"Smart Appointment Booking System"

A PROJECT REPORT SUBMITTED TO

THE NATIONAL INSTITUTE OF ENGINEERING, MYSURU

(An Autonomous Institution)

ESTD : 1946

In partial fulfillment of the requirements for Database Laboratory, fifth semester

Bachelor of Engineering

in

Computer Science & Engineering

Submitted by

Abhinav Gyan (4NI15CS003)

Aditya Sinha (4NI15CS006)

Ajeeta Asthana (4NICS008)

Akanksha Agrawal (4NI15CS009)

Under the Guidance of

Mr. G. Ramesh	**Smt. P. Nandini**
Assistant Professor	Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THE NATIONAL INSTITUTE OF ENGINEERING

Mysore-570 008

2017-2018

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
THE NATIONAL INSTITUTE OF ENGINEERING

ESTD : 1946

CERTIFICATE

This is to certify that the project work entitled “**Smart Appointment Booking System**” is a work carried out by **Abhinav Gyan (4NI15CS003), Aditya Sinha (4NI15CS006), Ajeeta Asthana (4NI15CS008)** and **Akanksha Agrawal (4NI15CS009)** in partial fulfillment for the Database Laboratory project work, fifth semester, Computer Science & Engineering, The National Institute of Engineering **(**Autonomous Institution under Vishveshwaraya Technological University, Belagavi) during the academic year 2017-2018. It is certified that all corrections and suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved in partial fulfillment as per academic regulations of The National Institute of Engineering, Mysuru.

Signature of Internal Guides		**Professor & Head**
Mr. G. Ramesh	**Smt. P. Nandini**	**Dr. H. D. Phaneendra**
Assistant Professor	**Assistant Professor**	**Professor & Head**
Department of CS&E	**Department of CS&E**	**Department of CS&E**
NIE, Mysuru – 570 008	**NIE, Mysuru – 570 008**	**NIE, Mysuru – 570 008**

ACKNOWLEDGEMENT

On completion of our project, we experienced many things that will help us in the coming years. We are extremely grateful to all of them who encouraged us to successfully complete the project and give us a lifetime opportunity to showcase our hard work and skill.

We would like to extend our sincere gratitude to Dr. H.D. Phaneendra, our HOD for giving us an opportunity to work on a challenging project and complete it successfully.

We are extremely happy to say that we have successfully completed our project and we credit our success to our project guides Mr. G. Ramesh and Smt. P. Nandini. They have encouraged us every time we felt discouraged , their guidance, knowledge and experience was way beyond our expectation.

Thank you.

Abhinav Gyan
Aditya Sinha
Ajeeta Asthana
Akanksha Agrawal

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 BACK END

A Back-End Database is a database that is accessed by users indirectly through an external application rather than by application programming stored within the database itself or by low level manipulation of the data (example through MySQL commands).

A back-end database stores data but does not include end-user application elements such as stored queries, forms,macros or reports.

The term back-end database is not widely used among developers using larger or enterprise database systems. This is because enterprise database systems enforce the use of the client–server model and do not have the option to include the application programming within their databases.

The end used here is MySQL on PHP. MySQL is the world’s most widely used open source relational database management system (RDBMS) that runs a server providing a multi-user access to a number of databases. The SQL phrase stands for Structured Query Language.

1.2 DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

CHAPTER 1

The database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world and generating reports from the data.

A database is not generally portable across different DBMS, but different DBMS can inter operate by using standards such as MySQL and JDBC to allow a single application to work with more than one DBMS.

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing system, which is usually managed manually, faces many problems, such as difficulty in data storage and retrieval, data inconsistency, data insecurity, risk of data loss, wastage of time in maintaining paper work and many more. All the information about a disease or a doctor must be looked up first, and then the hospital or clinic providing the service must be contacted in order to gain further knowledge. Another drawback is that most of the bookings cannot be done over the phone due to identity issues. Also, whenever a patient goes to the traditional appointment booking system, most of the things take a lot of time , and sometimes the appointments are also delayed. Hence, the traditional appointment booking systems are not efficient. The requirement analysis discovered the need of having automation and computerization of the entire working of the system.

CHAPTER 2

2.2 PROPOSED SYSTEM

This software automatizes and manages the working of appointment booking management systems. The database keeps track of the various data required. The whole project is divided into the following modules:

1. **Doctor Details:** This module keeps track of the details of the doctors enlisted, their work experience, qualification, fee structure, phone number and the amount of slots available to get their appointments.
2. **Hospital details:** This module keeps track of the the hospitals availability and the specialist doctors available in each of the hospitals and clinics.
3. **Bookings:** This module keeps a record of the details of the patients and doctors who have registered for the booking system. The details usually consists of the personal details of the registered user.
4. **Login:** This module deals with user logins, authentication and validates to prevent MySQL injection. Proper authorization will be done to take care of who is accessing the user.

CHAPTER 3

SYSTEM DESIGN

3.1. ENTITY DESCRIPTION AND ATTRIBUTE DETAILS

person:

It includes the person details like person id which is a primary key, person name, gender, date of birth, phone number and register date.

- personID
- firstName
- lastName
- gender
- dob
- phoneNumber
- registerDate

personLogin:

It includes patient login details like person id (foreign key), email (primary key), password, question, answer and last login. The password and the security answer are encrypted.

- personID
- email
- password
- question
- answer
- lastLogin

doctor:

It is a table similar to the person table along with some attributes added for the doctor profile like qualification, experience, fee, department and hospital or clinic. departmentID and buildingID are the foreign keys.

- doctorID
- firstName
- lastName
- gender
- dob
- phoneNumber
- registerDate
- qualification
- experience
- fee
- departmentID
- buildingID

doctorLogin:

It is a table which contains the login details for the user doctor. It is again similar to the table personLogin. doctorID is the foreign key.

- doctorID
- email
- password
- question
- answer

- lastLogin

department:

It consists of the details of the respective departments to which a registered doctor may belong to. It has a primary key departmentID.

- name
- departmentID

building:

It includes details of all the hospitals and clinics that may come under the patients interest of visit. buildingID is the primary key.

- buildingID
- name
- addressLine 1, 2, 3
- city
- pin
- longitude
- latitude

booking:

This includes booking details of a person who wants to book an appointment. bookingID is the primary key. personID and doctorID are the foreign key.

- bookingID
- bookingDate
- appointmentDate
- personID

- doctorID
- slot
- bookingStatus

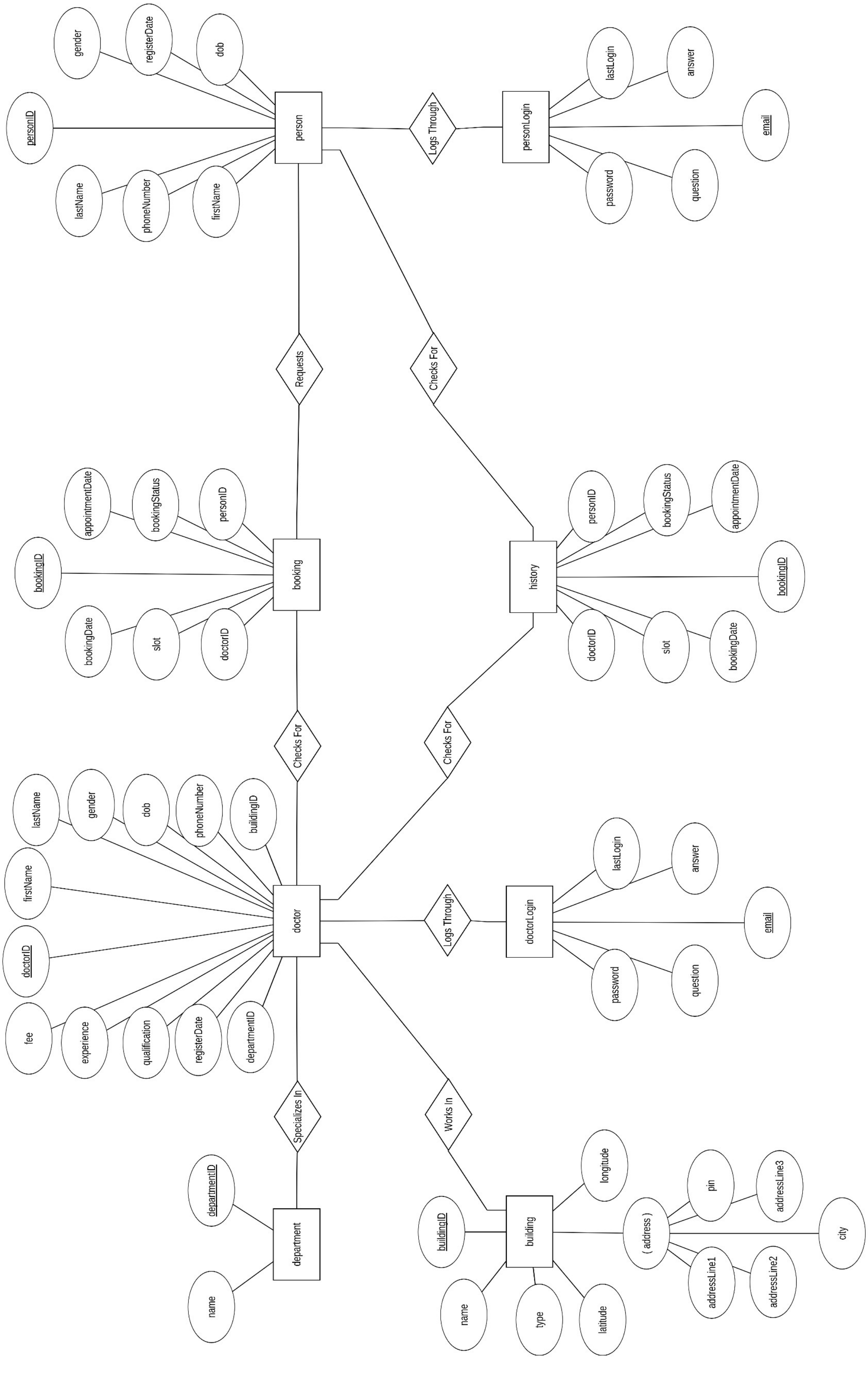
history:

This table consists of the details of past bookings that the patient or the doctor might have had. bookingID is the primary key. personID and doctorID are the foreign key.

- bookingID
- bookingDate
- appointmentDate
- personID
- doctorID
- slot
- bookingStatus

