HOSPITAL MANAGEMENT SYSTEM

ABSTRACT

The proposed project is a smart appointment booking system that provides patients or any user an easy way of booking a doctor's appointment. This is a application that overcomes the issue of managing and booking appointments according to user's choice or demands. The task sometimes becomes very tedious for the compounder or doctor himself in manually allotting appointments for the users as per their availability. Hence this project offers an effective solution where users can view various booking slots available and select the preferred date and time. The already booked space will be marked yellow and will not be available for anyone else for the specified time. This system also allows users to cancel their booking anytime. The system provides an additional feature of calculating monthly earnings of doctor. Doctor has to just feed the system regularly with daily earnings and the system automatically generates a report of total amount earned at the end of the month.

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 interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

It is having mainly two modules. One is at Administration Level and other one is of user I.e. of Admin, patients and doctors. The Application maintains authentication in order to access the application. Administrator task includes managing doctors information, patient's information. To achieve this aim a database was designed one for the patient and other for the doctors which the admin can access. The complaints which are given by user will be referred by authorities. The Patient modules include checking appointments, prescription. User can also pay doctor's Fee online.

CONTENT

1. INTRODUCTION

Our project Doctor Search and appointment portal includes registration of patients, storing their details into the system, and also computerized appointment and doctor details. Our project has the facility to provide a unique id for each and every patient and stores the details and information of each and every patient automatically. It contains a search facility to know the current status of each patient room. User can search availability of a doctor and the details of a patient using the id. It efficiently maintains the details about the patient. Simultaneously updates changes made to any data, item in the entire data base. It is faster than manual system. The "Doctor Search and Appointment Portal" is a project which is very useful in doctor's day to day life. It is system such that it can enter details using a user name and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. 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 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 multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow. Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes

2. REQUIREMENT SPECIFICATION

2.1 HARDWARE REQUIREMENTS

- Processor : Dual core processor 2.6.0 GHZ
- RAM : 4GB
- Hard disk : 320 GB
- Compact Disk : 650 Mb
- Keyboard : Standard keyboard

2.2 SOFTWARE REQUIREMENTS

- Operating system : Windows 11 OS
- Front End : PHP
- Back end : MYSQL Server
- Tool : Macromedia Dreamweaver 8

2.3 EXISTING SYSTEM

In the existing system the patient needs to visit the doctor for booking we need to wait and the booking will be done manually so to maintain everything is always a problem. Patient will come and case register then after doctor will check diseases. If report is required for checkup then does it otherwise tack the medicine and leave the hospital. All details regarding hospital is maintain in separate file so to display information for public it not possible.

DISADVANTAGES OF EXISTING SYSTEM

- This system is time consuming.
- More man power required.
- This Hospital has not any website.
- Existing system is costly.

2.4 PROPOSED SYSTEM

All the hospital information is on website so patient can easily use this system. All the schedule of doctor is prescribed. Hospital about all information is regarding in website. Online case registers facility available. Patient can see the schedule regarding booking of time of particular doctor. Website information is like doctor appointment details, Hospital images, appointment time. The main purpose of website is advertisements. This system helps to reduce the waiting time of the patient. User can select the appointment time according to his preference. Available and booked slots are shown in effective graphical user interface.

ADVANTAGES OF PROPOSED SYSTEM

- Data stored in computerized.
- This system is faster.
- Less time required.
- Save money and time.

2.5 SOFTWARE DESCRIPTION

Front End

PHP: Hypertext Preprocessor (the name is a recursive acronym) is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. As a general-purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as standalone interpreter on most operating systems and computing platforms. PHP was originally created by Rasmus Lerdorf in 1995 and has been in continuous development ever since. The main implementation of PHP is now produced by The PHP Group and serves as the de facto standard for PHP as there is no formal specification. PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) because restrictions exist regarding the use of the term PHP.

Hypertext refers to files linked together using hyperlinks, such as HTML (HyperText Markup Language) files. Preprocessing is executing instructions that modify the output. Below is a demonstration of the difference between HTML and PHP files.

Accessing an HTML Page

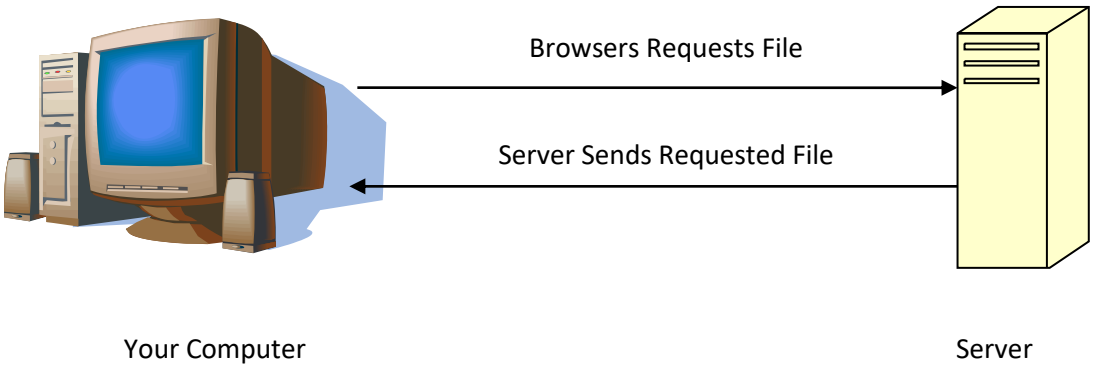
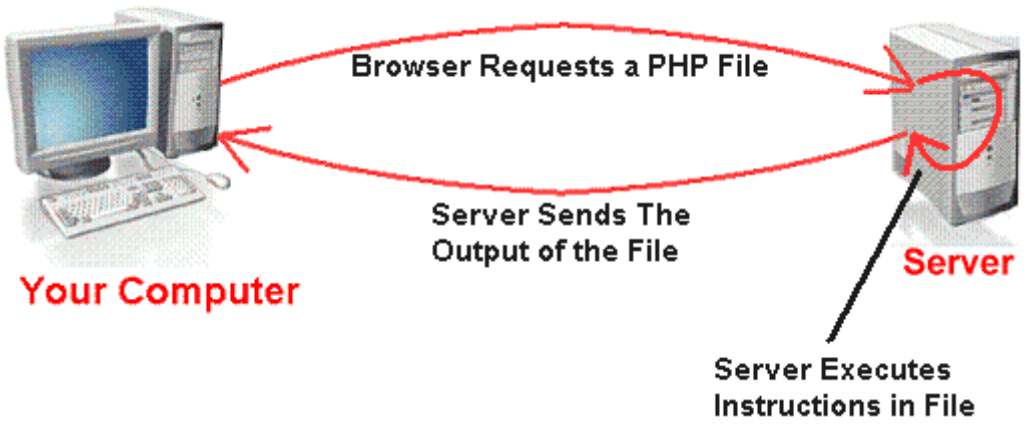


Fig 2 3.1 Accessing an HTML page

1. Your browser sends a request to that web page's server (computer) for the file (HTMLor image) you wish to view.
2. The web server (computer) sends the file requested back to your computer.
3. Your browser displays the file appropriately.
4. If you request a PHP file (ends with ".php"), the server handles it differently.

Accessing a PHP Page



Accessing a PHP Page

1. Your browser sends a request to that web page's server for the PHP file you wish to view.
2. The web server calls PHP to interpret and perform the operations called for in the PHP script.
3. The web server sends the output of the PHP program back to your computer.
4. Your browser displays the output appropriately.

Benefit of PHP

Because the server does processing, the output of PHP files changes when its input changes. For example, most of the pages on the Horticulture site have only two (2) PHP commands:

1. Include the header file that defines the links on the left, the banner, and the quick links at the top.
2. Include the footer file that displays the mission statement and Horticulture contact information.

Because including the files is performed every time the PHP file is accessed, when the header/footer files change, the new content will be immediately updated. In other words, if you add a new link, every page that includes the header will immediately display the new link.

Security

About 30% of all vulnerabilities listed on the National Vulnerability Database are linked to PHP. These vulnerabilities are caused mostly by not following best practice programming rules: technical security flaws of the language itself or of its core libraries are not frequent (23 in 2008, about 1% of the total). Recognizing that programmers make mistakes, some languages include taint checking to detect automatically the lack of input validation which induces many issues. Such a feature is being developed for PHP, but its inclusion in a release has been rejected several times in the past. There are advanced protection patches such as Suhosin and Hardening-Patch, especially designed for Web hosting environments.

PHPIDS adds security to any PHP application to defend against intrusions. PHPIDS detects attacks based on cross-site scripting (XSS), SQL injection, header injection, directory traversal, remote file execution, remote file inclusion, and denial-of-service (DoS)

Syntax

The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP (although non-PHP text is still subject to control structures described in PHP code). The most common delimiters are <?php to open and ?> to close PHP sections. <script language="php"> and </script> delimiters are also available, as are the shortened forms <?or<?= (which is used to echo back a string or variable) and ?> as well as ASP-style short forms <% or <%= and %>. While short delimiters are used, they make script files less portable as support for them can be disabled in the PHP configuration, and so they are discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

The first form of delimiters, <?php and ?>, in XHTML and other XML documents, creates correctly formed XML 'processing instructions'. This means that the resulting mixture of PHP code and other markup in the server-side file is itself well-formed XML.

Variables are prefixed with a dollar symbol, and a type does not need to be specified in advance. Unlike function and class names, variable names are case sensitive. Both double-quoted ("") and here-doc strings provide the ability to interpolate a variable's value into the string. PHP treats newlines as whitespace in the manner of a free-form language (except when inside string quotes), and statements are terminated by a semicolon. PHP has three types of comment syntax: /* */ marks block and inline comments; // as well as # are used for one-line comments. The echo statement is one of several facilities PHP provides to output text, e.g., to a Web browser.

In terms of keywords and language syntax, PHP is similar to most high level languages that follow the C style syntax. if conditions, for and while loops, and function returns are similar in syntax to languages such as C, C++, Java and Perl.

Data types

PHP stores whole numbers in a platform-dependent range, either a 64-bit or 32-bit signedinteger equivalent to the C-language long type. Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), octal, and hexadecimal notations. Floating point numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of scientific notation. PHP has a native Boolean type that is similar to the native Boolean types in Java and C++. Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in Perl and C++. The null data type represents a variable that has no value. The only value in the null data type is NULL. Variables of the "resource" type represent references to resources from external sources. These are typically created by functions from a particular extension, and can only be processed by functions from the same extension; examples include file, image, and database resources. Arrays can contain elements of any type that PHP can handle, including resources, objects, and even other arrays. Order is preserved in lists of values and in hashes with both keys and values, and the two can be intermingled. PHP also supports strings, which can be used with single quotes, double quotes, nowdoc or heredoc syntax.

Functions

PHP has hundreds of base functions and thousands more via extensions. These functions are well documented on the PHP site; however, the built-in library has a wide variety of naming conventions and inconsistencies. PHP currently has no functions for thread programming, although it does support multi-process programming on POSIX systems.

MY SQL

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, Word Press, phpBB, MyBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google(though not for searches), ImagebookTwitter, Flickr, Nokia.com, and YouTube.

Interimages

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

Graphical

The official MySQL Workbench is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).

MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

Command line

MySQL ships with some command line tools. Third-parties have also developed tools to manage a MySQL server, some listed below.

Maatkit - a cross-platform toolkit for MySQL, PostgreSQL and Memcached, developed in Perl Maatkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Maatkit is included with several GNU/Linux distributions such as CentOS and Debian and packages are available for Programming

MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, eCom Station, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, Open Solaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS, SCO Open Server, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists.[32]

MySQL is written in C and C++. Its SQL parser is written in yacc, and a home-brewed lexical analyzer. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBCinterimage called MyODBC allows additional programming languages that support the ODBC interimage to communicate with a MySQL database, such as ASP or ColdFusion. The HTSQL - URL-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

Features

As of April 2009, MySQL offered MySQL 5.1 in two different variants: the open source MySQL Community Server and the commercial Enterprise Server. MySQL 5.5 is offered under the same licences. They have a common code base and include the following features:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures
- Triggers
- Cursors
- Updatable Views
- Information schema
- Strict mode (ensures MySQL does not truncate or otherwise modify data to conform to an underlying data type, when an incompatible value is inserted into that type)
- X/Open XAdistributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine
- Independent storage engines (MyISAM for read speed, InnoDB for transactions and referential integrity, MySQL Archive for storing historical data in little space)
- Transactions with the InnoDB, and Cluster storage engines; savepoints with InnoDB
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Replication support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master, no automatic support for multiple masters per slave.
- Full-text indexing and searching using MyISAM engine
- Embedded database library
- Unicode support (however prior to 5.5.3 UTF-8 and UCS-2 encoded strings are limited to the BMP, in 5.5.3 and later use utf8mb4 for full unicode support)
- ACID compliance when using transaction capable storage engines (InnoDB and Cluster)
- Partititoned tables with pruning of partitions in optimiser

- Shared-nothing clustering through MySQL Cluster
- Hot backup (via mysqlhotcopy) under certain conditions
- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time): Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, Cluster, EXAMPLE, Maria, and InnoDB, which was made the default as of 5.5). Partner-developed storage engines (solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (formerly Brighthouse), Kickfire, XtraDB, IBM DB2). InnoDB used to be a partner-developed storage engine, but with recent acquisitions, Oracle now owns both MySQL core and InnoDB.

2.6 PROJECT DESCRIPTION

MODULES

Admin

- Login
- View Doctors Details
- View Patient Details
- View Appointments Details
- View Prescriptions List
- Add Doctor Details
- Delete Doctor Details
- Queries

Doctor

- Login
- View Appointments Details
- View Prescription List
- Accept/Reject Booking

Patient

- Register
- Login
- View Doctors Details
- Book Appointment
- View Appointment History
- Appointment Conformation
- View Prescriptions List
- Pay Bill

MODULE S DESCRIPTION

Admin

Admin has the full access to the system which means he is able to manage any activity with regard to the system. He is the highest privileged user who can access to the system.

- Login
  In this module, the admin can login in the system using his/her username and password.

- View Doctors Details
  In this module the admin can view the doctors list like doctor name, id, specialization fees etc.

- View Patient Details
  In this module the admin can view the patient details like patient name, id, address, location etc.

- View Appointments Details
  In this module the admin can view the appointments details like, appointments date and time, patient name and id, doctor name, Consultancy Fees etc.

- View Prescriptions List
  In this module the admin can view the Prescriptions List like doctor name, patient name, id, disease, allergy, prescriptions etc.

- Add Doctor Details
  In this module the admin only can register doctor details such as doctor name, password, gender, age, mobile number, mail id, specialization and so on.

- Delete Doctors

  In this module the admin only can delete the doctors for any reason.

- Queries

  In this module the admin only can view the queries like user name, E-mail, contact, message etc.

Doctor

Doctors can view the patient appointment list and provide the confirmation or make changes in the appointment list if required. Doctors have access to only records of those patients whom they are treating.

- Login

  In this module, the doctor can login in the system using his/her username and password. The user name and password is provided by admin.

- View Appointments Details

  In this module, the doctor can view the Appointments details in the system. The Appointments details are such as user name, Appointment time, date etc. These details are stored in the system.

- View Prescription List

  In this module the doctor can view the Prescriptions List like patient name, id, disease, allergy, prescriptions etc.

- Accept/Reject Booking

  In this module the doctor can accept or reject the user booking request.

Patient

Patients can choose the best preferred appointments from the options provided and can also change the appointment schedule or cancel it. After appt. is confirmed by the respective doctor they can pay their consultant fee online. Patients have access to only their records.

- Register

  There is registration form available where new patient can create their account by providing required information to the system. The registration form details are like name, email, gender, mobile number, address, and etc. These details are stored in the database. And then can getting to the username and password in the system.

- Login

  In this module, the patient can login in the system using his/her username and password.

- View Doctors Details

  In this module the patient can view the doctor details such as doctor name, gender, age, mail id, specialization and so on. After viewing all information the patient can book the particular doctor for appointment.

- Book Appointment

  In this module the patient can view the booking details like doctor name, date, time etc.

- View Appointment History

  In this module the patient can view the appointment details like doctor name, Consultancy Fees, appointment date and time, current status etc.

- Appointment Conformation

  In this module the patient can view the appointment conformation notification from doctor. The notification is in the form of SMS or email.

- View Prescriptions List

  In this module the patient can view the Prescriptions List like doctor name, appointment date and time, disease, allergy, prescriptions etc.

- Pay Bill

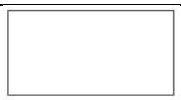
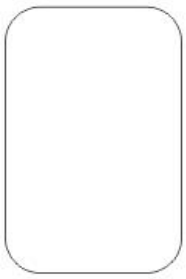
  In this module the patient can pay bill like bill paid successfully and view the bill receipt in pdf formet

3. DESIGN

3.1 DATA FLOW DIAGRAM

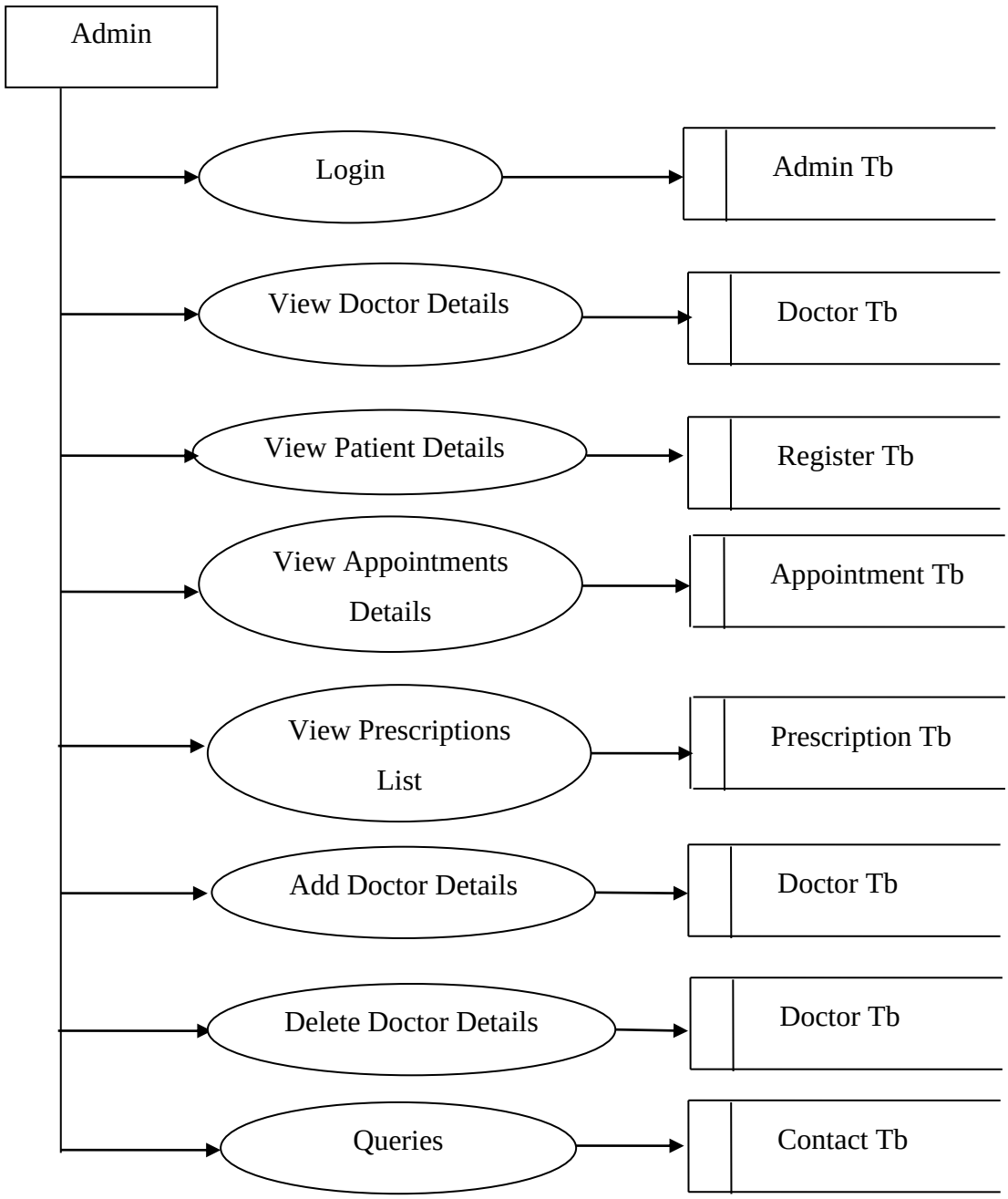
A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

Data flow Symbols

Symbol	Description
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.
	A **data flow**.

