A DBMS Lab Project Report

on

ONLINE APPOINTMENT MANAGEMENT SYSTEM

By

A.V.L. Likhita

1602-18-733-058

N.V. Kiran Reddy

1602-18-733-057

Department of Computer Science & Engineering

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2020-21

INDEX

CONTENTS

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to Smt. B. SYAMALA, our project guide, for her valuable guidance and constant support, along with her capable instruction and persistent encouragement.

We are grateful to our Head of Department, Dr T. ADILAKSHMI, for her steady support and the provision of every resource required for the completion of this project.

We would like to take this opportunity to thank our Principal, Dr S. V. RAMANA, as well as the management of the institute, for having designed an excellent learning atmosphere.

ABSTRACT

The Online Appointment Management system is a computerized management system. This system keeps the records of the appointments which is being registered by the customer and verified by the administrator.

This project has GUI based software that will help in storing, updating and retrieving the information through various user-friendly menu-driven modules.

The project “Online Appointment Management System” is aimed to develop to maintain the daily appointments of customers with different doctors and clinics. It helps the customer to think wisely to visit the doctor of his choice.

LIST OF TABLES

LIST OF FIGURES

LIST OF SCREENSHOTS

1. INTRODUCTION

1.1 Overview

Online Appointment Management System is a computerized management system. This system keeps the records of the Online Appointment being booked in this organization. The proposed system will keep a track of different appointment details including the doctors and clinic details. This project has GUI based software that will help in storing, updating, and retrieving the information through various user-friendly menu-driven modules. The project “Online Appointment Management System” is aimed to develop to maintain the day-to-day appointments. Main objective of this project is to provide appointments and clinic details to customers. This software application will help admin to handle customer information, doctor details and clinic details. Detailed explanation about modules and design are provided in project documentation. The existing system is a manually maintained system. All the appointment records are to be maintained for the details of each customer, doctor details, appointment registration, etc.

1.2 Motivation

One of the prime reasons that online appointment management system is gaining popularity in recent days is that the system provides an easier platform for receiving healthcare facilities to the general users. In this pandemic it is gaining popularity due to a smaller number of facilities.

1.3 Problem Definition

In the existing manual system, a lot of time is spent in communicating and sending the information across different branches and their independent website. The current world still works on the traditional, orthodox system of written entries of the registration and manually submitting all the information of booking appointment which is tedious and inefficient. There are more chances for the humans involved in the system to error, and this old-fashioned method also takes a long time to execute manually, even after not considering the high chances of mistakes. Also, due to the constant changing of the day-to-day life, it will lead to proportionate changes to the cost and appointments, which makes updating them dynamically everywhere tough.

All the old techniques and modus-operandi prove to be an inefficient and mammoth task, and we need to overcome this.

There is a need for an integrated automated system, which has some centralized control over the entire process. Conventional System makes use of huge amounts of paper for recording transactions. The existing system is a manually maintained system. All the appointment records are to be maintained for the details of each customers, doctor's details, clinic details and appointments details etc. All these details are entered and retrieved manually.

1.4 Objectives

The limited time and resources have restricted us to incorporate, in this project, only the main activities that are performed in a ‘Online Appointment Management System’, but utmost care has been taken to make the system efficient and user friendly.

“Online Appointment Management System” has been designed to computerize the functions for entering and retrieving the following details:

1) Booking: Username, Name, Gender, CID, DID, DOV, Timestamp, Status.
2) Clinic: CID, Address, Name, Town, City, Contact.
3) Doctor: DID, Name, Gender, DOB, Experience, Specialization, Contact, Address, Username, Password, Region.
4) Doctor Available: CID, DID, Day, Start time, End time.
5) Patients: Name, Gender, DOB, Phone, Username, Password, Email.
6) Deleted Doctors: DID, Name, Gender, DOB, Experience, Specialization, Contact, Address, Username, Password, Region.
7) Admin: Username, Password

1.5 Scope

Online appointment scheduling system is a system through which a user or simply, a patient can access the website of the doctor, and through the online software, the patient can easily make their appointments. Admin of the clinic can update the appointment status of the patients making it more informative.

2. ONLINE APPOINTMENT MANAGEMENT SYSTEM

2.1 Proposed System

Proposed system is a computerized version of the existing system which provides easy and quick access over the data keeping records of the land details so that the verification and registering the details is effective and efficient.

