PRAGJYOTISH COLLEGE

PROJECT REPORT

ON

HOSPITAL MANAGEMENT SYSTEM

SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OF DEGREE

For

BACHELOR OF COMPUTER APPLICATION (B.C.A)

SUBMITTED BY: ABHISHEK GHOSH

GU ROLL NUMBER: UT-191-032-0001

REGISTRATION NUMBER: 19014340

UNDER THE GUIDANCE OF: -

PROF. NAYAN MAHANTA (H.O.D)

MRS. NIBEDITA DAS (GUIDE)

UNDER THE GUIDANCE OF: -

PROF. NAYAN MAHANTA (H.O.D)

MRS. NIBEDITA DAS (PROJECT GUIDE)

DEPARTMENT OF BCA

.

ACKNOWLEDGEMENT

It is indeed with a great pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We are highly indebted to our Principal Dr. Manoj Kumar Mahanta, Pragjyotish college, for the facilities provided to accomplish this main project.

We would like to thank our Prof. Nayan Mahanta, Head of the Department of Computer Science, Pragjyotish College, for this constructive criticism throughout our project.

We feel elated in manifesting our sense of gratitude tour internal project guide Mrs. Nibedita Das, Department of Computer Science, Pragjyotish College, she has been a constant source of inspiration for us and we are very deeply thankful to her for her support and valuable advice.

We are extremely grateful to our Departmental staff members, Lab technicians and Non-teaching staff members for their extreme help throughout our project.

Finally, we express our heart full thanks to all of our friends who helped us in successful completion of this project.

ABSTRACT

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

INDEX

CHAPTER 1

INTRODUCTION

1.1 Introduction:

The project Hospital Management system includes registration of patients, storing their details into the system. The software has the facility to give a unique id for every patient and stores the details of every patient.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

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

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

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

1.2 Problem Introduction:

Lack of immediate retrievals: -

The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient's history, the user has to go through various registers. This results in inconvenience and wastage of time.

Lack of immediate information storage: -

The information generated by various transactions takes time and efforts to be stored at right place.

Lack of prompt updating: -

Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved.

Error prone manual calculation: -

Manual calculations are error prone and take a lot of time this may result in incorrect information. For example, calculation of patient's bill based on various treatments.

Preparation of accurate and prompt reports: -

This becomes a difficult task as information is difficult to collect from various register.

Objective: -

1) Define hospital
2) Recording information about the Patients that come.
3) Generating bills.
4) Recording information related to diagnosis given to Patients.
5) Keeping record of the Immunization provided to children/patients.
6) Keeping information about various diseases and medicines available to cure them.

These are the various jobs that need to be done in a Hospital by the operational staff and Doctors. All these works are done on papers.

Scope of the Project: -

1) Information about Patients is done by just writing the Patients name, age and gender. Whenever the Patient comes up his information is stored freshly.
2) Bills are generated by recording price for each facility provided to Patient on a separate sheet and at last they all are summed up.
3) Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.
4) Immunization records of children are maintained in pre-formatted sheets, which are kept in a file.
5) Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can’t remember them at that time.

1.3 MODULES:

Hospital Management System is web application for hospital which manages doctors and patients. In this project, we use PHP and MySQL database.

The entire project mainly consists of 3 modules, which are

- ❖ Admin module
- ❖ User module (patient)
- ❖ Doctor module

1.3.1 Admin module:

1. Dashboard: In this section, admin can view the Patients, Doctors, Appointments and New queries.
2. Doctors: In this section, admin can add doctor’s specialization and mange doctors (Add/Update).
3. Users: In this section, admin can view users detail (who take online appointment) and also have right to delete irrelevant user.
4. Patients: In this section, admin can view patient’s details.
5. Appointment History: In this section, admin can view appointment history.
6. Contact us Queries: In this section, admin can view queries which are send by users.
7. Doctor Session Logs: In this section, admin can see login and logout time of doctor.
8. User Session Logs: In this section, admin can see login and logout time of user.
9. Reports: In this section, admin can view reports of patients in particular periods.
10. Patient Search: In this section, admin can search patient with the help of patient name and mobile number.

Admin can also change his/her own password.

1.3.2 User module (patient):

1. Dashboard: In this section, patients can view the his/her profile, Appointments and Book Appointment.
2. Book Appointment: In this section, Patient can book his/her appointment.
3. Appointment History: In this section, Patients can see his/her own appointment history.
4. Medical History: In this section, Patients can see his/her own appointment history.

User can update his/her profile, change the password and recover the password.

1.3.3 Doctor module:

1. Dashboard: In this section, doctor can view his/her own profile and online appointments.
2. Appointment History: In this section, Doctor can see patient's appointment history.
3. Patients: In this section, doctor can manage patients (Add/Update).
4. Search: In this section, doctor can search patient with the help of patient name and mobile number.

Doctor can also update his profile, change the password and recover the password.

CHAPTER 2

REQUIREMENTSPECIFICATION

2.1 INTRODUCTION:

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as(computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

2.2 HARDWARE REQUIREMENTS:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

HARDWARE REQUIREMENTS FOR PRESENT PROJECT:

PROCESSOR	:	Intel dual Core ,i3
RAM	:	1 GB
HARD DISK	:	80 GB

2.3 SOFTWARE REQUIREMENTS:

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

SOFTWARE REQUIREMENTS FOR PRESENT PROJECT:

OPERATING SYSTEM	:	Windows 7/ XP/8
FRONT END	:	HTML, css , JavaScript query

SERVER SIDE SCRIPT	:	PHP
DATABASE	:	MySQL

CHAPTER 3

ANALYSIS

3.1EXISTING SYSTEM:

Hospitals currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the hospital management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

3.2 PROPOSED SYSTEM:

The Hospital Management System is designed for any hospital to replace their existing manual paper based system. The new system is to control the information of patients. 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.

3.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

3.3.1 Economic Feasibility

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products have to be purchased.

3.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

3.3.3 Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.4 SOFTWARE SPECIFICATION

HTML:

HTML or **Hypertext Markup Language** is the standard markup language used to create web pages.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

CASCADING STYLE SHEETS (CSS):

It is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.[1] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content .

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However, if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

MySQL:

MySQL is developed, distributed, and supported by Oracle Corporation. MySQL is a database system used on the web it runs on a server. MySQL is ideal for both small and large applications. It is very fast, reliable, and easy to use. It supports standard SQL. MySQL can be compiled on a number of platforms.

The data in MySQL is stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful when storing information categorically.

FEATURES OF MySQL:

Internals and portability:

- Written in C and C++.
- Tested with a broad range of different compilers.
- Works on many different platforms.
- Tested with Purify (a commercial memory leakage detector) as well as with Val grind, a GPL tool.
- Uses multi-layered server design with independent modules.

Security:

- A privilege and password system that is very flexible and secure, and that enables host-based verification.
- Password security by encryption of all password traffic when you connect to a server.

Scalability and Limits:

- Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
- Support for up to 64 indexes per table (32 before MySQL 4.1.2). Each index may consist of 1 to 16 columns or parts of columns. The maximum index width is 767 bytes for **InnoDB** tables, or

1000 for **MyISAM**; before MySQL 4.1.2, the limit is 500 bytes. An index may use a prefix of a column for **CHAR**, **VARCHAR**, **BLOB**, or **TEXT** column types.

CONNECTIVITY:

Clients can connect to MySQL Server using several protocols:

- Clients can connect using TCP/IP sockets on any platform.
- On Windows systems in the NT family (NT, 2000, XP, 2003, or Vista), clients can connect using named pipes if the server is started with the --enable-named-pipe option. In MySQL 4.1 and higher, Windows servers also support shared-memory connections if started with the --shared-memory option. Clients can connect through shared memory by using the --protocol=memory option.
- On UNIX systems, clients can connect using Unix domain socket files.

LOCALIZATION:

- The server can provide error messages to clients in many languages.
- All data is saved in the chosen character set.

CLIENTS AND TOOLS:

- MySQL includes several client and utility programs. These include both command-line programs such as **MySQL dump** and **mysqladmin**, and graphical programs such as MySQL Workbench.
- MySQL Server has built-in support for SQL statements to check, optimize, and repair tables. These statements are available from the command line through the **MySQL check** client. MySQL also includes **myisamchk**, a very fast command-line utility for performing these operations on **MyISAM** tables.
- MySQL programs can be invoked with the --help or -? option to obtain online assistance.

WHY TO USE MySQL:

- Leading open source RDBMS
- Ease of use – No frills
- Fast
- Robust
- Security
- Multiple OS support

- Free
- Technical support
- Support large database– up to 50 million rows, file size limit up to 8 Million TB

JAVASCRIPT:

JavaScript is the scripting language of the Web. All modern HTML pages are using JavaScript. A scripting language is a lightweight programming language. JavaScript code can be inserted into any HTML page, and it can be executed by all types of web browsers. JavaScript is easy to learn.

WHY TO USE JAVASCRIPT:

JavaScript is one of the 3 languages all web developers must learn:

1. HTML to define the content of web pages
2. CSS to specify the layout of web pages
3. JavaScript to specify the behavior of web pages

Example

```
x = document.getElementById("demo");  //Find the HTML element with id="demo"
x.innerHTML = "Hello JavaScript";     //Change the content of the HTML element
```

document. getElementById() is one of the most commonly used HTML DOM methods.

OTHER USES OF JAVASCRIPT:

- Delete HTML elements
- Create new HTML elements
- Copy HTML elements
- In HTML, JavaScript is a sequence of statements that can be executed by the web browser.

JAVASCRIPT STATEMENTS:

- JavaScript statements are "commands" to the browser.
- The purpose of the statements is to tell the browser what to do.
- This JavaScript statement tells the browser to write "Hello Dolly" inside an HTML element with id="demo":

Semicolon;

- Semicolon separates JavaScript statements.
- Normally you add a semicolon at the end of each executable statement.
- Using semicolons also makes it possible to write many statements on one line.

JAVASCRIPT CODE:

- JavaScript code (or just JavaScript) is a sequence of JavaScript statements.
- Each statement is executed by the browser in the sequence they are written.
- This example will manipulate two HTML elements:
- Example
- document.getElementById("demo").innerHTML="Hello Dolly";
  document.getElementById("myDIV").innerHTML="How are you?";

JAVASCRIPT PROPERTIES:

- Properties are the values associated with a JavaScript object.
- A JavaScript object is a collection of unordered properties.
- Properties can usually be changed, added, and deleted, but some are read only.

PHP:

WHAT IS PHP?

- PHP is an acronym for "PHP Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP costs nothing, it is free to download and use

WHAT IS PHP FILE?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

WHAT CAN PHP DO?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server

- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can restrict users to access some pages on your website
- PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

WHY PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net

CHAPTER 4

DESIGN

4.1 SYSTEM DESIGN:

4.1.1INTRODUCTION TO UML:

UML Design

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language, which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

Visualizing
Specifying
Constructing
Documenting

Visualizing

Through UML we see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize, how the components of the system communicate and interact with each other.

Specifying

Specifying means building, models that are precise, unambiguous and complete UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

Constructing

UML models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

Documenting

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating about a system during its developing requirements, architecture, desire, source code, project plans, tests, prototypes releasers, etc...

4.2 UML Approach

UML Diagram

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs. you draw diagram to visualize a system from different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams, or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams:

1. Class diagram
2. Object diagram
3. Use case diagram
4. Sequence diagram
5. Collaboration diagram
6. State chart diagram
7. Activity diagram
8. Component diagram
9. Deployment diagram

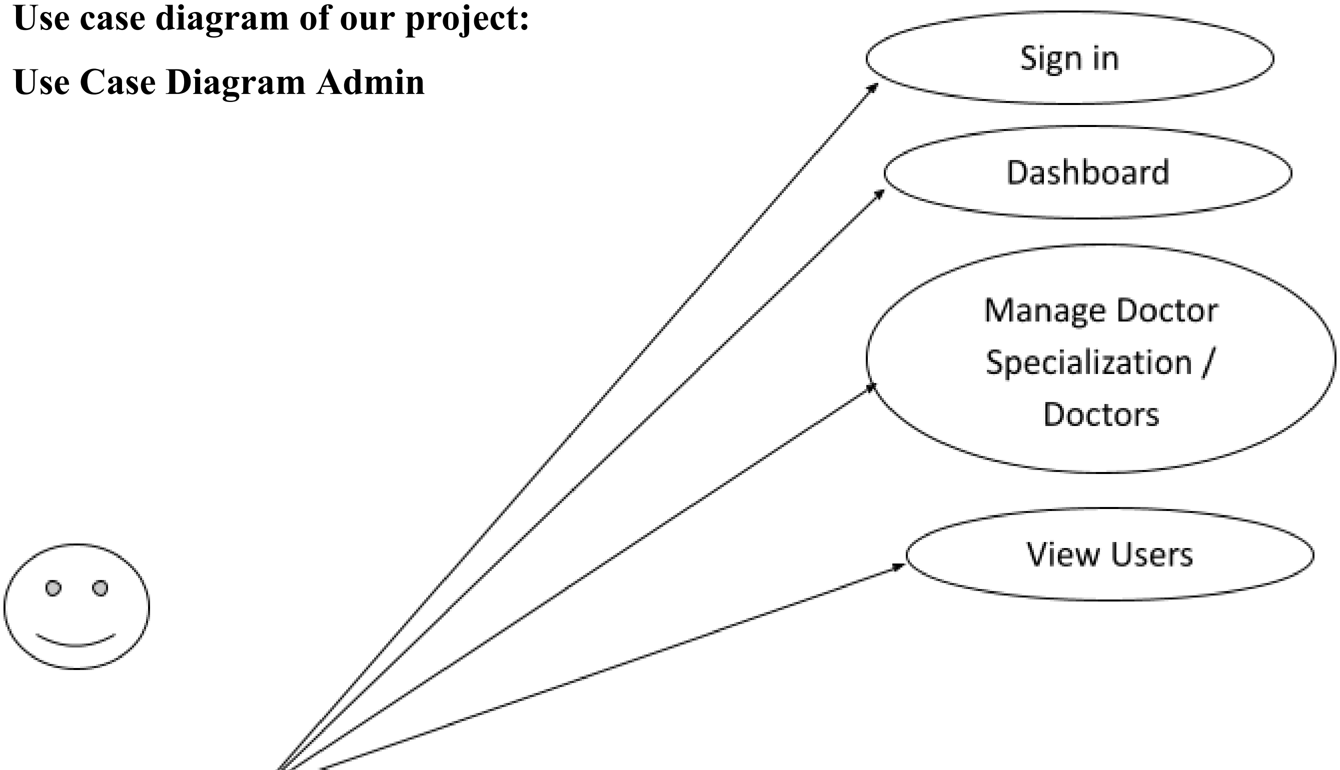
USE CASE DIAGRAM:

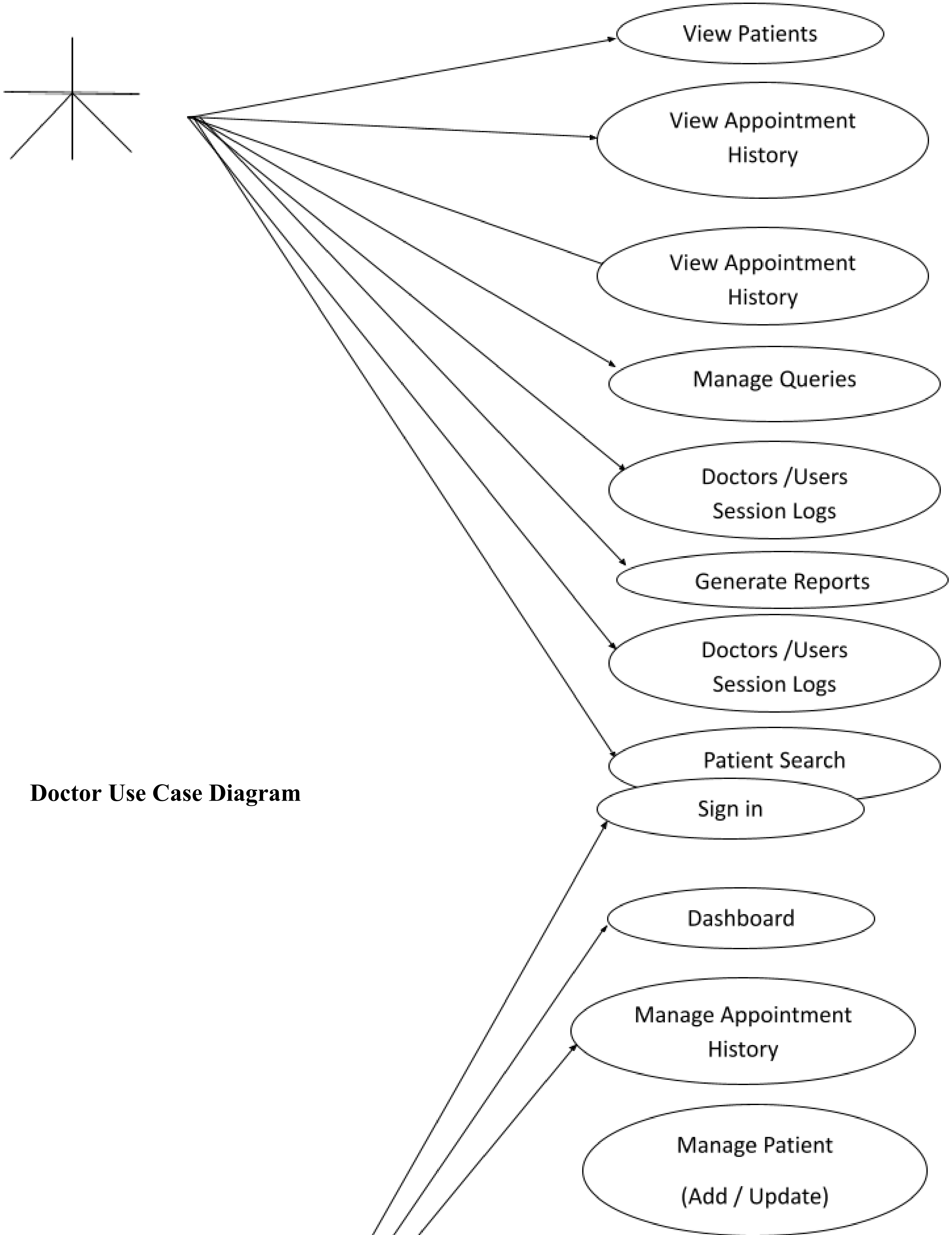
A use case diagram in the Unified Modeling Language(UML) is type of behavioral diagram defined by and created from a use-case analysis. its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

Use case diagrams are formally included in two modeling languages defined by the OMG: theunfied modeling language(UML) and the systems modeling language(sysML)

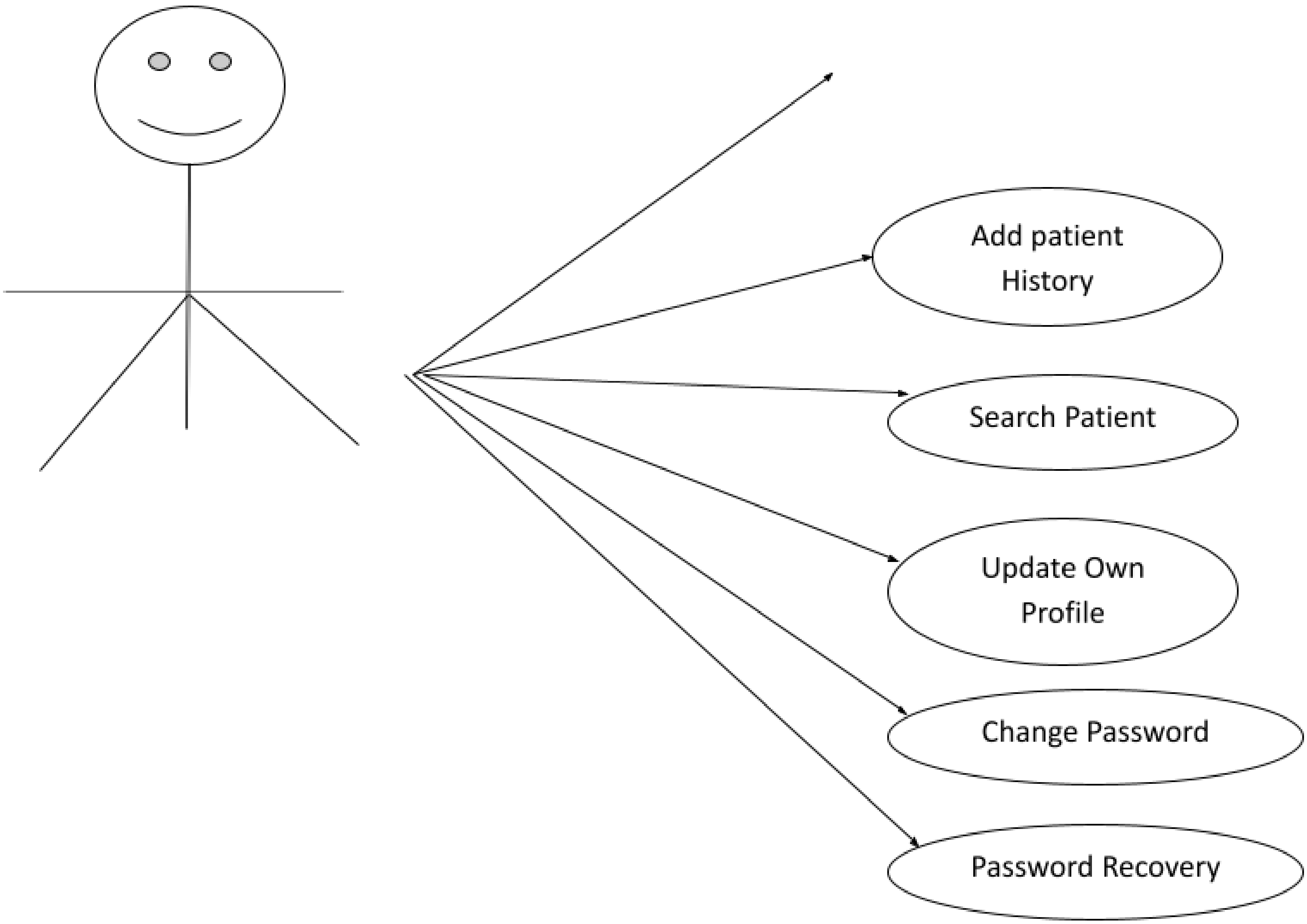
Use case diagram of our project:

Use Case Diagram Admin

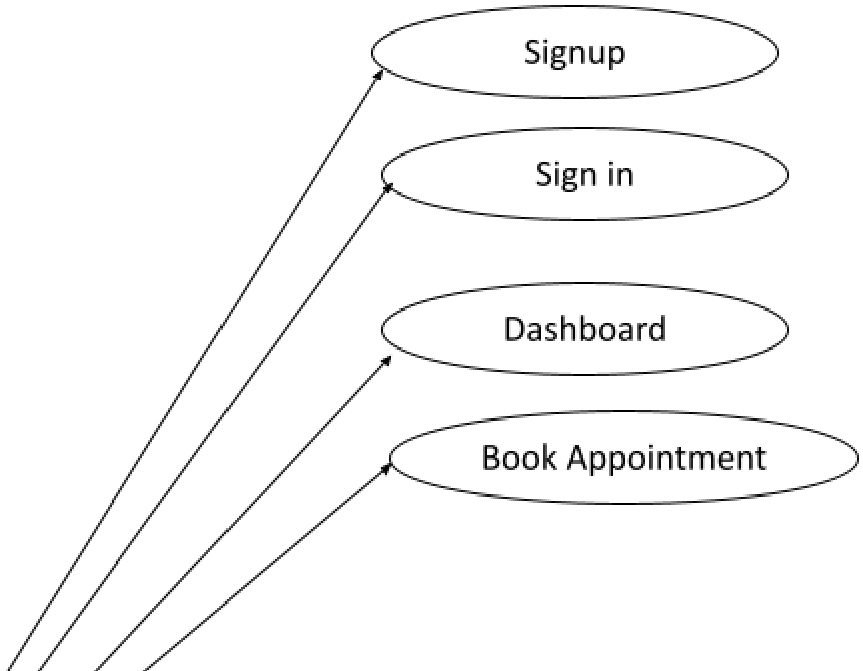


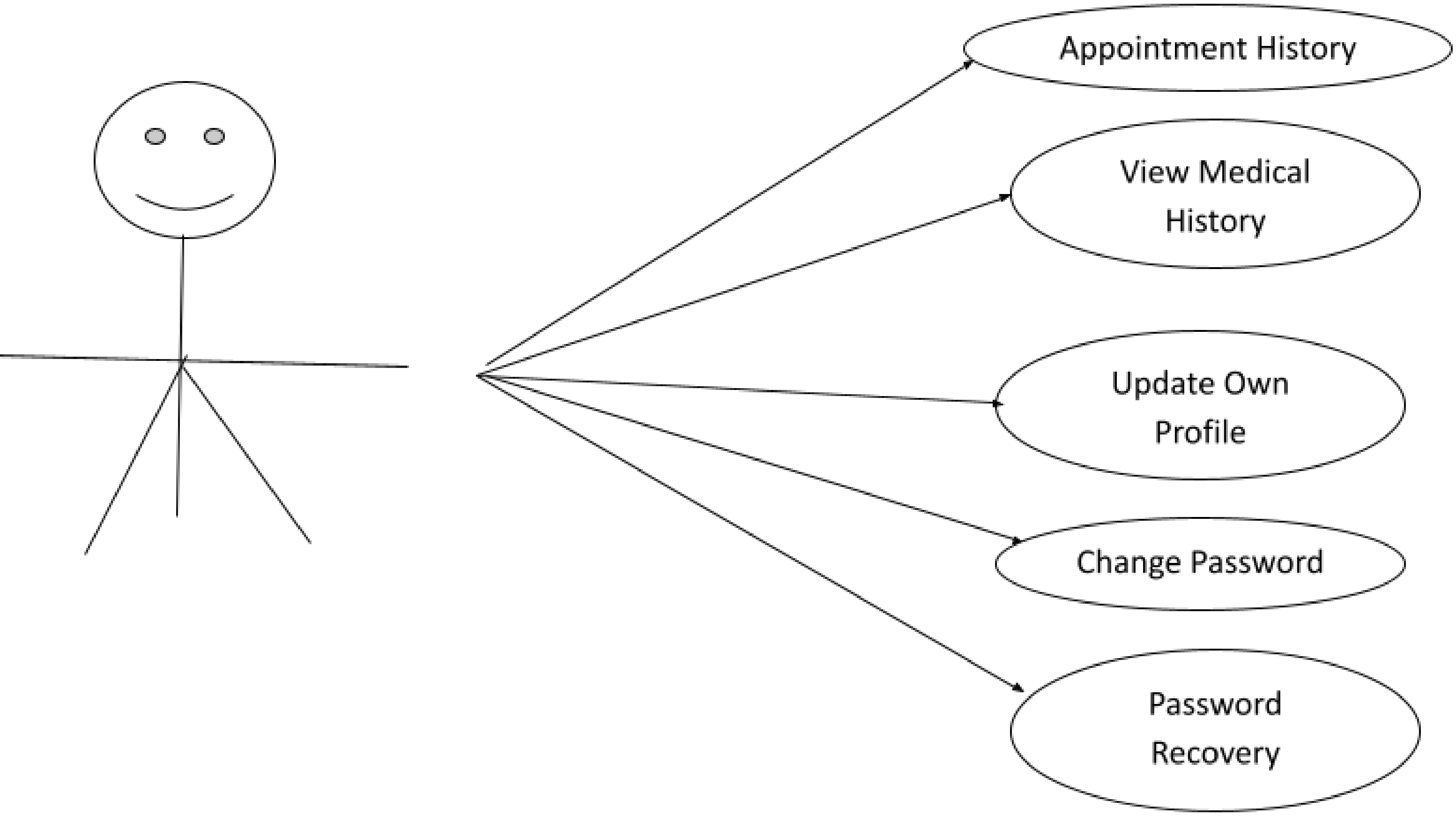


Doctor Use Case Diagram



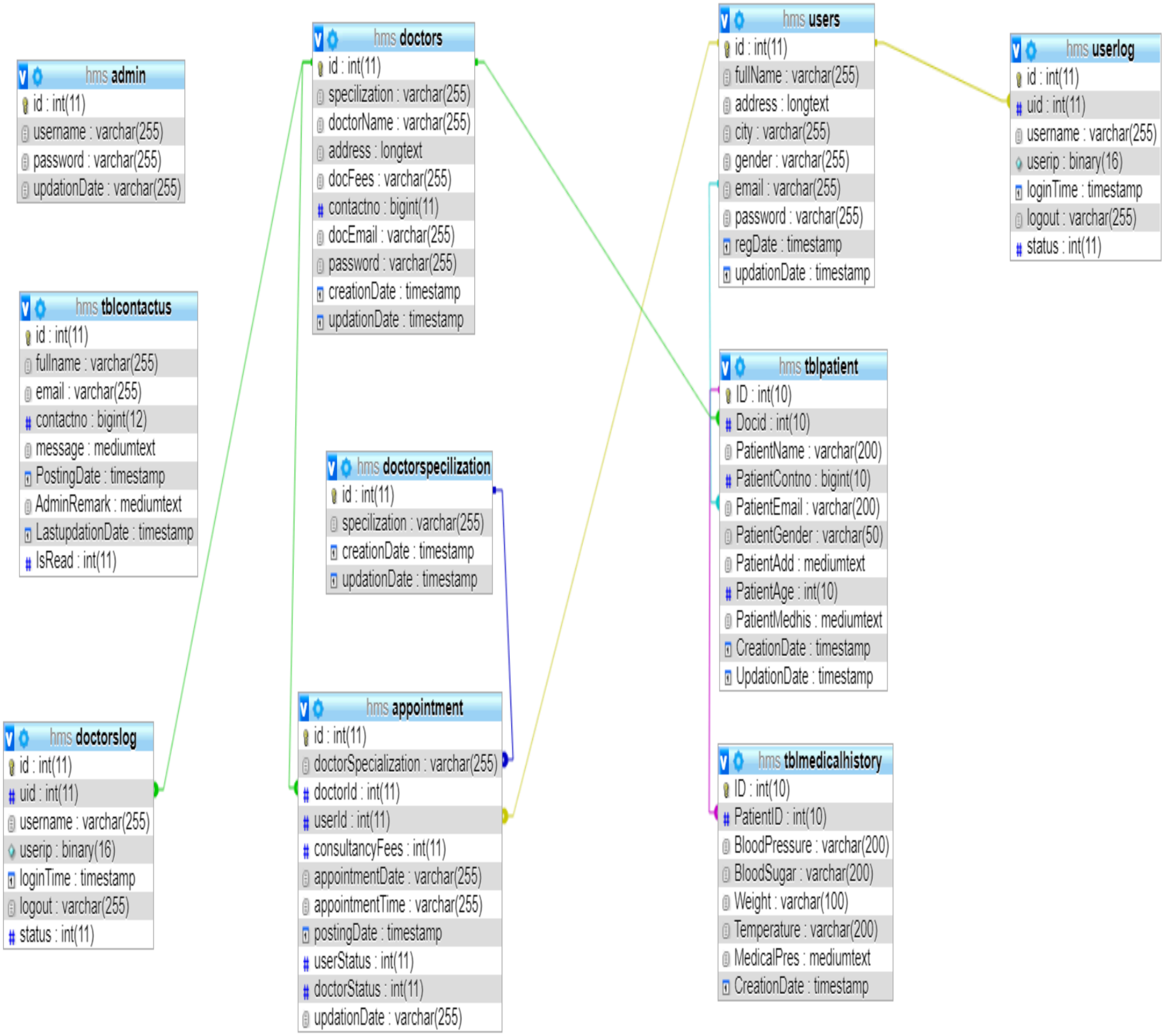
Patient Use Case Diagram



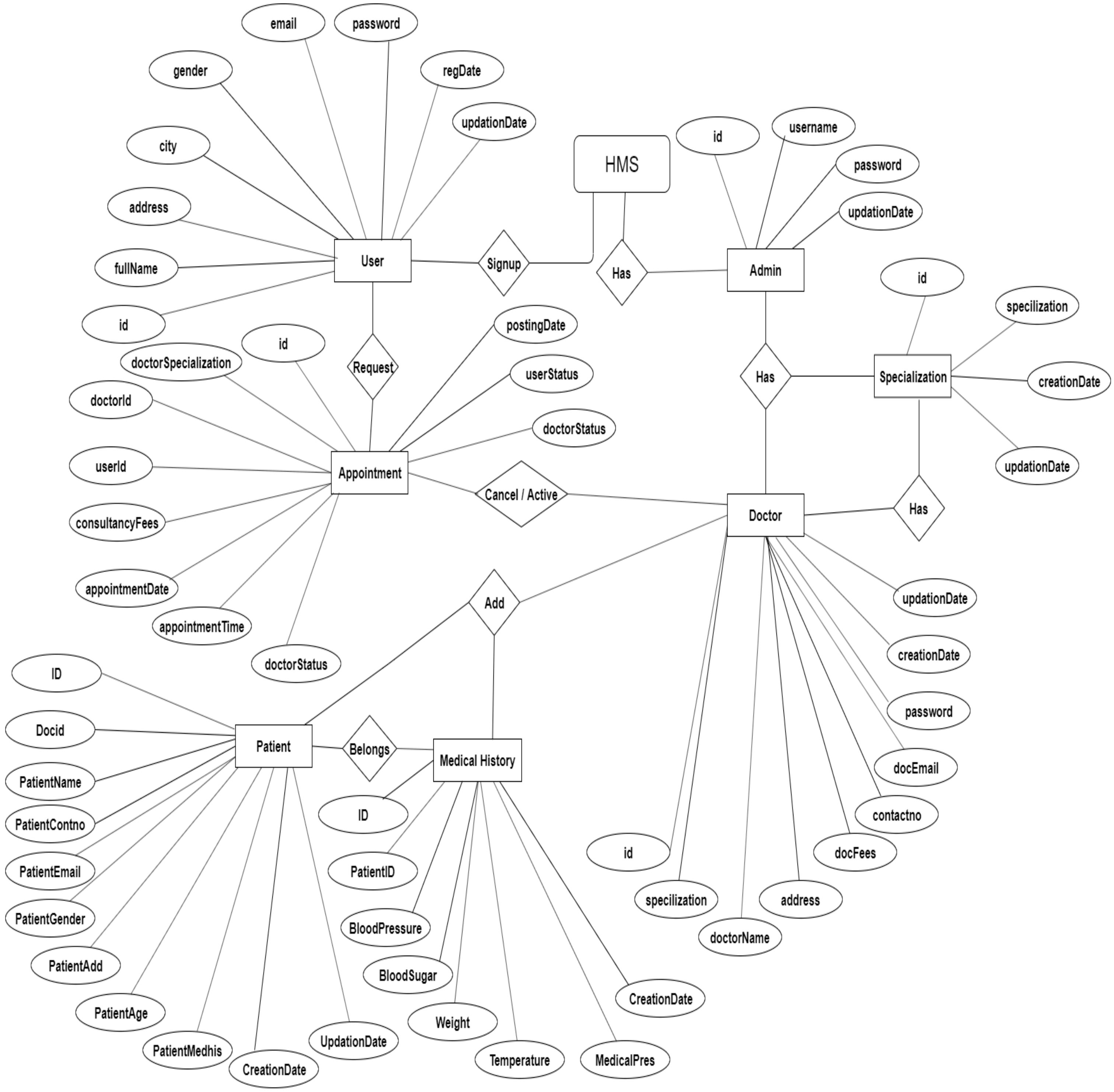


Class Diagram:

A Class is a category or group of things that has similar attributes and common behavior. A Rectangle is the icon that represents the class it is divided into three areas. The upper most area contains the name, the middle; area contains the attributes and the lowest areas show the operations. Class diagrams provides the representation that developers work from. Class diagrams help on the analysis side, too.

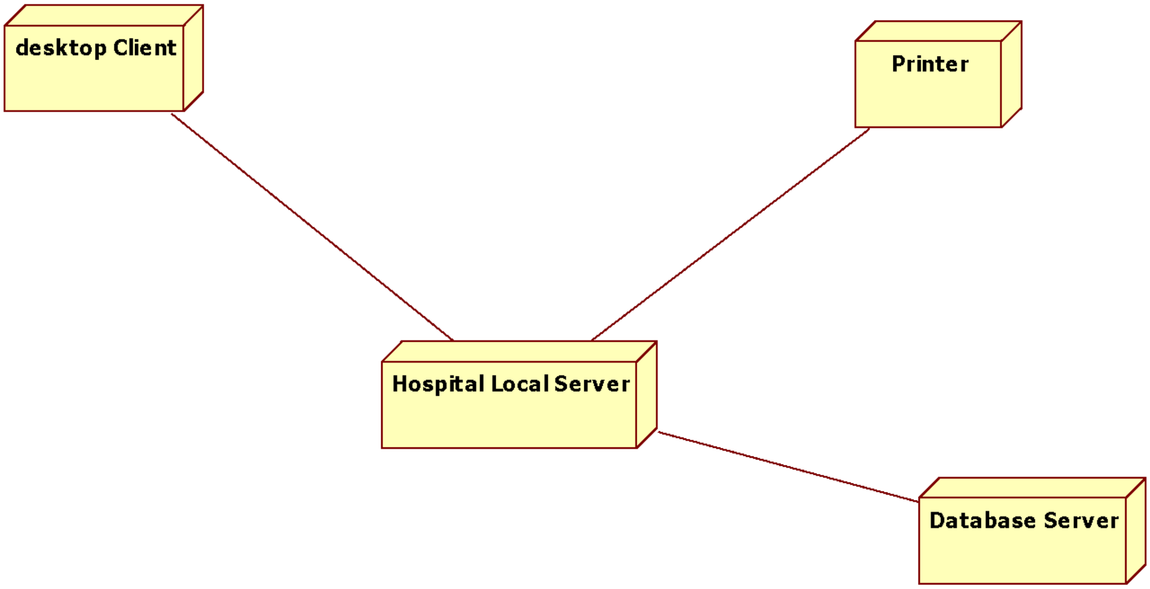

hms admin
id : int(11)
username : varchar(255)
password : varchar(255)
updationDate : varchar(255)
hms doctors
id : int(11)
specilization : varchar(255)
doctorName : varchar(255)
address : longtext
docFees : varchar(255)
contactno : bigint(11)
docEmail : varchar(255)
password : varchar(255)
creationDate : timestamp
updationDate : timestamp
hms users
id : int(11)
fullName : varchar(255)
address : longtext
city : varchar(255)
gender : varchar(255)
email : varchar(255)
password : varchar(255)
regDate : timestamp
updationDate : timestamp
hms userlog
id : int(11)
uid : int(11)
username : varchar(255)
userip : binary(16)
loginTime : timestamp
logout : varchar(255)
status : int(11)
hms tblcontactus
id : int(11)
fullname : varchar(255)
email : varchar(255)
contactno : bigint(12)
message : mediumtext
PostingDate : timestamp
AdminRemark : mediumtext
LastupdationDate : timestamp
IsRead : int(11)
hms doctorspecilization
id : int(11)
specilization : varchar(255)
creationDate : timestamp
updationDate : timestamp
hms tblpatient
ID : int(10)
Docid : int(10)
PatientName : varchar(200)
PatientContno : bigint(10)
PatientEmail : varchar(200)
PatientGender : varchar(50)
PatientAdd : mediumtext
PatientAge : int(10)
PatientMedhis : mediumtext
CreationDate : timestamp
UpdationDate : timestamp
hms doctorslog
id : int(11)
uid : int(11)
username : varchar(255)
userip : binary(16)
loginTime : timestamp
logout : varchar(255)
status : int(11)
hms appointment
id : int(11)
doctorSpecialization : varchar(255)
doctorId : int(11)
userId : int(11)
consultancyFees : int(11)
appointmentDate : varchar(255)
appointmentTime : varchar(255)
postingDate : timestamp
userStatus : int(11)
doctorStatus : int(11)
updationDate : varchar(255)
hms tblmedicalhistory
ID : int(10)
PatientID : int(10)
BloodPressure : varchar(200)
BloodSugar : varchar(200)
Weight : varchar(100)
Temperature : varchar(200)
MedicalPres : mediumtext
CreationDate : timestamp

ER Diagram



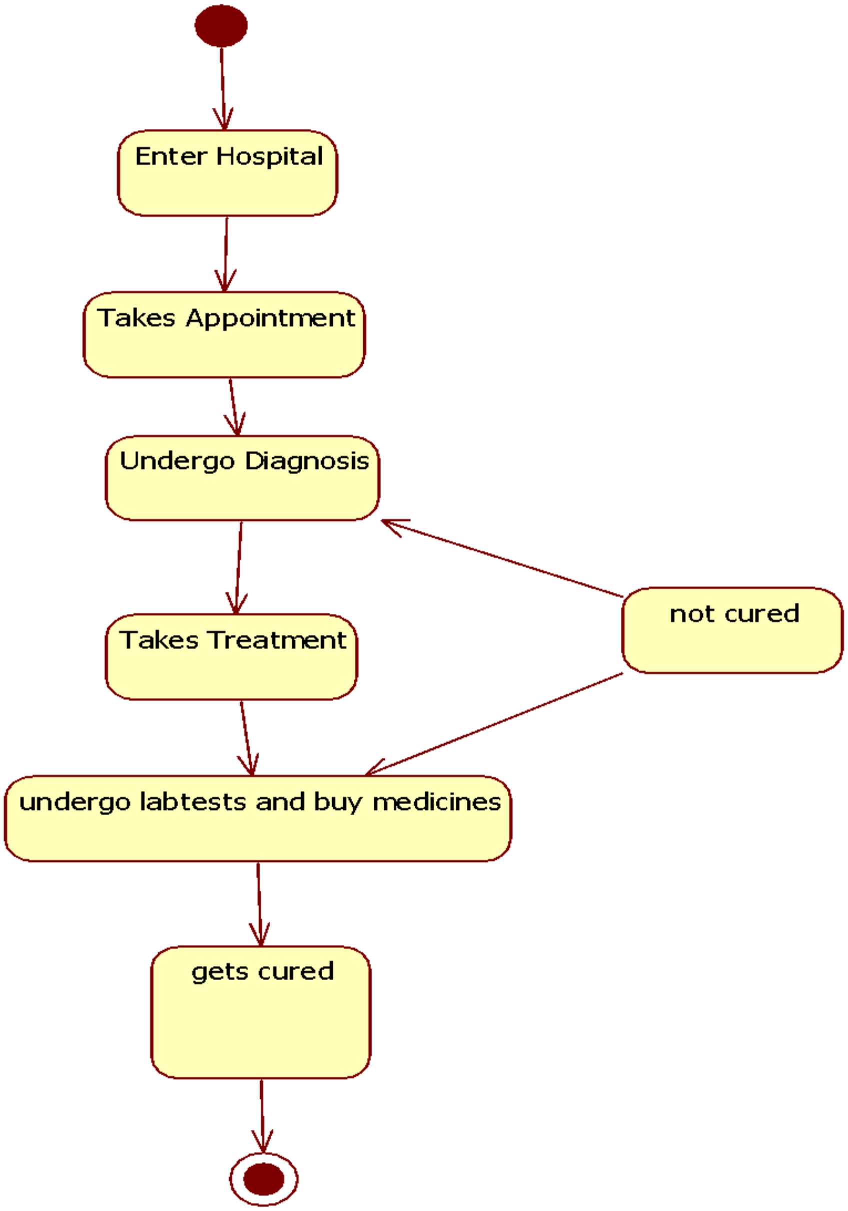
Deployment diagram:

A **Deployment Diagram** shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams address the static deployment view of architecture. They are related to component diagrams in that a node typically encloses one or more components.



State chart Diagrams:

The state diagram shows the states of an object and represents activities as arrows connecting the states. The Activity Diagram highlights the activities. Each activity is represented by a rounded rectangle-narrower and more oval-shaped than the state icon. An arrow represents the transition from the one activity to the next. The activity diagram has a starting point represented by filled-in circle, and an end point represented by bull’s eye.


Enter Hospital
Takes Appointment
Undergo Diagnosis
Takes Treatment
not cured
undergo labtests and buy medicines
gets cured

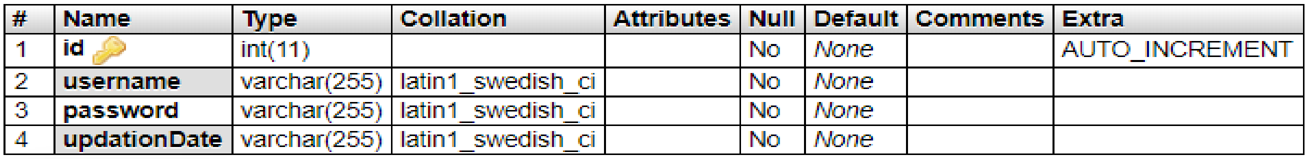
DATABASE DESIGN

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

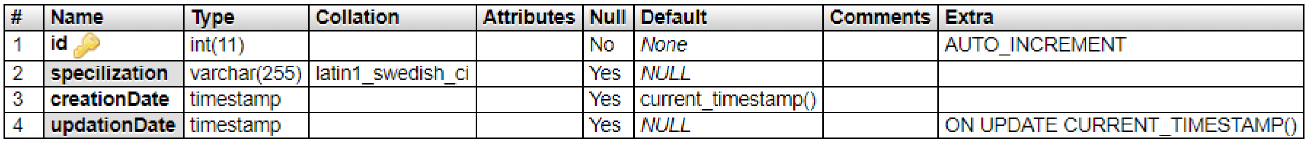
Hospital Management System (hms) contains 10 MySQL tables:

admin table Structure: This table store the login details of admin.



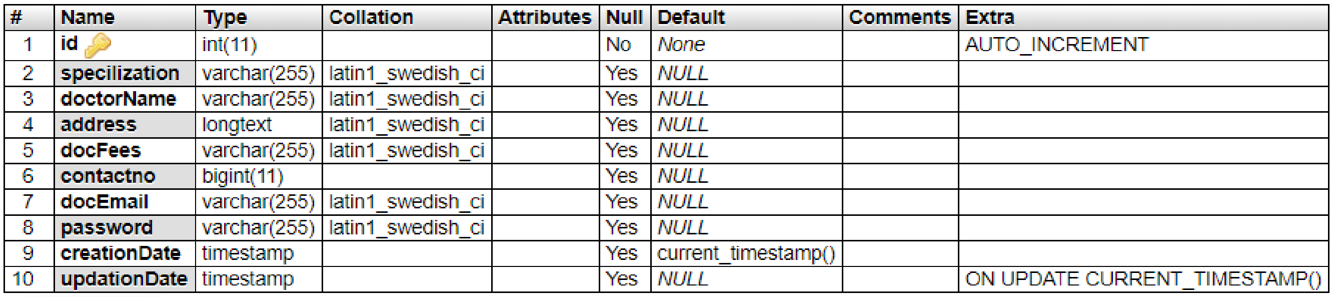
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	username	varchar(255)	latin1_swedish_ci		No	*None*		
3	password	varchar(255)	latin1_swedish_ci		No	*None*		
4	updationDate	varchar(255)	latin1_swedish_ci		No	*None*		

doctorspecilization table Structure: This table store the specializations of doctors.



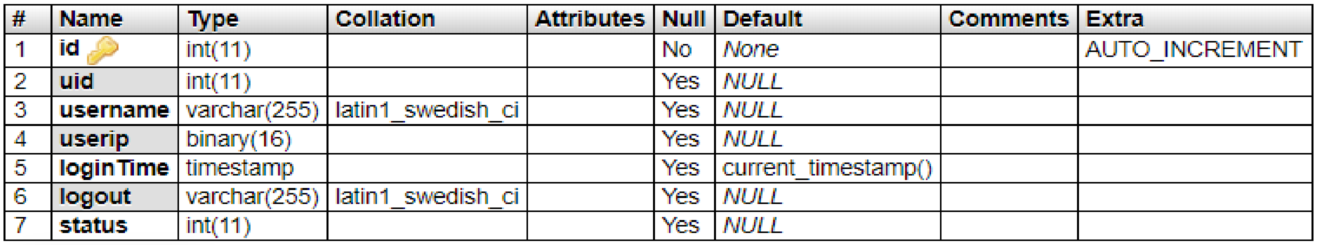
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	specilization	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
3	creationDate	timestamp			Yes	current_timestamp()		
4	updationDate	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()

doctors table Structure: This table store the logins and personal details of doctors.



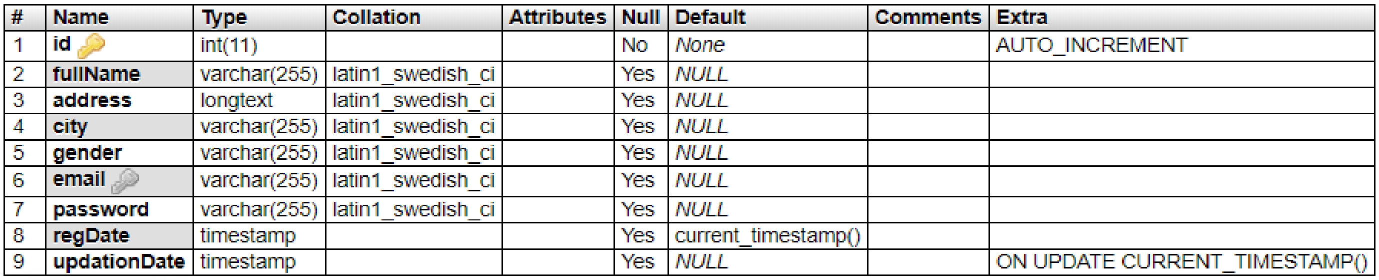
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	**specilization**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
3	**doctorName**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
4	**address**	longtext	latin1_swedish_ci		Yes	*NULL*		
5	**docFees**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
6	**contactno**	bigint(11)			Yes	*NULL*		
7	**docEmail**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
8	**password**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
9	**creationDate**	timestamp			Yes	current_timestamp()		
10	**updationDate**	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()

doctors log table Structure: This table store the doctor login and logout details.



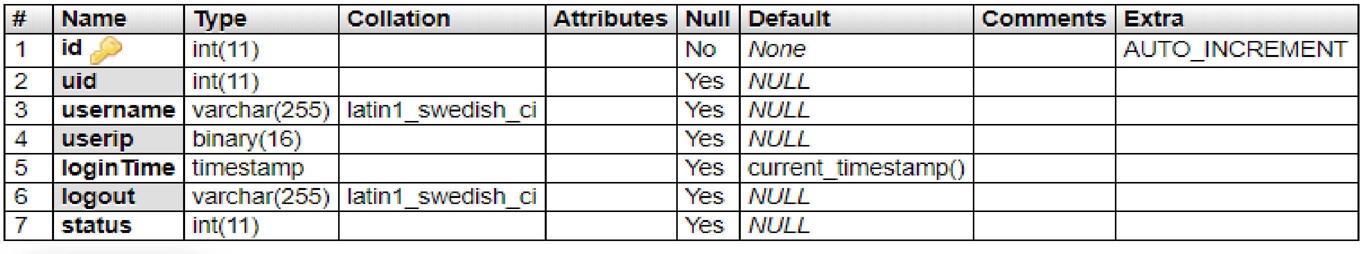
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	**uid**	int(11)			Yes	*NULL*		
3	**username**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
4	**userip**	binary(16)			Yes	*NULL*		
5	**loginTime**	timestamp			Yes	current_timestamp()		
6	**logout**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
7	**status**	int(11)			Yes	*NULL*		

users table Structure: This table store the user's login and personal details.



