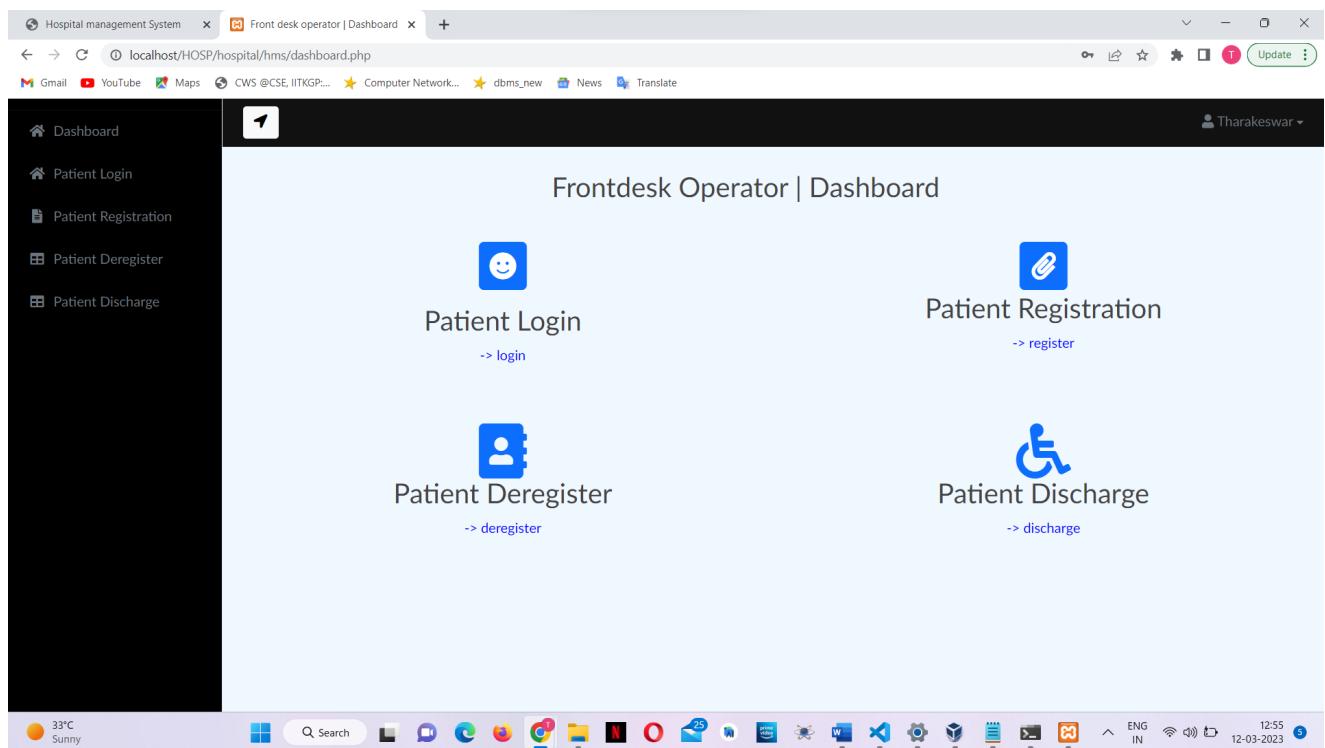
## Operator Login:



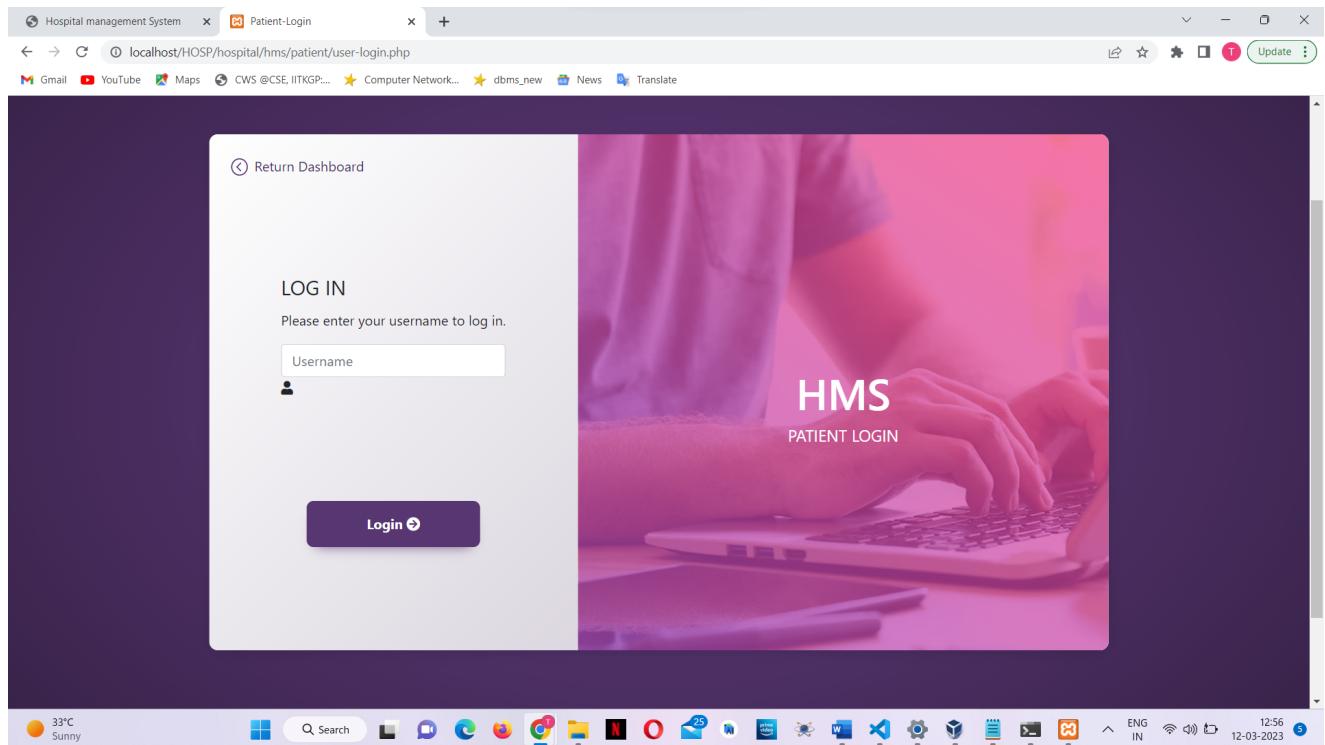
## Forgot Password:



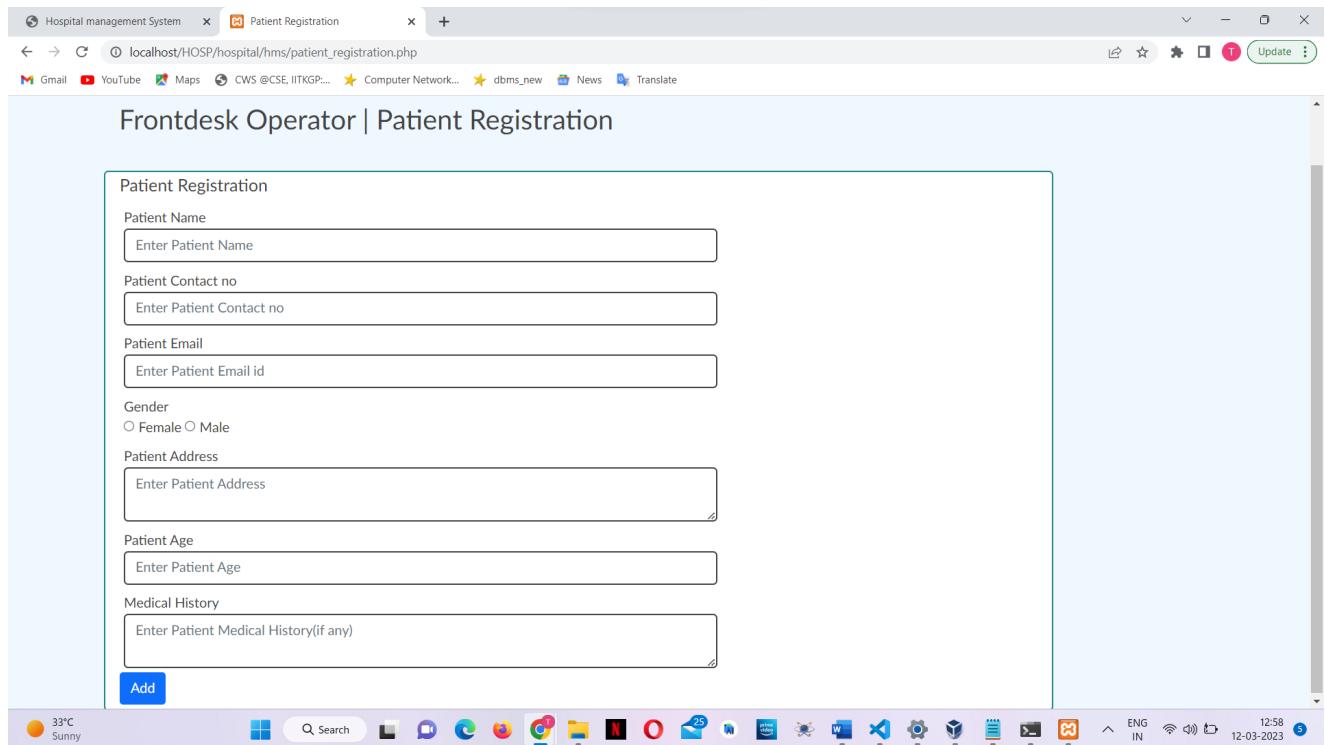
## Front-Desk Operator Dashboard:



## Patient Login:



## Patient Registration:



## Patient Deregistration:

Frontdesk Operator | Patient Deregister

#	Patient Name	Patient ID	Contact Number	Email	Gender	Address	Age
1.	Amit Kumar	1	1231231230	amitk@gmail.com	male	New Delhi india	35
2.	Amit	2	1231232230	amit@gmail.com	male	New Delhi india	33

Patient ID

Select Patient ID

Submit

33°C Sunny 12-03-2023 12:59 ENG IN

## Patient Discharge:

Frontdesk Operator | Patient Discharge

#	Patient Name	Stay ID	Room No	StayStart	StayEnd
---	--------------	---------	---------	-----------	---------

Stay ID

Select stayid

Submit

33°C Sunny 12-03-2023 13:00 ENG IN

## Data-Entry Operator Dashboard:

The screenshot shows a web browser window with three tabs open: "Hospital management System", "DataEntry Operator | Dashboard", and "Front desk operator | Dashboard". The "DataEntry Operator | Dashboard" tab is active. The main content area displays the "Data Entry Operator | Dashboard" title and a "Patient Health Information" section. This section includes a blue document icon, the text "Patient Health Information", and a placeholder "Enter username". On the left, there is a dark sidebar with links for "Dashboard" and "Patient Health Information". The bottom of the screen shows a Windows taskbar with various icons and system status.

## Patient Health Information:

The screenshot shows a web browser window with two tabs open: "Hospital management System" and "Reg Users | View Medical History". The "Reg Users | View Medical History" tab is active. The main content area displays the "Users | Medical History" title and a "View Medical History" table. The table has columns for #, Patient Name, Patient Contact Number, Patient Gender, Creation Date, Updation Date, and Action. One entry is listed: "1. Amit Kumar 1231231230 male 2022-11-06 18:48:31". The bottom of the screen shows a Windows taskbar with various icons and system status.

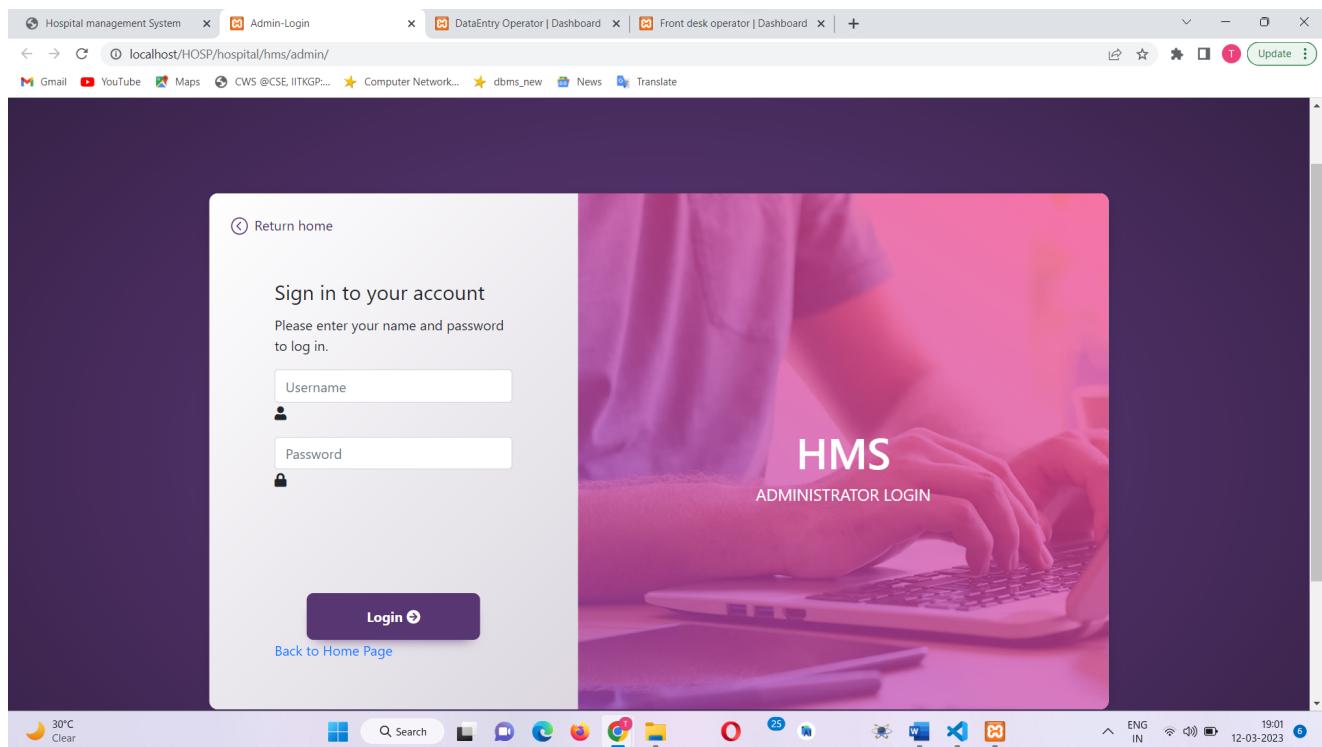
## Operator's Profile:

The screenshot shows a web browser window with the title 'User | Edit Profile'. The URL in the address bar is 'localhost/HOSP/hospital/hms/edit-profile.php'. The page displays 'Edit Profile' and 'Tharakeswar's Profile'. It shows the registration date as 'Profile Reg. Date: 2023-03-12 12:54:33'. There are fields for 'User Name' (Tharakeswar) and 'User Email' (tharak030620@gmail.com). A blue 'Update' button is visible. The browser's top bar includes tabs for 'Hospital management System' and 'User | Edit Profile', along with various system icons like search, file, and network. The status bar at the bottom shows weather (33°C, Sunny), system icons, and the date/time (12-03-2023, 13:02).

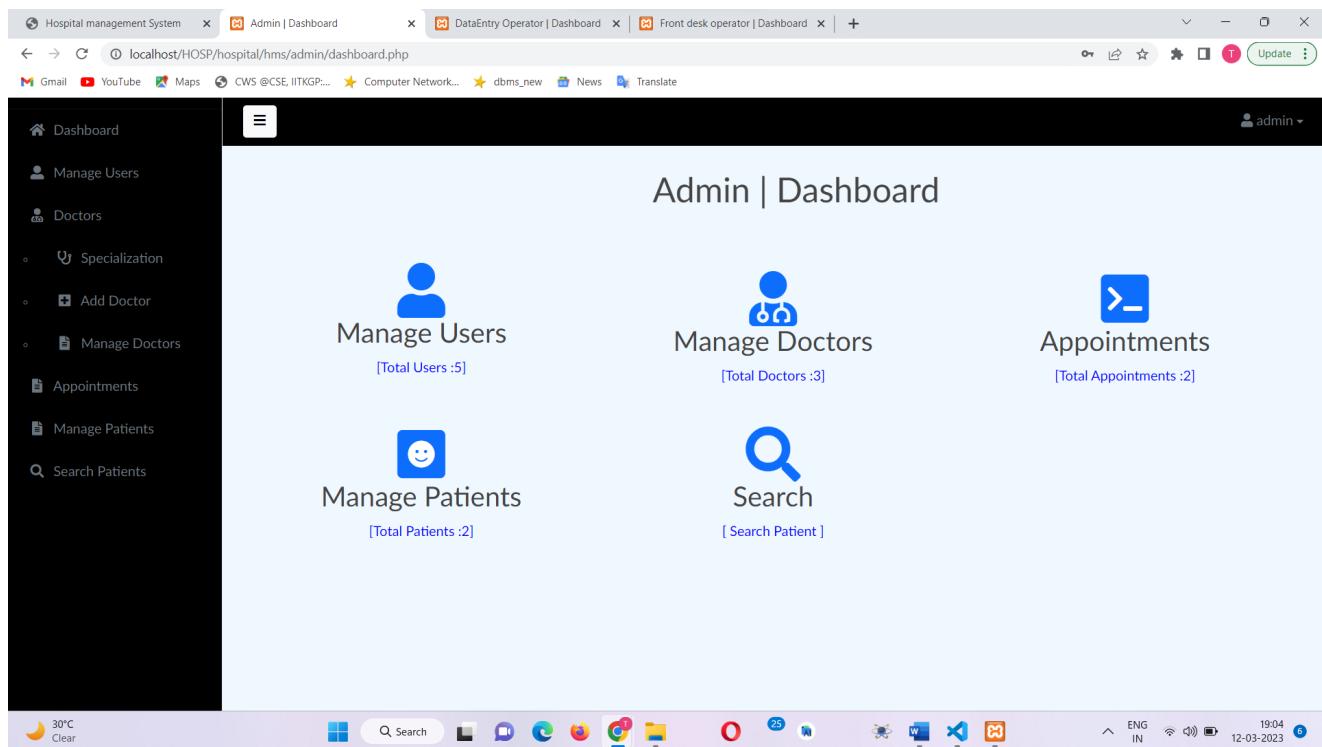
## Operator Change Password:

The screenshot shows a web browser window with the title 'User | Change Password'. The URL in the address bar is 'localhost/HOSP/hospital/hms/change-password.php'. The page displays 'Change Password' and fields for 'Current Password' (labeled 'Enter Current Password'), 'New Password', and 'Confirm Password'. A blue 'Submit' button is visible. The browser's top bar includes tabs for 'Hospital management System' and 'User | change Password', along with various system icons. The status bar at the bottom shows weather (33°C, Sunny), system icons, and the date/time (12-03-2023, 13:03).

## Admin Login:



## Admin Dashboard:



## Manage Users:

The screenshot shows a web browser window titled "Admin | Manage Users". The page displays a table of user information with the following columns: #, Full Name, Gender, Email, Creation Date, Updation Date, and Action. The data in the table is as follows:

#	Full Name	Gender	Email	Creation Date	Updation Date	Action
1.	John Doe		johndoe12@test.com	2022-11-06 17:43:56		X
2.	abhinav raj		abhi@gmai.com	2023-03-12 12:48:16		X
3.	rananaidu		rana@gmail.com	2023-03-12 12:50:34		X
4.	Tharakeswar		tharak030620@gmail.com	2023-03-12 12:54:33		X
5.	Giridhar		grdrdasari02@gmail.com	2023-03-12 13:09:29		X

The browser's address bar shows the URL: localhost/HOSP/hospital/hms/admin/manage-users.php. The top right corner of the browser window shows the user "admin". The taskbar at the bottom of the screen includes icons for various applications like File Explorer, Edge, and others, along with system status indicators like battery level and date/time (12-03-2023, 19:06).

## Manage Doctors:

The screenshot shows a web browser window titled "Admin | Manage Doctors". The page displays a table of doctor information with the following columns: #, Specialization, Doctor Name, Creation Date, and Action. The data in the table is as follows:

#	Specialization	Doctor Name	Creation Date	Action
1.	ENT	Anuj kumar	2022-10-30 23:46:52	X
2.	Endocrinologists	Charu Dua	2022-11-04 06:36:41	X
3.	Orthopedics	dinesh karthik	2023-03-11 19:37:45	X

The browser's address bar shows the URL: localhost/HOSP/hospital/hms/admin/manage-doctors.php. The top right corner of the browser window shows the user "admin". The taskbar at the bottom of the screen includes icons for various applications like File Explorer, Edge, and others, along with system status indicators like battery level and date/time (12-03-2023, 19:07).

## Appointments:

The screenshot shows a web browser window titled "Patients | Appointment History". The URL is "localhost/HOSP/hospital/hms/admin/appointment-history.php". The page displays a table with one row of data:

#	Doctor Name	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Anuj kumar	John Doe	ENT	500	2022-11-10 / 12:45 PM	2022-11-06 17:51:48	Cancel by Doctor	

The status column shows "Canceled" and the action column contains a blue button with a gear icon. The browser's address bar shows the full URL. The taskbar at the bottom includes icons for search, file, and various applications like Mail, Browser, and File Explorer.

## Manage Patients(Admin):

The screenshot shows a web browser window titled "Admin | View Patients". The URL is "localhost/HOSP/hospital/hms/admin/manage-patient.php". The page displays a table with two rows of data:

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Amit Kumar	1231231230	male	2022-11-06 18:48:31		
2.	Amit	1231232230	male	2022-11-08 18:48:31		

The browser's address bar shows the full URL. The taskbar at the bottom includes icons for search, file, and various applications like Mail, Browser, and File Explorer.

## Search patient:

Admin | View Patients

AdminView Patients

Search by Name/Mobile No.

