DOCTOR APPOINTMENT SYSTEM

MINOR PROJECT REPORT

Submitted by

BAVADHARANI S

22BCM508

Under the Guidance of

Dr. C. Nagarani M.Sc., MCA., M.Phil., Ph.D.,

Assistant Professor

Department of Computer Science

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

of Bharathiar University

DEPARTMENT OF COMPUTER SCIENCE

PSG COLLEGE OF ARTS & SCIENCE

An Autonomous College-Affiliated to Bharathiar University

Accredited with 'A++' grade by NAAC (4th Cycle)

College with Potential for Excellence

(Status Awarded by the UGC)

Star College Status Awarded by DBT - MST

An ISO 9001:2015 Certified Institution

Civil Aerodrome Post, Coimbatore -641 014

NOVEMBER 2024

DEPARTMENT OF COMPUTER SCIENCE

PSG COLLEGE OF ARTS & SCIENCE

An Autonomous College-Affiliated to Bharathiar University

Accredited with 'A++' grade by NAAC (4th Cycle)

College with Potential for Excellence

(Status Awarded by the UGC)

Star College Status Awarded by DBT - MST

An ISO 9001:2015 Certified Institution

Civil Aerodrome Post, Coimbatore -641 014

CERTIFICATE

This is to certify that this Project work entitled **"DOCTOR APPOINTMENT SYSTEM USING WEB TECHNOLOGIES"** is a bonafide record of work done by **BAVADHARANI S(22BCM508)** in partial fulfillment of the requirements for the award of Degree of **Bachelor of Science in Computer Science** of Bharathiar University.

_____ _____

Faculty Guide Head of the Department

Submitted for Viva-Voce Examination held on _____

_____ _____

Internal Examiner External Examiner

DECLARATION

I **BAVADHARANI S(22BCM508)**, hereby declare that this Project work entitled **"DOCTOR APPOINTMENT SYSTEM USING WEB TECHNOLOGIES"**, is submitted to PSG College of Arts & Science (Autonomous), Coimbatore in partial fulfillment for the award of Bachelor of Science in Computer Technology, is a record of original work done by me under the supervision and guidance of **Dr. C. Nagarani M.Sc., MCA., M.Phil., Ph.D.,** Assistant Professor in Department of Computer Science, PSG College of Arts & Science, Coimbatore.

This Project work has not been submitted by me for the award of any other Degree/Diploma/ Associate ship/ Fellowship or any other similar degree to any other university.

PLACE : Coimbatore BAVADHARANI S

DATE : **22BCM508**

ACKNOWLEDGEMENT

With great gratitude, I would like to acknowledge the help of those who contributed with their valuable suggestions and timely assistance to complete this work.

First and foremost, I would like to extend my heartfelt gratitude and place my sincere thanks to **Thiru. L. Gopalakrishnan Managing Trustee**, **PSG & Sons‟ Charities,** Coimbatore for providing all sorts of support and necessary facilities throughout the course.

I express my deep sense of gratitude to **Dr. T. Kannaian M.Sc., MTech., PhD., Secretary,** PSG College of Arts & Science for infrastructure provided to undertake this work.

I whole heartedly express my gratitude to **Dr. D. Brindha M.Sc., MPhil., PhD, MA(Yoga) Principal**, PSG College of Arts & Science for her academic support and constant source of inspiration throughout the course.

I am very thankful to **Dr. M Jayanthi, M.Com., MBA, MPhil., PhD., Vice Principal (Student Affairs), Prof. M Umarani, MBA., MPhil., faculty In-Charge (Student Affairs),** for their support.

I am thankful to **Dr. T. Revathi, MCA., MPhil., PhD., Dean, Department od Computer Science** for her support.

I owe my deepest gratitude **to Dr. S. Selvi, MCA., MPhil., PhD., Associate Professor & Head, Department of Computer Science** for her advice and encouragement to complete the project.

My sincere thanks to **Dr. C. Nagarani M.Sc., MCA., M.Phil., Ph.D., Assistant Professor, Department of Computer Science** for the valuable suggestions, support and guidance as my Internal Guide, without which my work would not have reached the present form.

Doctor appointment fixing system

SYNOPSIS

The purpose of Doctor Appointment system is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Doctor appointment system, as described above, can lead to error free, secure, reliable and fast management system. it can assist the user to concentrate on their other activities rather to concentrate on the record keeping .Thus it will help organization in better Utilization of resources. The organization can maintain computerized records without redundant entries. that means that one need not be distracted by information that is not relevant. while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data can be stored for longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients**.**

INTRODUCTION

The Doctor appointment system has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for particular need of the company to carry out operations in smooth and effective manner.

This application is reduced as much as possible to avoid errors while entering the data it also provides error message while entering invalid data. no formal knowledge is needed for the user to use this system. thus by this all it proves it is user-friendly. Doctor appointment system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. thus it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of appointment, doctor, booking, doctor schedule. Every Doctor appointment system has different doctor needs, therefore we design exclusive employee management Systems that are adopted to your managerial requirements. This is designed to assist in strategic planning and will help you ensure that your future goals. Also for those busy executive who will allow you to manage your workforce anytime. At all times. These systems will ultimately allow you better manage resources**.**

2. SYSTEM SPECIFICATION

2.1 HARDWARE SPECIFICATION

Processor	: Dual Core.
Ram	: 2 GB.
Hard Disk	: 300GB.
Compact Disk	: 650 Mb.
Input device	: Standard Keyboard and Mouse.
Output device	: VGA and High Resolution Monitor
Mother Board	:Intel
Speed	: 1 GHZ
System Bus	: 64 bit

2.2 SOFTWARE SPECIFICATION:

Operating System	: Windows XP
Server Used	:XAMPP server v3.2.2
Front End	: PHP
Back End	: My SQL

2.3 SOFTWARE DESCRIPTION:

PHP

PHP (recursive acronym for PHP: Hypertext Pre-processor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours. The fact that PHP was not originally designed but instead was developed organically has led to inconsistent naming of functions and inconsistent ordering of their parameters. In some cases, the function names were chosen to match the lower-level libraries which PHP was "wrapping",[1] while in some very early versions of PHP the length of the function names was used internally as a hash function, so names were chosen to improve the distribution of hash values.

Easy to read

The read() function reads from an open file. The first parameter of open() contains the name of the file to be opened and the second parameter specifies in which mode the file should be opened. The following example also generates a message if the open() function is unable to open the specified file The first parameter of read() contains the name of the file to read from and the second parameter specifies the maximum number of bytes to read.

Easy to Edit

(Anyone with PHP knowledge is welcome to comment on the code. If there are issues I haven't noticed, please let me know. Do realize that it is intended for

beginners, so I didn't want to do anything too advanced that might lead to confusion. Yes, I realize I could use OOP, or could separate some of these out into methods, etc. etc.) Basically, just imagine that you are in charge of a sports team, and you want to keep a list of all your player's contact information. The code I've created could be a starting point for that (it only includes fields for their first name/last name, but could obviously could be expanded to use more fields). How to create a system that allows a user to add/edit/remove data in a database seems to be a commonly asked topic, so I may adapt this into an actual tutorial at some point in the future.

More control

The switch statement is similar to a series of IF statements on the same expression. In many occasions, you may want to compare the same variable (or expression) with many different values, and execute a different piece of code depending on which value it equals to. This is exactly what the statement is for.

do-while loops are very similar to while loops, except the truth expression is checked at the end of each iteration instead of in the beginning. The main difference from regular while loops is that the first iteration of a do-while loop is guaranteed to run (the truth expression is only checked at the end of the iteration), whereas it may not necessarily run with a regular while loop (the truth expression is checked at the beginning of each iteration, if it evaluates to FALSE right from the beginning, the loop execution would end immediately).

Better Performance

PHP is a complex language that has suffered years of twists, bends, stretches, and hacks. It’s highly inconsistent and sometimes buggy. Each version has its own unique features, warts, and quirks, and it’s hard to keep track of what

version has what problems. It's easy to see why it gets as much hate as it does sometimes.

Despite that, it's the most popular language on the web today. Because of its long history, you'll find lots of tutorials on how to do basic things like password hashing and database access. The problem is that out of five tutorials, you have a good chance of finding five totally different ways of doing something. Which way is the "right" way? Do any of the other ways have subtle bugs or It's really hard to find out, and you'll be bouncing around the internet trying to pin down the right answer.

PHP Is Widely Available

That's also one of the reasons why new PHP programmers are so frequently blamed for ugly, outdated, or insecure code. They can't help it if the first Google result was a four-year-old article teaching a five-year-old method This document tries to address that. It's an attempt to compile a set of basic instructions for what can be considered best practices for common and confusing issues and tasks in PHP. If a low-level task has multiple and confusing approaches in PHP, it belongs here.