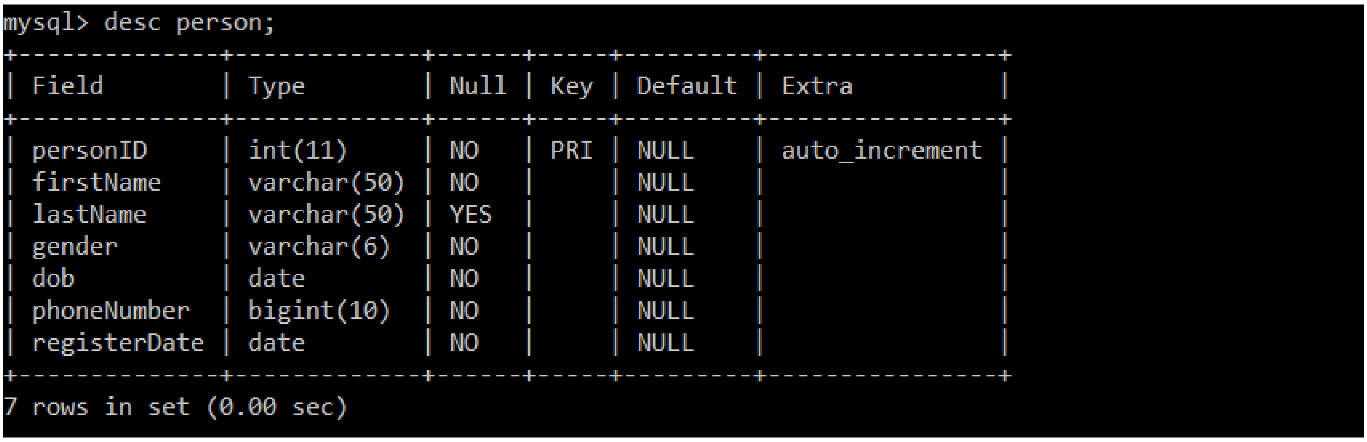
CHAPTER 3

3.2. ER DIAGRAM

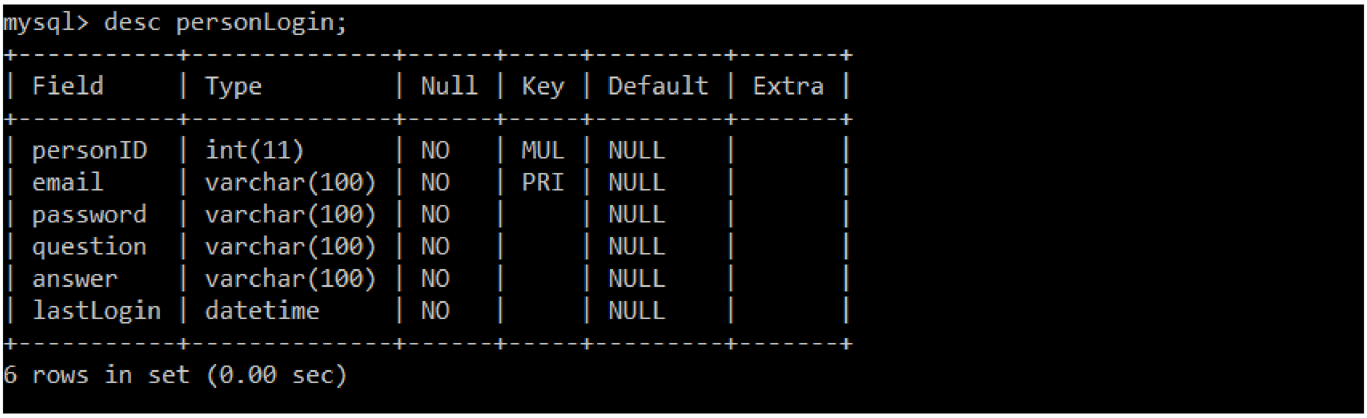


CHAPTER 3

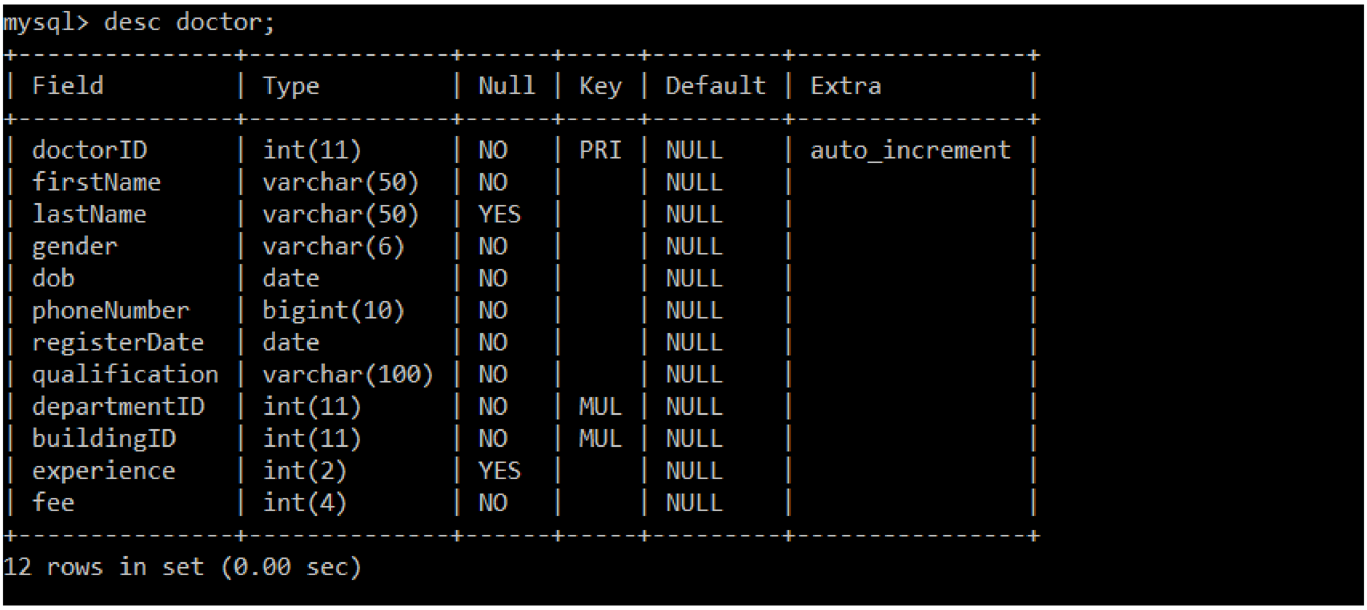
3.3 SCREEN SHOTS OF DESCRIPTION TABLES



```
mysql> desc person;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| personID     | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstName    | varchar(50) | NO   |     | NULL    |                |
| lastName     | varchar(50) | YES  |     | NULL    |                |
| gender       | varchar(6)  | NO   |     | NULL    |                |
| dob          | date        | NO   |     | NULL    |                |
| phoneNumber  | bigint(10)  | NO   |     | NULL    |                |
| registerDate | date        | NO   |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

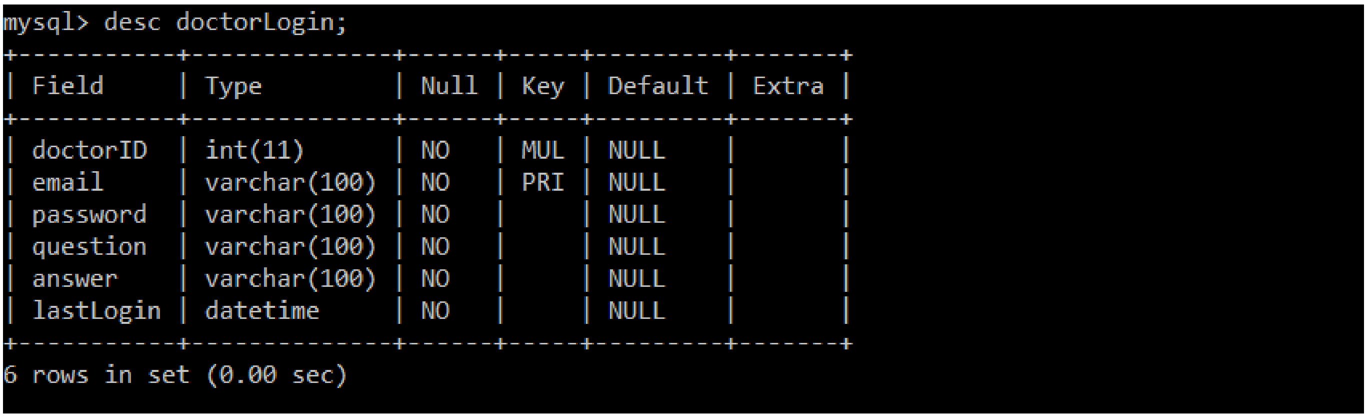


```
mysql> desc personLogin;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| personID  | int(11)      | NO   | MUL | NULL    |       |
| email     | varchar(100) | NO   | PRI | NULL    |       |
| password  | varchar(100) | NO   |     | NULL    |       |
| question  | varchar(100) | NO   |     | NULL    |       |
| answer    | varchar(100) | NO   |     | NULL    |       |
| lastLogin | datetime     | NO   |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

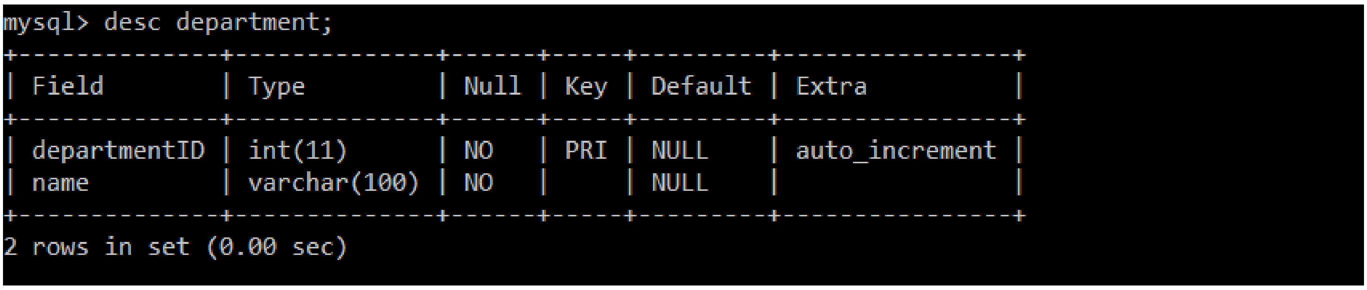


```
mysql> desc doctor;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| doctorID      | int(11)      | NO   | PRI | NULL    | auto_increment |
| firstName     | varchar(50)  | NO   |     | NULL    |                |
| lastName      | varchar(50)  | YES  |     | NULL    |                |
| gender        | varchar(6)   | NO   |     | NULL    |                |
| dob           | date         | NO   |     | NULL    |                |
| phoneNumber   | bigint(10)   | NO   |     | NULL    |                |
| registerDate  | date         | NO   |     | NULL    |                |
| qualification | varchar(100) | NO   |     | NULL    |                |
| departmentID  | int(11)      | NO   | MUL | NULL    |                |
| buildingID    | int(11)      | NO   | MUL | NULL    |                |
| experience    | int(2)       | YES  |     | NULL    |                |
| fee           | int(4)       | NO   |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
12 rows in set (0.00 sec)
```

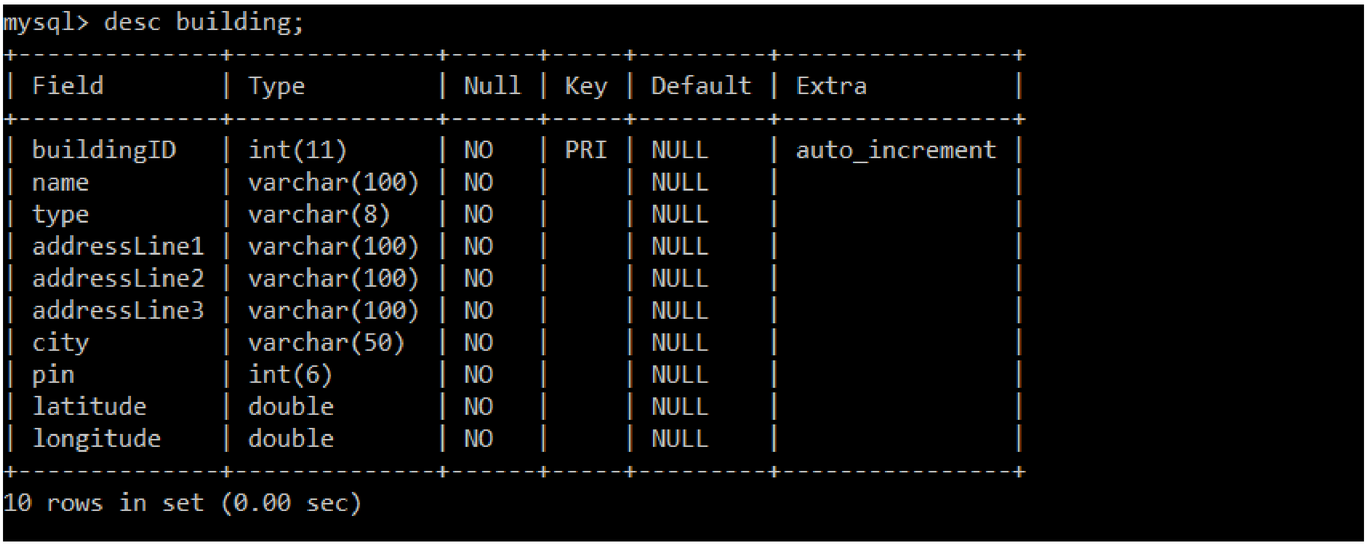
CHAPTER 3



```
mysql> desc doctorLogin;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| doctorID  | int(11)      | NO   | MUL | NULL    |       |
| email     | varchar(100) | NO   | PRI | NULL    |       |
| password  | varchar(100) | NO   |     | NULL    |       |
| question  | varchar(100) | NO   |     | NULL    |       |
| answer    | varchar(100) | NO   |     | NULL    |       |
| lastLogin | datetime     | NO   |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

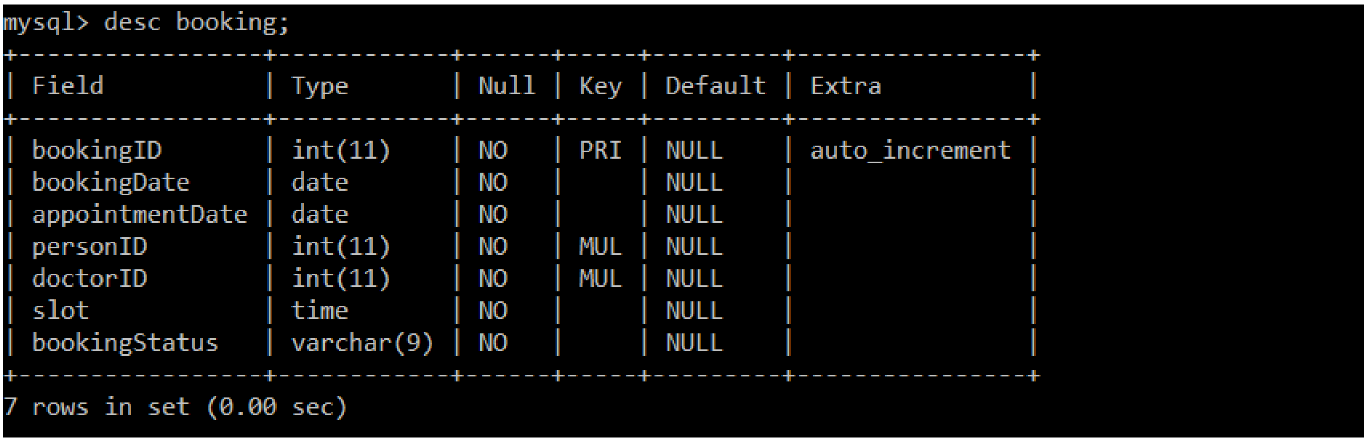


```
mysql> desc department;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| departmentID | int(11)      | NO   | PRI | NULL    | auto_increment |
| name         | varchar(100) | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

