CHAPTER 1

INTRODUCTION

1.1 Introduction

The Hospital Management System (HMS) is aimed at simplifying the tasks of doctors in keeping track of patient records, treatments and tests they have prescribed to the patients. With the help of this system, doctor can easily add new patients into the hospital, add new tests and treatments for the patients if the doctor feels that the current treatment is not sufficient to cure the disease. The HMS system provides a user-friendly interface so that the doctor can view the details of the patient, which contains the patient name, age, ward number in which the patient is admitted and the disease from which the patient is suffering. The doctor can also view the treatment and test that have been prescribed to a particular patient.

It not only helps doctors but also the patients who can view the treatments and tests that has been prescribed to them. It also facilitates patients to view their expenses in detail i.e., the cost of ward, treatment and test prescribed.

Thus, the HMS system will serve both the doctors and patients in faster keeping track of data, improving service and providing services to store the information centrally.

1.2 Problem Statement

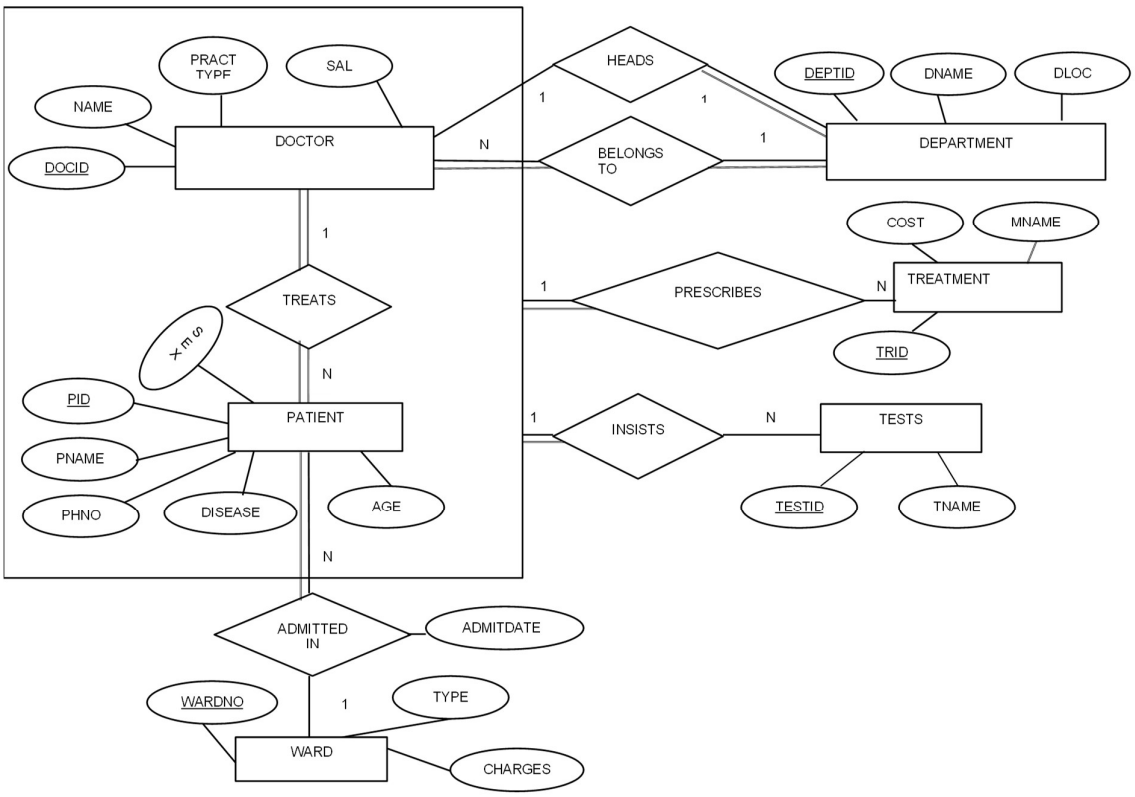
The main aim of this project is to maintain all the information pertaining to patients, doctors and help in managing this huge data in a best possible way. The Hospital Management System should help register patients, prescribe them treatments and tests and also allocate them the type of ward according to requirement. Also, it should help the patients by generating the bill.

