**HOSPITAL MANAGEMENT SYSTEM**

**ABSTRACT**

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

**CHAPTER ONE**

**INTRODUCTION**

**1.1 Introduction:**

The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id.

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

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

**1.2 Problem Introduction:**

**Lack of immediate retrievals:**

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

**Lack of immediate information storage:**

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

**Lack of prompt updating:**

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

**Error prone manual calculation:**

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

**Preparation of accurate and prompt reports: -**

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

**Objective:**

1. Recording information related to diagnosis given to Patients.
2. Keeping record of the Immunization provided to children/patients.
3. Keeping information about various diseases and medicines available to cure them.

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

**Scope of the Project:**

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

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

**1.3 MODULES:**

The entire project mainly consists of 7 modules, which are

1. Admin module
2. User module
3. Doctor module
4. Laboratories module
5. Patient module

**CHAPTER TWO**

**REQUIREMENT SPECIFICATION**

**2.1 INTRODUCTION:**

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

**2.2 HARDWARE REQUIREMENTS:**

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

**HARDWARE REQUIREMENTS FOR PRESENT PROJECT**:

PROCESSOR : Intel dual Core ,i3

RAM : 1 GB

HARD DISK : 80 GB

**2.3 SOFTWARE REQUIREMENTS:**

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

**SOFTWARE REQUIREMENTS FOR PRESENT PROJECT:**

OPERATING SYSTEM : Windows 7/ XP/8

FRONT END : Html,css,java script.

SERVER SIDE SCRIPT : Php

DATABASE : Mysql

**CHAPTER THREE**

**ANALYSIS AND DESIGN**

**3.1 EXISTING:**

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

**3.2 PROPOSED SYSTEM:**

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

**3.3 FEASIBILITY STUDY**

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

Three key considerations involved in the feasibility analysis are:

**3.3.1 Economic Feasibility**

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

**3.3.2 Technical Feasibility**

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

**3.3.3 Operational Feasibility**

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

**3.4 SOFTWARE SPECIFICATION**

**HTML:**

**HTML** or **Hypertext Markup Language** is the standard [markup language](http://en.wikipedia.org/wiki/Markup_language) used to create [web pages](http://en.wikipedia.org/wiki/Web_page).

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

The purpose of a [web browser](http://en.wikipedia.org/wiki/Web_browser) is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website [semantically](http://en.wikipedia.org/wiki/Semantic) along with cues for presentation, making it a [markup language](http://en.wikipedia.org/wiki/Markup_language) rather than a [programming language](http://en.wikipedia.org/wiki/Programming_language).

HTML elements form the building blocks of all [websites](http://en.wikipedia.org/wiki/Website). HTML allows [images and objects](http://en.wikipedia.org/wiki/Img_(HTML_element)) to be embedded and can be used to create [interactive forms](http://en.wikipedia.org/wiki/Fieldset). It provides a means to create [structured documents](http://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](http://en.wikipedia.org/wiki/Semantic) for text such as headings, paragraphs, lists, [links](http://en.wikipedia.org/wiki/Hyperlink), quotes and other items. It can embed [scripts](http://en.wikipedia.org/wiki/Scripting_language) written in languages such as [JavaScript](http://en.wikipedia.org/wiki/JavaScript) which affect the behavior of HTML web pages.

**CASCADING STYLE SHEETS** (**CSS**):

It is a [style sheet language](http://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [look and formatting](http://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](http://en.wikipedia.org/wiki/Markup_language). While most often used to style [web pages](http://en.wikipedia.org/wiki/Web_page) and [interfaces](http://en.wikipedia.org/wiki/Interface_(computing)) written in [HTML](http://en.wikipedia.org/wiki/HTML) and [XHTML](http://en.wikipedia.org/wiki/XHTML), the language can be applied to any kind of [XML](http://en.wikipedia.org/wiki/XML) document, including [plain XML](http://en.wikipedia.org/wiki/Plain_Old_XML), [SVG](http://en.wikipedia.org/wiki/Scalable_Vector_Graphics) and [XUL](http://en.wikipedia.org/wiki/XUL). CSS is a cornerstone specification of [the web](http://en.wikipedia.org/wiki/The_web) and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the [layout](http://en.wikipedia.org/wiki/Page_layout), [colors](http://en.wikipedia.org/wiki/Color), and [fonts](http://en.wikipedia.org/wiki/Typeface).[[1]](http://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-1) This separation can improve content [accessibility](http://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content .

