

University of Calcutta

A Project Report on

<<HOSPITAL MANAGEMENT SYSTEM>>

A project submitted as part of fulfillment of B.Sc(Hons.) in Computer Science,
Semester -VI, 2021 in the department of Computer Science, T.H.K Jain
College

Under the Supervision of
<<Prof. Suvadip Das>>
Department of Computer Science
T.H.K Jain College

Submitted by-
Name:Nidhi Jaiswal
Roll No: 183235-11-0013
Registration No: 235-1211-0142-18

Name:Dimpi Gupta
Roll No: 183235-11-0107
Registration No: 112-1211-0251-17



CERTIFICATE

This is to certify that the project entitled "**HOSPITAL MANAGEMENT SYSTEM**" submitted for the requirements of 6th Semester of B.Sc in Computer Science (Honours), University Of Calcutta; has been carried out by **Nidhi Jaiswal (Roll No:- 183235-11-0013 and Registration No:- 235-1211-0142-18) & Dimpi Gupta(Roll No:- 183235-11-0107 and Registration No:- 112-1211-0251-17)** under the supervision of Prof. Suvidip Das, T.H.K JAIN COLLEGE , University of Calcutta.

A handwritten signature of Prof. Suvidip Das.

Prof. Suvidip Das
Project Supervisor
Dept. of Computer Science
T.H.K Jain College

A handwritten signature of Prof. Rituparna Kundu.

Prof. Rituparna kundu
Department In-Charge
Dept. of Computer Science
T.H.K Jain College

A handwritten signature of the External Examiner.

External Examiner

ACKNOWLEDGEMENT

We wish to express our profound sense of gratitude of our project supervisor Prof. Suvadip Das, T.H.K Jain College, University of Calcutta for his support, inspiration and guidance. He has showed us different ways to approach a problem. We have also learned from him that an approach needs to be persistent to accomplish our goal. We are immensely grateful to him for giving his valuable time and constant advice for discussing various ideas related to our project work. It is being precious learning experience for us to work under tutelage.

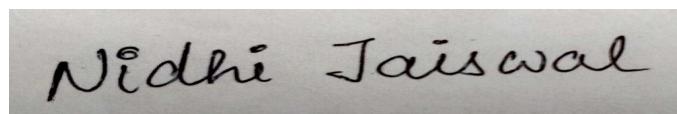
We are also thankful to our department; Department of Computer Science, T.H.K Jain College, University of Calcutta; for providing us with the required resources for working on this project.

Lastly we like to express our heartiest gratitude to our parents, seniors and our friends; and to all who have directly or indirectly extended their valuable guidance and advice during the preparation of this project; which will give us the continuous flow of inspiration to complete the project.

Date:- 7.08.2021

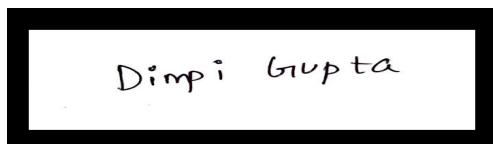
Name(s):

1. NIDHI JAISWAL



Nidhi Jaiswal

2. DIMPI GUPTA



Dimpri Gupta

CONTENT

Chapter No	Chapter Name	Page No
1	Introduction	7-8
2	RDBMS	9-10
3	Study about different Back End and Front End Applications	11-13
4	Important constructs used regarding Mysql and PHP	14-17
5	Problem Definition	18-19

6	Related Work	20
7	Objective	21
8	Scope	22
9	Proposed Algorithm to implement the “Automated Hospital Management System”	23-26
10	Proposed Software Design	27-29
11	Case Studies	30-63
12	Conclusion	64
13	Future Work	65
14	References	66

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. The goal of any system development is to develop and implement the system cost effectively; user-friendly and most suited to the user’s analysis is the heart of the process. Traditionally, it was done manually. 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. Our software has the facility to give a unique id for every patient and stores the details of every patient, doctor and the staff automatically. It includes a search facility to know the current status of each department or building. User can search availability of a doctor and the details of a patient using the id. The System also contains Special Corona Bed Booking for the Covid Positive Patients for the Booking of Bed. The data are well protected for personal use and makes the data processing very fast.

CHAPTER-1

INTRODUCTION

Hospital management system was introduced with the cause for helping hospitals speed up their processes. The System's aim is to develop a computerized system to maintain all the information of Hospital Entities. Hospital management system manages the data related to all departments of healthcare.

Hospital Management System was introduced to solve the complications coming from managing all the paper works of every patient associated with the various departments of hospitalization with confidentiality.

Hospital Management System provides the ability to manage all the paperwork in one place, reducing the work of staff in arranging and analyzing the paperwork of the patients.

1.1 Function

The main function of the system is to 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, doctor details, addmission details, diagnosis details, billing details, Icu details, department details, building details, staff details while system output is to get these details on to the screen. It includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs.

1.1.1 Functionality of the Automated System

Hospital Management System maintains the medical records of the patient, maintain the contact details of the patient, keep track of the appointment dates, save the insurance information for later reference, tracking the bill payments.

1.2 Advantages

Hospital Management System is time-saving Technology, Improved Efficiency by avoiding human errors, Reduces scope for Error, Data security and correct data retrieval made

possible,Cost effective and easily manageable,Easy access to patient data with correct patient history,Improved patient care made possible,Easy monitoring of supplies in inventory,Reduces the work of documentation,Better Audit controls and policy compliance.

1.3 Summary

Hospital Management System makes it possible to access all the data related to a patient via a system by the means of a few simple clicks. Information like patient history, current illness, doctors involved, tests reports taken, billing information and many more can be made visible to the user. These data will help to connect the dots about the patient, like specific diagnosis, related treatment, and medication.Overall,this project is developed to help the hospital administration,maintain the hospital management in the best way possible and also reduce the human efforts.

CHAPTER-2
***RELATIONAL DATABASE
MANAGEMENT SYSTEM(RDBMS)***

2.1 Manual System

As we know that, manual handling of the record is time consuming and highly prone to error. Not only we have to organize and store the files, hunting down the information when it is need can take time. When we are working with paper documents it is much harder to make changes. Paper document filing can be less secure than electronic filing systems. To overcome this issue we use file processing system.

2.2 File Processing System

It is possible to take faster and automatic back-up of database stored in files, store data compactly, access data remotely and retrieve data stored in files in easy and efficient way. It is easy to edit any information stored in computers in form of files. It is not necessary for a user to remain present at location where these data are kept. Data stored in files of can be shared among multiple users at a same time.

2.2.1 Demerits of File Processing System

It results in data redundancy and inconsistency, difficult in accessing data, data isolation, data integrity, concurrent access is not possible and have security problems.

To overcome all this difficulties we use database management system to manage the records online with having all this problems

2.3 Database Management System

Database Management System(DBMS) is a computerized system, whose overall purpose is to maintain the information and it is a collection of programs that enables user to create and maintain a database, i.e., it provides the users with the processes of defining, constructing and manipulating the database for various applications.

2.3.1 Advantages over File Processing System

In DBMS, redundancy can be reduced, inconsistency can be avoided, data can be shared, standards can be enforced, security restrictions can be applied, integrity can be maintained, data gathering can be possible, requirements can be balanced.

2.4 Relational Database Management System

RDBMS Stands for "Relational Database Management System."

An RDBMS is a DBMS designed specifically for relational databases. It is relational model in which data is stored in multiple tables where tables are related to each other using primary keys and foreign keys and indexes. A relational database refers to a database that stores data in a structured format, using rows and columns. This makes it easy to locate and access specific values within the database. It is "relational" because the values within each table are related to each other.

2.4.1 Functionality of RDBMS

It uses database normalization techniques to avoid redundancy in tables. It helps to fetch data faster using SQL query. It is widely used by enterprises and software developers to store large amount of complex data. Examples: SQL server, Oracle, MySQL, MariaDB, SQLite.

The relational structure makes it possible to run queries across multiple tables at once. While a relational database describes the type of database an RDBMS manages, the RDBMS refers to the database program itself. It is the software that executes queries on the data, including adding, updating, and searching for values. An RDBMS may also provide a visual representation of the data.

2.4.2 Practical Importance of RDBMS

RDBMS provides multiple interfaces with better backup and recovery procedures. Multiple users can access the database which is not possible in DBMS. Proper database management systems help increase organizational accessibility to data, which in turn helps the end users share the data quickly and effectively across the organization. A management system helps get quick solutions to database queries, thus making data access faster and more accurate.

CHAPTER-3

STUDY ABOUT DIFFERENT BACKEND RDBMS AND FRONT END APPLICATIONS

3.1 Introduction

There are different types of database management systems, but the relational database management system (RDBMS) is the most popular type of DBMS. With the right RDBMS, companies can create and manage relational databases with all kinds of structured data such as employee data, customer data, sales data, phone numbers, census records, legal records, and library catalogs.

3.2 Front-end

It is ‘User-side’ application which provides an interface to fetch, store and display data. Data inputted here and sent to backend database.

3.2.1 Some Frontend Applications

3.2.1.1. HTML

HTML stands for Hypertext Markup Language. It is used to design the front-end portion of web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text documentation within the tag which defines the structure of web pages.

3.2.1.2. CSS

Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

3.2.1.3 JavaScript

JavaScript is a famous scripting language used to create magic on the sites to make the site interactive for the user. It is used to enhancing the functionality of a website to running cool games and web-based software.

3.3 Cross Platform

3.3.1 XAMP

XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the Ps s.tand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl.

3.3.2 Platforms used for Frontend and backend connections

3.3.2.1 PHP

It is the backend scripting language primarily used for web development. PHP allows users to create dynamic websites and applications. It can be installed on every platform and supports a variety of database management systems

3.3.2.2 MySQL

It is a cross-platformed, leading open source and free relational database management system which is dominating the software industry for the past 25 years.

It is super fast and easy to use, having high performance for large databases (more than 50 million rows in a table).

3.3.2.3 MariaDB

MariaDB Server is one of the most popular free open-source relational databases made by the original developers of MySQL and guaranteed to stay open source.

3.3.2.4 MongoDB

MongoDB stores data in JSON-like documents which is much more expressive and powerful than the traditional row-column model. Though MongoDB is a NoSQL database it has all the power of a relational database and even more.

3.4 Backend

It is ‘Server-side’ database which stores data (tables) and is connected to front-end through an API.

3.4.1 Some Backend Database

3.4.1.1 MariaDB

It is an enhanced drop-in replacement for MySQL and a powerful database server made for MySQL developers providing a platform for turning data into structured information by using a wide array of features.

3.4.1.2 Db2 Express-C

It is a free community edition of Db2 database software that has a fast setup, easy to use, ideal for academia, small businesses and multi-branch companies.

3.4.1.3 SQLite

It is the most widely deployed database in the world with very many applications which includes very many high-profile projects and it implements a self-contained, server less, zero configuration and transactional SQL database engine.

3.4.1.4 Firebird

It is an open source and relational database that provides excellent performance and scales impressively by use of embedded and single user model that provides many SQL and ANSI features

3.4.1.5 Oracle Database XE

It is an entry-level, small-footprint database on the Oracle Database 11g Release 2 code base.

3.4.1.6 Sequel Pro

It is a fast, easy to use Mac database management application that works with MySQL databases.

3.4.1.7 PostgreSQL

It is a powerful open source object-relational database system. It has a strong reputation in terms of reliability, data integrity, and correctness.

3.4.1.8 CUBRID

It is an open source and free database engine that provides high level of SQL capability along with MySQL and also features built-in enterprise grade features such as high performance data caching.

3.4.1.9 MySQL

It is the world's most popular open source database. It's the leading database choice for web-based applications. It supports SQL as its database language. It has views that relate to an SQL query that is viewed as a distinct database object and makes it possible for one to view a particular part of a database.

3.4.1.10 SQL Server Express

It is a free database software that works great with any web application framework like PHP, and applications such as WordPress, and Drupal and can be used to store and access the information stored in many different databases.

3.5 Implementation platform for our project

The Front end and Backend used in our project are html,css, javascript and php and MySQL for Hospital Management System. PHP combined with MySQL are cross-platform (we can develop in Windows and serve on a Unix platform). With PHP, we can connect to and manipulate databases. MySQL is the most popular database system used with PHP.

3.5.1 Reason behind using these platforms

HTML and CSS are at the heart of Front End development. These languages are relatively easy to learn and offer plenty of flexibility and creativity. JavaScript is a very important tool for a front-end web developer. Without it, webpages wouldn't have become the dynamic web applications they are today.

MySQL is a first choice of PHP developers. As an open source Relational Database Management System (RDBMS) that uses SQL language, MySQL database helps to automate data retrieving and provide great support in PHP MySQL web application development.

CHAPTER-4

IMPORTANT CONSTRUCTS USED REGARDING MYSQL AND PHP

4.1 Introduction

Mysql database set up with a database user and password and a web development environment using mysql or mariadb, apache and php and a simple text editor.

It establishes a mysql connection using php on a web page, connecting to a mysql table and retrieving the results and displaying them back on the web page.

4.2 PHP syntax:

4.2.1 mysqli_connect

Functionality:

connecting to a MySQL database.

Syntax:

```
$db = mysqli_connect('localhost','root','root','database_name');
```

4.2.2 mysqli_query

Functionality:

performing a database query.

Code Syntax:

```
$query = "SELECT * FROM table_name";  
mysqli_query($db, $query);
```

4.2.3 mysqli_fetch_array

Functionality:

Adding the table and rows to the Database.

Code Syntax:

```
$result = mysqli_query($db, $query);  
$row = mysqli_fetch_array($result)
```

4.2.4 mysqli_close

Functionality:

closing off the connection.

Code Syntax:

```
mysqli_close($db);
```

4.3 MySQL commands:

4.3.1 Create Database

Functionality:

creates a new database

Code Syntax:

```
CREATE DATABASE databasename;
```

4.3.2 Alter Database

Functionality:

modifies a database.

Code Syntax:

```
ALTER DATABASE database_name
```

```
[COLLATE collation_name ]
```

4.3.3 Create Table

Functionality:

creates a new table.

Code Syntax:

```
CREATE TABLE table_name (
```

```
column_1 datatype,
```

```
column_2 datatype,
```

```
column_3 datatype );
```

4.3.4 Alter table

Functionality:

This query add, delete, or modify columns in an existing table.