CHAPTER 2

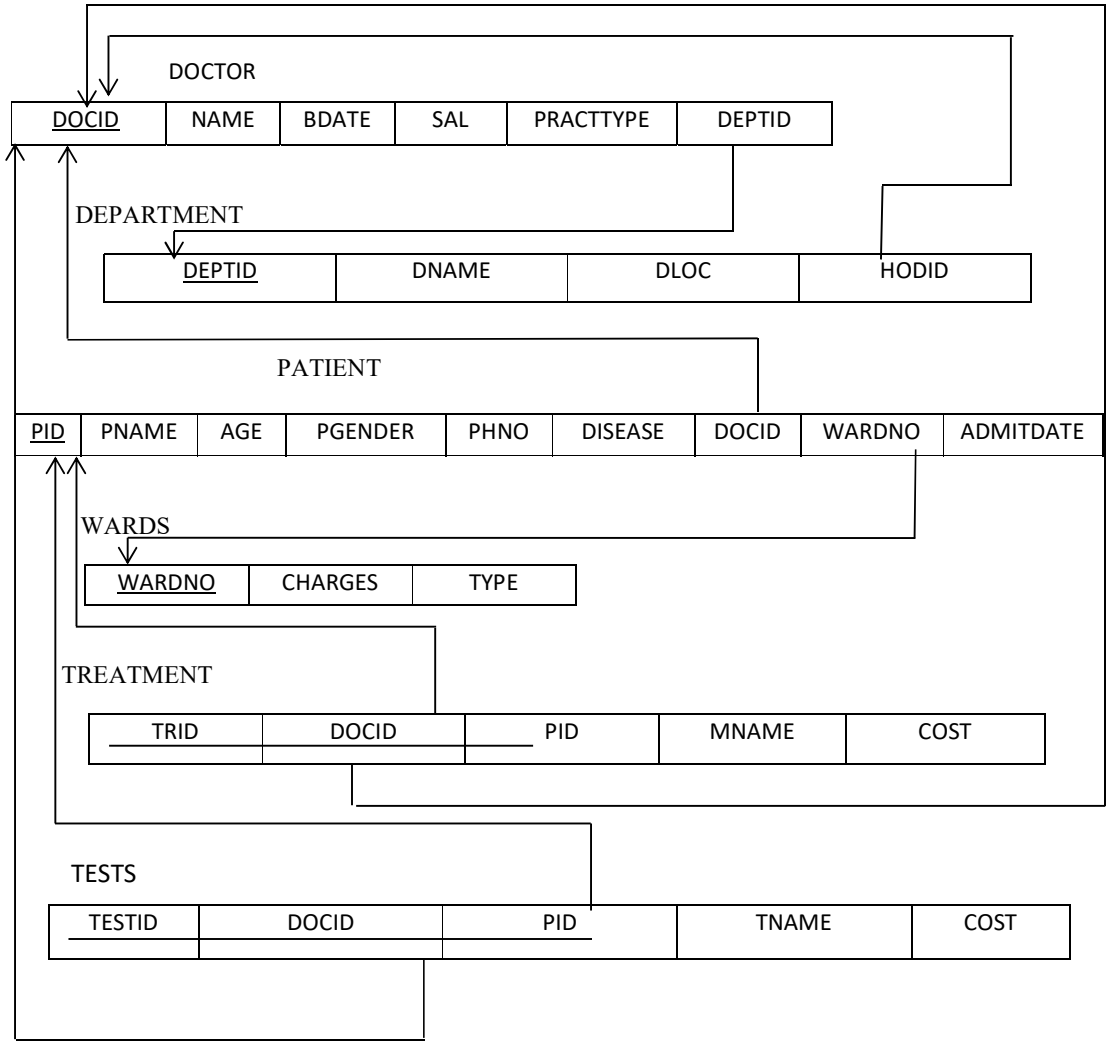
BACK END DESIGN

2.1 ER Diagram

ER DIAGRAM



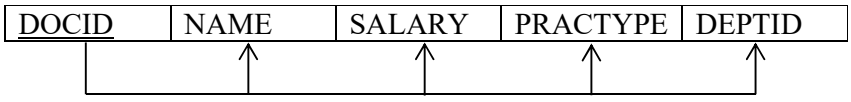
2.2 ER MAPPING



2.3Normalization

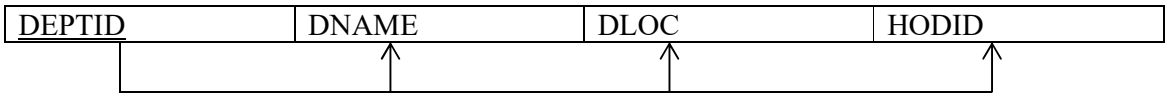
Functional Dependencies

1.Doctor

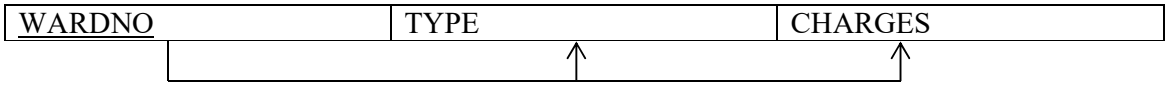


2.Patient

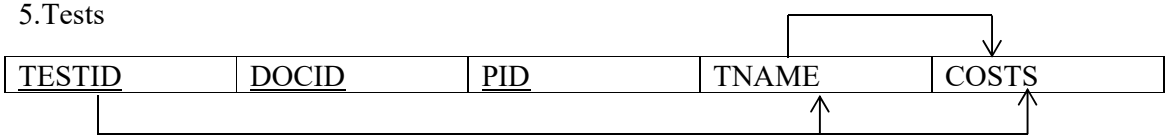
3.Department



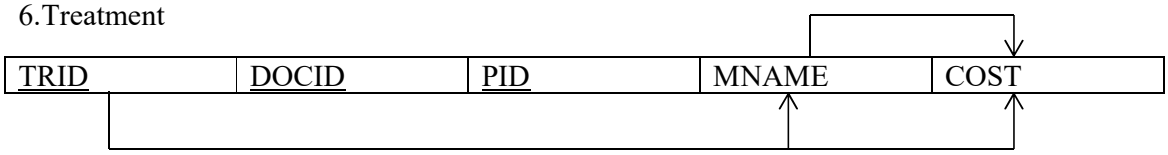
4.Wards



5.Tests



6.Treatment



1st Normal Form

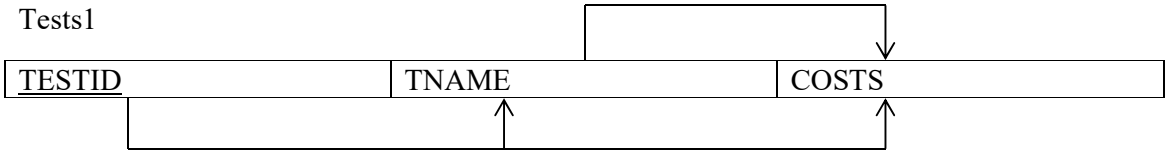
It defines that all attributes in a relation must have atomic domains. In all of the above relations, all the attributes are atomic attributes. Hence the above relations satisfy 1st normal form.

2nd Normal Form

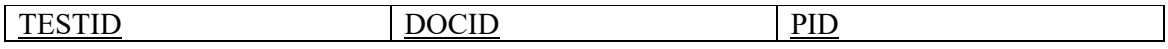
It defines that all the non-prime attributes in a relation must be fully functionally dependent on the prime key attribute.

Consider the relation tests, here tname and costs depend entirely on testid. This is a violation to 2nd normal form. This relation can be decomposed into the following relations:

Tests1

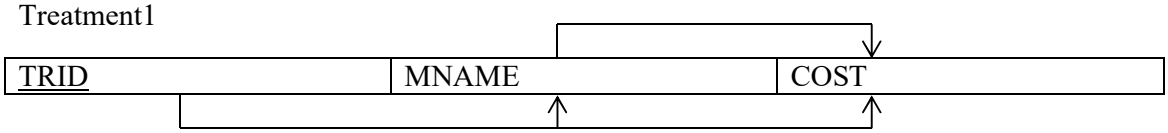


Tests 2



Consider the relation treatment, here mname and cost are only dependent on trid. This is a violation to 2nd normal form. This relation can be decomposed into the following relations:

Treatment1



Treatment 2

TRID	DOCID	PID

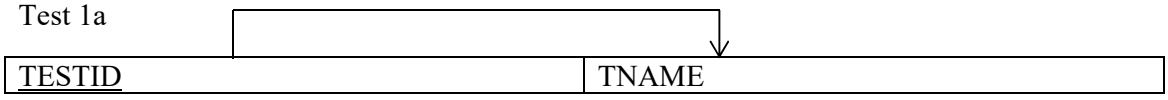
In all of the above relations, all the non-prime attributes are fully functionally dependent on the key attribute. Hence these relations satisfy 2nd normal form.

3rd Normal Form

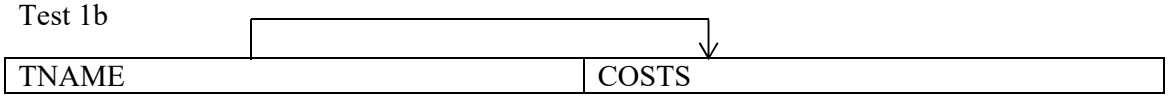
It defines that no non-prime attributes should be transitively dependent.

Consider the relation tests1, here the costs can be determined by both testid and tname, in turn tname can be determined by testid. Hence costs is having transitive dependency which is a violation to 3rd normal form. This relation can be decomposed into the following relations:

Test 1a

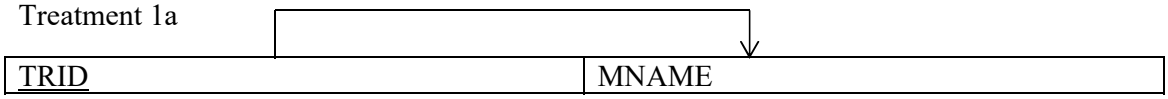


Test 1b

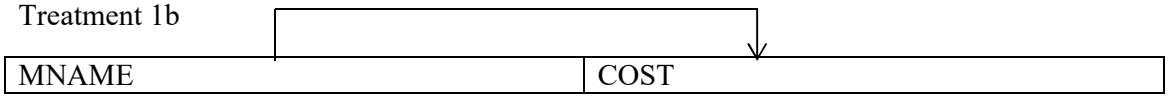


Consider the relation treatment 1, here the cost can be determined by both trid and mname, in turn mname can be determined by trid. Hence cost is having transitive dependency which is a violation to 3rd normal form. This relation can be decomposed into the following relations:

Treatment 1a



Treatment 1b



In all of the above relations, no non-prime attributes are transitively dependent. Hence these relations satisfy 3rd normal form.

CHAPTER 3

FRONT END DESIGN

3.1Screen layout for web pages and forms

3.1.1 Screen layout for login page

```
<!DOCTYPE html>
<html>
<head>
	<style>
		body {
			text-align:center;
			background-image: url("./imgs/subtle.jpg");
			background-size: cover;
		}
		h2 {
			font-size:36px;
			font-weight:bold;
			color:#ffffff;
			margin:30px;
		}
		h3 {
			font-size:30px;
			color:#f1e100;
			margin:5px;
			height:80px;
		}
		hr {
			border: 1px solid black;
		}
		.loginbox {
			border: 3px solid black;
			border-radius: 12px;
			background-color: #f1f1f1;
			width:40%;
		}
		.tab {
		 overflow: hidden;
		 background-color: #f1f1f1;
```

```
 border: 3px solid #ccc;
 border-radius: 12px;
}
.tab button {
  background-color: inherit;
  float: left;
  border: none;
  outline: none;
  cursor: pointer;
  padding: 14px 16px;
  transition: 0.3s;
  font-size: 17px;
}
.tab button:hover {
  background-color: #ddd;
        color:red;
}
.tab button.active {
        background-color: #bbb;
}
.tablinks {
        font-size: 30px !important;
}
.tabcontent {
  padding: 6px 12px;
  border-top: none;
}
.doc_avatar, .pat_avatar {
        border-radius: 50%;
        width: 200px;
        height: 200px;
}
input[type=text], input[type=password] {
        width: 100%;
        padding: 12px 20px;
        margin: 8px 0;
```

```
                display: inline-block;
                border-radius: 12px;
                border: 2px solid black;
                box-sizing: border-box;
        }
        button[type=submit] {
                padding: 12px;
                background-color: green;
                color: white;
                font-size: 18px;
                border-radius: 12px;
                opacity:0.7;
        }
        button[type=submit]:hover {
                opacity: 1.0;
                cursor: pointer;
        }
</style>
<script>
        function loginfunc(evt, loginType) {
                var i, tabcontent, tablinks;
                var start_session_add_dct;
                start_session_add_dct = document.getElementsByClassName("dct");
                start_session_add_dct.className +=" active";
                tabcontent = document.getElementsByClassName("tabcontent");
                for (i = 0; i < tabcontent.length; i++) {
                                tabcontent[i].style.display = "none";
                }
                tablinks = document.getElementsByClassName("tablinks");
                for (i = 0; i < tablinks.length; i++) {
                    tablinks[i].className = tablinks[i].className.replace(" active", "");
                }
                document.getElementById(loginType).style.display = "block";
                evt.currentTarget.className += " active";
        }
</script>
```

```
</head>
<body>
        <div class="welcome">
        <h2>HEALTH CARE </h2>
                <h2>Hospital Management System </h2><hr>
        <h3>Login to Continue </h3>
        </div>
        <div class="loginbox">
                <div class="tab">
                  <button class="active tablinks col-sm-6" onclick="loginfunc(event,
'Doctor')">Doctor</button>
                  <button class="tablinks col-sm-6" onclick="loginfunc(event,
'Patient')">Patient</button>
                </div>
                <div id="Doctor" class="tabcontent" style="display:block">
                        <form action="doctorlogin.php" method="POST">
                                <img class="doc_avatar" src="./imgs/doc.png"><br>
                                <span><b>DocID</b></span>
                                <input type="text" placeholder="Enter DocID" name="docid"
required><br>
                                <span><b>Password</b></span>
                                <input type="password" placeholder="Enter Password"
name="password" required><br>
                                <button type="submit">LOGIN</button>
                        </form>
                </div>
                <div id="Patient" class="tabcontent" style="display:none">
                        <form  action="patientlog.php" method="POST">
                                <img class="pat_avatar" src="./imgs/pat.png" <br>
                                <span><b>PatientID</b></span>
                                <div id=patient">
                                <input type="text" placeholder="Enter PatientID"
name="patid" required><br>
                                <span><b>Password</b></span>
                                <input type="password" placeholder="Enter Password"
name="passwd1" required><br>
                                <button type="submit">LOGIN</button></div>
                        </form>
                </div>
```

```
        </div><br>
</body>
</html>
```