In older versions of PHP, every time a script was executed it would have to be compiled from scratch, even if it had been compiled before. Opcode caches were additional software that saved previously compiled versions of PHP, speeding things up a bit. There were various flavours of caches you could choose from.

MYSQL

Oracle MySQL Cloud Service delivers a secure, cost-effective and enterprise-grade MySQL database service. Built on MySQL Enterprise Edition

and powered by the Oracle Cloud, it provides a simple, automated, integrated and enterprise ready MySQL cloud service, enabling organizations to increase business agility and reduce costs.

MySQL Enterprise High Availability

My SQL Inno DB Cluster tightly integrates My SQL Server with Group Replication, My SQL Router, and My SQL Shell, so you don't have to rely on external tools, scripts or other components. Plus, it leverages proven My SQL features including InnoDB, GTIDs, binary logs, multi-threaded slave execution, multi-source replication and Performance Schema. A MySQL InnoDB Cluster can be set up in less than five minutes and managed using the scriptable AdminAPI in the MySQL Shell.

Group Replication provides native high availability with built-in group membership management, data consistency guarantees, conflict detection and handling, node failure detection and database failover related operations, all without the need for manual intervention or custom tooling.

Group Replication implements both a single-primary mode with automatic leader election and a multi-master update everywhere mode. By using a powerful new group communication system, which provides an in-house implementation of the popular Paxos algorithm, the group automatically coordinates on data replication, consistency, and membership. This provides all of the built-in mechanisms necessary for making your MySQL databases highly available.

Client /server Architecture

Client/server architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client. This type of architecture has one or more client computers connected to a central server over a network or internet connection. This system shares computing resources. Client/server architecture is also known as a networking computing model or client/server network because all the requests and services are delivered over a network. The **client–server model** is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers .which await incoming requests. Examples of computer applications that use the client–server model are Email, network printing, and the World Wide Web.

3. SYSTEM ANALYSIS

3.1 EXISITING SYSTEM

If anybody is ill and wants to visit a doctor for check-up, he or she needs to visit the hospital and waits until the doctor is available. The patient also waits in a queue while getting
appointment. If the doctor cancels the appointment for some emergency reasons then the patient is not able to know about the cancelation of the appointment unless or until he or she visits the hospital. As the mobile communication technology is developing rapidly, therefore, one can use the mobile's applications to overcome such problems and inconvenience for the patients.

DRAWBACKS

- The appointment record maintenance takes more time consuming.
- Slow retrieval of appointment information.
- It is not satisfy user requirements.
- Hard to operate manual appointment fixing.

3.2 PROPOSED SYSTEM

Life is becoming too busy to get medical appointments in person and to maintain a proper health care. he main idea of this work is to provide ease and comfort to patients while taking appointment from doctors and it also resolves the problems that the patients has to face while making an appointment.

The proposed system contain two modules.

One module is the application designed for the patient that contains a login screen. The patient has to register himself before logging in to the application. After logging in, the patient can select a hospital and can view the hospital details. The patient has the option of selecting a doctor from the list of doctors and can view the doctor's details. The patient can request for an appointment on his/her preferred day/time. The selected day/time slot will be reserved and patient will receive the notification of the successfully added appointment. second module is the admin module that is designed on the website. The admin views all details of doctors and all appointments by the admin. The admin can add doctor, view patient's details and doctor's details and can view appointments also. All the doctors of the specific clinic are registered by the admin. Doctors cannot register themselves.

.

FEATURES

- Creating and changing issue at ease.
- User accounts to control the access and maintain security
- Simple status and resolutions
- Robust database back-end
- Accuracy in work.
- It contain better storage capacity.
- Easy and fast retrieval of information
- Work become very speedy.
- Easy to update information

4. SYSTEM DESIGN AND DEVELOPMENT

4.1 FILE DESIGN

The file design manages access to the data and the metadata of the files, and manages the available space of the device(s) which contain it. Ensuring reliability is a major responsibility of

a file system. A file design organizes data in an efficient manner, and may be tuned to the characteristics of the backing device.

Some file design are used on data storage devices, to maintain the locations of the files on the device. Others provide access to files residing on a server, by acting as clients for a network protocol. Others provide access to data that is not stored on a persistent device, and/or may be computed on request. This is distinguished from a directory service and registry.

4.2 INPUT DESIGN

It is the process of converting the user oriented input to computer based format. In the system design phase the expanded dataflow diagram identifies logical data flow, data stores destination. Input data is collected and organized into groups of similar data.

The goal begin designing input data is to make the data entry easy, and make it free from logical errors ,the input entry to all customer user name , password .only if the are valid the customer is allowed to enter in to the software.

Thus careful design of input stages has taken place by giving attention to error handling, controls, batching and validation procedures

In the project, The input design is made in web application form with various technique.

For example, in the patient creation table, the empty patient name is not allowed. the patient name is exist in the database does not allow to create patient. The patient table contain Fields like name, doctorname,date,time,disease.

4.3 OUTPUT DESIGN

Compute Output is the most important and direct source of information to the user efficient intelligible output design should improve system's relationship with the user and health in decision making. Output design generally refers to the results generated by the system. For many end users on the base of the output . the evaluate the usefulness of the applications. An efficient software must be able to produce efficient and effective reports.

The reports may be of

a. Administrator side report

b. patient side report

In the project output forms are containing reports format like medical practitioner authority number and etc.

4.4 DATABASE DESIGN

Database is to make information access easy , quick ,inexpensive and flexible for the user .In database design several specific object is are considered: A General theme begin a database is to handle information as an integrated whole

.A database is a collection of inter related data stored with minimum redundancy to serve many users quickly and efficiently .The general objective .

CONTROLED REDUNDANCY

Redundant data occupies space and there fore, is wasteful if versions of the same date are in different phase of updating a system often gives conflicting information .A unique aspect of database design is storing data only once ,which controls redundancy and improves system performance

DATA INDEPENDENCY

An important database objective is changing hardware and storage procedures for adding new data without having to read write application programs.

ACCURACY AND INTEGRITY

The accuracy and database ensures that data quality content remain constant .Integrity controls detect data inaccuracies where the occurred.

SYSTEM SECURITY

For data to remain private, security measures must be taken to prevent unauthorized access. Database security means the data are protected from various forms of destruction .users must be positively identified and actions monitored.

Managing the database requires a database administrator (DBA) whose key functions are to be managed data activities the database structure, and the DBMS.

In the project database name doctor. Table names are tbl_admin,tbl_appointment,tbl_clientregistration,tbl_description,tbl_doctor,tbl_feedback,

tbl_patientregistration. The patient registration table contain Fields like Name, Email, Password, Mobile, Gender, Address, dob.

4.5 SYSTEM DEVELOPMENT

Top-Down approaches emphasis planning and a complete understanding of the system. it is inherent that no coding can begin until a sufficient level of detail has been reached in the design of at least some part of system. Programming, actually writing software code is just one part of the process, which is why people prefer to be called developers rather than merely programmers.

Top-down programming is a programming style, the mainstay of traditional procedural language, in which design begins by specifying complex task and then dividing them is not separate modules, Eventually, the components are specific enough to be coded and the program is written.

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards through a phases of initiation, analysis, design, code, testing, implementation and maintenance.

5. MODULE DESCRIPTION

Administrator login

The administrator interface allows the administrator to be able to implement editing, inserting,deleting and to check the overall result of all registered outpatients. An administrator could be a doctor or nurse who is able to schedule appointment, generate appointment report and view the appointment report of patients that has registered all the necessary information needed. The patient has

a patient identification number in order to search for the information stored in the database of the system.

Patient Registration Module:

There is need for every new patient to register before they can access the system, a patient must have a username and password which they will use to login before they can book for medical appointment. Any patient that does not register will not be allowed to access appointment booking Menu because to login to the appointment Menu requires a username and a password.

Patient Appointment Home:

The patient appointment Menu is to allow the patient to book for medical appointment based on the calendar already viewed. All the fields provided must be entered and then clink on submit button, the output Menu will be display. The appointment output Menu shows the confirmation of the appointment already booked by a patient with a doctor. The appointment will then be treated by the administrator for final confirmation, the appointment can then be approved or disapproved based on the doctor availability. The patient can also print as a confirmation for their appointment. The patient can then view the appointment status we either the appointment has been approved or not.

6. TESTING AND IMPLEMENTATION

System Testing

System testing makes a logical assumption that if all the pairs of the system are correct. Software testing is a crucial element of the software quality, assurance and represents interesting anomaly for the software during earlier definition and development phases, it was attempted to build software from an abstract concept to a tangible implementation.

The testing phases of the developed system using various test data preparation. This plays a vital role in system testing. After preparing the test data the system under study was tested using those tested data. While testing the system, errors were found and corrected by using the following testing steps and corrections are also noted for future use.