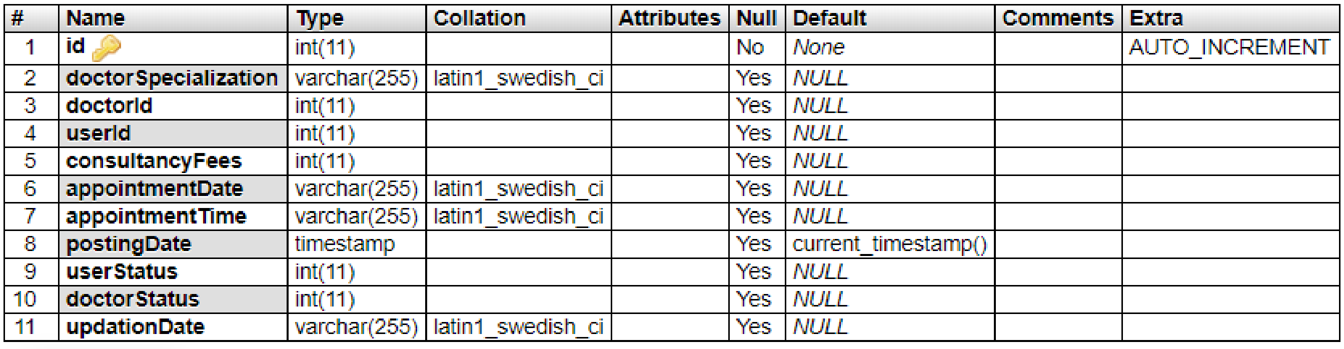
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	**fullName**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
3	**address**	longtext	latin1_swedish_ci		Yes	*NULL*		
4	**city**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
5	**gender**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
6	**email** 🔑	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
7	**password**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
8	**regDate**	timestamp			Yes	current_timestamp()		
9	**updationDate**	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()

user log table Structure: This table store the users login and personal details.



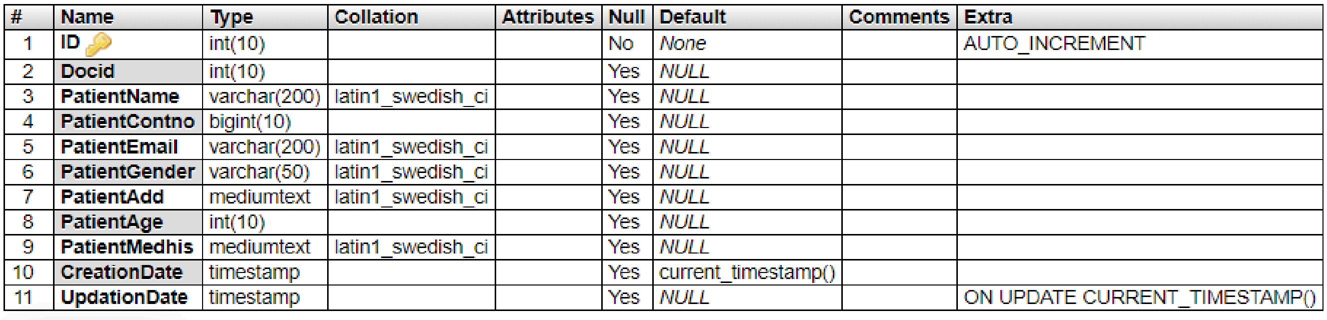
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	**uid**	int(11)			Yes	*NULL*		
3	**username**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
4	**userip**	binary(16)			Yes	*NULL*		
5	**loginTime**	timestamp			Yes	current_timestamp()		
6	**logout**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
7	**status**	int(11)			Yes	*NULL*		

appointment table Structure: This table store the user's appointment details.



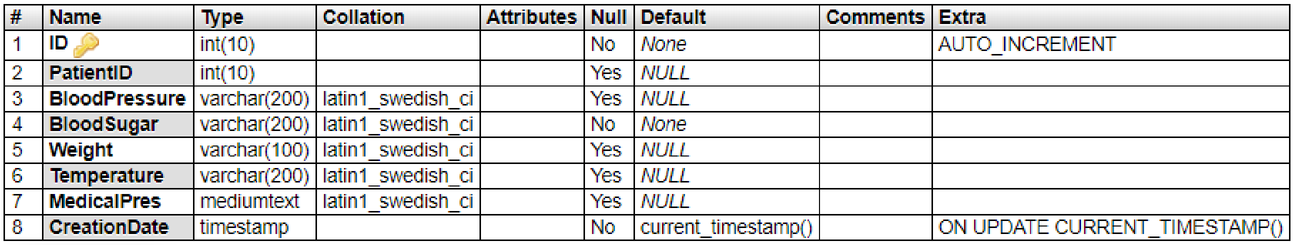
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	**doctorSpecialization**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
3	**doctorId**	int(11)			Yes	*NULL*		
4	**userId**	int(11)			Yes	*NULL*		
5	**consultancyFees**	int(11)			Yes	*NULL*		
6	**appointmentDate**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
7	**appointmentTime**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
8	**postingDate**	timestamp			Yes	current_timestamp()		
9	**userStatus**	int(11)			Yes	*NULL*		
10	**doctorStatus**	int(11)			Yes	*NULL*		
11	**updationDate**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		

tblpatient table Structure: This table store the patient details.



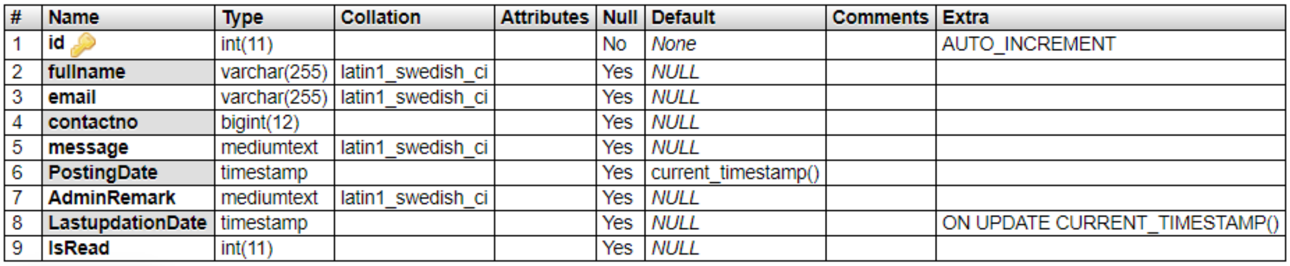
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**ID** 🔑	int(10)			No	*None*		AUTO_INCREMENT
2	**Docid**	int(10)			Yes	*NULL*		
3	**PatientName**	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
4	**PatientContno**	bigint(10)			Yes	*NULL*		
5	**PatientEmail**	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
6	**PatientGender**	varchar(50)	latin1_swedish_ci		Yes	*NULL*		
7	**PatientAdd**	mediumtext	latin1_swedish_ci		Yes	*NULL*		
8	**PatientAge**	int(10)			Yes	*NULL*		
9	**PatientMedhis**	mediumtext	latin1_swedish_ci		Yes	*NULL*		
10	**CreationDate**	timestamp			Yes	current_timestamp()		
11	**UpdationDate**	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()

tblmedicalhistory table Structure: This table store the patient medical history.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**ID** 🔑	int(10)			No	*None*		AUTO_INCREMENT
2	**PatientID**	int(10)			Yes	*NULL*		
3	**BloodPressure**	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
4	**BloodSugar**	varchar(200)	latin1_swedish_ci		No	*None*		
5	**Weight**	varchar(100)	latin1_swedish_ci		Yes	*NULL*		
6	**Temperature**	varchar(200)	latin1_swedish_ci		Yes	*NULL*		
7	**MedicalPres**	mediumtext	latin1_swedish_ci		Yes	*NULL*		
8	**CreationDate**	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

tblcontactus table Structure: This table store the contact us query details.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id** 🔑	int(11)			No	*None*		AUTO_INCREMENT
2	**fullname**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
3	**email**	varchar(255)	latin1_swedish_ci		Yes	*NULL*		
4	**contactno**	bigint(12)			Yes	*NULL*		
5	**message**	mediumtext	latin1_swedish_ci		Yes	*NULL*		
6	**PostingDate**	timestamp			Yes	current_timestamp()		
7	**AdminRemark**	mediumtext	latin1_swedish_ci		Yes	*NULL*		
8	**LastupdationDate**	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()
9	**IsRead**	int(11)			Yes	*NULL*		

CHAPTER 5

SYSTEM IMPLEMENTATION

5. IMPLEMENTATION:

5.1 Introduction:

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

5.2 Sample code:

index.html:

```
<! DOCTYPE HTML>
<html>
<head>
<title>Hospital Management System</title>
<link href="css/style.css" rel="stylesheet" type="text/css"  media="all" />
<link href='http://fonts.googleapis.com/css?family=Ropa+Sans' rel='stylesheet'
type='text/css'>
<link rel="stylesheet" href="css/responsiveslides.css">
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<scriptsrc="js/responsiveslides.min.js"></script>
<script> // You can also use "$(window).load(function() {"
                    $(function () {
                       // Slideshow 1
                       $("#slider1").responsiveSlides({
                    maxwidth: 1600,
                    speed: 600
                       });
                    });
               </script>
          </head>
          <body>
               <!--start-wrap-->
```

```
<!--start-header-->
<div class="header">
    <div class="wrap">
    <!--start-logo-->
    <div class="logo">
        <a href="index.html" style="font-size: 30px;">Hospital Management system</a>
    </div>
    <!--end-logo-->
    <!--start-top-nav-->
    <div class="top-nav">
        <ul>
            <li class="active"><a href="index.html">Home</a></li>


            <li><a href="contact.php">contact</a></li>
        </ul>
    </div>
    <div class="clear"></div>
    <!--end-top-nav-->
</div>
<!--end-header-->
```

```
            </div>
            <div class="clear"></div>
                <!--start-image-slider---->
                        <div class="image-slider">
                            <!-- Slideshow 1 -->
                        <ul class="rslides" id="slider1">
                        <li><imgsrc="images/slider-image1.jpg"
alt=""></li>
                        <li><imgsrc="images/slider-image2.jpg"
alt=""></li>
                        <li><imgsrc="images/slider-image1.jpg"
alt=""></li>
                        </ul>
                            <!-- Slideshow 2 -->
                        </div>
                        <!--End-image-slider---->
            <div class="clear"></div>
            <div class="content-grids">
                <div class="wrap">
                <div class="section group">
                <div class="listview_1_of_3 images_1_of_3">
                        <div class="listing listimg_1_of_2">
                            <img src="images/grid-img3.png">
                        </div>
                        <div class="text list_1_of_2">
                            <h3>Patients</h3>
                            <p>Register & Book Appointment</p>
                            <div class="button"><span><a
href="hms/user-login.php">Click Here</a></span></div>
                    </div>
                    </div>

                    <div class="listview_1_of_3 images_1_of_3">
                        <div class="listimg listimg_1_of_2">
                            <imgsrc="images/grid-img1.png">
                        </div>
                        <div class="text list_1_of_2">
                            <h3>Doctors Login</h3>

                            <div class="button"><span><a
href="hms/doctor/">Click Here</a></span></div>
                        </div>
                    </div>
<div class="listview_1_of_3 images_1_of_3">
                        <div class="listimg listimg_1_of_2">
                            <imgsrc="images/grid-img2.png">
                        </div>
                        <div class="text list_1_of_2">
                            <h3>Admin Login</h3>
<div class="button"><span><a href="hms/admin">Click Here</a></span></div>
                    </div>
                    </div>
                </div>
            </div></div>
            <div class="wrap">
            <div class="content-box">
            <div class="section group">
                    <div class="col_1_of_3 span_1_of_3 frist">
                    </div>
                    <div class="col_1_of_3 span_1_of_3 second">
                </div>
                    <div class="col_1_of_3 span_1_of_3 frist">
                </div>
                </div>
            </div>
            </div>
            <div class="clear"></div>
            <div class="footer">
                <div class="wrap">
                <div class="footer-left">
```

```
                              <ul>

                                  <li><a
href="index.html">Home</a></li>

                                  <li><a
href="contact.php">contact</a></li>

                              </ul>

                      </div>

                      <div class="clear"></div>

                  </div>

                  </div>

                  <!--end-wrap-->

          </body>

</html>
```

Doctor/index.php

```
<?php

session_start();

include("include/config.php");

error_reporting(0);

if(isset($_POST['submit']))

{

$ret=mysqli_query($con,"SELECT * FROM doctors WHERE docEmail='".$_POST['username']."'
and password='".md5($_POST['password'])."'");

$num=mysqli_fetch_array($ret);

if($num>0)
```

```
{

$extra="dashboard.php";

$_SESSION['dlogin']=$_POST['username'];

$_SESSION['id']=$num['id'];

$uip=$_SERVER['REMOTE_ADDR'];

$status=1;

$log=mysqli_query($con,"insert into doctorslog(uid,username,userip,status)
values('".$_SESSION['id']."','".$_SESSION['dlogin']."','$uip','$status')");

$host=$_SERVER['HTTP_HOST'];

$uri=rtrim(dirname($_SERVER['PHP_SELF']),'/\\');

header("location:http://$host$uri/$extra");

exit();

}

else

{

$host  = $_SERVER['HTTP_HOST'];

$_SESSION['dlogin']=$_POST['username'];

$uip=$_SERVER['REMOTE_ADDR'];

$status=0;

mysqli_query($con,"insert into doctorslog(username,userip,status)
values('".$_SESSION['dlogin']."','$uip','$status')");

$_SESSION['errmsg']="Invalid username or password";

$extra="index.php";

$uri  = rtrim(dirname($_SERVER['PHP_SELF']),'/\\');

header("location:http://$host$uri/$extra");

exit();
}

}

?>



<! DOCTYPE html>

<html lang="en">

        <head>

                <title>Doctor Login</title>



                <link
href="http://fonts.googleapis.com/css?family=Lato:300,400,400italic,600,700|Raleway:300,400,500,
600,700|Crete+Round:400italic" rel="stylesheet" type="text/css" />

                <link rel="stylesheet" href="vendor/bootstrap/css/bootstrap.min.css">

                <link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">

                <link rel="stylesheet" href="vendor/themify-icons/themify-icons.min.css">

                <link href="vendor/animate.css/animate.min.css" rel="stylesheet" media="screen">

                <link href="vendor/perfect-scrollbar/perfect-scrollbar.min.css" rel="stylesheet"
media="screen">

                <link href="vendor/switchery/switchery.min.css" rel="stylesheet" media="screen">

                <link rel="stylesheet" href="assets/css/styles.css">

                <link rel="stylesheet" href="assets/css/plugins.css">

                <link rel="stylesheet" href="assets/css/themes/theme-1.css" id="skin_color" />

        </head>

        <body class="login">

                <div class="row">
```

```
            <div class="main-login col-xs-10 col-xs-offset-1 col-sm-8 col-sm-offset-2
col-md-4 col-md-offset-4">
                <div class="logo margin-top-30">
                <a href="../../index.html">    <h2> HMS | Doctor Login</h2></a>
                </div>

                <div class="box-login">
                    <form class="form-login" method="post">
                        <fieldset>
                            <legend>
                                Sign in to your account
                            </legend>
                            <p>
                                Please enter your name and password to
log in.<br />
                                <span style="color:red;"><?php echo
$_SESSION['errmsg']; ?><?php echo $_SESSION['errmsg']="";?></span>
                            </p>
                            <div class="form-group">
                                <span class="input-icon">
                                    <input type="text"
class="form-control" name="username" placeholder="Username">
                                    <i class="fa fa-user"></i>
</span>
                            </div>
                            <div class="form-group form-actions">
                                <span class="input-icon">
```

```
                                    <input type="password"
class="form-control password" name="password" placeholder="Password">

                                    <i class="fa fa-lock"></i>

                                     </span>

                                     <a href="forgot-password.php">

                                    Forgot Password ?

                                </a>

                            </div>

                            <div class="form-actions">



                                <button type="submit" class="btn
btn-primary pull-right" name="submit">

                                    Login <i class="fa
fa-arrow-circle-right"></i>

                                </button>

                            </div>





                        </fieldset>

                    </form>



                    <div class="copyright">

                        &copy; <span class="current-year"></span><span
class="text-bold text-uppercase"> HMS</span>. <span>All rights reserved</span>

                    </div>



                </div>
```

```
				</div>
			</div>
			<script src="vendor/jquery/jquery.min.js"></script>
			<script src="vendor/bootstrap/js/bootstrap.min.js"></script>
			<script src="vendor/modernizr/modernizr.js"></script>
			<script src="vendor/jquery-cookie/jquery.cookie.js"></script>
			<script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
			<script src="vendor/switchery/switchery.min.js"></script>
			<script src="vendor/jquery-validation/jquery.validate.min.js"></script>

			<script src="assets/js/main.js"></script>

			<script src="assets/js/login.js"></script>
			<script>
				jQuery(document).ready(function() {
					Main.init();
					Login.init();
				});
			</script>

		</body>
		<!-- end: BODY -->
	</html>
```

Doctor/appointment-history.php

```
<?php
session_start();
error_reporting(0);
include('include/config.php');
include('include/checklogin.php');
check_login();
if(isset($_GET['cancel']))
                {
                        mysqli_query($con,"update appointment set doctorStatus='0' where id =
'".$_GET['id']."'");
            $_SESSION['msg']="Appointment canceled !!";
                }
?>
<!DOCTYPE html>
<html lang="en">
        <head>
                <title>Doctor | Appointment History</title>

                <link
href="http://fonts.googleapis.com/css?family=Lato:300,400,400italic,600,700|Raleway:300,400,500,
600,700|Crete+Round:400italic" rel="stylesheet" type="text/css" />
                <link rel="stylesheet" href="vendor/bootstrap/css/bootstrap.min.css">
                <link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
                <link rel="stylesheet" href="vendor/themify-icons/themify-icons.min.css">
                <link href="vendor/animate.css/animate.min.css" rel="stylesheet" media="screen">
                <link href="vendor/perfect-scrollbar/perfect-scrollbar.min.css" rel="stylesheet"
media="screen">
```

```
        <link href="vendor/switchery/switchery.min.css" rel="stylesheet" media="screen">

        <link href="vendor/bootstrap-touchspin/jquery.bootstrap-touchspin.min.css"
rel="stylesheet" media="screen">

        <link href="vendor/select2/select2.min.css" rel="stylesheet" media="screen">

        <link href="vendor/bootstrap-datepicker/bootstrap-datepicker3.standalone.min.css"
rel="stylesheet" media="screen">

        <link href="vendor/bootstrap-timepicker/bootstrap-timepicker.min.css"
rel="stylesheet" media="screen">

        <link rel="stylesheet" href="assets/css/styles.css">

        <link rel="stylesheet" href="assets/css/plugins.css">

        <link rel="stylesheet" href="assets/css/themes/theme-1.css" id="skin_color" />

    </head>

    <body>

        <div id="app">

<?php include('include/sidebar.php');?>

            <div class="app-content">



                    <?php include('include/header.php');?>

                <!-- end: TOP NAVBAR -->

                <div class="main-content" >

                    <div class="wrap-content container" id="container">

                        <!-- start: PAGE TITLE -->

                        <section id="page-title">

                            <div class="row">

                                <div class="col-sm-8">
```

```
                                    <h1 class="mainTitle">Doctor  |
Appointment History</h1>

                <div>
                                <ol class="breadcrumb">
                                    <li>
                                        <span>Doctor </span>
                                    </li>
                                    <li class="active">
                                        <span>Appointment
History</span>
                                    </li>
                                </ol>
                            </div>
                        </section>
                        <!-- end: PAGE TITLE -->
                        <!-- start: BASIC EXAMPLE -->
                        <div class="container-fluid container-fullw bg-white">



                                    <div class="row">
                                <div class="col-md-12">


                                    <p style="color: red;"><?php
echo htmlentities($_SESSION['msg']);?>
                                <?php echo
htmlentities($_SESSION['msg']="");?></p>
```

```
                                                            <table class="table table-hover"
id="sample-table-1">

                                                                  <thead>

                                                                        <tr>

                                                                              <th
class="center">#</th>

                                                                              <th
class="hidden-xs">Patient Name</th>

<th>Specialization</th>

<th>Consultancy Fee</th>

<th>Appointment Date / Time </th>

<th>Appointment Creation Date </th>

<th>Current Status</th>

<th>Action</th>


                                                                        </tr>

                                                                  </thead>

                                                                  <tbody>

<?php

$sql=mysqli_query($con,"select users. full Name as fame, appointment. *  from appointment join
users on users.id=appointment. user ID where appointment. doctor ID='".$_SESSION['id']."'");

$cnt=1;

while($row=mysqli_fetch_array($sql))
```

```
{
?>

                                                                <tr>
                                                                        <td
class="center"><?php echo $cnt;?>.</td>
                                                                        <td
class="hidden-xs"><?php echo $row['fname'];?></td>
                                                                        <td><?php
echo $row['doctor Specialization'];?></td>
                                                                        <td><?php
echo $row['consultancy Fees'];?></td>
                                                                        <td><?php
echo $row['appointment Date'];?> / <?php echo

$row['appointment Time'];?>
                                                                        </td>
                                                                        <td><?php
echo $row['posting Date'];?></td>
                                                                        <td>
<? php if(($row['userStatus']==1) && ($row['doctorStatus']==1))
{
        echo "Active";
}
if(($row['userStatus']==0) && ($row['doctorStatus']==1))
{
        echo "Cancel by Patient";
}
```

```
if(($row['userStatus']==1) && ($row['doctorStatus']==0))
{
    echo "Cancel by you";
}

                                                ?></td>
                                                <td >
                                                <div
class="visible-md visible-lg hidden-sm hidden-xs">
                            <?php if(($row['userStatus']==1) &&
($row['doctorStatus']==1))
{ ?>

    <a href="appointment-history.php? id=<?php echo $row['id']?>&cancel=update"
onClick="return confirm ('Are you sure you want to cancel this appointment ?')"class="btn
btn-transparent btn-xs tooltips" title="Cancel Appointment" tooltip-placement="top"
tooltip="Remove">Cancel</a>
    <? php } else {

        echo "Canceled";
        } ?>
                                                </div>
                                                </td>
```

```
                                            </tr>

                                            <? php
$cnt=$cnt+1;
                                             }?>


                                        </tbody>
                                    </table>
                                </div>
                            </div>
                                </div>

                        <!-- end: BASIC EXAMPLE -->
                        <!-- end: SELECT BOXES -->

                    </div>
                </div>
            </div>
            <! -- start: FOOTER -->
    <? php include ('include/footer. Php');?>
            <! -- end: FOOTER -->

            <! -- start: SETTINGS -->
    <? php include('include/setting. Php');?>
    <! -- end: SETTINGS -->
</div>
<! -- start: MAIN JAVASCRIPTS -->
<script src="vendor/jQuery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.min.js"></script>
<script src="vendor/modernizer/modernizr.js"></script>
<script src="vendor/jQuery-cookie/jquery.cookie.js"></script>
<script src="vendor/perfect-scrollbar/perfect-scrollbar.min.js"></script>
<script src="vendor/switchery/switchery.min.js"></script>
<! -- end: MAIN JAVASCRIPTS -->
<! -- start: JAVASCRIPTS REQUIRED FOR THIS PAGE ONLY -->
<script src="vendor/masked input/jquery.maskedinput.min.js"></script>
<script src="vendor/bootstrap-touchspin/jquery.bootstrap-touchspin.min.js"></script>
<script src="vendor/autosize/autosize.min.js"></script>
<script src="vendor/selectFx/classie.js"></script>
<script src="vendor/selectFx/selectFx.js"></script>
<script src="vendor/select2/select2.min.js"></script>
<script src="vendor/bootstrap-datepicker/bootstrap-datepicker.min.js"></script>
<script src="vendor/bootstrap-timepicker/bootstrap-timepicker.min.js"></script>
<! -- end: JAVASCRIPTS REQUIRED FOR THIS PAGE ONLY -->
<! -- start: CLIP-TWO JAVASCRIPTS -->
<script src="assets/js/main.js"></script>
<! -- start: JavaScript Event Handlers for this page -->
<script src="assets/js/form-elements.js"></script>
```

```
        <script>
            jQuery(document). ready (function () {
                Main.init();
                FormElements.init();
            });
        </script>
        <! -- end: JavaScript Event Handlers for this page -->
        <! -- end: CLIP-TWO JAVASCRIPTS -->
    </body>
</html>
```

CHAPTER 6

TESTING

6.1 INTRODUCTION TO SYSTEM TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTING:

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one steps up – software applications at the company level – interact without error.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

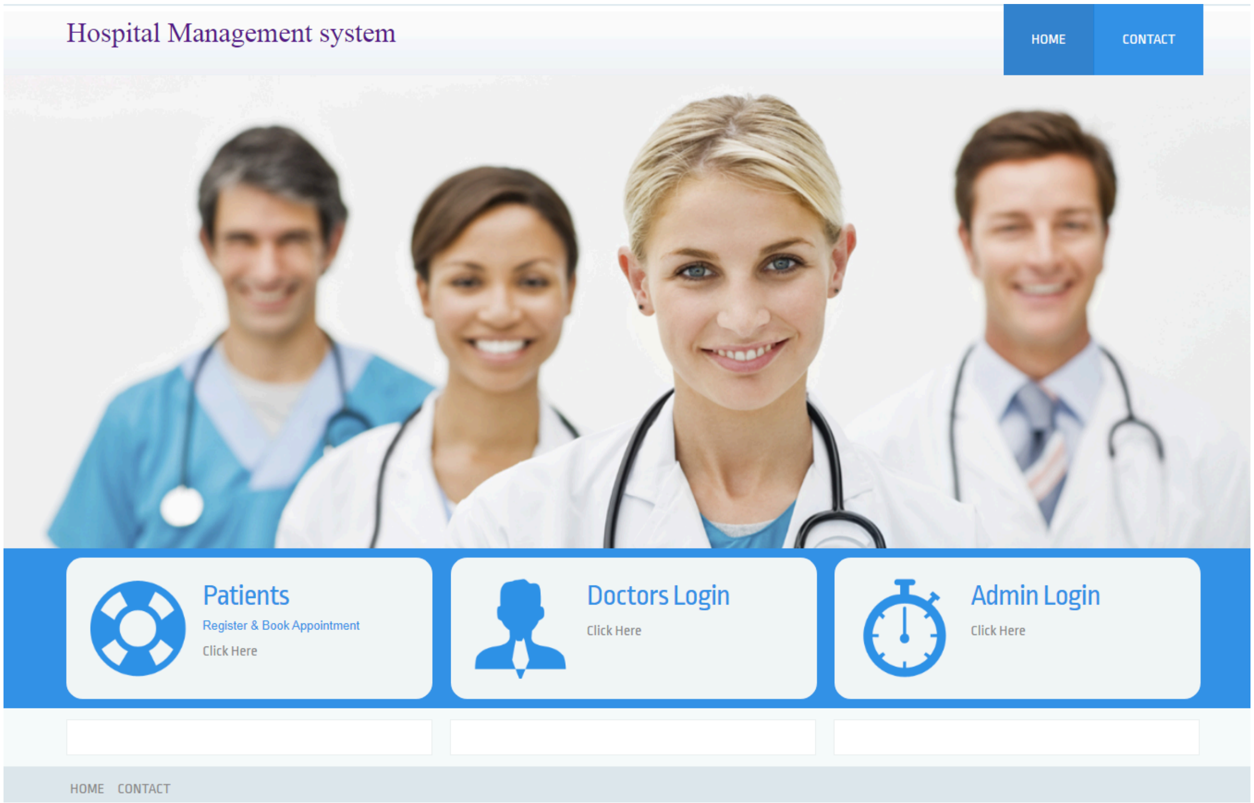
Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