3.1.2 Screen layout for all web pages

```
<style>
                body {
                        background-image: url("imgs/ref.jpg");
                }
                .profile_pic {
                        border-radius:50%;
                        width:50px;
                        height:50px;
                        float:left;
                }
                .profile_banner {
                        font-size: 25px;
                        font-family: 'Google Sans';
                        color:white;
                }
                input.logout_but {
                        background-color:#f00;
                        color:white;
                        font-size: 18px;
                        border: 2px solid black;
                        margin:10px;
                        padding: 5px;
                }
                input.logout_but:hover {
                        background-color:#880;
                        transition-delay:0.1s;
                        transform: scale(1.02);
                }
                hr {
                        border: 1px solid black;
                }
                .your_details {
```

```
                    border: 2px solid black;
                    border-radius: 12px;
                    padding: 20px;
                    font-size: 20px;
                    font-family: monospace;
                    margin: 125px 25px 25px 500px;
                    width: 25%;
                }
                .view_buttons {
                    margin-top:50px;
                    text-align:center;
                    border: 2px solid black;
                    border-radius: 6px;
                    padding:8px;
                    width:100%;
                    font-size:20px;
                    background-color: grey;
                    color:white;
                    cursor: pointer;
                }
                .view_buttons:hover {
                    background-color: green;
                }
  </style>
</head>
<body>
    <nav class="navbar fixed-top navbar-dark bg-dark" style="height:60px">
        <div class="container-fluid">
            <img class="profile_pic" src="./imgs/doc.png"></img>
            <div class="profile_banner">Profile:Dr.<?php echo $name?></div>
            <input class="logout_but" type="button" value="Logout"
onclick="window.location.href='http://localhost/loginpage.php'">
        </div>
    </nav>
```

3.1.3 Screen Layout for Doctor Mainpage

```
 <html>
```

```
<style>
                .your_patients {
        font-size: 20px;
        font-family: monospace;
        margin: 75px 25px 25px 475px;
        width: 35%;
    }
    .treating, .testing {
        border: 2px solid black;
        border-radius: 12px;
        padding: 20px;
        font-size: 20px;
        font-family: monospace;
        margin: 25px 25px 25px 500px;
        width: 25%;
    }
    table {
        border: 3px solid black;
    }
    th,td {
        border:2px solid black;
        padding: 10px;
    }
</style>
<div class="your_details">
        <div><b>DocID: </b><?php echo $doctid ?></div>
        <hr>
        <div><b>Practice Type: </b><?php echo $prac ?></div>
        <hr>
        <div><b>Department:  </b><?php echo $dname ?></div>
        <hr>
    </div>
    <div class="your_patients">
        <h2 style="font-size:30px;text-align:center"><u>Your
Patients</u></h2>
```

```
                    <table>
                    <tr>
                    <th>PID</th>
                    <th>Patient Name</th>
                    <th>Age</th>
                    <th>Disease</th>
                    <th>Ward No</th>
        </table></div>
            <center>
            <div class="container">
                <div class="row">
                        <div class="col"><button class="view_buttons"
onclick="window.location.href='http://localhost/add_patient.php'"> Add New Patient
</button></div>
                        <div class="col"><button class="view_buttons"
onclick="window.location.href='http://localhost/add_test.php'"> Add New Test
</button></div>
                        <div class="col"><button class="view_buttons"
onclick="window.location.href='http://localhost/add_treatment.php'"> Add New Treatment
</button></div>
                        <div class="col"><button class="view_buttons"
onclick="window.location.href='http://localhost/view_test.php'">View Test</button></div>
                        <div class="col"><button class="view_buttons"
onclick="window.location.href='http://localhost/view_tr.php'">View
Treatment</button></div></div><br></div><br>
                <div class="row">
                        <div class="col"><form action="<?php $_PHP_SELF ?>"
method="POST">
                            <input type="text" style="padding:4px;border:2px solid
black;border-radius:6px" name="pid4" placeholder="Enter Patient ID" />
                            <input type="submit" class="hidden_block del_pat"
name="del_pat" value="Delete Patient" /></form>
```

3.1.4 Screen layout for patient main page

```
<style>
        .your_details,.your_bill {
            border: 2px solid black;
            border-radius: 12px;
            padding: 20px;
```

```
                font-size: 20px;
                font-family: monospace;
                margin: 25px 25px 25px 500px;
                width: 25%;
        }
 </style>
    <script>
                function showBill() {
   document.getElementById('billing').style.display = "block";
  }
    </script>
</head>
<body style="height:1500px">
    <nav class="navbar fixed-top navbar-dark bg-dark" style="height:70px">
        <div class="container-fluid">
            <img class="profile_pic" src="./imgs/pat.png"></img>
            <div class="profile_banner">Profile: <?php echo $pname?></div>
            <button class="logout_but" type="button"
onclick='window.location.href="http://localhost/loginpage.php"'>Logout</a></button>
        </div>
    </nav>
    <div class="maincontent">
        <div class="your_details">
            <div><b>Your ID: </b><?php echo $pid ?></div>
            <hr>
            <div><b>Age: </b><?php echo $age ?></div>
            <hr>
            <div><b>Phone Number:  </b><?php echo $phno ?></div>
            <hr>
            <div><b>Your Doctor: </b><?php echo "Dr." . $dname ?></div>
            <hr>
            <div><b>Your Ward: </b><?php echo $wno ?></div>
            <hr>
            <div><b>Admitted Date: </b><?php echo $adate ?></div>
            <hr>
        </div>
    </div>
```

```
        <div id="billing" style="display:none;">
                <div class="your_bill">
                        <h2 style="font-weight:800;color:blue"><u>Your Expenses</u></h2><br>
                        <div><b>Test Costs: </b><?php echo $tcost?></div>
                        <hr>
                        <div><b>Treatment Costs: </b><?php echo $trcost?></div>
                        <hr>
                        <div><b>Ward Costs: </b><?php echo $wcost?></div>
                        <hr>
                        <div><b>Total Costs: </b><?php echo $total?></div>
                        <hr>
                </div>
        </div>
        <div class="container">
                <div class="row"><button class="view_buttons" onclick="window.location.href = './pat_test.php'"> Click to view your Tests </button></div>
                <div class="row"><button class="view_buttons" onclick="window.location.href='./pat_tr.php'"> Click to view your Treatments </button></div>
        </div>
        <div class="container">
                <div class="row"><input type="button" class="view_bill" value="Click to display your Bill" onclick="showBill()"></div>
        </div>
        <br><br>
    </div>
</body>
</html>
```

3.1.5 Screen layout for forms

```
<html>
    <head>
    <style>
            form {
                text-align:center;
                font-size: 20px;
            }
```

```
                input[type="radio"]:hover {
                        cursor: pointer;
                }
                input[type="text"] {
                        border: 2px solid black;
                        padding: 7px;
                        border-radius: 12px;
                }
                input[type="password"] {
                        border: 2px solid black;
                        padding: 7px;
                        border-radius: 12px;
                }
                button {
                        margin-top:80px;
                        font-size:20px;
                        background-color:red !important;
                }
                input[type="submit"], button {
                        background-color: black;
                        color:white;
                        border: 3px solid black;
                        border-radius:10px;
                        padding: 6px;
                }
                input[type="submit"]:hover, button:hover {
                        background-color: green !important;
                        cursor: pointer;
                }
        </style>
    </head>
```

3.2 Connectivity

The database is connected to the front end through the following piece of code.

```
<?php
        $servername="localhost";
        $username="root";
```

```
        $password="nischaya#89$&";
        $dbname="hdbms";
        if($conn->connect_error) {
                die("connection failed:" .$conn->connect_error);
        }
        $conn=mysqli_connect($servername,$username,$password,$dbname);
?>
```

CHAPTER 4

MODULES ALONG WITH FUNCTIONALITIES

MODULES

The major modules in the project are doctor and patient. The functionalities of these modules are described below.

4.1 Doctor Module

Doctor uses the docid and the password to login.

On login the details of the doctor such as docid, practice type and the department to which the doctor belongs are listed and the functions which the doctor can perform are also displayed.