Testing Methodologies

Anything may be the system; testing phase is the final and important phase for it to be success. It is the stage of implementation, which is aimed at ensuring that the system works accurately and effectively before live operation commences. System testing makes a logical assumption that if all parts of the system are correct the goal will be successful. The user tests the developed system and changes are made according to their needs.

The various types of testing done on the system are,

- Unit testing
- Integrated testing
- database testing
- Validation Testing

UNIT TESTING

This kind of testing is to verify the smallest unit of the software module. This is also known as “Module Testing”. This test is carried out during the programming stage. This test ensures the expected output from each of the module. Exceptions have been handled and appropriate Error messages have been given in each module so as to avoid abnormal termination of the program.

Patient Login module

If entered Patient login details integrate the database and check the Patient registration table .if registration table value matches login page redirected to Patient home page. else the error message has been displayed.

INTEGRATION TESTING

This kind of testing is a systematic testing for constructing tests to uncover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole.

The system underwent a series of Integration tests that recorded smooth transmission of data from one module to the other. In this project the developed system is tested after integrating various modules together, and the detected errors were corrected.

Patient Login and Appointment status module

The Patient login module and Appointment status module are integrated and tested together, For example if Patient registered him details .that data will be displayed in admin module. The admin login will be allocated status for Patient in Patient status table entries done by using Patient login module

DATABASE TESTING

Database Testing is checking the schema, tables, triggers, etc. of the database under test. It may involve creating complex queries to load/stress test the database and check its responsiveness. It Checks data integrity and consistency.

VALIDATION TESTING:

It is done to validate the input given by the user. The user inputs are checked for their correctness and range. If there are errors, the error message is given and the user is prompted to enter a new value. If the user types numeric value in the place of character values, an error message is displayed.

For Example,

The Patient appointment status allocation table contain patient name,age and registration number. To check whether the registration number is entered or not, To check appointment date and time will be delivered to patient home page. The patient name must not be empty. The date and time must be entered.

7. CONCUSION

the development of web based doctor appointment booking system, patients are able to book and manage their own appointment with ease. They will be reminded of their appointments via login to him website that will be promptly sent to them before their appointment date.

The system itself also provides a quick view of their appointment at the Home page. These functions could indirectly help to reduce the number of missed medical appointments and patients no-show up for their appointment.

8.SCOPE FOR FUTURE ENHANCEMENT

The system delegates some administrative work to the patients by allowing them to manage their
own appointment and personal profiles. Time will not be wasted on converting paper based
appointment record into electronic -based. The system further helps to reduces healthcare
personnel workload by allowing them to generate medical reports easily. They could now
Maximize their competence and allocate more time to maximize service quality.

Enhancement can be done in efficient manner. We can update the following details.

- The doctor appointment system web portal providing automatic remainder for new ideas.
- The doctor appointment system website used send notification through mail or sms.