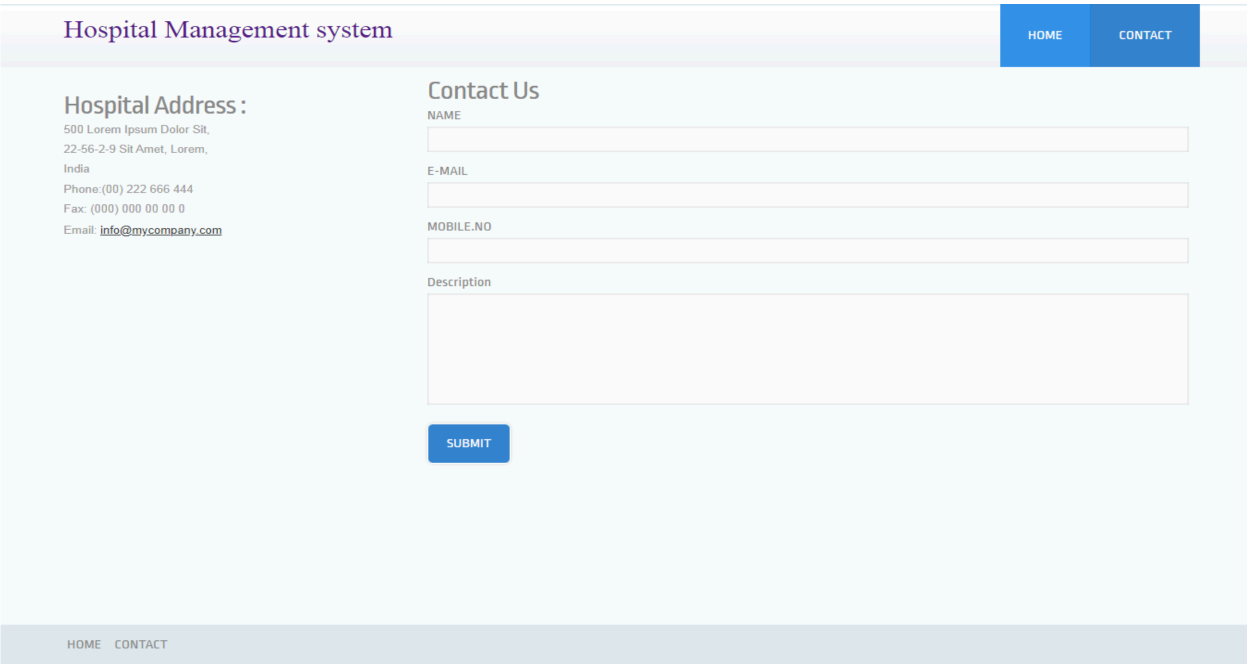
CHAPTER 7

SAMPLE SCREENSHOTS

Home Page



Contact Us



Hospital Address :

500 Lorem Ipsum Dolor Sit,
22-56-2-9 Sit Amet, Lorem,
India
Phone:(00) 222 666 444
Fax: (000) 000 00 00 0
Email: info@mycompany.com

Contact Us

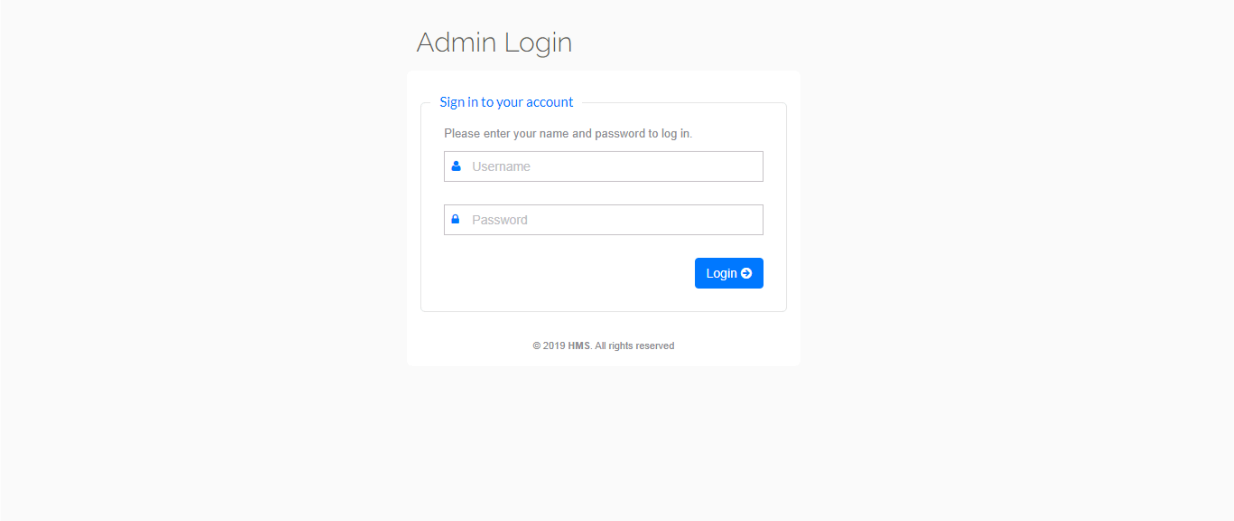
NAME

E-MAIL

MOBILE.NO

Description

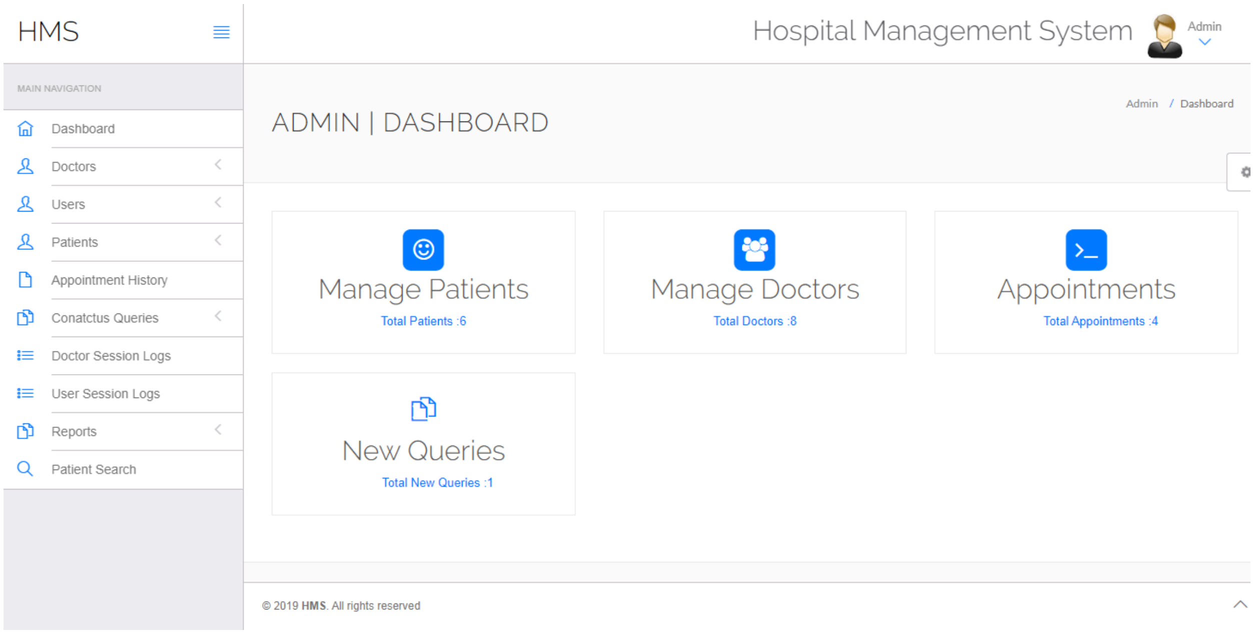
Admin Login



Please enter your name and password to log in.

© 2019 HMS. All rights reserved

Admin Dashboard



ADMIN | DASHBOARD

Manage Patients
Total Patients :6

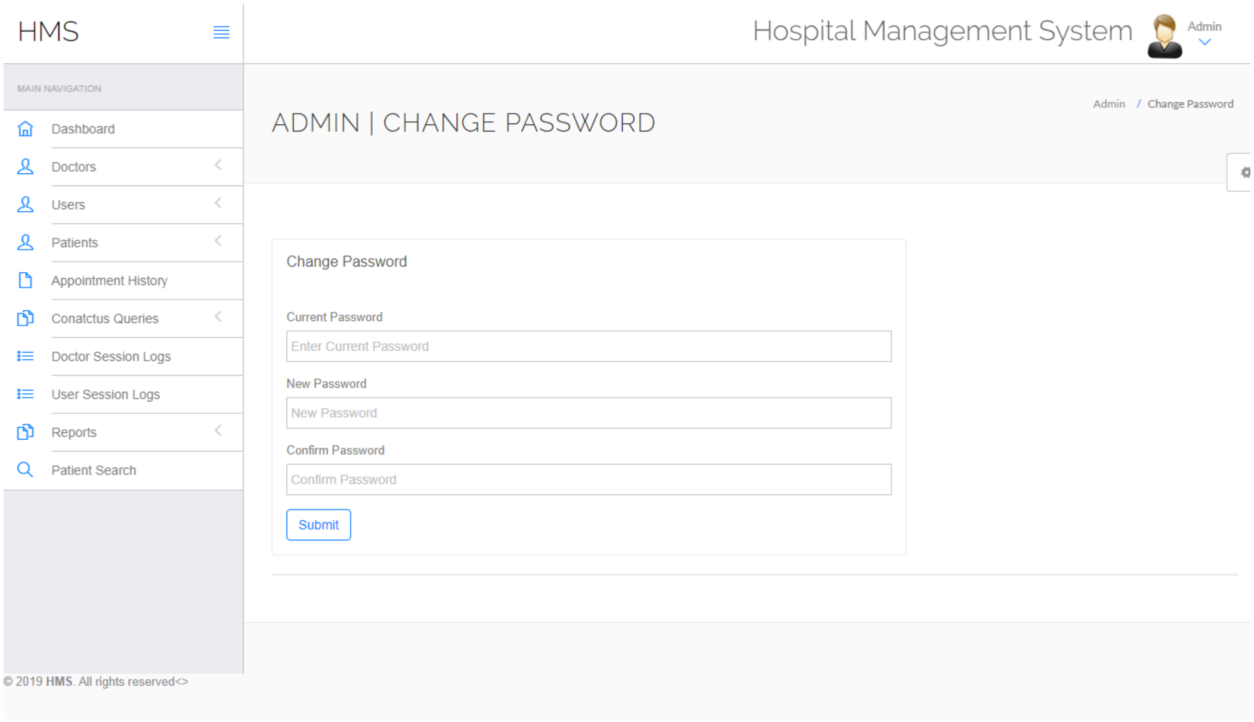
Manage Doctors
Total Doctors :8

Appointments
Total Appointments :4

New Queries
Total New Queries :1

© 2019 **HMS**. All rights reserved

Change Password



ADMIN | CHANGE PASSWORD

Change Password

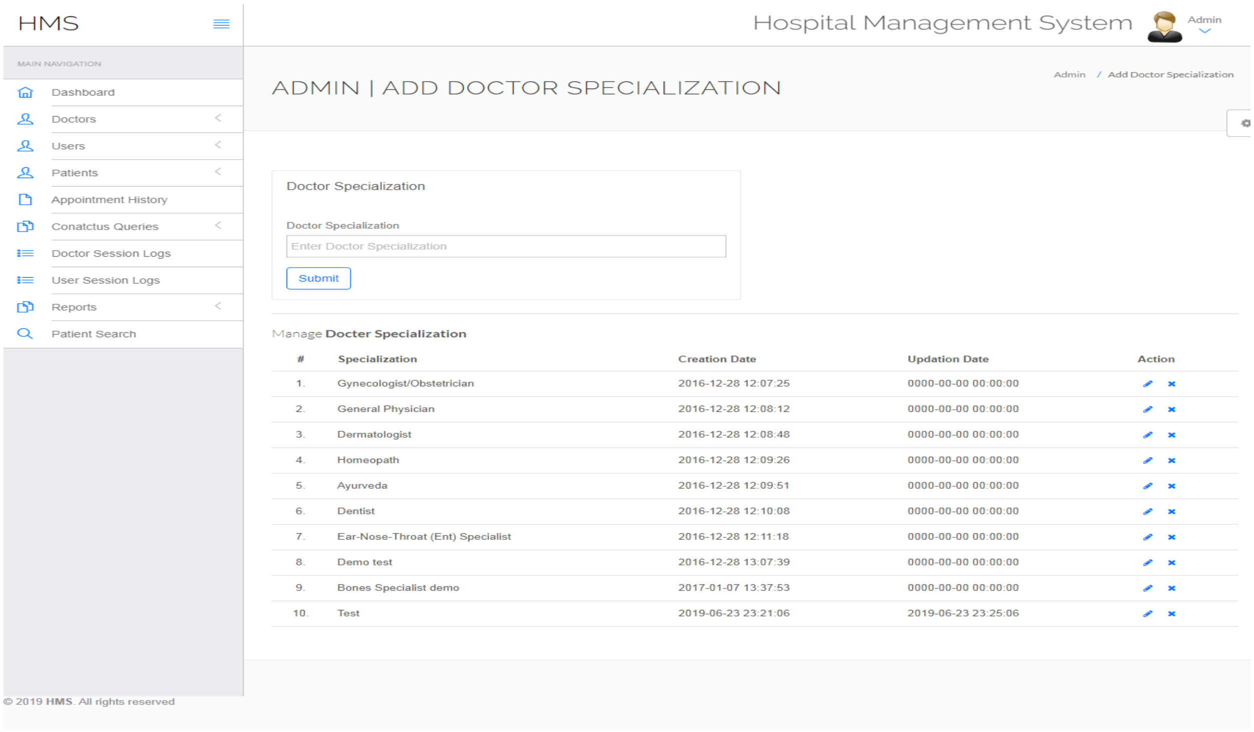
Current Password: Enter Current Password

New Password: New Password

Confirm Password: Confirm Password

© 2019 HMS. All rights reserved<>

Add Doctor Specialization



ADMIN | ADD DOCTOR SPECIALIZATION

Doctor Specialization

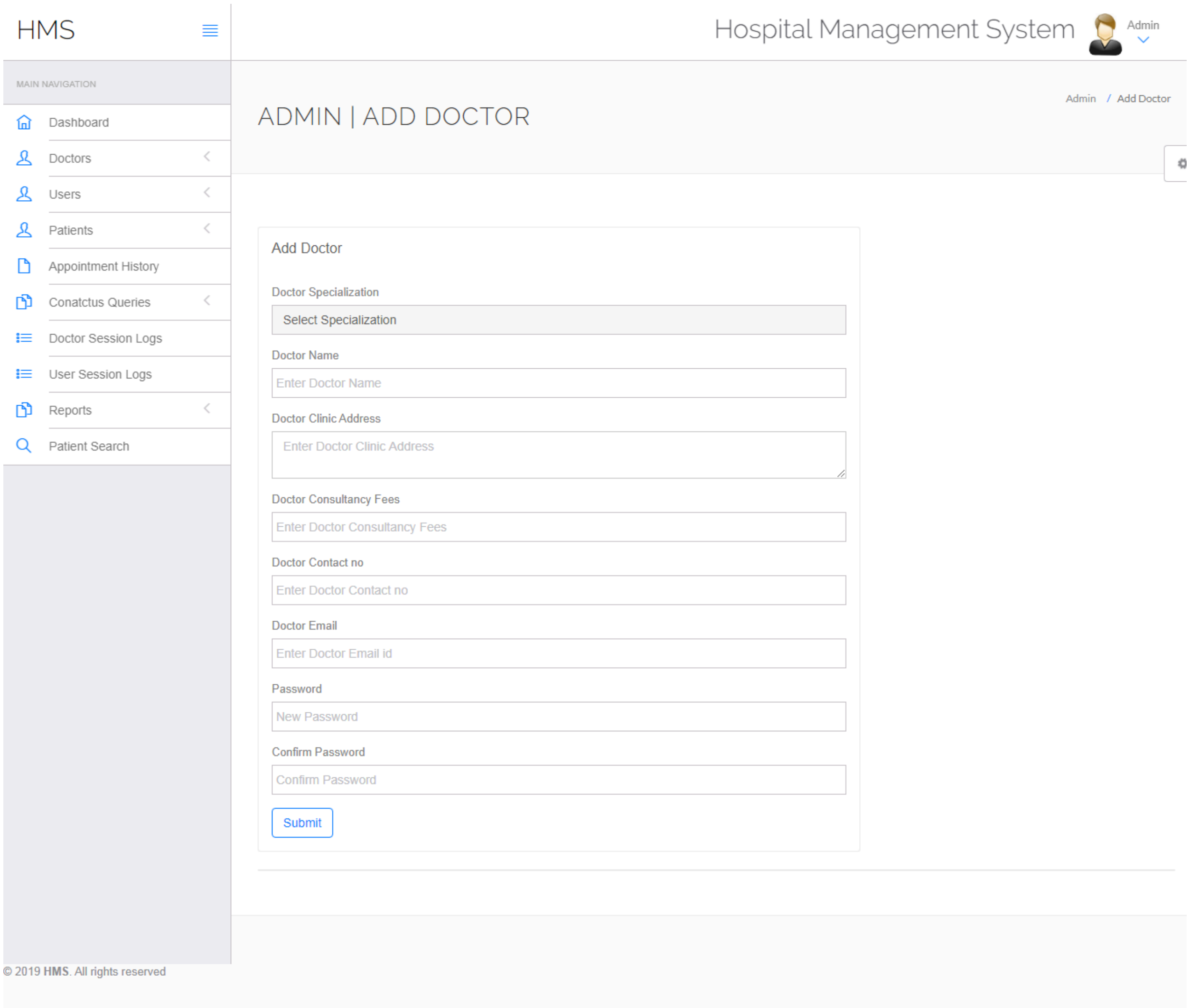
Doctor Specialization: Enter Doctor Specialization

Manage **Docter Specialization**

#	Specialization	Creation Date	Updation Date	Action
1.	Gynecologist/Obstetrician	2016-12-28 12:07:25	0000-00-00 00:00:00	✎ ✖
2.	General Physician	2016-12-28 12:08:12	0000-00-00 00:00:00	✎ ✖
3.	Dermatologist	2016-12-28 12:08:48	0000-00-00 00:00:00	✎ ✖
4.	Homeopath	2016-12-28 12:09:26	0000-00-00 00:00:00	✎ ✖
5.	Ayurveda	2016-12-28 12:09:51	0000-00-00 00:00:00	✎ ✖
6.	Dentist	2016-12-28 12:10:08	0000-00-00 00:00:00	✎ ✖
7.	Ear-Nose-Throat (Ent) Specialist	2016-12-28 12:11:18	0000-00-00 00:00:00	✎ ✖
8.	Demo test	2016-12-28 13:07:39	0000-00-00 00:00:00	✎ ✖
9.	Bones Specialist demo	2017-01-07 13:37:53	0000-00-00 00:00:00	✎ ✖
10.	Test	2019-06-23 23:21:06	2019-06-23 23:25:06	✎ ✖

© 2019 HMS. All rights reserved

Add Doctor



ADMIN | ADD DOCTOR

Add Doctor

Doctor Specialization

Select Specialization

Doctor Name

Enter Doctor Name

Doctor Clinic Address

Enter Doctor Clinic Address

Doctor Consultancy Fees

Enter Doctor Consultancy Fees

Doctor Contact no

Enter Doctor Contact no

Doctor Email

Enter Doctor Email id

Password

New Password

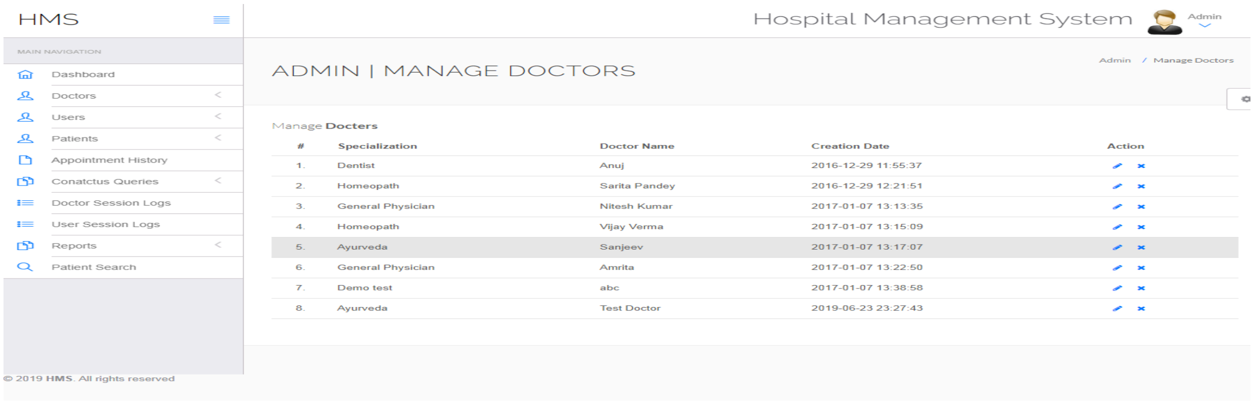
Confirm Password

Confirm Password

Submit

© 2019 **HMS**. All rights reserved

Manage Hospital



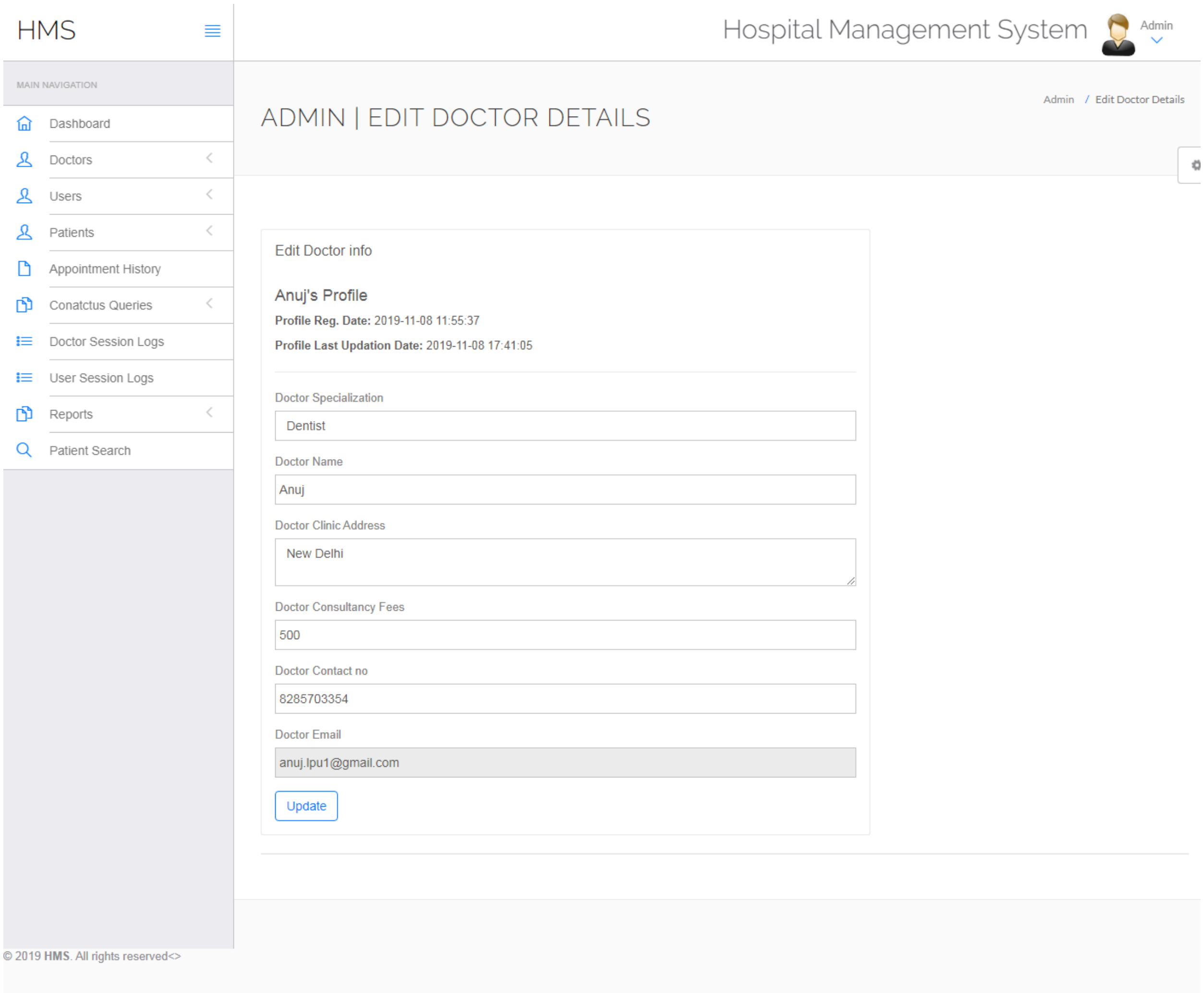
ADMIN | MANAGE DOCTORS

Manage **Docters**

#	Specialization	Doctor Name	Creation Date	Action
1.	Dentist	Anuj	2016-12-29 11:55:37	✎ ✖
2.	Homeopath	Sarita Pandey	2016-12-29 12:21:51	✎ ✖
3.	General Physician	Nitesh Kumar	2017-01-07 13:13:35	✎ ✖
4.	Homeopath	Vijay Verma	2017-01-07 13:15:09	✎ ✖
5.	Ayurveda	Sanjeev	2017-01-07 13:17:07	✎ ✖
6.	General Physician	Amrita	2017-01-07 13:22:50	✎ ✖
7.	Demo test	abc	2017-01-07 13:38:58	✎ ✖
8.	Ayurveda	Test Doctor	2019-06-23 23:27:43	✎ ✖