2.1.1 Advantages of present system

1. Storing customer details and appointment details properly
2. Maintains accuracy of data and reduces error
3. Less time consuming
4. Accuracy of data
5. Less prone to errors

2.2 System Requirement Specification

2.2.1 Software Specifications

- Operating System - Windows 95/98/2000/XP/8/8.1/10
- Application Server - Apache
- Front End - HTML, CSS
- Back End – PHP
- Database Connectivity – MySQL
- Connecting front end and back end - PHP

2.2.2 Hardware Specifications

Online appointment management system is a web interface where the user can be able to access a web portal using a mobile phone or a personal computer. The devices need to have a web browser which supports java script, cookies, sessions, an interface to enter the required details using keyboard and a mouse to navigate.

3. MODULE DESCRIPTION

3.1 ADMIN MODULE

This module provides administrator related functionalities. Administrator can view the registered customers and their details, doctors available, appointment details along with the current approval status. In this module admin must verify all the details of the customer and then he either approve or reject the appointment of the doctor that the customer is requesting. Admin will give the responses to customer based on submitted details always checking the doctor details and authenticity of the details.

This module also requires the admin logged in to keep all the customer information confidential and not disclose it to any other third party, not compromising the data to unauthorized access.

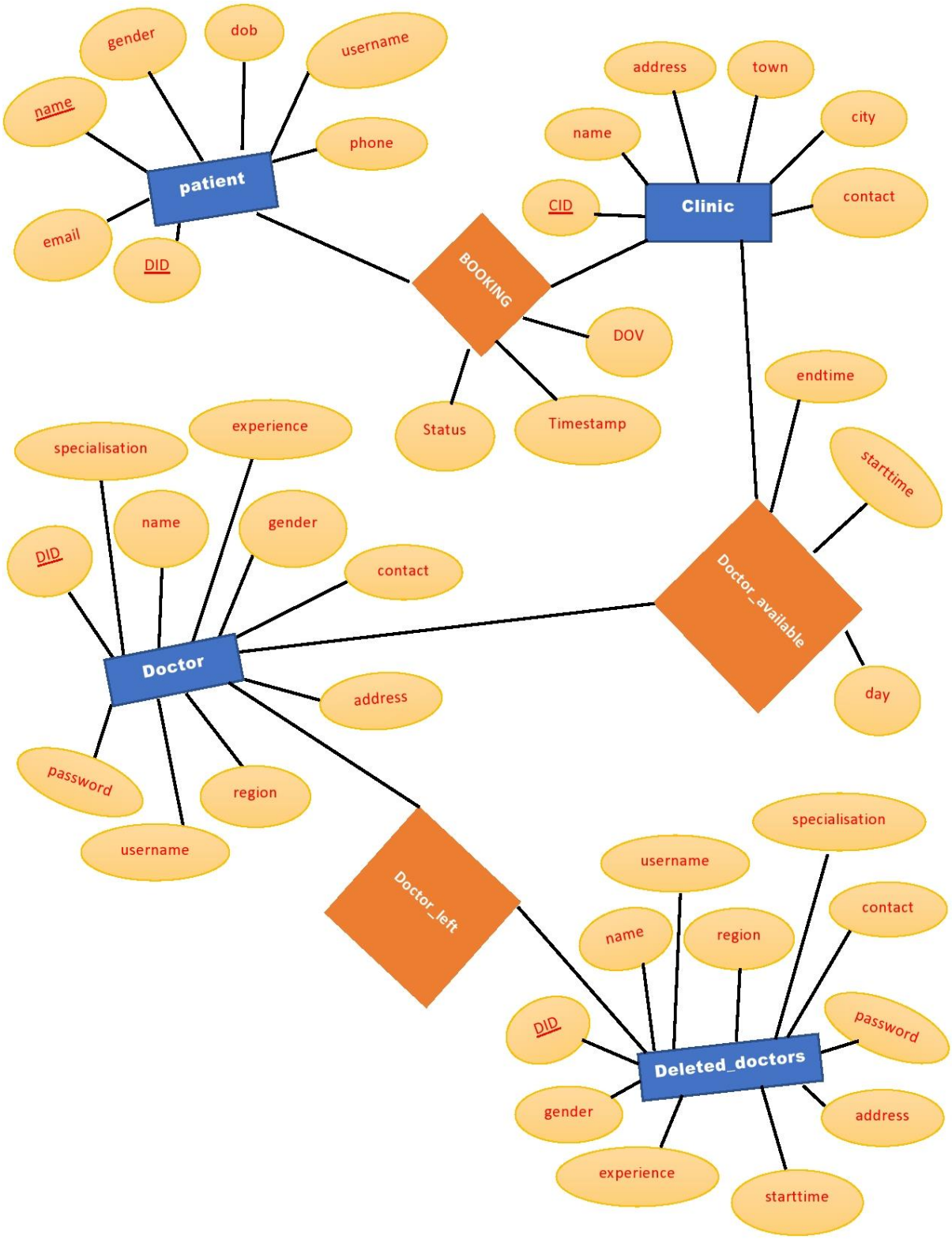
3.2 USER MODULE

This module is about users of this portal. By using this module user, i.e., the customer can give their details along with all the necessary details. User must be registered with the system. Customer booking details must be kept confidential after that he/she will give their personal details through the website. By using this login id, the customer just needs to book appointments giving all the details of the appointment such as username, city, clinic, doctor, and date of visit. Then the customer must wait until any progress is updated. After the booking the customer can check the booking status for the respective doctor.