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

**MySQL:**

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

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

**REASONS FOR USING MySQL:**

* Leading open source RDBMS
* Ease of use – No frills
* Fast
* Robust
* Security
* Multiple OS support
* Free
* Technical support
* Support large database– up to 50 million rows, file size limit up to 8 Million TB

**JAVASCRIPT:**

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

**UML Design**

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

The UML is a language for:

1. Visualizing
2. Specifying
3. Constructing
4. Documenting

**Visualizing**

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

**Specifying**

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

**Constructing**

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

**Documenting**

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

**3.5 UML Approach**

**UML Diagram**

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

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

**3.5.1 USE CASE DIAGRAM:**

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

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

**Use case diagram:**

****

**3.5.2 Class Diagram**:

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



**3.5.3 Sequence diagram:**

A Sequence Diagramis an interaction diagram that emphasis the time ordering of messages; a collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other.

****

**3.5.4 Collaboration diagram:**

A Collaboration Diagramalso called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects. The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

****

**3.5.5 Deployment diagram:**

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

****

**3.5.6 Statechart Diagrams:**

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



**CHAPTER 4**

**IMPLEMENTATION**

This chapter details the development of the software while also discussing the results and tools that were used during this process. The application was built using python programming language and can be deployed on a Windows, Mac and Linux based system.