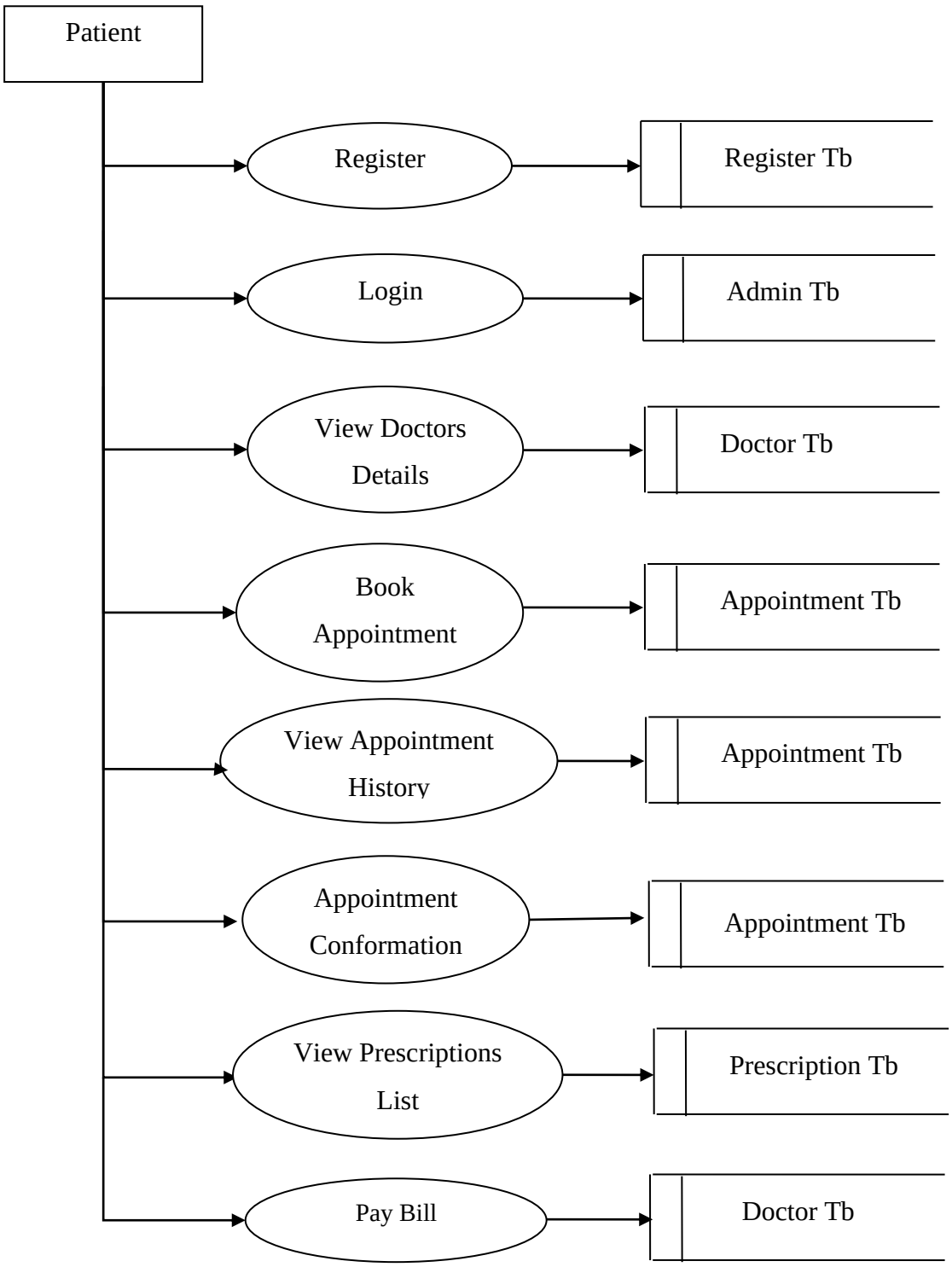
LEVEL 0

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.



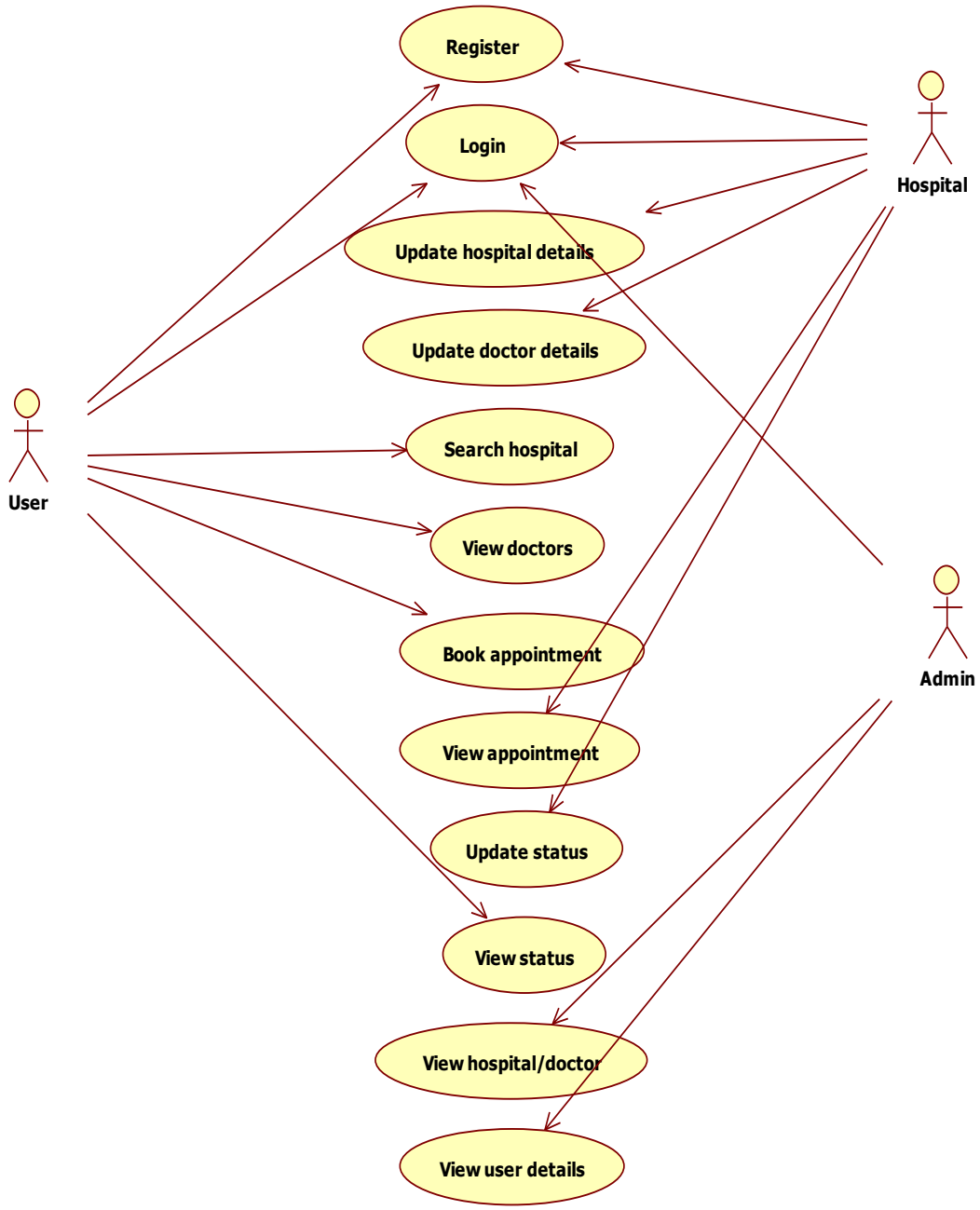
LEVEL 1

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manage on screen or paper.



3.2 USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.



3.3 TABLE DESIGN

A table is a data structure that organizes information into rows and columns. It can be used to both store and display data in a structured format. For example, databases store data in tables so that information can be quickly accessed from specific rows. Websites often use tables to display multiple rows of data on page. Spreadsheets combine both purposes of a table by storing and displaying data in a structured format.

Databases often contain multiple tables, with each one designed for a specific purpose. For example, a company database may contain separate tables for employees, clients, and suppliers. Each table may include its own set of fields, based on what data the table needs to store. In database tables, each field is considered a column, while each entry (or record), is considered a row. A specific value can be accessed from the table by requesting data from an individual column and row.

Database myhmsdb

Table Name: admintb

Column	Type	Null	Default
username	varchar(50)	No	NULL
password	varchar(30)	No	NULL

Table Name: appointmenttb

Column	Type	Null	Default
pid	int(11)	No	NULL
ID	int(11)	No	NULL
fname	varchar(20)	No	NULL
lname	varchar(20)	No	NULL
gender	varchar(10)	No	NULL
age	Int(2)	No	NULL
email	varchar(30)	No	NULL
contact	varchar(10)	No	NULL
doctor	varchar(30)	No	NULL
docFees	int(5)	No	NULL
appdate	date	No	NULL
apptime	time	No	NULL
userStatus	int(5)	No	NULL
doctorStatus	int(5)	No	NULL

Table Name: contact

Column	Type	Null	Default
name	varchar(30)	No	NULL
email	text	No	NULL
contact	varchar(10)	No	NULL
message	varchar(200)	No	NULL

Table Name: doctb

Column	Type	Null	Default
username	varchar(50)	No	NULL
password	varchar(50)	No	NULL
email	varchar(50)	No	NULL
spec	varchar(50)	No	NULL
docFees	int(10)	No	NULL

Table Name: patreg

Column	Type	Null	Default
pid	int(11)	No	NULL
fname	varchar(20)	No	NULL
lname	varchar(20)	No	NULL
gender	varchar(10)	No	NULL
email	varchar(30)	No	NULL
contact	varchar(10)	No	NULL
password	varchar(30)	No	NULL
cpassword	varchar(30)	No	NULL

Table Name: prestb

Column	Type	Null	Default
doctor	varchar(50)	No	NULL
pid	int(11)	No	NULL
ID	int(11)	No	NULL
fname	varchar(50)	No	NULL
lname	varchar(50)	No	NULL
appdate	date	No	NULL
apptime	time	No	NULL
disease	varchar(250)	No	NULL
allergy	varchar(250)	No	NULL
prescription	varchar(1000)	No	NULL

4. IMPLEMENTATION

4.1 SOURCE CODE

index.php

```
<html>
<head>
 <title>HMS</title>
 <link rel="shortcut icon" type="image/x-icon" href="images/favicon.png" />
<link rel="stylesheet" type="text/css" href="style1.css">
<link href="https://fonts.googleapis.com/css?family=IBM+Plex+Sans&display=swap"
rel="stylesheet">
<!-- <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
integrity="sha384-
HSMxcRTRxnN+Bdg0JdbxYKrThecOKuH5zCYotlSAcp1+c8xmyTe9GYg1l9a69psu"
crossorigin="anonymous"> -->

<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

<link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">

<style >
 .form-control {
 border-radius: 0.75rem;
}
</style>
```

```
<script>
 var check = function() {
 if (document.getElementById('password').value ==
 document.getElementById('cpassword').value) {
 document.getElementById('message').style.color = '#5dd05d';
 document.getElementById('message').innerHTML = 'Matched';
 } else {
 document.getElementById('message').style.color = '#f55252';
 document.getElementById('message').innerHTML = 'Not Matching';
 }
}

function alphaOnly(event) {
 var key = event.keyCode;
 return ((key >= 65 && key <= 90) || key == 8 || key == 32);
};

function checklen()
{
 var pass1 = document.getElementById("password");
 if(pass1.value.length<6){
 alert("Password must be at least 6 characters long. Try again!");
 return false;
 }
}

</script>
</head>
<!------ Include the above in your HEAD tag ---------->
<body>
<nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav" >
 <div class="container">
```

```
<a class="navbar-brand js-scroll-trigger" href="#" style="margin-top: 10px;margin-left:-65px;font-family: 'IBM Plex Sans', sans-serif;"><h4><i class="fa fa-user-plus" aria-hidden="true"></i>  Hospital Management System</h4></a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarResponsive">
<ul class="navbar-nav ml-auto">
<li class="nav-item" style="margin-right: 40px;">
<a class="nav-link js-scroll-trigger" href="index.php" style="color: white;font-family: 'IBM Plex Sans', sans-serif;"><h6>HOME</h6></a>
</li>

<li class="nav-item" style="margin-right: 40px;">
<a class="nav-link js-scroll-trigger" href="services.html" style="color: white;font-family: 'IBM Plex Sans', sans-serif;"><h6>ABOUT US</h6></a>
</li>

<li class="nav-item">
<a class="nav-link js-scroll-trigger" href="contact.html" style="color: white;font-family: 'IBM Plex Sans', sans-serif;"><h6>CONTACT</h6></a>
</li>
</ul>
</div>
</div>
</nav>
<div class="container register" style="font-family: 'IBM Plex Sans', sans-serif;">
<div class="row">
<div class="col-md-3 register-left" style="margin-top: 10%;right: 5%">
<img src="images/hmsicon.png" alt=""/>
<h3>Welcome To Andavan Hospitals</h3>
```

```
</div>
<div class="col-md-9 register-right" style="margin-top: 40px;left: 80px;">
<ul class="nav nav-tabs nav-justified" id="myTab" role="tablist" style="width: 40%;">
<li class="nav-item">
<a class="nav-link active" id="home-tab" data-toggle="tab" href="#home" role="tab" aria-controls="home" aria-selected="true">Patient</a>
</li>
<li class="nav-item">
<a class="nav-link" id="profile-tab" data-toggle="tab" href="#profile" role="tab" aria-controls="profile" aria-selected="false">Doctor</a>
</li>
<li class="nav-item">
<a class="nav-link" id="profile-tab" data-toggle="tab" href="#admin" role="tab" aria-controls="admin" aria-selected="false">Receptionist</a>
</li>
</ul>
<div class="tab-content" id="myTabContent">
<div class="tab-pane fade show active" id="home" role="tabpanel" aria-labelledby="home-tab">
<h3 class="register-heading">Register as Patient</h3>
<form method="post" action="func2.php">
<div class="row register-form">

<div class="col-md-6">
<div class="form-group">
<input type="text" class="form-control" placeholder="First Name *" name="fname" onkeydown="return alphaOnly(event);" required/>
</div>
<div class="form-group">
<input type= "numbersOnly" minlength="1" maxlength="2" name="Age" class="form-control" placeholder="Age *" />
</div>
<div class="form-group">
<input type="email" class="form-control" placeholder="Your Email *" name="email" />
```

```
</div>
<div class="form-group">
<input type="password" class="form-control" placeholder="Password *" id="password" name="password" onkeyup='check();' required/>
</div>

<div class="form-group">
<div class="maxl">
<label class="radio inline">
<input type="radio" name="gender" value="Male" checked>
<span> Male </span>
</label>
<label class="radio inline">
<input type="radio" name="gender" value="Female">
<span>Female </span>
</label>
<label class="radio inline">
<input type="radio" name="gender" value="Others">
<span>Others </span>
</label>
</div>
<a href="index1.php">Already have an account?</a>
</div>
</div>

<div class="col-md-6">
<div class="form-group">
<input type="text" class="form-control" placeholder="Last Name *" name="lname" onkeydown="return alphaOnly(event);" required/>
</div>
<div class="form-group">
<input type="text" minlength="10" maxlength="10" name="contact" class="form-control" placeholder="contact *" />
</div>
```

```
<div class="form-group">
<input type="tel" minlength="10" maxlength="10" name="date" class="form-control" placeholder="dd-mm-yyyy *" />
</div>
<div class="form-group">
<input type="password" class="form-control" id="cpassword" placeholder="Confirm Password *" name="cpassword" onkeyup='check();' required/><span id='message'></span>
</div>
<input type="submit" class="btnRegister" name="patsub1" onclick="return checklen();" value="Register"/>
</div>

</div>
</form>
</div>


<div class="tab-pane fade show" id="profile" role="tabpanel" aria-labelledby="profile-tab">
<h3 class="register-heading">Login as Doctor</h3>
<form method="post" action="func1.php">
<div class="row register-form">
<div class="col-md-6">
<div class="form-group">
<input type="text" class="form-control" placeholder="User Name *" name="username3" onkeydown="return alphaOnly(event);" required/>
</div>
</div>
<div class="col-md-6">
<div class="form-group">
<input type="password" class="form-control" placeholder="Password *" name="password3" required/>
</div>

<input type="submit" class="btnRegister" name="docsub1" value="Login"/>
```

```
</div>
</div>
</form>
</div>

<div class="tab-pane fade show" id="admin" role="tabpanel" aria-labelledby="profile-tab">
<h3 class="register-heading">Login as Admin</h3>
<form method="post" action="func3.php">
<div class="row register-form">
<div class="col-md-6">
<div class="form-group">
<input type="text" class="form-control" placeholder="User Name *" name="username1" onkeydown="return alphaOnly(event);" required/>
</div>


</div>
<div class="col-md-6">
<div class="form-group">
<input type="password" class="form-control" placeholder="Password *" name="password2" required/>
</div>

<input type="submit" class="btnRegister" name="adsub" value="Login"/>
</div>
</div>
</form>
</div>
</div>

</div>
</div>
```

```
</div>
</body>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js" integrity="sha384-aJ21OjlMXNL5UyIl/XNwTMqvzeRMZH2w8c5cRVpzpU8Y5bApTppSuUkhZXN0VxHd" crossorigin="anonymous"></script>
</html>
```

index1.php

```
<?php
include("header.php");
?>
<!DOCTYPE html>
<html lang="en">
 <head>
 <!-- Required meta tags -->
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