4. SYSTEM DESIGN

4.1 DATABASE DESIGN

4.1.1 ER Diagram



4.1.2 Database tables

Online Appointment Management System consists of seven tables. It is used by a Customer and admin to store and retrieve details regarding doctors, clinics, bookings.

The database uses the following tables for maintaining the details:

- Booking
- Clinic
- Doctor
- Doctor Available
- Patients
- Deleted Doctors
- Admin

The description of the tables is as follows:

Table 4.1.2.1 **Booking:** This table consists of details of the appointments

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
Username	Varchar (30)	Primary key	Username of customer
Fname	Varchar (30)	Not Null	Name of the customer
Gender	Varchar (10)	Not Null	Gender of customer
CID	Int (11)	Not Null	Clinic ID
DID	Int (11)	Not Null	Doctor ID
DOV	Date	Not Null	Date of Visit
Timestamp	Datetime	Not Null	Date and time the booking is done
Status	Varchar (50)	Not Null	Status of appointment

Table 4.1.2.2 **Clinic:** This table consists of the clinic details.

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
CID	Int (11)	Primary Key	The ID accepts the integer which is used to identify the clinic.
Name	Varchar (30)	Not Null	Name of the clinic
Address	Varchar (30)	Not Null	Address of the clinic
Town	Varchar (20)	Not Null	Town where the clinic is located
City	Varchar (20)	Not Null	City where the clinic is located
Contact	Varchar (10)	Not Null	Contact of the clinic

Table 4.1.2.3 **Doctor:** This table consists of the doctor details

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
DID	Int (11)	Primary Key	The ID accepts the integer which is used to identify the doctor.
Name	Varchar (30)	Not Null	Name of the doctor
Gender	Varchar (10)	Not Null	Gender of the doctor
DOB	Date	Not Null	Doctor's Date of Birth
Experience	Varchar (30)	Not Null	Doctor's working experience in years
Specialisation	Varchar (30)	Not Null	An expert or specialist of doctor
Contact	Varchar (10)	Not Null	Doctor contact details
Address	Varchar (40)	Not Null	Doctors address or home resident
Username	Varchar (30)	Not Null	Username of doctor
Password	Varchar (20)	Not Null	Doctors password
Region	Varchar (20)	Not Null	Doctor resident city

Table 4.1.2.4 **Doctor Available:** Table consists of the doctor's availability in the clinic

COLOUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
CID	Int (11)	Not Null	The id accepts the integer which is used to identify the clinic
DID	Int (11)	Primary key	The id accepts the integer which is used to identify the doctor
Day	Varchar (20)	Not Null	The day doctor is available in the clinic
Start time	Time	Not Null	Clinic opening time
End time	Time	Not Null	Clinic closing time

Table 4.1.2.5 **Patients:** This table consists of the details about the customer.

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
Name	Varchar (30)	Primary Key	Name of the customer
Gender	Varchar (10)	Not Null	Gender of the customer.
DOB	Varchar (50)	Not Null	Customer's Date of Birth
Phone	Varchar (10)	Not Null	Phone number of the customer.
Username	Varchar (20)	Unique	Username of the customer.
Password	Varchar (30)	Not Null	Customer's secured password used while logging in.
Email	Varchar (30)	Not Null	Customer's email used to send mails regarding appointments

Table 4.1.2.6 **Deleted Doctors:** This table consists of the doctor details deleted by admin(Trigger).

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
DID	Int (11)	Primary Key	The ID accepts the integer which is used to identify the doctor.
Name	Varchar (30)	Not Null	Name of the doctor
Gender	Varchar (10)	Not Null	Gender of the doctor
DOB	Date	Not Null	Doctor's Date of Birth
Experience	Varchar (30)	Not Null	Doctor's working experience in years
Specialisation	Varchar (30)	Not Null	An expert or specialist of doctor
Contact	Varchar (10)	Not Null	Doctor contact details
Address	Varchar (40)	Not Null	Doctors address or home resident
Username	Varchar (30)	Not Null	Username of doctor

Table 4.1.2.7 **ADMIN:** This table consists of admin details.