**Home Page:**

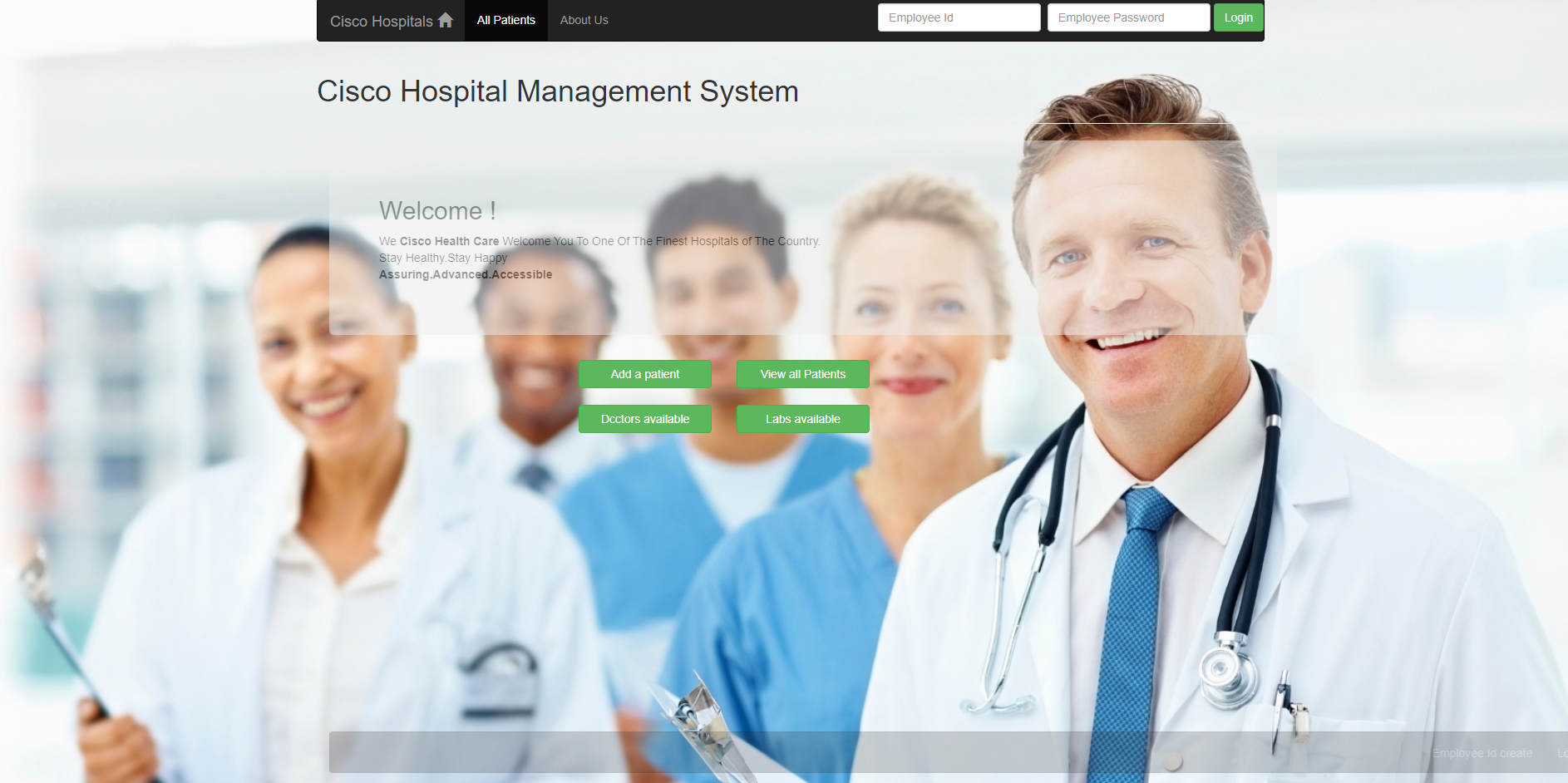
****

Figure 4.1

Image of the login Screen

**Adding New Patient Details:**

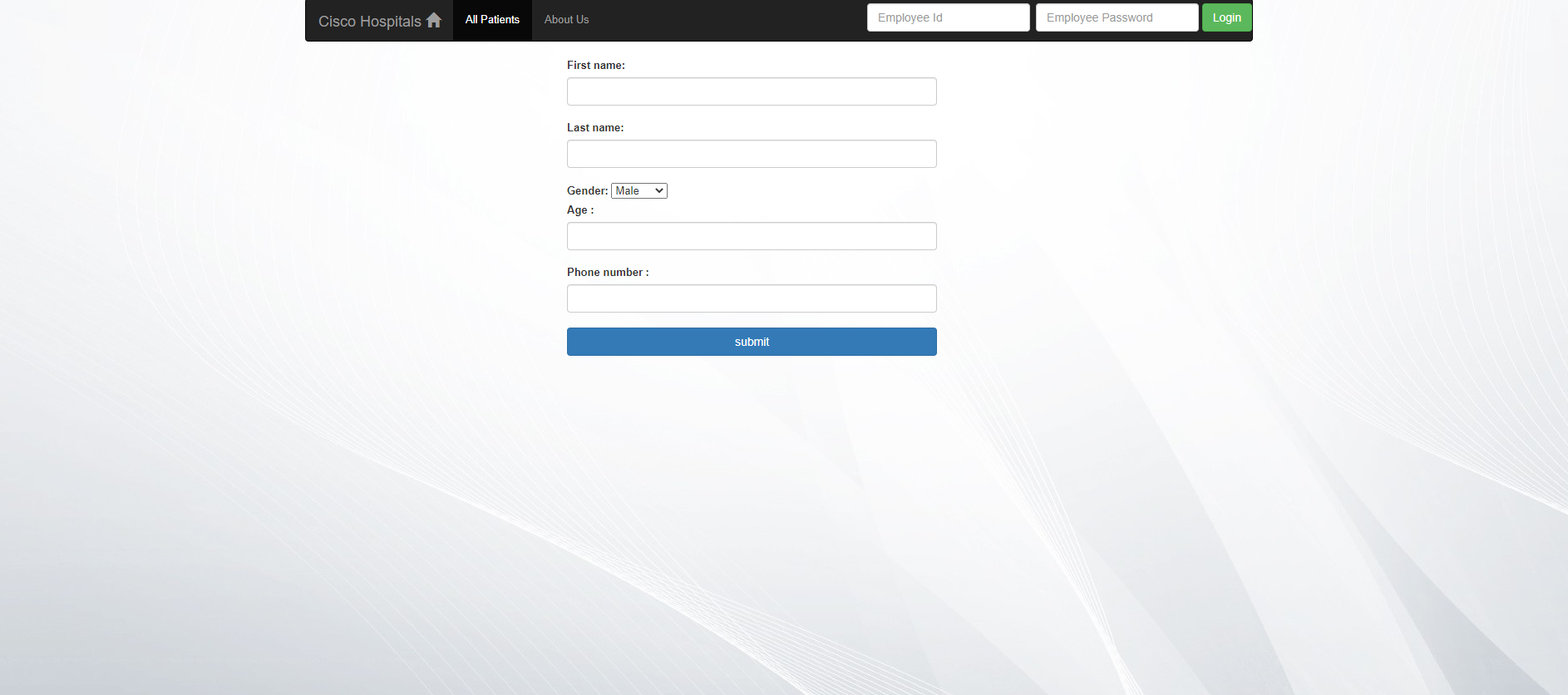
****

Figure 4.2

Adding a new patient Records

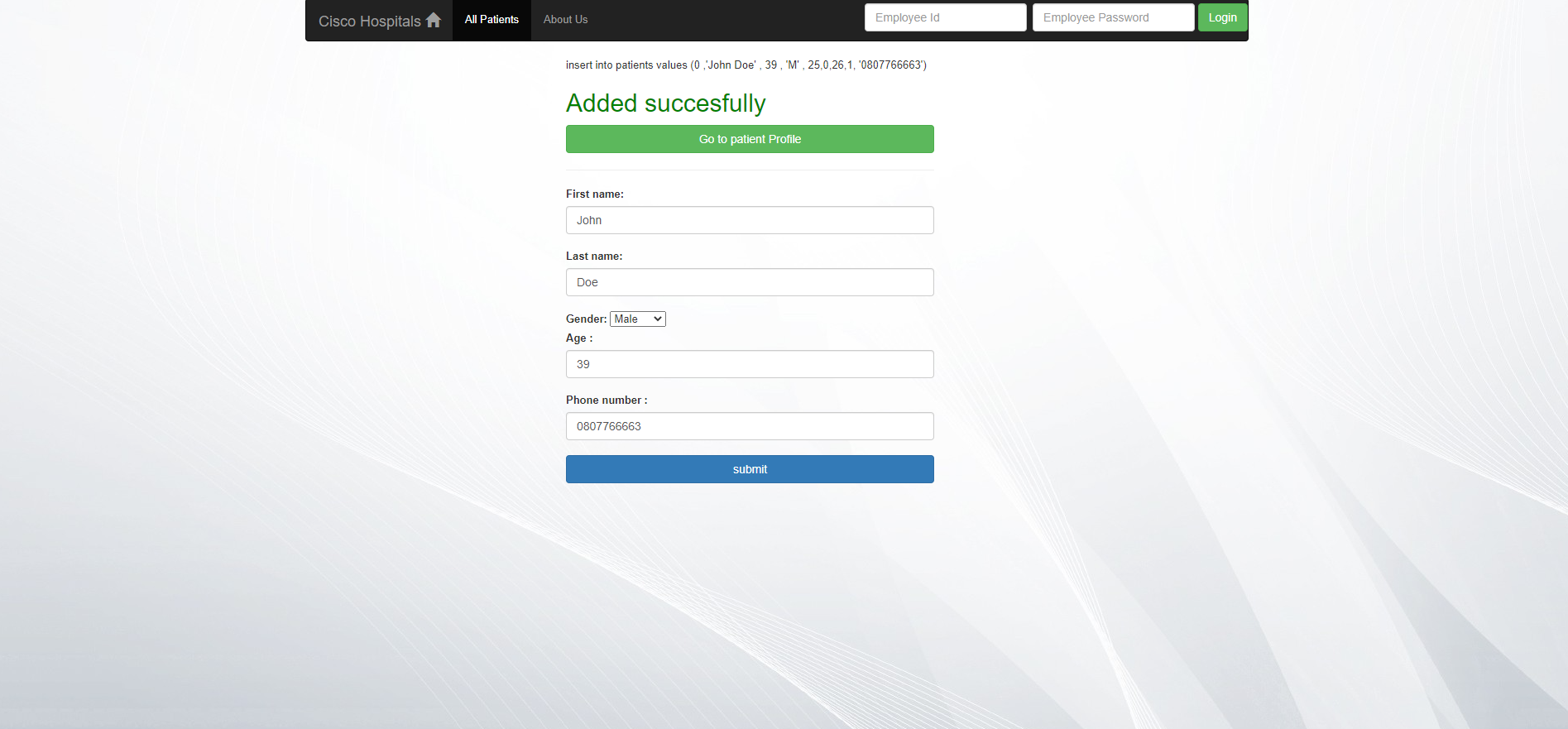
****

Figure 4.3

Patient Added to the System

**View Patient Details:**

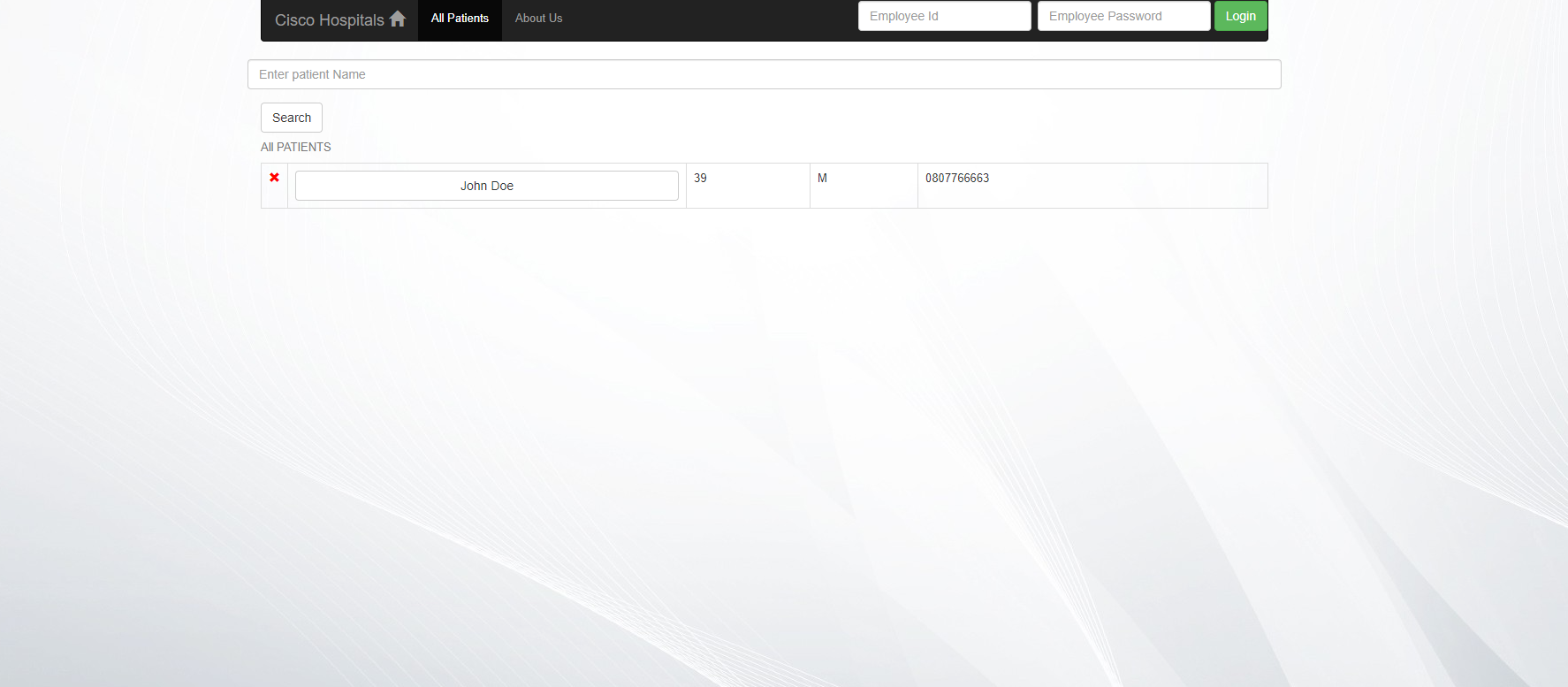
****

Figure 4.4

View Added Records

**Doctor List:**

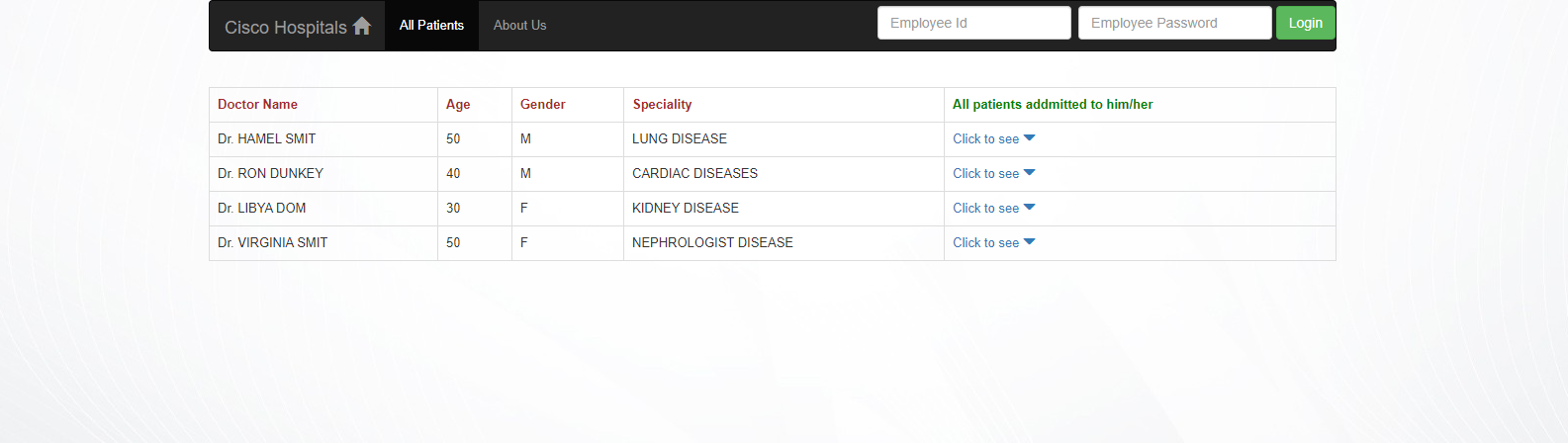
****

Figure 4.5

Doctors and their specialty

**Labs List**

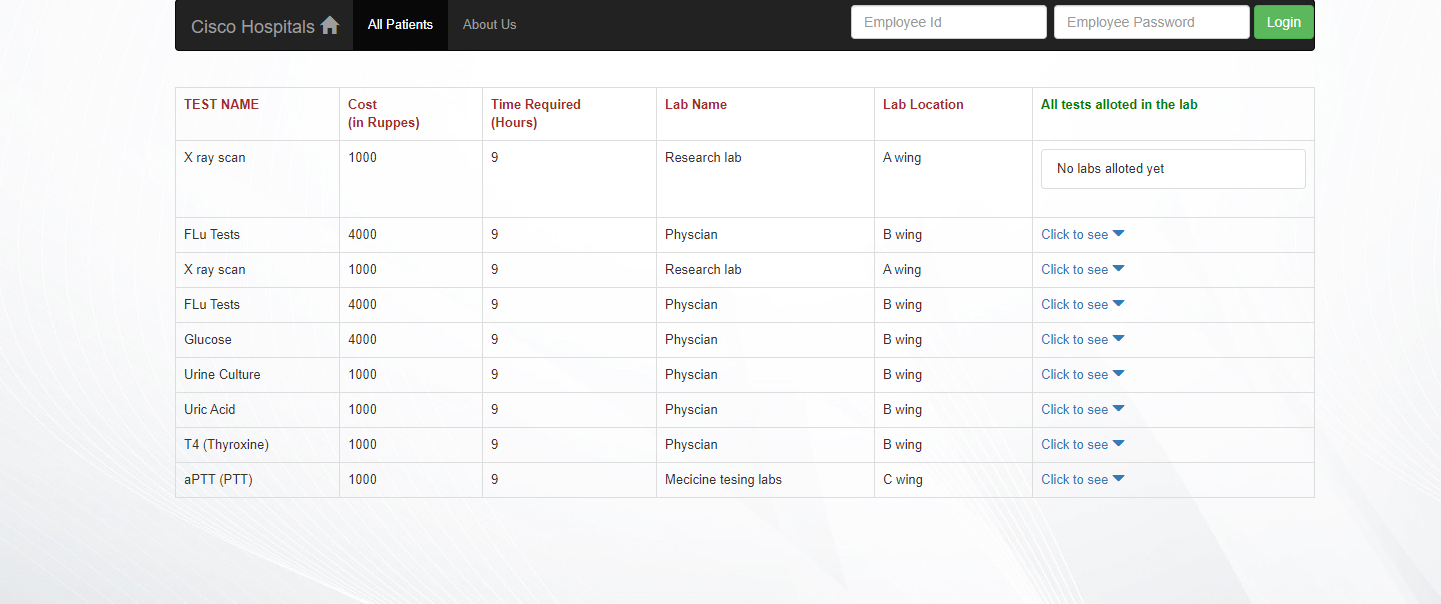
****

Figure 4.6

Lab records

**CHAPTER 5**

**INTRODUCTION**

This chapter details the development of the software while also discussing the results and tools that were used during this