```
<!-- Bootstrap CSS -->
<link href="https://fonts.googleapis.com/css?family=IBM+Plex+Sans&display=swap" rel="stylesheet">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M" crossorigin="anonymous">
<link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
<link href="font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="style2.css">
</head>
<style type="text/css">
#inputbtn:hover{cursor:pointer;}
.card{
background: #f8f9fa;
border-top-left-radius: 5% 5%;
border-bottom-left-radius: 5% 5%;
border-top-right-radius: 5% 5%;
border-bottom-right-radius: 5% 5%;
}

</style>
<body style="background: -webkit-linear-gradient(left, #3931af, #00c6ff); background-size: cover;">
<nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav" >
<div class="container">

<a class="navbar-brand js-scroll-trigger" href="index.php" style="margin-top: 10px;margin-left:-65px;font-family: 'IBM Plex Sans', sans-serif;"><h4><i class="fa fa-user-plus" aria-hidden="true"></i>  Hospital Management System</h4></a>
```

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">

</button>

<ul class="navbar-nav ml-auto">
<li class="nav-item" style="margin-right: 40px;">
<h6>HOME</h6>

<li class="nav-item" style="margin-right: 40px;">
<h6>ABOUT US</h6>

<li class="nav-item">
<h6>CONTACT</h6>

</nav>

<div style="-webkit-animation: mover 2s infinite alternate;
animation: mover 1s infinite alternate;">
<img src="images/ambulance1.png" alt="" style="width: 20%;padding-left: 40px;margin-top: 150px;margin-left: 45px;margin-bottom:15px">

```
<div style="color: white;">
<h4 style="font-family: 'IBM Plex Sans', sans-serif;"> We are here for you!</h4>
</div>
</div>
<div class="col-md-4" style="margin-top: 5%;right: 8%">
<div class="card" style="font-family: 'IBM Plex Sans', sans-serif;">
<div class="card-body">
<center>
<i class="fa fa-hospital-o fa-3x" aria-hidden="true" style="color:#0062cc"></i>
<br>
<h3 style="margin-top: 10%">Patient Login</h3><br>
<form class="form-group" method="POST" action="func.php">
<div class="row" style="margin-top: 10%">
<div class="col-md-4"><label>Email-ID: </label></div>
<div class="col-md-8"><input type="text" name="email" class="form-control" placeholder="enter email ID" required/></div><br><br>
<div class="col-md-4" style="margin-top: 8%"><label>Password: </label></div>
<div class="col-md-8" style="margin-top: 8%"><input type="password" class="form-control" name="password2" placeholder="enter password" required/></div><br><br><br>
</div>
<div class="row">
<div class="col-md-4" style="padding-left: 160px;margin-top: 10%">
<center><input type="submit" id="inputbtn" name="patsub" value="Login" class="btn btn-primary"></center></div>
<!-- <div class="col-md-8" style="margin-top: 10%">
<a href="index.php" class="btn btn-primary">Back</a></div> -->
</div>
</form>
</center>
</div>
</div>
</div>
</div>
</div>
```

```
 <!-- Optional JavaScript -->
 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.11.0/umd/popper.min.js" integrity="sha384-b/U6ypiBEHpOf/4+1nzFpr53nxSS+GLCkfwBdFNTxtclqqenISfwAzpKaMNFNmj4" crossorigin="anonymous"></script>
 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/js/bootstrap.min.js" integrity="sha384-h0AbiXch4ZDo7tp9hKZ4TsHbi047NrKGLO3SEJAg45jXxnGIfYzk4Si90RDIqNm1" crossorigin="anonymous"></script>
 </body>
</html>
```

index2.php

```
<?php
include("header.php");
?>
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <meta name="description" content="">
 <meta name="author" content="">
 <title>DOCTOR - Responsive HTML &amp;amp; Bootstrap Template</title>
 <link rel="stylesheet" href="doctor/css/font-awesome.min.css">
 <link rel="stylesheet" href="doctor/css/bootstrap.min.css">
 <link rel="stylesheet" href="doctor/css/style.css">
```

```
 <link     href='http://fonts.googleapis.com/css?family=Open+Sans:600italic,400,800,700,300'
rel='stylesheet' type='text/css'>
 <link  href='http://fonts.googleapis.com/css?family=BenchNine:300,400,700'  rel='stylesheet'
type='text/css'>
 <script src="doctor/js/modernizr.js"></script>
 <!--[if lt IE 9]>
 <script src="js/html5shiv.js"></script>
 <script src="js/respond.min.js"></script>
 <![endif]-->
<link rel="stylesheet" href="doctor/js/jquery-ui.css">

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




 <!-- CUSTOM SCRIPTS -->
 <script src="doctor/assets/js/custom.js"></script>
</head>
<body>
 <?php
 include "config.php";

 if(isset($_POST["sign_up"]))
 {

 $email = $_POST['user_email'];
 $pass =$_POST['user_pass'];
 $usrNm=""; $usrNm=$_POST['user_name'];
 $usemail=""; $usemail=$_POST['user_email'];
 $pswd=""; $pswd=$_POST['user_pass'];

 $gen=""; $gen=$_POST['gender'];
 $cont=""; $cont=$_POST['user_cont'];
```

```
$add=""; $add=$_POST['user_grp'];
$dob=""; $dob=$_POST['dob'];

$today = date("Y-m-d");
$diff = date_diff(date_create($dob), date_create($today));
$age=$diff->format('%y');


$sql1="insert into patient_detail (user_name,email,place,contact, dob, gender,tbl_age,time)
values('$usrNm','$usemail','$add','$cont','$dob','$gen','$age','$dt $tim')";
$sql1=$conn->query($sql1);
$lId= $conn->insert_id;

$sql="insert into tbl_login (tbl_email,tbl_pswd,tbl_contact,type,user_loginid) values
('$usemail','$pass','$cont','1','$lId')";
$sql=$conn->query($sql);

echo "<script>alert('Patient Registration done successfully...');</script>";
echo "<script>window.location='index.php';</script>";
//header('Location:index.php');

//exit;
}
if(isset($_POST["login_id"])){
$user=$_POST['username'];
$psawd=$_POST['password'];
$sql = "select * from tbl_login where tbl_email='$user' And tbl_pswd='$psawd'";
$row = $conn->query($sql);
$nRow = mysqli_num_rows($row);
$udtl = mysqli_fetch_array($row);
if($nRow>=1)
{
session_start();
```

```
</div>
<div class="row">

document.frmbill.textarea.focus();
return false;
}
else if(document.frmbill.paid.value == "")
{
alert("Paid Amount should not be empty..");
document.frmbill.paid.focus();
return false;
}
else if(!document.frmbill.paid.value.match(numericExpression))
{
alert("Paid Amount not valid..");
document.frmbill.paid.focus();
return false;
}
else if(document.frmbill.Dtime.value == "")
{
alert("Discharge Time should not be empty..");
document.frmbill.Dtime.focus();
return false;
}
else if(document.frmbill.Ddate.value == "")
{
alert("Discharge Date should not be empty..");
document.frmbill.Ddate.focus();
return false;
}
else
{
return true;
```

```
 }

}
</script>
```

contact.php

```
<?php
$con=mysqli_connect("localhost","root","","myhmsdb");
if(isset($_POST['btnSubmit']))
{
 $name = $_POST['txtName'];
 $email = $_POST['txtEmail'];
 $contact = $_POST['txtPhone'];
 $message = $_POST['txtMsg'];

 $query="insert into contact(name,email,contact,message)
values('$name','$email','$contact','$message');";
 $result = mysqli_query($con,$query);

 if($result)
 {
 echo '<script type="text/javascript">';
 echo 'alert("Message sent successfully!");';
 echo 'window.location.href = "contact.html";';
 echo '</script>';
 }
}
```

contactus.php

```
<?php
include("header.php");
 <div class="panel-body">
 <span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i class="fa fa-list fa-stack-1x fa-inverse"></i> </span>
 <h4 class="StepTitle" style="margin-top: 5%;"> View Appointments</h4>
 <script>
 function clickDiv(id) {
 document.querySelector(id).click();
 }
 </script>
 <p class="links cl-effect-1">
 <a href="#list-app" onclick="clickDiv('#list-app-list')">
 Appointment List
 </a>
 </p>
 </div>
 </div>
 </div>

 <div class="col-sm-4" style="left: 15%">
 <div class="panel panel-white no-radius text-center">
 <div class="panel-body">
 <span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i class="fa fa-list-ul fa-stack-1x fa-inverse"></i> </span>
 <h4 class="StepTitle" style="margin-top: 5%;"> Prescriptions</h4>

 <p class="links cl-effect-1">
 <a href="#list-pres" onclick="clickDiv('#list-pres-list')">
 Prescription List
 </a>
 </p>
```

```
</div>
</div>
</div>

</div>
</div>
</div>


<div class="tab-pane fade" id="list-app" role="tabpanel" aria-labelledby="list-home-list">

<table class="table table-hover">
<thead>
<tr>
<th scope="col">Patient ID</th>
<th scope="col">Appointment ID</th>
<th scope="col">First Name</th>
<th scope="col">Last Name</th>
<th scope="col">Gender</th>
<th scope="col">Email</th>
<th scope="col">Contact</th>
<th scope="col">Appointment Date</th>
<th scope="col">Appointment Time</th>
<th scope="col">Current Status</th>
<th scope="col">Action</th>
<th scope="col">Prescribe</th>

</tr>
</thead>
<tbody>
<?php
$con=mysqli_connect("localhost","root","","myhmsdb");
global $con;
$dname = $_SESSION['dname'];
```

```
$query = "select pid,ID,fname,lname,gender,email,contact,appdate,apptime,userStatus,doctorStatus from appointmenttb where doctor='$dname';";
<td><input type="password" name="cnfirmpassword" id="cnfirmpassword" value="<?php echo $rsedit[password]; ?>"/></td>
</tr>
<tr>
<td>Education</td>
<td><input type="text" name="education" id="education" value="<?php echo $rsedit[education]; ?>" /></td>
</tr>
<tr>
<td>Experience</td>
<td><input type="text" name="experience" id="experience" value="<?php echo $rsedit[experience]; ?>"/></td>
</tr>
<tr>
<td>Consultancy Charge</td>
<td><input type="text" name="consultancy_charge" id="consultancy_charge" value="<?php echo $rsedit[experience]; ?>"/></td>
</tr>
<tr>
<td>Status</td>
<td><select name="select" id="select">
<option value="">Select</option>
<?php
$arr= array("Active","Inactive");
foreach($arr as $val)
{
if($val == $rsedit[status])
{
echo "<option value='$val' selected>$val</option>";
}
else
```

```
{
echo "<option value='$val'>$val</option>";
}
}
?>
</select></td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit" name="submit" id="submit" value="Submit" /></td>
</tr>
</tbody>
</table>
</form>
<p> </p>
</div>
</div>
</div>
<div class="clear"></div>
</div>
</div>
<?php
include("footers.php");
?>
<script type="application/javascript">
var alphaExp = /^[a-zA-Z]+$/; //Variable to validate only alphabets
var alphaspaceExp = /^[a-zA-Z\s]+$/; //Variable to validate only alphabets and space
var numericExpression = /^[0-9]+$/; //Variable to validate only numbers
var alphanumericExp = /^[0-9a-zA-Z]+$/; //Variable to validate numbers and alphabets
var emailExp = /^[\w\-\.\+]+\@[a-zA-Z0-9\.\-]+\.[a-zA-z0-9]{2,4}$/; //Variable to validate Email ID

function validateform()
{
```

```
if(document.frmdoct.doctorname.value == "")
{
alert("Doctor name should not be empty..");
{
$lastinsid =$_SESSION[patientid];
}
else
{
$dt = date("Y-m-d");
$tim = date("H:i:s");
$sql ="INSERT INTO patient(patientname,admissiondate,admissiontime,address,city,mobileno,loginid,password,gender,dob,status)
values('$_POST[patiente]','$dt','$tim','$_POST[textarea]','$_POST[city]','$_POST[mobileno]','$_POST[loginid]','$_POST[password]','$_POST[select6]','$_POST[dob]','Pending')";
if($qsql = mysqli_query($con,$sql))
{
/* echo "<script>alert('patient record inserted successfully...');</script>"; */
}
else
{
echo mysqli_error($con);
}
$lastinsid = mysqli_insert_id($con);
}

$sqlappointment="SELECT * FROM appointment WHERE appointmentdate='$_POST[appointmentdate]' AND appointmenttime='$_POST[appointmenttime]' AND doctorid='$_POST[doct]' AND status='Approved'";
$qsqlappointment = mysqli_query($con,$sqlappointment);
if(mysqli_num_rows($qsqlappointment) >= 1)
{
echo "<script>alert('Appointment already scheduled for this time..');</script>";
```

```
 }
 else
 {
 $sql ="INSERT INTO appointment(appointmenttype,patientid,appointmentdate,appointmenttime,app_reason,status,departmentid,doctorid)
values('ONLINE','$lastinsid','$_POST[appointmentdate]','$_POST[appointmenttime]','$_POST[app_reason]','Pending','$_POST[department]','$_POST[doct]')";
 if($qsql = mysqli_query($con,$sql))
 {
 echo "<script>alert('Appointment record inserted successfully...');</script>";
 }
 else
 {
 echo mysqli_error($con);
 }
 }
}
if(isset($_GET[editid]))
{
 $sql="SELECT * FROM appointment WHERE appointmentid='$_GET[editid]' ";
 $qsql = mysqli_query($con,$sql);
 $rsedit = mysqli_fetch_array($qsql);

}
if(isset($_SESSION[patientid]))
{
$sqlpatient = "SELECT * FROM patient WHERE patientid='$_SESSION[patientid]' ";
$qsqlpatient = mysqli_query($con,$sqlpatient);
$rspatient = mysqli_fetch_array($qsqlpatient);
$readonly = " readonly";
}
?>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

```
<div class="wrapper col2">
 <div id="breadcrumb">
 <ul>
 <li class="first">Add New Appointment</li></ul>
 </div>
</div>
<div class="wrapper col4">
 <div id="container">

 <?php
if(isset($_POST[submit]))
{
 if(mysqli_num_rows($qsqlappointment) >= 1)
 {
 echo "<h2>Appointment already scheduled for ". date("d-M-Y", strtotime($_POST[appointmentdate])) . " " . date("H:i A", strtotime($_POST[appointmenttime])) . " .. </h2>";
 }
 else
 {
 if(isset($_SESSION[patientid]))
 {
 echo "<h2>Appointment taken successfully.. </h2>";
 echo "<p>Appointment record is in pending process. Kinldy check the appointment status. </p>";
 echo "<p> <a href='viewappointment.php'>View Appointment record</a>. </p>";
 }
 else
 {
 echo "<h2>Appointment taken successfully.. </h2>";
 echo "<p>Appointment record is in pending process. Please wait for confirmation message.. </p>";
 echo "<p> <a href='patientlogin.php'>Click here to Login</a>. </p>";
 }
```

```
 }
}
else
{
 ?>
 <form method="post" action="" name="frmpatapp" onSubmit="return validateform()">
 <table width="532" border="3">
 <tbody>
 <tr>
 <td width="34%">Patient Name</td>
 <td width="66%"><input type="text" name="patiente" id="patiente" value="<?php echo $rspatient[patientname]; ?>" <?php echo $readonly; ?> ></td>
 </tr>
 <tr>
 <td height="62">Address</td>
 <td><textarea name="textarea" id="textarea" <?php echo $readonly; ?>><?php echo $rspatient[address]; ?></textarea></td>
 </tr>
 <tr>
 <td>City</td>
 <td><input type="text" name="city" id="city" value="<?php echo $rspatient[city]; ?>" <?php echo $readonly; ?> ></td>
 </tr>
 <tr>
 <td>Mobile Number</td>
 <td><input type="text" name="mobileno" id="mobileno" value="<?php echo $rspatient[mobileno]; ?>" <?php echo $readonly; ?> ></td>

 </tr>
<?php
 if(!isset($_SESSION[patientid]))
 {
?>
 <tr>
```

```
<td>Login-ID</td>
<td><input type="text" name="loginid" id="loginid" value="<?php echo $rspatient[loginid];
?>" <?php echo $readonly; ?> ></td>
</tr>

<tr>
<td>Password</td>
<td><input type="password" name="password" id="password" value="<?php echo
$rspatient[patientname]; ?>" <?php echo $readonly; ?> ></td>
</tr>
<?php
}
?>
<tr>
<td>Gender</td>
<td>
<?php
if(isset($_SESSION[patientid]))
{
echo $rspatient[gender];
}
else
{
?>
<select name="select6" id="select6">
<option value="">Select</option>
<?php
$arr = array("Male","Female");
foreach($arr as $val)
{
echo "<option value='$val'>$val</option>";
}
?>
</select>
```

```
<?php
}
?>
</td>

</tr>
<tr>
<td>DOB</td>
<td><input type="date" name="dob" id="dob" value="<?php echo $rspatient[dob]; ?>" <?php echo $readonly; ?> ></td>
</tr>
<tr>
<td><strong>Enter Appointment Date</strong></td>
<td><input type="date" min="<?php echo date("Y-m-d"); ?>" name="appointmentdate" id="appointmentdate"></td>
</tr>
<tr>
<td><strong>Enter Appointment Time</strong></td>
<td><input type="time" name="appointmenttime" id="appointmenttime"></td>
</tr>
<tr>
<td><strong>Department</strong></td>
<td>
<select name="department" id="department" onchange="loaddoctor(this.value)">
<option value="">Select department</option>
<?php
$sqldept = "SELECT * FROM department WHERE status='Active'";
$qsqldept = mysqli_query($con,$sqldept);
while($rsdept = mysqli_fetch_array($qsqldept))
{
echo "<option value='$rsdept[departmentid]'>$rsdept[departmentname]</option>";
}
?>
</select>
```

```
</td>
</tr>
<tr>
<td><strong>Doctors</strong></td>
<td>
<div id="divdoc">
<select name="doct" id="doct">
<option value="">Select doctor</option>
</select>
</div>
</td>
</tr>
<tr>
<td><strong>Appointment reason</strong></td>
<td><textarea name="app_reason"></textarea></td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit" name="submit" id="submit" value="Submit" /></td>
</tr>
</tbody>
</table>
</form>
<p> </p>
<?php
}
?>
</div>
</div>
</div>
<div class="clear"></div>
</div>
</div>
<?php
```

```
include("footer.php");
?>
<script type="application/javascript">
var alphaExp = /^[a-zA-Z]+$/; //Variable to validate only alphabets
var alphaspaceExp = /^[a-zA-Z\s]+$/; //Variable to validate only alphabets and space
var numericExpression = /^[0-9]+$/; //Variable to validate only numbers
var alphanumericExp = /^[0-9a-zA-Z]+$/; //Variable to validate numbers and alphabets
var emailExp = /^[\w\-\.\+]+\@[a-zA-Z0-9\.\-]+\.[a-zA-z0-9]{2,4}$/; //Variable to validate Email ID

function validateform()
{
 if(document.frmpatapp.patiente.value == "")
 {
 alert("Patient name should not be empty..");
 document.frmpatapp.patiente.focus();
 return false;
 }
 else if(!document.frmpatapp.patiente.value.match(alphaspaceExp))
 {
 alert("Patient name not valid..");
 document.frmpatapp.patiente.focus();
 return false;
 }
 else if(document.frmpatapp.textarea.value == "")
 {
 alert("Address should not be empty..");
 document.frmpatapp.textarea.focus();
 return false;
 }
 else if(document.frmpatapp.city.value == "")
 {
 alert("City should not be empty..");
 document.frmpatapp.city.focus();
```

```
return false;
}
else if(!document.frmpatapp.city.value.match(alphaspaceExp))
{
alert("City name not valid..");
document.frmpatapp.city.focus();
return false;
}
else if(document.frmpatapp.mobileno.value == "")
{
alert("Mobile number should not be empty..");
document.frmpatapp.mobileno.focus();
return false;
}
else if(!document.frmpatapp.mobileno.value.match(numericExpression))
{
alert("Mobile number not valid..");
document.frmpatapp.mobileno.focus();
return false;
}
else if(document.frmpatapp.loginid.value == "")
{
alert("login ID should not be empty..");
document.frmpatapp.loginid.focus();
return false;
}
else if(!document.frmpatapp.loginid.value.match(alphanumericExp))
{
alert("login ID not valid..");
document.frmpatapp.loginid.focus();
return false;
}
else if(document.frmpatapp.password.value == "")
{
```

```
alert("Password should not be empty..");
document.frmpatapp.password.focus();
return false;
}
else if(document.frmpatapp.password.value.length < 8)
{
alert("Password length should be more than 8 characters...");
document.frmpatapp.password.focus();
return false;
}
else if(document.frmpatapp.select6.value == "")
{
alert("Gender should not be empty..");
document.frmpatapp.select6.focus();
return false;
}
else if(document.frmpatapp.dob.value == "")
{
alert("Date Of Birth should not be empty..");
document.frmpatapp.dob.focus();
return false;
}
else if(document.frmpatapp.appointmentdate.value == "")
{
alert("Appointment date should not be empty..");
document.frmpatapp.appointmentdate.focus();
return false;
}
else if(document.frmpatapp.appointmenttime.value == "")
{
alert("Appointment time should not be empty..");
document.frmpatapp.appointmenttime.focus();
return false;
}
```

```
 else
 {
 return true;
 }
}

function loaddoctor(deptid)
{
 if (window.XMLHttpRequest) {
 // code for IE7+, Firefox, Chrome, Opera, Safari
 xmlhttp = new XMLHttpRequest();
 } else {
 // code for IE6, IE5
 xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
 }
 xmlhttp.onreadystatechange = function() {
 if (this.readyState == 4 && this.status == 200) {
 document.getElementById("divdoc").innerHTML = this.responseText;
 }
 };
 xmlhttp.open("GET","departmentDoctor.php?deptid="+deptid,true);
 xmlhttp.send();
}
</script>
```