Search

30°C Clear

ENG IN 12-03-2023 19:24

## Admin's Change Password :

Admin | Change Password

AdminChange Password

Change Password

Current Password

Enter Current Password

New Password

New Password

Confirm Password

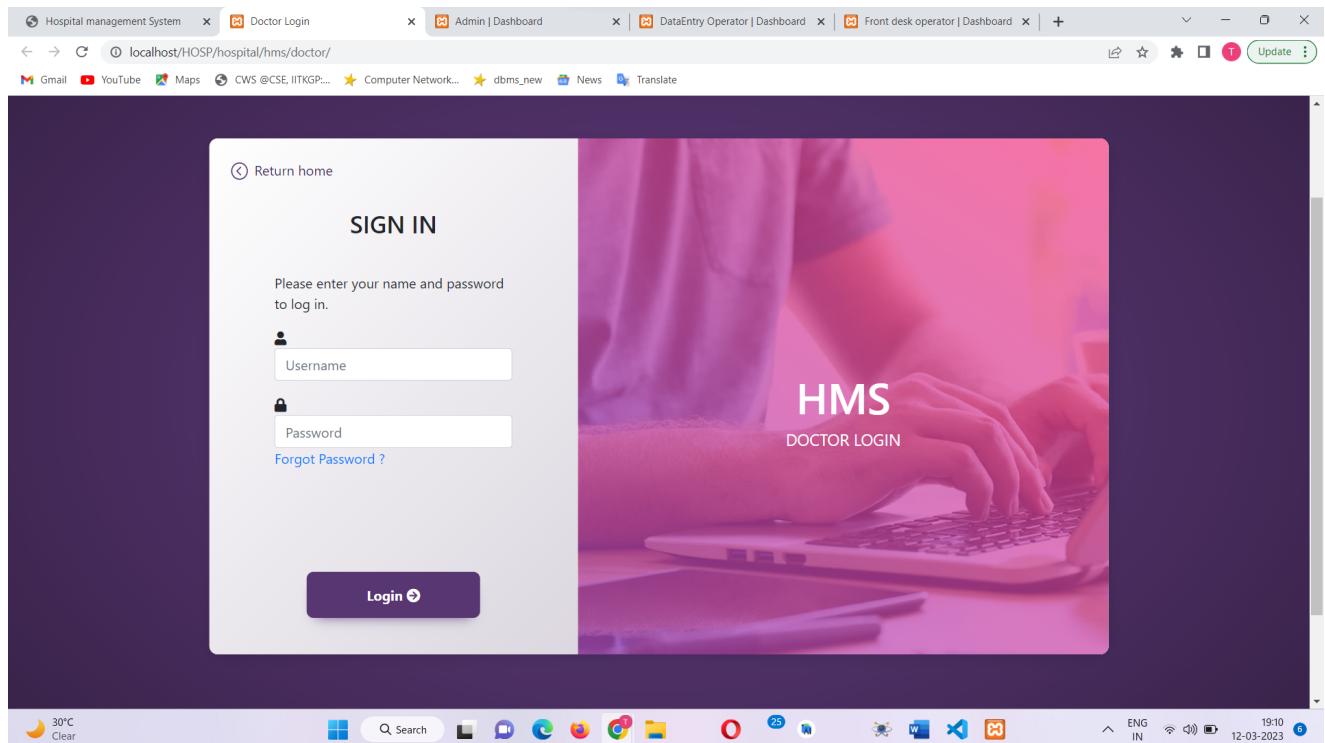
Confirm Password

Submit

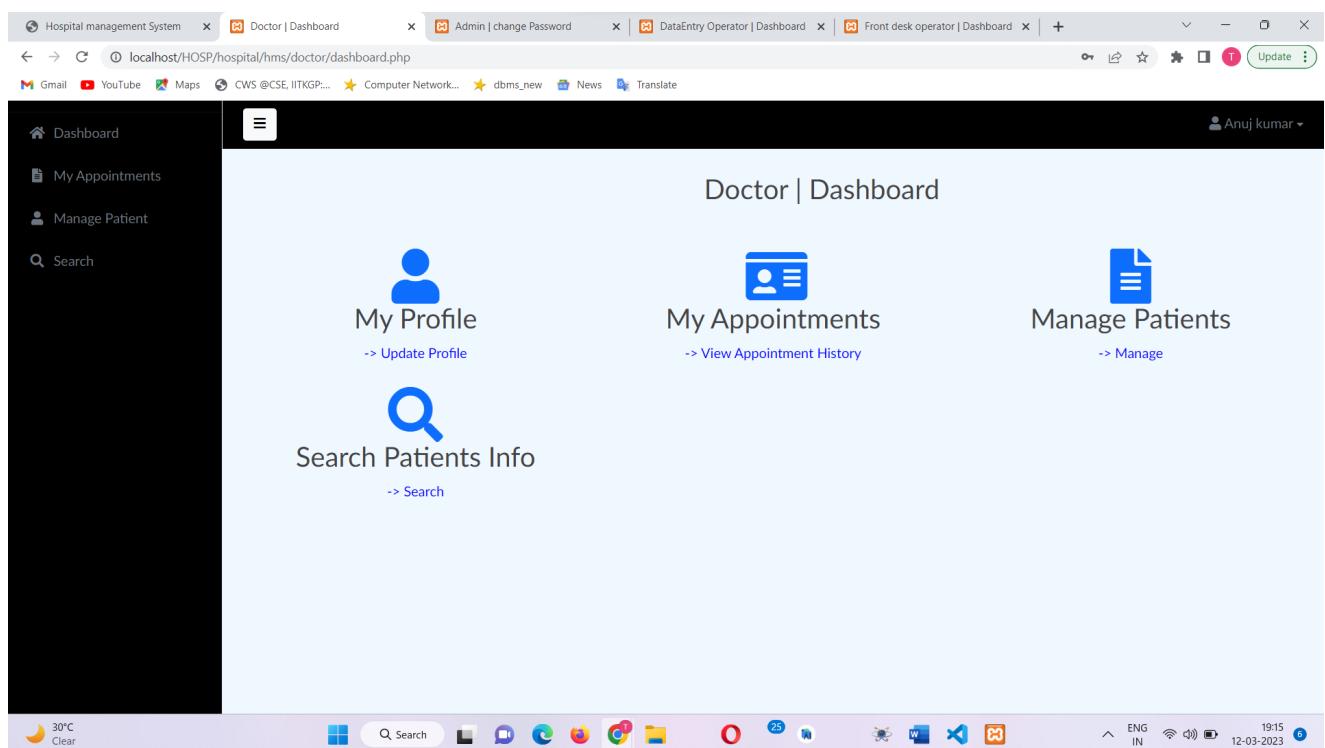
30°C Air: Moderate

ENG IN 12-03-2023 19:12

## Doctor Login:



## Doctor Dashboard:



## Doctor's Profile:

Edit Doctor  
Profile Reg. Date: 2022-10-30 23:46:52  
Profile Last Updation Date: 2022-11-06 18:50:17

Doctor Name  
Anuj kumar

Doctor Clinic Address  
A 123 XYZ Apartment Raj Nagar Ext Ghaziabad

Doctor Consultancy Fees  
500

Doctor Contact no  
142536250

Doctor Email  
anujk123@test.com

Update

## Doctor's Appointments:

Doctor | Appointment History

#	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Amit Kumar	ENT	500	2022-11-10 / 12:45 PM	2022-11-06 17:51:48	Cancel by you	Canceled
2.	Amit	ENT	500	2022-11-17 / 7:00 PM	2022-11-06 18:46:18	Active	<a href="#">Cancel</a>

## Manage Patient(Doctor):

Doctor | Manage Patients

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	View	Edit
1.	Amit Kumar	1231231230	male	2022-11-06 18:48:31			
2.	Amit	1231232230	male	2022-11-08 18:48:31			

30°C Clear

Search

ENG IN 19:20 12-03-2023

## Patient Info Search:

Doctor | Search Patients

Search by Name/Mobile No.

Search

30°C Clear

Search

ENG IN 19:22 12-03-2023