© 2019 **HMS**. All rights reserved

Doctor Info



ADMIN | EDIT DOCTOR DETAILS

Edit Doctor info

Anuj's Profile

Profile Reg. Date: 2019-11-08 11:55:37

Profile Last Updation Date: 2019-11-08 17:41:05

Doctor Specialization: Dentist

Doctor Name: Anuj

Doctor Clinic Address: New Delhi

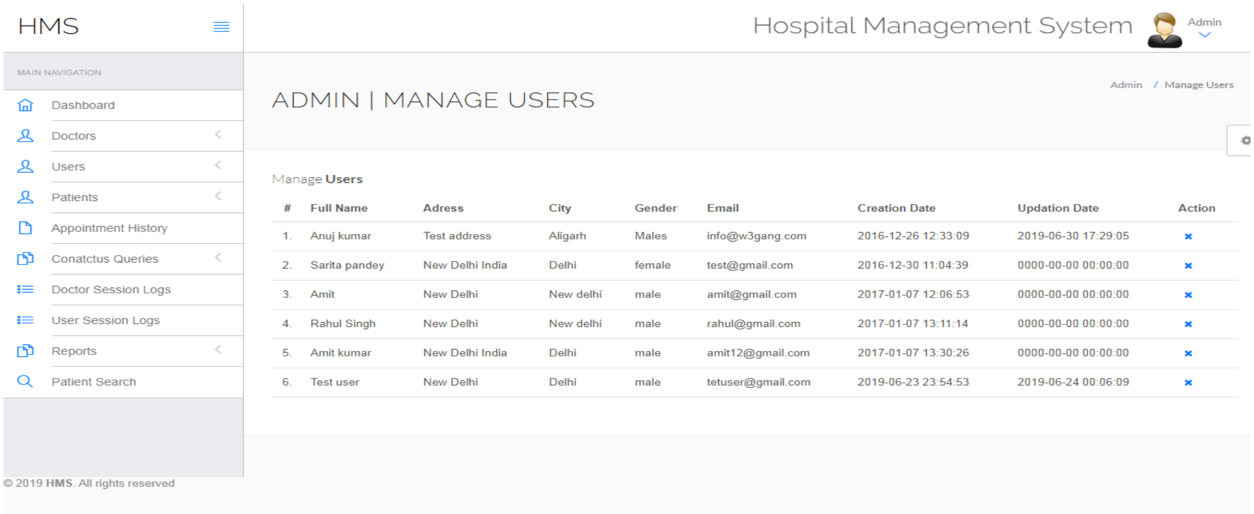
Doctor Consultancy Fees: 500

Doctor Contact no: 8285703354

Doctor Email: anuj.lpu1@gmail.com

© 2019 **HMS**. All rights reserved<>

Manage Users



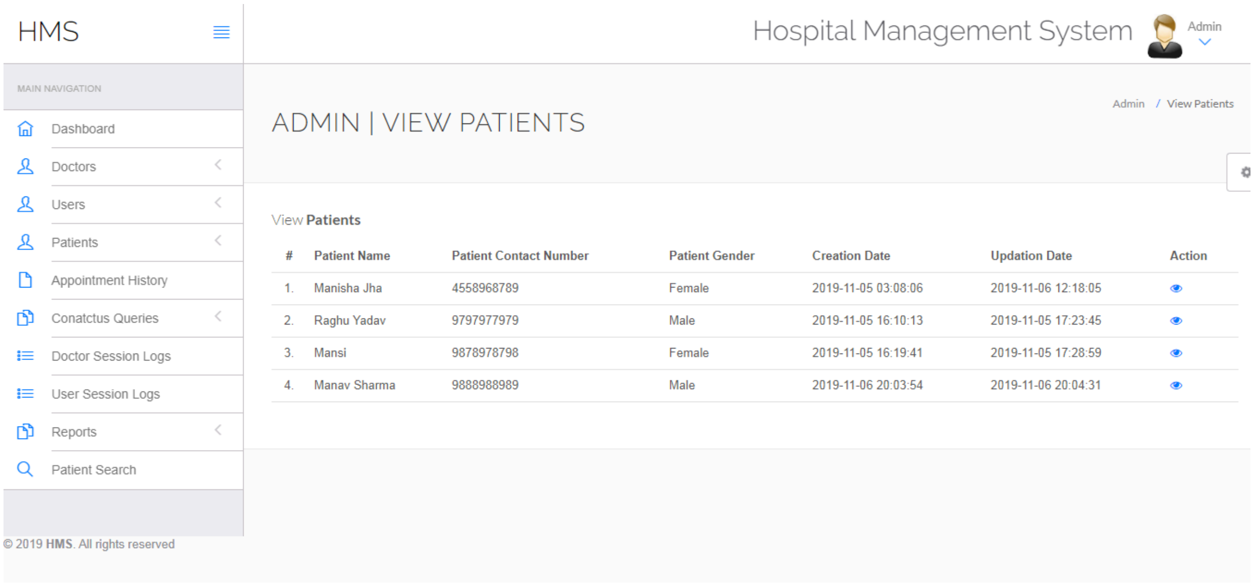
ADMIN | MANAGE USERS

Manage **Users**

#	Full Name	Adress	City	Gender	Email	Creation Date	Updation Date	Action
1.	Anuj kumar	Test address	Aligarh	Males	info@w3gang.com	2016-12-26 12:33:09	2019-06-30 17:29:05	✖
2.	Sarita pandey	New Delhi India	Delhi	female	test@gmail.com	2016-12-30 11:04:39	0000-00-00 00:00:00	✖
3.	Amit	New Delhi	New delhi	male	amit@gmail.com	2017-01-07 12:06:53	0000-00-00 00:00:00	✖
4.	Rahul Singh	New Delhi	New delhi	male	rahul@gmail.com	2017-01-07 13:11:14	0000-00-00 00:00:00	✖
5.	Amit kumar	New Delhi India	Delhi	male	amit12@gmail.com	2017-01-07 13:30:26	0000-00-00 00:00:00	✖
6.	Test user	New Delhi	Delhi	male	tetuser@gmail.com	2019-06-23 23:54:53	2019-06-24 00:06:09	✖

© 2019 **HMS**. All rights reserved

View Patient

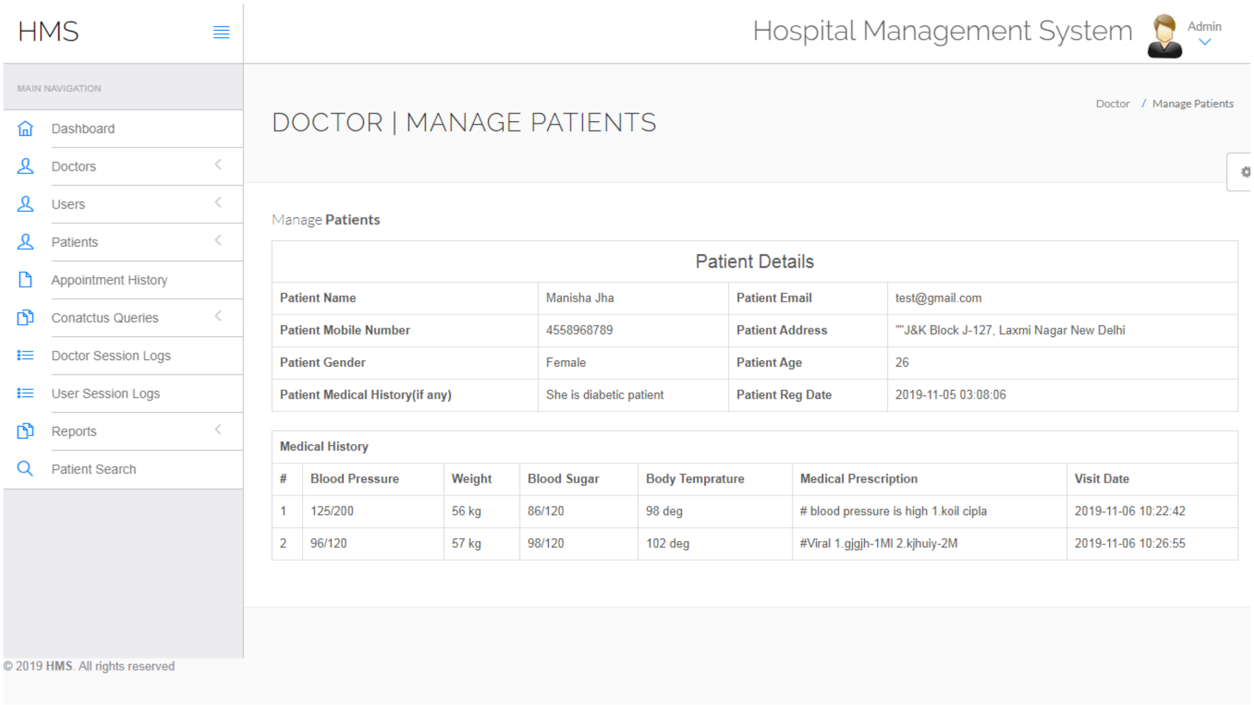


ADMIN | VIEW PATIENTS

View **Patients**

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Manisha Jha	4558968789	Female	2019-11-05 03:08:06	2019-11-06 12:18:05	👁
2.	Raghu Yadav	9797977979	Male	2019-11-05 16:10:13	2019-11-05 17:23:45	👁
3.	Mansi	9878978798	Female	2019-11-05 16:19:41	2019-11-05 17:28:59	👁
4.	Manav Sharma	9888988989	Male	2019-11-06 20:03:54	2019-11-06 20:04:31	👁

© 2019 **HMS**. All rights reserved



DOCTOR | MANAGE PATIENTS

Manage **Patients**

Patient Details

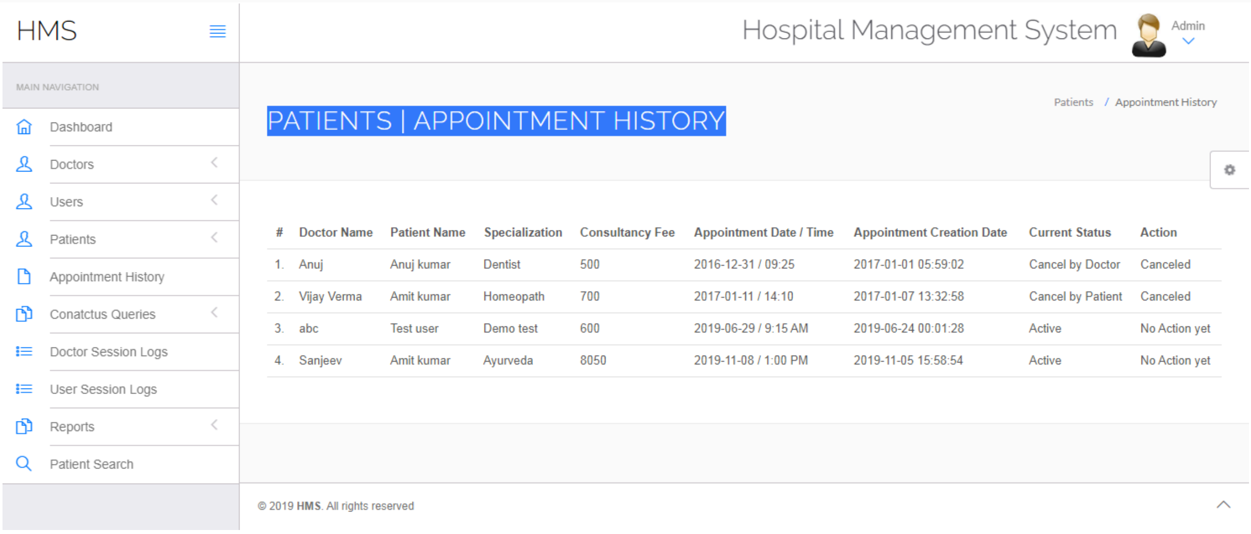
Patient Name	Manisha Jha	**Patient Email**	test@gmail.com
Patient Mobile Number	4558968789	**Patient Address**	""J&K Block J-127, Laxmi Nagar New Delhi
Patient Gender	Female	**Patient Age**	26
Patient Medical History(if any)	She is diabetic patient	**Patient Reg Date**	2019-11-05 03:08:06

Medical History

#	Blood Pressure	Weight	Blood Sugar	Body Temprature	Medical Prescription	Visit Date
1	125/200	56 kg	86/120	98 deg	# blood pressure is high 1.koil cipla	2019-11-06 10:22:42
2	96/120	57 kg	98/120	102 deg	#Viral 1.gjgjh-1Ml 2.kjhuiy-2M	2019-11-06 10:26:55

© 2019 **HMS**. All rights reserved

PATIENTS APPOINTMENT HISTORY

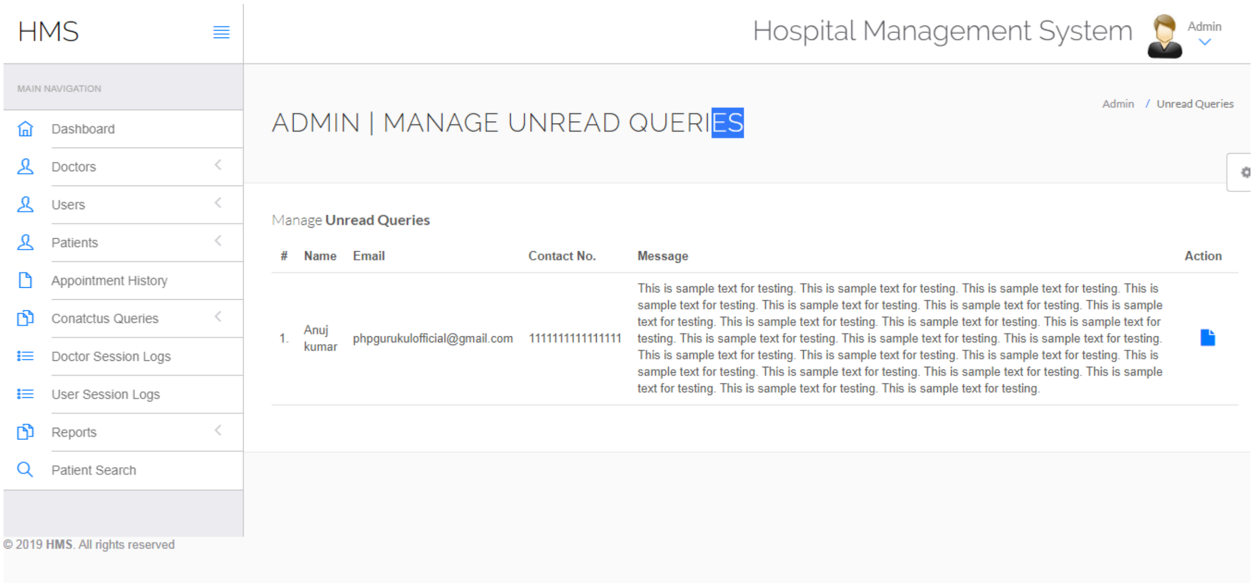


PATIENTS | APPOINTMENT HISTORY

#	Doctor Name	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Anuj	Anuj kumar	Dentist	500	2016-12-31 / 09:25	2017-01-01 05:59:02	Cancel by Doctor	Canceled
2.	Vijay Verma	Amit kumar	Homeopath	700	2017-01-11 / 14:10	2017-01-07 13:32:58	Cancel by Patient	Canceled
3.	abc	Test user	Demo test	600	2019-06-29 / 9:15 AM	2019-06-24 00:01:28	Active	No Action yet
4.	Sanjeev	Amit kumar	Ayurveda	8050	2019-11-08 / 1:00 PM	2019-11-05 15:58:54	Active	No Action yet

© 2019 **HMS**. All rights reserved

Manage Unread Queries



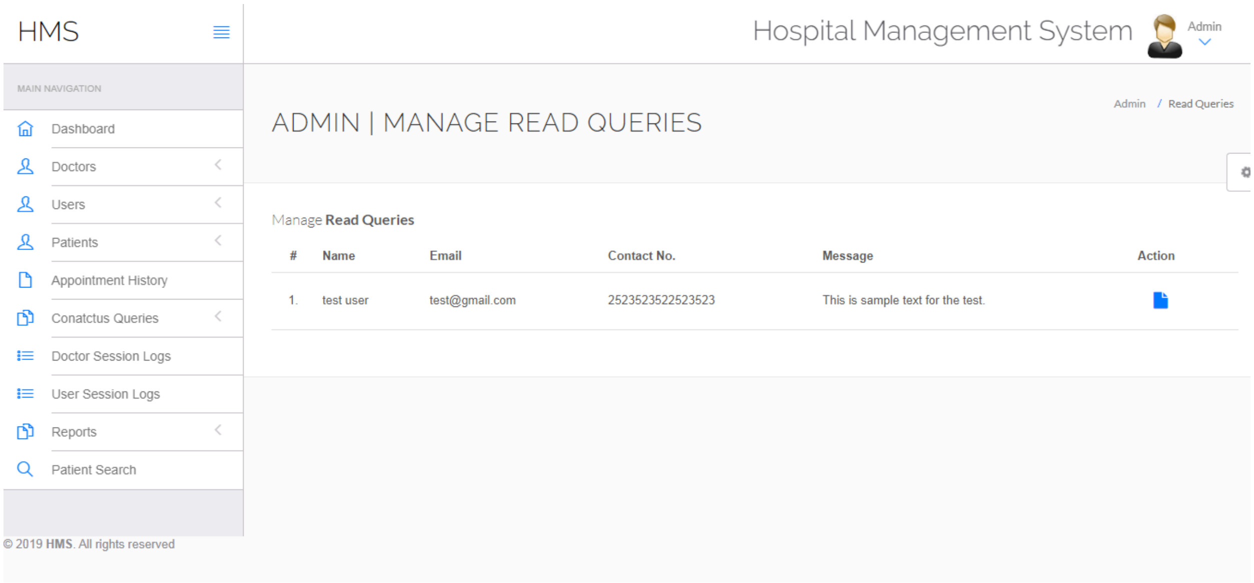
ADMIN | MANAGE UNREAD QUERIES

Manage **Unread Queries**

#	Name	Email	Contact No.	Message	Action
1.	Anuj kumar	phpgurukulofficial@gmail.com	111111111111111	This is sample text for testing. This is sample text for testing.	

© 2019 **HMS**. All rights reserved

Manage Read Queries



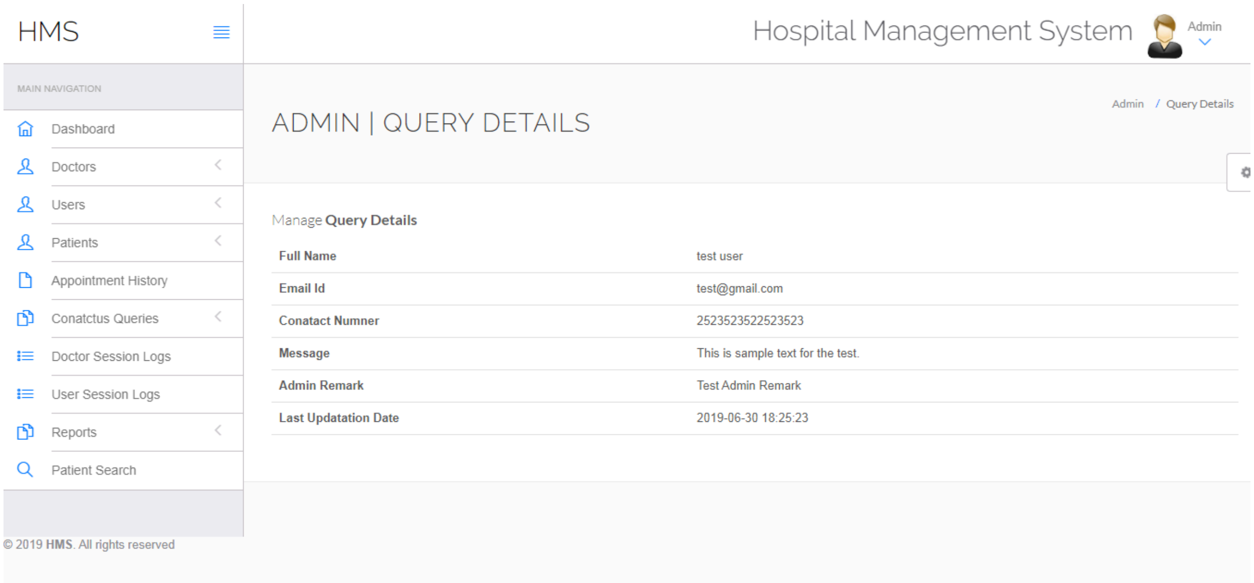
ADMIN | MANAGE READ QUERIES

Manage **Read Queries**

#	Name	Email	Contact No.	Message	Action
1.	test user	test@gmail.com	2523523522523523	This is sample text for the test.	📄

© 2019 **HMS**. All rights reserved

Query Detail



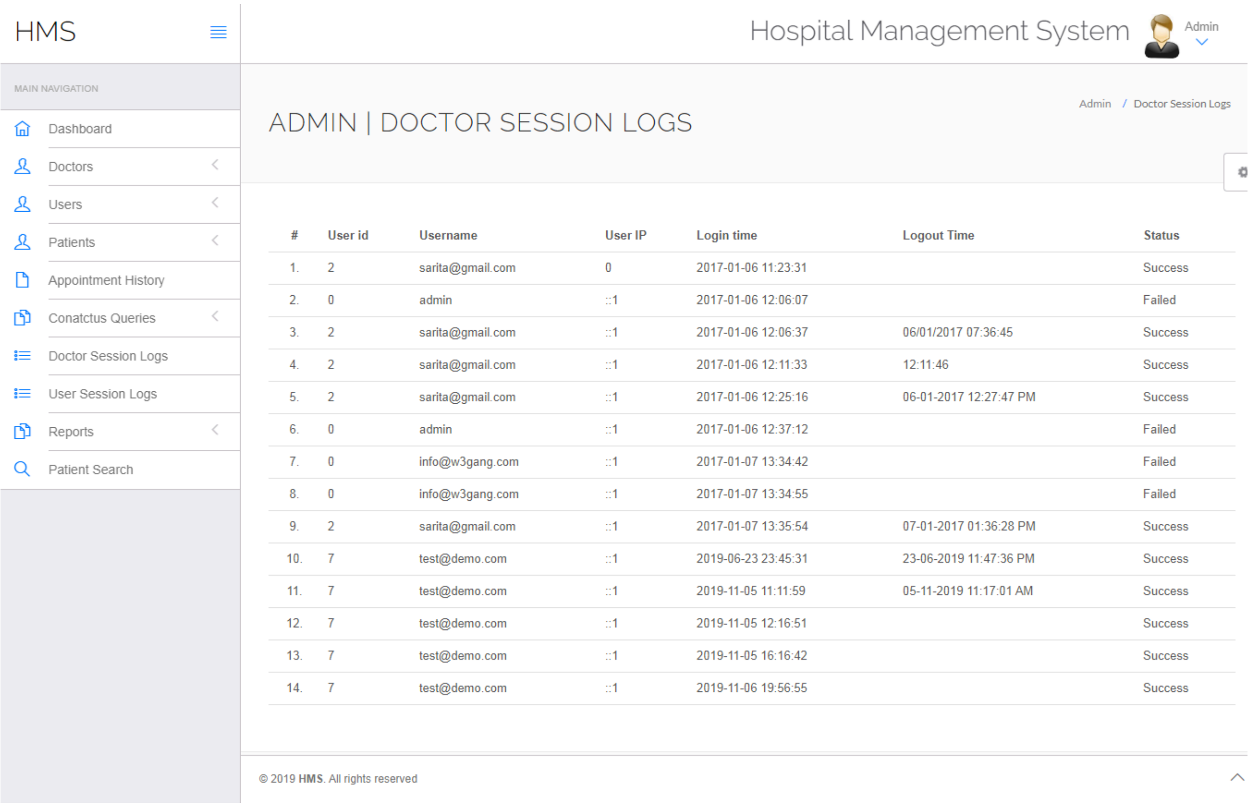
ADMIN | QUERY DETAILS

Manage **Query Details**

Full Name	test user
Email Id	test@gmail.com
Conatact Numner	2523523522523523
Message	This is sample text for the test.
Admin Remark	Test Admin Remark
Last Updatation Date	2019-06-30 18:25:23