The functionalities in doctor module are:

4.1.1 Add new patient

The doctor main page includes a button, which upon clicking takes the doctor to the form to add new patient using which new patient details has to be entered i.e., ID to be assigned for the new patient, patient name, phno age, etc. Upon successful execution, new patient will be created.

4.1.2 Add new test

When the doctor wants to add a new test for an existing patient, the doctor has to fill the add new test form by entering the details such as testid, test name, ID of the patient to whom the test is prescribed and the cost of the test. On successful execution new test will be created.

4.1.3 Add new treatment

When the doctor wants to add a new test for an existing patient, the doctor has to fill the add new test form by entering the details such as treatment ID, treatment name, ID of the patient to whom the treatment is prescribed and the cost of the treatment. On successful execution new treatment will be created.

4.1.4 View Tests

When the doctor wants to view the already prescribed tests, the doctor can just enter the ID of the patient whose tests he/she wants to view and then the prescribed test of the respective patient will be displayed.

4.1.5 View Treatment

When the doctor wants to view the already prescribed treatment, the doctor can just enter the ID of the patient whose treatment he/she wants to view and then the prescribed treatment of the respective patient will be displayed.

4.1.6 Delete Patient

The doctor can delete an already existing patient by entering the ID of the patient whose details the doctor wishes to delete.

4.2 Patient Module

The patient can use the assigned patient ID and the password to login to the system.

On login the details of the patient such as patient ID, treating doctor's name, name of the disease from which the patient is suffering are displayed. The main page also contains the functions such as viewing tests, treatment and the bill.

4.2.1 View Test

The patient can view the details of the tests i.e, testID, name of the test and the cost of the test prescribed to him/her.

4.2.2 View Treatment

The patient can view the details the treatment i.e, treatment ID, name of the treatment and the cost of the treatment prescribed to him/her.

4.2.3 View Bill

The patient can view his/her bill which is the total cost of test, treatment and ward.

CHAPTER 5

IMPLEMENTATION USING MYSQL AND PHP

5.1Table Creation

1.Department Table

```
CREATE TABLE `department` (
  `deptid` int(11) NOT NULL,
  `dname` varchar(45) DEFAULT NULL,
  `dloc` varchar(45) DEFAULT NULL,
  `hodid` int(11) DEFAULT NULL,
  PRIMARY KEY (`deptid`),
  KEY `hodid_idx` (`hodid`),
   CONSTRAINT `hodidfk` FOREIGN KEY (`hodid`) REFERENCES `doctor` (`docid`) ON DELETE CASCADE ON UPDATE CASCADE
);
```

2.Doctor Table

```
CREATE TABLE `doctor` (
  `docid` int(11) NOT NULL,
  `name` varchar(45) DEFAULT NULL,
  `sal` int(11) DEFAULT NULL,
  `practype` varchar(45) DEFAULT NULL,
  `deptid` int(11) DEFAULT NULL,
  PRIMARY KEY (`docid`),
  KEY `deptfk_idx` (`deptid`),
  CONSTRAINT `deptfk` FOREIGN KEY (`deptid`) REFERENCES `department` (`deptid`) ON DELETE CASCADE ON UPDATE CASCADE
);
```

3.Patient Table

```
CREATE TABLE `patient` (
  `pid` int(11) NOT NULL,
  `pname` varchar(45) DEFAULT NULL,
  `age` int(11) DEFAULT NULL,
  `phno` char(10) DEFAULT NULL,
  `disease` varchar(45) DEFAULT NULL,
  `docid` int(11) DEFAULT NULL,
  `admitdate` date DEFAULT NULL,
  `wardno` int(11) DEFAULT NULL,
  `pgender` char(1) DEFAULT NULL,
```

```
  `passwd` varchar(100) NOT NULL,
  PRIMARY KEY (`pid`),
  KEY `pdocfk_idx` (`docid`),
  KEY `pdwfk_idx` (`wardno`),
  CONSTRAINT `pdocfk` FOREIGN KEY (`docid`) REFERENCES `doctor` (`docid`) ON DELETE SET NULL ON UPDATE CASCADE,
  CONSTRAINT `pdwfk` FOREIGN KEY (`wardno`) REFERENCES `wards` (`wardno`) ON DELETE SET NULL ON UPDATE CASCADE
);
```

4.Wards table

```
CREATE TABLE `wards` (
  `wardno` int(11) NOT NULL,
  `type` varchar(45) DEFAULT NULL,
  `charges` int(11) DEFAULT NULL,
  PRIMARY KEY (`wardno`)
);
```

5.Tests table

```
CREATE TABLE `tests` (
  `testid` int(11) NOT NULL,
  `docid` int(11) NOT NULL,
  `pid` int(11) NOT NULL,
  `tname` varchar(45) DEFAULT NULL,
  `costs` int(11) DEFAULT NULL,
  PRIMARY KEY (`testid`,`docid`,`pid`),
  KEY `tdocfk_idx` (`docid`),
  KEY `tpifk_idx` (`pid`),
  CONSTRAINT `tdocfk` FOREIGN KEY (`docid`) REFERENCES `doctor` (`docid`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `tpifk` FOREIGN KEY (`pid`) REFERENCES `patient` (`pid`) ON DELETE CASCADE ON UPDATE CASCADE
);
```

6.Treatment Table

```
CREATE TABLE `tests` (
  `testid` int(11) NOT NULL,
  `docid` int(11) NOT NULL,
  `pid` int(11) NOT NULL,
```

```
 `tname` varchar(45) DEFAULT NULL,
 `costs` int(11) DEFAULT NULL,
 PRIMARY KEY (`testid`,`docid`,`pid`),
 KEY `tdocfk_idx` (`docid`),
 KEY `tpifk_idx` (`pid`),
 CONSTRAINT `tdocfk` FOREIGN KEY (`docid`) REFERENCES `doctor` (`docid`) ON DELETE CASCADE ON UPDATE CASCADE,
 CONSTRAINT `tpifk` FOREIGN KEY (`pid`) REFERENCES `patient` (`pid`) ON DELETE CASCADE ON UPDATE CASCADE
);
```

5.2 Insertion into tables

Department table

```
insert into department values(1,'admin','ground',null);
select * from department;
+--------+------------------+---------+-------+
| deptid | dname            | dloc    | hodid |
+--------+------------------+---------+-------+
|      1 | admin            | ground  |  NULL |
|      2 | emergency        | first   |   119 |
|      3 | cardiology       | second  |   106 |
|      4 | neuorology       | third   |   109 |
|      5 | oncology         | cgblock |   108 |
|      6 | gastroenterology | fourth  |   115 |
|      7 | anaesthetics     | fourth  |   117 |
|      8 | gynecology       | fifth   |   107 |
|      9 | urology          | fifth   |   122 |
+--------+------------------+---------+-------+
```

Doctor table

```
select * from doctor;
+-------+----------------------+--------+--------------------+----
----+
| docid | name                 | sal    | practype           |
deptid |
+-------+----------------------+--------+--------------------+----
----+
|   101 | Manjunath Shastry    |  35000 | consultant         |
6 |
|   102 | Jayanthi Shastry     |  25000 | consultant         |
8 |
|   103 | Ajay Shastry         |  85000 | neuro surgeon      |
4 |
|   104 | Ananya Shastry       |  80000 | cardiac surgeon    |
3 |
|   105 | Srihari Kulakarni    |  10000 | team doctor        |
7 |
|   106 | Sugandhi Sharma      | 100000 | cardiac surgeon    |
3 |
|   107 | Lalitha Iyer         | 100000 | gynecologist       |
8 |
```

```
|   108 | Padmasini Sridharan   | 200000 | oncologist          |
5 |
|   109 | Shelavapille Iyyangar | 300000 | neuro surgeon       |
4 |
|   110 | Snehal Kulakarni      |  30000 | team doctor         |
5 |
|   111 | Shruthi Nayar         |  20000 | team doctor         |
4 |
|   112 | Anika Agarwal         |  20000 | team doctor         |
3 |
|   113 | Arushi Ghosh          |  15000 | team doctor         |
6 |
|   114 | Nishitha Gowda        |  55000 | surgeon             |
5 |
|   115 | Nikhil Gowda          |  45000 | laproscopic surgeon |
6 |
|   116 | Sanjana Hadimani      |  15000 | team doctor         |
7 |
|   117 | Sripad Desai          |  75000 | Anaesthetic expert  |
7 |
|   118 | Debjani Chatterjee    |  15000 | team doctor         |
8 |
|   119 | Nischal Mishra        |  35000 | surgeon             |
2 |
|   120 | Arna Nidu             |  25000 | surgeon             |
2 |
|   121 | Adishwar Reddy        |  25000 | surgeon             |
2 |
|   122 | Adisheshulu Naidu     |  75000 | urologist           |
9 |
|   123 | Sumadhwa Kulakarni    |  45000 | consultant          |
9 |
|   124 | Adhokshaj Prahlad     |  15000 | team doctor         |
9 |
+-------+-----------------------+--------+---------------------+----
----+
```