patientchangepassword.php

```
<?php
session_start();
include("headers.php");
include("dbconnection.php");
if(isset($_POST[submit]))
{
```

```
$sql = "UPDATE patient SET password='$_POST[newpassword]' WHERE password='$_POST[oldpassword]' AND patientid='$_SESSION[patientid]'";
$qsql= mysqli_query($con,$sql);
if(mysqli_affected_rows($con) == 1)
{
echo "<script>alert('Password has been updated successfully..');</script>";
}
else
{
echo "<script>alert('Failed to update password..');</script>";
}
}
?>

<div class="wrapper col2">
 <div id="breadcrumb">
 <ul>
 <li class="first">Add New Change Password</li></ul>
 </div>
</div>
<div class="wrapper col4">
 <div id="container">
 <h1>Add new Change Password record</h1>
 <form method="post" action="" name="frmpatchange" onSubmit="return validateform()">
 <table width="200" border="3">
 <tbody>
 <tr>
 <td width="34%">Old Password</td>
 <td width="66%"><input type="password" name="oldpassword" id="oldpassword" /></td>
 </tr>
 <tr>
 <td>New Password</td>
 <td><input type="password" name="newpassword" id="newpassword" /></td>
 </tr>
```

```
<tr>
<td>Confirm Password</td>
<td><input type="password" name="password" id="password" /></td>
</tr>
<tr>
<td height="36" colspan="2" align="center"><input type="submit" name="submit" id="submit" value="Submit" /></td>
</tr>
</tbody>
</table>
</form>
<p> </p>
</div>
</div>
</div>
<div class="clear"></div>
</div>
</div>
<?php
include("footers.php");
?>
<script type="application/javascript">
function validateform()
{
if(document.frmpatchange.oldpassword.value == "")
{
alert("Old password should not be empty..");
document.frmpatchange.oldpassword.focus();
return false;
}
else if(document.frmpatchange.newpassword.value == "")
{
alert("New Password should not be empty..");
document.frmpatchange.newpassword.focus();
```

```
 return false;
 }
 else if(document.frmpatchange.newpassword.value.length < 8)
 {
 alert("New Password length should be more than 8 characters...");
 document.frmpatchange.newpassword.focus();
 return false;
 }
 else if(document.frmpatchange.newpassword.value != document.frmpatchange.password.value )
 {
 alert(" New Password and confirm password should be equal..");
 document.frmpatchange.password.focus();
 return false;
 }
 else
 {
 return true;
 }
}
</script>
```

patientdetial.php

```
<?php
include("dbconnection.php");
if(isset($_POST[submitpat]))
{
 $sql ="INSERT INTO patient(patientname,admissiondate,admissiontime,address,mobileno,gender,dob) values('$_POST[patientname]','$_POST[admissiondate]','$_POST[admissiontime]','$_POST[address]','$_POST[mobilenumber]','$_POST[select]','$_POST[dateofbirth]')";
 if($qsql = mysqli_query($con,$sql))
 {
```

```
 echo "<script>alert('patients record inserted successfully...');</script>";
 }
 else
 {
 echo mysqli_error($con);
 }
}

if(isset($_GET[editid]))
{
 $sql="SELECT * FROM patient WHERE patientid='$_GET[editid]' ";
 $qsql = mysqli_query($con,$sql);
 $rsedit = mysqli_fetch_array($qsql);

}
?>
<?php
if(!isset($_GET[patientid]))
{
?>
<form method="post" action="" name="frmpatdet" onSubmit="return validateform()">
 <table width="808" border="1">
 <tbody>
 <tr>
 <td width="17%"><strong>Patient Name </strong></td>
 <td width="41%"><input type="text" name="patientname" id="patientname"/></td>
 <td width="16%"><strong>Patient ID</strong></td>
 <td width="26%"><input type="text" name="patientid" id="patientid" /></td>
 </tr>
 <tr>
 <td><strong>Address</strong></td>
 <td align="right"><textarea name="address" id="address" cols="45" rows="5">
</textarea></td>
 <td><strong>Gender</strong></td>
```

```
<td><label for="select"></label>
<select name="select" id="select">
<option value="">Select</option>
<option value="Male">Male</option>
<option value="Female">Female</option>
</select></td>
</tr>
<tr>
<td><strong>Contact Number</strong></td>
<td><input type="text" name="mobilenumber" id="mobilenumber"/></td>
<td><strong>Date Of Birth </strong></td>
<td><input type="date" name="dateofbirth" id="dateofbirth" /></td>
</tr>
<tr>
<td colspan="4" align="center"><input type="submit" name="submitpat" id="submitpat" value="Submit" /></td>
</tr>
</tbody>
</table>
</form>
<?php
}
else
{
$sqlpatient = "SELECT * FROM patient where patientid='$_GET[patientid]'";
$qsqlpatient = mysqli_query($con,$sqlpatient);
$rspatient=mysqli_fetch_array($qsqlpatient);
?>

<table width="808" border="1">
<tbody>
<tr>
<td width="16%"><strong>Patient Name </strong></td>
<td width="34%"> <?php echo $rspatient[patientname]; ?></td>
```

```
<td width="16%"><strong>Patient ID</strong></td>
<td width="34%"> <?php echo $rspatient[patientid]; ?></td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td> <?php echo $rspatient[address]; ?></td>
<td><strong>Gender</strong></td>
<td> <?php echo $rspatient[gender];?></td>
</tr>
<tr>
<td><strong>Contact Number</strong></td>
<td> <?php echo $rspatient[mobileno]; ?></td>
<td><strong>Date Of Birth </strong></td>
<td> <?php echo $rspatient[dob]; ?></td>
</tr>
</tbody>
</table>
<?php
}
?>
<script type="application/javascript">
function validateform()
{
 if(document.frmpatdet.patientname.value == "")
 {
 alert("Patient name should not be empty..");
 document.frmpatdet.patientname.focus();
 return false;
 }
 else if(document.frmpatdet.patientid.value == "")
 {
 alert("Patient ID should not be empty..");
 document.frmpatdet.patientid.focus();
 return false;
```

```
}
else if(document.frmpatdet.admissiondate.value == "")
{
alert("Admission date should not be empty..");
document.frmpatdet.admissiondate.focus();
return false;
}
else if(document.frmpatdet.admissiontime.value == "")
{
alert("Admission time should not be empty..");
document.frmpatdet.admissiontime.focus();
return false;
}
else if(document.frmpatdet.address.value == "")
{
alert("Address should not be empty..");
document.frmpatdet.address.focus();
return false;
}
else if(document.frmpatdet.select.value == "")
{
alert("Gender should not be empty..");
document.frmpatdet.select.focus();
return false;
}
else if(document.frmpatdet.mobilenumber.value == "")
{
alert("Contact number should not be empty..");
document.frmpatdet.mobilenumber.focus();
return false;
}
else if(document.frmpatdet.dateofbirth.value == "")
{
alert("Date Of Birth should not be empty..");
```

```
 document.frmpatdet.dateofbirth.focus();
 return false;
 }

 else
 {
 return true;
 }
}
</script>
```

payment.php

```
<?php
session_start();
include("header.php");
include("dbconnection.php");
if(isset($_POST[submit]))
{
 $sql ="INSERT INTO payment(patientid,appointmentid,paiddate,paidtime,paidamount,status) values('$_GET[patientid]','$_GET[appointmentid]','$_POST[date]','$_POST[time]','$_POST[paidamount]','Active')";
 if($qsql = mysqli_query($con,$sql))
 {
 echo "<script>alert('payment record inserted successfully...');</script>";
 }
 else
 {
 echo mysqli_error($con);
 }

 if($_POST[discountamount] != '0')
 {
```

```
 $sql ="UPDATE billing SET discount=$_POST[discountamount]+ discount, discountreason=CONCAT('$_POST[discountreason] , ', discountreason) WHERE appointmentid='$_GET[appointmentid]'";
 $qsql = mysqli_query($con,$sql);
 echo mysqli_error($con);

 }
}
if(isset($_SESSION[patientid]))
{
 $sql="SELECT * FROM payment WHERE paymentid='$_GET[editid]' ";
 $qsql = mysqli_query($con,$sql);
 $rsedit = mysqli_fetch_array($qsql);

}
$billappointmentid = $_GET[appointmentid];
?>

<div class="wrapper col2">
 <div id="breadcrumb">
 <ul>
 <li class="first">Payment Detail</li></ul>
 </div>
</div>
<div class="wrapper col4">
 <div id="container">
 <form method="post" action="" name="frmpatprfl" onSubmit="return validateform()">
 <table width="515" border="3">
 <thead>
 <tr>
 <th colspan="2"> Add payment details.. </th>
 </tr>
 </thead>
 <tbody>
```

```
 <tr>
 <td width="34%">Paid Date</td>
 <td width="66%"><input type="date" value="<?php echo date("Y-m-d"); ?>" name="date" id="date"></td>
 </tr>
 <tr>
 <td>Paid Time</td>
 <td><input type="text" readonly="readonly" value="<?php echo date("H:i:s"); ?>" name="time" id="time"></td>
 </tr>
 <tr>
 <td>Paid Amount</td>
 <td><input name="paidamount" type="text" id="paidamount" value="0"></td>
 </tr>
 <tr>
 <td colspan="2" align="center"><input type="submit" name="submit" id="submit" value="Submit" /></td>
 </tr>
 </tbody>
 </table>

 <p> </p>
 </form>

<?php
include("viewpaymentreport.php");
?>
<table width="342" border="3">
<thead>
 <tr>
 <td colspan="2" align="center"><a href='patientreport.php?patientid=<?php echo $_GET[patientid]; ?>'><strong>View Patient Report>></strong></a></td>
 </tr>
 </thead>
```

```
 </table>
 <p> </p>
 </div>
</div>
<div class="clear"></div>
 </div>
</div>
<?php
include("footer.php");
?>
```

paymentdischarge.php

```
<?php
session_start();
include("header.php");
include("dbconnection.php");
if(isset($_POST["submitfullamount"]))
{
 $sql ="INSERT INTO payment(patientid,appointmentid,paiddate,paidtime,paidamount,status) values('$_GET[patientid]','$_GET[appointmentid]','$dt','$tim','$_POST[paidamount]','Active')";
 if($qsql = mysqli_query($con,$sql))
 {
 echo "<script>alert('payment record inserted successfully...');</script>";
 }
 else
 {
 echo mysqli_error($con);
 }

 $sql ="UPDATE billing SET discount=$_POST[discountamount]+ discount, discountreason=CONCAT('$_POST[discountreason] , ',
```

```
discountreason),discharge_time='$_POST[dischargetime]',discharge_date='$_POST[discharg
edate]' WHERE appointmentid='$_GET[appointmentid]'";
 $qsql = mysqli_query($con,$sql);
 echo mysqli_error($con);

 echo
"<script>window.location='patientreport.php?patientid=$_GET[patientid]&appointmentid=$
_GET[appointmentid]'</script>";
}
if(isset($_SESSION[patientid]))
{
 $sql="SELECT * FROM payment WHERE paymentid='$_GET[editid]' ";
 $qsql = mysqli_query($con,$sql);
 $rsedit = mysqli_fetch_array($qsql);

}
$billappointmentid = $_GET[appointmentid];
?>

<div class="wrapper col4">
 <div id="container">
<?php
include("viewpaymentreport.php");
?>

<br>
<form method="post" action="">
<table width="515" border="3">
 <thead>
 <tr>
 <th colspan="2">Discharge</th>
 </tr>
 </thead>
 <tbody>
```

```
<tr>
<td>Discharge date</td>
<td><input name="dischargedate" type="text" id="dischargedate" value="<?php echo date("Y-m-d"); ?>" readonly></td>
</tr>
<tr>
<td>Discharge time</td>
<td><input name="dischargetime" type="text" id="dischargetime" value="<?php echo date("h:i:s"); ?>" readonly></td>
</tr>
<tr>
<td>Balance amount</td>
<td><input name="balamt" type="text" id="balamt" value="<?php echo $balanceamt; ?>" readonly onkeyup="calculatepayable()"></td>
</tr>
<tr>
<td>Discount</td>
<td><input name="discountamount" type="text" id="discountamount" value="0" onkeyup="calculatepayable()"></td>
</tr>
<tr>
<td>Payable amount</td>
<td><input name="paidamount" type="text" id="paidamount" value="<?php echo $balanceamt; ?>" readonly></td>
</tr>
<tr>
<td>Discount reason</td>
<td><textarea name="discountreason" id="discountreason"></textarea></td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit" name="submitfullamount" id="submitfullamount" value="Submit" /></td>
</tr>
```

```
 </tbody>
 </table>

</form>

<table width="342" border="3">

<thead>
 <tr>
 <td colspan="2" align="center"><a href='patientreport.php?patientid=<?php echo $_GET[patientid]; ?>&appointmentid=<?php echo $_GET[appointmentid]; ?>'><strong>View Patient Report>></strong></a></td>
 </tr>
 </thead>
 </table>
 <p> </p>
 </div>
</div>
<div class="clear"></div>
 </div>
</div>
<?php

include("footer.php");
?>
<script type="application/javascript">
function calculatepayable()
{
 document.getElementById("paidamount").value = document.getElementById("balamt").value - document.getElementById("discountamount").value
}
</script>
```

search.php

```
<?php
session_start();
$con=mysqli_connect("localhost","root","","myhmsdb");
if(isset($_POST['search_submit'])){
 $contact=$_POST['contact'];
 $docname = $_SESSION['dname'];
 $query="select * from appointmenttb where contact='$contact' and doctor='$docname';";
 $result=mysqli_query($con,$query);
 echo '<!DOCTYPE html>
<html lang="en">
 <head>
 <!-- Required meta tags -->
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

 <!-- Bootstrap CSS -->
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-beta/css/bootstrap.min.css" integrity="sha384-/Y6pD6FV/Vv2HJnA6t+vslU6fwYXjCFtcEpHbNJ0lyAFsXTsjBbfaDjzALeQsN6M" crossorigin="anonymous">
 </head>
 <body style="background-color:#342ac1;color:white;text-align:center;padding-top:50px;">
 <div class="container" style="text-align:left;">
 <center><h3>Search Results</h3></center><br>
 <table class="table table-hover">
 <thead>
 <tr>
 <th>First Name</th>
 <th>Last Name</th>
 <th>Email</th>
 <th>Contact</th>
 <th>Appointment Date</th>
```

```
<th>Appointment Time</th>
</tr>
</thead>
<tbody>
';
while($row=mysqli_fetch_array($result)){
$fname=$row['fname'];
$lname=$row['lname'];
$email=$row['email'];
$contact=$row['contact'];
$appdate=$row['appdate'];
$apptime=$row['apptime'];
echo '<tr>
<td>'.$fname.'</td>
<td>'.$lname.'</td>
<td>'.$email.'</td>
<td>Action</td>
</tr>
<?php
$sql ="SELECT * FROM doctor_timings where doctorid='$_SESSION[doctorid]'";
$qsql = mysqli_query($con,$sql);
while($rs = mysqli_fetch_array($qsql))
{
$sqldoctor = "SELECT * FROM doctor WHERE doctorid='$rs[doctorid]'";
$qsqldoctor = mysqli_query($con,$sqldoctor);
$rsdoctor = mysqli_fetch_array($qsqldoctor);

$sqldoct = "SELECT * FROM doctor_timings WHERE
doctor_timings_id='$rs[doctor_timings_id]'";
$qsqldoct = mysqli_query($con,$sqldoct);
$rsdoct = mysqli_fetch_array($qsqldoct);

echo "<tr>
<td> $rsdoctor[doctorname]</td>
```

```
 <td> $rsdoct[start_time] - $rsdoct[end_time]</td>
 <td> $rs[status]</td>
 <td> <a href='doctortimings.php?editid=$rs[doctor_timings_id]'>Edit</a> | <a href='viewdoctortimings.php?delid=$rs[doctor_timings_id]'>Delete</a> </td>
 </tr>";
 }
 ?>

 </tbody>
 </table>
 <p> </p>
 </div>
</div>
</div>
 <div class="clear"></div>
 </div>
</div>
<?php
include("footers.php");
?>
```