process. The application was built using python programming language and can be deployed on a Windows, Mac and Linux based system.

**CONCLUSION**

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

**RECOMMENDATION**

After the completion of this project the following recommendations were discovered:

1. The users of the system should be trained before total deployment.
2. Certain medical functionalities can be exposed to APIs for extension by opensource community.

**9. BIBLOGRAPHY**

1. PHP MySQL Website Programming: Problem - Design – Solution byChris

Lea, Mike Buzzard, Dilip Thomas , Jessey White-Cinis

2. Beginning PHP5, Apache, and MySQL Web Development (Programmer to

Programmer) by Elizabeth Naramore

3. MySQL/PHP Database Applications, 2nd Edition by Brad Bulger

4. How to Do Everything with PHP and MySQL by Vikram Vaswani

**Home.html:**

<!DOCTYPE html>

<html>

<body>

<table width="1350" height="640" border="1" >

<tr>

<td colspan="2" style="background-color:#FFF5EE;">

<h1>HOSPITAL MANAGEMENT SYSTEM</h1>

<h3 align="center">ADMIN PANEL</h3>

</td>

</tr>

<tr>

<td style="background-color:#00FFFF;width:50px;height:400px;">

<table align="center">

<tr><td><form action="doctor.php" align="center">

<input type="submit" align="center" value=" doctor ">

</form></td>

</tr>

<tr>

<td><form action="nurse.php" align="center">

<input type="submit" align="center" value=" nurse ">

</form></td>

</tr>

<tr>

<td><form action="patient.php" align="center">

<input type="submit" align="center" value=" patient ">

</form></td>

</tr>

<tr>

<td><form action="pharmacist.php" align="center">

<input type="submit" align="center" value=" pharamacist ">

</form></td>

</tr>

<tr>

<td><form action="laboratorist.php" align="center">

<input type="submit" align="center" value=" laboratorist ">

</form></td><tr>

<td><form action="accountant.php" align="center">

<input type="submit" align="center" value=" accountant ">

</form></td>

</tr>

</table>

</td>

<td style="background-color:#eeeeee;height:200px;width:400px;height:400px;"><h3 align="center">Advanced, powerfull, flexible complete management software for hospital, clinic and medical institutes. Integrates and facilitates all user area of a hospital: </h3><h4>align="center">Administrator</h4>