Patient table

INSERT INTO `patient` VALUES (301,'Thimmegowda',65,'9845190852','liver cirrohsis',101,'2018-08-23',1501,'m');

```
select * from patient;
+-----+---------------+------+-----------+-------------------+----
---+-----------+--------+---------+
| pid | pname         | age  | phno      | disease           |
docid | admitdate  | wardno | pgender |
+-----+---------------+------+-----------+-------------------+----
---+-----------+--------+---------+
| 301 | Thimmegowda   |   65 | 9845190852 | liver cirrohsis   |
101 | 2018-08-23 |   1501 | m       |
| 302 | Niharika      |   48 | 9765190802 | Fibroid Uterus    |
102 | 2018-09-11 |   1304 | f       |
| 303 | Advika        |   28 | 9723898020 | PCOS              |
107 | 2018-02-10 |   1306 | f       |
| 304 | Hari Kumar    |   54 | 8523898025 | cardiac arrest    |
104 | 2018-03-22 |   1201 | m       |
```

```
| 305 | Harish          |   64 | 8590778028 | brain tumor       |
103 | 2018-05-22 |   1101 | m       |
| 306 | Karibasappa     |   69 | 7590772327 | colon cancer      |
110 | 2018-05-22 |   1101 | m       |
| 307 | Nagappa         |   39 | 8890772329 | stroke            |
109 | 2018-05-24 |   1403 | m       |
| 308 | Narendra        |   44 | 9023077232 | leukamia          |
108 | 2018-06-15 |   1401 | m       |
| 309 | Smrithi         |    5 | 7234509816 | septal hole       |
106 | 2018-06-15 |   1406 | f       |
| 310 | Sharada         |   95 | 7234902341 | gall stones       |
115 | 2018-07-25 |   1403 | f       |
| 311 | Mrinalini       |   35 | 8034902341 | cerebral palsy    |
111 | 2018-04-12 |   1303 | f       |
| 312 | Mallika         |   55 | 9834902341 | breast cancer     |
108 | 2018-05-11 |   1504 | f       |
| 313 | srinidhi        |   43 | 9123902341 | head injury       |
119 | 2018-06-01 |   1404 | m       |
| 314 | sumathi         |   45 | 8053902341 | liver enlargement |
113 | 2018-07-04 |   1405 | f       |
| 315 | abhishek        |   42 | 7789002803 | kidney failure    |
123 | 2018-07-06 |   1302 | m       |
| 316 | nischal         |   22 | 8123900280 | kidney stones     |
124 | 2018-04-09 |   1303 | m       |
| 317 | basheer         |   67 | 8723334580 | appendicitis      |
113 | 2018-02-25 |   1406 | m       |
| 318 | peter           |   27 | 9033334580 | kidney stones     |
124 | 2018-02-25 |   1406 | m       |
| 319 | isabella        |   35 | 7233884580 | parkinsons        |
104 | 2018-03-08 |   1505 | f       |
| 320 | sheikh abdulla |   55 | 8233884588 | liver failure     |
101 | 2018-01-10 |   1502 | m       |
+-----+----------------+------+------------+-------------------+----
---+------------+--------+---------+
```

Wards table

INSERT INTO `wards` VALUES (1101,'ICU',25000);

```
select * from wards;
+--------+---------------+---------+
| wardno | type          | charges |
+--------+---------------+---------+
|   1101 | ICU           |   25000 |
|   1201 | ICU 2         |   25000 |
|   1301 | special       |   15000 |
|   1302 | special       |   15000 |
|   1303 | semi special  |   10000 |
|   1304 | semi special  |   10000 |
|   1305 | General       |    5000 |
|   1306 | General       |    5000 |
|   1401 | special       |   15000 |
|   1402 | semi special  |   10000 |
|   1403 | semi special  |   10000 |
|   1404 | special       |   15000 |
|   1405 | General       |    5000 |
|   1406 | General       |    5000 |
|   1501 | special       |   15000 |
|   1502 | special       |   15000 |
```

```
|   1503 | semi special  |   10000 |
|   1504 | semi special  |   10000 |
|   1505 | General       |    5000 |
|   1506 | General       |    5000 |
+--------+---------------+---------+
```

Tests table

insert into tests values(61,101,301,'endoscopy',2500);

```
select * from tests;
+--------+-------+-----+------------------+-------+
| testid | docid | pid | tname            | costs |
+--------+-------+-----+------------------+-------+
|     61 |   101 | 301 | endoscopy        |  2500 |
|     62 |   102 | 302 | pelvic exam      |  1500 |
|     63 |   107 | 303 | ultrasound scan  |  2000 |
|     64 |   104 | 304 | angiogram        |  5000 |
|     65 |   103 | 305 | MRI              |  7000 |
|     66 |   110 | 306 | colonoscopy      |  7000 |
|     67 |   110 | 306 | biopsy           |  7000 |
|     68 |   109 | 307 | CPSS             |  1000 |
|     69 |   108 | 308 | biopsy           |  5000 |
|     70 |   106 | 309 | ECG              |  1000 |
|     71 |   115 | 310 | colongioscopy    |  4000 |
|     72 |   111 | 311 | MRI              | 10000 |
|     73 |   108 | 312 | Mammography      | 12000 |
|     74 |   119 | 313 | MRI              |  7000 |
|     75 |   113 | 314 | endoscopy        |  2500 |
|     76 |   123 | 315 | serum creatinine |  5500 |
|     77 |   124 | 316 | ultrasound       |  3500 |
|     78 |   113 | 317 | colonoscopy      |  2500 |
|     79 |   124 | 318 | ultrasound       |  3500 |
|     80 |   104 | 319 | CT scan          |  6000 |
|     81 |   101 | 320 | Endoscopy        |  2500 |
+--------+-------+-----+------------------+-------+
```

Treatment table

insert into treatment values(5618,101,320,'Liver transplant',100000);

```
select * from treatment;
+------+-------+-----+--------------------------------+--------+
| trid | docid | pid | mname                          | cost   |
+------+-------+-----+--------------------------------+--------+
| 5601 |   101 | 301 | NSAIDS                         |   2500 |
| 5602 |   102 | 302 | hysterectomy                   |  60000 |
| 5603 |   107 | 303 | Laparoscopic Ovarine Drilling |  20000 |
| 5604 |   104 | 304 | Bypass Surgery                 |  70000 |
| 5605 |   103 | 305 | Tumor removal surgery          | 200000 |
| 5606 |   108 | 312 | Chemotherapy                   | 100000 |
| 5606 |   110 | 306 | Chem0therapy                   | 100000 |
| 5607 |   109 | 307 | Physiotherapy                  |  10000 |
| 5608 |   108 | 308 | Immunetherapy                  |  50000 |
| 5609 |   106 | 309 | cardiac catheterization        |  45000 |
| 5610 |   115 | 310 | laparoscopy                    |  55000 |
| 5611 |   111 | 311 | CSV                            |  65000 |
| 5612 |   119 | 313 | RP Surgery                     |  30000 |
| 5613 |   113 | 314 | Liver ASD                      |  30000 |
| 5614 |   123 | 315 | Kidney transplant              | 300000 |
```

```
| 5615 |   124 | 316 | StoneCR surgery               |  38000 |
| 5615 |   124 | 318 | StoneCR surgery               |  38000 |
| 5616 |   113 | 317 | APR surgery                   |  40000 |
| 5617 |   104 | 319 | Stemcell treatment            |  98000 |
| 5618 |   101 | 320 | Liver transplant              | 100000 |
+------+-------+-----+-------------------------------+--------+
```

5.3 Trigger

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers can be defined on the table, view, schema, or database with which the event is associated. The main difference between a trigger and a stored procedure is that a trigger is called automatically when a data modification event is made against a table whereas a stored procedure must be called explicitly.

```
CREATE DEFINER=`root`@`localhost` TRIGGER `patient_AFTER_INSERT` BEFORE
INSERT ON `patient` FOR EACH ROW

BEGIN

        if char_length(new.phno)<10

                then set new.phno=null;

        end if;

END
```

5.4 Stored Procedure

A stored procedure is a subroutine available to applications that access a relational_database management_system. Such procedures are stored in the database data_dictionary. A stored procedure is nothing more than prepared SQL code that is used when the same code is executed over and over again.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `bill`(in ppid int,out total int)
BEGIN
  select tr.cost,t.costs,w.charges,(tr.cost+t.costs+w.charges) as total
  from treatment tr,tests t,wards w,patient p
  where tr.pid=p.pid and t.pid=p.pid
  and w.wardno=p.wardno
  and p.pid=ppid;
END ;;
DELIMITER ;
```

5.5.1 PHP Code for add new patient

```
<?php
        $pid=$_POST['pid'];
        $pname=$_POST['pname'];
```

```
$age=$_POST['age'];

$phno=$_POST['phno'];

$disease=$_POST['disease'];

$docid=$_SESSION['docid'];

$admitdate=$_POST['admitdate'];

$wardno=$_POST['wardno'];

$pgender=$_POST['pgender'];

$pass=$_POST['pass'];

$paswd=password_hash($pass,PASSWORD_DEFAULT);

$stmt=$conn->prepare("INSERT INTO `patient` (`pid`, `pname`, `age`, `phno`,
`disease`, `docid`, `admitdate`, `wardno`, `pgender`,`passwd`) VALUES(?,?,?,?,?,?,?,?,?,?)");