Code Syntax:

```
ALTER TABLE table_name
```

```
ADD column_name datatype;
```

4.3.5 Insert Record

Functionality:

inserts new data into a database

Code Syntax:

```
INSERT INTO table_name (column_1, column_2, column_3)
```

```
VALUES (value_1, 'value_2', value_3);
```

4.3.6 Update Record

Functionality:

updates data in a database

Code Syntax:

```
UPDATE table_name SET some_column = some_value WHERE some_column some_value;
```

4.3.7 Delete Record

Functionality:

deletes data from a database

Code Syntax:

```
DELETE FROM table_name
```

```
WHERE some_column = some_value;
```

4.3.8 Select Record

Functionality:

extracts data from a database.

Code Syntax:

```
SELECT * FROM table_name;
```

4.3.9 Drop Table

Functionality:

deletes a table.

Code Syntax:

```
DROP TABLE table_name;
```

4.4 Css commands:

4.4.1 The linear-gradient() function: It is an inbuilt function in CSS which is used to set the linear gradient as the background image.Syntax: `background-image: linear-gradient(direction, color1, color2, ...).`

4.4.2 Selector – A selector is an HTML tag at which a style will be applied. This could be any tag like `<h1>` or `<table>` etc.

4.4.3 Property – A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.

4.4.4 Value – Values are assigned to properties. For example, color property can have value either red or `#F1F1F1` etc.

4.4.5 Colors in CSS: In this CSS commands, colors can be specified in RGB formula. Each parameter defines the intensity of these colors and defines a new color. For example, to display black all color parameters should be set to `RGB(0,0,0)`.

4.5 Methodology for frontend to backend connection

1. Make the connection.
2. Select the database.
3. perform the query on the table.

4. print out the data.
5. close the connection.

4.6 Quickview for php and MySQL connection:

```
<?php  
$db = mysqli_connect('localhost','root','root','database_name'); //connecting to a database  
?>  
<?php  
$query = "SELECT * FROM table_name"; //display table data  
mysqli_query($db, $query); //performing database query  
$result = mysqli_query($db, $query); //store it in result  
while ($row = mysqli_fetch_array($result)) {  
echo $row['FirstName'] . ' ' . $row['LastName'] . ':' . $row['Email'] . ' ' . $row['City'] . '<br />'; }  
mysqli_close($db);?> //close the connection
```

CHAPTER-5

PROBLEM DEFINITION

5.1 Introduction

Since Hospital is associated with the lives of common people and their day-to-day routines so we decided to work on this project.

The manual handling of the record is time consuming and highly prone to error. So, to overcome this issue file oriented system had come to use but it has also some of the issues which cannot maintain system properly. So, DBMS came into use which overcome all the issues which was faced in manual handling and file oriented system.

5.2 Main problems of storing data manually and file systems

The information is very difficult to retrieve and to find particular information like for example, to find out about the patient's history, the user has to go through various registers. This results in inconvenience and wastage of time. The information generated by various transactions takes time and efforts to be stored at right place.

Various changes to information like patient details is difficult to make as paper work is involved. 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.

In the File System, Duplicate data is created in many places because all the programs have their own files which create data redundancy resulting in wastage of memory. Users have to back up the files in regular intervals which lead to waste of time and resources.

5.3 Advantages of using Database Management System

To solve all the problems faced by manual handling and file oriented system, DBMS is used to overcome all this issue.

The advantages of using DBMS are:

5.3.1 Data security level

Data can be shared between authorized users of the database in DBMS. Any unauthorized access is restricted.

5.3.2 Data redundancy

In DBMS, all the files are integrated in a single database. So there is no chance of duplicate data.

5.3.3 Data integrity

Data integrity makes unification of so many files into a single file. DBMS allows data integrity which makes it easy to decrease data duplicity. DBMS allows multiple users to access the same database at a time without any conflicts.

5.3.4 Data consistency

Data consistency is nothing but if we want to update data in any files then all the files should not be updated again. In DBMS, data is stored in a single database so data becomes more consistent in comparison to file processing systems.

5.3.5 Backup

DBMS solves the problem of taking backup automatically and recovery of the database.

5.4 Purpose

The purpose of this project is to automate or make online, the process of day-to-day activities like Room activities, Admission of New Patient, Discharge of Patient, Assign a Doctor, and finally compute the bill etc. We have tried our best to make the complicated process Hospital Management System as simple as possible using Structured & Modular technique & Menu oriented interface. We have tried to design the software in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. Even though we cannot claim that this work to be entirely exhaustive, the main purpose of this is to perform each Hospital's activity in computerized way rather than manually which is time consuming.

CHAPTER-6

RELATED WORK

1.Reducing the of Child Abuse: Impact of a New Hospital Management System

Eli H. Newberger, M.D., John J. Hagenbuch, A.C.S.W., Nancy B. Ebeling, A.C.S.W., Elizabeth Pivchik

Colligan, A.C.S.W., Jane S. Sheehan, R.N., and Susan H. McVeigh, B.A.

2.ThedaCare's Business Performance System: Sustaining Continuous Daily Improvement Through Hospital Management in a Lean Environment

Kim barnas.

3.A Study of Advanced Hospital Managaement System

Kumaran S,Dr.Drusphagaran,Kalai Selvi3,Christopher,Deepak.

CHAPTER-7

OBJECTIVE

7.1 Introduction

Hospital are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc.

It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, staff, building, department, etc, that keep the hospital running smoothly & successfully.

7.2 Objective of our Project

In our Hospital Management System project, we have first designed the Entity Relationship Diagram (ERD) and Data Flow Diagram (DFD). Then we have considered all the hospitals data in terms of domain such as patient details, doctor details, admission details, diagnosis details, billing details, Icu details, department details, building details and staff details which are controlled by this section and with the respect of all domain we have defined the Er diagram and Data flow diagram.

Then one by one we have designed this in the backend database MYSQL and then we have developed front end application by using HTML,css,php and javascript for performing several operations such as 'Insert','Delete','Update','View' for this system by considering two domain admin section and patient section where all the operations in the frontend has been accessed by the admin. Then we have performed several queries and Join queries in for associating more than one table to view them in frontend. lastly, we have drawn conclusion by designing this whole Hospital Management System having an excellent future scope.

We are confident that this software package can be readily used by non-programming personal avoiding human handled chance of error.

CHAPTER-8

SCOPE

8.1 Introduction

The proposed software product is the Hospital Management system (HMS). It is not intended to particular organisation. This will develop a generic software, which can be applied by any business organisation. The system will be used in any hospital, clinic, dispensary or pathology labs. Clinic, dispensary or pathology to get the information from the patients and then storing that data for future usages.

The current system in use is a paper based system. It is too slow and cannot provide updated lists of patients within reasonable time-frame. keeping track of all the activities and their records on paper is very cumbersome and error prone. It also very inefficient and a time-consuming process for observing the continuous increase in population and number of people visiting the hospital.

Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper.

The intention of the system is to reduce over-time pay and increase the number of patients that can be treated accurately. Requirement statements in these documents are both functional and non-functional.

8.2 Aim

As, keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system.

The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and also provides robust & reliable storage and backup facilities.

CHAPTER-9

PROPOSED ALGORITHM

TO IMPLEMENT THE

AUTOMATED HOSPITAL

MANAGEMENT SYSTEM

9.1 Proposed Methodology Regarding Algorithm

9.1.1 SQL table

It is a collection of data which is organized in terms of rows and columns. In DBMS, the table is known as relation and row as a tuple. Table is a simple form of data storage. A table is also considered as a convenient representation of relations. some of the queries for creating employee table:

```
CREATE TABLE EMPLOYEE (
    EMP_ID INT(10),
    NAME VARCHAR(50),
    CITY VARCHAR(50),
    PHONE_NO INT(10),
    PRIMARY KEY (ID )
);
```

So, the empty employee table will be created and now data can be inserted by using query:

```
INSERT INTO EMPLOYEE VALUES(1,'KRISTEN','WASHINGTON',7289201223);
```

EMP_ID	NAME	CITY	PHONE_NO
1	Kristen	Washington	7289201223
2	Anna	Franklin	9378282882
3	Jackson	Bristol	9264783838
4	Kellan	California	7254728346
5	Ashley	Hawaii	9638482678

In the above table, "EMPLOYEE" is the table name, "EMP_ID", "NAME", "CITY", "PHONE_NO" are the column names. The combination of data of multiple columns forms a row, e.g., 1, "Kristen", "Washington" and 7289201223 are the data of one row.

9.1.2 Metadata

Metadata is simply defined as data about data. It means it is a description and context of the data. It helps to organize, find and understand data.

For example, every time we take a photo with today's cameras a bunch of metadata is gathered and saved with it:

1. Date and time
2. Filename,
3. Camera settings,
4. Geolocation.

9.1.2.1 Meta data in Relational database

Relational databases store and provide access not only data but also metadata in a structure called data dictionary or system catalog. It holds information about:

1. tables,
2. columns,
3. data types,
4. table relationship,
5. constraints etc.

9.1.3 Attribute

An attribute is a property or characteristic of an entity. An entity may contain any number of attributes. One of the attributes is considered as the primary key. In an Entity-Relationship model, attributes are represented in an elliptical shape.

Example: Student has attributes like name, age, roll number and many more. To uniquely identify the student, we use the primary key as roll number as it is not repeated.

9.1.4 Connections for Frontend to backend

The frontend is the part of the website users can see and interact with such as the graphical user interface (GUI) and the command line including the design, navigating menus, texts, images, videos, etc. It is also referred to as the 'client side' of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and navigation menu.

Backend, on the contrary, is the part of the website users cannot see and interact with. Backend is the server-side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine.

Frontend and backend communicate with each other - via Http requests. The frontend, for example, send entered data to the backend. The backend might then again validate that data (since frontend code can be tricked) and finally store it in some database.

The Front end and Backend used in our project are html,css, javascript and php and MySQL for Hospital Management System. PHP combined with MySQL are cross-platform (we can develop in Windows and serve on a Unix platform). With PHP, we can connect to and manipulate databases. MySQL is the most popular database system used with PHP.

9.1.5 Different Frontend functions and their Efficiency

We have used html,css and javascript as a frontend applications in our project.

9.1.5.1 Html functions

1. Block-level elements start on a new line in the document and take up their own space. Examples of these elements include headings and paragraph tags.
2. Inline elements do not start on a new line in the document and only take up necessary space. These elements usually format the contents of block-level elements. Examples of inline elements include hyperlinks and text format tags.

9.1.5.1.1 Efficiency

It is easy to use and understand, supported by all the browsers, simple to edit, lightweight, integrate easily with other languages and user friendly.

9.1.5.2 Css functions

1. The linear-gradient() function: It is an inbuilt function in CSS which is used to set the linear gradient as the background image. Syntax: background-image: linear-gradient(direction, color1, color2, ...).
2. Selector – A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
3. Property – A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.
4. Value – Values are assigned to properties. For example, color property can have value either red or #F1F1F1 etc.
5. Colors in CSS: In this CSS commands, colors can be specified in RGB formula. Each parameter defines the intensity of these colors and defines a new color. For example, to display black all color parameters should be set to RGB(0,0,0).

9.1.5.2.1 Efficiency

CSS has excellent support in today's browsers. It's relatively easy to learn and produces better and cleaner code than applying all those styles directly to your HTML.

9.1.5.3 JavaScript functions

The <script> tag is used to embed a client-side script (JavaScript). The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

9.1.5.3.1 Efficiency

Client-side JavaScript is very fast because it can be run immediately within the client-side browser. Unless outside resources are required, JavaScript is unhindered by network calls to a backend server. Being client-side reduces the demand on the website server.

9.2 Algorithm Steps

(Backend Database Design)

1. Consider a Database 'db'.
2. Consider a Automated Hospital Management System in terms of two domain,i.e,
 - i)Admin Section
 - ii)Patient Section.
3. Create and consider the Hospital's Patient Section as an independent relation within 'db'. 'Patient id' as the Primary key integrity constraint declaring for it.
4. Consider Admin Section in terms of all subdomain patient details, doctor details, addmission details, diagnosis details, billing details, Icu details, department details, building details and staff details which are controlled by it.
5. Create each of the subdomain as an independent relation within 'db' and also declare necessary Integrity constraints for each individual relation.

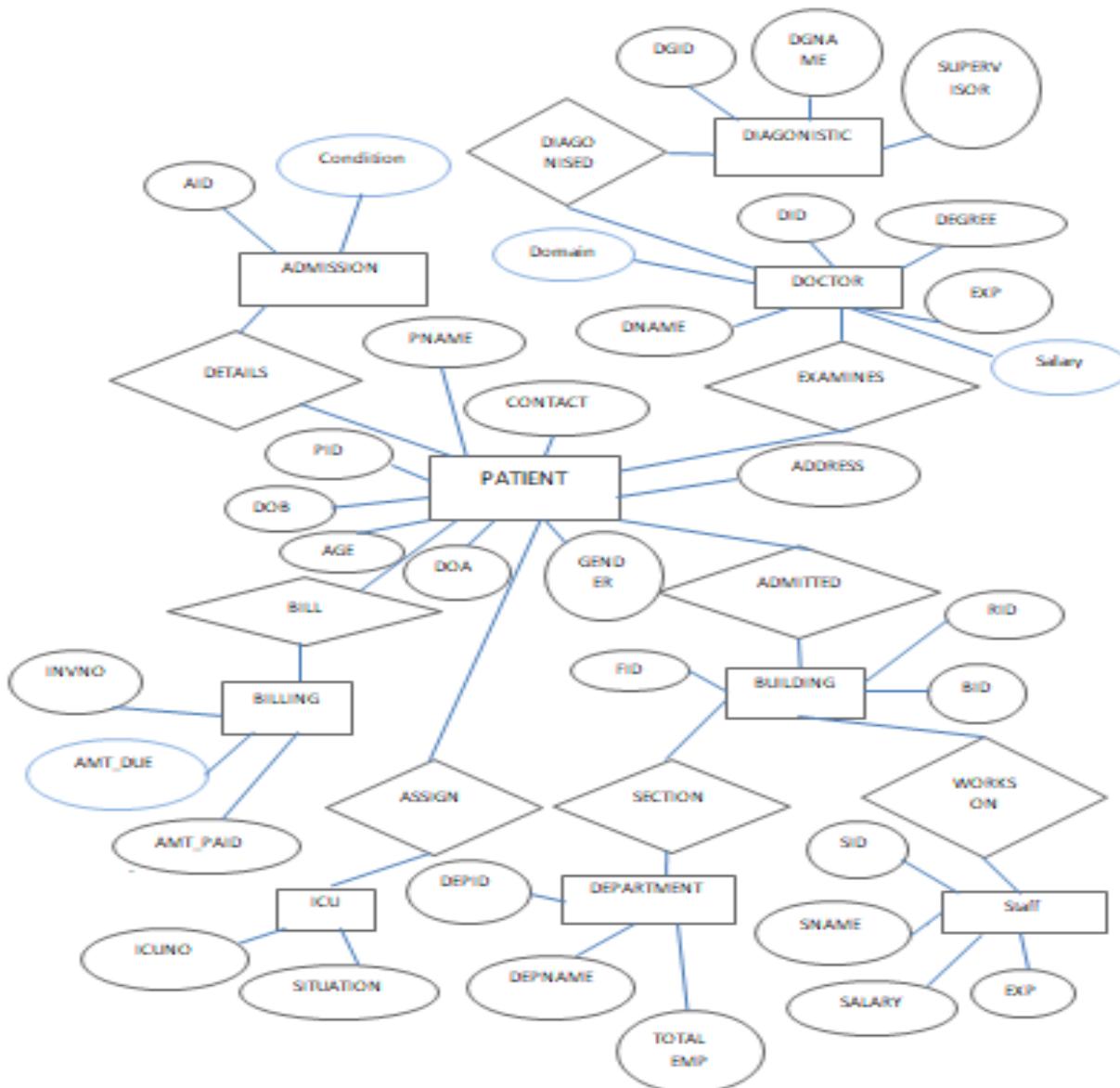
(Frontend Database Design)

6. Design a suitable frontend interfaces,each individual relation in php which not only accepts and store data from Patient or Admin,but also perform necessary 'Delete','View','Update' operations. Here relationship is being established between Frontend and Backend in such a way that any action has been taken from Frontend immediately which affects the Backend Databases.
7. Lastly,create necessary frontend interfaces which will initiate some Query operation considering one or more than one subdomains at a time for 'db' and display possible results.