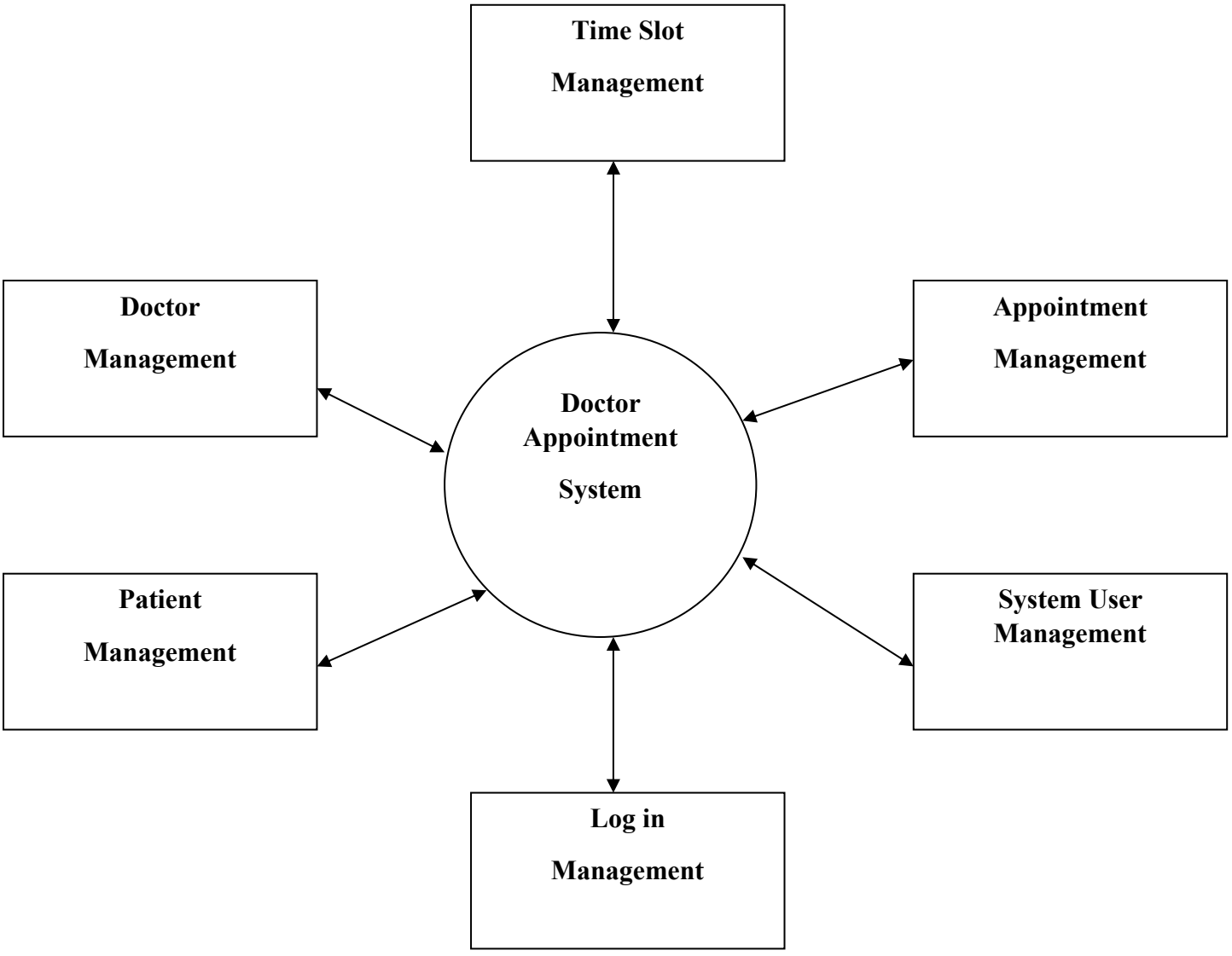
9.BIBIOGRAPHY

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

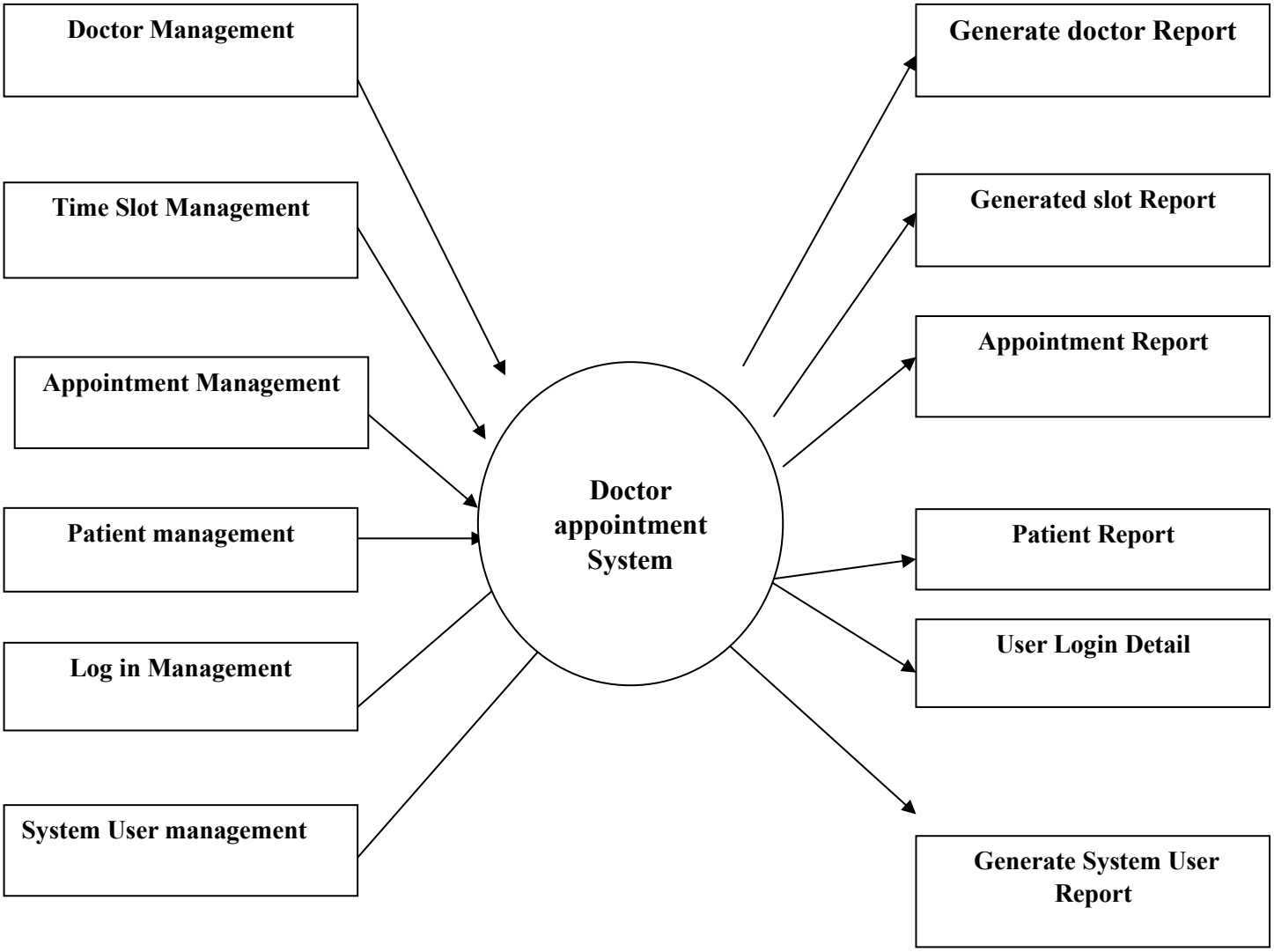
10.SYSTEM DESIGN

10.1DATA FLOW DIAGRAM

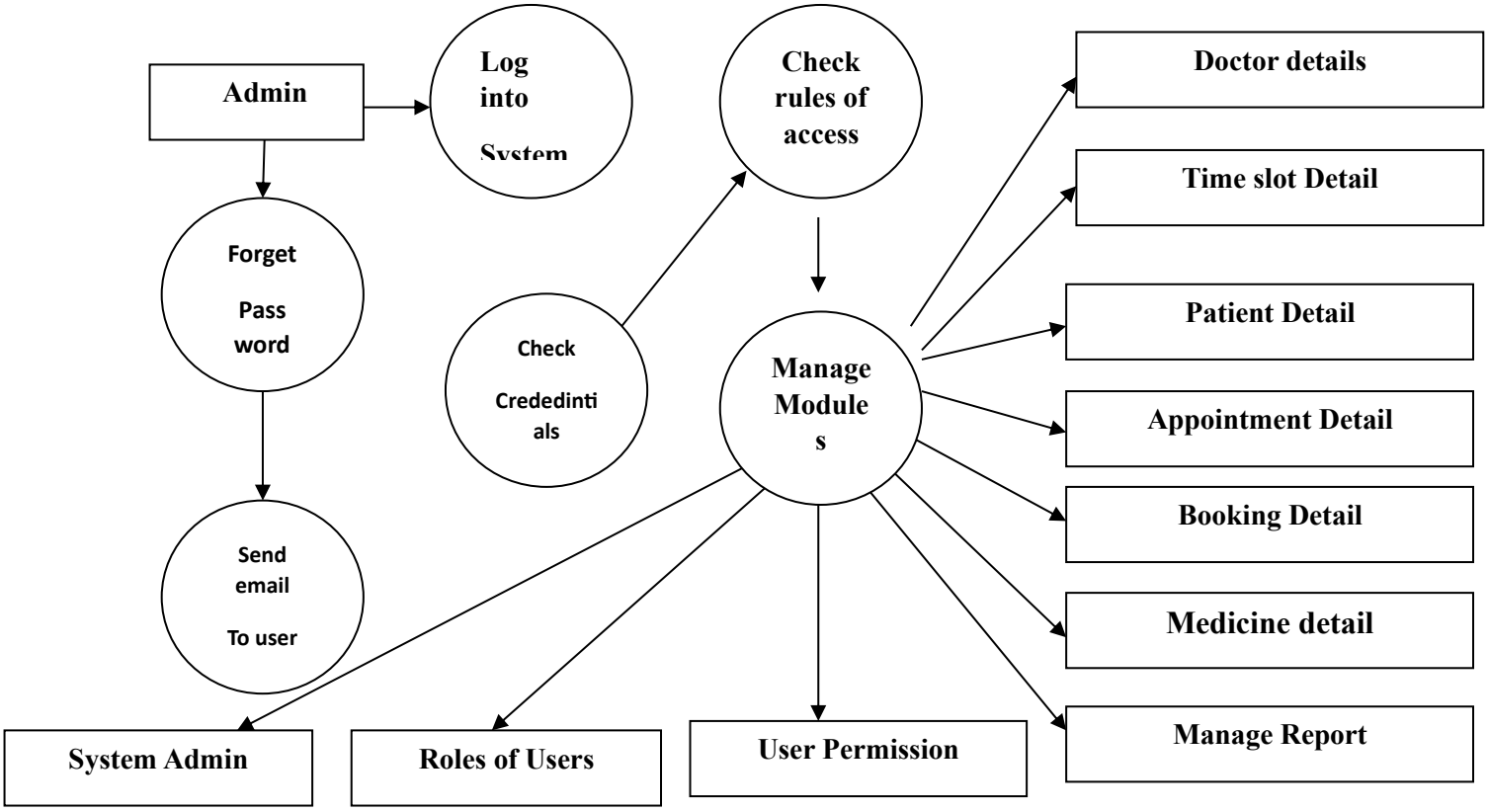
0 level Dfd



1 level diagram:-



2 Level Diagram:-



10.2.TABLE STRUCTURE

Table name : tbl_admin

S.NO	FIELD NAME	FIELD TYPE	SIZE	DESCRIPTION
1	Username	VARCHAR	5000	USER name
2	Password	VARCHAR	45	Pass word

Table name: tbl_appointment

S.NO	FIELD NAME	FIELD TYPE	SIZE	DESCRIPTION
1	Appointment id	INT	10	Appoinment id
2	Category	VARCHAR	5000	Category
3	Doctor	VARCHAR	5000	Doctor
4	Appointment	VARCHAR	5000	Appointment
5	Timing	VARCHAR	45	Timing
6	email	VARCHAR	5000	Email id
7	Doctor mail	VARCHAR	5000	Doctor mail id

Table name: tbl_clientregistration

S.NO	FIELD NAME	FIELD TYPE	SIZE	DESCRIPTION
1	Name	VARCHAR	5000	Name detail
2	Mail	VARCHAR	5000	Mail id
3	Password	VARCHAR	5000	Password
4	Mobile	VARCHAR	5000	Mobile number
5	Gender	VARCHAR	50	Gender
6	Address	VARCHAR	5000	Address
7	dop	VARCHAR	5000	Date of birth

Table name: tbl_description

S.NO	FIELD NAME	FIELD TYPE	SIZE	DESCRIPTION
1	User id	INT	10	User id
2	Name	VARCHAR	4500	Name detail
3	Treatment for	VARCHAR	5000	Treatment for detail
4	Treatment	VARCHAR	45	Treatment

5	Note	VARCHAR	45	Note

Table name: tbl_doctor

S.NO	FIELD NAME	FIELD TYPE	SIZE	DESCRIPTION
1	Doctor name	INT	10	DOCTOR NAME
2	Doctored	VARCHAR	500	Doctor id
3	Name	VARCHAR	5000	Name
4	Address	VARCHAR	5000	Address
5	Mobile number	VARCHAR	5000	Mobile number
6	Category	VARCHAR	5000	Category
7	email	VARCHAR	5000	e-mail id

Table name: tbl_feedback

S.NO	FIELD NAME	FIELD TYPE	SIZE	DESCRIPTION
1	feedback	VARCHAR	8000	Feedback detail

Table name: tbl_patientregistration

S.NO	FIELD NAME	FIELD TYPE	SIZE	DESCRIPTION
1	Name	VARCHAR	5000	Name detail
2	Email	VARCHAR	5000	e-mail id
3	Password	VARCHAR	5000	password
4	Mobile	VARCHAR	5000	Mobile
5	Gender	VARCHAR	45	Gendr
6	Address	VARCHAR	5000	Address