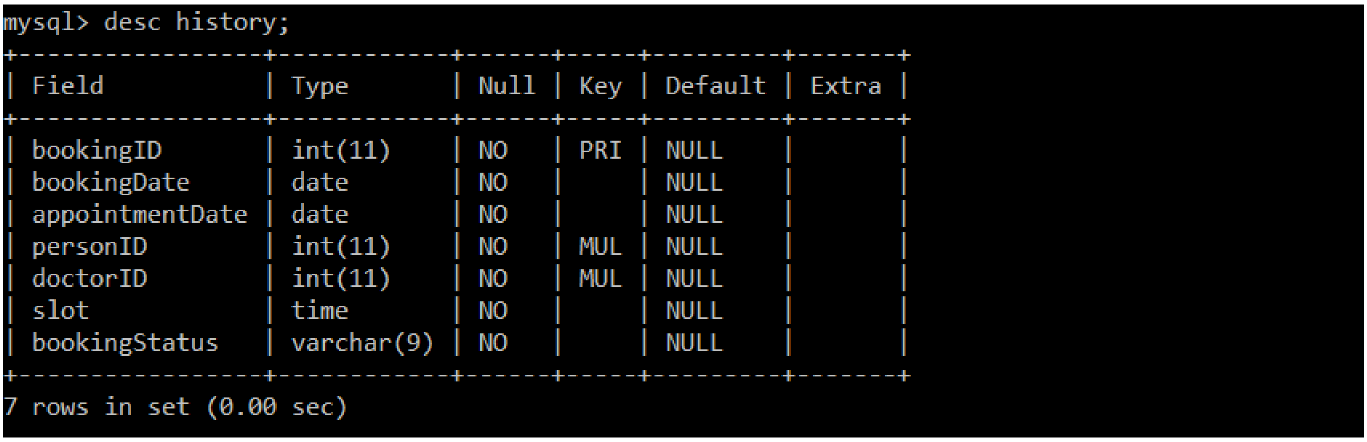


```
mysql> desc building;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| buildingID   | int(11)      | NO   | PRI | NULL    | auto_increment |
| name         | varchar(100) | NO   |     | NULL    |                |
| type         | varchar(8)   | NO   |     | NULL    |                |
| addressLine1 | varchar(100) | NO   |     | NULL    |                |
| addressLine2 | varchar(100) | NO   |     | NULL    |                |
| addressLine3 | varchar(100) | NO   |     | NULL    |                |
| city         | varchar(50)  | NO   |     | NULL    |                |
| pin          | int(6)       | NO   |     | NULL    |                |
| latitude     | double       | NO   |     | NULL    |                |
| longitude    | double       | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
10 rows in set (0.00 sec)
```

CHAPTER 3



```
mysql> desc booking;
+-----------------+------------+------+-----+---------+----------------+
| Field           | Type       | Null | Key | Default | Extra          |
+-----------------+------------+------+-----+---------+----------------+
| bookingID       | int(11)    | NO   | PRI | NULL    | auto_increment |
| bookingDate     | date       | NO   |     | NULL    |                |
| appointmentDate | date       | NO   |     | NULL    |                |
| personID        | int(11)    | NO   | MUL | NULL    |                |
| doctorID        | int(11)    | NO   | MUL | NULL    |                |
| slot            | time       | NO   |     | NULL    |                |
| bookingStatus   | varchar(9) | NO   |     | NULL    |                |
+-----------------+------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```



```
mysql> desc history;
+-----------------+------------+------+-----+---------+-------+
| Field           | Type       | Null | Key | Default | Extra |
+-----------------+------------+------+-----+---------+-------+
| bookingID       | int(11)    | NO   | PRI | NULL    |       |
| bookingDate     | date       | NO   |     | NULL    |       |
| appointmentDate | date       | NO   |     | NULL    |       |
| personID        | int(11)    | NO   | MUL | NULL    |       |
| doctorID        | int(11)    | NO   | MUL | NULL    |       |
| slot            | time       | NO   |     | NULL    |       |
| bookingStatus   | varchar(9) | NO   |     | NULL    |       |
+-----------------+------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

CHAPTER 4

HARDWARE AND SOFTWARE REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS

Front end:	HTML, CSS, Bootstrap, JavaScript
Back end:	MySQL
Scripting Language:	PHP
Environment:	XamppServer, Nginx

4.2 HARDWARE REQUIREMENTS

OS:	Any desktop version
Processor:	1.2 GHz CPU
Memory:	1 GB RAM
Disk:	2 GB HDD Space
Display:	1024 x 768 or higher
Network:	512 Kbps or faster

A web browser with cookies and JavaScript enabled.

CHAPTER 5

SYSTEM IMPLEMENTATION

The database software used here is MySQL.
Doctor, doctorLogin, person, personLogin, booking, history, department, building are used as the tables in our project.

Doctor and doctorLogin tables are used store information of doctors. Person and personLogin are used to store information of patients. Booking and History tables are used to store information of the future bookings and past bookings respectively. Department table stores department name of a particular doctor and building table stores the name of the hospital/clinic in which a particular doctor is working.

All the records are maintained/edited by writing queries in MySQL through PHP.

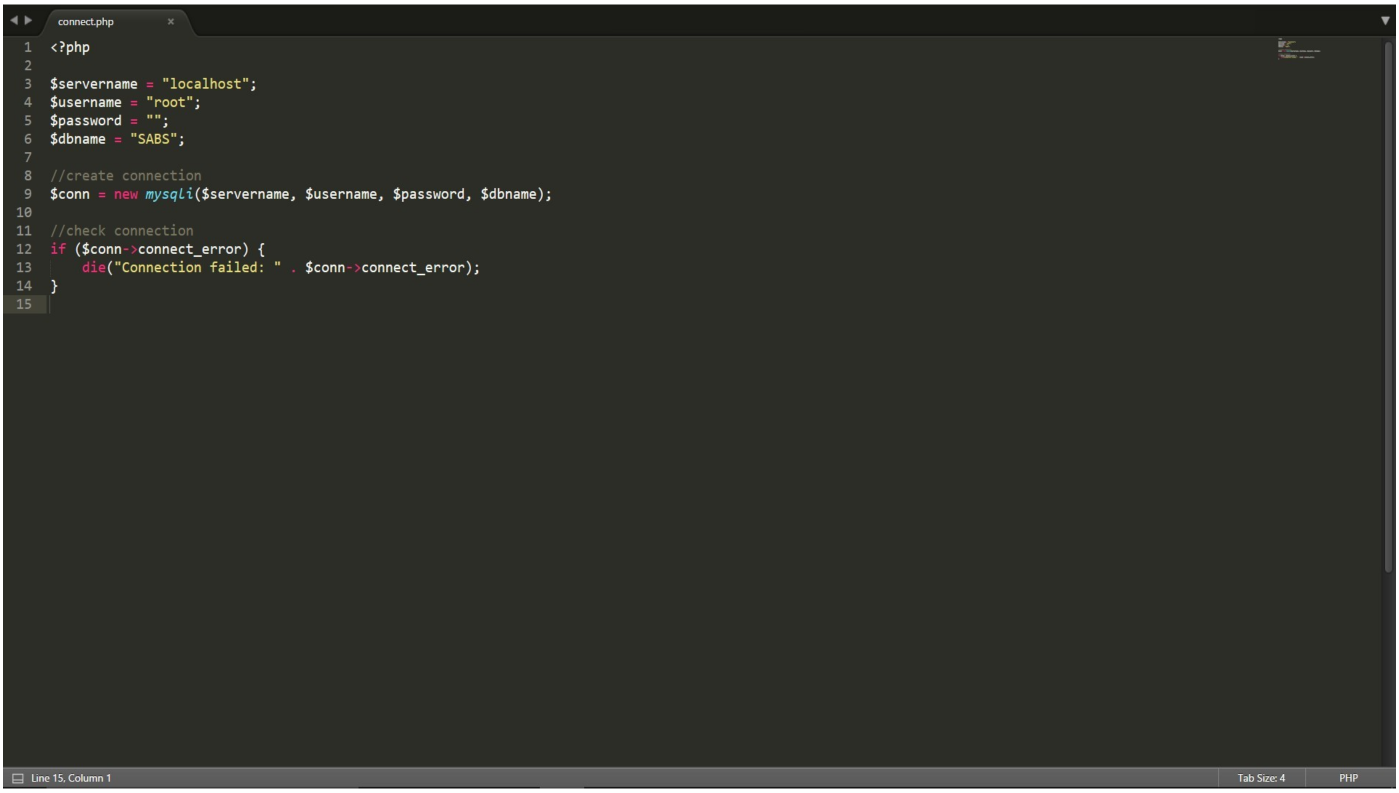
Patient should first sign up and login to book a doctor’s slot. All booking information will be present in the profile of the patient.

Doctor should login to see all the bookings and cancel it if necessary from their profile page.

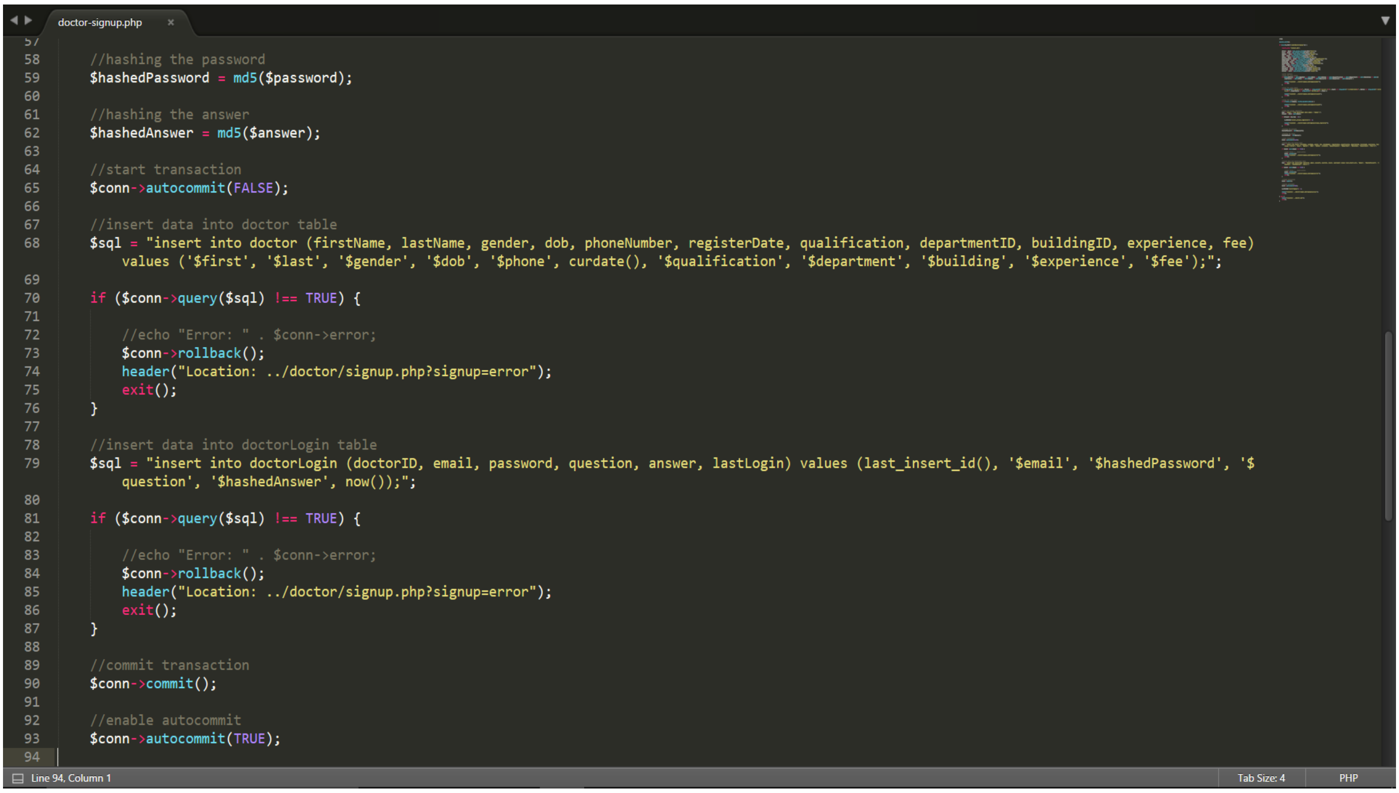
Algorithm:
MySQL uses a B+ tree data structure for primary and secondary indexes in MyISAM and InnoDB storage engines.