<h4 align="center">Doctor</h4>

<h4 align="center">Patient</h4>

<h4 align="center">Nurse</h4>

<h4 align="center">Pharmacist</h4>

<h4 align="center">Laboratorist</h4>

<h4 align="center">Accountant</h4>

</td>

</tr>

<tr>

<td colspan="2" style="background-color:#9ACD32;text-align:center;">

<table align="right">

<th>

<tr>

<form action="appointment.php" align="center">

<input type="submit" align="center" value=" appointment ">

</form>

</tr>

<tr>

<form action="payment.php" align="center">

<input type="submit" align="center" value=" payment ">

</form>

</tr>

<tr>

<form action="bloodbank.php" align="center">

<input type="submit" align="center" value=" bloodbank ">

</form>

</tr>

<tr>

<form action="medicine.php" align="center">

<input type="submit" align="center" value=" medicine ">

</form>

</tr>

<tr>

<form action="operations.php" align="center">

<input type="submit" align="center" value=" operations ">

</form>

</tr>

<tr>

<form action="birthreport.php" align="center">

<input type="submit" align="center" value=" birthreport ">

</form>

</tr>

<tr>

<form action="deathreport.php" align="center">

<input type="submit" align="center" value=" deathreport ">

</form>

</tr>

<tr>

<form action="bedallotment.php" align="center">

<input type="submit" align="center" value=" bedallotment ">

</form>

</tr>

</th>

</table>

</td>

</tr></table></body></html>

**Doctor.PHP**

<!DOCTYPE html>

<html>

<body>

<table width="1350" height="640" border="1" >

<tr>

<td colspan="2" style="background-color:#FFF5EE;">

<h1>HOSPITAL MANAGEMENT SYSTEM</h1>

<h3 align="center">ADMIN PANEL</h3>

</td>

</tr>

<tr>

<td style="background-color:#00FFFF;width:50px;height:400px;">

<table align="center">

<tr>

<td><form action="nurse.php" align="center">

<input type="submit" align="center" value=" nurse ">

</form></td>

</tr>

<tr>

<td><form action="patient.php" align="center">

<input type="submit" align="center" value=" patient ">

</form></td>

</tr>

<tr>

<td><form action="pharmacist.php" align="center">

<input type="submit" align="center" value=" pharamacist ">

</form></td>

</tr>

<tr>

<td><form action="laboratorist.php" align="center">

<input type="submit" align="center" value=" laboratorist ">

</form></td>

<tr>

<td><form action="accountant.php" align="center">

<input type="submit" align="center" value=" accountant ">

</form></td>

</tr>

</table>

</td>

<td style="background-color:#eeeeee;height:200px;width:400px;height:400px;">

<?php

$host='localhost';

$username='root';

$password='';

$dbname='hospital';

$con=mysql\_connect($host,$username,$password);

mysql\_select\_db($dbname);

$result = mysql\_query("SELECT \* FROM doctor");

echo "<h4 align='center'> doctors list </h4>";

echo "<table border=1 align=center><tr><th>s.no</th><th>name</th><th>d\_id</th><th>qualification</th><th>speciality</th><th>age</th></tr>";

while($row = mysql\_fetch\_array($result))

{

echo "<tr>";

echo "<td>" . $row['s\_no'] . "</td>";

echo "<td>" . $row['name'] . "</td>";

echo "<td>" . $row['d\_id'] . "</td>";

echo "<td>" . $row['qualification'] . "</td>";

echo "<td>" . $row['speciality'] . "</td>";

echo "<td>" . $row['age'] . "</td>";

echo "</tr>";

}

echo "</table>";

mysql\_close($con);