viewpaymentreport.php

```
<?php
session_start();
include("dbconnection.php");
if(isset($_GET[delid]))
{
 $sql ="DELETE FROM billing_records WHERE billingid='$_GET[delid]'";
 $qsql=mysqli_query($con,$sql);
 if(mysqli_affected_rows($con) == 1)
 {
 echo "<script>alert('billing record deleted successfully..');</script>";
 }
```

```
}

?>
 <section class="container">
<?php
$sqlbilling_records ="SELECT * FROM billing WHERE appointmentid='$billappointmentid'";
$qsqlbilling_records = mysqli_query($con,$sqlbilling_records);
$rsbilling_records = mysqli_fetch_array($qsqlbilling_records);
?>
 <table width="1713" border="3">
 <tbody>
 <tr>
 <th scope="col"><div align="right">Bill number   </div></th>
 <td><?php echo $rsbilling_records[billingid]; ?></td>
 <td>Appointment Number  </td>
 <td><?php echo $rsbilling_records[appointmentid]; ?></td>
 </tr>
 <tr>
 <th width="442" scope="col"><div align="right">Billing Date   </div></th>
 <td width="413"><?php echo $rsbilling_records[billingdate]; ?></td>
 <td width="413">Billing time  </td>
 <td width="413"><?php echo $rsbilling_records[billingtime] ; ?></td>
 </tr>

 <tr>
 <th scope="col"><div align="right"></div></th>
 <td></td>
 <th scope="col"><div align="right">Bill Amount   </div></th>
 <td><?php
 $sql ="SELECT * FROM billing_records where billingid='$rsbilling_records[billingid]'";
 $qsql = mysqli_query($con,$sql);
 $billamt= 0;
 while($rs = mysqli_fetch_array($qsql))
```

```
 {
 $billamt = $billamt + $rs[bill_amount];
 }
?>
  Rs. <?php echo $billamt; ?></td>
 </tr>
 <tr>
 <th width="442" scope="col"><div align="right"></div></th>
 <td width="413"> </td>
 <th width="442" scope="col"><div align="right">Tax Amount (5%)   </div></th>
 <td width="413"> Rs. <?php echo $taxamt = 5 * ($billamt / 100); ?></td>
 </tr>

 <tr>
 <th scope="col"><div align="right">Disount reason</div></th>
 <td rowspan="4" valign="top"><?php echo $rsbilling_records[discountreason]; ?></td>
 <th scope="col"><div align="right">Discount   </div></th>
 <td> Rs. <?php echo $rsbilling_records[discount]; ?></td>
 </tr>

 <tr>
 <th rowspan="3" scope="col"> </th>
 <th scope="col"><div align="right">Grand Total   </div></th>
 <td> Rs. <?php echo $grandtotal = ($billamt + $taxamt) -
$rsbilling_records[discount] ; ?></td>
 </tr>
 <tr>
 <th scope="col"><div align="right">Paid Amount </div></th>
 <td>Rs. <?php
 $sqlpayment ="SELECT sum(paidamount) FROM payment where
appointmentid='$billappointmentid'";
 $qsqlpayment = mysqli_query($con,$sqlpayment);
 $rspayment = mysqli_fetch_array($qsqlpayment);
 echo $rspayment[0];
```

```
?></td>
</tr>
<tr>
<th scope="col"><div align="right">Balance Amount</div></th>
<td>Rs. <?php echo $balanceamt = $grandtotal - $rspayment[0] ; ?></td>
</tr>
</tbody>
</table>
<p><strong>Payment report:</strong></p>
<?php
$sqlpayment = "SELECT * FROM payment where appointmentid='$billappointmentid'";
$qsqlpayment = mysqli_query($con,$sqlpayment);
if(mysqli_num_rows($qsqlpayment) == 0)
{
 echo "<strong>No transaction details found..</strong>";
}
else
{
?>
 <table width="840" border="3">
 <tbody>
 <tr>
 <th scope="col">Paid Date</th>
 <th scope="col">Paid time</th>
 <th scope="col">Paid amount</th>
 </tr>
<?php
 while($rspayment = mysqli_fetch_array($qsqlpayment))
 {
 ?>
 <tr>
 <td> <?php echo $rspayment[paiddate]; ?></td>
 <td> <?php echo $rspayment[paidtime]; ?></td>
 <td> Rs. <?php echo $rspayment[paidamount]; ?></td>
```

```
 </tr>
 <?php
 }
?>

 </tbody>
 </table>
<?php
}
?>
 <p><strong></strong></p>
</section>
```

viewtreatmentrecord.php

```
<?php
session_start();
include("dbconnection.php");
if(isset($_GET[delid]))
{
 $sql ="DELETE FROM billing_records WHERE billingid='$_GET[delid]'";
 $qsql=mysqli_query($con,$sql);
 if(mysqli_affected_rows($con) == 1)
 {
 echo "<script>alert('billing record deleted successfully..');</script>";
 }
}

?>
 <section class="container">
<?php
$sqlbilling_records ="SELECT * FROM billing WHERE
appointmentid='$billappointmentid'";
```

```
$qsqlbilling_records = mysqli_query($con,$sqlbilling_records);
$rsbilling_records = mysqli_fetch_array($qsqlbilling_records);
?>
 <table width="1713" border="3">
 <tbody>
 <tr>
 <th scope="col"><div align="right">Bill number   </div></th>
 <td><?php echo $rsbilling_records[billingid]; ?></td>
 <td>Appointment Number  </td>
 <td><?php echo $rsbilling_records[appointmentid]; ?></td>
 </tr>
 <tr>
 <th width="442" scope="col"><div align="right">Billing Date   </div></th>
 <td width="413"><?php echo $rsbilling_records[billingdate]; ?></td>
 <td width="413">Billing time  </td>
 <td width="413"><?php echo $rsbilling_records[billingtime] ; ?></td>
 </tr>

 <tr>
 <th scope="col"><div align="right"></div></th>
 <td></td>
 <th scope="col"><div align="right">Bill Amount   </div></th>
 <td><?php
 $sql ="SELECT * FROM billing_records where billingid='$rsbilling_records[billingid]'";
 $qsql = mysqli_query($con,$sql);
 $billamt= 0;
 while($rs = mysqli_fetch_array($qsql))
 {
 $billamt = $billamt + $rs[bill_amount];
 }
?>
  Rs. <?php echo $billamt; ?></td>
 </tr>
 <tr>
```

```
<th width="442" scope="col"><div align="right"></div></th>
<td width="413"> </td>
<th width="442" scope="col"><div align="right">Tax Amount (5%)   </div></th>
<td width="413"> Rs. <?php echo $taxamt = 5 * ($billamt / 100); ?></td>
</tr>

<tr>
<th scope="col"><div align="right">Disount reason</div></th>
<td rowspan="4" valign="top"><?php echo $rsbilling_records[discountreason]; ?></td>
<th scope="col"><div align="right">Discount   </div></th>
<td> Rs. <?php echo $rsbilling_records[discount]; ?></td>
</tr>

<tr>
<th rowspan="3" scope="col"> </th>
<th scope="col"><div align="right">Grand Total   </div></th>
<td> Rs. <?php echo $grandtotal = ($billamt + $taxamt) -
$rsbilling_records[discount] ; ?></td>
</tr>
<tr>
<th scope="col"><div align="right">Paid Amount </div></th>
<td>Rs. <?php
$sqlpayment ="SELECT sum(paidamount) FROM payment where
appointmentid='$billappointmentid'";
$qsqlpayment = mysqli_query($con,$sqlpayment);
$rspayment = mysqli_fetch_array($qsqlpayment);
echo $rspayment[0];
?></td>
</tr>
<tr>
<th scope="col"><div align="right">Balance Amount</div></th>
<td>Rs. <?php echo $balanceamt = $grandtotal - $rspayment[0] ; ?></td>
</tr>
</tbody>
```

```
 </table>
 <p><strong>Payment report:</strong></p>
<?php
$sqlpayment = "SELECT * FROM payment where appointmentid='$billappointmentid'";
$qsqlpayment = mysqli_query($con,$sqlpayment);
if(mysqli_num_rows($qsqlpayment) == 0)
{
 echo "<strong>No transaction details found..</strong>";
}
else
{
?>
 <table width="840" border="3">
 <tbody>
 <tr>
 <th scope="col">Paid Date</th>
 <th scope="col">Paid time</th>
 <th scope="col">Paid amount</th>
 </tr>
<?php
 while($rspayment = mysqli_fetch_array($qsqlpayment))
 {
 ?>
 <tr>
 <td> <?php echo $rspayment[paiddate]; ?></td>
 <td> <?php echo $rspayment[paidtime]; ?></td>
 <td> Rs. <?php echo $rspayment[paidamount]; ?></td>
 </tr>
 <?php
 }
?>

 </tbody>
 </table>
```

```
<?php
}
?>
 <p><strong></strong></p>
</section>
```

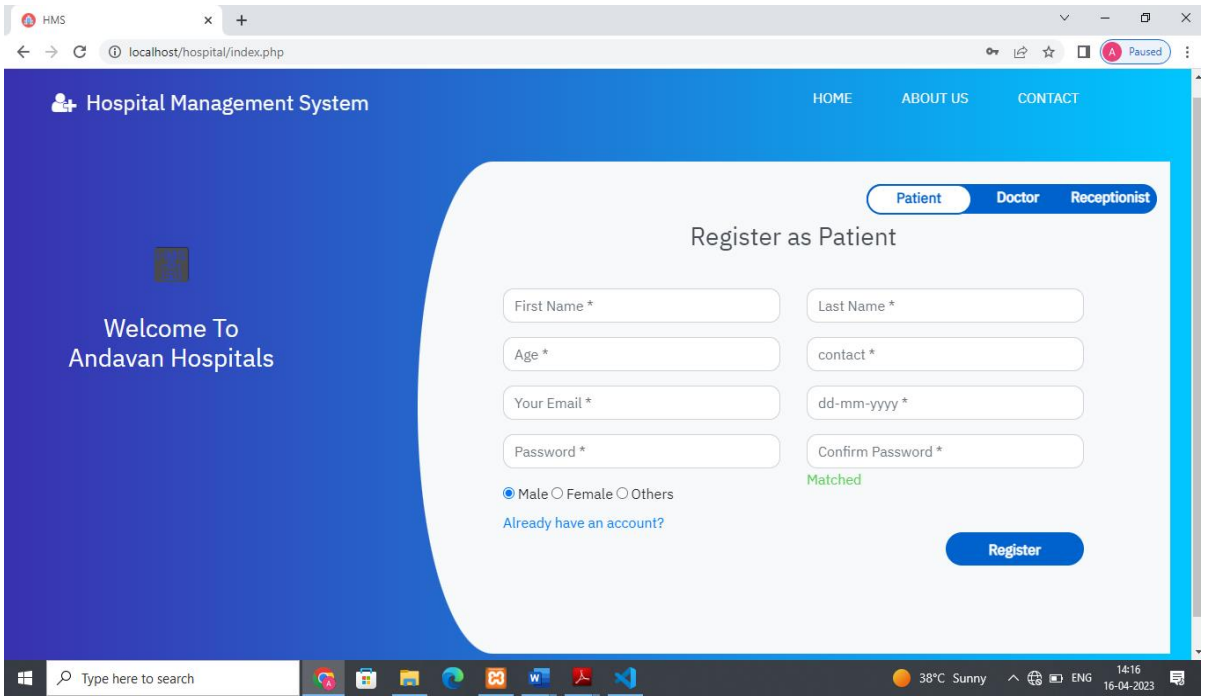
viewservicetype.php

```
<?php
include("header.php");
include("dbconnection.php");
if(isset($_GET[delid]))
{
 $sql ="DELETE FROM service_type WHERE service_type_id='$_GET[delid]'";
 $qsql=mysqli_query($con,$sql);
 if(mysqli_affected_rows($con) == 1)
 {
 echo "<script>alert('Service type deleted successfully..');</script>";
 }
}
?>
<div class="wrapper col2">
 <div id="breadcrumb">
 <ul>
 <li class="first">View service type</li></ul>
 </div>
</div>
<div class="wrapper col4">
 <div id="container">
 <h1>View Service type record</h1>
 <table width="200" border="3">
 <tbody>
 <tr>
 <td>Service Type</td>
```

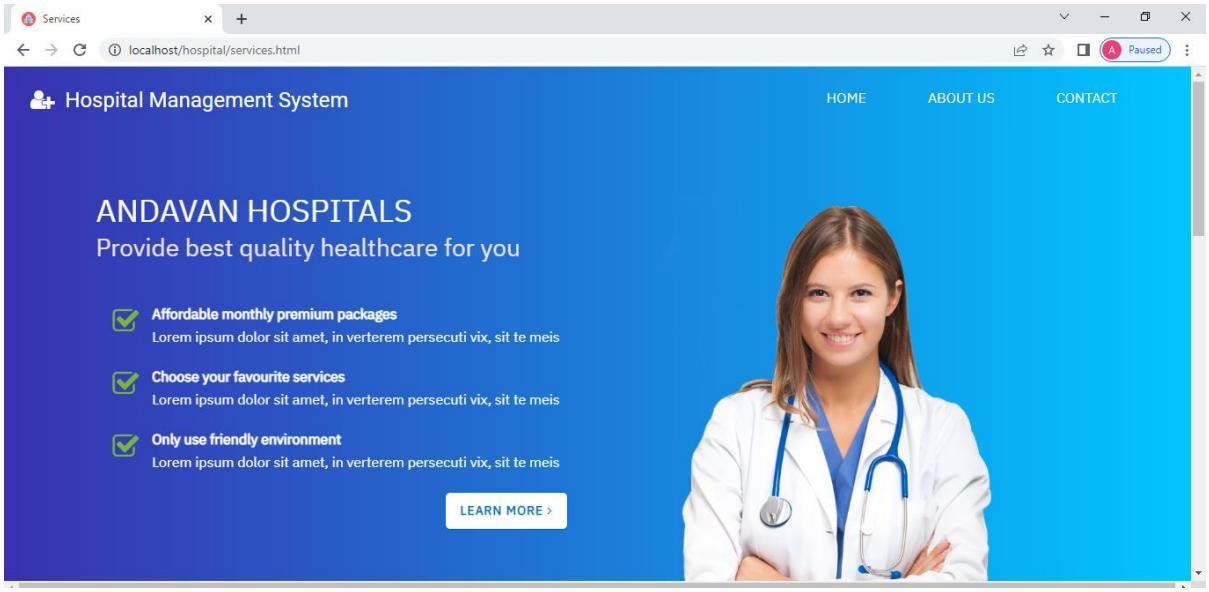
```
<td>Service Charge</td>
<td>Description</td>
<td>Status</td>
<td>Action</td>
</tr>
<?php
$sql ="SELECT * FROM service_type";
$qsql = mysqli_query($con,$sql);
while($rs = mysqli_fetch_array($qsql))
{
echo "<tr>
<td> $rs[service_type]</td>
<td> $rs[servicecharge]</td>
<td> $rs[description]</td>
<td> $rs[status]</td>
<td> 
<a href='servicetype.php?editid=$rs[service_type_id]'>Edit</a> | <a href='viewservicetype.php?delid=$rs[service_type_id]'>Delete</a> </td>
</tr>";
}
?>
</tbody>
</table>
<p> </p>
</div>
</div>
</div>
<div class="clear"></div>
</div>
</div>
<?php
include("footer.php");
?>
```

4.2 SCREENSHOTS

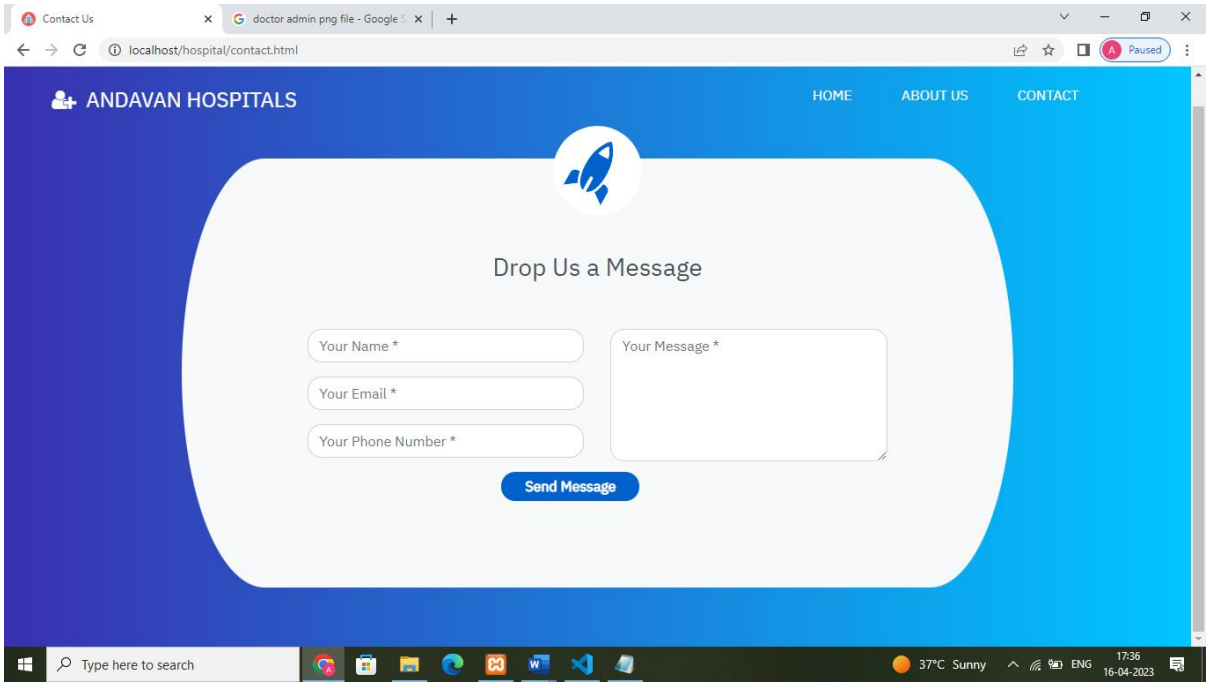
HOME PAGE :