## Patient Dashboard(FDO):

The screenshot shows a web browser window titled "Patient | Dashboard" with the URL "localhost/HOSP/hospital/hms/patient/dashboard.php". The page has a dark header bar with a user profile for "Amit Kumar". On the left is a sidebar with links: "Dashboard", "Book Appointment", "Appointment History", "Medical History", and "Book Room". The main content area is titled "Patient | Dashboard" and contains five cards:

- My Profile**: Icon of a person, link to "Update Profile".
- My Appointments**: Icon of a clipboard with a pen, link to "View Appointment History".
- Book Appointment**: Icon of a plus sign inside a blue square, link to "Book Appointment".
- Medical History**: Icon of a person with a list, link to "View Medical History".
- Book Room**: Icon of a bed, link to "Book Room".

The taskbar at the bottom shows various application icons and the date/time: 12-03-2023, 12:57.

## Patient Dashboard(DEO):

The screenshot shows a web browser window titled "Patient | Dashboard" with the URL "localhost/HOSP/hospital/hms/patient/dashboard1.php". The page has a dark header bar with a user profile for "Amit Kumar". On the left is a sidebar with links: "Dashboard" and "Medical History". The main content area is titled "Patient | Dashboard" and contains three cards:

- My Profile**: Icon of a person, link to "Update Profile".
- Medical History**: Icon of a clipboard with a list, link to "View Medical History".
- Book Room**: Icon of a bed, link to "Book Room".