© 2019 **HMS**. All rights reserved

Doctor Logs

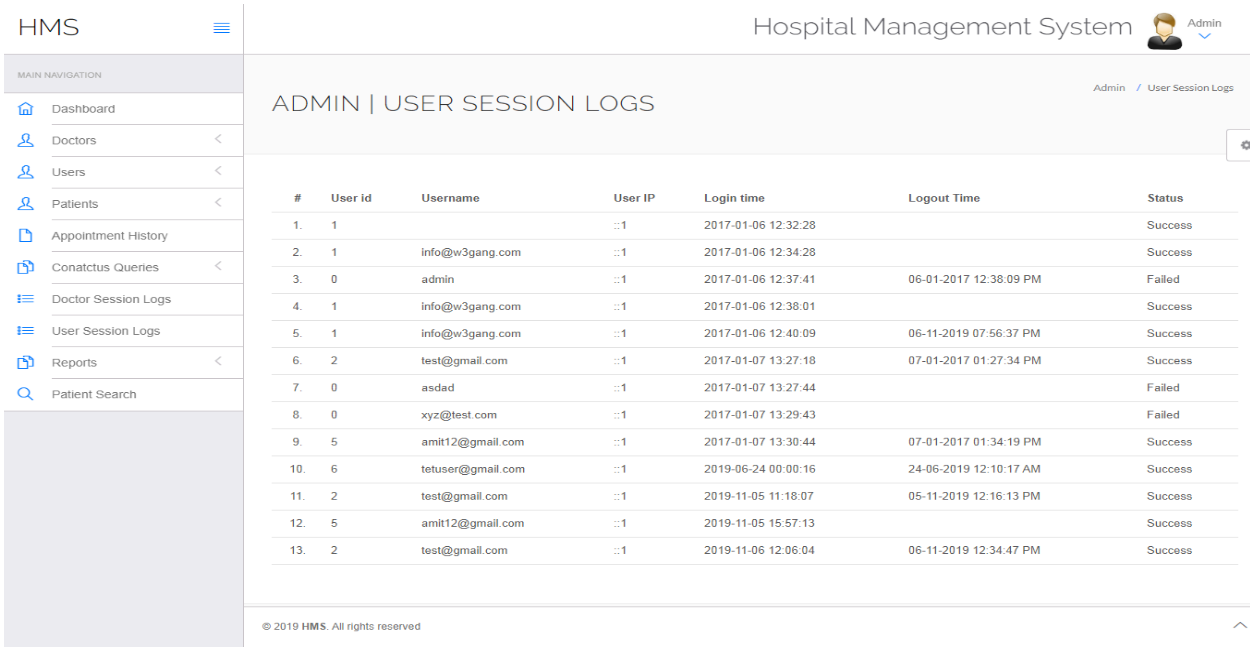


ADMIN | DOCTOR SESSION LOGS

#	User id	Username	User IP	Login time	Logout Time	Status
1.	2	sarita@gmail.com	0	2017-01-06 11:23:31		Success
2.	0	admin	::1	2017-01-06 12:06:07		Failed
3.	2	sarita@gmail.com	::1	2017-01-06 12:06:37	06/01/2017 07:36:45	Success
4.	2	sarita@gmail.com	::1	2017-01-06 12:11:33	12:11:46	Success
5.	2	sarita@gmail.com	::1	2017-01-06 12:25:16	06-01-2017 12:27:47 PM	Success
6.	0	admin	::1	2017-01-06 12:37:12		Failed
7.	0	info@w3gang.com	::1	2017-01-07 13:34:42		Failed
8.	0	info@w3gang.com	::1	2017-01-07 13:34:55		Failed
9.	2	sarita@gmail.com	::1	2017-01-07 13:35:54	07-01-2017 01:36:28 PM	Success
10.	7	test@demo.com	::1	2019-06-23 23:45:31	23-06-2019 11:47:36 PM	Success
11.	7	test@demo.com	::1	2019-11-05 11:11:59	05-11-2019 11:17:01 AM	Success
12.	7	test@demo.com	::1	2019-11-05 12:16:51		Success
13.	7	test@demo.com	::1	2019-11-05 16:16:42		Success
14.	7	test@demo.com	::1	2019-11-06 19:56:55		Success

© 2019 **HMS**. All rights reserved

User Logs

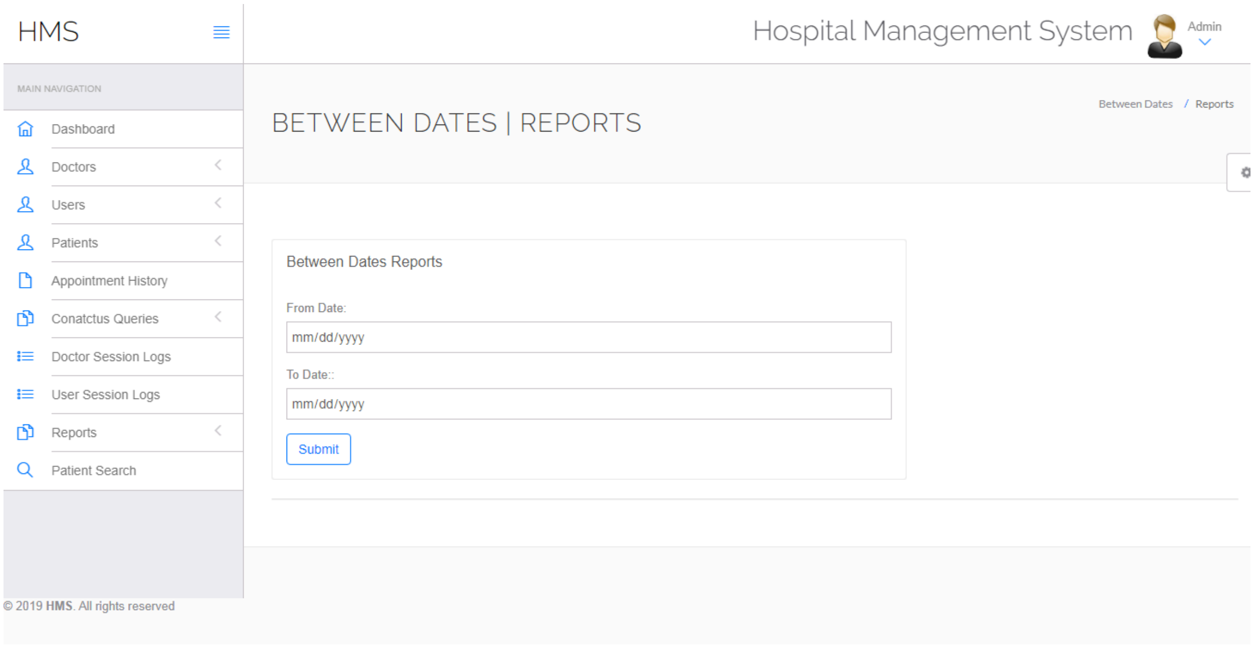


ADMIN | USER SESSION LOGS

#	User id	Username	User IP	Login time	Logout Time	Status
1.	1		::1	2017-01-06 12:32:28		Success
2.	1	info@w3gang.com	::1	2017-01-06 12:34:28		Success
3.	0	admin	::1	2017-01-06 12:37:41	06-01-2017 12:38:09 PM	Failed
4.	1	info@w3gang.com	::1	2017-01-06 12:38:01		Success
5.	1	info@w3gang.com	::1	2017-01-06 12:40:09	06-11-2019 07:56:37 PM	Success
6.	2	test@gmail.com	::1	2017-01-07 13:27:18	07-01-2017 01:27:34 PM	Success
7.	0	asdad	::1	2017-01-07 13:27:44		Failed
8.	0	xyz@test.com	::1	2017-01-07 13:29:43		Failed
9.	5	amit12@gmail.com	::1	2017-01-07 13:30:44	07-01-2017 01:34:19 PM	Success
10.	6	tetuser@gmail.com	::1	2019-06-24 00:00:16	24-06-2019 12:10:17 AM	Success
11.	2	test@gmail.com	::1	2019-11-05 11:18:07	05-11-2019 12:16:13 PM	Success
12.	5	amit12@gmail.com	::1	2019-11-05 15:57:13		Success
13.	2	test@gmail.com	::1	2019-11-06 12:06:04	06-11-2019 12:34:47 PM	Success

© 2019 **HMS**. All rights reserved

Reports



BETWEEN DATES | REPORTS

Between Dates Reports

From Date:

mm/dd/yyyy

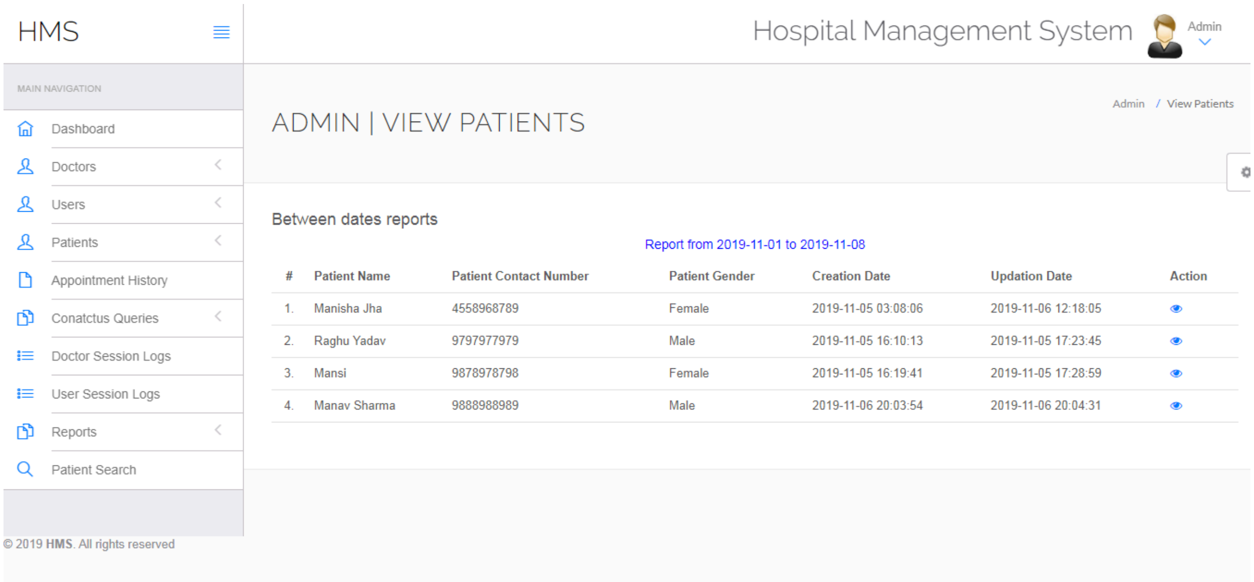
To Date::

mm/dd/yyyy

Submit

© 2019 **HMS**. All rights reserved

Between Dates Reports



ADMIN | VIEW PATIENTS

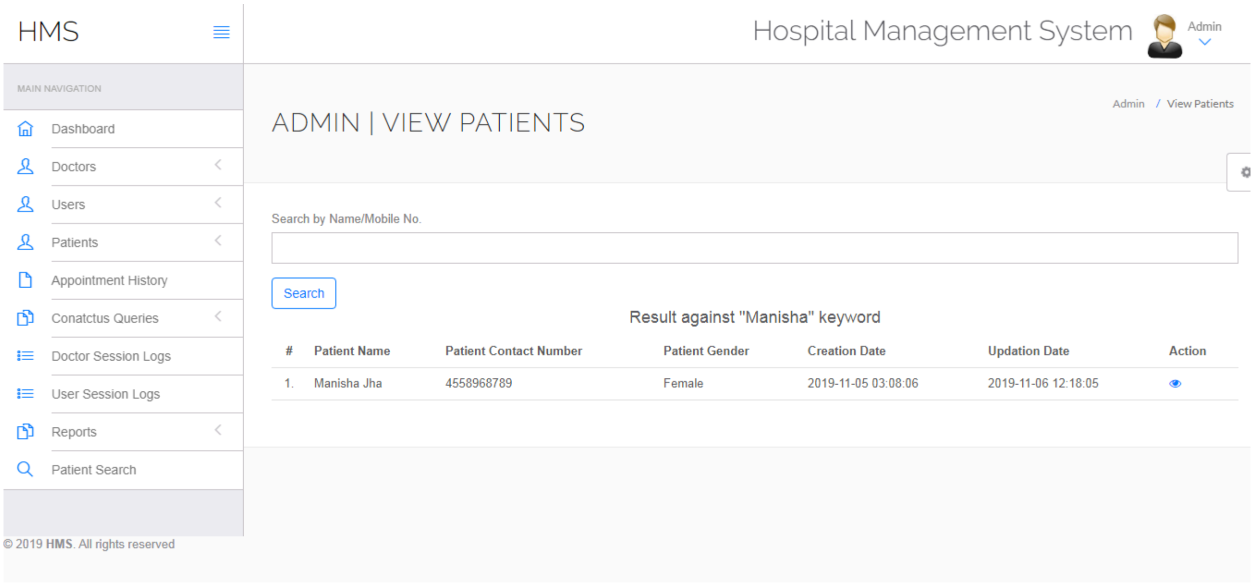
Between dates reports

Report from 2019-11-01 to 2019-11-08

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Manisha Jha	4558968789	Female	2019-11-05 03:08:06	2019-11-06 12:18:05	👁
2.	Raghu Yadav	9797977979	Male	2019-11-05 16:10:13	2019-11-05 17:23:45	👁
3.	Mansi	9878978798	Female	2019-11-05 16:19:41	2019-11-05 17:28:59	👁
4.	Manav Sharma	9888988989	Male	2019-11-06 20:03:54	2019-11-06 20:04:31	👁

© 2019 **HMS**. All rights reserved

Patient Search



ADMIN | VIEW PATIENTS

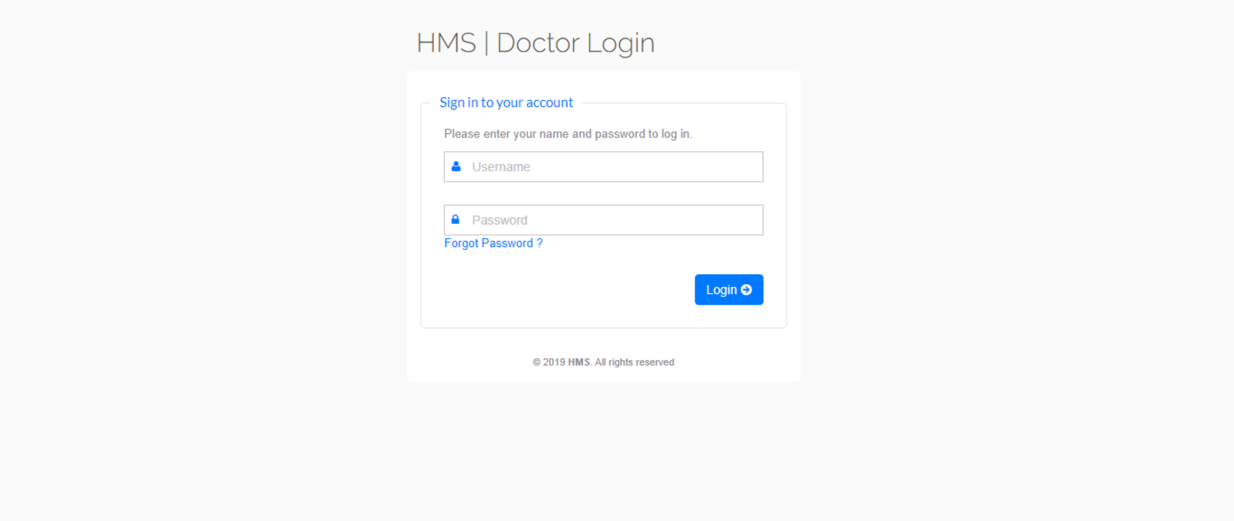
Search by Name/Mobile No.

Result against "Manisha" keyword

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Manisha Jha	4558968789	Female	2019-11-05 03:08:06	2019-11-06 12:18:05	👁

© 2019 **HMS**. All rights reserved

Doctor Login



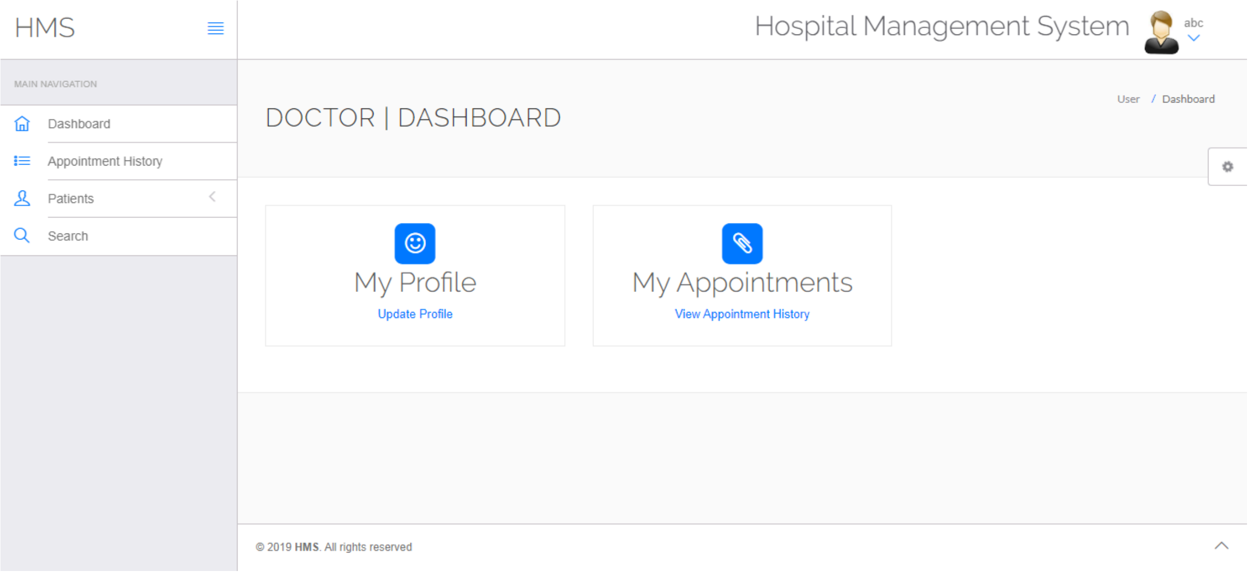
HMS | Doctor Login

Sign in to your account

Please enter your name and password to log in.

© 2019 HMS. All rights reserved

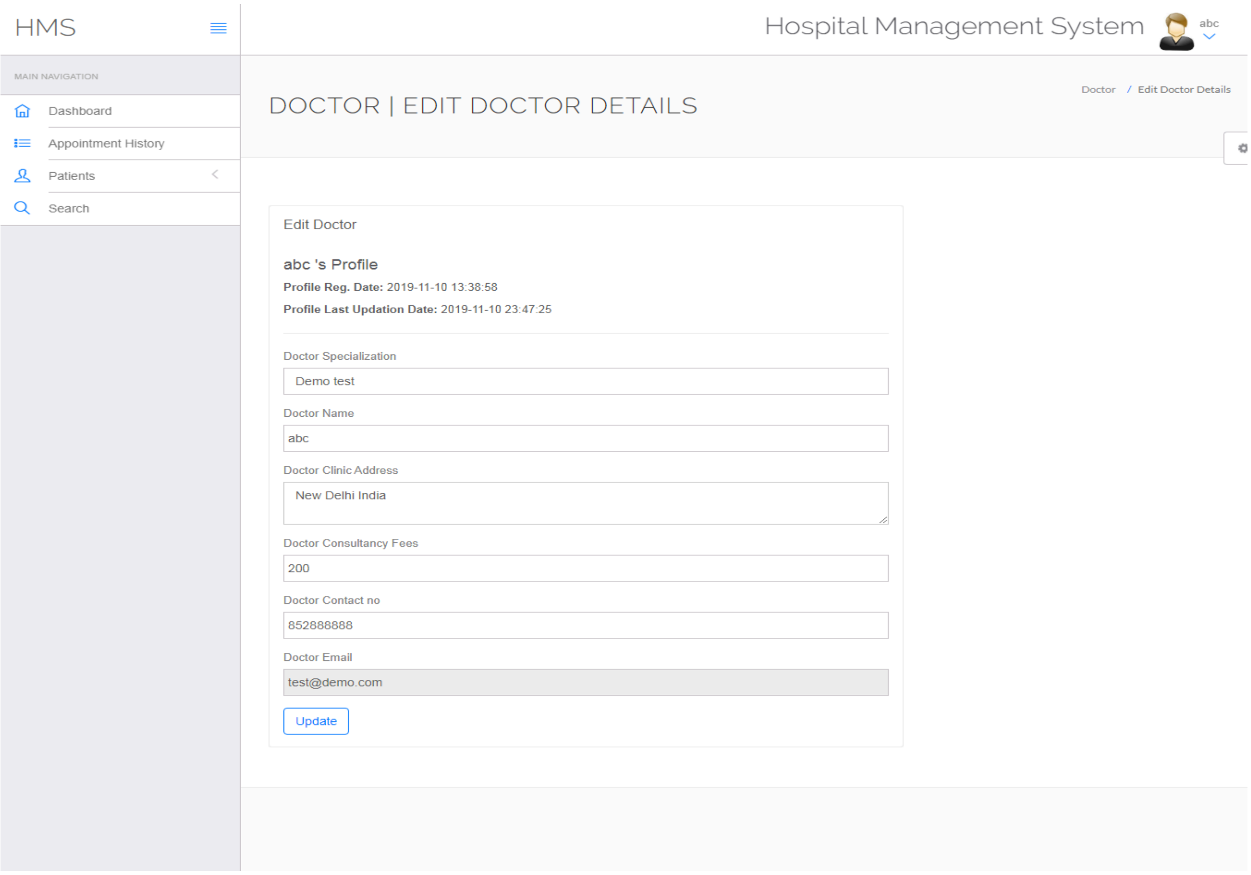
Doctor Dashboard



DOCTOR | DASHBOARD

© 2019 HMS. All rights reserved

Profile



DOCTOR | EDIT DOCTOR DETAILS

Edit Doctor

abc 's Profile

Profile Reg. Date: 2019-11-10 13:38:58

Profile Last Updation Date: 2019-11-10 23:47:25

Doctor Specialization: Demo test

Doctor Name: abc

Doctor Clinic Address: New Delhi India

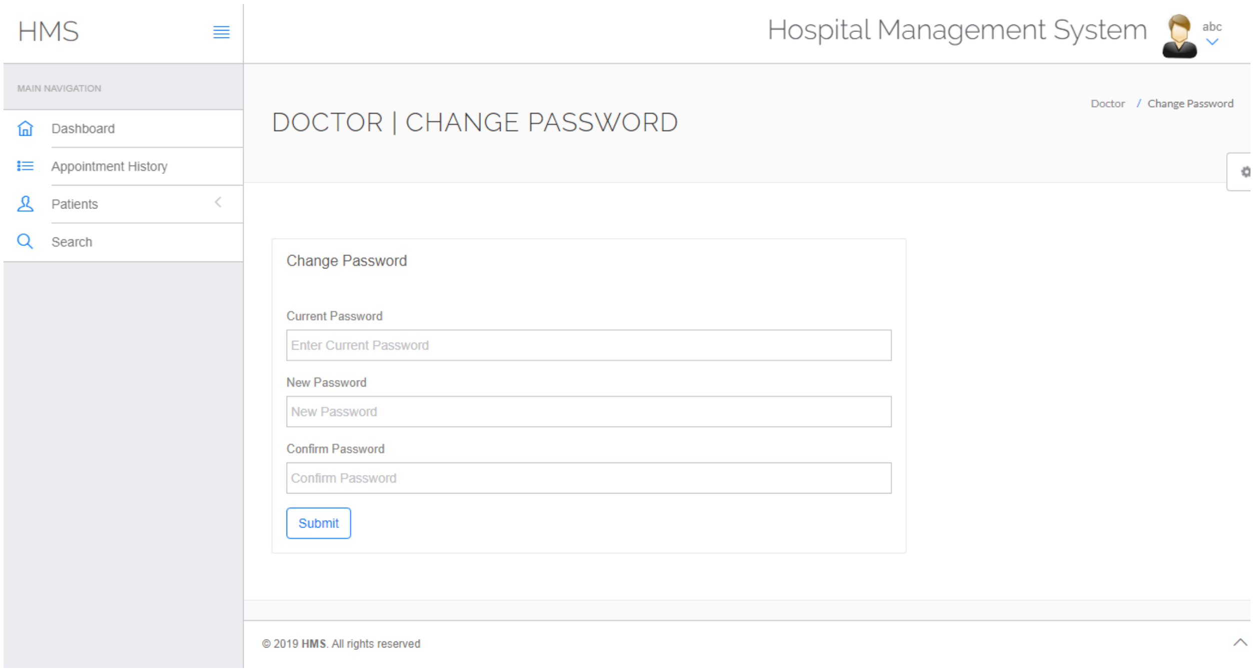
Doctor Consultancy Fees: 200

Doctor Contact no: 852888888

Doctor Email: test@demo.com

Update

Change Password



DOCTOR | CHANGE PASSWORD

Change Password

Current Password

Enter Current Password

New Password

New Password

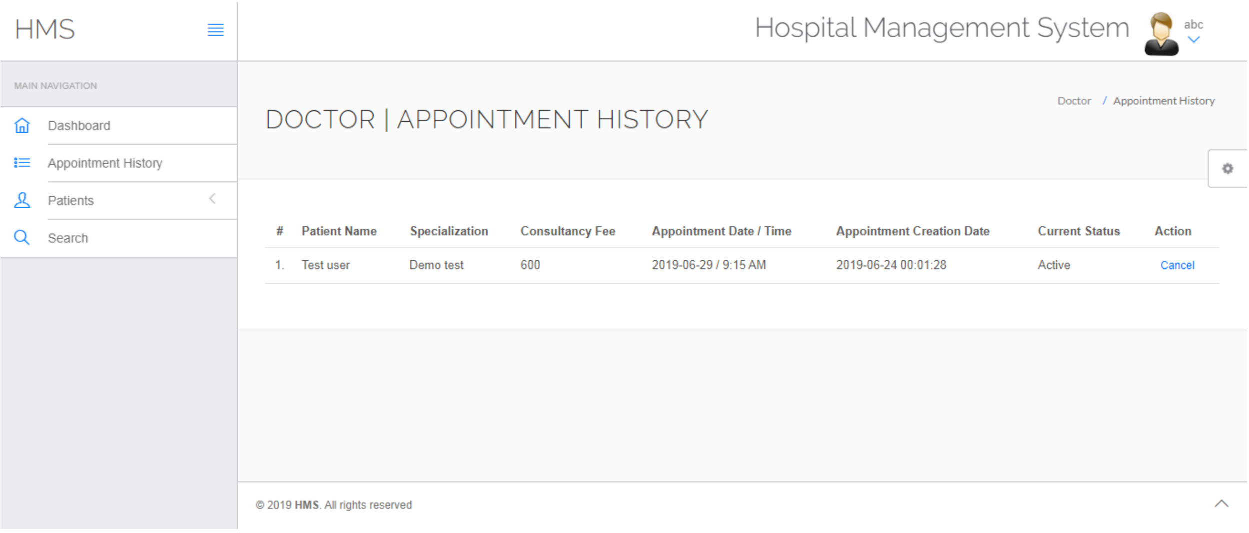
Confirm Password

Confirm Password

Submit

© 2019 **HMS**. All rights reserved

Appointment History

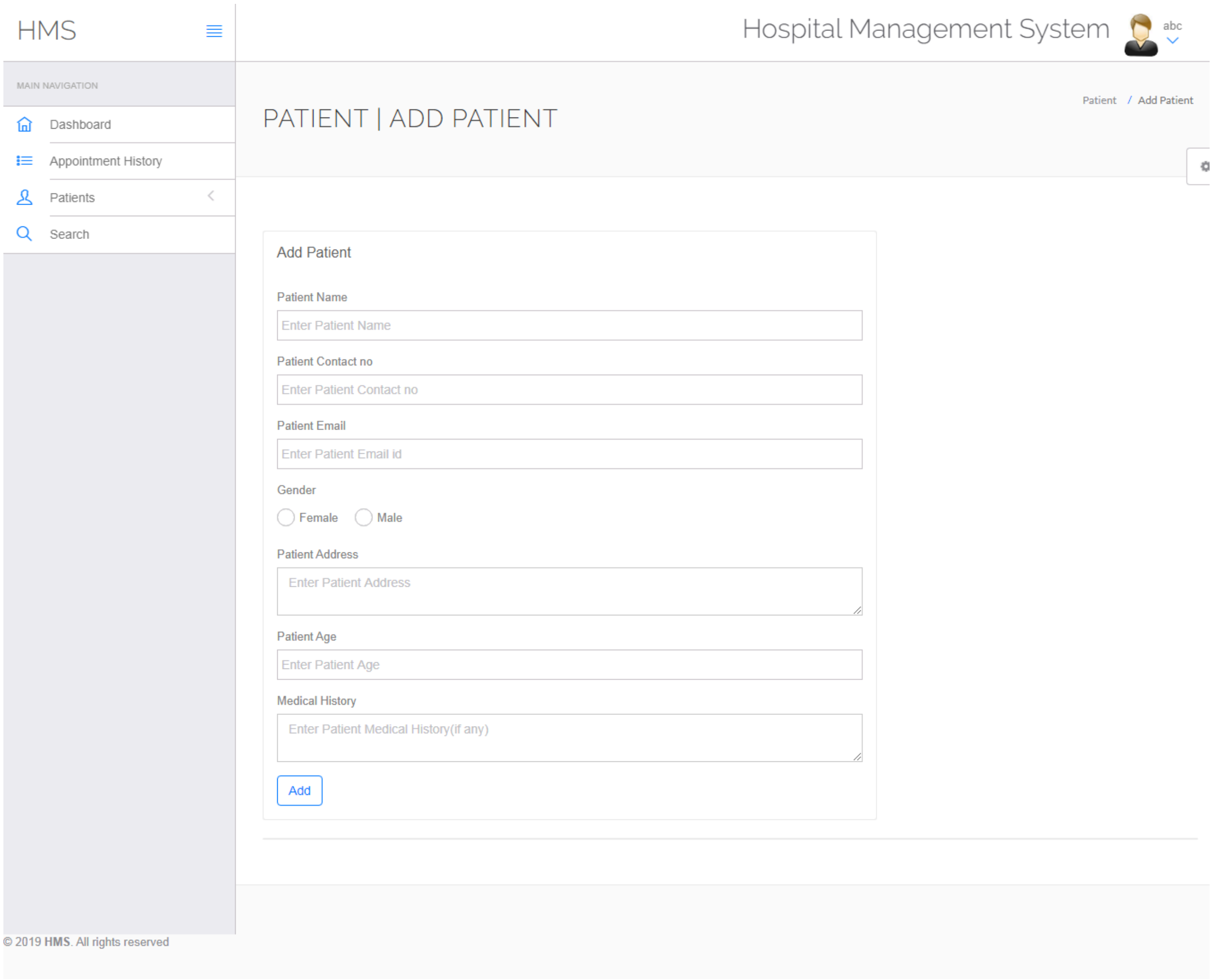


DOCTOR | APPOINTMENT HISTORY

#	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Test user	Demo test	600	2019-06-29 / 9:15 AM	2019-06-24 00:01:28	Active	Cancel