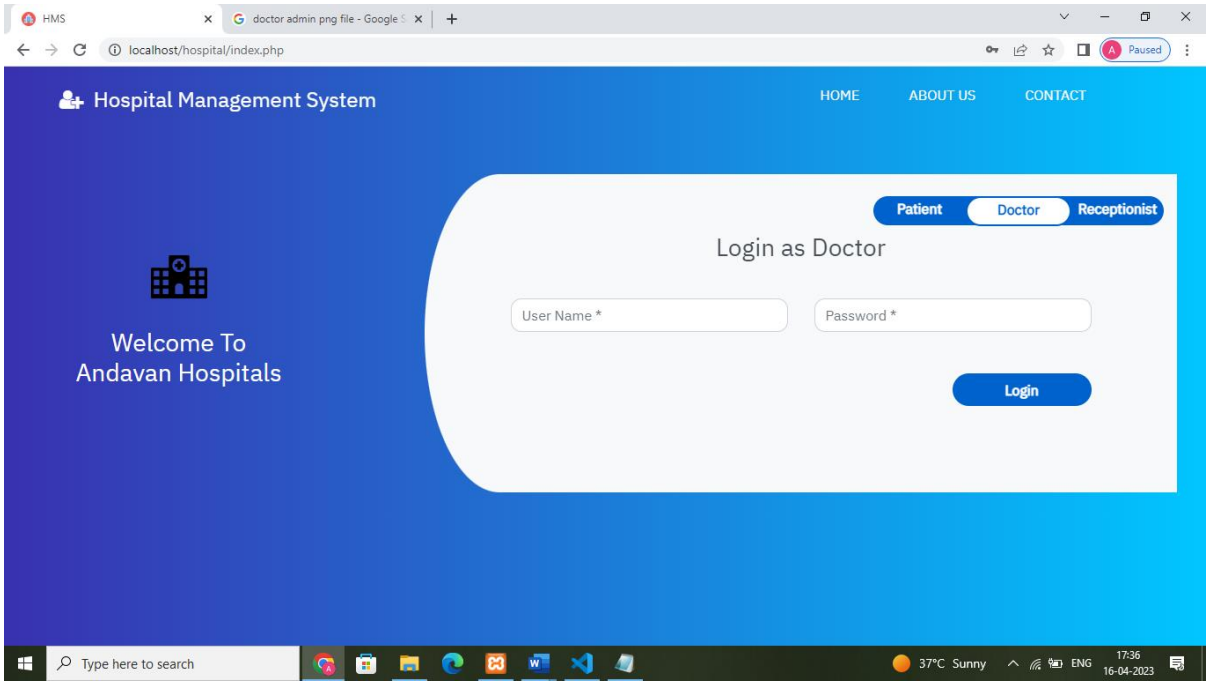
ABOUT US :


Services
localhost/hospital/services.html
Paused
Hospital Management System
HOME
ABOUT US
CONTACT
ANDAVAN HOSPITALS
Provide best quality healthcare for you
Affordable monthly premium packages
Lorem ipsum dolor sit amet, in verterem persecuti vix, sit te meis
Choose your favourite services
Lorem ipsum dolor sit amet, in verterem persecuti vix, sit te meis
Only use friendly environment
Lorem ipsum dolor sit amet, in verterem persecuti vix, sit te meis
LEARN MORE ›

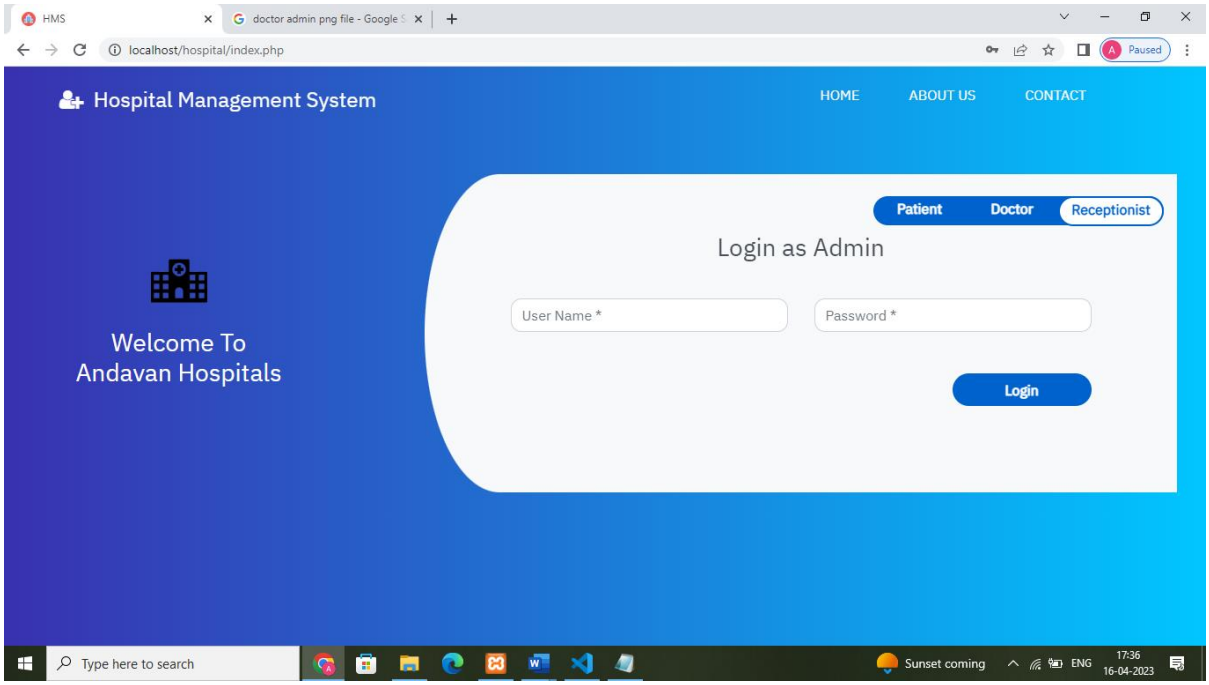
CONTACT :


Contact Us
doctor admin png file - Google
localhost/hospital/contact.html
Paused
ANDAVAN HOSPITALS
HOME
ABOUT US
CONTACT
Drop Us a Message
Your Name *
Your Email *
Your Phone Number *
Your Message *
Send Message
Type here to search
37°C Sunny
ENG
17:36
16-04-2023

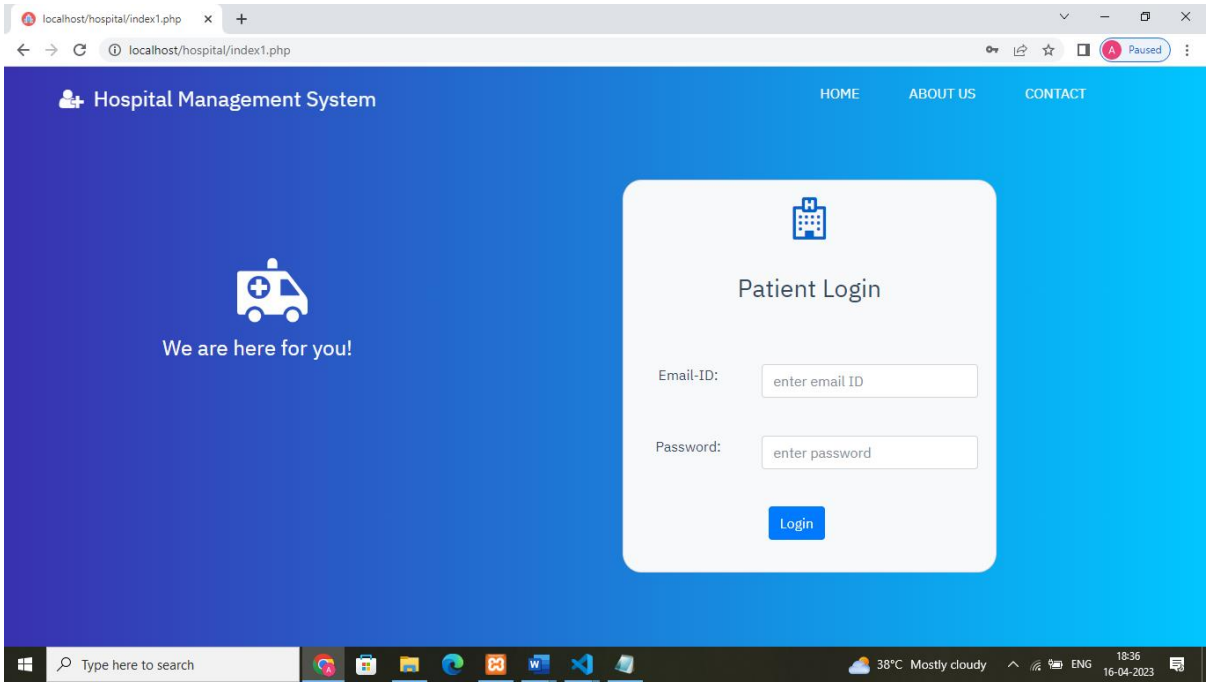
DOCTOR LOGIN PAGE :



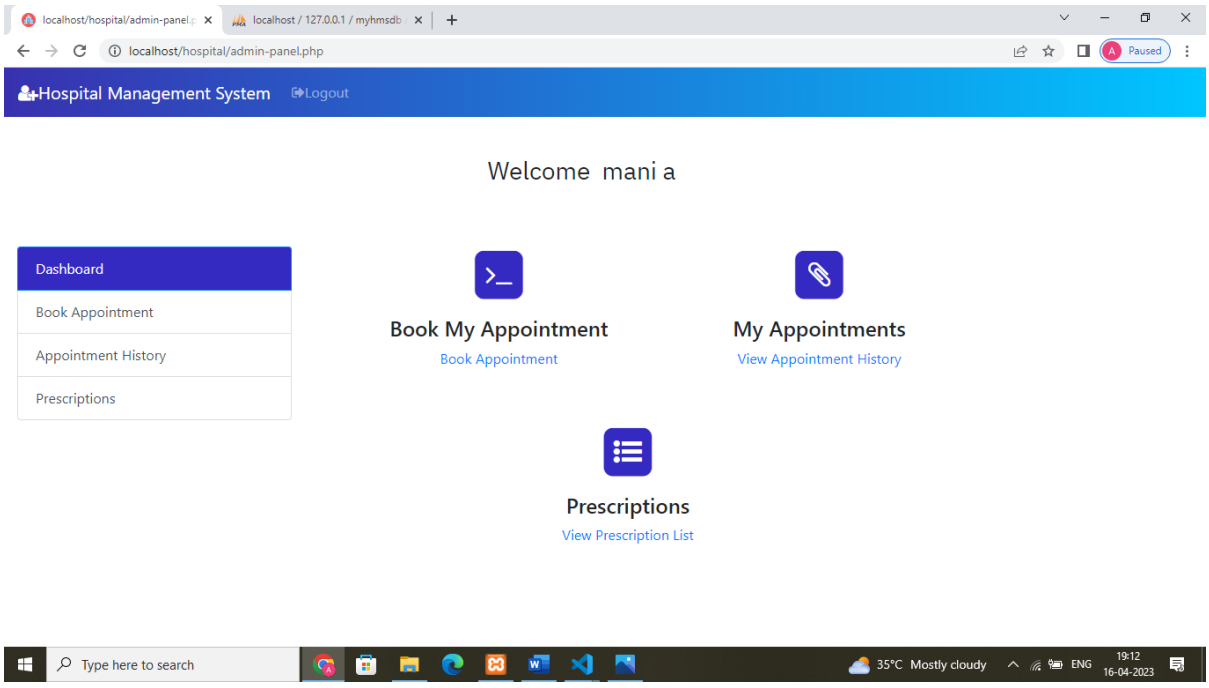
ADMIN LOGIN PAGE :



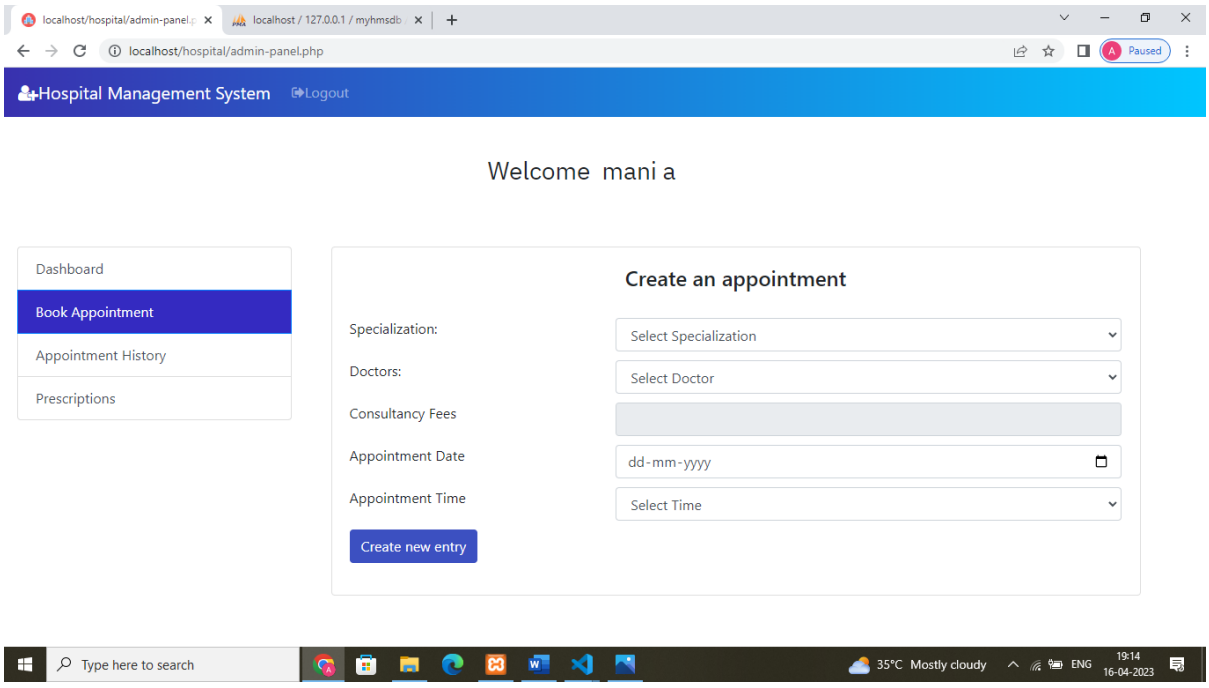
PATIENT LOGIN PAGE :



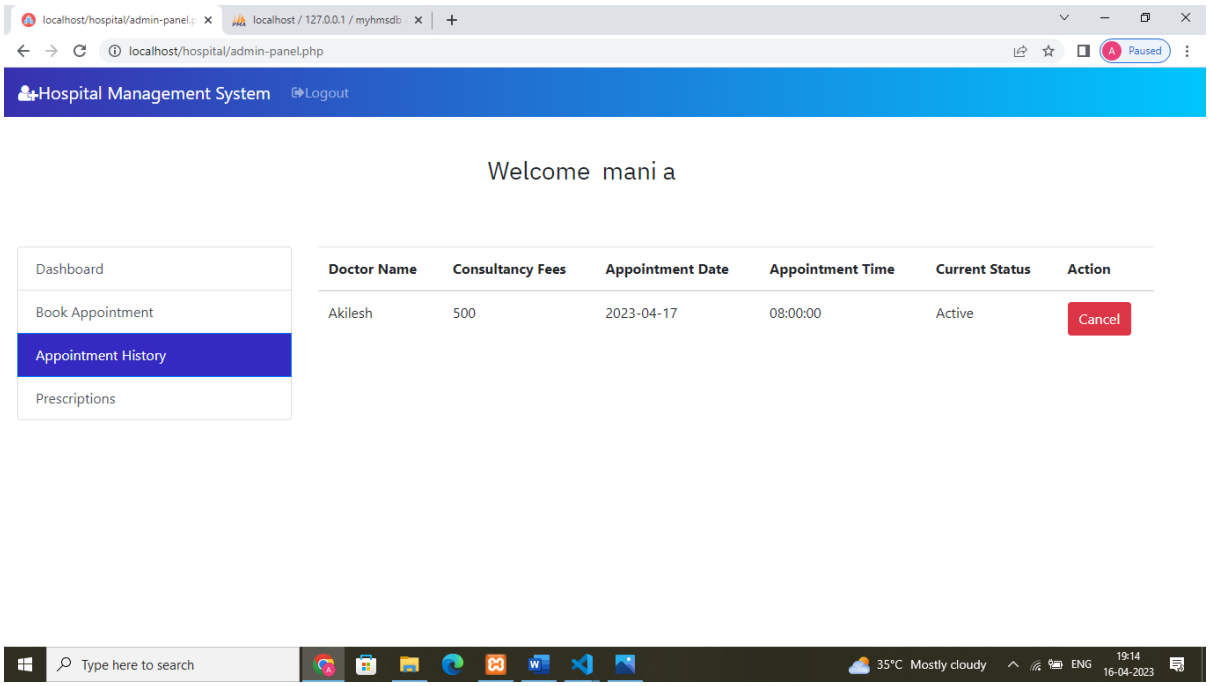
PATIENT PAGE :



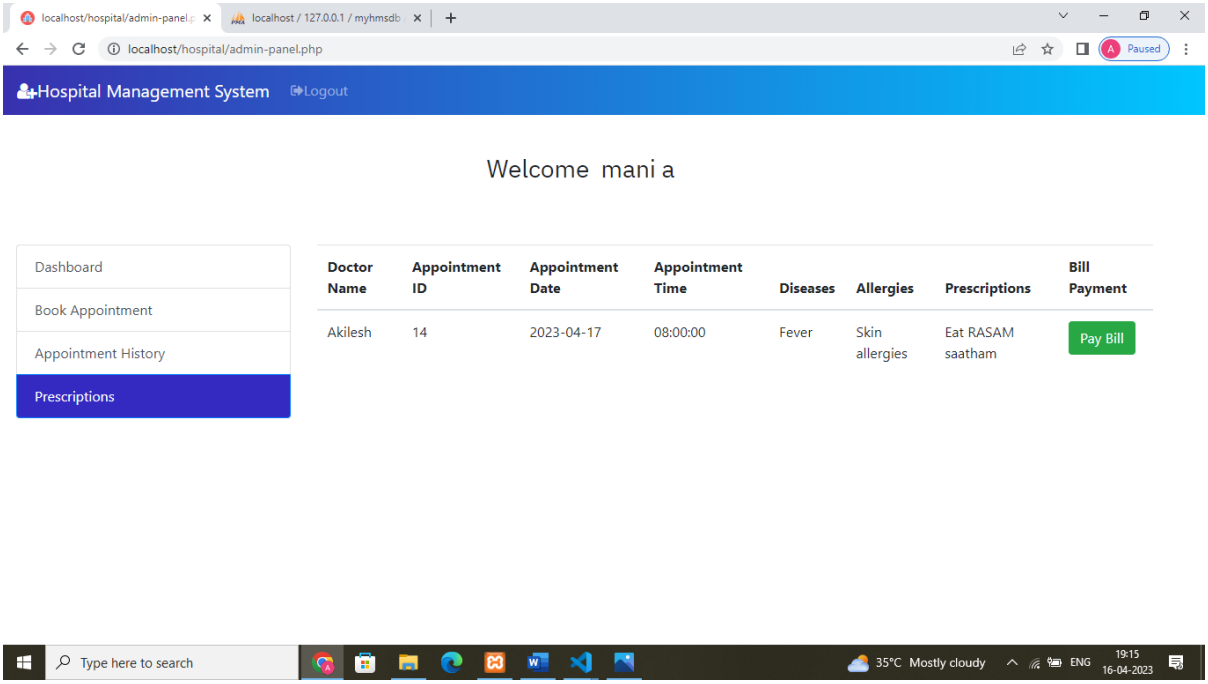
BOOK APPOINTMENT :