CHAPTER 6

CODE SNIPPET



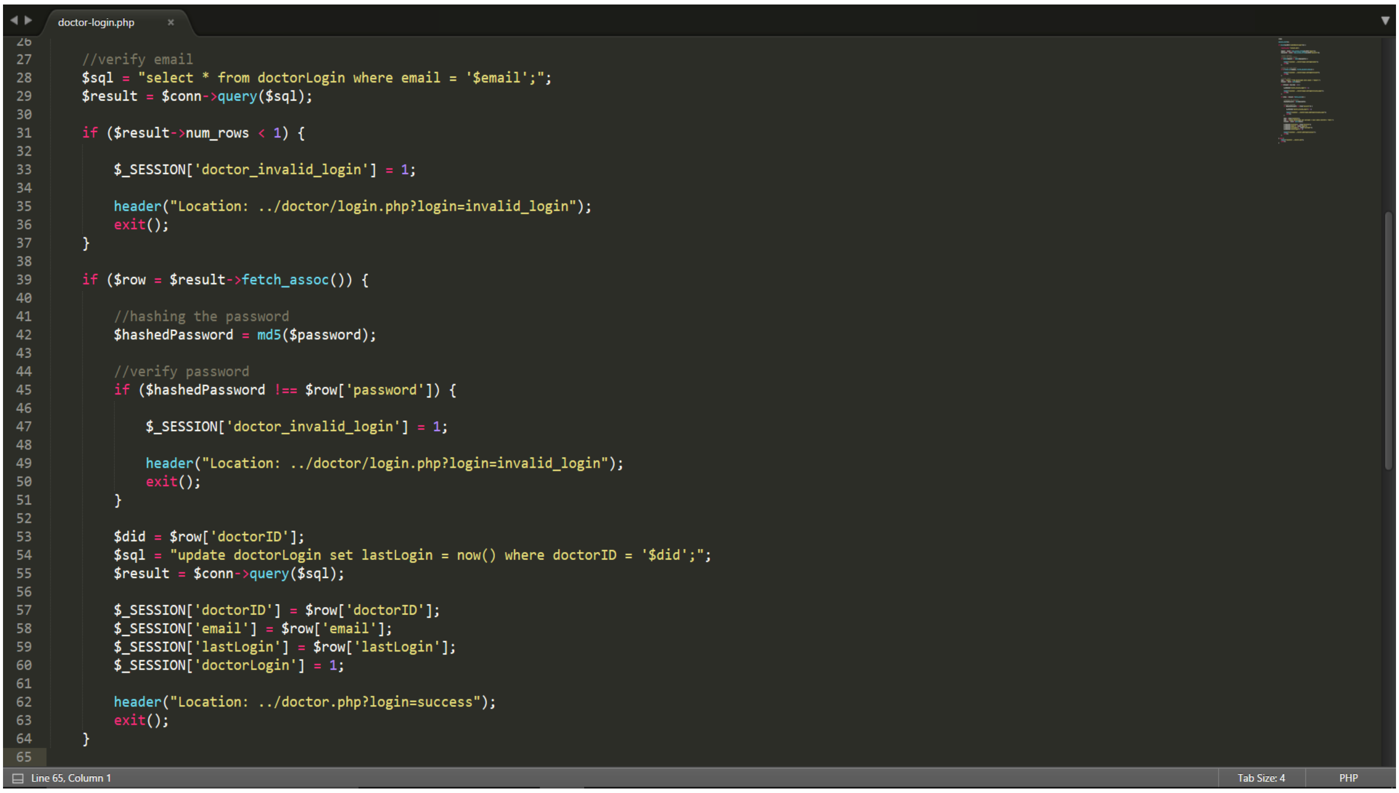
```php
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "SABS";

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

//check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```



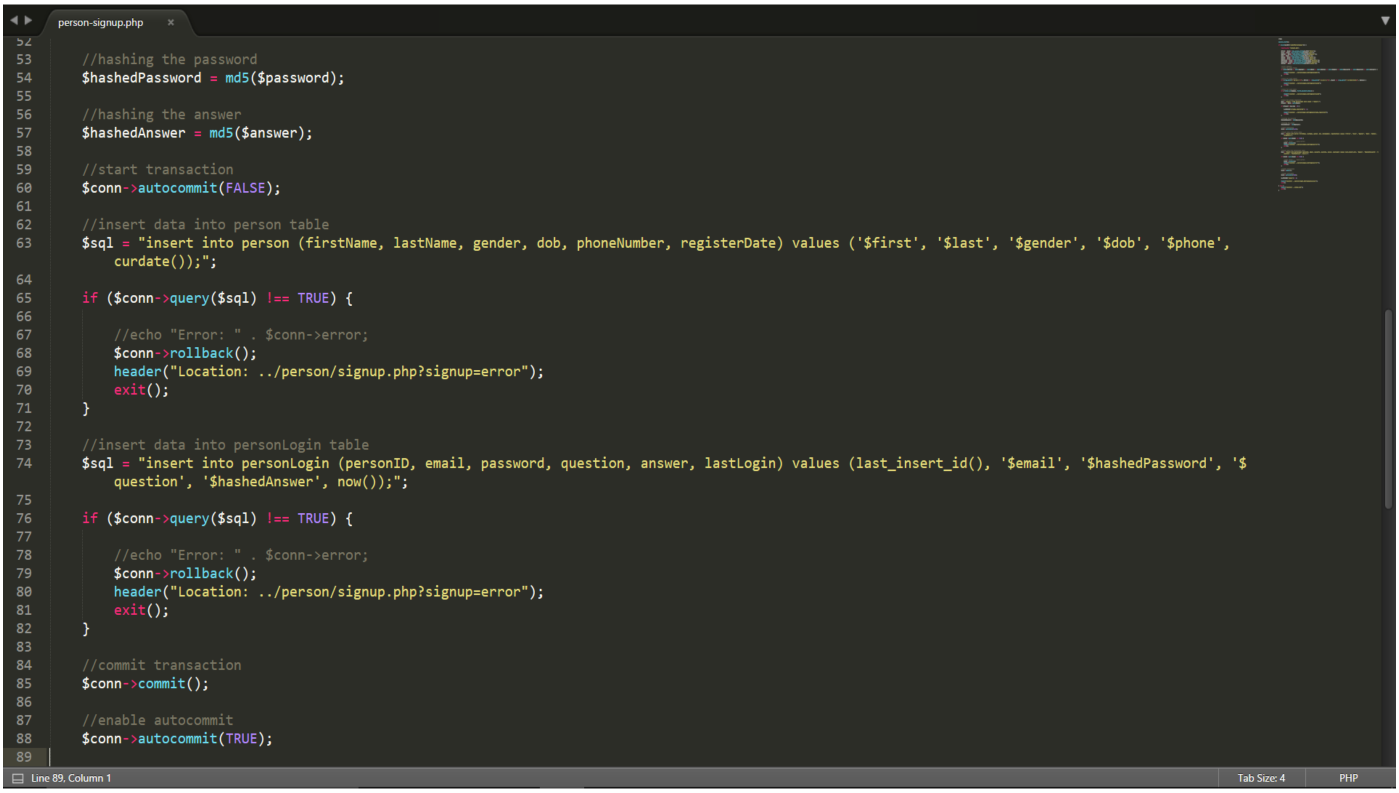
```php
    //hashing the password
    $hashedPassword = md5($password);

    //hashing the answer
    $hashedAnswer = md5($answer);

    //start transaction
    $conn->autocommit(FALSE);

    //insert data into doctor table
    $sql = "insert into doctor (firstName, lastName, gender, dob, phoneNumber, registerDate, qualification, departmentID, buildingID, experience, fee)
        values ('$first', '$last', '$gender', '$dob', '$phone', curdate(), '$qualification', '$department', '$building', '$experience', '$fee');";

    if ($conn->query($sql) !== TRUE) {

        //echo "Error: " . $conn->error;
        $conn->rollback();
        header("Location: ../doctor/signup.php?signup=error");
        exit();
    }

    //insert data into doctorLogin table
    $sql = "insert into doctorLogin (doctorID, email, password, question, answer, lastLogin) values (last_insert_id(), '$email', '$hashedPassword', '$
        question', '$hashedAnswer', now());";

    if ($conn->query($sql) !== TRUE) {

        //echo "Error: " . $conn->error;
        $conn->rollback();
        header("Location: ../doctor/signup.php?signup=error");
        exit();
    }

    //commit transaction
    $conn->commit();

    //enable autocommit
    $conn->autocommit(TRUE);

```

CHAPTER 6

CODE SNIPPET

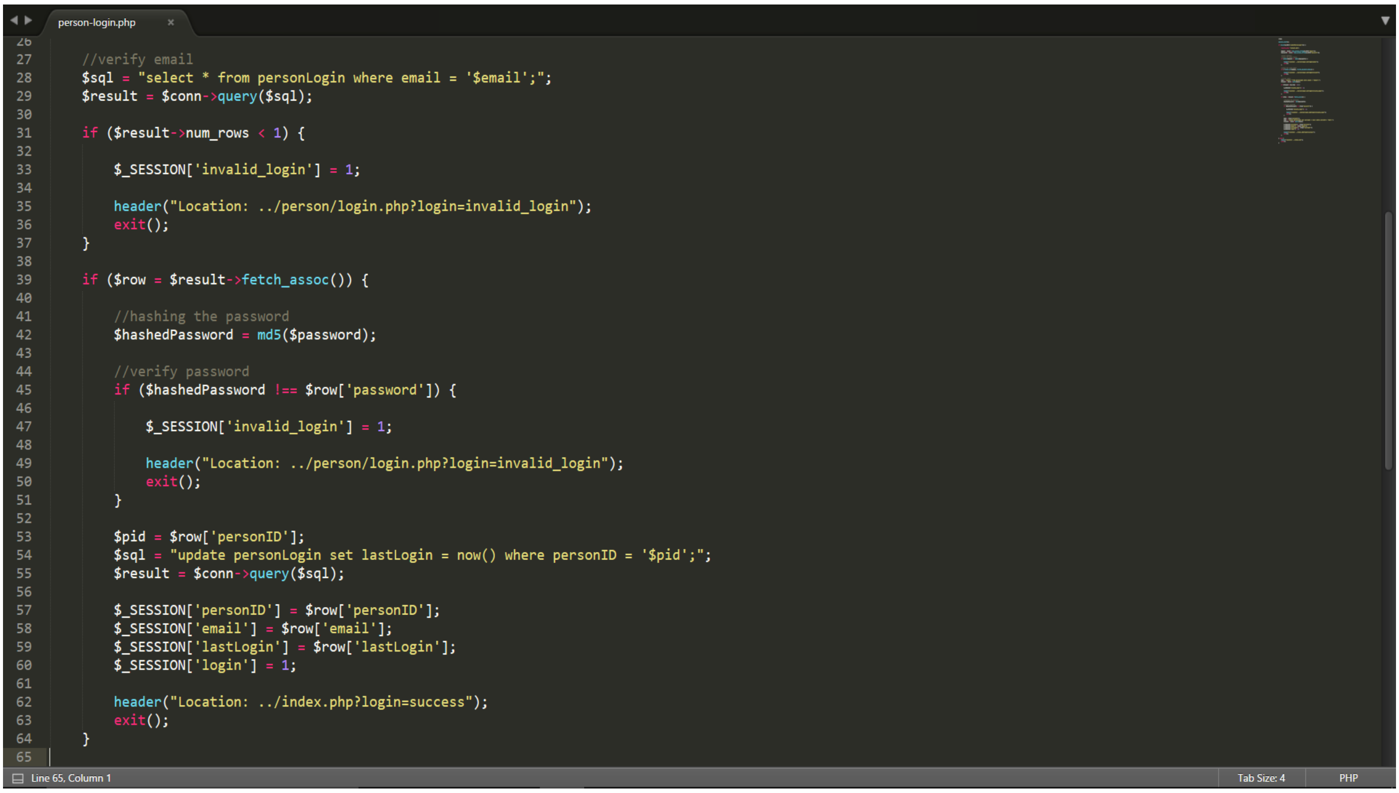


```php
//verify email
$sql = "select * from doctorLogin where email = '$email';";
$result = $conn->query($sql);

if ($result->num_rows < 1) {

    $_SESSION['doctor_invalid_login'] = 1;

    header("Location: ../doctor/login.php?login=invalid_login");
    exit();
}

if ($row = $result->fetch_assoc()) {

    //hashing the password
    $hashedPassword = md5($password);

    //verify password
    if ($hashedPassword !== $row['password']) {

        $_SESSION['doctor_invalid_login'] = 1;

        header("Location: ../doctor/login.php?login=invalid_login");
        exit();
    }

    $did = $row['doctorID'];
    $sql = "update doctorLogin set lastLogin = now() where doctorID = '$did';";
    $result = $conn->query($sql);

    $_SESSION['doctorID'] = $row['doctorID'];
    $_SESSION['email'] = $row['email'];
    $_SESSION['lastLogin'] = $row['lastLogin'];
    $_SESSION['doctorLogin'] = 1;

    header("Location: ../doctor.php?login=success");
    exit();
}

```

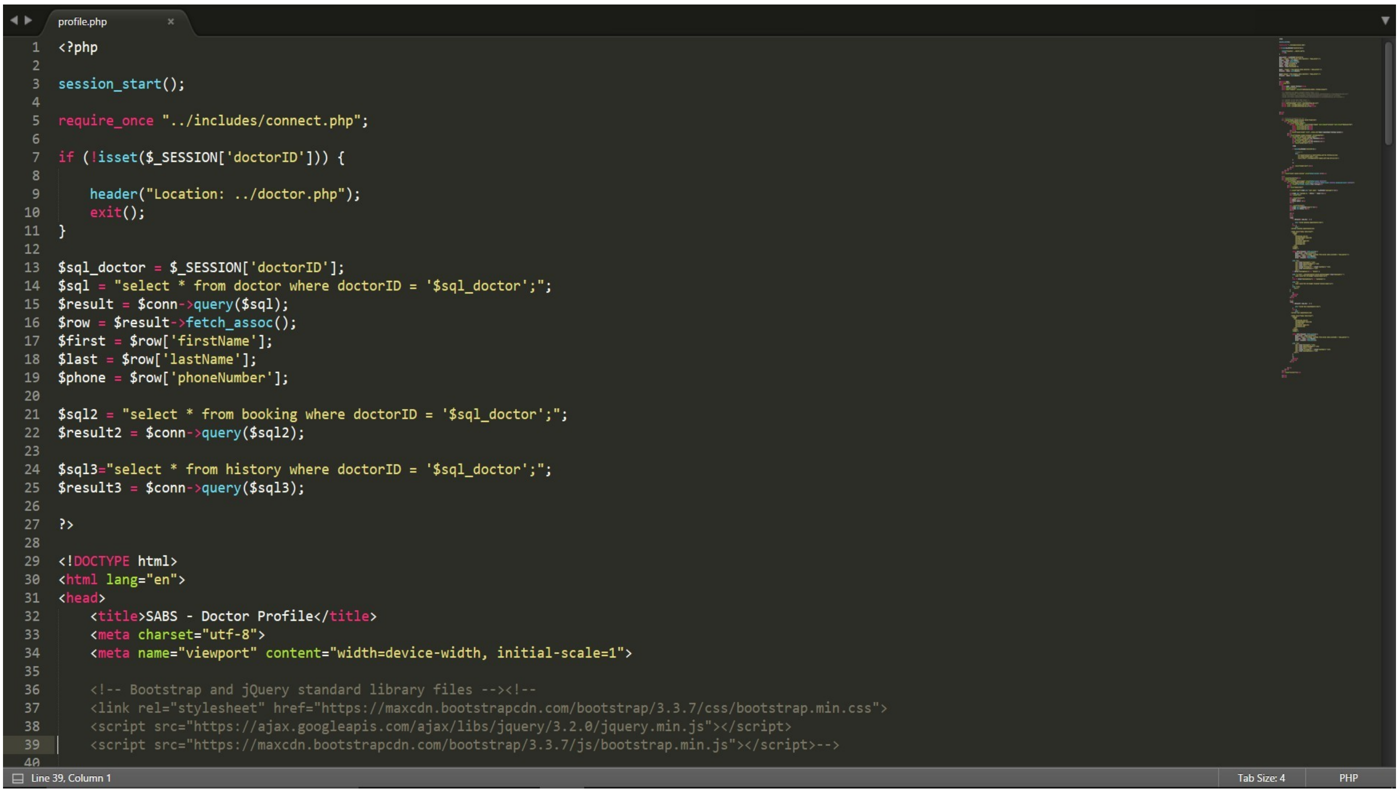