$stmt-
>bind_param('isisssisiss',$pid,$pname,$age,$phno,$disease,$docid,$admitdate,$wardno,$pgen
der,$paswd);

$stmt->execute();  ?>
```

5.5.2 PHP Code for add new test

```
<?php

$testid=$_POST['testid'];

$pid=$_POST['pid'];

$tname=$_POST['tname'];

$cs=$_POST['costs'];

$stmt=$conn->prepare( "INSERT INTO `tests` (`testid`, `docid`, `pid`, `tname`,
`costs`) VALUES(?,?,?,?,?)") ;

$stmt->bind_param('iiisi',$testid,$docid,$pid,$tname,$cs);

$stmt->execute();  ?>
```

5.5.3 Add new treatment

```
<?php

$trid=$_POST['trid'];

$pid=$_POST['pid'];

$mn=$_POST['mname'];

$cost=$_POST['cost'];

$stm=$conn->prepare( "INSERT INTO `treatment` (`trid`, `docid`, `pid`, `mname`,
`cost`) VALUES (?,?,?,?,?)");

$stm->bind_param('iiisi',$trid,$docid,$pid,$mn,$cost);

$stm->execute();  ?>
```

5.5.4 View test

```
<?php
```

```
$pid1=$_POST['pid1'];
$q="select testid,tname,costs from tests where pid=$pid1";
$re=mysqli_query($conn,$q);
while($row = $re->fetch_assoc()) {
        $tid= $row['testid'];
        $tn= $row['tname'];
        $cs= $row['costs'];
} ?>
```

5.5.5 View treatment

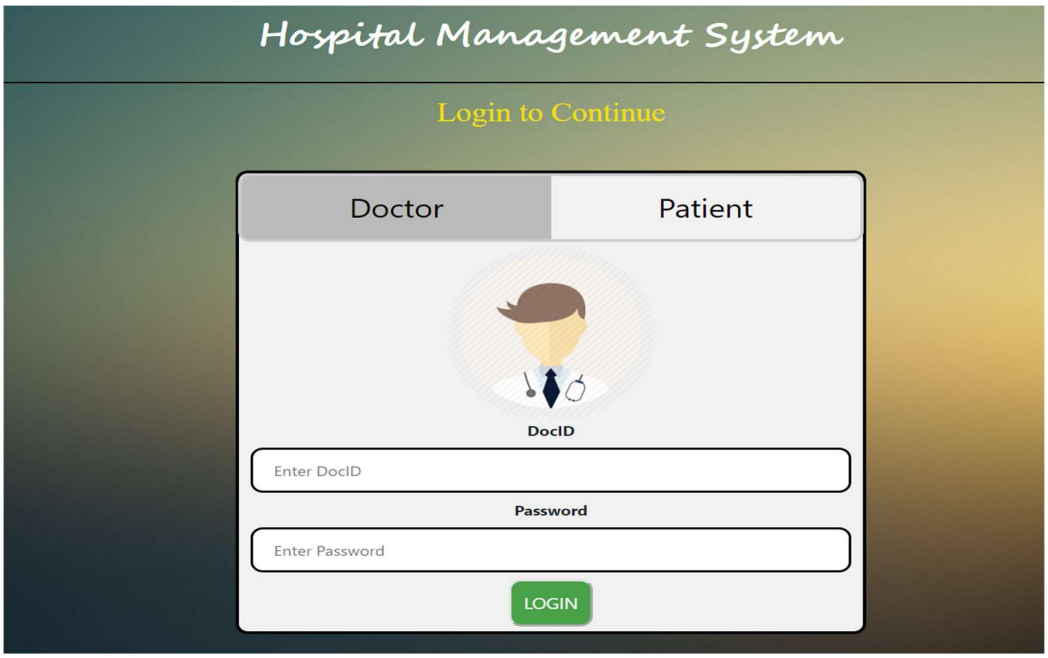
```
<?php
$pid2=$_POST['pid2'];
$q1="select trid,mname,cost from treatment where pid=$pid2";
$rt=mysqli_query($conn,$q1);
while($row = $rt->fetch_assoc()) {
        $tmid=$row['trid'];
        $mn=$row['mname'];
        $ct=$row['cost'];
} ?>
```

5.5.6 Delete Patient

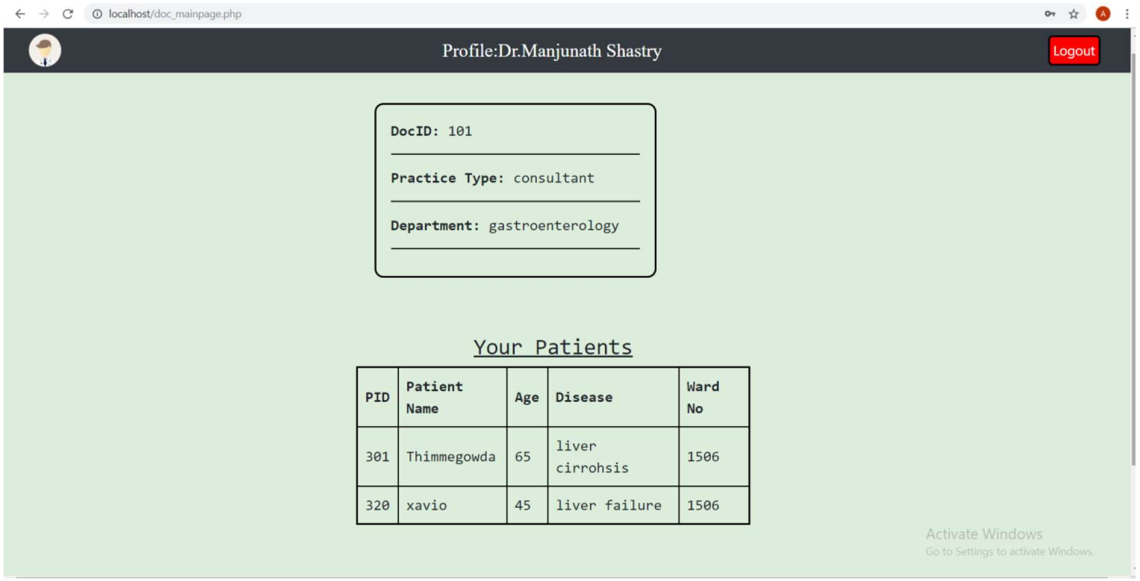
```
<?php
$pid4=$_POST['pid4'];
$q2="delete from patient where pid=$pid4";
$rs=mysqli_query($conn,$q2);
if($conn->query($q2) === TRUE) {
        echo "patient deleted successfully";
} ?>
```

CHAPTER 6

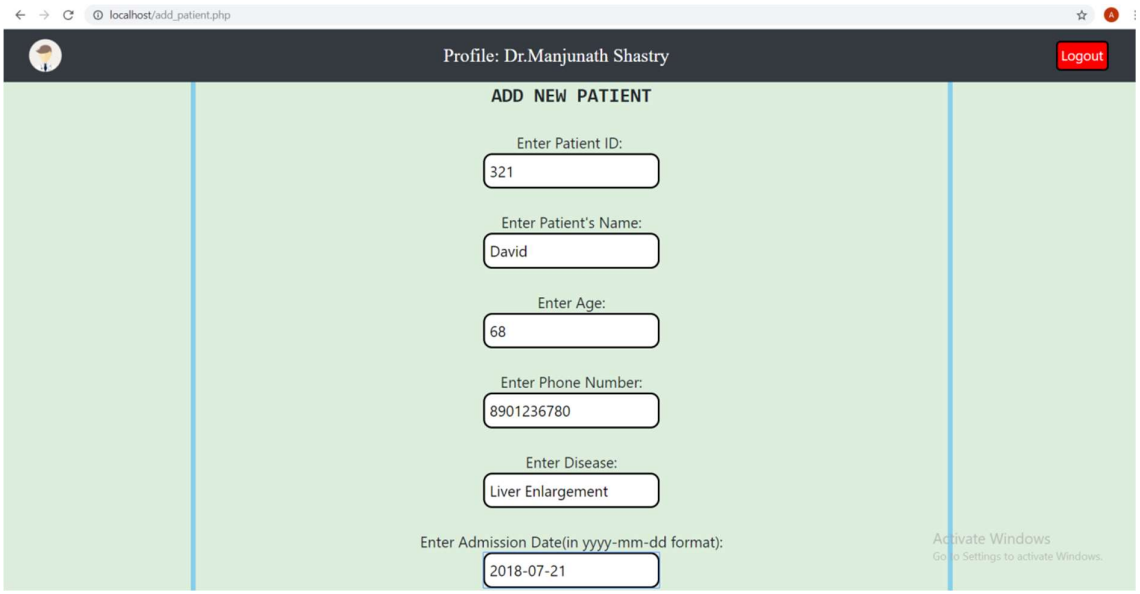
SNAPSHOTS



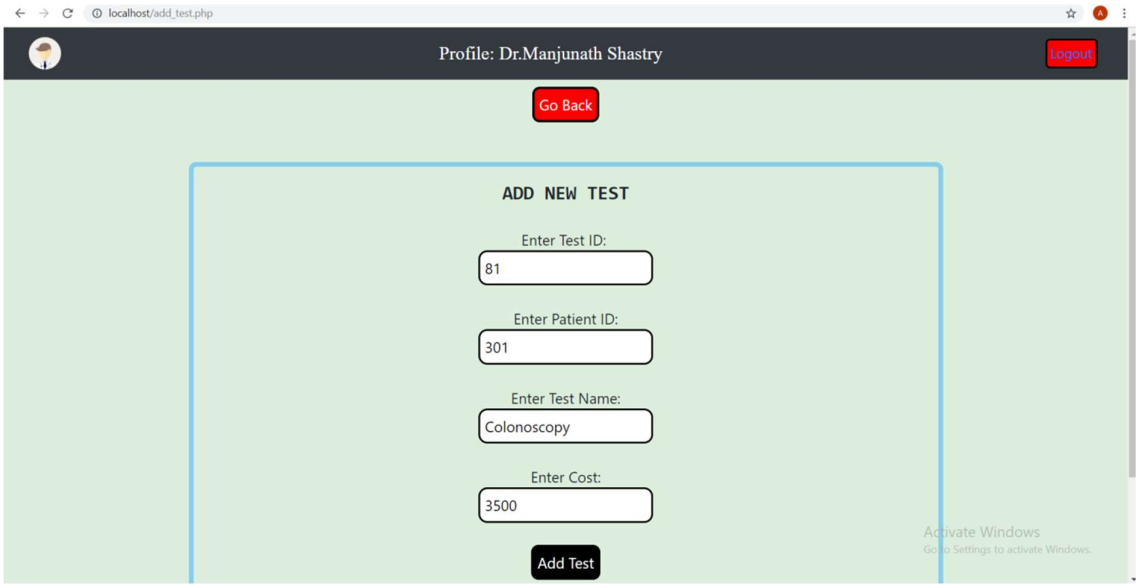
Doctor/Patient Login Page.



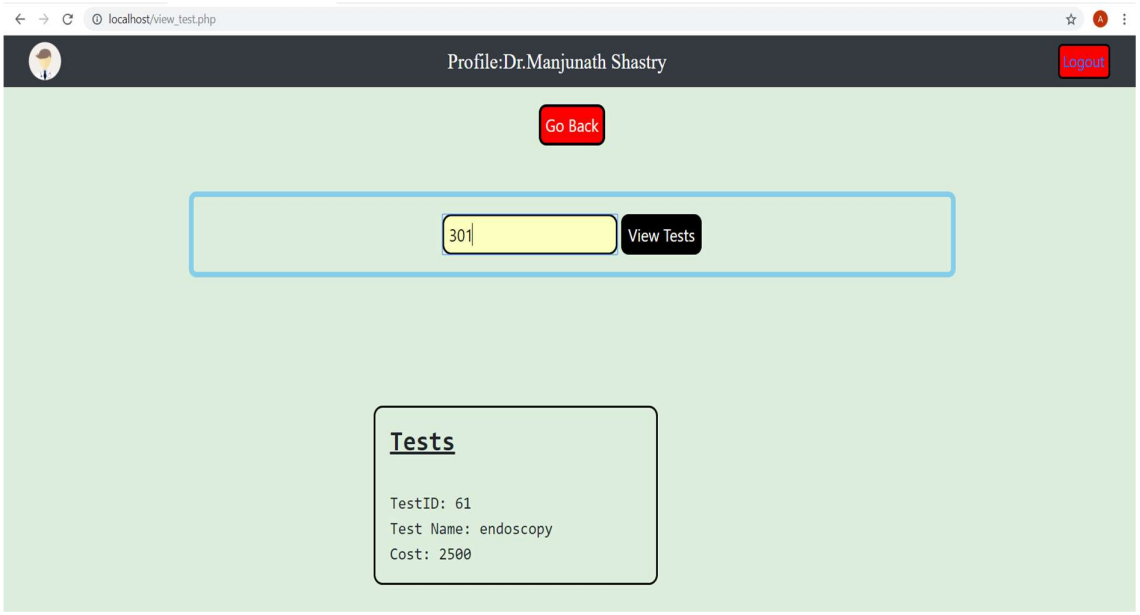
Doctor main page



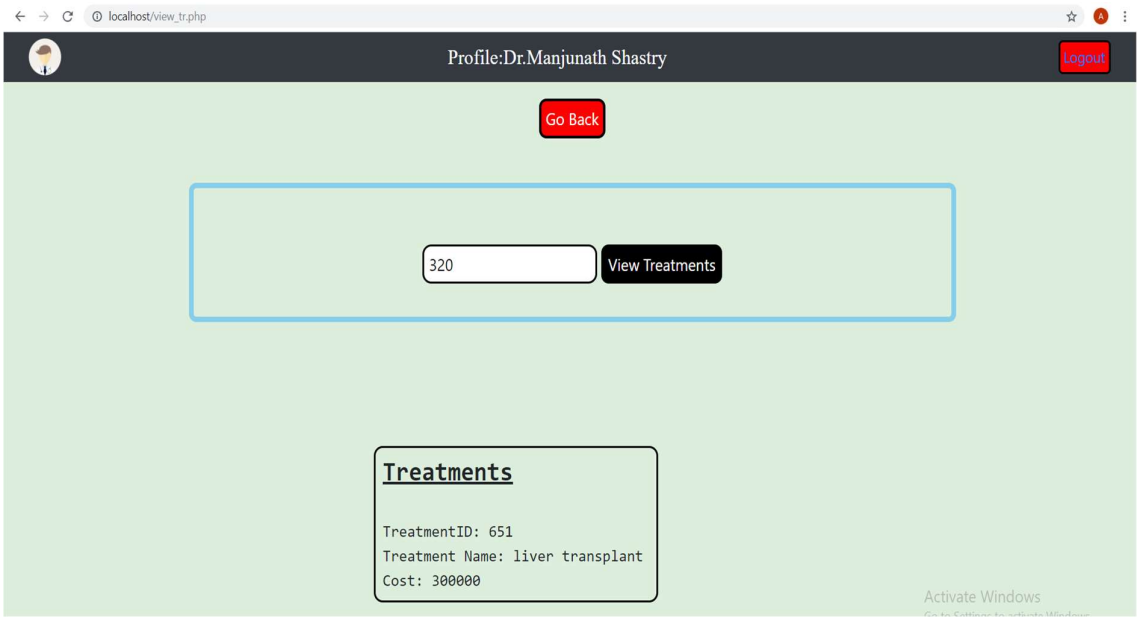
New patient addition by doctor.



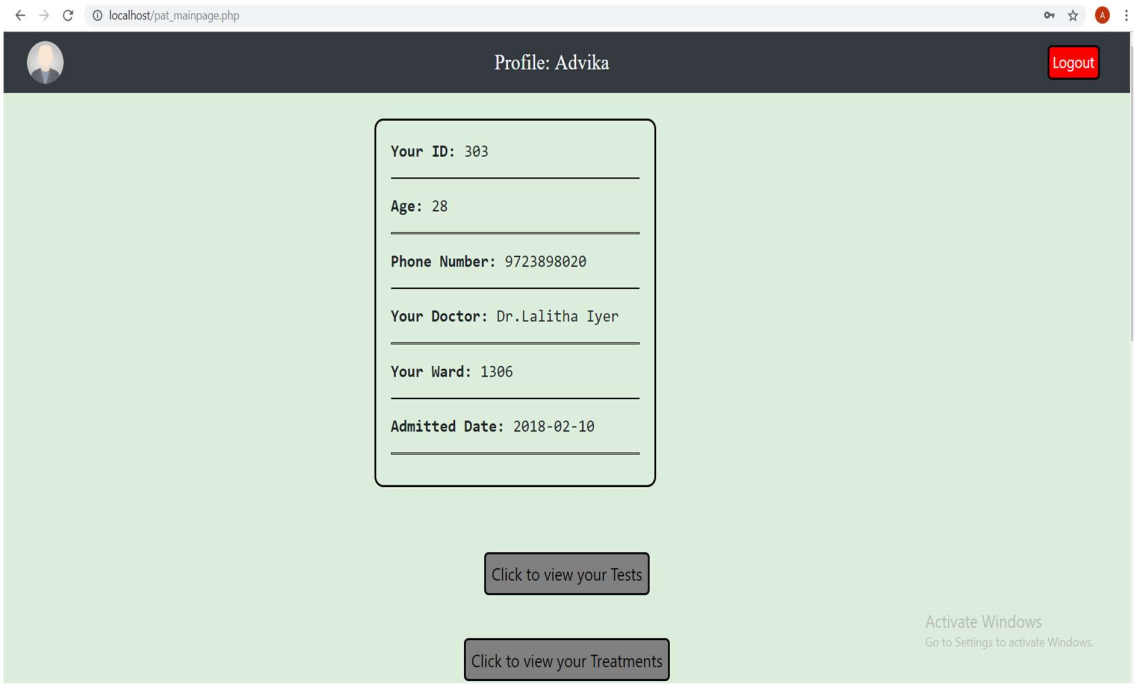
New test addition page.