7	dob	VARCHAR	5000	Date of birth

11.APPENDICES

11.1. SAMPLE CODING

Admin Home

```
<?php
include('connection.php');
session_start();?>
<script type="text/javascript" src="js/jquery-1.7.1.min.js"></script>
      <script type="text/javascript" src="js/jquery-ui-1.8.17.custom.min.js"></script>
      <link rel="stylesheet" type="text/css"
       href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css" />
      <script type="text/javascript">
            $(document).ready(function(){
                $("#date").datepicker();
            });
      </script>
<?php
extract($_POST);
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "online_notice";

// Create connection
```

```
$conn = new mysqli($servername, $username, $password, $dbname);

if(isset($save)) {

if($e=="" || $p=="") {

$err="<font color='red'>fill all the fileds first</font>"; }

Else {

$pass=$p;

$sql=mysqli_query($conn,"select * from user where email='$e' and pass='$pass'");

echo "select * from user where email='$e' and pass='$pass'";

$r=mysqli_num_rows($sql);

if($r==true) {

$_SESSION['user']=$e;

header('location:User_Home.php'); }

else {

$err="<font color='red'>Invalid login details</font>";

} } } ?>

<form method="post" >

<img src="images/head.jpg" />

<table width="940" border="2">

<tr><td width="147" height="39"><a href="Add_doctor.php">Add Doctor</a></td>

<td width="148"><a href="View_Doctor.php">View Doctor</a></td>

<td width="289"><a href="View_Customer.php">View Customer</a></td>

<td width="163"><a href="View_Appoinment.php">View Appoinment</a></td>

<td width="163"><a href="View_Description.php">View Description</a></td>

<td width="157"><a href="Home.php">Logout</a></td>
```

```
</tr>

</table>

</form>
```

Appointment_act.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Untitled Document</title>

</head>

<?php

include('connection.php');

session_start();?>

<body>


<?php


$Aid = $_POST['Aid'];

$Category = $_POST['category'];

$doctor = $_POST['doctor'];

$email =$_POST['email'];

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

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

$dd =$_POST['dd'];

$time =$_POST['Time'];

$user=$_SESSION['user'];

echo $Aid;

echo $Category;

echo $doctor;

echo $time;

echo $yy;

echo $mm;

echo $dd;

$dob=$yy.'-'.$mm.'-'.$dd;

echo $dob;

$servername = "localhost";

$username = "root";

$password = "root";

$dbname = "doctor";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect_error)

{

   die("Connection failed: " . $conn->connect_error);

}
```

```
$query = mysqli_query($conn,"SELECT * FROM tbl_appointment where appointmentdate like '$dob' and timings like '$time' ;");


  if (mysqli_num_rows($query) != 0)
  {
     echo "<script>
alert(' Appoinment already booked Try another time');
window.location.href='Patient_Home.php';
</script>";
  }
  else{

$sql = "INSERT INTO tbl_appointment VALUES ('$Aid', '$Category', '$doctor','$dob','$time','$user','$email')";
echo $sql;
if ($conn->query($sql) === TRUE) {
    echo "New Appoinment registered successfully";
                    echo "<script>
alert('New Appoinment registered successfully');
window.location.href='Patient_Home.php';
</script>";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
}
```

```
$conn->close();

?>

</body>

</html>

.
```

Bookappointment.php

```
<?php

include('connection.php');

session_start();?>

<script type="text/javascript" src="js/jquery-1.7.1.min.js"></script>

      <script type="text/javascript" src="js/jquery-ui-1.8.17.custom.min.js"></script>
```

```
<link rel="stylesheet" type="text/css"
 href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css" />
<script type="text/javascript">
        $(document).ready(function(){
            $("#date").datepicker();
        });
</script>
<?php
extract($_POST);
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "online_notice";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
if(isset($save)) {
if($e=="" || $p=="") {
$err="<font color='red'>fill all the fileds first</font>"; }
Else {
$pass=$p;
$sql=mysqli_query($conn,"select * from user where email='$e' and pass='$pass'");
echo "select * from user where email='$e' and pass='$pass'";
$r=mysqli_num_rows($sql);
```

```
if($r==true) {
$_SESSION['user']=$e;
header('location:User_Home.php'); }
else {
$err="<font color='red'>Invalid login details</font>";
} } } ?>

<form method="post" >

<img src="images/head.jpg" />
<table width="940" border="2">
<tr><td width="101"><a href="Mydetails.php">My Details</a></td>
<td width="114"><a href="Book_Appoinment.php">Book Appoinment</a></td>
<td width="212"><a href="View_Booking.php">View Booking</a></td>
<td width="119"><a href="Cancel_Booking.php">Cancel Booking</a></td>
<td width="107"><a href="Search_Doctor.php">Search Doctor</a></td>
<td width="126"><a href="Feedback.php">FeedBack</a></td>
<td width="113"><a href="Home.php">Logout</a></td>

</tr>
</table>

<h2 align="center">New Booking</h2>
<table align="center" bgcolor="#00CC99" border="2"><tr><td width="226">
```

<label>Appoinment ID </label></td><td width="210"><input type="text" name="bid" />

</td></tr>

<tr><td><label>Select Category</label> </td><td>

<select name="Category">

<option value="select" selected="selected">------Select------</option>

<option value="GeneralPhysician">GeneralPhysician</option>

<option value="Bone">Bone</option>

<option value="Heart">Heart</option>

<option value="Dentist">Dentist</option>

<option value="Kidney">Kidney</option>

<option value="Plastic Surgeon">Plastic Surgeon</option>

<option value="Nuralagist">Nuralagist</option>

<option value="Cardiologist">Cardiologist</option>

</select>

</td></tr>

<tr><td><label>Select Doctor</label> </td><td>

<select name="doctor">

<option value="select" selected="selected">------Select------</option>

<option value="Abinaya">Abinaya</option>

<option value="Nithya">Nithya</option>

<option value="Divya">Divya</option>

<option value="Senthil">Senthil</option>

<option value="Priya">Priya</option>

```
<option value="Mahalakshmi">Mahalakshmi</option>

<option value="Saranya">Saranya</option>

<option value="Prabha">Prabha</option>


</select>

</td></tr>


<tr>

                <td>Choose Appoinment Date </td>

                <Td>

<select name="yy" required>

                <option value="">2017</option>

<?php

for($i=2017;$i<=2020;$i++) {

echo "<option>".$i."</option>"; } ?> </select>

<select name="mm" required>

                <option value="">4</option>

<?php

for($i=1;$i<=12;$i++) {

echo "<option>".$i."</option>"; } ?>

</select><select name="dd" required>

<option value="">10</option>

<?php

for($i=1;$i<=31;$i++) {
```

```
echo "<option>".$i."</option>"; }?>

</select> </td></tr>


<tr><td colspan="2" align="center"><input type="button" value="check"/>  </td></tr>


<tr><td> choose timings<td><select ><option>9.30-10.00</option> <option>10.00-
10.30</option> <option>10.30-11.00</option> <option>11.00-11.30</option>
<option>11.30-12.00</option>

<option>1.00-1.30</option>  <option>1.30-2.00</option>  <option>2.00-2.30</option>
<option>2.30-3.00</option>  <option>3.00-3.30</option>  <option>4.00-4.30</option>


<option>6.00-6.30</option>  <option>6.30-7.00</option>  <option>7.00-7.30</option>
<option>7.30-8.00</option>  <option>8.30-9.30</option>  <option>9.30-10.00</option>

</select>

</td></tr>

<tr align="center"><td colspan="2"><input type="submit" value="Book"></td></tr>

</table>

<?php

?>

</form>
```

Viewappointment.php

```
<?php

include('connection.php');

session_start();?>

<script type="text/javascript" src="js/jquery-1.7.1.min.js"></script>
```

```
<script type="text/javascript" src="js/jquery-ui-1.8.17.custom.min.js"></script>
<link rel="stylesheet" type="text/css"
 href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css" />
<script type="text/javascript">
        $(document).ready(function(){
            $("#date").datepicker();
        });
</script>
<?php
extract($_POST);
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "online_notice";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
if(isset($save)) {
if($e=="" || $p=="") {
$err="<font color='red'>fill all the fileds first</font>"; }
Else {
$pass=$p;
$sql=mysqli_query($conn,"select * from user where email='$e' and pass='$pass'");
echo "select * from user where email='$e' and pass='$pass'";
```

```
$r=mysqli_num_rows($sql);

if($r==true) {

$_SESSION['user']=$e;

header('location:User_Home.php'); }

else {

$err="<font color='red'>Invalid login details</font>";