```php
//hashing the password
$hashedPassword = md5($password);

//hashing the answer
$hashedAnswer = md5($answer);

//start transaction
$conn->autocommit(FALSE);

//insert data into person table
$sql = "insert into person (firstName, lastName, gender, dob, phoneNumber, registerDate) values ('$first', '$last', '$gender', '$dob', '$phone',
    curdate());";

if ($conn->query($sql) !== TRUE) {

    //echo "Error: " . $conn->error;
    $conn->rollback();
    header("Location: ../person/signup.php?signup=error");
    exit();
}

//insert data into personLogin table
$sql = "insert into personLogin (personID, email, password, question, answer, lastLogin) values (last_insert_id(), '$email', '$hashedPassword', '$
    question', '$hashedAnswer', now());";

if ($conn->query($sql) !== TRUE) {

    //echo "Error: " . $conn->error;
    $conn->rollback();
    header("Location: ../person/signup.php?signup=error");
    exit();
}

//commit transaction
$conn->commit();

//enable autocommit
$conn->autocommit(TRUE);

```

CODE SNIPPET



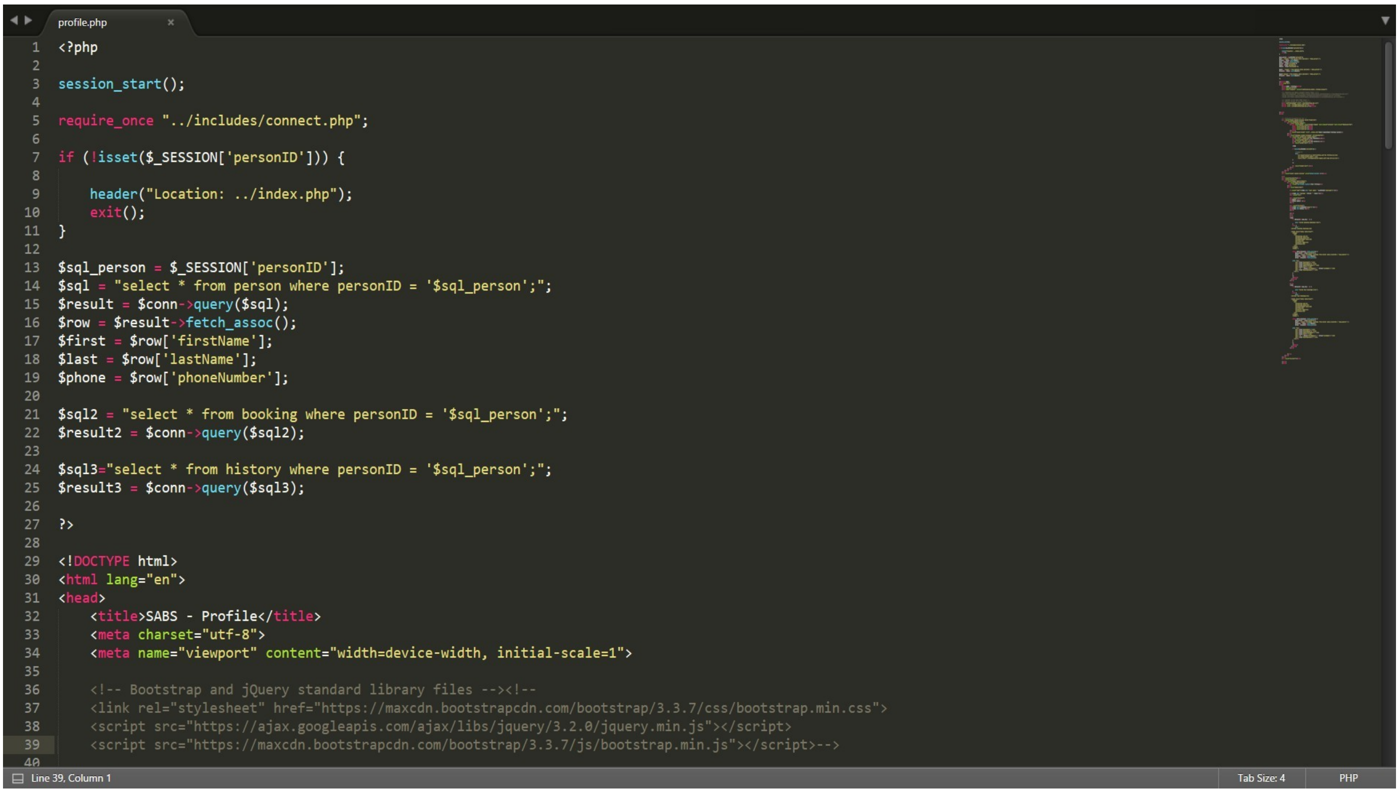
```php
    //verify email
    $sql = "select * from personLogin where email = '$email';";
    $result = $conn->query($sql);

    if ($result->num_rows < 1) {

        $_SESSION['invalid_login'] = 1;

        header("Location: ../person/login.php?login=invalid_login");
        exit();
    }

    if ($row = $result->fetch_assoc()) {

        //hashing the password
        $hashedPassword = md5($password);

        //verify password
        if ($hashedPassword !== $row['password']) {

            $_SESSION['invalid_login'] = 1;

            header("Location: ../person/login.php?login=invalid_login");
            exit();
        }

        $pid = $row['personID'];
        $sql = "update personLogin set lastLogin = now() where personID = '$pid';";
        $result = $conn->query($sql);

        $_SESSION['personID'] = $row['personID'];
        $_SESSION['email'] = $row['email'];
        $_SESSION['lastLogin'] = $row['lastLogin'];
        $_SESSION['login'] = 1;

        header("Location: ../index.php?login=success");
        exit();
    }

```



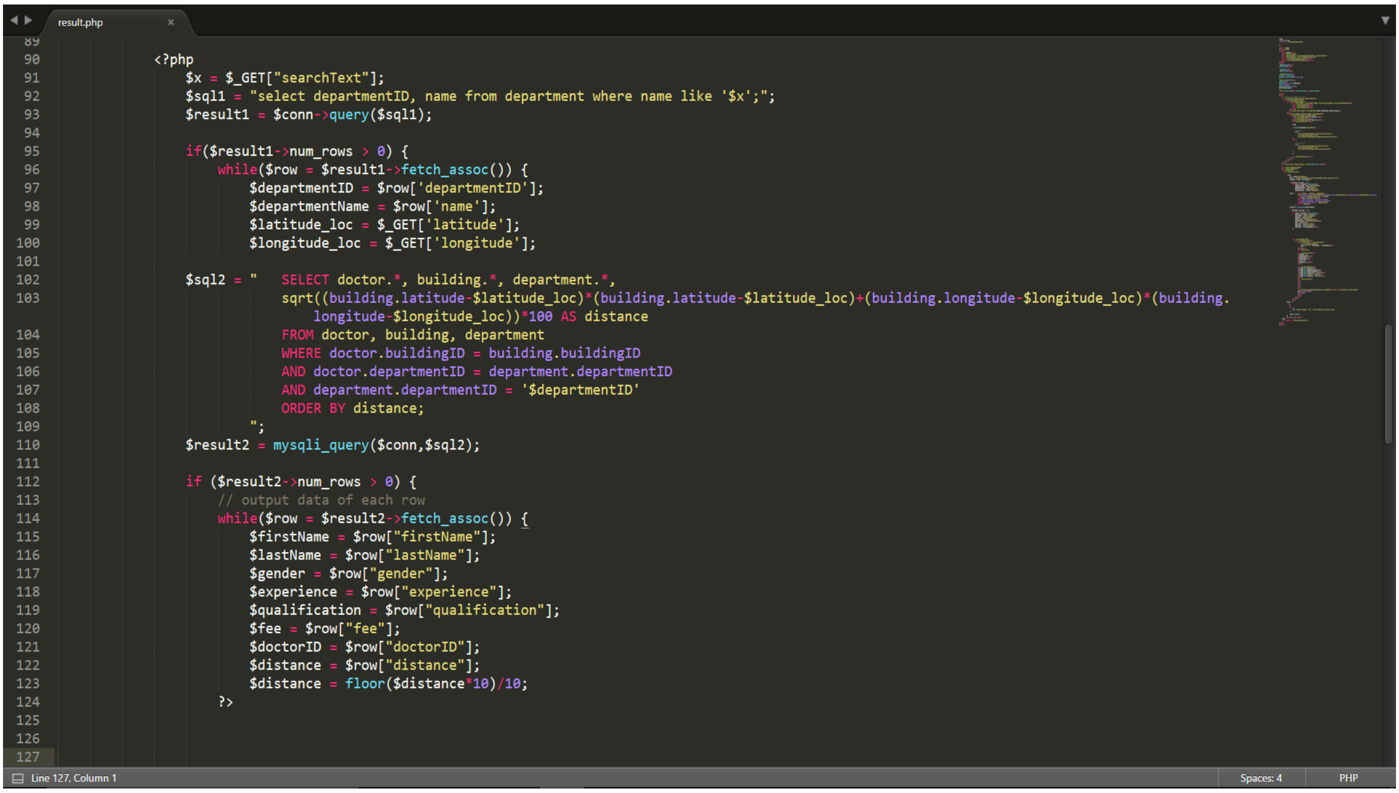
```php
<?php

session_start();

require_once "../includes/connect.php";

if (!isset($_SESSION['doctorID'])) {

    header("Location: ../doctor.php");
    exit();
}

$sql_doctor = $_SESSION['doctorID'];
$sql = "select * from doctor where doctorID = '$sql_doctor';";
$result = $conn->query($sql);
$row = $result->fetch_assoc();
$first = $row['firstName'];
$last = $row['lastName'];
$phone = $row['phoneNumber'];

$sql2 = "select * from booking where doctorID = '$sql_doctor';";
$result2 = $conn->query($sql2);

$sql3="select * from history where doctorID = '$sql_doctor';";
$result3 = $conn->query($sql3);

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <title>SABS - Doctor Profile</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap and jQuery standard library files --><!--
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>-->

```

CHAPTER 6

CODE SNIPPET



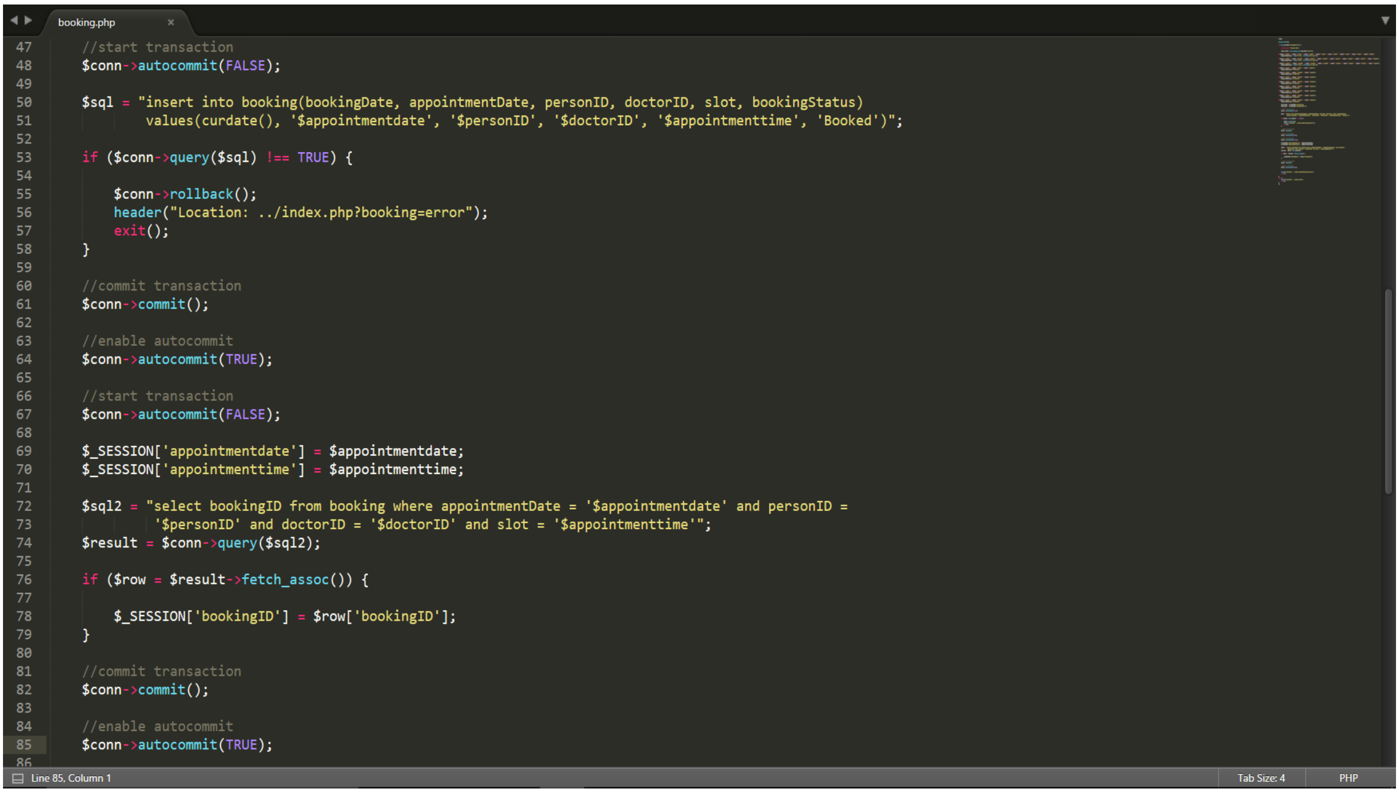
```php
<?php

session_start();

require_once "../includes/connect.php";

if (!isset($_SESSION['personID'])) {

    header("Location: ../index.php");
    exit();
}

$sql_person = $_SESSION['personID'];
$sql = "select * from person where personID = '$sql_person';";
$result = $conn->query($sql);
$row = $result->fetch_assoc();
$first = $row['firstName'];
$last = $row['lastName'];
$phone = $row['phoneNumber'];