CHAPTER-10

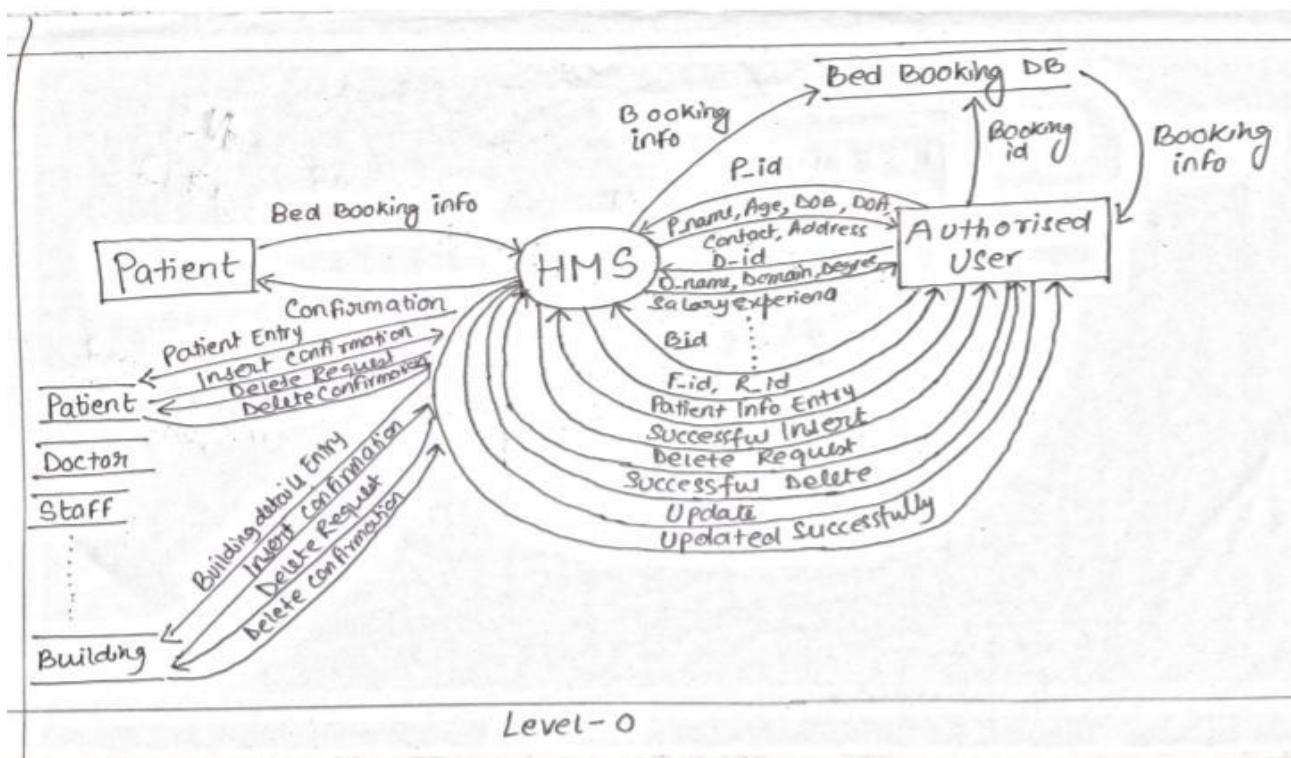
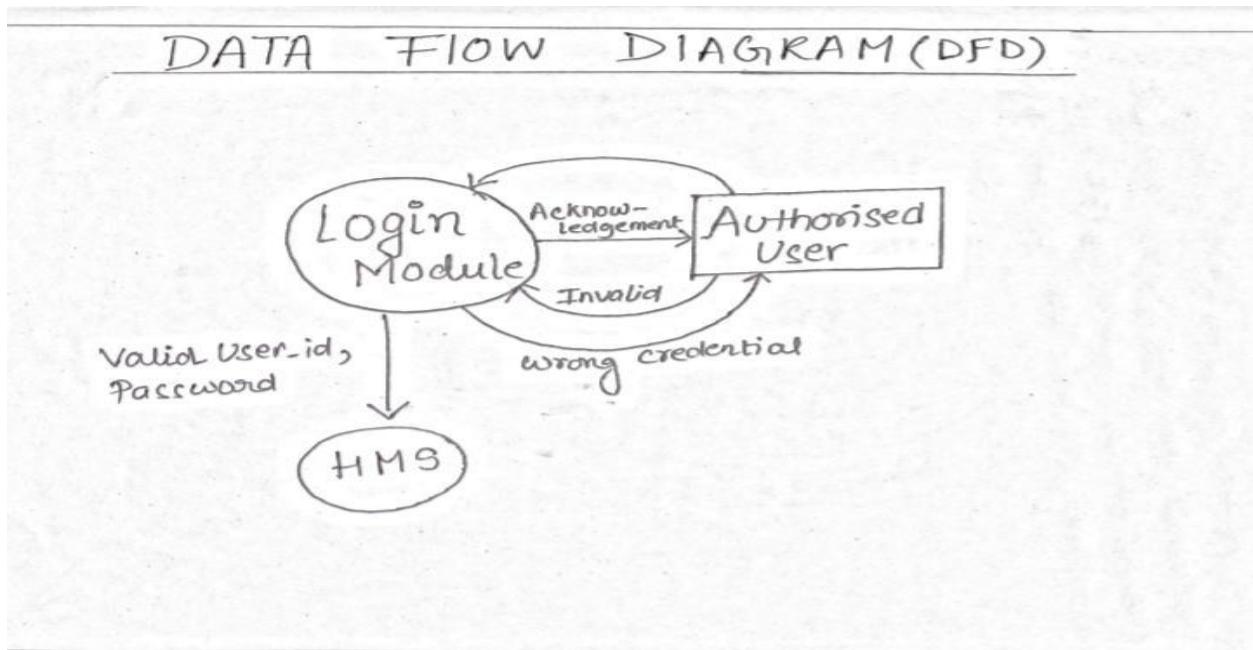
PROPOSED SOFTWARE DESIGN

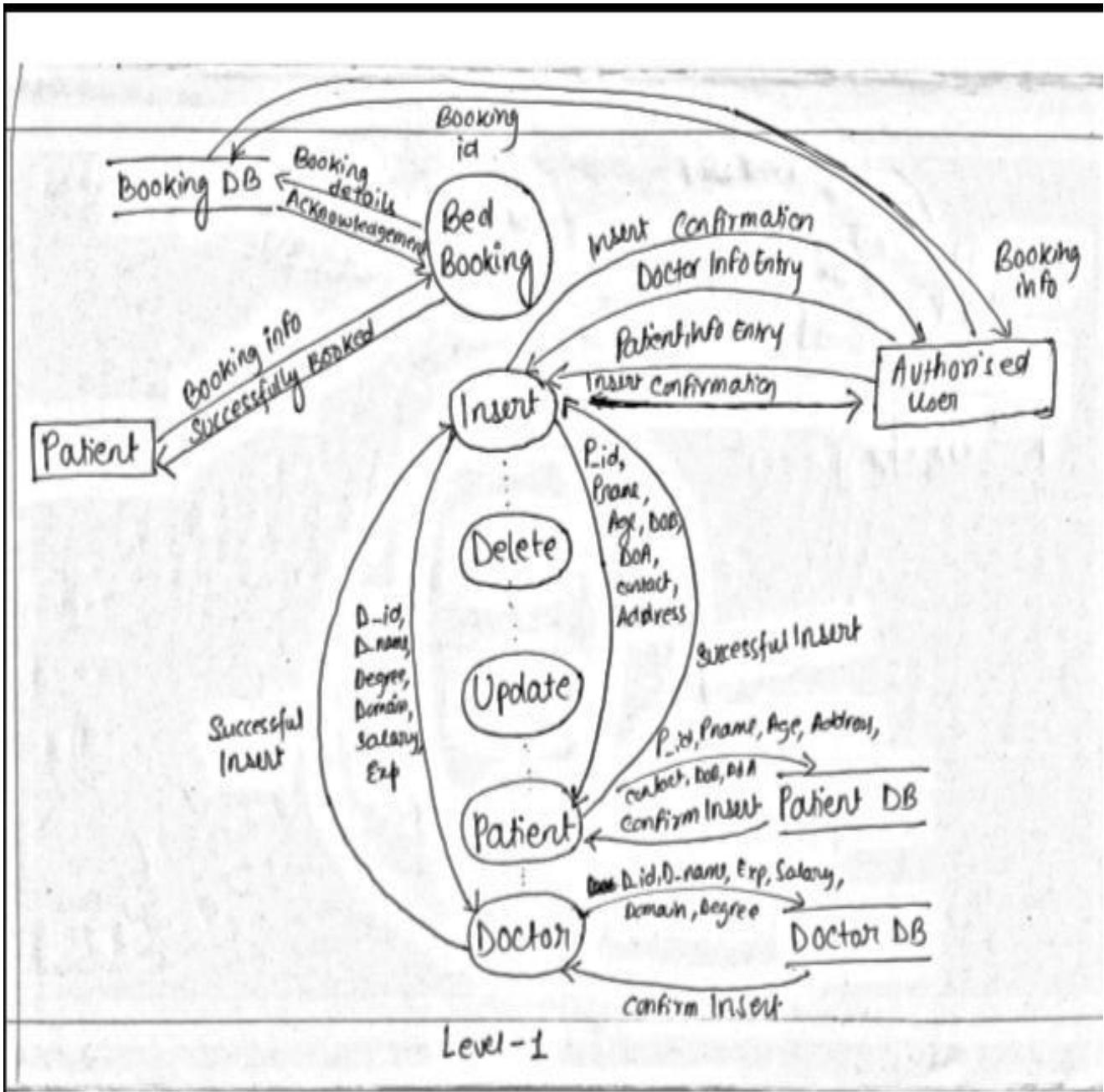
10.1 Entity Relationship Diagram



1

10.2 Data flow Diagram





CHAPTER-11

CASE STUDIES

Code For Admin Login Page:-

```
<html>
<head>
<title>hospital management system</title>
<link rel = "stylesheet" href = "stylehospital.css">
</head>
<body>
<header>
<div class="title">
<h1><u>CITY HOSPITAL</u></h1>
</div>
<div class="button">
<a href="admin.php" class="btn">ADMIN LOGIN</a>&ampnbsp&ampnbsp&ampnbsp
<a href="bedbooking.php" class="btn">CORONA BED BOOKING</a>
</div>
</header>
</body>
</html>
```

Css styling code for Admin Login page

```
* {
margin:0;
padding:0;
font-family:century gothic;
}

header{
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(img.jpg);
height:100vh;
background-size:cover;
background-position:center;
}

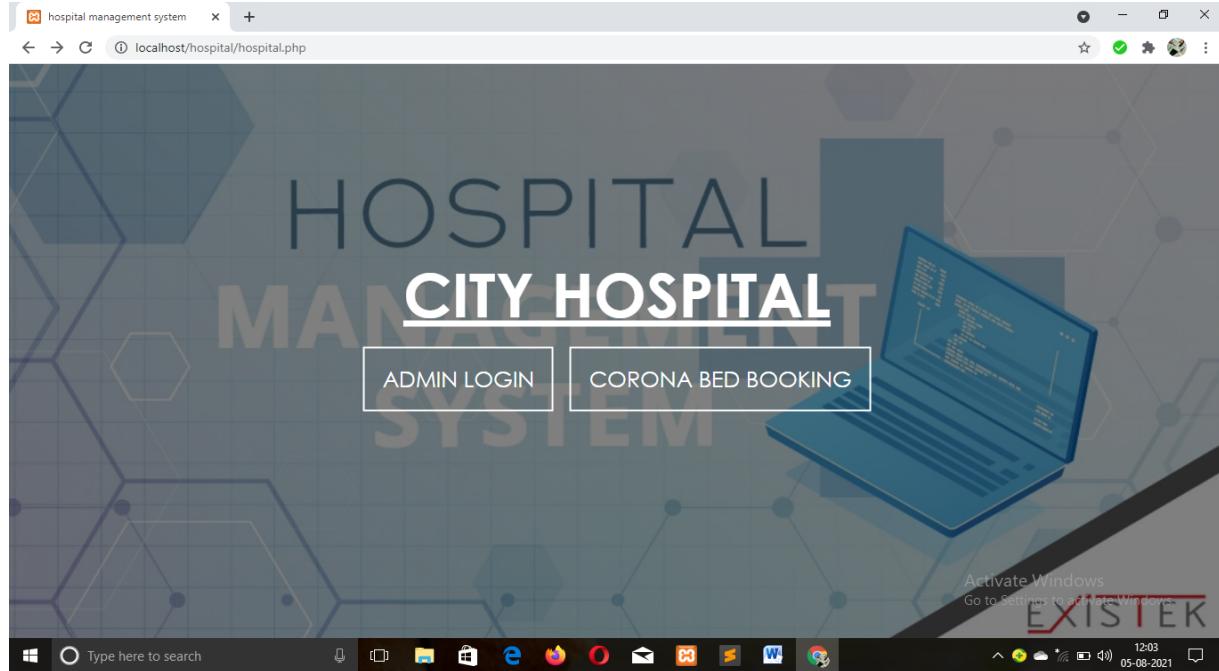
.title{
position: absolute;
top:40%;
left:50%;
transform: translate(-50%, -50%);
}

.title h1{
color:white;
```

```

font-size:70px;
}
.button{
position: absolute;
top:55%;
left:50%;
transform: translate(-50%, -50%);
}
.btn{
color: white;
padding: 20px 20px;
font-size:25px;
transition:0.6s ease;
border:2px solid white;
text-decoration:none;
}
.btn: hover{
background-color:white;
color:black;
}