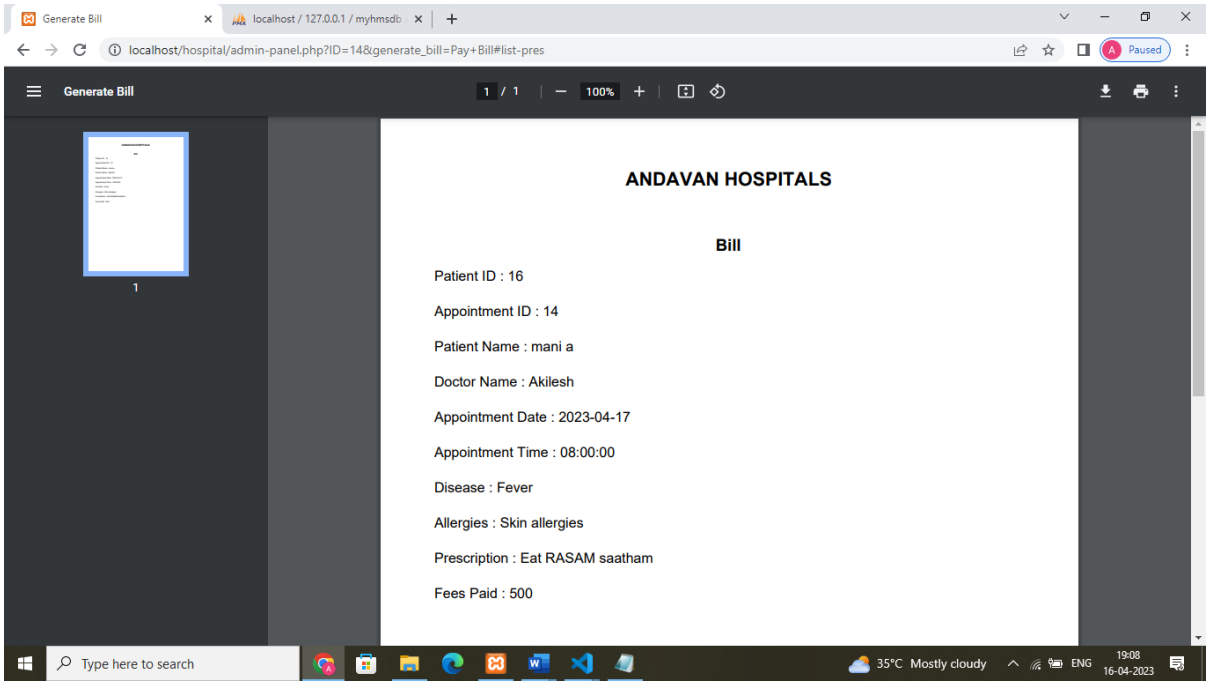
APPOINTMENT HISTORY :



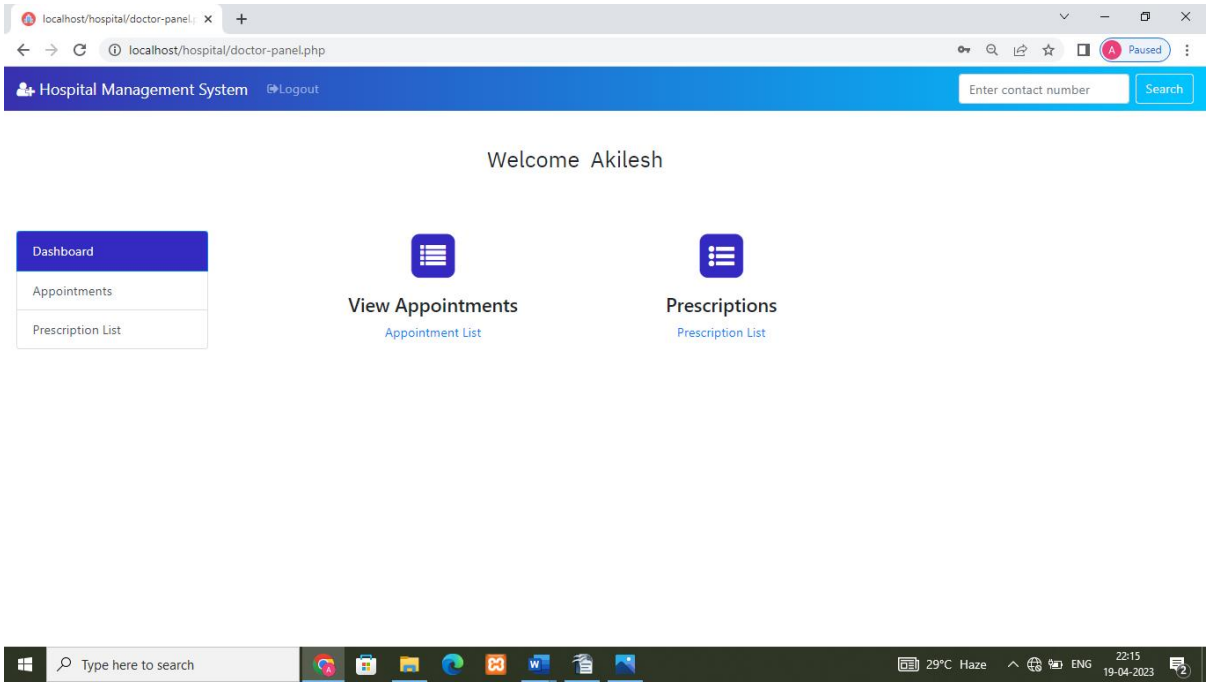
PRESCRIPTIONS :



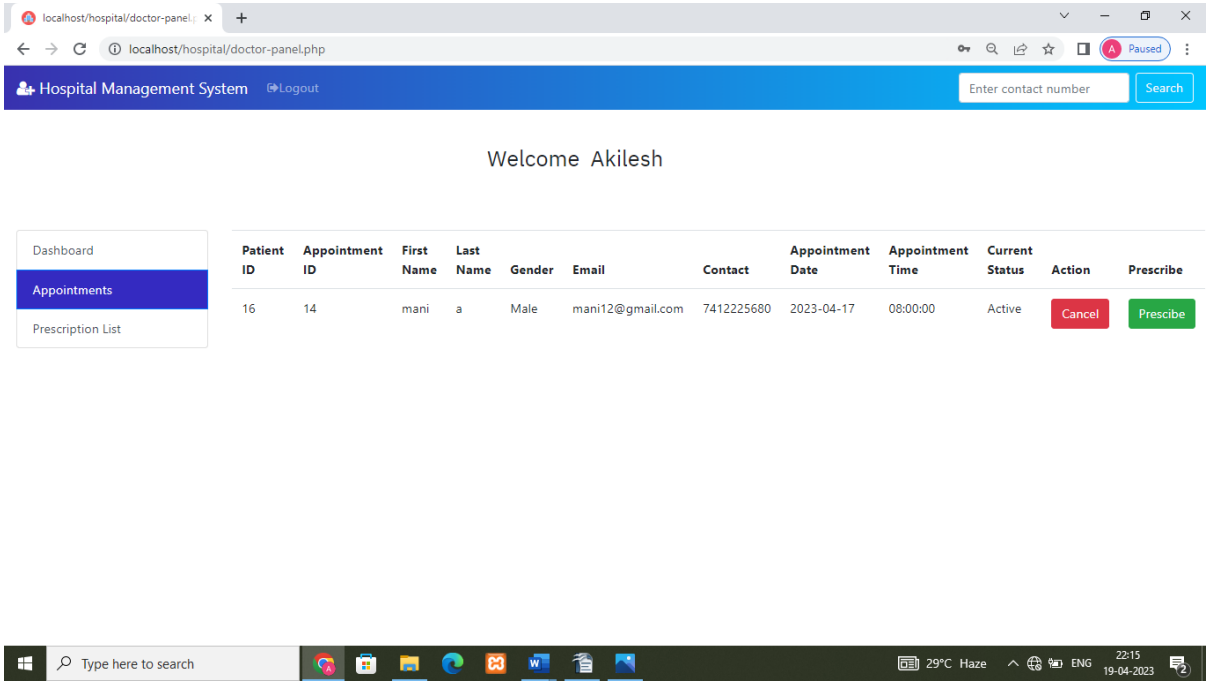
BILL PDF :


Generate Bill
localhost / 127.0.0.1 / myhmsdb
localhost/hospital/admin-panel.php?ID=14&generate_bill=Pay+Bill#list-pres
Paused
Generate Bill
1 / 1
100%
1
ANDAVAN HOSPITALS
Bill
Patient ID : 16
Appointment ID : 14
Patient Name : mani a
Doctor Name : Akilesh
Appointment Date : 2023-04-17
Appointment Time : 08:00:00
Disease : Fever
Allergies : Skin allergies
Prescription : Eat RASAM saatham
Fees Paid : 500
Type here to search
35°C Mostly cloudy
ENG
19:08
16-04-2023

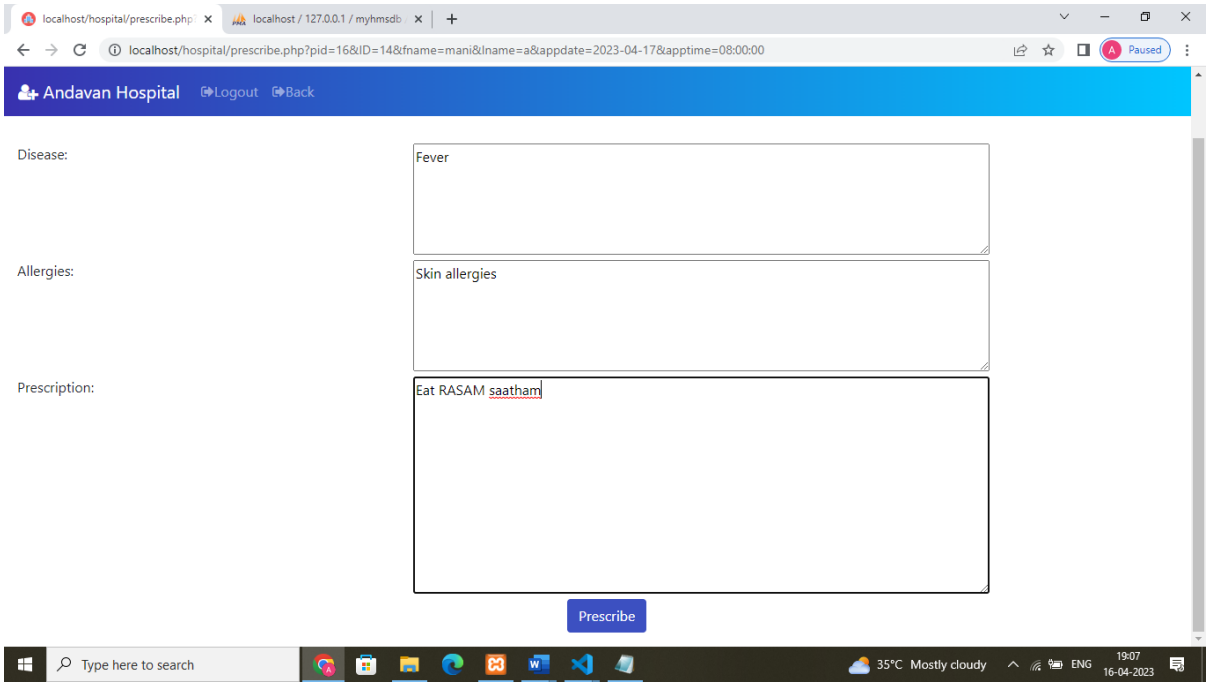
DOCTOR PAGE :


localhost/hospital/doctor-panel.php
Paused
Hospital Management System
Logout
Enter contact number
Search
Welcome Akilesh
Dashboard
Appointments
Prescription List
View Appointments
Appointment List
Prescriptions
Prescription List
Type here to search
29°C Haze
ENG
22:15
19-04-2023

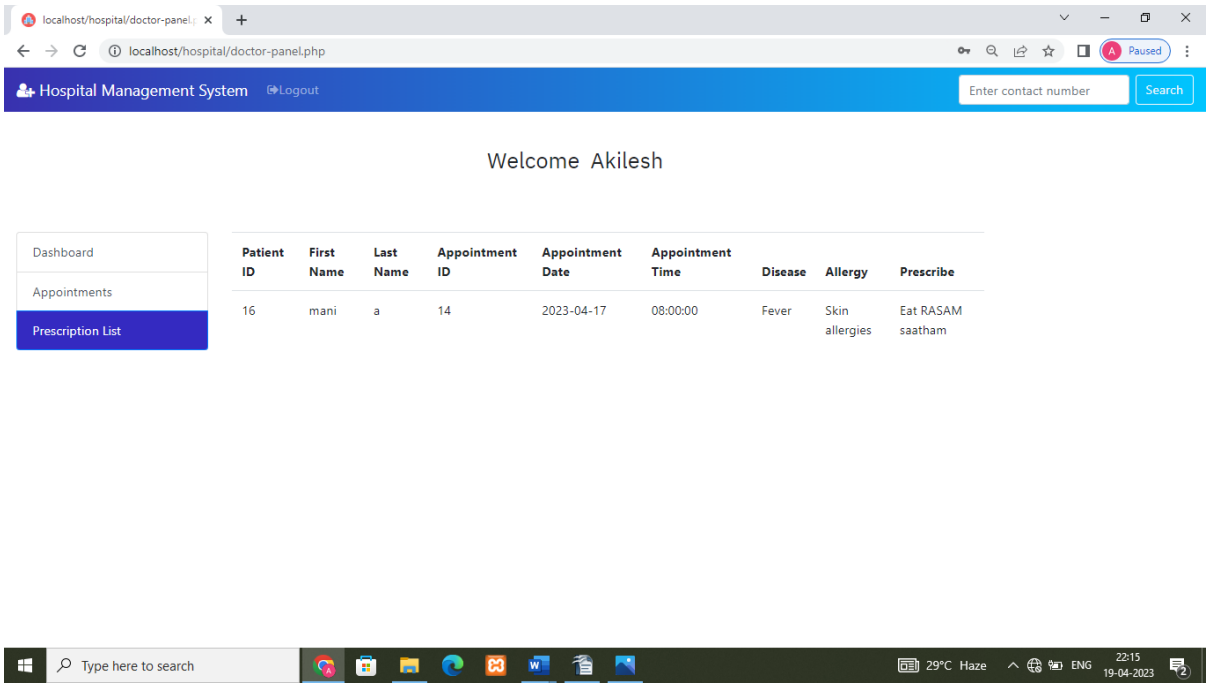
APP0INTMENT DETAILS :



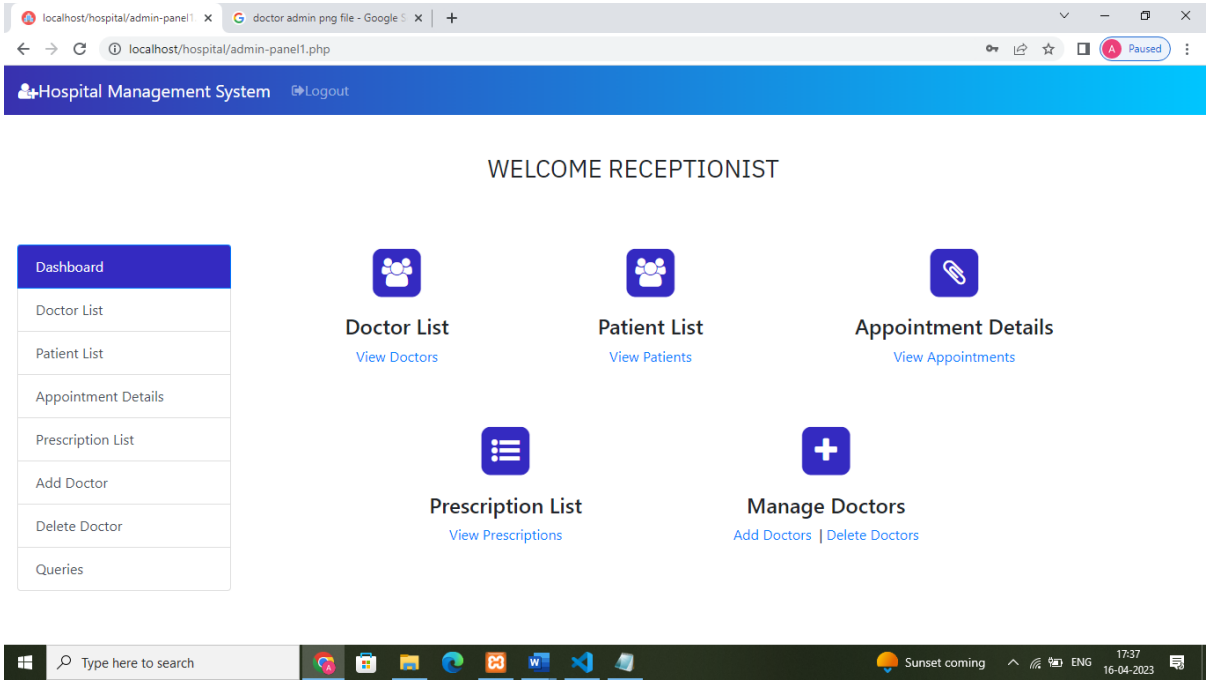
EDIT PRESCRIPTION :