The taskbar at the bottom shows various application icons and the date/time: 12-03-2023, 19:31.

## Patient's Profile:

The screenshot shows a web browser window titled "Patient | Edit Profile". The page displays a form for editing a patient's profile. The patient's name is listed as "Amit Kumar Profile". The form includes fields for "Profile Creation Date", "PatientName" (Amit Kumar), "PatientContno" (1231231230), "PatientEmail" (amitk@gmail.com), "PatientGender" (male), "PatientAdd" (New Delhi india), and "PatientAge" (35). A blue "Update" button is located at the bottom left of the form area. The browser's address bar shows the URL "localhost/HOSP/hospital/hms/patient/edit-profile.php". The taskbar at the bottom of the screen shows various application icons and the date/time as 12-03-2023 19:31.

## Patient's Appointments:

The screenshot shows a web browser window titled "Patient | Appointment History". The page displays a table of the patient's appointments. The table has columns for #, Doctor Name, Specialization, Consultancy Fee, Appointment Date / Time, Appointment Creation Date, Current Status, and Action. There is one entry in the table: "1. Anuj kumar ENT 500 2022-11-10 / 12:45 PM 2022-11-06 17:51:48 Canceled". The browser's address bar shows the URL "localhost/HOSP/hospital/hms/patient/appointment-history.php". The taskbar at the bottom of the screen shows various application icons and the date/time as 12-03-2023 19:33.

## Book Appointment:

The screenshot shows a web browser window titled "Patient | Book Appointment". The main content area is a form titled "Book Appointment" with the following fields:

- Doctor Specialization: A dropdown menu labeled "Select Specialization".
- Doctors: A dropdown menu labeled "Select Doctor".
- Consultancy Fees: A text input field.
- Date: A text input field.
- Time: A text input field showing "7:45 PM" with a placeholder "eg : 10:00 PM".

A blue "Submit" button is located at the bottom left of the form. The browser's address bar shows the URL "localhost/HOSP/hospital/hms/patient/book-appointment.php". The taskbar at the bottom includes icons for various applications like Gmail, YouTube, Maps, and News, along with system status indicators for battery, signal, and volume.

## Book Room:

The screenshot shows a web browser window titled "User | Book Room". The main content area is a form titled "Book Room" with the following field:

- Room Type: A dropdown menu labeled "Select Room Type".

A blue "Submit" button is located at the bottom left of the form. The browser's address bar shows the URL "localhost/HOSP/hospital/hms/patient/book-room.php". The taskbar at the bottom includes icons for various applications like Gmail, YouTube, Maps, and News, along with system status indicators for battery, signal, and volume. A user profile for "Amit Kumar" is visible in the top right corner.

## Medical History(FDO):

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Users | Medical History'. The page title is 'Users | Medical History'. The content area contains a table for 'Patient Details' and another table for 'Medical History'.

**Patient Details**

Patient Name	Amit Kumar	Patient Email	amitk@gmail.com
Patient Mobile Number	1231231230	Patient Address	New Delhi India
Patient Gender	male	Patient Age	35
Patient Medical History(if any)	NA	Patient Reg Date	2022-11-06 18:48:31

**Medical History**

#	MID	Doctor Name	Blood Pressure	Weight	Blood Sugar	Body Temperature	Medical Prescription	Visit Date

## Medical History(DEO):

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Users | Medical History'. The page title is 'Users | Medical History'. The content area contains a table titled 'View Medical History'.

**View Medical History**

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Amit Kumar	1231231230	male	2022-11-06 18:48:31		<a href="#">Edit</a>

# Manage Patients(DEO):

## Data entry | Manage Patients

Manage Patients

Patient Details								
Patient Name			Amit Kumar		Patient Email		amitk@gmail.com	
Patient Mobile Number			1231231230		Patient Address		New Delhi india	
Patient Gender			male		Patient Age		35	
Patient Medical History(if any)			NA		Patient Reg Date		2022-11-06 18:48:31	
Medical History								
#	MID	Doctor Name	Blood Pressure	Weight	Blood Sugar	Body Temprature	Medical Prescription	Visit Date
Test Results								
#	mhid	testresult				Visit Date		
1	1	blood cells are less				2022-11-06 18:49:41		
Images								
#	mhid	Images			Upload date			
Add Images								

## **CONCLUSION**

The HMS was designed in such a way that future modifications can be done easily. The following conclusion can be deduced from the development of the project.

- Automation of the entire system improves efficiency.
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating information becomes so easy.
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.