```



Code for Login Page:-

```
<html>
<head>
    <title>Login Page</title>
    <link rel="stylesheet" href="styleadmin.css">
</head>
<body>
    <h1><u>CITY HOSPITAL</u></h1>
    <form action="login.php" method="post">
        <h2><center><i><u>LOGIN FOR ACCESSING THE ADMIN
PAGE</u></i></center></h2><br><br>
        <h3><center>Enter Your Credentials</center></h3><br>
        <?php
            if(isset($_GET['error']))
            {
                ?>
                <p class="error">
                <?php
                    echo $_GET['error'];
                ?>
                </p>
            <?php
            }
            ?>
            <center>
                <input type="text" name="username" placeholder="Enter your
Email"><br><br>
                <input type="password" name="password" placeholder="Enter your
Password"><br><br>
            <button type="submit" class="button">LOGIN</button>
            </center>
        </form>
    </body>
</html>
```

Css styling code for Login Page:-

```
* {
margin:0;
padding:0;
font-family:century gothic;
}
body{
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(image.jpg);
```

```
height:100vh;
background-size:cover;
background-position:center;
}

h1{
position: absolute;
top:40%;
left:50%;
color:black;
transform: translate(-50%, -50%);
font-size:60px;
}

h2,h3{
color:black;
font-size:20px;
}

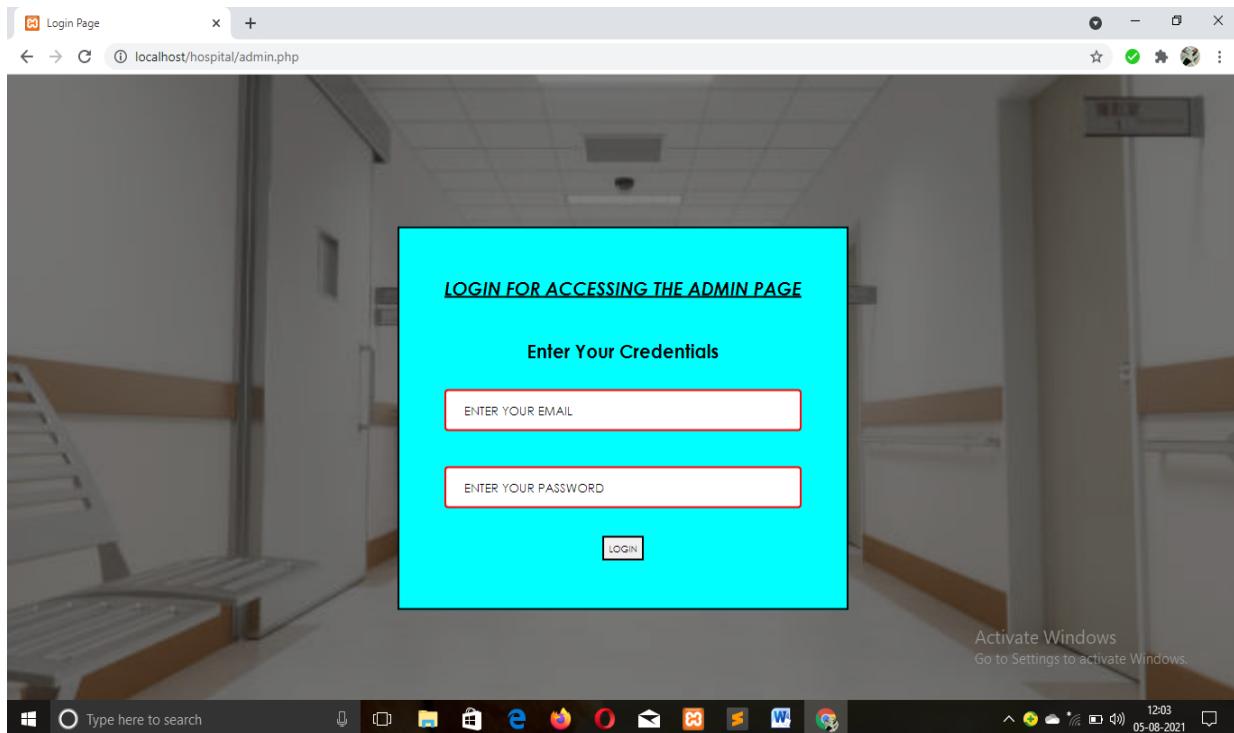
form{
top:55%;
left:50%;
position:absolute;
border:2px solid black;
background-color:aqua;
form-size:50px;
padding:50px 50px;
transform: translate(-50%, -50%);
color:black;
}

text{
padding:50px50px;
border-radius:4px;
box-sizing:border-box;
}

::placeholder {
color: black;
text-transform: uppercase;
}

button{
color: black;
padding: 5px 5px;
font-size:10px;
transition:0.6s ease;
border:2px solid ;
}
```

```
text-decoration:none;  
}  
  
button: hover{  
background-color:white;  
color:black;  
}  
  
input {  
width: 100%;  
padding: 12px 20px;  
margin: 8px 0;  
box-sizing: border-box;  
border: 2px solid red;  
border-radius: 4px;  
}
```



Code for Home Page:-

```
<html>
<head>
    <title>HOME PAGE</title>
    <link rel = "stylesheet" href = "stylehome.css">
</head>
<body>
<header>
    <?php
        session_start();
    ?>
<div class="title">
<h1><u>WELCOME TO CITY HOSPITAL</u></h1>
</div>
<div class="button">
    <a href="insert.php" class ="btn">INSERT DATA</a>&nbsp&nbsp
    <a href="view.php" class ="btn">DELETE/UPDATE DATA</a>&nbsp&nbsp
    <a href="display.php" class ="btn">VIEW DATA</a>
</div>
</header>
</body>
</html>
```

Css styling code for Home Page:-

```
* {
margin:0;
padding:0;
font-family:century gothic;
}

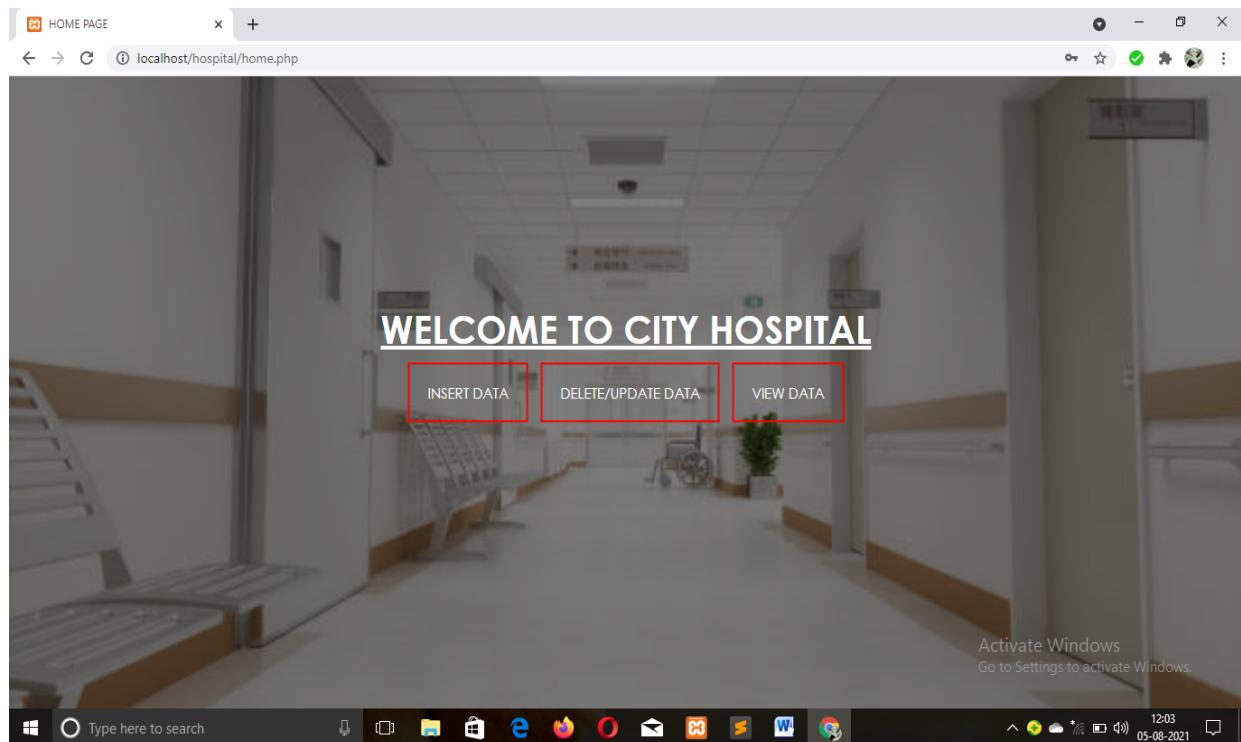
header{
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(image.jpg);
height:100vh;
background-size:cover;
background-position:center;
}

.title{
position: absolute;
top:40%;
left:50%;
transform: translate(-50%, -50%);
}

.title h1{
color:white;
```

```
font-size:40px;
}
.button{
position: absolute;
top:50%;
left:50%;
transform: translate(-50%, -50%);
}
.btn{
color: white;
padding: 20px 20px;
font-size:15px;
transition:0.6s ease;
border:2px solid red;
text-decoration:none;
}
.btn: hover{
background-color:white;
color:black;
}

}
```



Code for Insert Page:-

```
<html>
<head>
<link rel = "stylesheet" href = "styleinsert.css">
</head>
<body>
<header>
<div class="title">
<h1><u>CITY HOSPITAL</u></h1>
</div>
<div class="button">
    <a href="insert_patient.php" class ="btn">INSERT INTO PATIENT
    TABLE</a>&nbsp;&nbsp;&nbsp;
    <a href="insert_doctor.php" class ="btn">INSERT INTO DOCTOR TABLE</a>
</div>
</header>
</body>
</html>
```

Css styling for Insert Page:-

```
* {
margin:0;
padding:0;
font-family:century gothic;
}

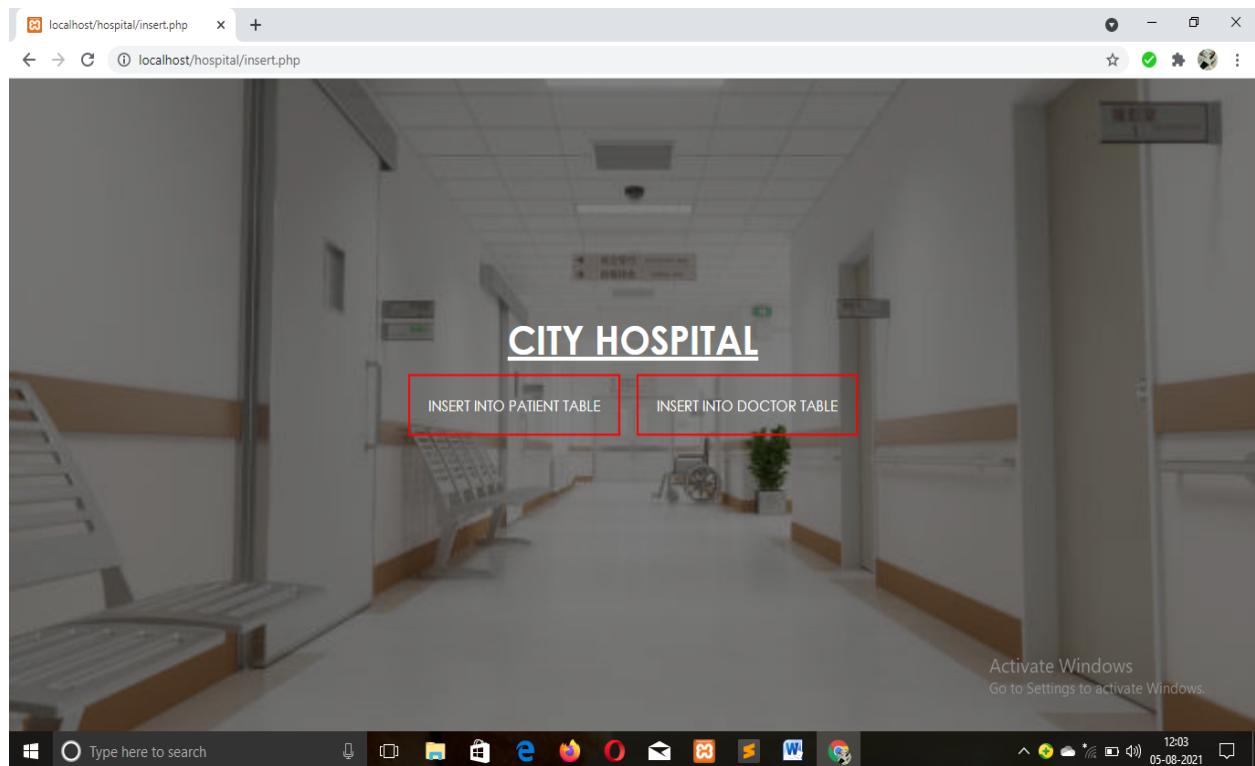
.header{
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(image.jpg);
height:100vh;
background-size:cover;
background-position:center;
}

.title{
position: absolute;
top:40%;
left:50%;
transform: translate(-50%, -50%);
}

.title h1{
color:white;
font-size:40px;
}

.button{
```

```
position: absolute;  
top:50%;  
left:50%;  
transform: translate(-50%, -50%);  
}  
.btn{  
color: white;  
padding: 20px 20px;  
font-size:15px;  
transition:0.6s ease;  
border:2px solid red;  
text-decoration:none;  
}  
.btn: hover{  
background-color:white;  
color:black;  
}
```



