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

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:

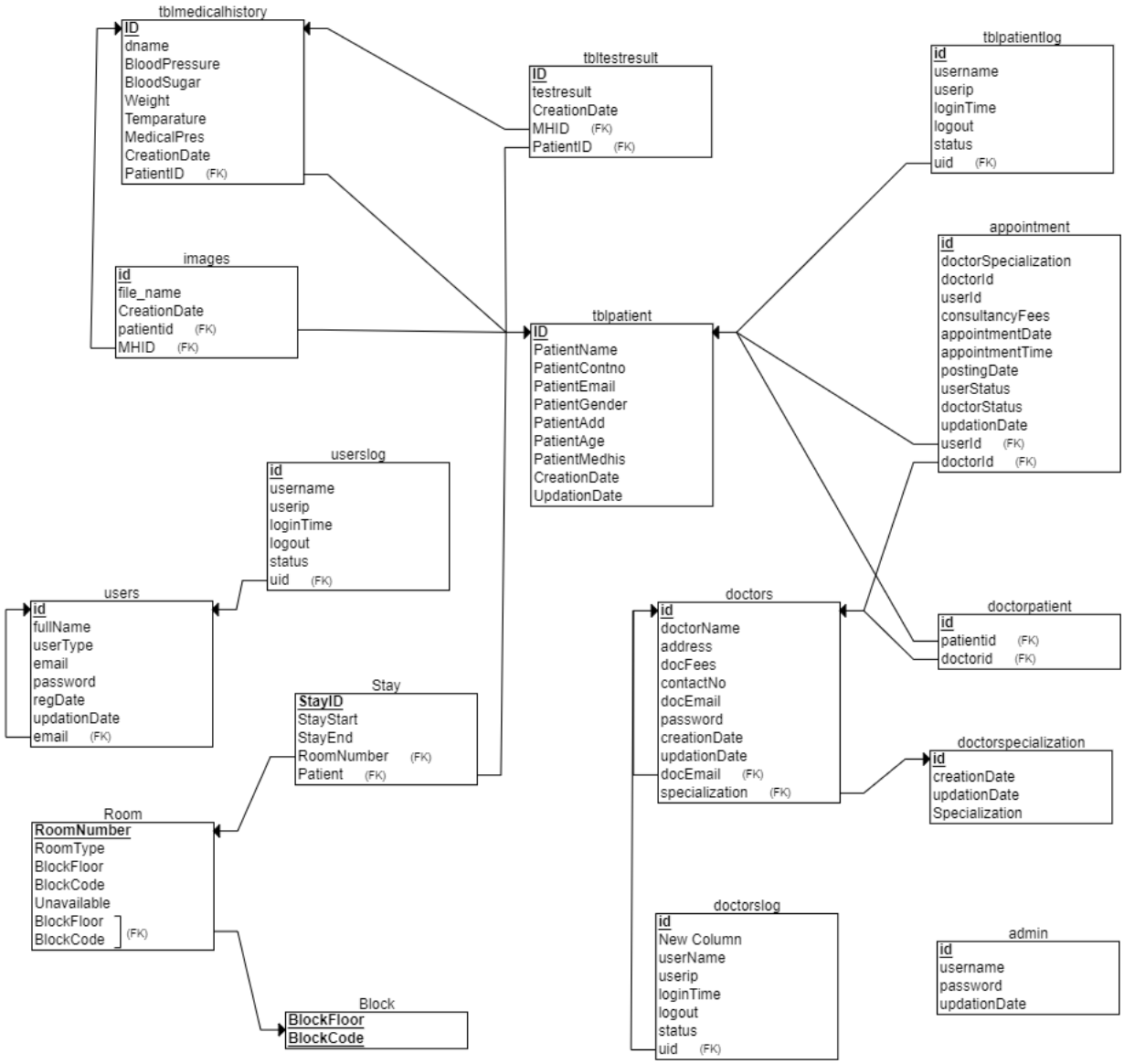
Hardware Requirements:-

- 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

RELATIONAL SCHEMA FOR HMS:



PROJECT MODULES

- ➢ Operators
- ➢ Admin
- ➢ Doctor
- ➢ Patient

Module-1: Operators

Description: The Operators module handles various enquiries connected with the frost-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, Codelgniter 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='$puname' and password='$ppwd' and usertype =
'$usertype' and usertype = 'dataentry'
```

```
insert into userlog(uid,username,userip,status)
values('$pid','$puname','$uip','$status')
```

```
insert into userlog(username,userip,status) values('$puname','$uip','$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='$puname' and password='$ppwd' and usertype = '$usertype' and usertype = 'frontdesk'

insert into userlog(uid,username,userip,status)
values('$pid','$puname','$uip','$status')
```

```
insert into userlog(username,userip,status) values('$puname','$uip','$status')
```

Patient_registration.php:

```
insert into
tblpatient(PatientName,PatientContno,PatientEmail,PatientGender,PatientAdd,PatientAge,PatientMedhis)
values('$patname','$patcontact','$patemail','$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='$puname' and password='$ppwd' and usertype =
'$usertype'
```

```
insert into userlog(uid,username,userip,status)
values('$pid','$puname','$uip','$status')
```

```
insert into userlog(username,userip,status) values('$puname','$uip','$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 doctorspecilization
```

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
```

betweendates-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-speacialization.php:

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

```
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
specilization='$docspecialization',doctorName='$docname',address='$docaddress'
,docFees='$docfees',contactno='$doccontactno',docEmail='$docemail' where
id='$did'
```

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

```
select * from doctorspecilization
```

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='$npass', updationDate='$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','$uip','$status')
```

```
insert into doctorslog(username,userip,status)
values('$uname','$uip','$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 tbltestresult  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 doctorspecilization
```

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='$Patie
ntEmail',PatientGender='$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='$Patie
ntEmail',PatientGender='$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='$puname'
```

```
insert into tblpatientlog(uid,username,userip,status)
values('$pid','$puname','$uip','$status')
```

```
insert into tblpatientlog(username,userip,status)
values('$puname','$uip','$status')
```

User-Login1.php:

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

```
insert into tblpatientlog(uid,username,userip,status)
values('$pid','$puname','$uip','$status')
```

```
insert into tblpatientlog(username,userip,status)
values('$puname','$uip','$status')
```

View-medhistory.php:

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

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

```
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 ;
```

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.