© 2019 **HMS**. All rights reserved

Add Patient



HMS

Hospital Management System abc

MAIN NAVIGATION

- Dashboard
- Appointment History
- Patients
- Search

PATIENT | ADD PATIENT

Patient / Add Patient

Add Patient

Patient Name

Enter Patient Name

Patient Contact no

Enter Patient Contact no

Patient Email

Enter Patient Email id

Gender

○ Female ○ Male

Patient Address

Enter Patient Address

Patient Age

Enter Patient Age

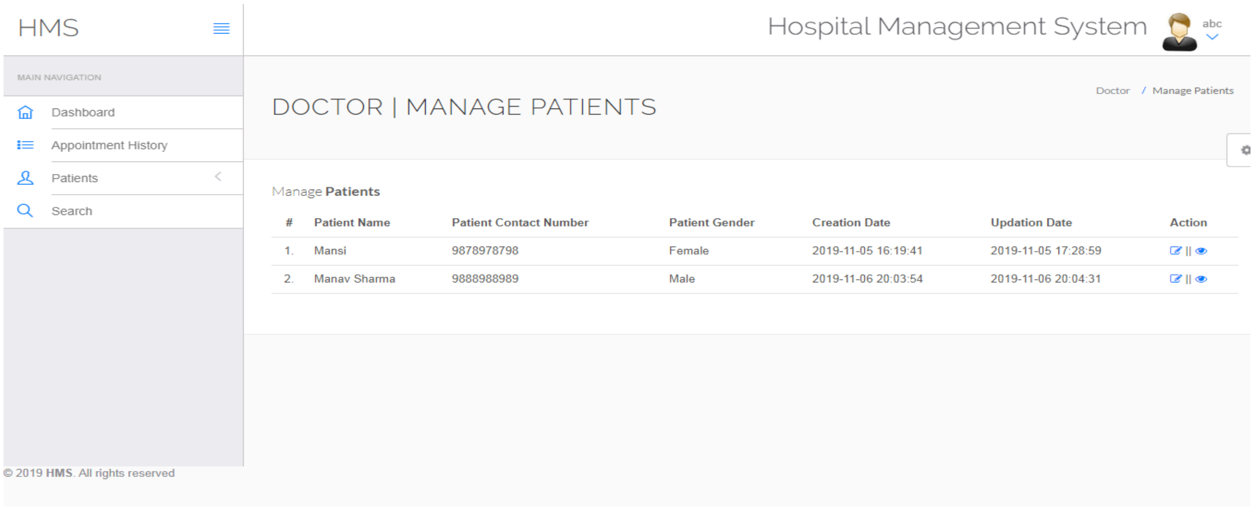
Medical History

Enter Patient Medical History(if any)

Add

© 2019 **HMS**. All rights reserved

Manage Patient



HMS

Hospital Management System abc

MAIN NAVIGATION

- Dashboard
- Appointment History
- Patients
- Search

DOCTOR | MANAGE PATIENTS

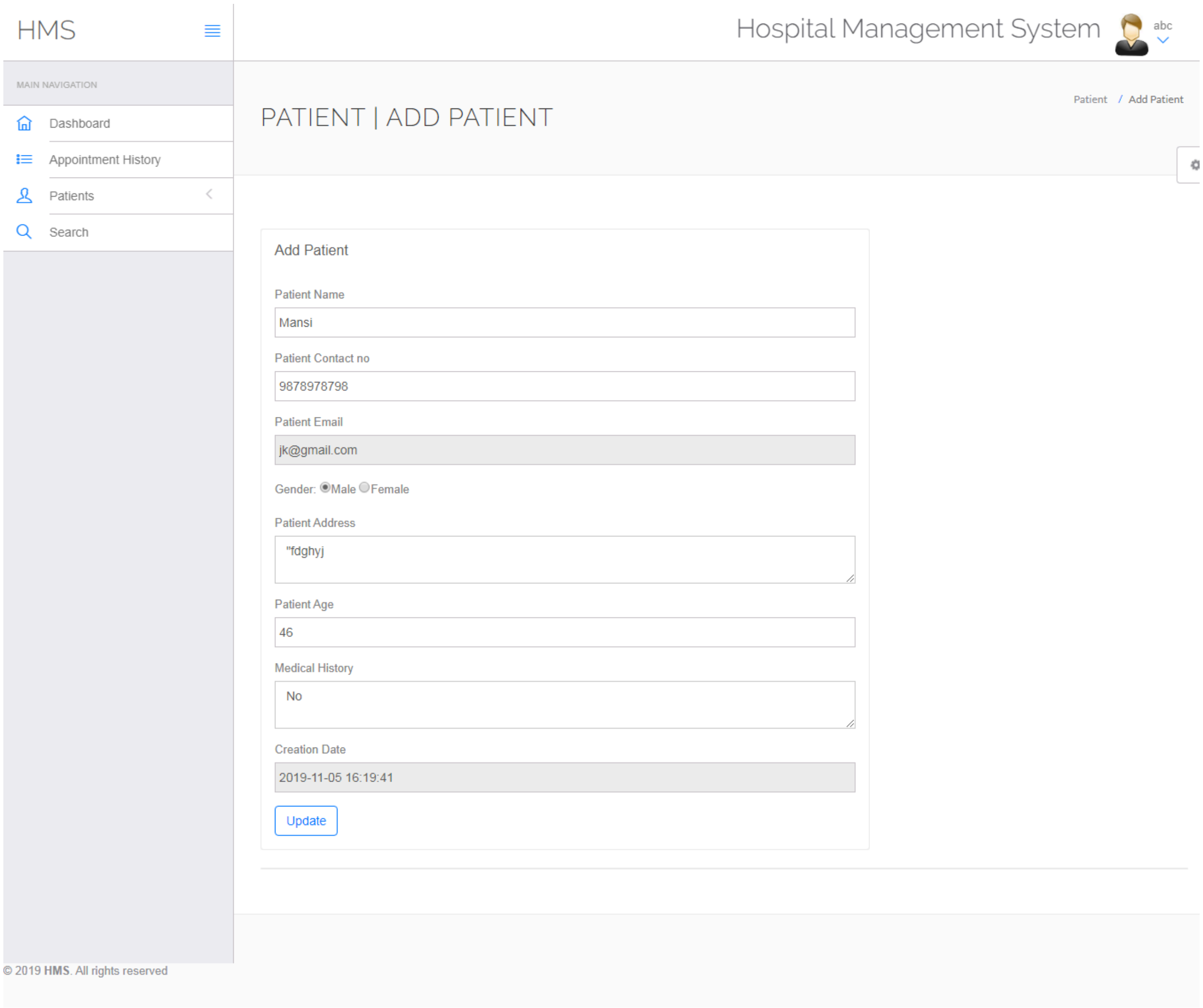
Doctor / Manage Patients

Manage **Patients**

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Mansi	9878978798	Female	2019-11-05 16:19:41	2019-11-05 17:28:59	✎ || 👁
2.	Manav Sharma	9888988989	Male	2019-11-06 20:03:54	2019-11-06 20:04:31	✎ || 👁

© 2019 **HMS**. All rights reserved

Edit Patient



PATIENT | ADD PATIENT

Add Patient

Patient Name

Mansi

Patient Contact no

9878978798

Patient Email

jk@gmail.com

Gender: ● Male ○ Female

Patient Address

"fdghyj

Patient Age

46

Medical History

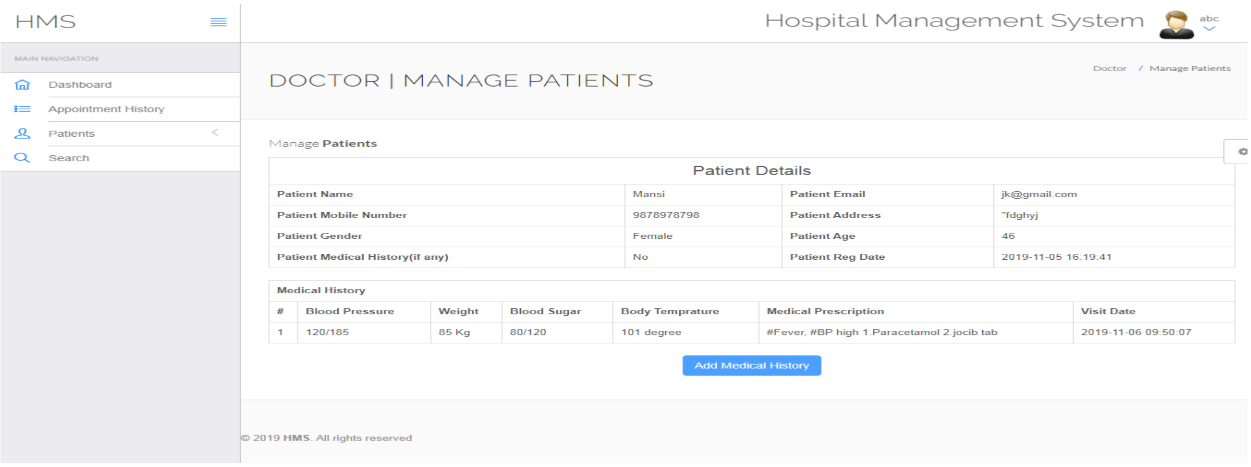
No

Creation Date

2019-11-05 16:19:41

© 2019 **HMS**. All rights reserved

View Patient Medical history



DOCTOR | MANAGE PATIENTS

Manage **Patients**

Patient Details

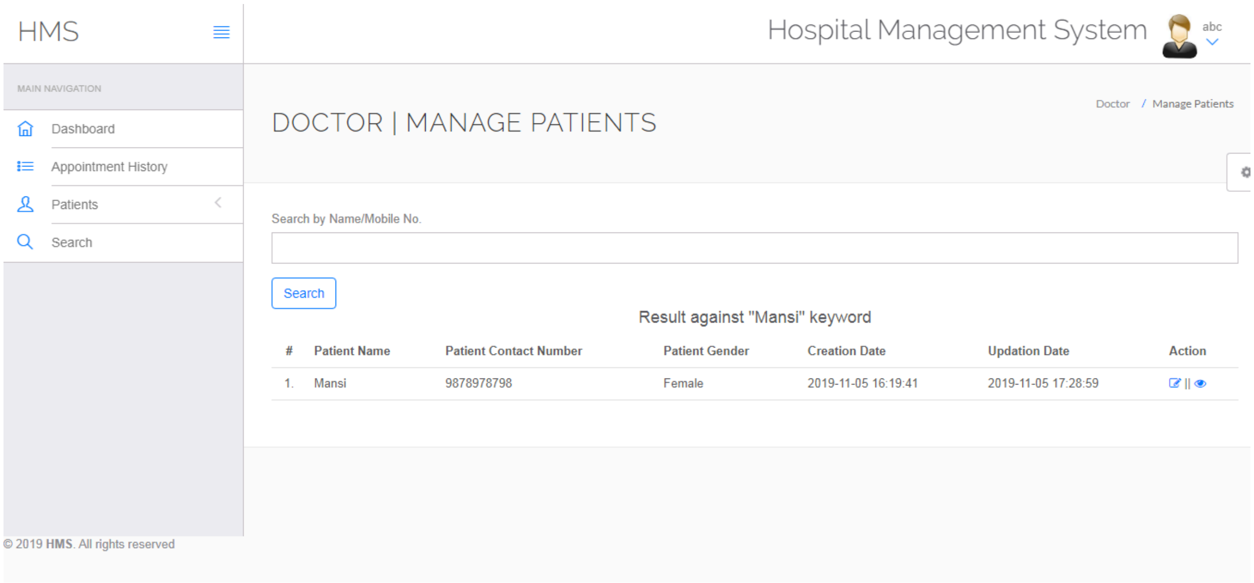
Patient Name	Mansi	**Patient Email**	jk@gmail.com
Patient Mobile Number	9878978798	**Patient Address**	"fdghyj
Patient Gender	Female	**Patient Age**	46
Patient Medical History(if any)	No	**Patient Reg Date**	2019-11-05 16:19:41

Medical History

#	Blood Pressure	Weight	Blood Sugar	Body Temprature	Medical Prescription	Visit Date
1	120/185	85 Kg	80/120	101 degree	#Fever, #BP high 1.Paracetamol 2.jocib tab	2019-11-06 09:50:07

© 2019 **HMS**. All rights reserved

Search



DOCTOR | MANAGE PATIENTS

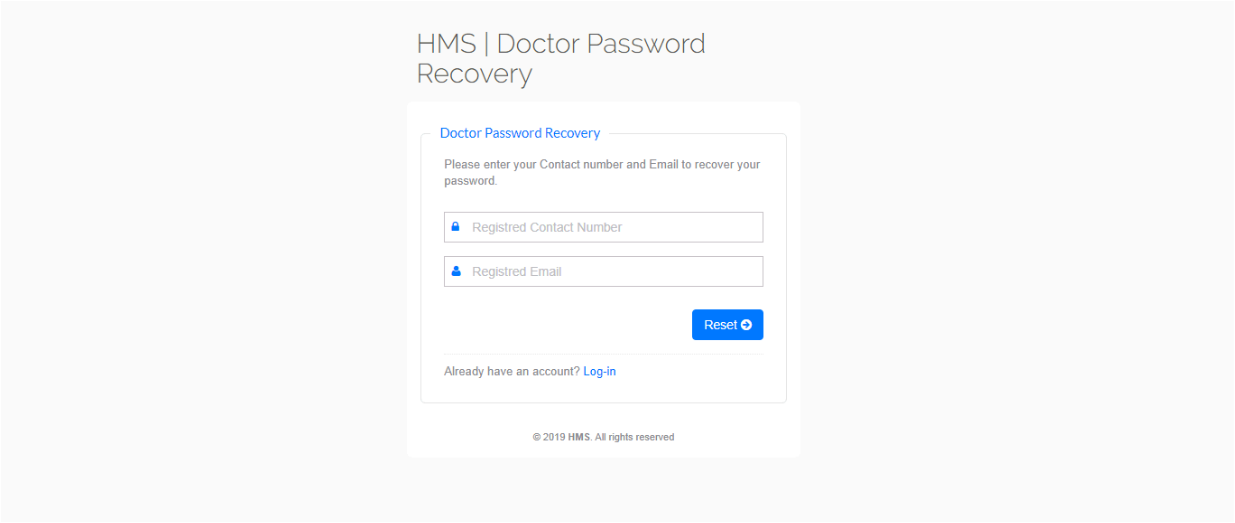
Search by Name/Mobile No.

Result against "Mansi" keyword

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Mansi	9878978798	Female	2019-11-05 16:19:41	2019-11-05 17:28:59	✎ || 👁

© 2019 **HMS**. All rights reserved

Forgot Password



HMS | Doctor Password Recovery

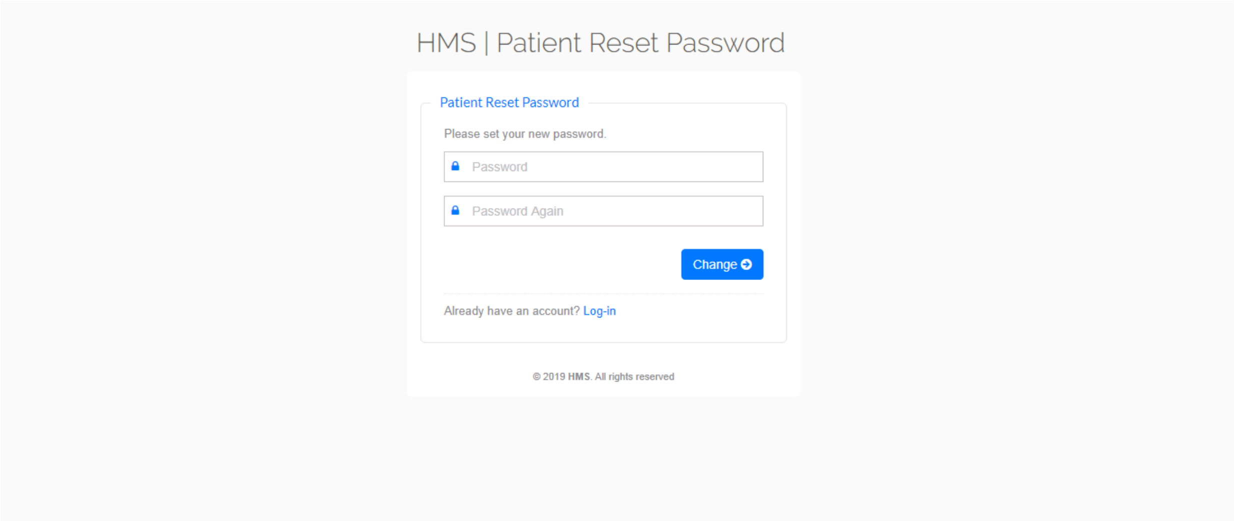
Doctor Password Recovery

Please enter your Contact number and Email to recover your password.

Already have an account? Log-in

© 2019 **HMS**. All rights reserved

Reset Password



HMS | Patient Reset Password

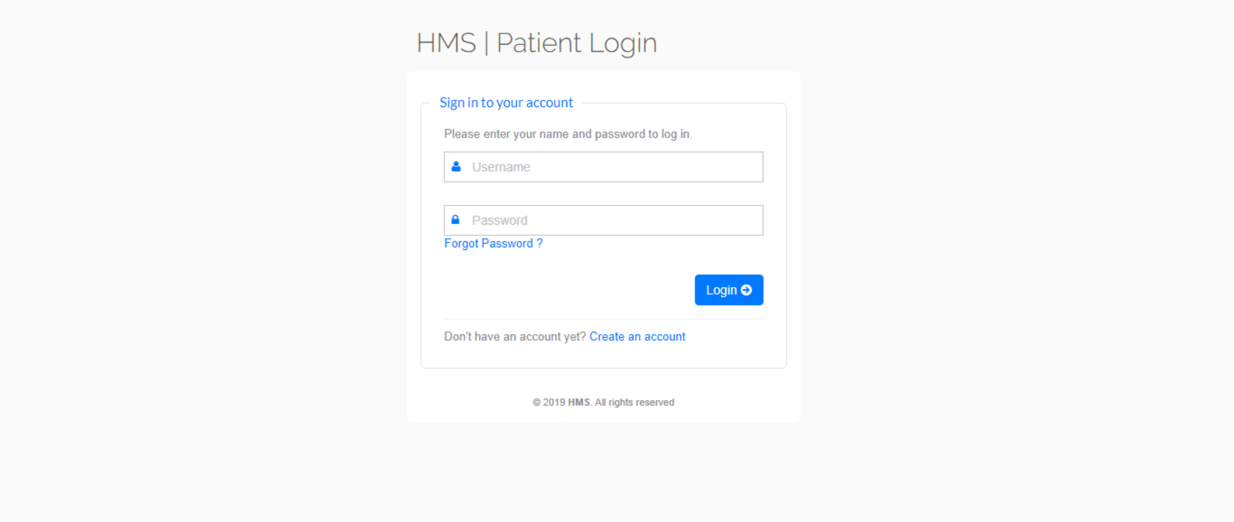
Patient Reset Password

Please set your new password.

Already have an account? Log-in

© 2019 **HMS**. All rights reserved

User Login



HMS | Patient Login

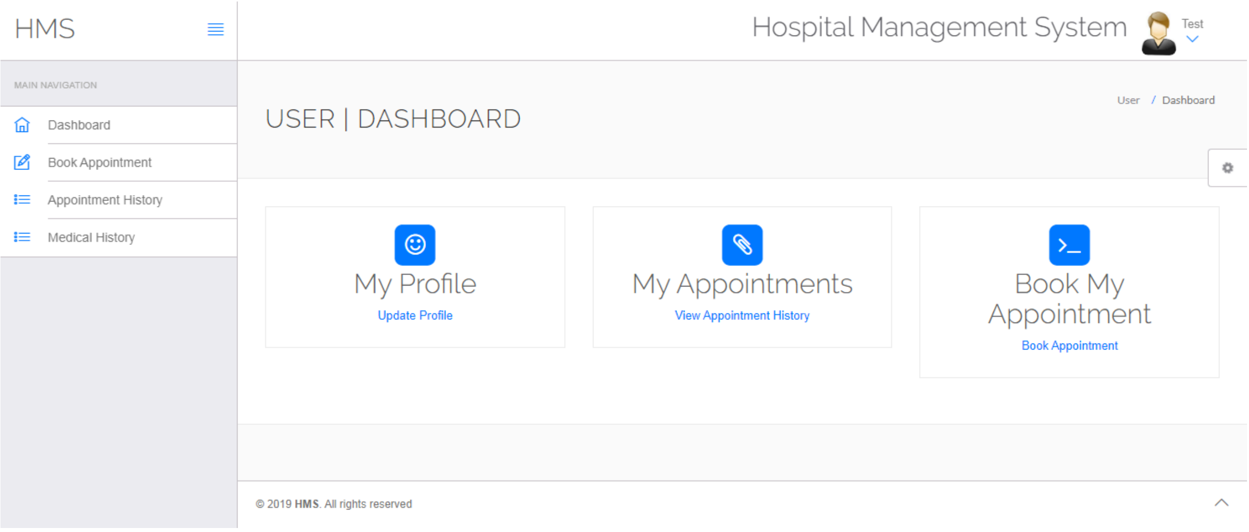
Sign in to your account

Please enter your name and password to log in.