Code for Insert Doctor Details Page:-

```
<?php
include("db_conn.php");
error_reporting(0);
?>
<html>
<head>
    <title> FORM FOR INSERT </title>
    <link rel = "stylesheet" href = "styleinsert_doctor.css">
</head>
<body>
    <form method="GET">
        <center><h2>DOCTOR DETAILS</h2></center>
        D_ID <input type="text" name="did" placeholder="ENTER DOCTOR'S ID"
value="" /><br>
        D_NAME <input type="text" name="dname" placeholder="ENTER DOCTOR'S NAME"
value="" /><br>
        DEGREE <input type="text" name="degree" placeholder="ENTER DOCTOR'S
DEGREE" value="" /><br>
        DOMAIN <input type="text" name="domain" placeholder="ENTER DOCTOR'S
DOMAIN" value="" /><br>
        SALARY <input type="text" name="salary" placeholder="ENTER DOCTOR'S
SALARY" value="" /><br>
        EXPERIENCE <input type="text" name="exp" placeholder="ENTER DOCTOR'S
EXPERIENCE" value="" /><br>
        <center><button type="submit" class="button">INSERT</button></center>
    </form>
    <?php
        if($_GET['submit'])
        {
            $a=$_GET['did'];
            $b=$_GET['dname'];
            $c=$_GET['degree'];
            $d=$_GET['domain'];
            $e=$_GET['salary'];
            $f=$_GET['exp'];
            if($a!="" && $b!="" && $c!="" && $d['gender']!="" && $e!="" &&
$f!="")
            {
                $query="INSERT INTO Doctor VALUES
('$a','$b','$c','$d','$e','$f')";
                $data=mysqli_query($conn,$query);
                if($data)

```

```

        {
            echo "DATA INSERTED INTO THE DATABASE";
        }
    }
else
{
    echo "ALL FIELDS ARE REQUIRED";
}
}

?>

```

</body>

</html>

Css styling for Insert Doctor Details Page:-

```

*{
margin:0;
padding:0;
font-family:century gothic;
}

body{
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(booking.jpg);
height:100vh;
background-size:cover;
background-position:center;
}

h1{
position: absolute;
top:30%;
left:50%;
color:black;
transform: translate(-50%, -50%);
font-size:60px;
}

h2{
color:black;
font-size:20px;
}

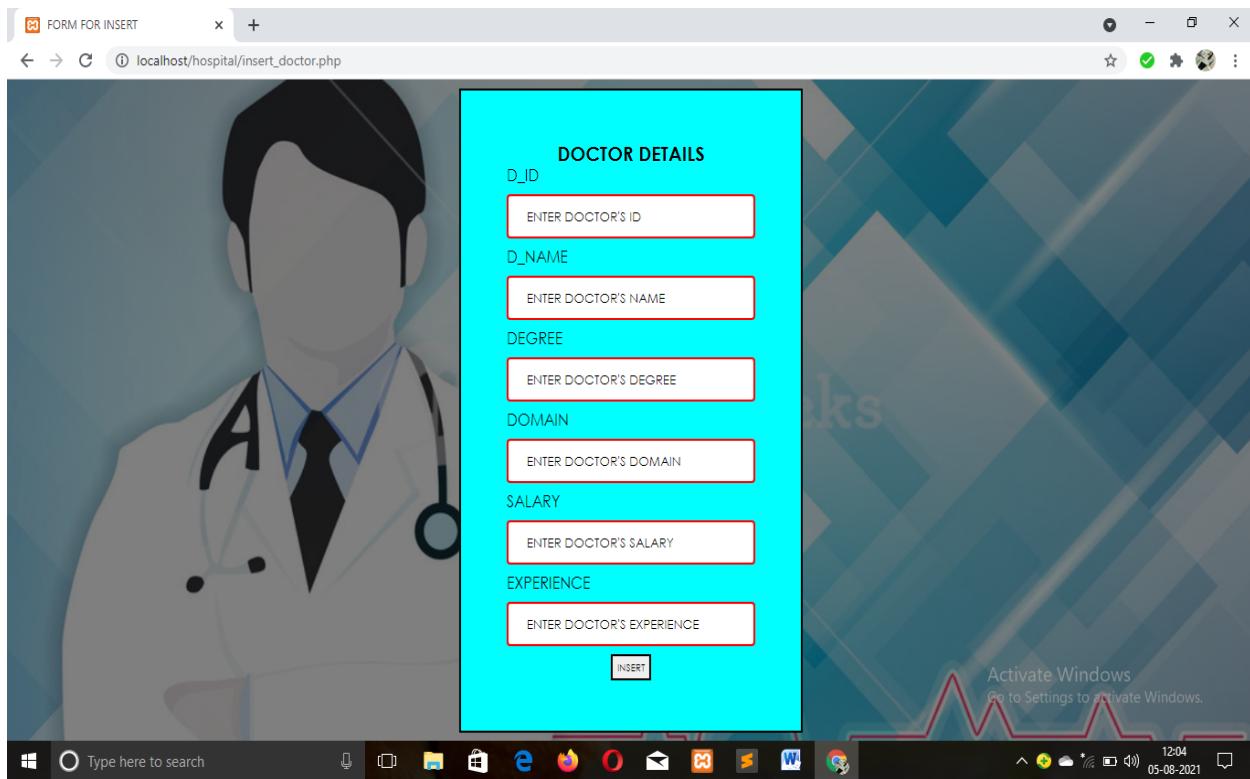
form{
top:50%;
left:50%;
position:absolute;
border:2px solid black;
background-color:aqua;
}
```

```
form-size:50px;
padding:50px 50px;
transform: translate(-50%, -50%);
color:black;
}

text{
padding:50px50px;
border-radius:4px;
box-sizing:border-box;
}

::placeholder {
    color: black;
    text-transform: uppercase;
}

button{
    color: black;
    padding: 5px 5px;
    font-size:10px;
    transition:0.6s ease;
    border:2px solid ;
    text-decoration:none;
}
button: hover{
    background-color:white;
    color:black;
}
input {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;
    border: 2px solid red;
    border-radius: 4px;
}
```



Code for Update/Delete Page:-

```
<style>
    td
    {
        padding: 10px;
    }
    body{
        background-image: linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(first.jpg);
        height:100vh;
        background-size:cover;
        background-position:center;
    }
</style>
<?php
include("db_conn.php");
error_reporting(0);
$query="SELECT * FROM Patient";
$data=mysqli_query($conn,$query);
$total=mysqli_num_rows($data);
if($total!=0)
```

```

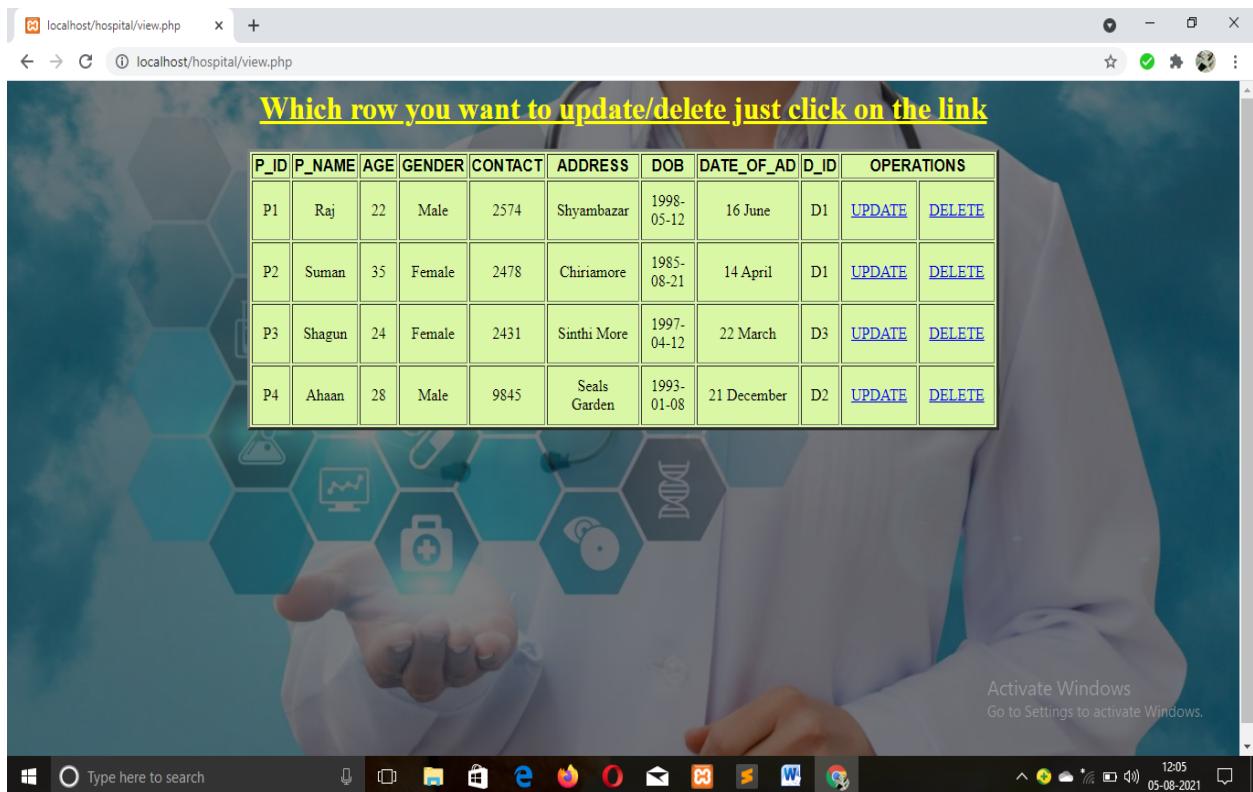
{
    ?>
    <h1 style="color: yellow"><center><u><b>Which row you want to
update/delete just click on the link</b></u></center></h1>
    <center>
        <table border="3" cellspacing="2" cellpadding="2" style="width: 50%;
text-align: center; background-color: #DAF7A6;">
            <tr>
                <th> <font face ="Arial">P_ID</font></th>
                <th> <font face="Arial">P_NAME</font></th>
                <th> <font face="Arial">AGE</font></th>
                <th> <font face="Arial">GENDER</font></th>
                <th> <font face="Arial">CONTACT</font></th>
                <th> <font face="Arial">ADDRESS</font></th>
                <th> <font face="Arial">DOB</font></th>
                <th> <font face="Arial">DATE_OF_AD </font></th>
                <th> <font face="Arial">D_ID</font></th>
                <th colspan="2" > <font face="Arial">OPERATIONS</font></th>
            </tr>
        <?php
            while($result=mysqli_fetch_assoc($data))
            {
                echo "<tr>
                    <td>".$result['Pid']."'</td>
                    <td>".$result['Pname']."'</td>
                    <td>".$result['Age']."'</td>
                    <td>".$result['Gender']."'</td>
                    <td>".$result['Contact']."'</td>
                    <td>".$result['Address']."'</td>
                    <td>".$result['DOB']."'</td>
                    <td>".$result['Date_of_ad']."'</td>
                    <td>".$result['Did']."'</td>
                    <td><a href ='update_patient.php?a=$result[Pid] &
b=$result[Pname] & c=$result[Age] & d=$result[Gender] & e=$result[Contact] &
f=$result[Address] & g=$result[DOB] & h=$result[Date_of_ad] &
i=$result[Did]'>UPDATE</a></td>
                    <td><a href='delete.php?a=$result[Pid]'>DELETE</a></td>
                </tr>";
            }
        else
        {
            echo "NO RECORDS ARE FOUND";
        }
    }
}

```

```

        }
    ?>
</table></center>
<script>
    function checkdelete()
    {
        return confirm("Are you sure you want to delete this data??");
    }
</script>

```



Code for Update Patient :-

```

<style>
    *{
        margin:0;
        padding:0;
        font-family:century gothic;
    }
    body{
        background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(image.jpg);
        height:100vh;
    }