$sql2 = "select * from booking where personID = '$sql_person';";
$result2 = $conn->query($sql2);

$sql3="select * from history where personID = '$sql_person';";
$result3 = $conn->query($sql3);

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <title>SABS - Profile</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap and jQuery standard library files --><!--
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>-->
```



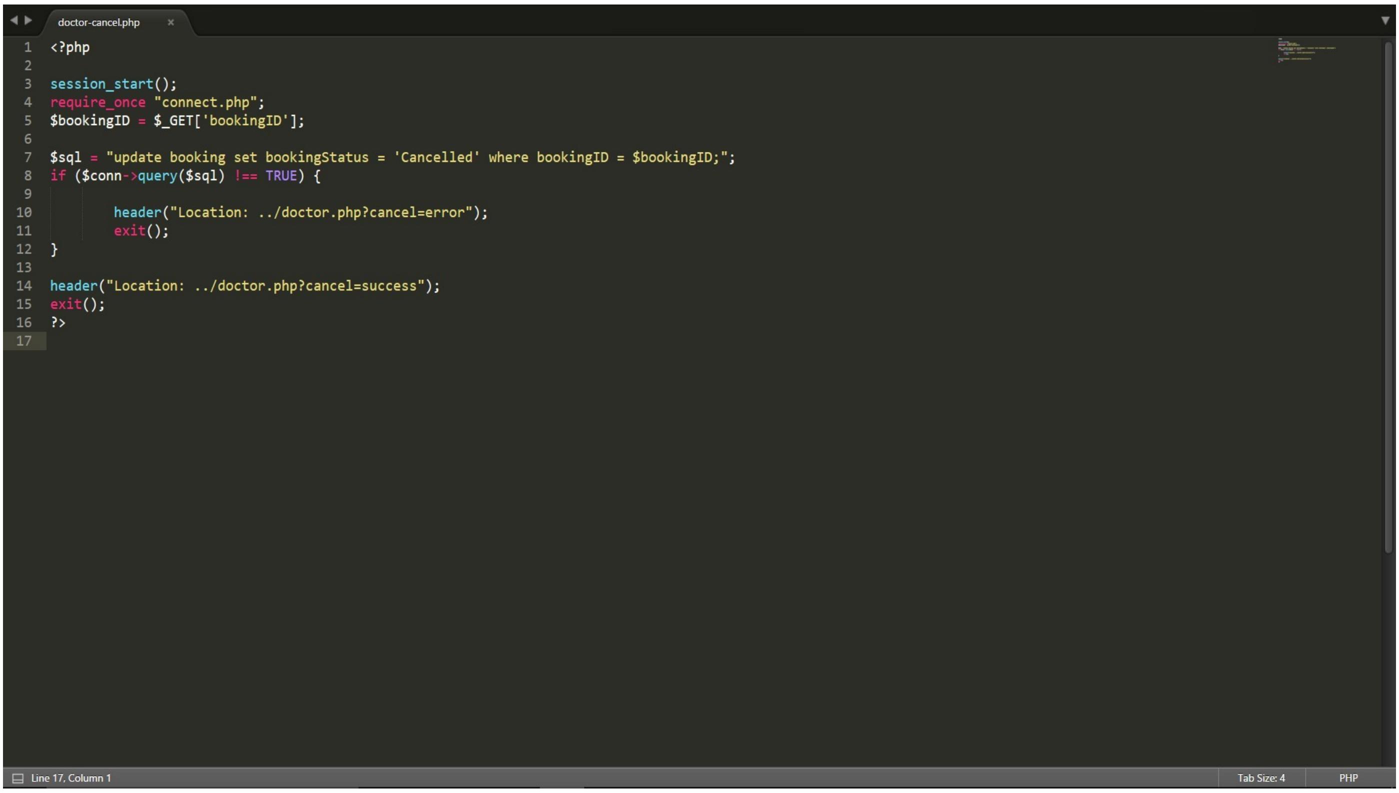
```php
            <?php
                $x = $_GET["searchText"];
                $sql1 = "select departmentID, name from department where name like '$x';";
                $result1 = $conn->query($sql1);

                if($result1->num_rows > 0) {
                    while($row = $result1->fetch_assoc()) {
                        $departmentID = $row['departmentID'];
                        $departmentName = $row['name'];
                        $latitude_loc = $_GET['latitude'];
                        $longitude_loc = $_GET['longitude'];

                $sql2 = "   SELECT doctor.*, building.*, department.*,
                            sqrt((building.latitude-$latitude_loc)*(building.latitude-$latitude_loc)+(building.longitude-$longitude_loc)*(building.
                                longitude-$longitude_loc))*100 AS distance
                            FROM doctor, building, department
                            WHERE doctor.buildingID = building.buildingID
                            AND doctor.departmentID = department.departmentID
                            AND department.departmentID = '$departmentID'
                            ORDER BY distance;
                        ";
                $result2 = mysqli_query($conn,$sql2);

                if ($result2->num_rows > 0) {
                    // output data of each row
                    while($row = $result2->fetch_assoc()) {
                        $firstName = $row["firstName"];
                        $lastName = $row["lastName"];
                        $gender = $row["gender"];
                        $experience = $row["experience"];
                        $qualification = $row["qualification"];
                        $fee = $row["fee"];
                        $doctorID = $row["doctorID"];
                        $distance = $row["distance"];
                        $distance = floor($distance*10)/10;
                    ?>
```

CHAPTER 6

CODE SNIPPET



```php
    //start transaction
    $conn->autocommit(FALSE);

    $sql = "insert into booking(bookingDate, appointmentDate, personID, doctorID, slot, bookingStatus)
            values(curdate(), '$appointmentdate', '$personID', '$doctorID', '$appointmenttime', 'Booked')";

    if ($conn->query($sql) !== TRUE) {

        $conn->rollback();
        header("Location: ../index.php?booking=error");
        exit();
    }

    //commit transaction
    $conn->commit();

    //enable autocommit
    $conn->autocommit(TRUE);

    //start transaction
    $conn->autocommit(FALSE);

    $_SESSION['appointmentdate'] = $appointmentdate;
    $_SESSION['appointmenttime'] = $appointmenttime;

    $sql2 = "select bookingID from booking where appointmentDate = '$appointmentdate' and personID =
             '$personID' and doctorID = '$doctorID' and slot = '$appointmenttime'";
    $result = $conn->query($sql2);

    if ($row = $result->fetch_assoc()) {

        $_SESSION['bookingID'] = $row['bookingID'];
    }

    //commit transaction
    $conn->commit();

    //enable autocommit
    $conn->autocommit(TRUE);
```



```php
<?php

session_start();
require_once "connect.php";
$bookingID = $_GET['bookingID'];

$sql = "update booking set bookingStatus = 'Cancelled' where bookingID = $bookingID;";
if ($conn->query($sql) !== TRUE) {