} } } ?>


<form method="post" >


<img src="images/head.jpg" />

<table width="940" border="2">

<tr><td width="147" height="39"><a href="Add_doctor.php">Add Doctor</a></td>

<td width="148"><a href="View_Doctor.php">View Doctor</a></td>

<td width="289"><a href="View_Customer.php">View Customer</a></td>

<td width="163"><a href="View_Appoinment.php">View Appoinment</a></td>


<td width="157"><a href="Home.php">Logout</a></td>


</tr>

</table>


<h2 align="center">Appoinments </h2>
```

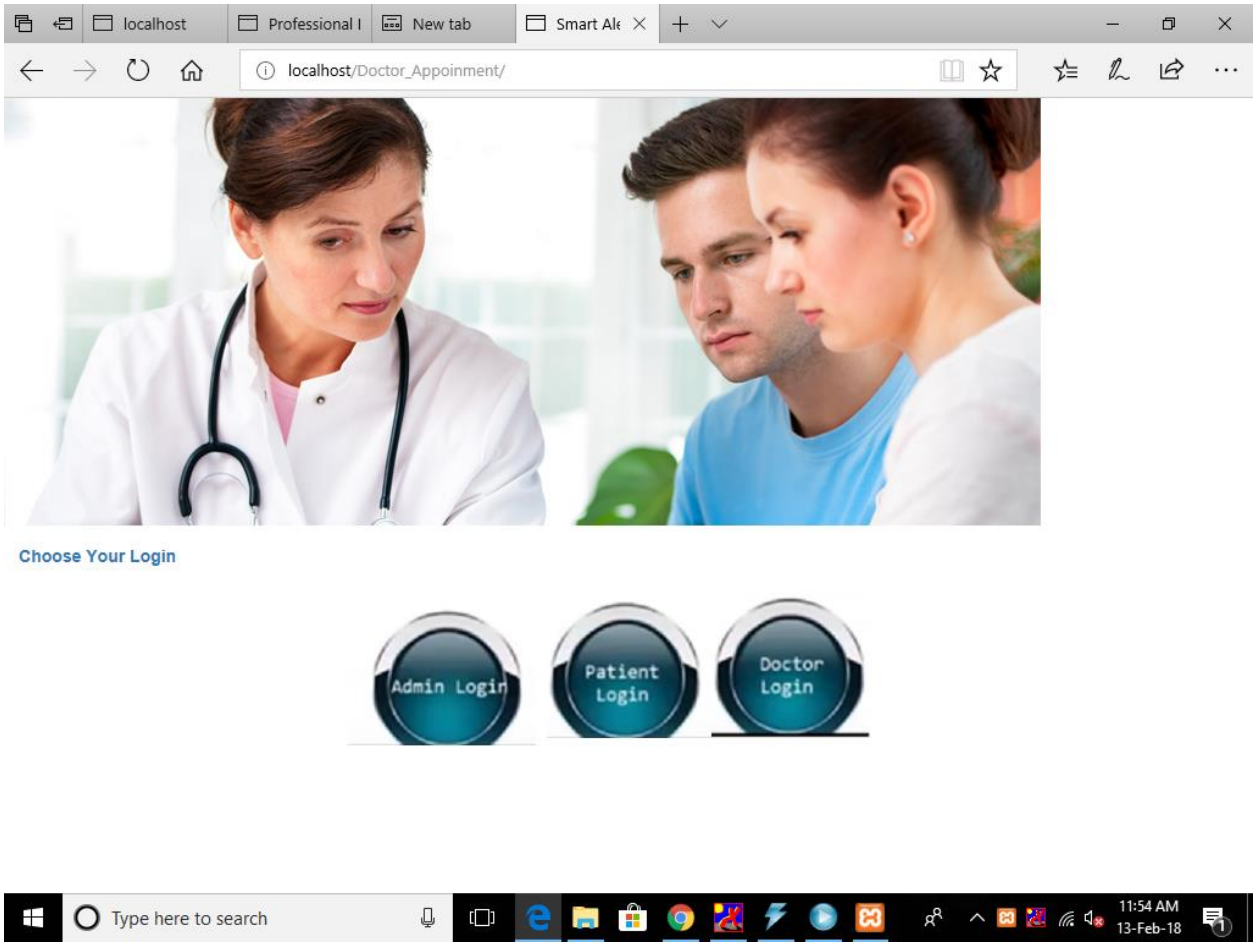
```
<table width="425" border="1" align="center">
		<tr>
			<td width="63"> </td>
			<td width="89">User Name </td>
			<td width="78">Appoinment Date </td>
			<td width="101">Appoinment Time </td>
			<td width="60"> </td>
		</tr>
		 <?php
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "doctor";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$user=$_SESSION['user'];

$sql = "SELECT * FROM tbl_appointment";
```

```
$result = $conn->query($sql);


if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "<tr><td> ".$row["email"]."</td>" ;
                    echo "<td> ".$row["doctor"]."</td>" ;
                     echo "<td> ".$row["appointmentdate"]."</td>" ;
                      echo "<td> ".$row["timings"]."</td></tr>" ;