```

```

background-size:cover;
background-position:center;
}

</style>
<?php
include("db_conn.php");
error_reporting(0);
$_GET['a'];
$_GET['b'];
$_GET['c'];
$_GET['d'];
$_GET['e'];
$_GET['f'];
$_GET['g'];
$_GET['h'];
$_GET['i'];
?>
<html>
<head>
</head>
<body>
<form action="" method="GET">
P_ID<input type="text" name="pid" value="<?php echo $_GET['a'];?>"/><br><br>
P_NAME<input type="text" name="pname" value="<?php echo $_GET['b'];?>"/><br><br>
AGE<input type="text" name="age" value="<?php echo $_GET['c'];?>"/><br><br>
GENDER<input type="text" name="gender" value="<?php echo $_GET['d'];?>"/><br ><br>
CONTACT<input type="text" name="con" value="<?php echo $_GET['e'];?>"/><br><br>
ADDRESS<input type="text" name="add" value="<?php echo $_GET['f'];?>"/><br><br>
DOB<input type="text" name="dob" value="<?php echo $_GET['g'];?>"/><br><br>
DATE_OF_AD<input type="text" name="ad" value="<?php echo $_GET['h'];?>"/><br><br>
D_ID<input type="text" name="did" value="<?php echo $_GET['i'];?>"/><br><br>
<input type="submit" name="submit" value="update"/>
</form>
<?php
if($_GET['submit'])
{
    $A=$_GET['pid'];
    $B=$_GET['pname'];
    $C=$_GET['age'];
    $D=$_GET['gender'];
    $E=$_GET['con'];
    $F=$_GET['add'];
}

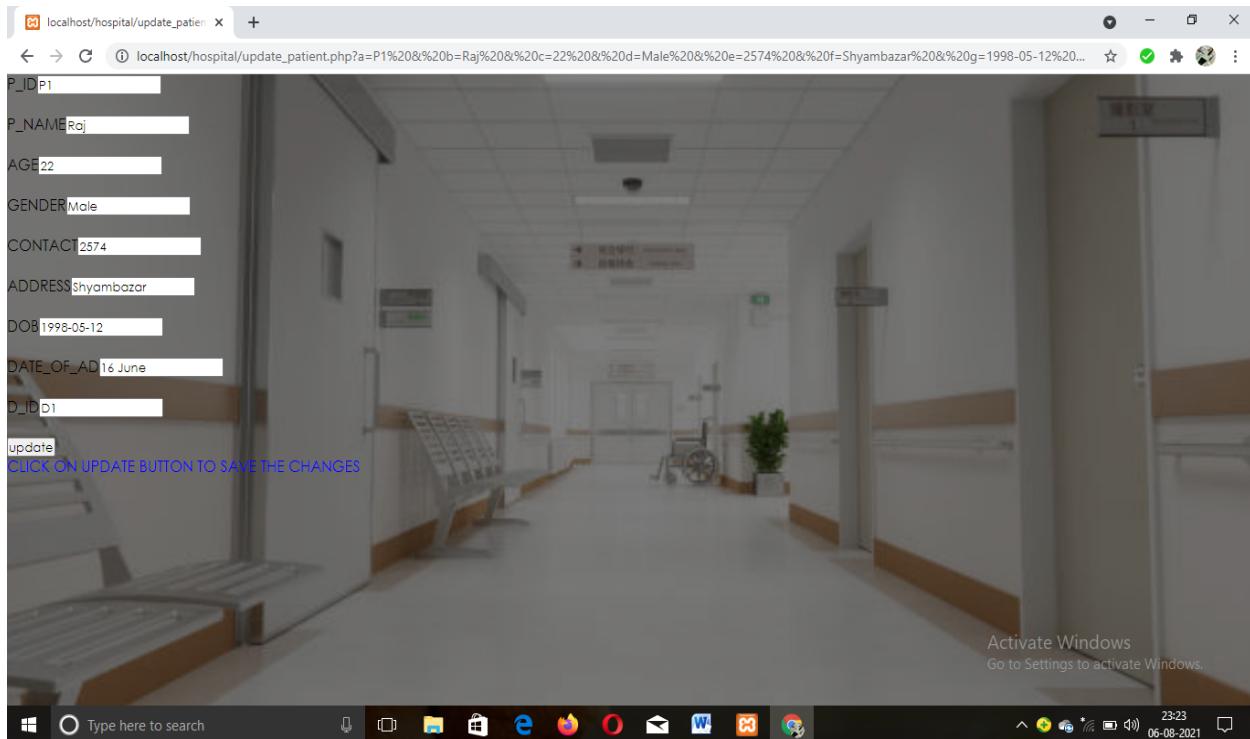
```

```

$G=$_GET['dob'];
$H=$_GET['ad'];
$I=$_GET['did'];
$query= "UPDATE patient SET Pname='$B' , Age
 ='$C' , Gender='$D' , Contact='$E' , Address='$F' , DOB='$G' , Date_of_ad='$H' ,
Did='$I' WHERE Pid='$A' ";
$data=mysqli_query($conn,$query);
if($data)
{
    echo "<font color='green'>Record Updated
Successfully.<a href='view.php'>Check Updated list here</a>";
}
else
{
    echo "<font color='red'>RECORDS NOT
UPDATED. <a href='view.php'>Check Updated list here</a>";
}
else
{
    echo "<font color='blue'>CLICK ON UPDATE BUTTON TO
SAVE THE CHANGES";
}

?>
</body>
</html>

```



Code for Display Page:-

```

<html>
<head>
<link rel = "stylesheet" href = "styledisplay.css">
</head>
<body>
<header>
    <div class="title">
        <h1><u>CITY HOSPITAL</u></h1>
    </div>
    <div class="button">
        &nbsp;&nbsp;&nbsp;&nbsp;<a href="display_patient.php" class="btn">PATIENT</a><br><br>&nbsp;&nbsp;&nbsp;
        <a href="display_doctor.php" class="btn">DOCTOR</a><br><br>&nbsp;&nbsp;&nbsp;
        <a href="display_dept.php" class="btn">DEPARTMENT</a><br><br>&nbsp;&nbsp;&nbsp;
        <a href="display_icu.php" class="btn">ICU </a><br><br>&nbsp;&nbsp;&nbsp;
        <a href="display_diagonistic.php" class="btn">DIAGONISIC</a><br><br>&nbsp;&nbsp;&nbsp;
        <a href="join-doc-pat.php" class="btn">PATIENT, ADDMISSION AND DOCTOR</a><br><br>&nbsp;&nbsp;&nbsp;
        <a href="join-pat-bill.php" class="btn">PATIENT AND BILL</a><br><br>&nbsp;&nbsp;&nbsp;
        <a href="join-staff-build.php" class="btn">STAFF AND BUILDING</a>
    </div>
</header>

```

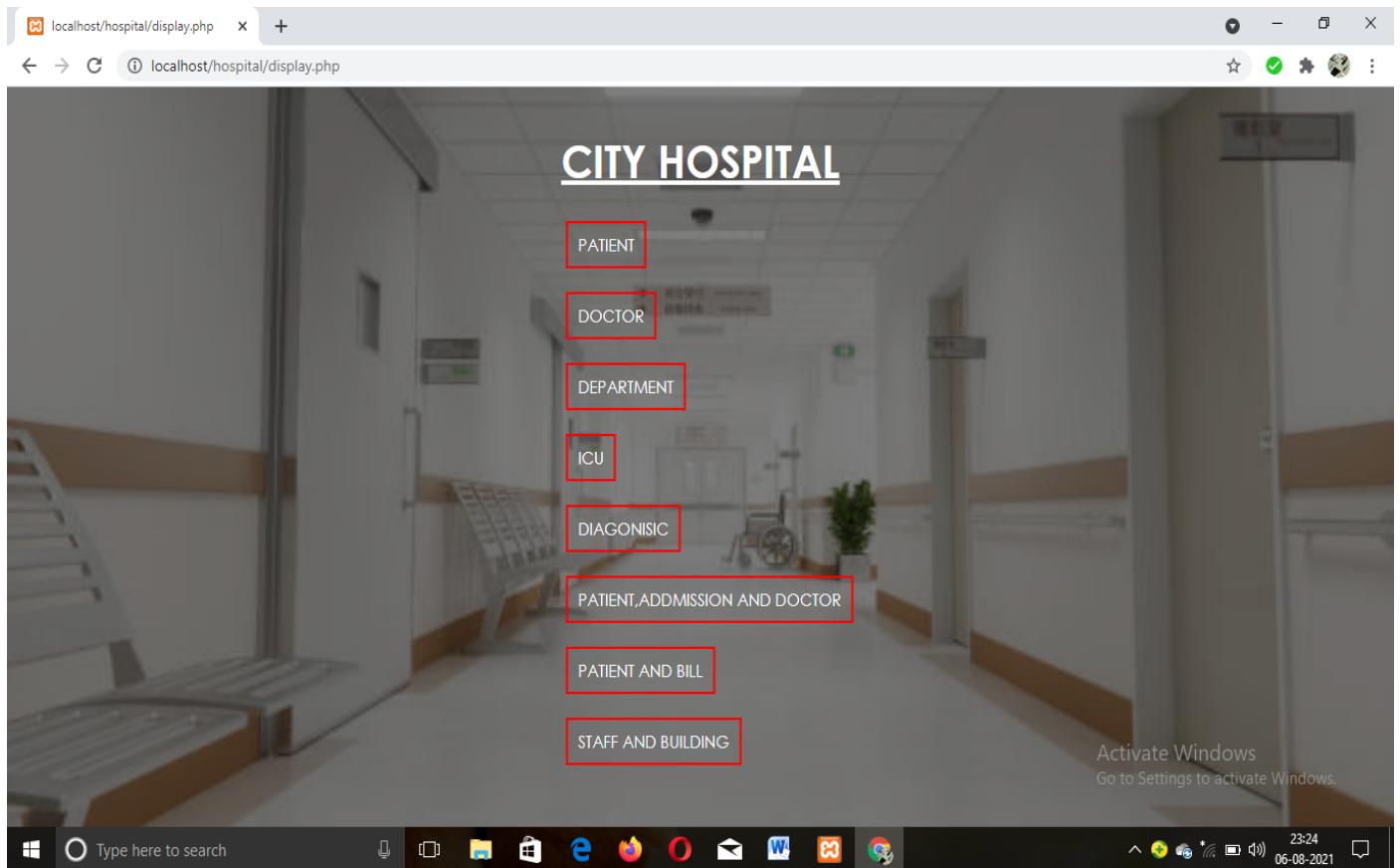
```

        </div>
    </header>
</body>
</html>

Css styling code for Display Page:-
* {
margin:0;
padding:0;
font-family:century gothic;
}
header{
background-image:linear-gradient(rgb(0,0,0,0.5),rgb(0,0,0,0.5)),url(image.jpg);
height:100vh;
background-size:cover;
background-position:center;
}
.title{
position: absolute;
top:10%;
left:50%;
transform: translate(-50%, -50%);
}
.title h1{
color:white;
font-size:40px;
}
.button{
position: absolute;
top:55%;
left:50%;
transform: translate(-50%, -50%);
}
.btn{
color: white;
padding: 10px 10px;
font-size:15px;
transition:0.6s ease;
border:2px solid red;
text-decoration:none;
}
.btn: hover{
background-color:white;
}

```

```
color:black;  
}
```



Code for Display Patient Page:-

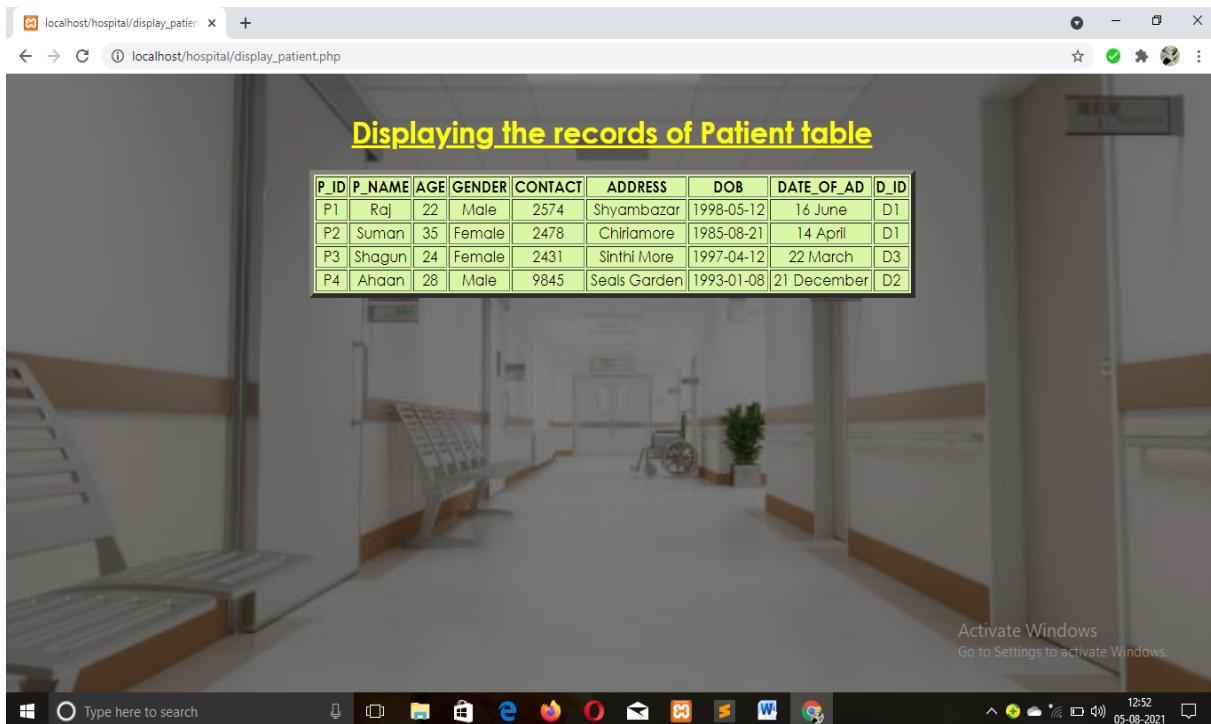
```
<style>  
*{  
    margin:0;  
    padding:0;  
    font-family:century gothic;  
}  
body{  
  
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(image.jpg);  
height:100vh;  
background-size:cover;  
background-position:center;  
}  
</style>  
<?php  
include("db_conn.php");
```

```

error_reporting(0);
$query="SELECT * FROM Patient";
$data=mysqli_query($conn,$query);
$total=mysqli_num_rows($data);
if($total!=0)
{
    ?
    <br><br>
    <center><h1 style="color: yellow"><b><u>Displaying the records of Patient
table </u></b></h2></center><br>
    <center>
        <table border="5" cellspacing="2" cellpadding="2" style="width: 50%;
text-align: center; background-color: #DAF7A6;">
            <tr>
                <th> <font face ="Arial">P_ID</font></th>
                <th> <font face="Arial">P_NAME</font></th>
                <th> <font face="Arial">AGE</font></th>
                <th> <font face="Arial">GENDER</font></th>
                <th> <font face="Arial">CONTACT</font></th>
                <th> <font face="Arial">ADDRESS</font></th>
                <th> <font face="Arial">DOB</font></th>
                <th> <font face="Arial">DATE_OF_AD </font></th>
                <th> <font face="Arial">D_ID</font></th>
            </tr>
        </table>
    <?php
        while($result=mysqli_fetch_assoc($data))
        {
            echo "<tr>
                    <td>".$result['Pid']."'</td>
                    <td>".$result['Pname']."'</td>
                    <td>".$result['Age']."'</td>
                    <td>".$result['Gender']."'</td>
                    <td>".$result['Contact']."'</td>
                    <td>".$result['Address']."'</td>
                    <td>".$result['DOB']."'</td>
                    <td>".$result['Date_of_ad']."'</td>
                    <td>".$result['Did']."'</td>
                </tr>";
        }
    else
    {
        echo "NO RECORDS ARE FOUND";
    }
}