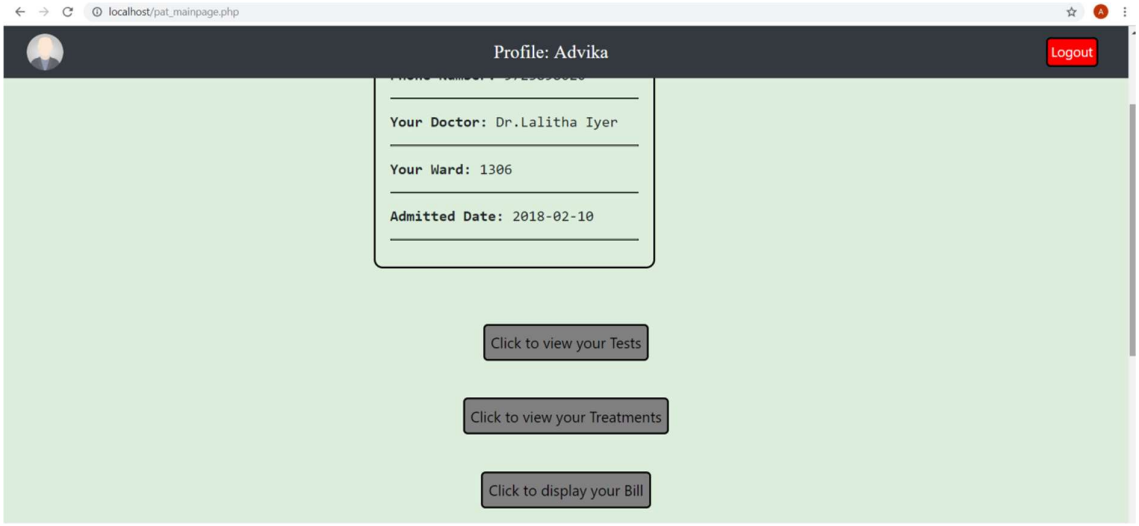
View test by entering patient ID by doctor.

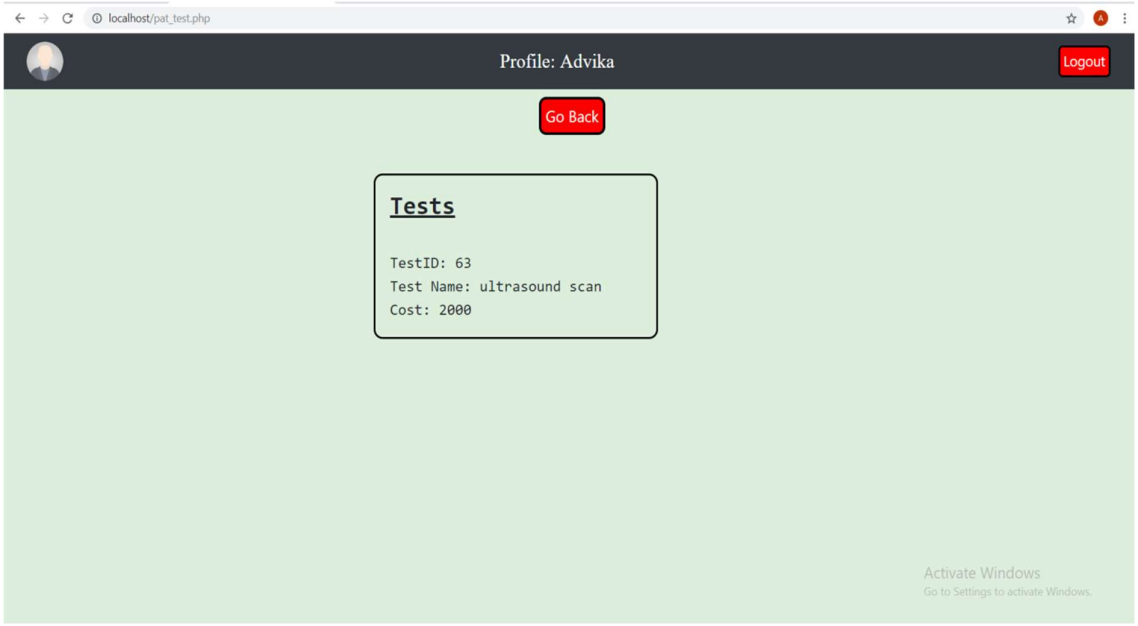


View treatment by entering patient ID by doctor.



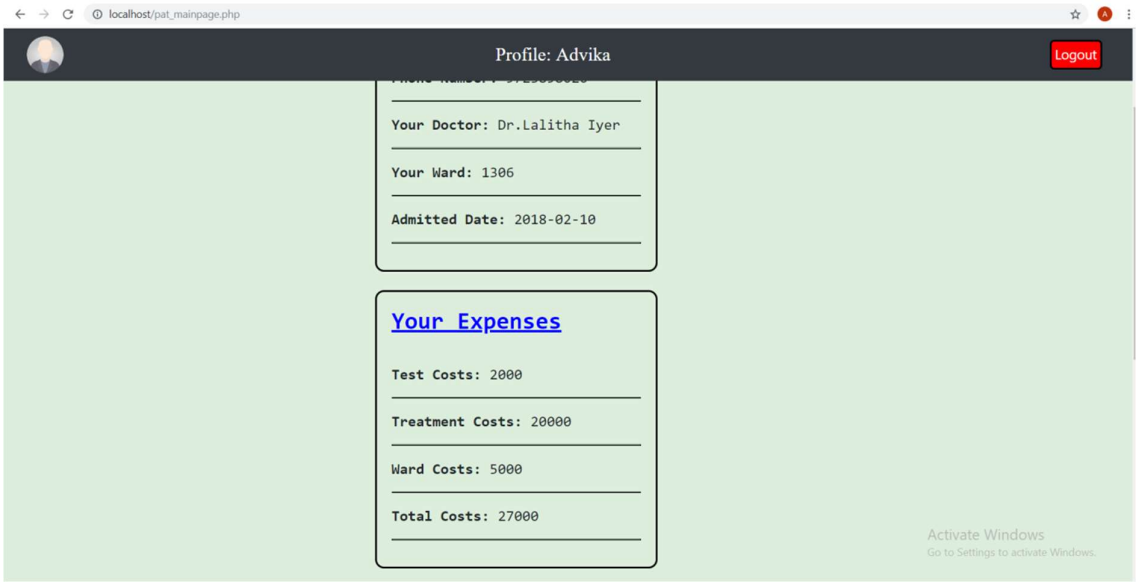
Patient Main Page.





View tests by patient.

View treatment by patient.



View bill by patient.

CHAPTER 7

APPLICATIONS

APPLICATIONS

Due the evolution of Database management system, companies are getting more from their work because they can keep records of everything. Also, it makes them faster to search information and records about any people or product that makes them more effective in work. A Database Management System (DBMS) provides us with an interface or a tool, to perform various operations like creating database, storing data in it, updating data, creating tables in the database and a lot more. Database Management Systems (DBMS) also provide protection and added security features to the databases. In addition, it also maintains data consistency in case of multiple users.

The applications of the Hospital Management System range from small nursing homes to super speciality hospitals which implies the scalability of the HMS application.

The HMS system can also be adopted in diagnostic centres with a few changes like exclusion of wards and treatment and including samples.

The HMS system can also be adopted without any changes in department specific hospitals like mental hospitals, maternity homes, cancer hospitals etc.

The HMS system's scope can be reduced so that it can be used in clinics of general physicians, eye clinics, ENT clinics and also dental clinics.

The working of HMS system can be enhanced by including the case history of patients which helps the doctors in determining the exact treatment for a particular patient. HMS system can be made more user friendly by providing online payment options for patients which not only helps the patients but also the management by making the service hassle-free.