    }
} else {
    echo "0 results";
}
$conn->close();
?> </table>
</form>
```

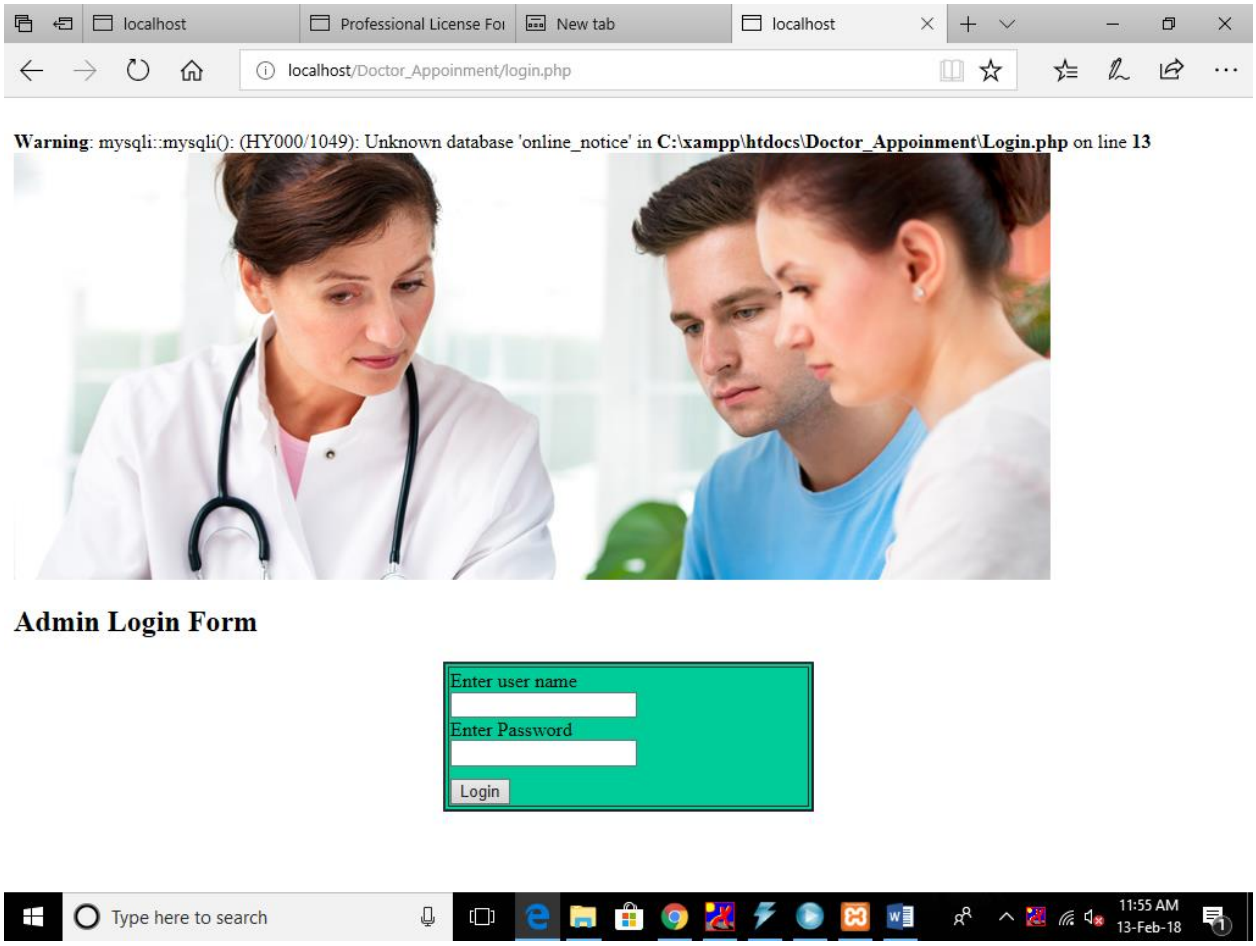
11.2.SAMPLE INPUT

INPUT DESIGN

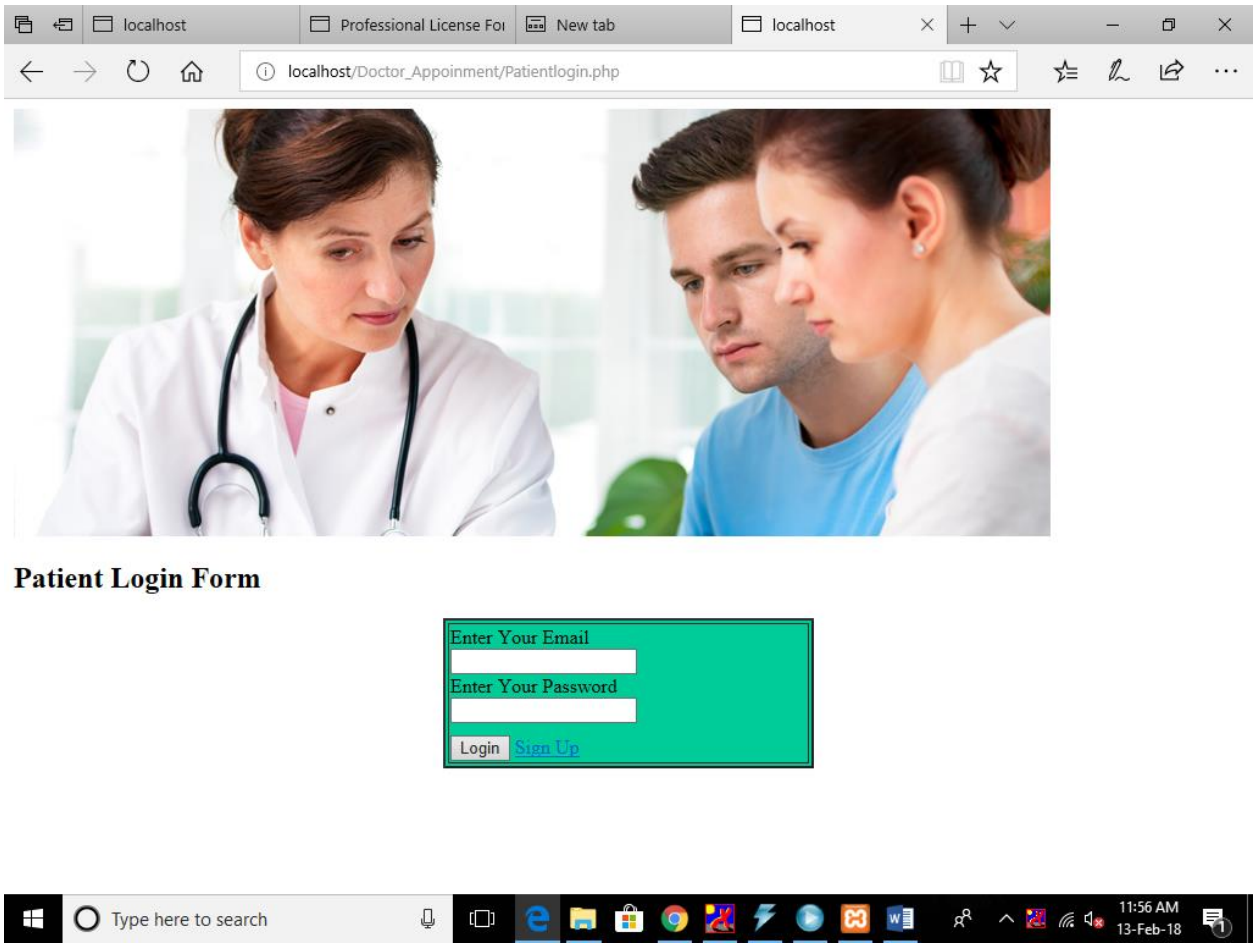
Home



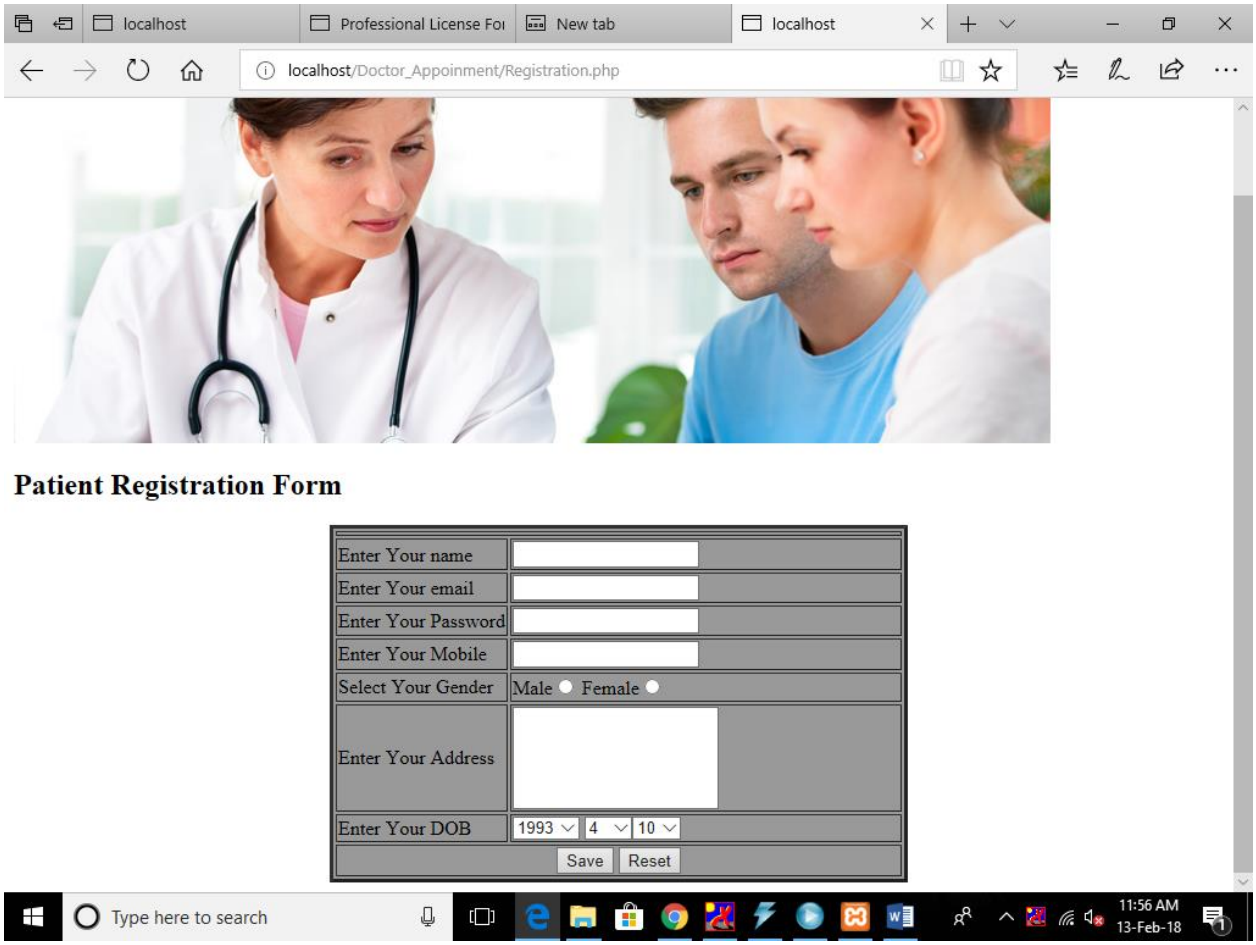
Admin login form


localhost
Professional License For
New tab
localhost
localhost/Doctor_Appoinment/login.php
Warning: mysqli::mysqli(): (HY000/1049): Unknown database 'online_notice' in C:\xampp\htdocs\Doctor_Appoinment\Login.php on line 13
Admin Login Form
Enter user name
Enter Password