```

```
?>  
</table></center>
```



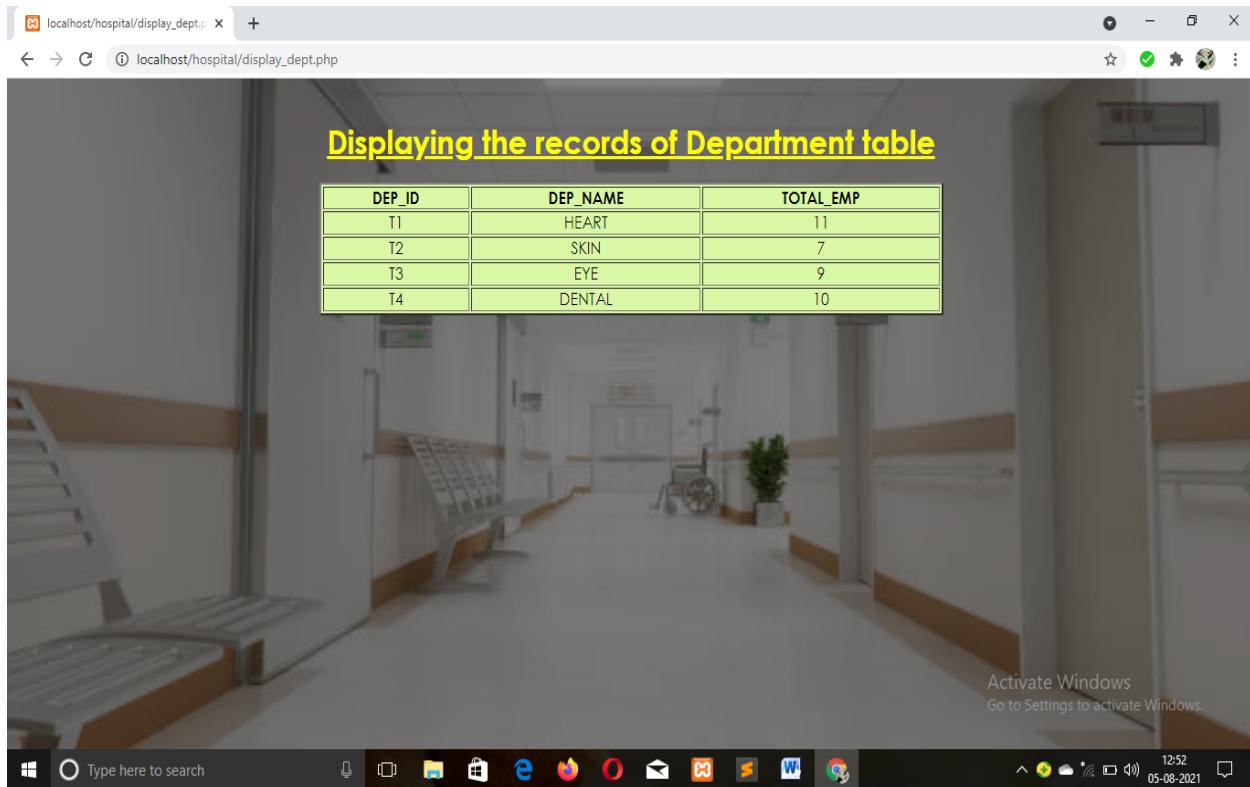
Code for Display Department Page:-

```
<style>
    * {
        margin:0;
        padding:0;
        font-family:century gothic;
    }
    body{
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(image.jpg);
        height:100vh;
        background-size:cover;
        background-position:center;
    }
</style>
<?php
    include("db_conn.php");
```

```

error_reporting(0);
$query="SELECT * FROM DEPARTMENT";
$data=mysqli_query($conn,$query);
$total=mysqli_num_rows($data);
if($total!=0)
{
    ?
    <br><br>
    <center> <h1 style="color: yellow"> <b> <u> Displaying the records
of Department table </u> </b> </h2> </center><br>
    <center>
        <table border="2" cellspacing="2" cellpadding="2" style="width:
50%; text-align: center; background-color: #DAF7A6;">
            <tr>
                <th> <font face ="Arial">DEP_ID</font></th>
                <th> <font face="Arial">DEP_NAME</font></th>
                <th> <font face="Arial">TOTAL_EMP</font></th>
            </tr>
            <?php
while($result=mysqli_fetch_assoc($data))
{
    echo "<tr>
                    <td>".$result['DepId']."'</td>
                    <td>".$result['DepName']."'</td>
                    <td>".$result['TotalEmp']."'</td>
                </tr>";
}
else
{
    echo "NO RECORDS ARE FOUND";
}
?>
</table></center>

```



Code for Display ICU Page:-

```

<style>
  * {
    margin:0;
    padding:0;
    font-family:century gothic;
  }
  body{
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(first.jpg);
height:100vh;
background-size:cover;
background-position:center;
}
</style>

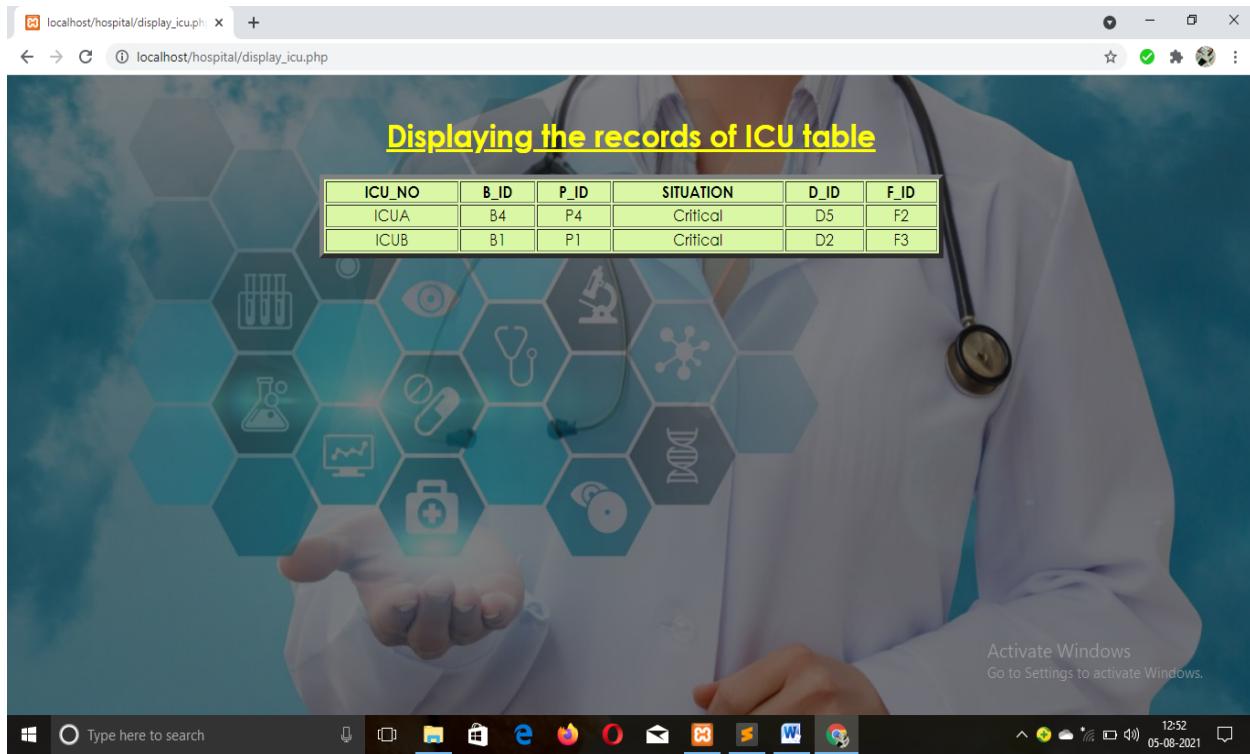
<?php
include("db_conn.php");
error_reporting(0);
echo "<br>";
echo "<br>";

```

```

$query="SELECT * FROM ICU";
$data=mysqli_query($conn,$query);
$total=mysqli_num_rows($data);
if($total!=0)
{
    ?
    <h1 style="color: yellow"><center><b><u>Displaying the records of ICU
table</u></b></center></h1> <br>
    <center>
        <table border="5" cellspacing="2" cellpadding="2" style="width: 50%;
text-align: center; background-color: #DAF7A6;">
            <tr>
                <th> <font face ="Arial">ICU_NO</font></th>
                <th> <font face="Arial">B_ID</font></th>
                <th> <font face="Arial">P_ID</font></th>
                <th> <font face="Arial">SITUATION</font></th>
                <th> <font face="Arial">D_ID</font></th>
                <th> <font face="Arial">F_ID</font></th>
            </tr>
            <?php
                while($result=mysqli_fetch_assoc($data))
                {
                    echo "<tr>
                        <td>".$result['icu_no']."'</td>
                        <td>".$result['Bid']."'</td>
                        <td>".$result['Pid']."'</td>
                        <td>".$result['Situation']."'</td>
                        <td>".$result['Did']."'</td>
                        <td>".$result['Fid']."'</td>
                    </tr>";
                }
            }
        else
        {
            echo "NO RECORDS ARE FOUND";
        }
    ?>
</table></center>

```



Code for Display the Join of Patient, Admission and Doctor Page:-

```

<style>
* {
    margin:0;
    padding:0;
    font-family:century gothic;
}
body{
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(image.jpg);
height:100vh;
background-size:cover;
background-position:center;
}
</style>

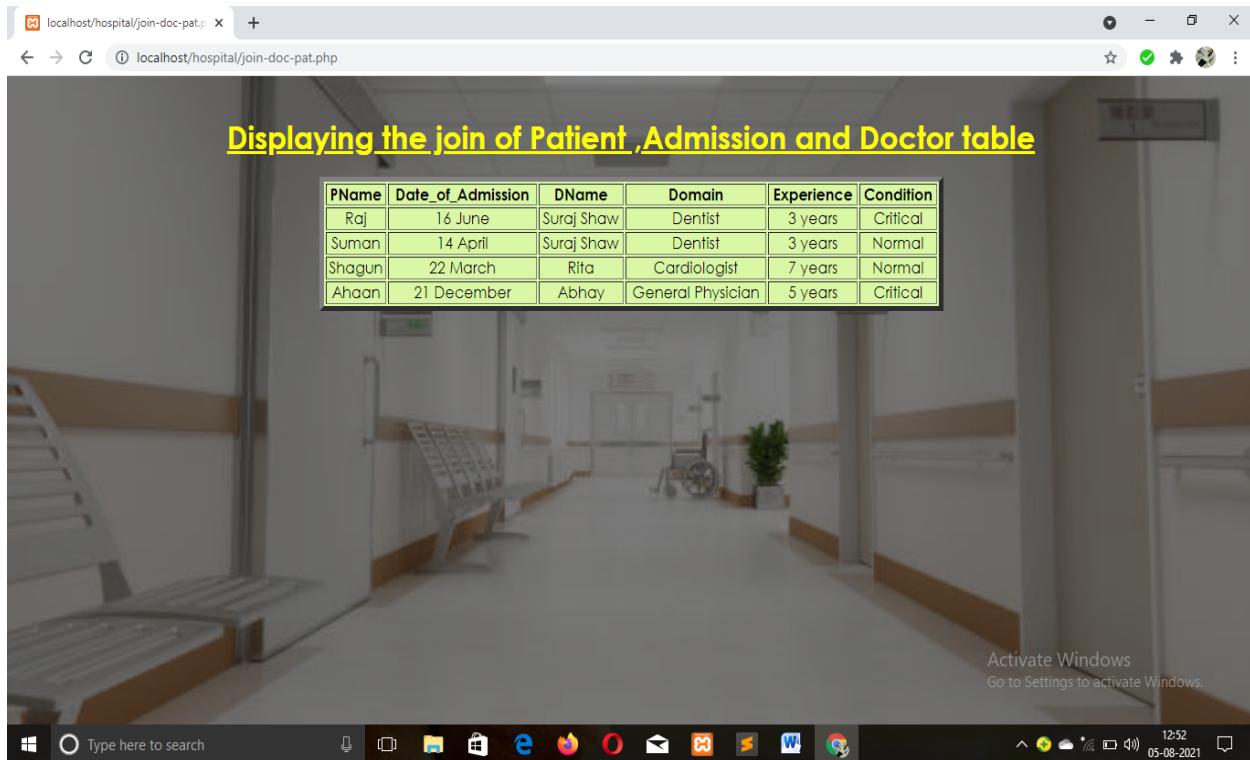
<?php
include("db_conn.php");
error_reporting(0);
$query="SELECT P.Pname,P.Date_of_ad,D.Dname,D.Domain,D.Exp,A.Con FROM PATIENT P,DOCTOR D,ADMISSION A WHERE P.Did=D.Did AND P.Pid=A.Pid";
$data=mysqli_query($conn,$query);
$total=mysqli_num_rows($data);
if($total!=0)

```

```

{
    ?>
    <br><br>
    <center><h1 style="color: yellow"><u><b>Displaying the join of Patient
,Admission and Doctor table </u></b></h1></center>
    <br>
    <center>
        <table border="5" cellspacing="2" cellpadding="2" style="width: 50%;
text-align: center; background-color: #DAF7A6;">
            <tr>
                <th> <font face="Arial">PName</font></th>
                <th> <font face="Arial">Date_of_Admission </font></th>
                <th> <font face="Arial">DName</font></th>
                <th> <font face="Arial">Domain</font></th>
                <th> <font face="Arial">Experience</font></th>
                <th> <font face="Arial">Condition</font></th>
            </tr>
            <?php
                while($result=mysqli_fetch_assoc($data))
                {
                    echo "<tr>
                        <td>".$result['Pname']."'</td>
                        <td>".$result['Date_of_ad']."'</td>
                        <td>".$result['Dname']."'</td>
                        <td>".$result['Domain']."'</td>
                        <td>".$result['Exp']."'</td>
                        <td>".$result['Con']."'</td>
                    </tr>";
                }
            }
        else
        {
            echo "NO RECORDS ARE FOUND";
        }
    ?>
</table></center>

```



Code for Display the Join of Patient and Bill Table:-

```

<style>
  * {
    margin:0;
    padding:0;
    font-family:century gothic;
  }
  body{
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(first.jpg);
  height:100vh;
  background-size:cover;
  background-position:center;
  }
</style>

<?php
include("db_conn.php");
error_reporting(0);
$query="SELECT P.Pid,P.Pname,B.InvNo,B.AmtPaid,B.AmtDue FROM BILL B,PATIENT P WHERE
P.Pid=B.Pid ";

```

```

$data=mysqli_query($conn,$query);
$total=mysqli_num_rows($data);
if($total!=0)
{
    ?
    <br><br>
    <center><h1 style="color: yellow;"><u><b>Displaying the join of Patient
and Bill table </b></u></h1></center>
    <br>
    <CENTER>
        <table border="5" cellspacing="2" cellpadding="2" style="width: 50%;
text-align: center; background-color: #DAF7A6;">
            <tr>
                <th> <font face="Arial"> P_ID </font></th>
                <th> <font face="Arial"> P_NAME </font></th>
                <th> <font face="Arial"> INV_NO </font></th>
                <th> <font face="Arial"> AMT_PAID </font></th>
                <th> <font face="Arial"> AMT_DUE </font></th>
            </tr>
            <?php
                while($result=mysqli_fetch_assoc($data))
                {
                    echo "<tr>
                        <td>".$result['Pid']."'</td>
                        <td>".$result['Pname']."'</td>
                        <td>".$result['InvNo']."'</td>
                        <td>".$result['AmtPaid']."'</td>
                        <td>".$result['AmtDue']."'</td>
                    </tr>";
                }
            }
        else
        {
            echo "NO RECORDS ARE FOUND";
        }
    ?>
</table>
</CENTER>
</table>