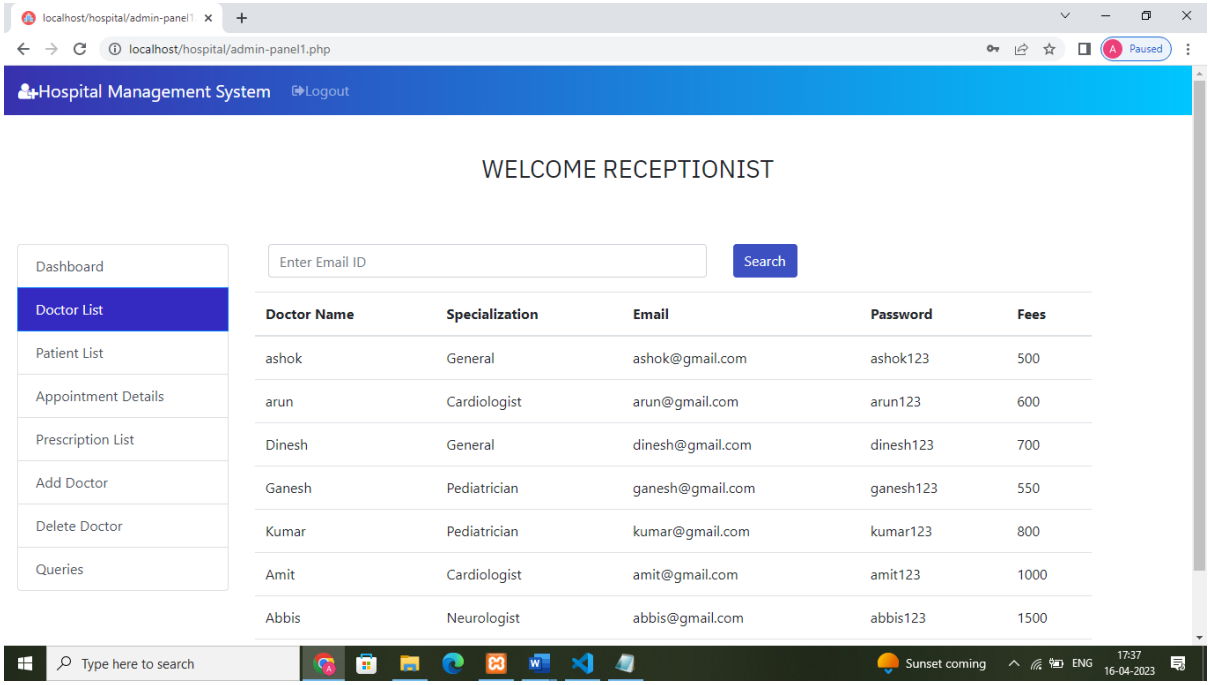
PRESCRIPTION LIST :



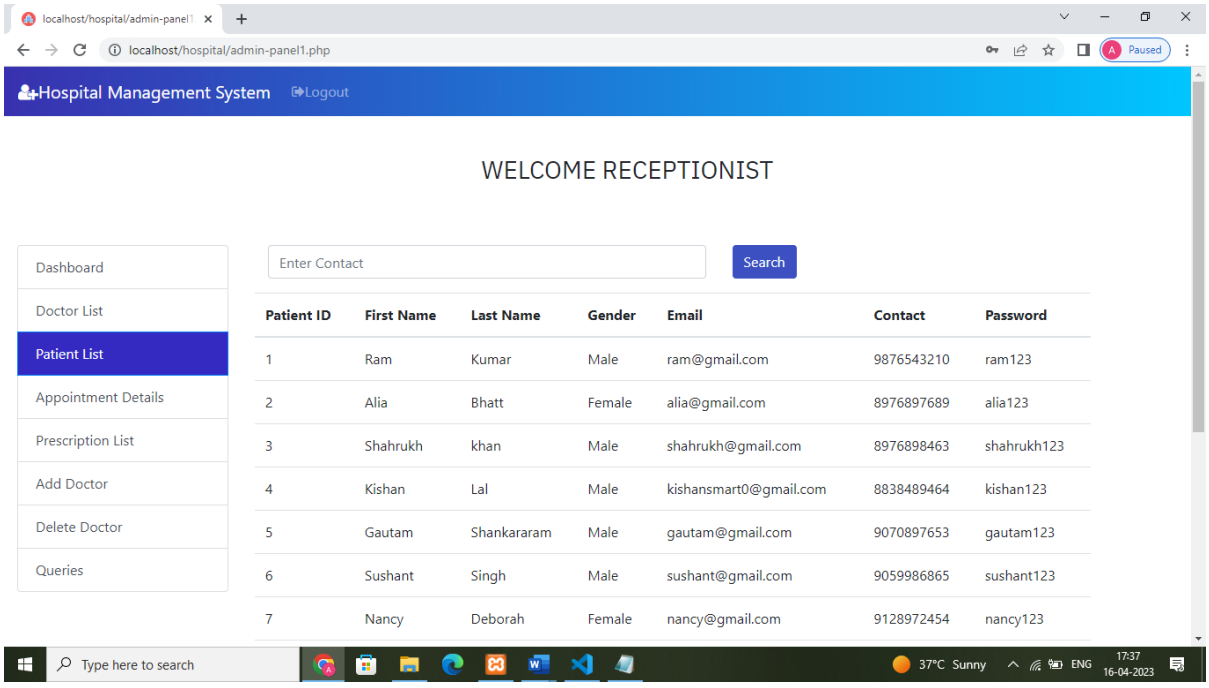
ADMIN PAGE :


localhost/hospital/admin-panel1
doctor admin png file - Google S
localhost/hospital/admin-panel1.php
Paused
Hospital Management System
Logout
WELCOME RECEPTIONIST
Dashboard
Doctor List
Patient List
Appointment Details
Prescription List
Add Doctor
Delete Doctor
Queries
Doctor List
View Doctors
Patient List
View Patients
Appointment Details
View Appointments
Prescription List
View Prescriptions
Manage Doctors
Add Doctors | Delete Doctors
Type here to search
Sunset coming
ENG
17:37
16-04-2023

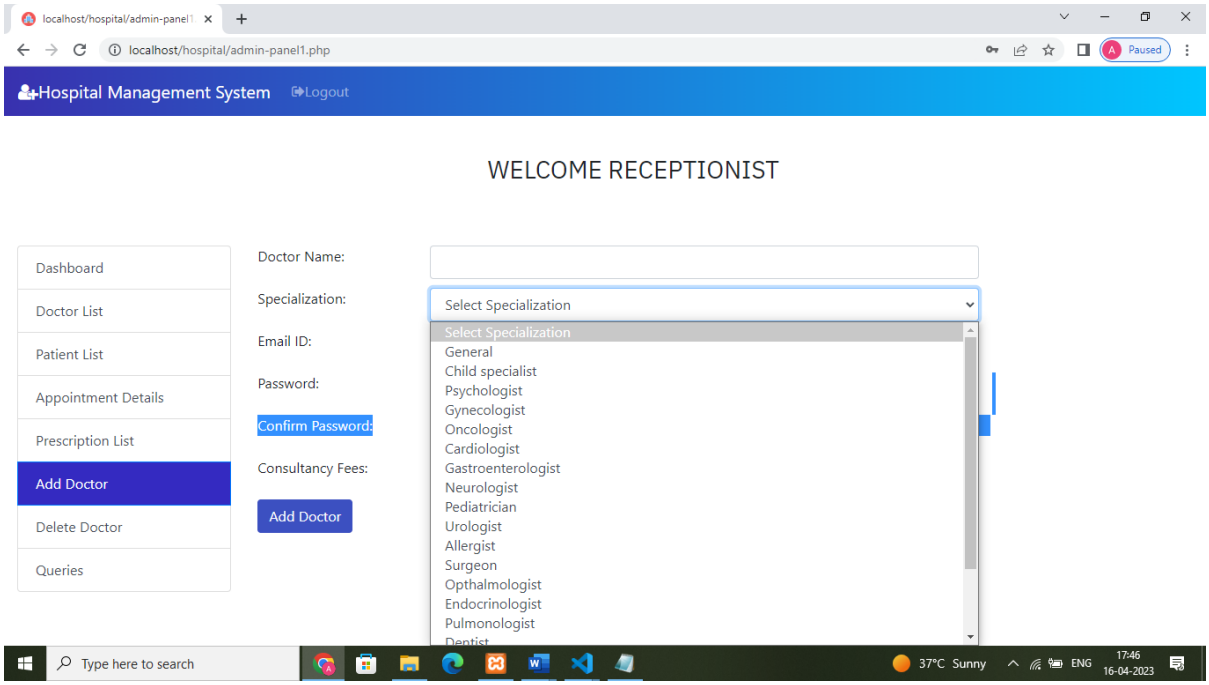
DOCTOR LIST :


localhost/hospital/admin-panel1
localhost/hospital/admin-panel1.php
Paused
Hospital Management System
Logout
WELCOME RECEPTIONIST
Dashboard
Doctor List
Patient List
Appointment Details
Prescription List
Add Doctor
Delete Doctor
Queries
Enter Email ID
Search
Doctor Name
Specialization
Email
Password
Fees
ashok
General
ashok@gmail.com
ashok123
500
arun
Cardiologist
arun@gmail.com
arun123
600
Dinesh
General
dinesh@gmail.com
dinesh123
700
Ganesh
Pediatrician
ganesh@gmail.com
ganesh123
550
Kumar
Pediatrician
kumar@gmail.com
kumar123
800
Amit
Cardiologist
amit@gmail.com
amit123
1000
Abbis
Neurologist
abbis@gmail.com
abbis123
1500
Type here to search
Sunset coming
ENG
17:37
16-04-2023

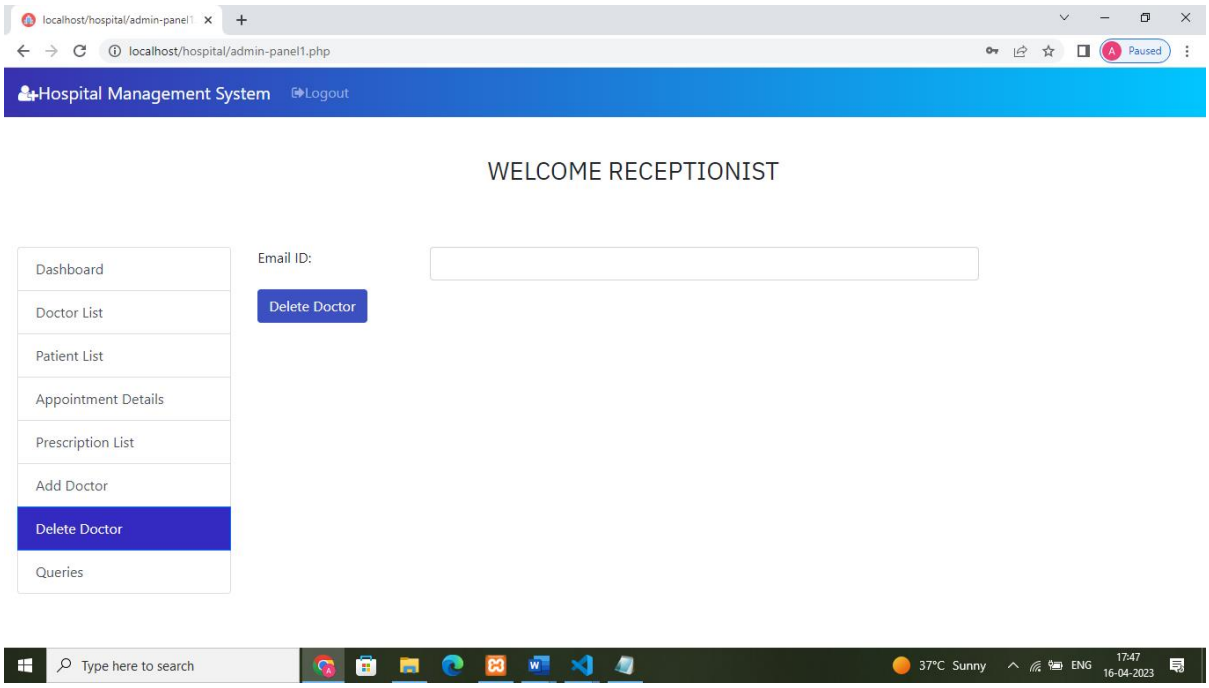
PATIENT LIST :



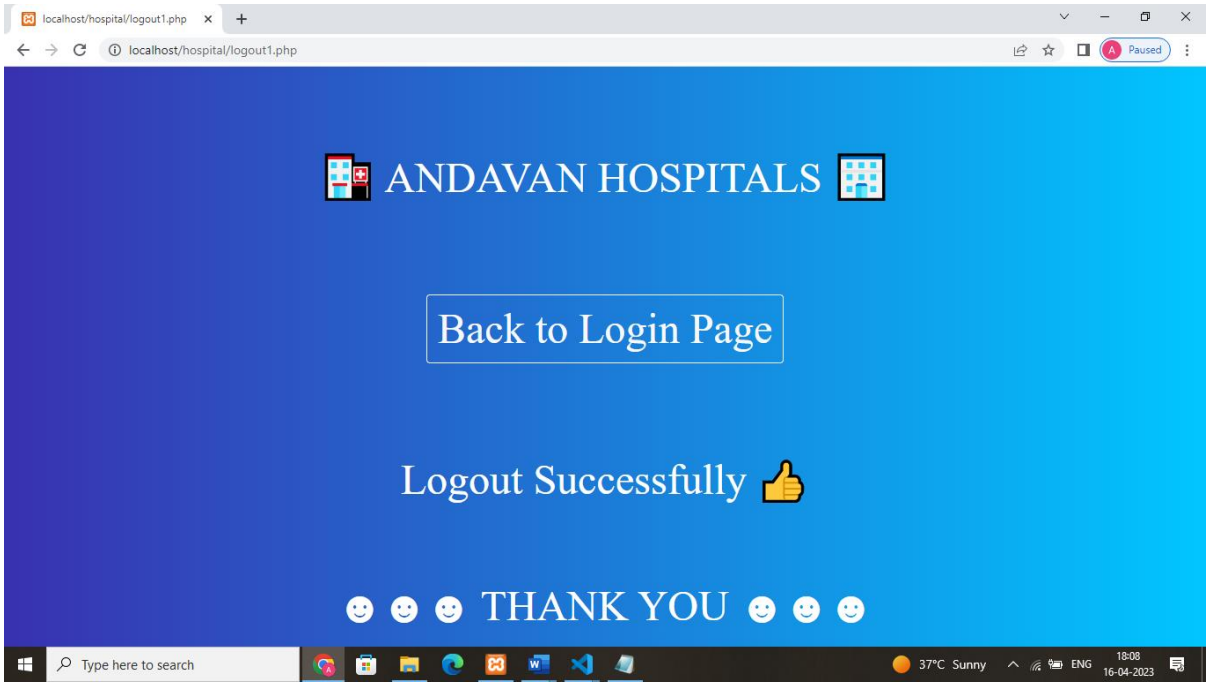
ADD DOCTOR PAGE :



DELETE DOCTOR PAGR :



LOGOUT :


localhost/hospital/logout1.php
localhost/hospital/logout1.php
Paused
ANDAVAN HOSPITALS
Back to Login Page
Logout Successfully
THANK YOU
Type here to search
37°C Sunny
ENG
18:08
16-04-2023

5. TESTING

Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated and performed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system. The philosophy behind testing is to find the errors. A good test is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as an input.

TYPES OF TESTING DONE

SYSTEM TESTING

After a system has been verified, it needs to be thoroughly tested to ensure that every component of the system is performing in accordance with the specific requirements and that it is operating as it should including when the wrong functions are requested or the wrong data is introduced.

Testing measures consist of developing a set of test criteria either for the entire system or for specific hardware, software and communications components. For an important and sensitive system such as an electronic voting system, a structured system testing program may be established to ensure that all aspects of the system are thoroughly tested.

Testing measures that could be followed include:

- Applying functional tests to determine whether the test criteria have been met
- Applying qualitative assessments to determine whether the test criteria have been met.
- Conducting tests in “laboratory” conditions and conducting tests in a variety of “real life” conditions.
- Conducting tests over an extended period of time to ensure systems can perform consistently.

- Conducting “load tests”, simulating as close as possible likely conditions while using or exceeding the amounts of data that can be expected to be handled in an actual situation.

Test measures for hardware may include:

- Applying “non-operating” tests to ensure that equipment can stand up to expected levels of physical handling.
- Testing “hard wired” code in hardware (firmware) to ensure its logical correctness and that appropriate standards are followed.

Tests for software components also include:

- Testing all programs to ensure its logical correctness and that appropriate design, development and implementation standards have been followed.
- Conducting “load tests”, simulating as close as possible a variety of “real life” conditions using or exceeding the amounts of data that could be expected in an actual situation.
- Verifying that integrity of data is maintained throughout its required manipulation.

UNIT TESTING

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components (nodes and vertices) of a product to ensure their correct behavior prior to system integration.

INTEGRATION TESTING

Testing in which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is release.

VALIDATION TESTING

Valid and invalid data should be created and the program should be made to process this data to catch errors. When the user of each module wants to enter into the page by the login page using the use rid and password .If the user gives the wrong password or use rid then the information is provided to the user like “you must enter user id and password”. Here the inputs given by the user are validated. That is password validation, format of date are correct, textbox validation. Changes that need to be done after result of this testing.

6. CONCLUSION

With the development of web based scheduling medical appointment booking system, patients are able to book and manage their own appointment with ease. The system itself also provides a quick view of their appointment at the Home page. These functions could indirectly help to reduce the number of missed medical appointments and patients no-show up for their appointment. The system is mainly designed to facilitate appointment booking between the patient and the health personnel. And allow a flexible management of patient's appointment. Time will not be wasted on converting paper-based appointment record into electronic-based.

Working on the project was an excellent experience. It helped us to understand the importance of planning, designing and implementation so far we have learnt in our theory books. It helped us unleashing our creativity while working in a team. It also realized the importance of team working, communication as a part of this project.

The project was successfully completed after a lot of efforts and work hours. This project underwent number of compiling, debugging, removing errors, making it bug free, adding more facilities in Hospital Management System and interactivity making it more reliable and useful.

7. BIBLIOGRAPHY

BOOK REFERENCE

- Gerken, Till, and Tobias Ratschiller. Web Application Development with PHP. New Riders Publishing, 2000.
- Gutmans, Andi, Stig Bakken, and Derick Rethans. PHP 5 Power Programming (Bruce Perens' Open Source Series). Prentice Hall PTR, 2004.
- Ware, Brent. Open source development with LAMP: using Linux, Apache, MySQL and PHP. Addison-Wesley Longman Publishing Co., Inc., 2002.
- Yank, Kevin. Build your own database driven website using PHP & MySQL. SitePoint Pty Ltd, 2004.

WEBSITE REFERENCE

- www.freetechbooks.com
- www.slideshare.com
- www.w3schools.com
- www.programmersheaven.com
- www.phpreferencebook.com