        header("Location: ../doctor.php?cancel=error");
        exit();
}

header("Location: ../doctor.php?cancel=success");
exit();
?>
```

CHAPTER 7

Testing

Testing is the set of activities that can be planned in advance and conducted systematically.

There are many strategies that can be used to test conventional software. Testing can be done once the entire software is complete. However, the results in buggy software are simply not effective.

Another approach would be to test the software on a daily basis, whenever any part of the software is constructed. The approach is effective as bugs are eliminated as the software is constructed and leads to more efficient software once all the modules are integrated. A testing strategy chosen falls between two extremes. It takes an incremental view of testing, starting with the testing of individual program units, moving to tests designed to facilitate the integration of the units, and ending with tests that cover the entire software. This project focuses on ‘Unit testing’. This testing mainly considers the above mentioned factors for each and every module implemented in the project.

Unit testing focuses on the smallest unit of the software design. Smallest unit include the module which are integrated to produce the final project. The unit testing focuses on the internal logic and data structures within the boundaries of the component. Test considerations can be the data structures, boundary conditions, independent paths, error handling paths, etc. Unit testing was done on verifying the email and password for accessing the database. The following results were obtained:

CHAPTER 7

Unit testing of each modules:

7.1 Test case for Login:

Test Cases	E-Mail Id	Password	Test Result
TC1	Correct Email Id	Correct password	Successful login, Main page is displayed
TC2	Correct Email Id	Incorrect password	Prompt saying that "Wrong Password!!!".
TC3	Incorrect Email Id	Correct password	Prompt saying that "Email id not yet registered".
TC4	Incorrect Email Id	Incorrect password	Prompt saying that "Email not yet registered ".

Fig 1: Test results for successful and unsuccessful login.

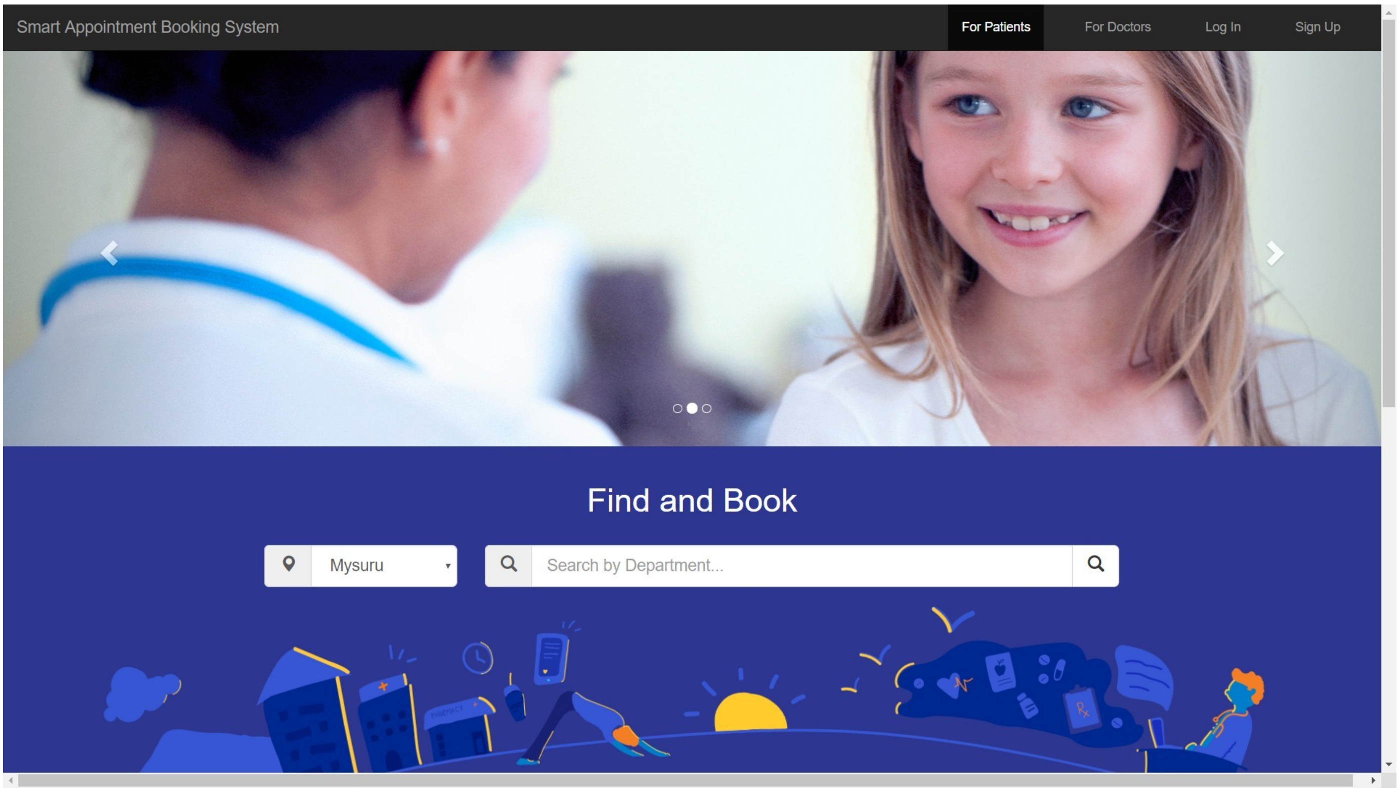
7.2 Test Cases for Registration :

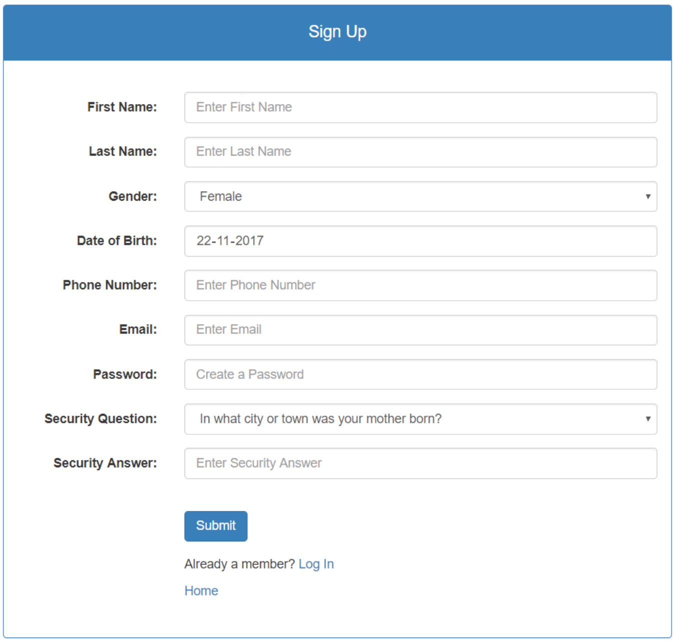
Test Cases	E-Mail Id	Test Result
TC1	Unregistered Email Id	Successful Registration, Login page is displayed
TC2	Registered Email Id	Prompt saying that "Email already exists!!!".

Fig 2: Test results for successful and unsuccessful registration.

CHAPTER 8

SCREEN SHOTS



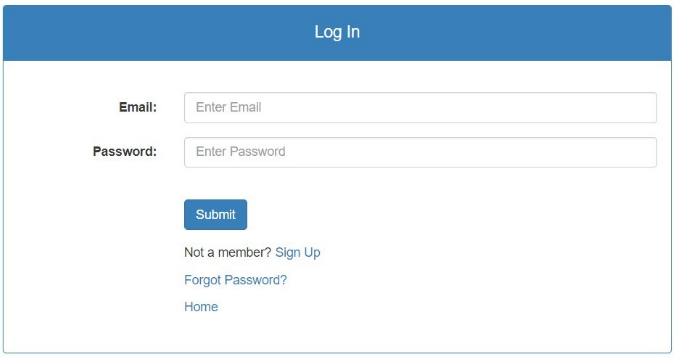


First Name:	Enter First Name
Last Name:	Enter Last Name
Gender:	Female
Date of Birth:	22-11-2017
Phone Number:	Enter Phone Number
Email:	Enter Email
Password:	Create a Password
Security Question:	In what city or town was your mother born?
Security Answer:	Enter Security Answer

Already a member? Log In

Home

CHAPTER 8



Log In

Email: Enter Email

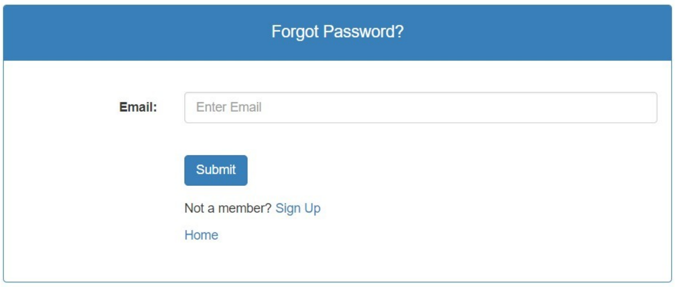
Password: Enter Password

Submit

Not a member? Sign Up

Forgot Password?

Home



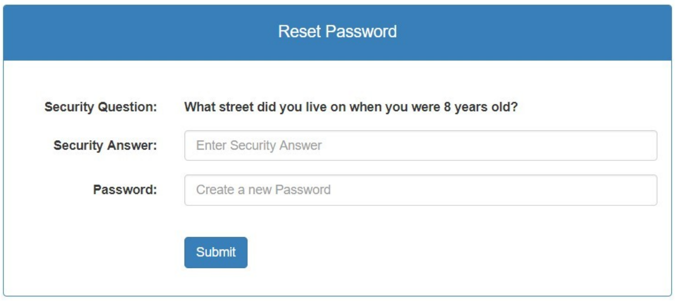
Forgot Password?

Email: Enter Email

Submit

Not a member? Sign Up

Home



Reset Password

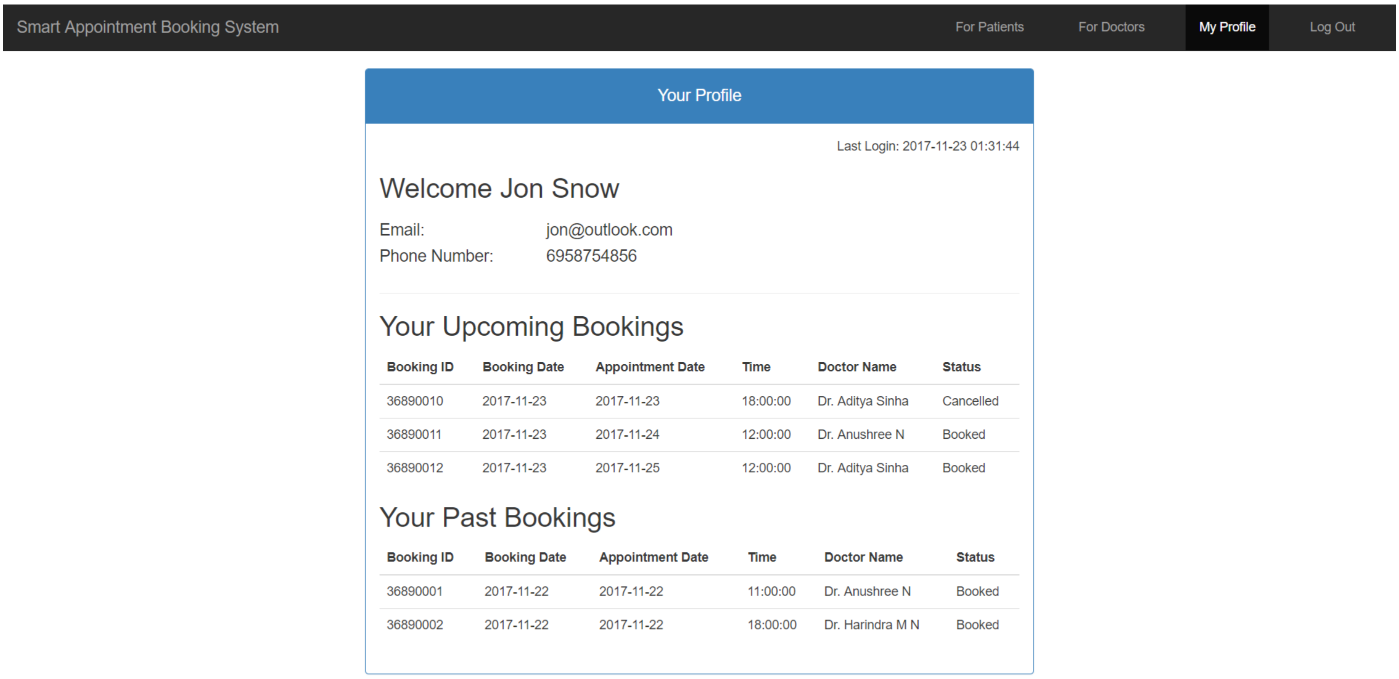
Security Question: What street did you live on when you were 8 years old?

Security Answer: Enter Security Answer

Password: Create a new Password

Submit

CHAPTER 8



Your Profile

Last Login: 2017-11-23 01:31:44

Welcome Jon Snow

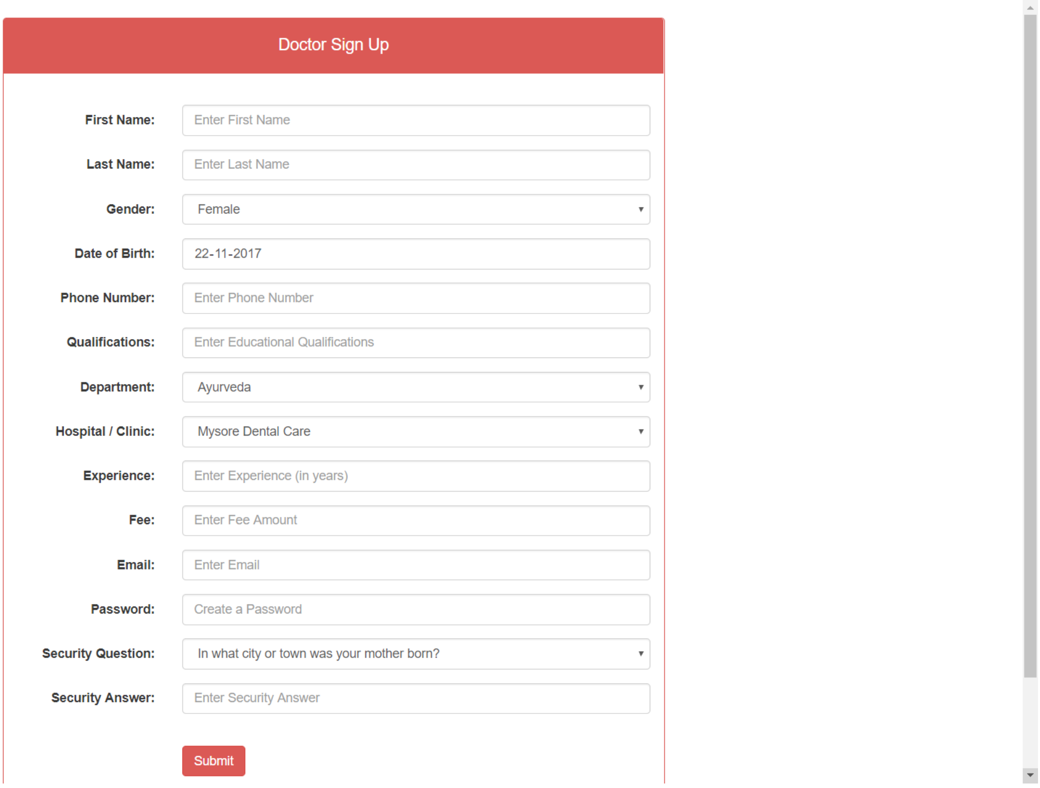
Email: jon@outlook.com
Phone Number: 6958754856

Your Upcoming Bookings

Booking ID	Booking Date	Appointment Date	Time	Doctor Name	Status
36890010	2017-11-23	2017-11-23	18:00:00	Dr. Aditya Sinha	Cancelled
36890011	2017-11-23	2017-11-24	12:00:00	Dr. Anushree N	Booked
36890012	2017-11-23	2017-11-25	12:00:00	Dr. Aditya Sinha	Booked

Your Past Bookings

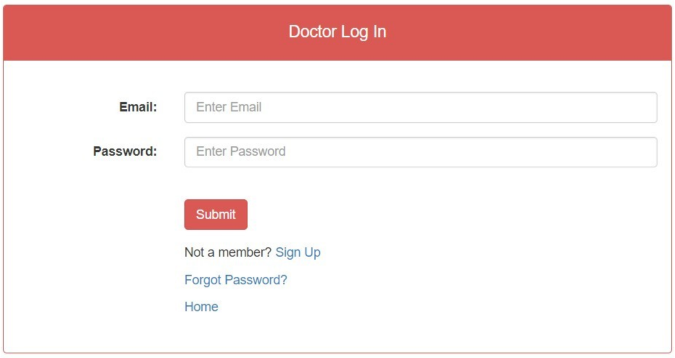
Booking ID	Booking Date	Appointment Date	Time	Doctor Name	Status
36890001	2017-11-22	2017-11-22	11:00:00	Dr. Anushree N	Booked
36890002	2017-11-22	2017-11-22	18:00:00	Dr. Harindra M N	Booked



Doctor Sign Up