```



Code for Display the Join of Staff and Building table:-

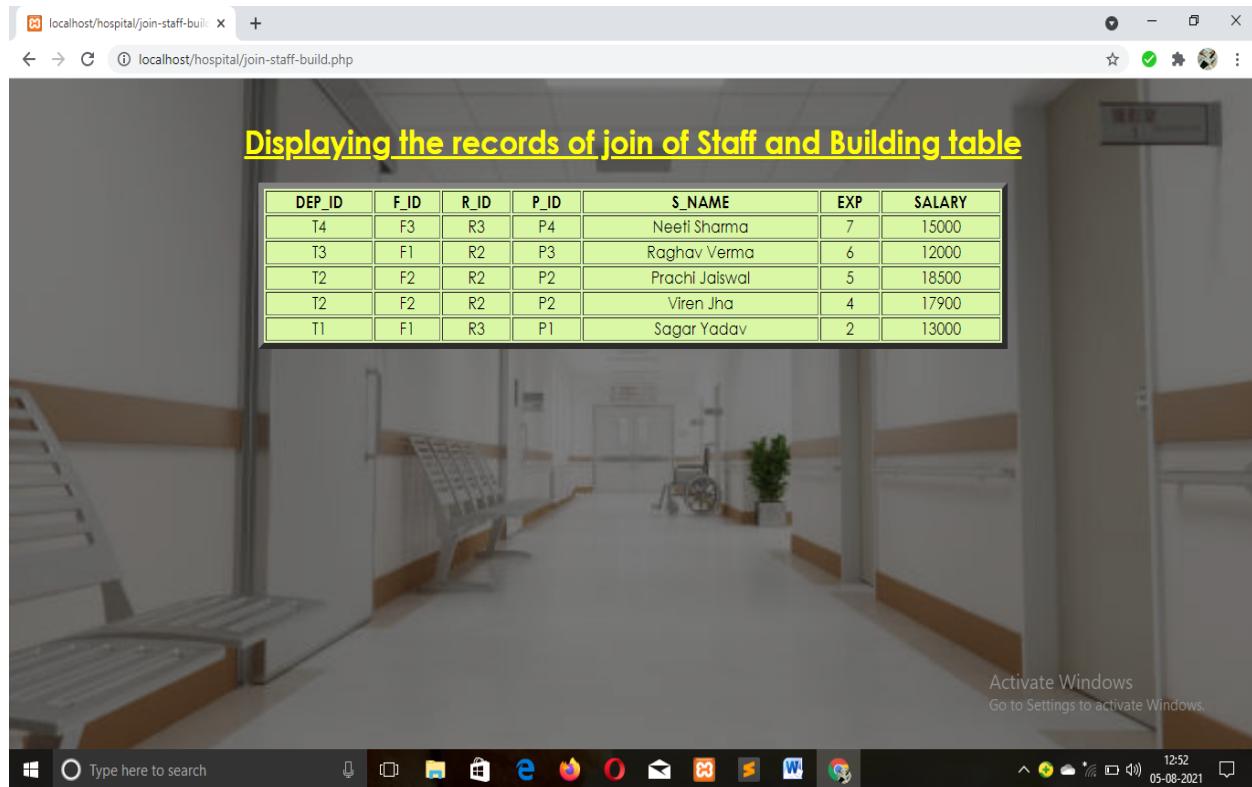
```
<style>
    * {
        margin:0;
        padding:0;
        font-family:century gothic;
    }
    body{
        background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(image.jpg);
        height:100vh;
        background-size:cover;
        background-position:center;
    }
</style>

<?php
include("db_conn.php");
error_reporting(0);
$query="SELECT BU.DepId,BU.Fid,BU.Rid,BU.Pid,S.Sname,S.Exp,S.Salary FROM BUILDING
BU,STAFF S WHERE BU.Bid=S.Bid";
$data=mysqli_query($conn,$query);
$total=mysqli_num_rows($data);
```

```

if($total!=0)
{
    ?
    <br><br>
    <center> <h1 style="color: yellow"> <b> <u> Displaying the records of
join of Staff and Building table </u> </b> </h2> </center><br>
    <center>
        <table border="6" cellspacing="2" cellpadding="2" style="width: 60%;
text-align: center; background-color: #DAF7A6;">
            <tr>
                <th> <font face="Arial">DEP_ID</font></th>
                <th> <font face="Arial">F_ID</font></th>
                <th> <font face="Arial">R_ID</font></th>
                <th> <font face="Arial">P_ID</font></th>
                <th> <font face="Arial">S_NAME</font></th>
                <th><font face="Arial">EXP</font></th>
                <th><font face="Arial">SALARY</font></th>
            </tr>
<?php
while($result=mysqli_fetch_assoc($data))
{
    echo "<tr>
                    <td>".$result['DepId']."'</td>
                    <td>".$result['Fid']."'</td>
                    <td>".$result['Rid']."'</td>
                    <td>".$result['Pid']."'</td>
                    <td>".$result['Sname']."'</td>
                    <td>".$result['Exp']."'</td>
                    <td>".$result['Salary']."'</td>
                </tr>";
}
}
else
{
    echo "NO RECORDS ARE FOUND";
}
?>
</table></center>

```



Code for Corona Bed Booking:-

```
<?php
include("db_conn.php");
error_reporting(0);
?>
<html>
<head>
    <title>CORONA BED BOOKING</title>
<link rel = "stylesheet" href = "stylebedbooking.css">
</head>
<body>
    <center>
        <h1><u>CITY HOSPITAL</u></h1></center>

<form action="display_booking.php" method="POST">
    <center><h2>CORONA BED BOOKING</h2><br></center>
    NAME <input type = "text" placeholder = "Enter Patient's Name" name = "pname" id="pname"><br><br>
    AGE <input type = "text" placeholder = "Enter Patient's Age" name = "age" id="age"><br><br>
    DATE <input type = "text" placeholder = "Enter Date" name = "date" id="date"><br><br>
```

```

TIME <input type = "text" placeholder = " Enter Time" name = "time"
id="time"><br><br>
<input type="file" id="myFile" name="filename"><br><br>
<center><button type="submit" class="button">SUBMIT</button></center>
</form>
</center>
</body>
</html>

```

Css styling code for Corona Bed Booking:-

```

* {
margin:0;
padding:0;
font-family:century gothic;
}

body{
background-image:linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url(booking.jpg);
height:100vh;
background-size:cover;
background-position:center;
}

h1{
position: absolute;
top:50%;
left:50%;
color:black;
transform: translate(-50%, -50%);
font-size:60px;
}

h2{
color:black;
font-size:20px;
}

form{
top:70%;
left:50%;
height:450px;
width:50%;
position:absolute;
border:2px solid black;
background-color:aqua;
padding:40px 40px;
transform: translate(-50%, -50%);
color:black;
}

```

```

}

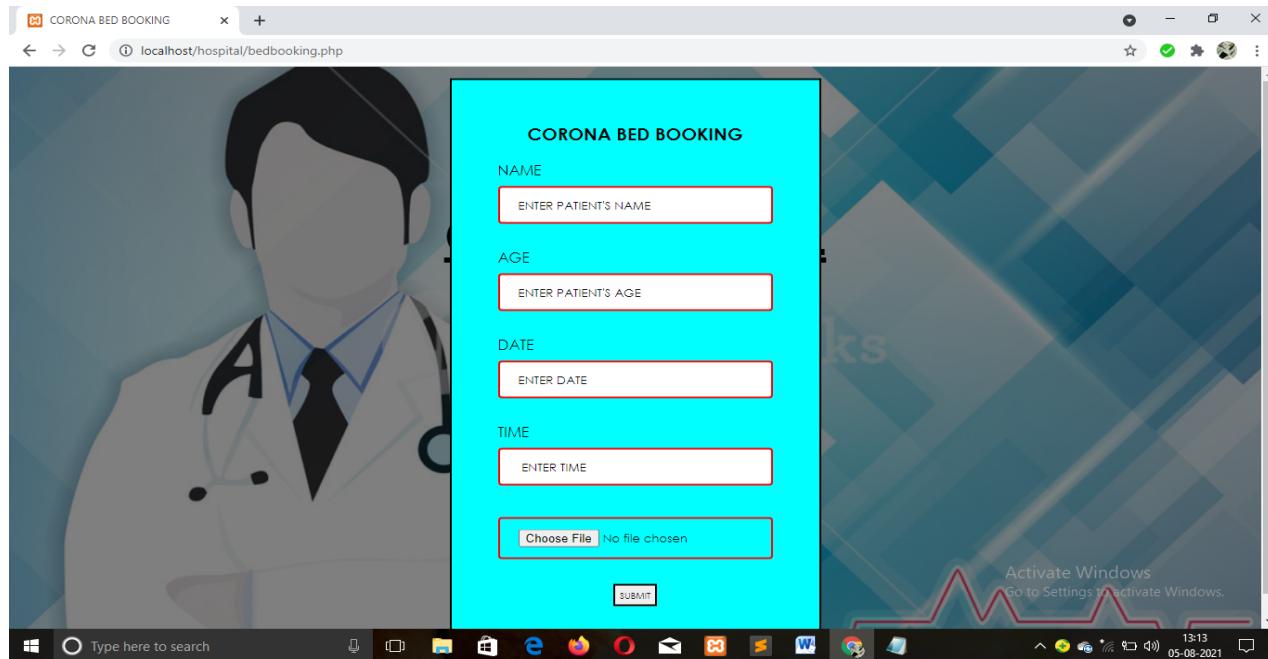
input[type=text] {
    width:80%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;
    border: 2px solid red;
    border-radius: 4px;
}

::placeholder {
    color: black;
    text-transform: uppercase;
}

button{
    color: black;
    padding: 5px 5px;
    font-size:10px;
    transition:0.6s ease;
    border:2px solid ;
    text-decoration:none;
}

button: hover{
    background-color:white;
    color:black;
}

```



CHAPTER-12

CONCLUSION

The project entitled "HOSPITAL MANAGEMENT SYSTEM" is developed by using HTML,CSS,PHP as a front-end platforms and MYSQL database as a back-end platform to computerize the process of getting admitted to the hospital or for online appointments by keeping several records or details in the system. This system will reduce the manual operation required to maintain all records with a paperless motives. The main aim of system development is to develop and implement the system cost effectively; user-friendly and most suited to the user's analysis. Since the Hospital Management System is essential for maintaining detail about the Doctor, Patient, Hospital staff etc. we understand that by using of Hospital Management System project the work became very easy and we save lot of time. Hospital administrators would be able to significantly improve the operational control and thus streamline operations. This would enable to improve the response time to the demands of patient care because it automates the process of collecting, collating and retrieving patient information. Accounting sometimes becomes awfully pathetic and complex. This product will eliminate any such complexity.

CHAPTER-13

FUTURE WORK

1. IN OUR PROJECT WE HAVE MADE BED BOOKING SECTION WHERE ALL THE DETAILS OF PATIENT IS STORED IN DATABASE BUT PDF IS NOT PRINTED. IN FUTURE, WE WANT TO MAKE AN INTERFACE WHERE WE CAN DO BOTH THE WORK I.E. PRINTING THE PDF OF CORONA PATIENT AND STORING IT IN DATABASE.
2. IN FUTURE, WE ARE ALSO THINKING ABOUT APPLYING PROPOSED ALGORITHM WHERE WE CAN DEPEND ON SOME FACTORS FROM CORONA BED BOOKING DETAILS IN WHICH, WHOEVER PATIENT NEEDED THE BED MOST I.E. SEVERE PATIENT WILL BE ALLOTTED THE BED, THERE WILL BE NO FIRST COME SERVICE WILL BE PERFORMED. HERE, THE INITIAL STAGE AND EXTREME STAGE IS THE PARAMETERS AND WE WILL ALSO ATTACH DOCTOR CERTIFICATE AND MANY OTHER FACTORS WHICH WILL BE STORED IN DATABASE AND WHOEVER COMES TO BOOK THE BED, THERE BOOKING REQUEST WILL BE TAKEN BUT THE BED NO. ACKNOWLEDGEMENT WILL BE GIVEN TO THE SEVERE PATIENT WHO NEEDED THE BED MOST.
3. IN BACKEND DATABASE WE HAVE TAKEN RELATIONAL DATABASE, SO IN FUTURE WE WANT TO CHANGE THIS AND WANT TO ESTABLISH OUR DATABASE TO NON RELATIONAL DATABASE WHICH ARE DIVIDED INTO DOCUMENT ORIENTED DATABASE(MONGODB), COLUMNAR DATABASE(CASSANDRA), KEYVALUE ORIENTED DATABASE(REDIS), GRAPH DATABASE. IN NON RELATIONAL DATABASE, WE CAN ADD IMAGE, AUDIO, VLOG, ETC EASILY. IT IS ALSO SCHEMA FREE, OPEN SOURCE LANGUAGE LIKE MONGODB.
4. IN FUTURE WE WANT TO USE OLAP(ONLINE ANALYTICAL PROCESSING) DATABASE WHICH WILL PERFORM COMPLEX ANALYSIS BY SURVEYING THE PATIENT HISTORY AND GENERATE SOME RESULT. SUCH AS, IN WHICH TIME HOW MANY PERSON IS AFFECTED BY MORE SEVERE DISEASE GOT ADMITTED AND DIED AND WHAT WILL BE THE MOST GOOD MEDICATION FOR THE PATIENT WILL BE DEPICTED BY THIS. MEDICATION, DISEASE, PATIENT ID, TIME ARE THE DIFFERENT ATTRIBUTES. THIS THREE FIELDS MEDICINE, TIME AND DISEASE ARE THE ATTRIBUTES WHERE GROUP BY OPERATION WILL BE PERFORMED AND SHOW THE RESULT HOW TO RECOVER THE PATIENT FROM THE DISEASE EASILY.

CHAPTER-14

REFERENCES

1. PHP MySQL Website Programming: Problem- Design- Solution by ChrisLea, 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.
4. Deepak Thomas- "Beginning PHP 4 Databases", Wrox Press Ltd. Paperback-17, October,2002.
5. WWW.Google.com
6. WWW.Yahoo.com
7. WWW.Youtube.com