The HMS system currently considers only in-patients. But it can be extended to include out-patients. With this modification, services like fixing the consultation time for out-patients can be done by taking into consideration the visiting of time of doctors.

For now, HMS is a small application of front-end and back-end clubbed together. HMS can be employed anywhere in the healthcare sector with the slightest of the modifications.

CHAPTER 8

CONCLUSION

CONCLUSION

The need for the Hospital to computerize the application processing and servicing the patients request through automated modules is very much necessary and now has become inevitable.

This project is a humble venture to provide a user-friendly application for the overall management of the hospital. An effort has been made to build an application to simplify the tasks of managing the hospital. The application hopes to make the lives easier for the people involved in hospital management.

Since the application is developed as a web application, it can be accessed anywhere on the globe. As the application has used an efficient database management system, there is no compromise on speed and also the security.

The HMS application enables the doctors to view all the patient related information in one place. This helps doctors to easily keep track of the treatment and progress in the health of the patient. The HMS system also provides doctor a user-friendly interface to add new treatments and tests required for the diagnosis of the disease. It also enables the doctors to add a new patient into the hospital. The HMS system thus saves time for the doctor by making the patient records management much easier and efficient.

The HMS application not only takes into consideration the doctors but also the patients and provides them a clean interface to view all their treatments and tests in just the single tap of a button. It also facilitates patients by displaying their detailed bill which is the stored procedure in the application.