?>

<br><br>

<table align="right">

<th>

<tr>

<form action="adddoctor.php" align="center">

<input type="submit" align="center" value=" add new doctor ">

</form>

</tr>

<tr>

<form action="deletedoctor.php" align="center">

<input type="submit" align="center" value=" delete doctor ">

</form></tr>

<tr>

<form action="viewcompletedoctor.php" align="center">

<input type="submit" align="center" value=" viewcomplete ">

</form>

</tr>

<tr>

<form action="admin.html" align="center">

<input type="submit" align="center" value=" home ">

</form>

</tr></table>

</td>

</tr><tr>

<td colspan="2" style="background-color:#9ACD32;text-align:center;">

<table align="right">

<th>

<tr><form action="appointment.php" align="center">

<input type="submit" align="center" value=" appointment ">

</form>

</tr><tr><form action="payment.php" align="center">

<input type="submit" align="center" value=" payment ">

</form>

</tr><tr><form action="bloodbank.php" align="center">

<input type="submit" align="center" value=" bloodbank ">

</form>

</tr><tr>

<form action="medicine.php" align="center">

<input type="submit" align="center" value=" medicine ">

</form>

</tr><tr><form action="operations.php" align="center">

<input type="submit" align="center" value=" operations ">

</form>

</tr><tr>

<form action="birthreport.php" align="center">

<input type="submit" align="center" value=" birthreport ">

</form>

</tr><tr><form action="deathreport.php" align="center">

<input type="submit" align="center" value=" deathreport ">

</form>

</tr><tr><form action="bedallotment.php" align="center">

<input type="submit" align="center" value=" bedallotment ">

</form>

</tr></th> </table>

</td></tr></table>

</body>

</html>

**Appointment.php**

<!DOCTYPE html>

<html>

<body>

<table width="1350" height="640" border="1" ><tr>

<td colspan="2" style="background-color:#FFF5EE;">

<h1>HOSPITAL MANAGEMENT SYSTEM</h1>

<h3 align="center">DOCTOR PANEL</h3>

</td>

</tr>

<tr>

<td style="background-color:#00FFFF;width:50px;height:400px;">

<table align="center">

<tr> <td><form action="docappointment.php" align="center">

<input type="submit" align="center" value=" Appointment ">

</form> </td></tr>

<tr> <td><form action="docperscription.php" align="center">

<input type="submit" align="center" value=" perscription ">

</form> </td> </tr>

<tr> <td> <form action="docoperation.php" align="center">

<input type="submit" align="center" value=" Operation ">

</form> </td></tr>

<tr> <td><form action="docmedicines.php.php" align="center">

<input type="submit" align="center" value=" Add Medicines ">

</form></td></tr>

<tr> <td> <form action="doctests.php" align="center">

<input type="submit" align="center" value=" Add Tests ">

</form></td>

</table>

</td>

<td style="background-color:#eeeeee;height:200px;width:400px;height:400px;">

<h2 align="center"> Appointments </h2>

<?php

$host='localhost';

$username='root';

$password='';

$dbname='hospital';

$con=mysql\_connect($host,$username,$password);

mysql\_select\_db($dbname);

$result = mysql\_query("SELECT \* FROM appointment WHERE d\_id='$a'");

echo "<table border=1 align=center> <tr> <th>s.no</th> <th>pid</th> <th>name</th> <th>problem</th> <th>date</th> <th>time</th> <th>status</th> <th> update</th> </tr>";

while($row = mysql\_fetch\_array($result))

{

echo "<tr>";

echo "<td>" . $row['s\_no'] . "</td>";

echo "<td>" . $row['p\_id'] . "</td>";

echo "<td>" . $row['name'] . "</td>";

echo "<td>" . $row['problem'] . "</td>";

echo "<td>" . $row['date\_of\_app'] . "</td>";

echo "<td>" . $row['time\_of\_app'] . "</td>";

echo "<td>" . $row['status'] . "</td>";

echo "<td>" ;?> <form action="updateappointment.php" align="center" method="POST">

<input type="hidden" name="sno" value=" <?php echo $row['s\_no']; ?> ">

<input type="hidden" name="pid" value=' <?php echo $row['p\_id']; ?> '>

<input type="submit" align="center" value=" update ">

</form> <?php echo "<td>";

echo "</tr>";

}

echo "</table>";

mysql\_close($con);

?>

<br><br>

<table align="center">

<tr>

<td><form action="allappointment.php" align="center">

<input type="submit" align="center" value=" all Appointment ">

</form> </td>

<td><form action="pendingappointment.php" align="center">

<input type="submit" align="center" value=" pending Appointment ">

</form> </td>

<td> <form action="upcomingappointment.php" align="center">

<input type="submit" align="center" value=" upcoming appointment ">

</form> </td>

<td><form action="completedappointment.php" align="center">

<input type="submit" align="center" value=" completed Appointment ">

</form></td></table>

</td></tr>

<tr>

<td colspan="2" style="background-color:#9ACD32;text-align:center;">

<table align="center"> <tr> <td> Doctor name </td> <td> </td> <td> Doctor id </td> <td> </td> </tr> </table>

</td></tr>

</table></body></html>