First Name:	Enter First Name
Last Name:	Enter Last Name
Gender:	Female
Date of Birth:	22-11-2017
Phone Number:	Enter Phone Number
Qualifications:	Enter Educational Qualifications
Department:	Ayurveda
Hospital / Clinic:	Mysore Dental Care
Experience:	Enter Experience (in years)
Fee:	Enter Fee Amount
Email:	Enter Email
Password:	Create a Password
Security Question:	In what city or town was your mother born?
Security Answer:	Enter Security Answer

CHAPTER 8



Doctor Log In

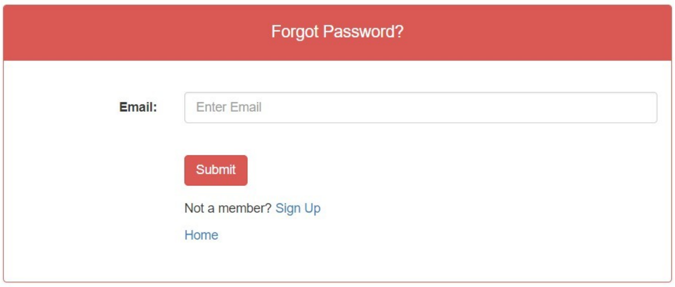
Email: Enter Email

Password: Enter Password

Not a member? Sign Up

Forgot Password?

Home

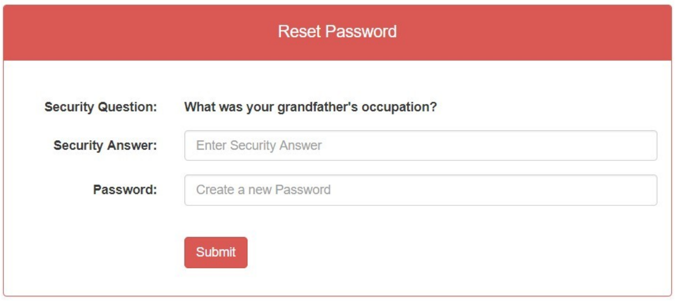


Forgot Password?

Email: Enter Email

Not a member? Sign Up

Home



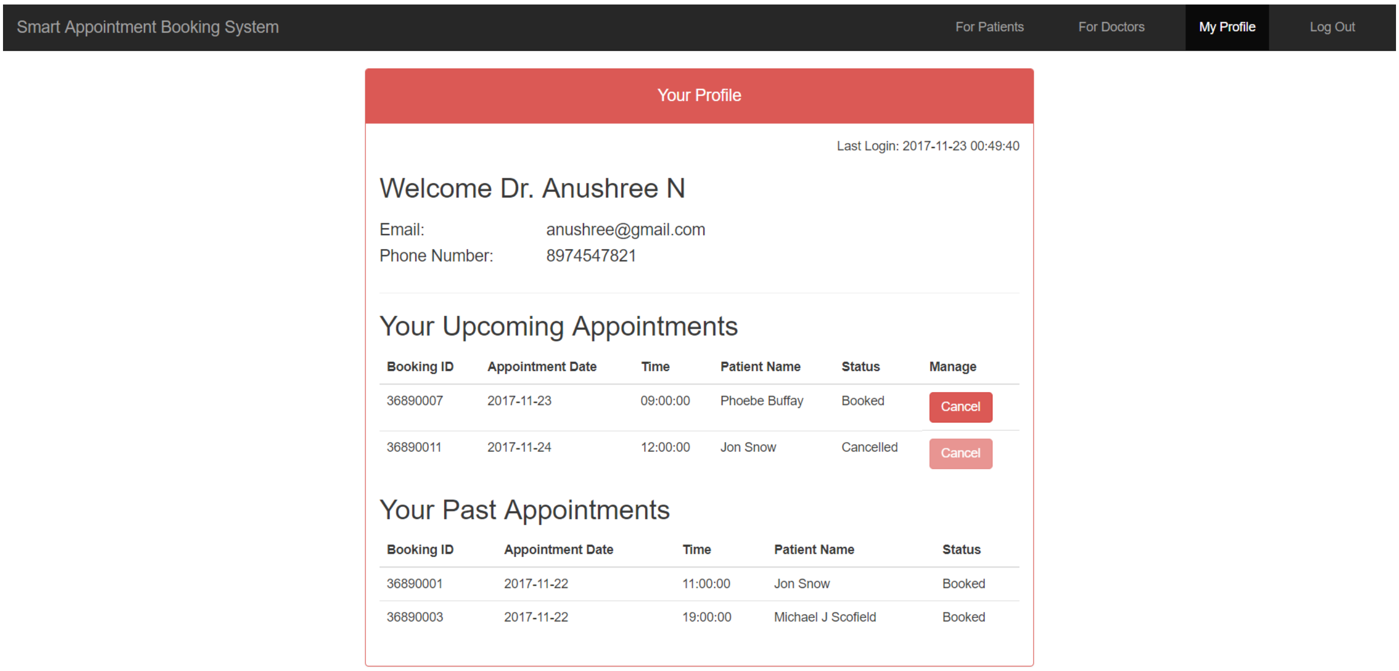
Reset Password

Security Question: What was your grandfather's occupation?

Security Answer: Enter Security Answer

Password: Create a new Password

CHAPTER 8



Last Login: 2017-11-23 00:49:40

Welcome Dr. Anushree N

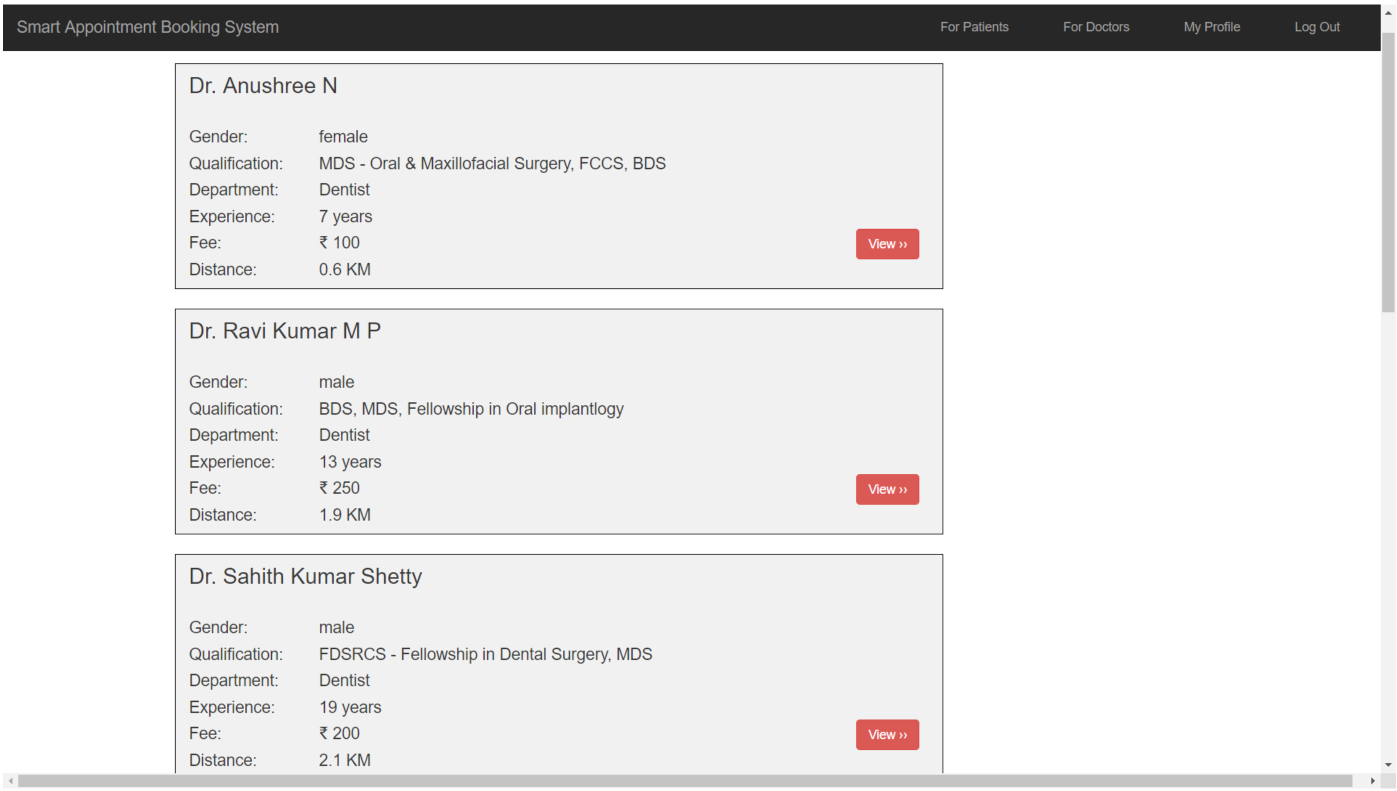
Email:	anushree@gmail.com
Phone Number:	8974547821

Your Upcoming Appointments

Booking ID	Appointment Date	Time	Patient Name	Status	Manage
36890007	2017-11-23	09:00:00	Phoebe Buffay	Booked	Cancel
36890011	2017-11-24	12:00:00	Jon Snow	Cancelled	Cancel

Your Past Appointments

Booking ID	Appointment Date	Time	Patient Name	Status
36890001	2017-11-22	11:00:00	Jon Snow	Booked
36890003	2017-11-22	19:00:00	Michael J Scofield	Booked



Dr. Anushree N

Gender:	female
Qualification:	MDS - Oral & Maxillofacial Surgery, FCCS, BDS
Department:	Dentist
Experience:	7 years
Fee:	₹ 100
Distance:	0.6 KM

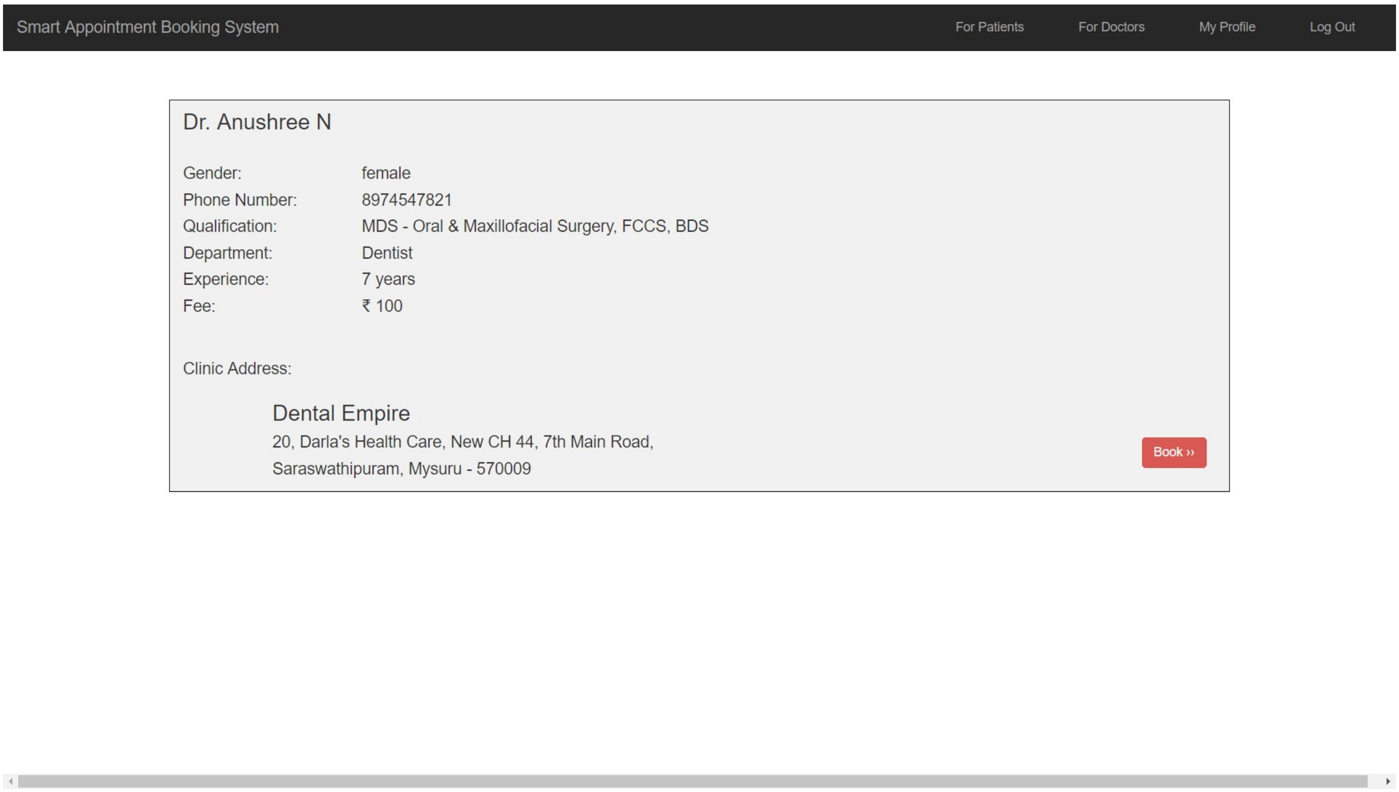
Dr. Ravi Kumar M P

Gender:	male
Qualification:	BDS, MDS, Fellowship in Oral implantlogy
Department:	Dentist
Experience:	13 years
Fee:	₹ 250
Distance:	1.9 KM

Dr. Sahith Kumar Shetty

Gender:	male
Qualification:	FDSRCS - Fellowship in Dental Surgery, MDS
Department:	Dentist
Experience:	19 years
Fee:	₹ 200
Distance:	2.1 KM

CHAPTER 8

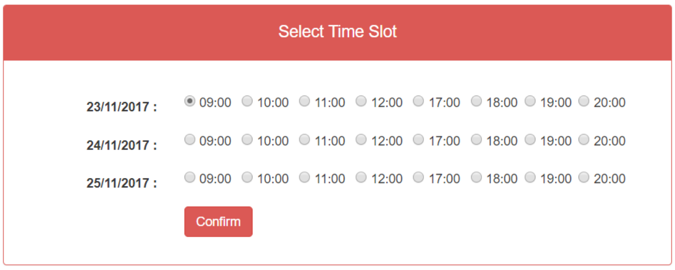


Dr. Anushree N

Gender:	female
Phone Number:	8974547821
Qualification:	MDS - Oral & Maxillofacial Surgery, FCCS, BDS
Department:	Dentist
Experience:	7 years
Fee:	₹ 100

Clinic Address:

Dental Empire
20, Darla's Health Care, New CH 44, 7th Main Road,
Saraswathipuram, Mysuru - 570009



Select Time Slot

23/11/2017 :	● 09:00	○ 10:00	○ 11:00	○ 12:00	○ 17:00	○ 18:00	○ 19:00	○ 20:00
24/11/2017 :	○ 09:00	○ 10:00	○ 11:00	○ 12:00	○ 17:00	○ 18:00	○ 19:00	○ 20:00
25/11/2017 :	○ 09:00	○ 10:00	○ 11:00	○ 12:00	○ 17:00	○ 18:00	○ 19:00	○ 20:00

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENTS

CONCLUSION:

This application reduces the unwanted time taken to manually book appointment to a specific doctor.
Doctor can check their profile to see the timings he has to treat a patient. Moreover, he/she can also cancel the appointment if he/she is too busy to treat at the specific time.

Also, the application cannot make use of the users' private data. The user's password and answer are encrypted by PHP.

The data in the database is backed up frequently to recover the data during system failure using transactions.

FUTURE ENHANCEMENTS:

This application provides only the basic picture of the project which is applicable for a single constituency, the fully developed version can be made with the help of the MySQL and other software required for the further detailed version.

This application can be further extended by making the application available in other cities also.

Moreover, we can add review system such that patient can rate their experience with the doctor. We will make the mobile app version of the project such that it can be used anywhere anytime.

CHAPTER 9

We intend to add emergency button such that in emergency case, an ambulance can be dispatched to the specified location.

REFERENCES

WEBSITES:

https://www.mysql.com/

https://stackoverflow.com/

https://www.php.net/

https://www.apachefriends.org/

https://www.w3schools.com/

TEXTBOOKS:

“DATABASE SYSTEMS” – Ramez Elmasri, Shamkant B. Navathe.

“THE DATABASE BOOK – PRINCIPLE AND PRACTICE USING MYSQL” – Narain Gehani.