Login
Type here to search
11:55 AM
13-Feb-18

Patient login form

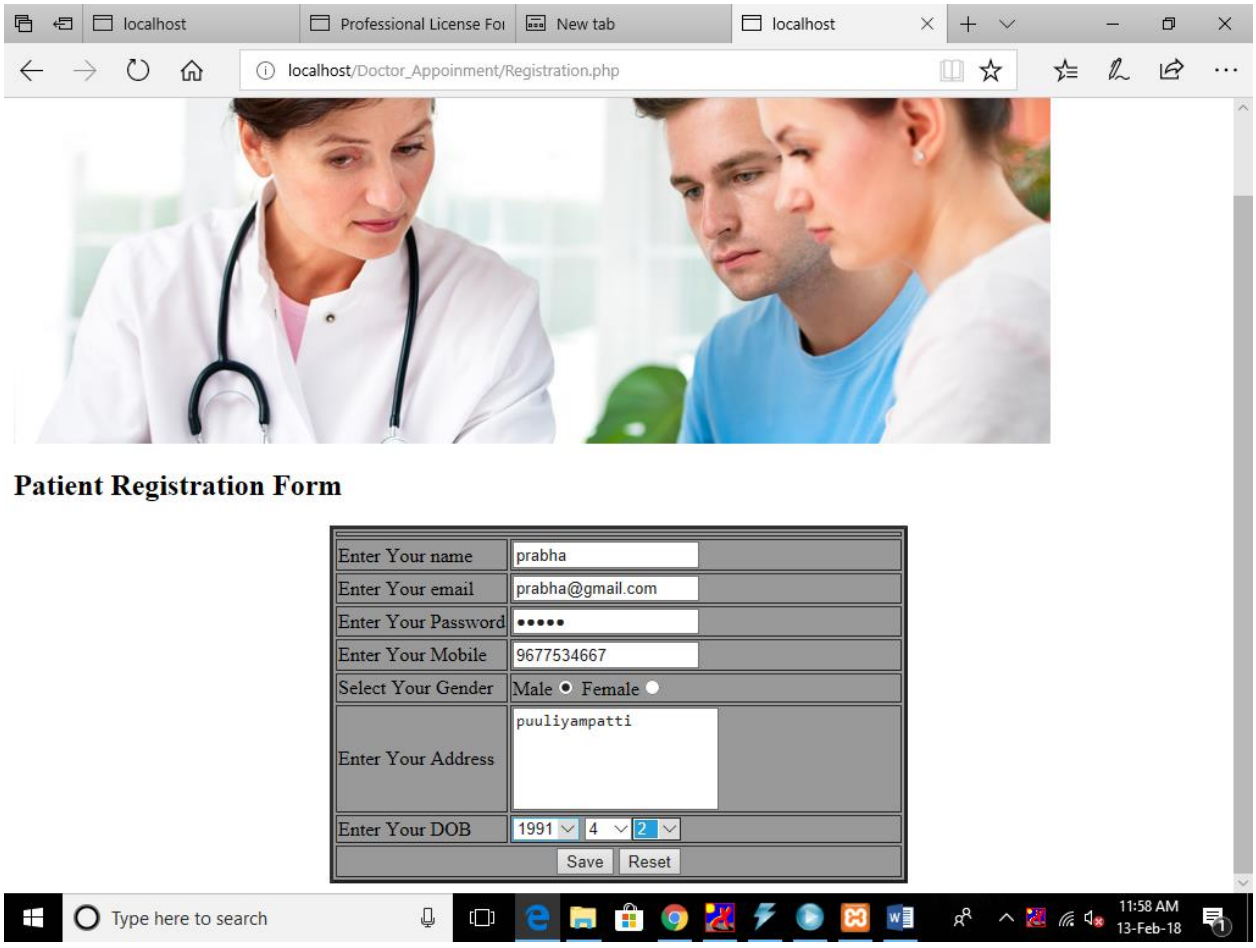


Registration form

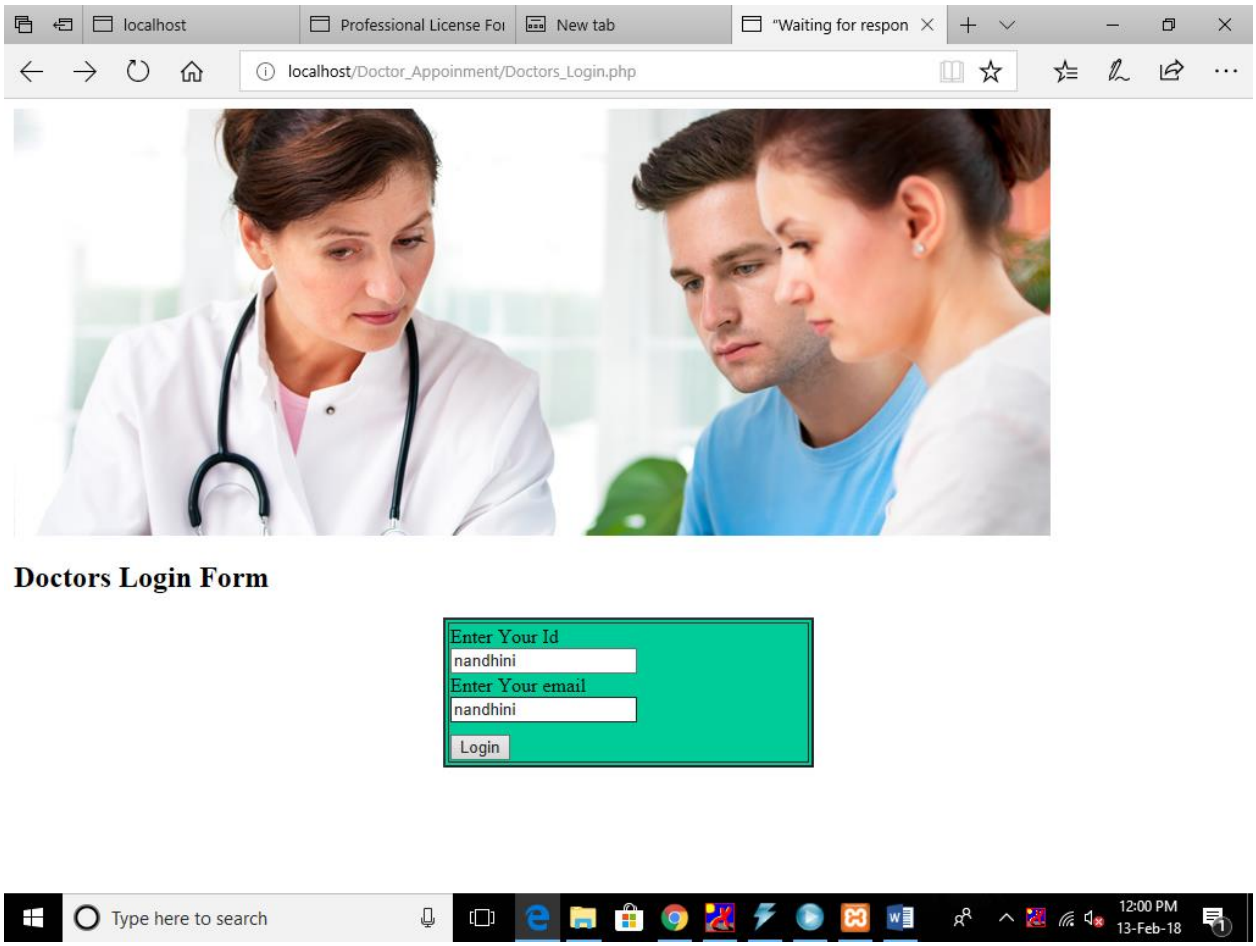


D.SAMPLE OUTPUT

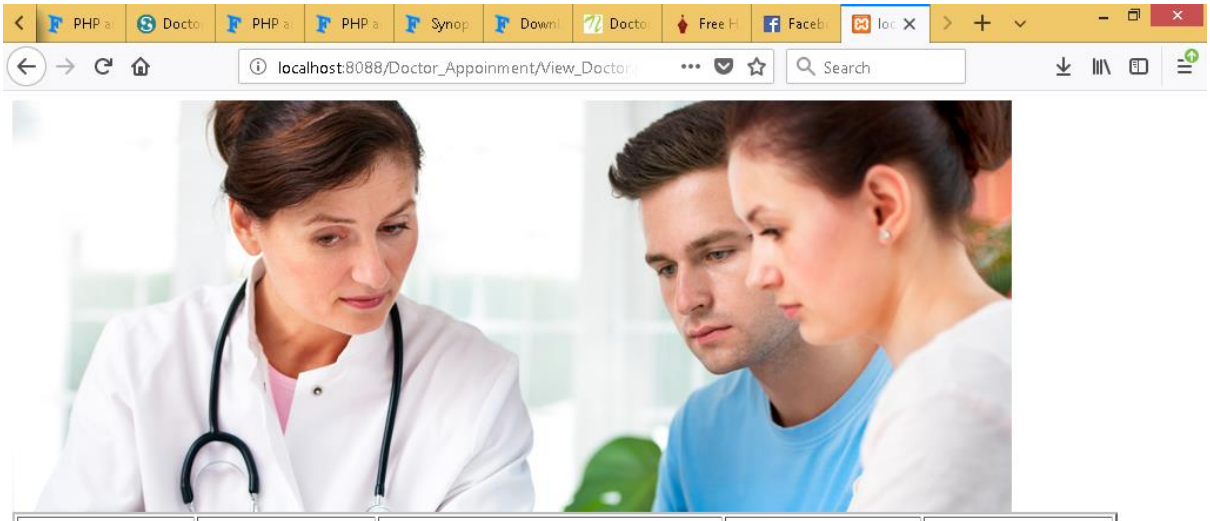
Enter registration form



Enter login form



View Doctor



Add Doctor	View Doctor	View Customer	View Appoinment	Logout

View Doctor

DID	Name	Address	Mobile	Category
107	hari	cbe	8754060427	Heart
1000	senthil	coimbatore	8754060427	General Physician
1002	kumar	coimbatore	9787332528	General Physician