Don't have an account yet? Create an account

© 2019 **HMS**. All rights reserved

User Dashboard



HMS

Hospital Management System

Test

MAIN NAVIGATION

- Dashboard
- Book Appointment
- Appointment History
- Medical History

USER | DASHBOARD

User / Dashboard

My Profile

Update Profile

My Appointments

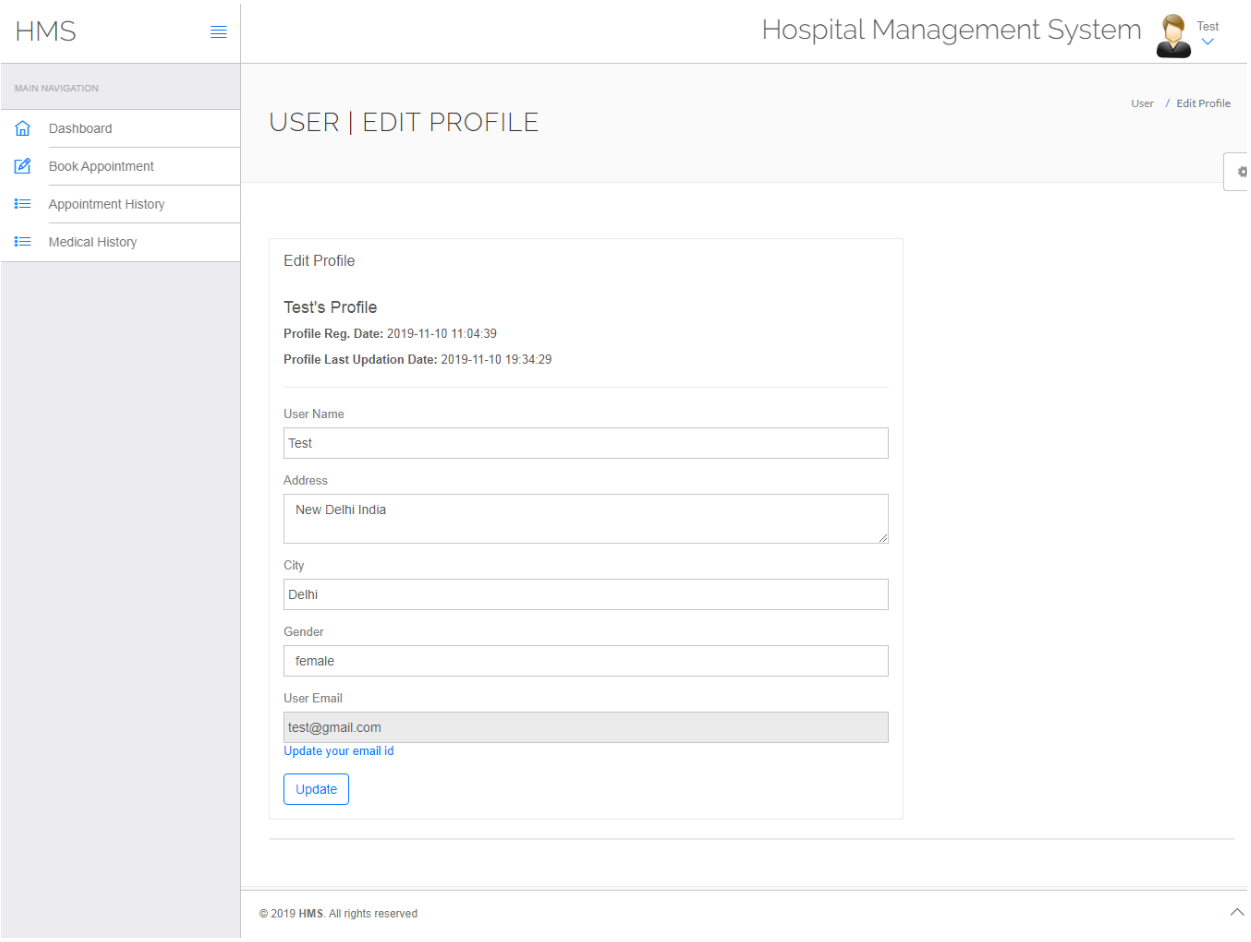
View Appointment History

Book My Appointment

Book Appointment

© 2019 **HMS**. All rights reserved

Edit Profile



HMS

Hospital Management System

Test

MAIN NAVIGATION

- Dashboard
- Book Appointment
- Appointment History
- Medical History

USER | EDIT PROFILE

User / Edit Profile

Edit Profile

Test's Profile

Profile Reg. Date: 2019-11-10 11:04:39

Profile Last Updation Date: 2019-11-10 19:34:29

User Name

Test

Address

New Delhi India

City

Delhi

Gender

female

User Email

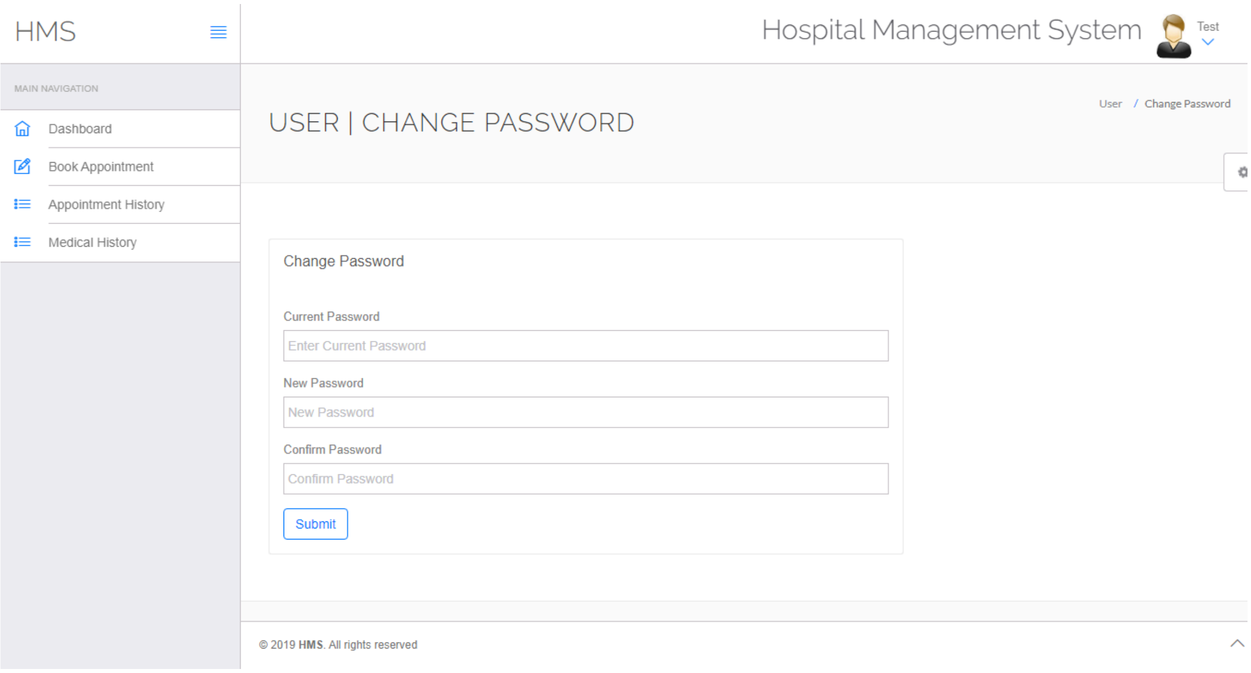
test@gmail.com

Update your email id

Update

© 2019 **HMS**. All rights reserved

Change Password



Change Password

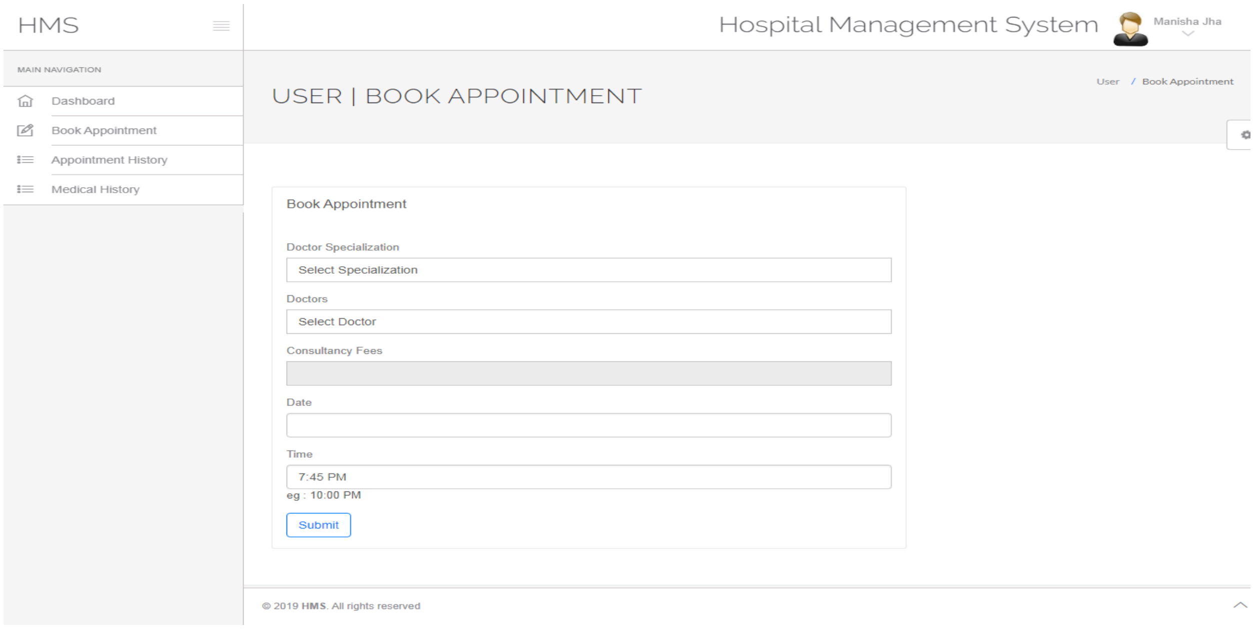
Current Password: Enter Current Password

New Password: New Password

Confirm Password: Confirm Password

© 2019 **HMS**. All rights reserved

Book Appointment



Book Appointment

Doctor Specialization: Select Specialization

Doctors: Select Doctor

Consultancy Fees:

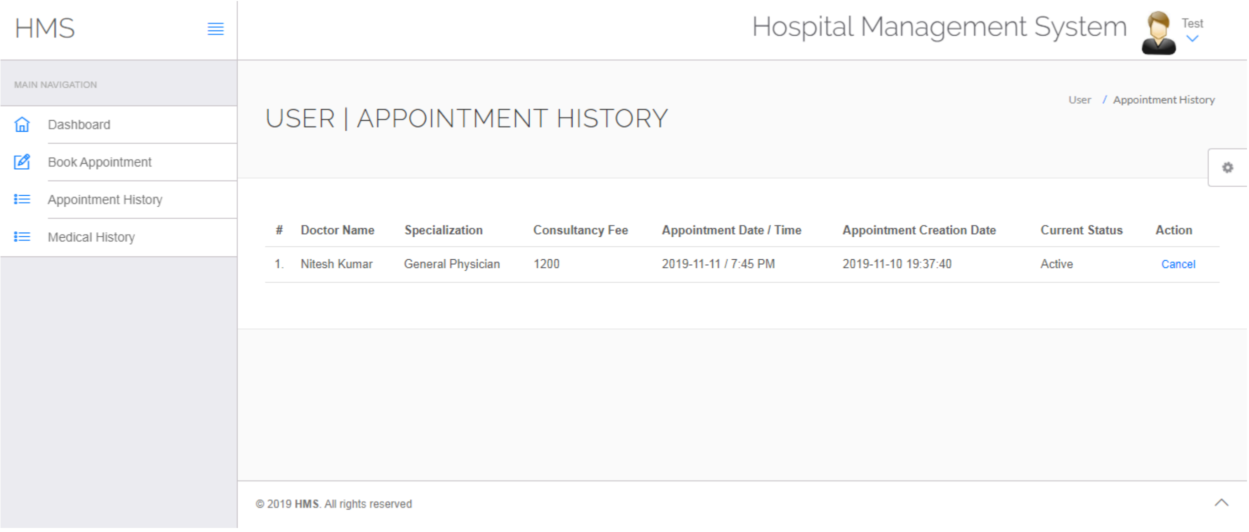
Date:

Time: 7:45 PM

eg : 10:00 PM

© 2019 **HMS**. All rights reserved

Appointment History

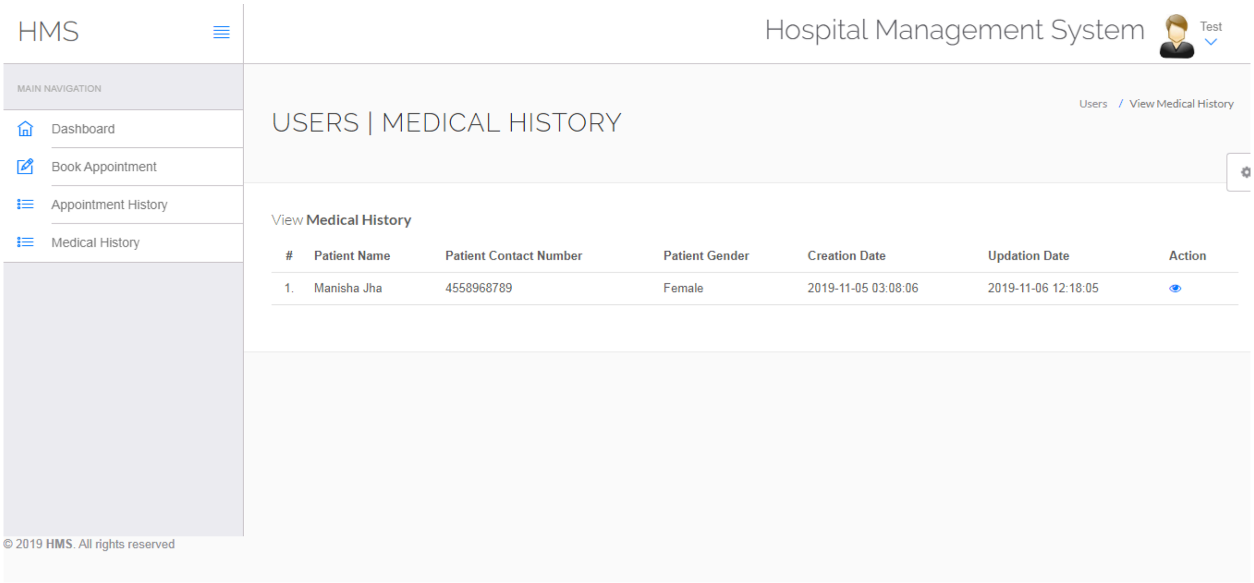


USER | APPOINTMENT HISTORY

#	Doctor Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Nitesh Kumar	General Physician	1200	2019-11-11 / 7:45 PM	2019-11-10 19:37:40	Active	Cancel

© 2019 **HMS**. All rights reserved

Medical History



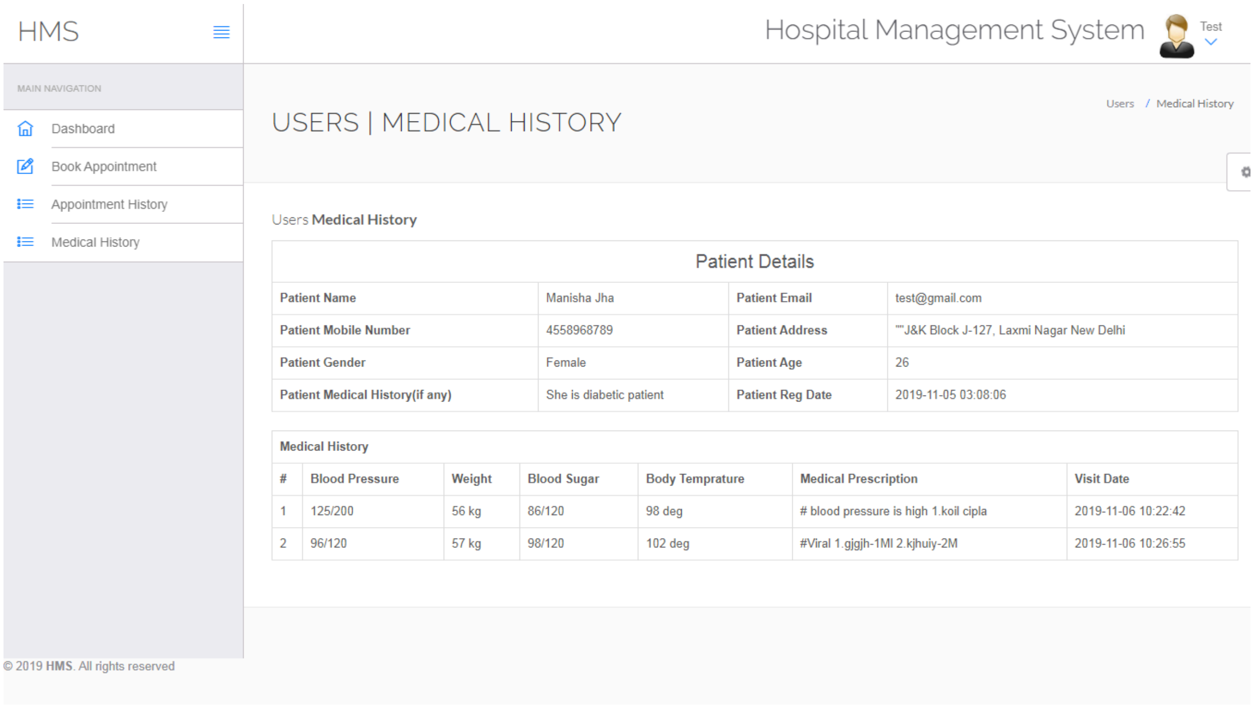
USERS | MEDICAL HISTORY

View **Medical History**

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Manisha Jha	4558968789	Female	2019-11-05 03:08:06	2019-11-06 12:18:05	👁

© 2019 **HMS**. All rights reserved

View Medical History



USERS | MEDICAL HISTORY

Users **Medical History**

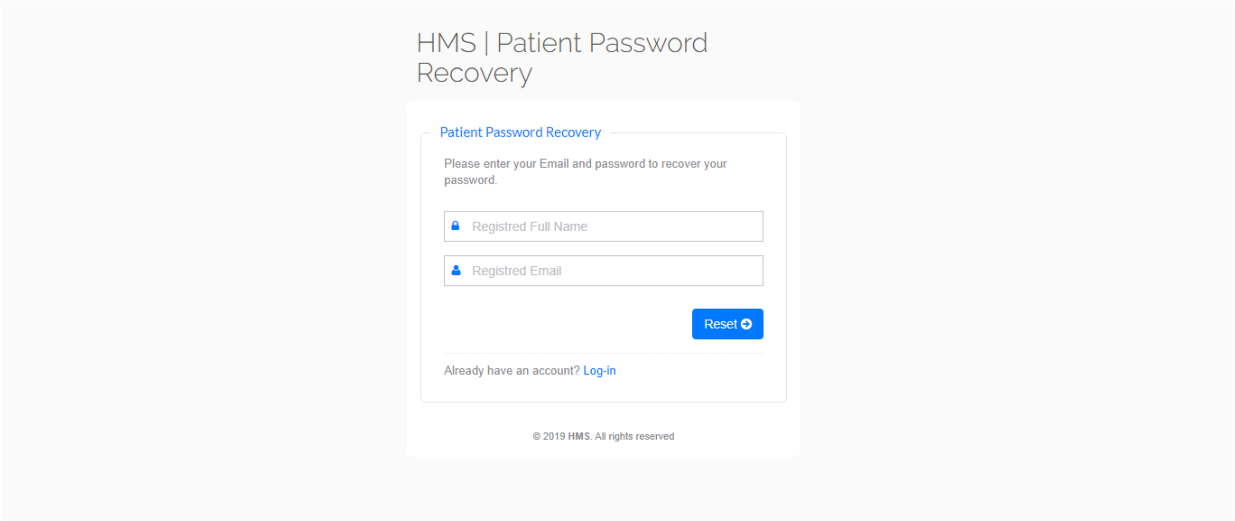
Patient Details			
Patient Name	Manisha Jha	**Patient Email**	test@gmail.com
Patient Mobile Number	4558968789	**Patient Address**	""J&K Block J-127, Laxmi Nagar New Delhi
Patient Gender	Female	**Patient Age**	26
Patient Medical History(if any)	She is diabetic patient	**Patient Reg Date**	2019-11-05 03:08:06

Medical History

#	Blood Pressure	Weight	Blood Sugar	Body Temprature	Medical Prescription	Visit Date
1	125/200	56 kg	86/120	98 deg	# blood pressure is high 1.koil cipla	2019-11-06 10:22:42
2	96/120	57 kg	98/120	102 deg	#Viral 1.gjgjh-1Ml 2.kjhuiy-2M	2019-11-06 10:26:55

© 2019 **HMS**. All rights reserved

Forgot Password



HMS | Patient Password Recovery

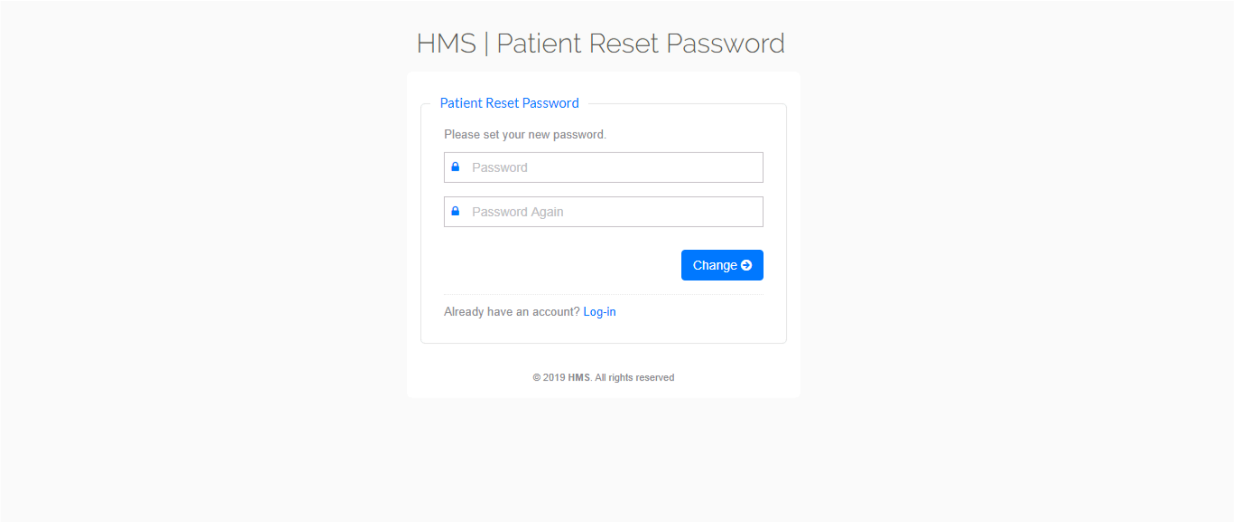
Patient Password Recovery

Please enter your Email and password to recover your password.

Already have an account? Log-in

© 2019 **HMS**. All rights reserved

Reset Password



HMS | Patient Reset Password

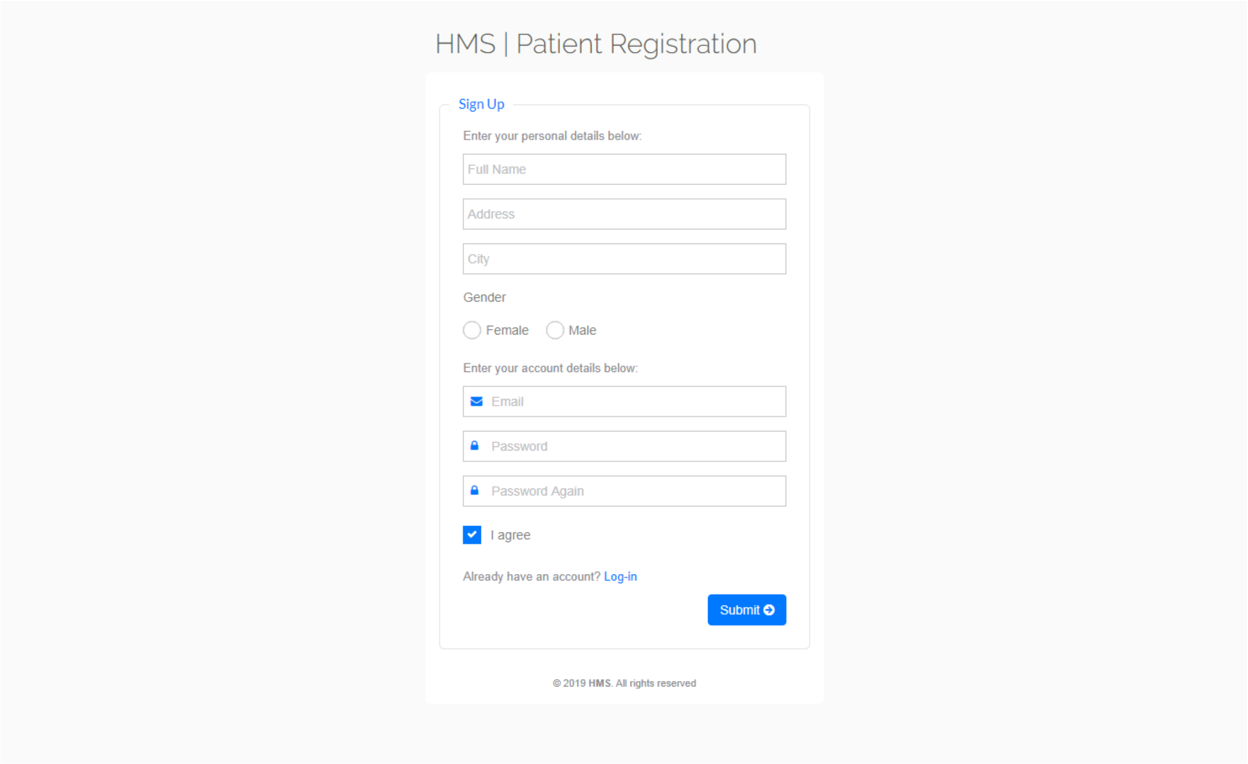
Patient Reset Password

Please set your new password.

Already have an account? Log-in

© 2019 HMS. All rights reserved

Patient Registration



HMS | Patient Registration

Sign Up

Enter your personal details below:

Gender

○ Female ○ Male

Enter your account details below:

☑ I agree

Already have an account? Log-in

© 2019 HMS. All rights reserved

8.CONCLUSION:

Since we are entering details of the patients electronically in the” Hospital Management System”, data will be secured. Using this application, we can retrieve patient’s history with a single click. Thus processing information will be faster. It guarantees accurate maintenance of Patient details. It easily reduces the book keeping task and thus reduces the human effort and increases accuracy speed.

9. BIBLOGRAPHY

1. PHP MySQL Website Programming: Problem - Design – Solution by Chris Lea, Mike Buzzard, DilipThomas, Jessy White-Cines’
2. Beginning PHP5, Apache, and MySQL Web Development (Programmer to Programmer) by Elizabeth Naramore
3. MySQL/PHP Database Applications, 2nd Edition by Brad Bulger
4. How to Do Everything with PHP and MySQL by VikramVaswani.