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
Username	Varchar (20)	Primary Key	Username of the admin
Password	Varchar (10)	Not Null	Password of the admin

5. ACTIVITIES PERFORMED

To achieve the project goal effectively, there are some specific objectives implemented.

The following were specific objectives for this project.

- To provide easy access for booking appointment.
- To provide easy access to manage appointments.
- To provide easy access to patients for checking doctor availablity .
- To provide easy access to customer records to admin.
- To minimize human errors.

6. IMPLEMENTATION

6.1 SAMPLE CODE

Appointment.sql:

```
--

DELIMITER $$
--
-- Procedures
--
CREATE DEFINER=`root`@`localhost` PROCEDURE `Src` ()  NO SQL
SELECT *FROM patient$$

DELIMITER ;

-- --------------------------------------------------------

--
-- Table structure for table `booking`
--

CREATE TABLE `booking` (
  `username` varchar(30) NOT NULL,
  `Fname` varchar(30) NOT NULL,
  `gender` varchar(10) NOT NULL,
  `CID` int(11) NOT NULL,
  `DID` int(11) NOT NULL,
  `DOV` date NOT NULL,
  `Timestamp` datetime NOT NULL,
  `Status` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```

```
-- Dumping data for table `booking`
--
-- Table structure for table `clinic`
--

CREATE TABLE `clinic` (
  `CID` int(11) NOT NULL,
  `name` varchar(30) NOT NULL,
  `address` varchar(30) NOT NULL,
  `town` varchar(20) NOT NULL,
  `city` varchar(20) NOT NULL,
  `contact` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `clinic`
--
-- Table structure for table `deleted_doctors`
--

CREATE TABLE `deleted_doctors` (
  `DID` int(11) NOT NULL,
  `name` varchar(30) NOT NULL,
  `gender` varchar(10) NOT NULL,
  `dob` date NOT NULL,
  `experience` varchar(30) NOT NULL COMMENT '(years)',
  `specialisation` varchar(30) NOT NULL,
  `contact` varchar(10) NOT NULL,
  `address` varchar(40) NOT NULL,
  `username` varchar(30) NOT NULL,
  `password` varchar(20) NOT NULL,
  `region` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Dumping data for table `deleted_doctors`
--
-- Table structure for table `doctor`
--

CREATE TABLE `doctor` (
  `DID` int(11) NOT NULL,
  `name` varchar(30) NOT NULL,
  `gender` varchar(10) NOT NULL,
  `dob` date NOT NULL,
  `experience` varchar(30) NOT NULL COMMENT '(years)',
  `specialisation` varchar(30) NOT NULL,
  `contact` varchar(10) NOT NULL,
  `address` varchar(40) NOT NULL,
  `username` varchar(30) NOT NULL,
  `password` varchar(20) NOT NULL,
  `region` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `doctor`
--
-- Triggers `doctor`
--
DELIMITER $$
CREATE TRIGGER `deletedDoc` AFTER DELETE ON `doctor` FOR EACH ROW INSERT
INTO deleted_doctors(DID,name,gender,dob,experience,specialisation,contact
        ,address,username,password,region) VALUES
(old.DID,old.name,old.gender,old.dob,old.experience,old.specialisation,
        old.contact,old.address,old.username,old.password,old.region)
$$
DELIMITER ;
```

```
-- --------------------------------------------------------

--
-- Table structure for table `doctor_available`
--

CREATE TABLE `doctor_available` (
  `CID` int(11) NOT NULL,
  `DID` int(11) NOT NULL,
  `day` varchar(20) NOT NULL,
  `starttime` time NOT NULL,
  `endtime` time NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `doctor_available`
--
-- Table structure for table `patient`
--

CREATE TABLE `patient` (
  `name` varchar(30) NOT NULL,
  `gender` varchar(10) NOT NULL,
  `dob` date NOT NULL,
  `phone` varchar(10) NOT NULL,
  `username` varchar(20) NOT NULL,
  `password` varchar(30) NOT NULL,
  `email` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Dumping data for table `patient`

-- Indexes for dumped tables
```

```
-- Indexes for table `booking`
--
ALTER TABLE `booking`
  ADD PRIMARY KEY (`username`);
--
-- Indexes for table `clinic`
--
ALTER TABLE `clinic`
  ADD PRIMARY KEY (`CID`);
--
-- Indexes for table `deleted_doctors`
--
ALTER TABLE `deleted_doctors`
  ADD PRIMARY KEY (`DID`);
--
-- Indexes for table `doctor`
--
ALTER TABLE `doctor`
  ADD PRIMARY KEY (`DID`);
--
-- Indexes for table `patient`
--
ALTER TABLE `patient`
  ADD PRIMARY KEY (`name`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

6.2 SCREENSHOTS OF DATABASE

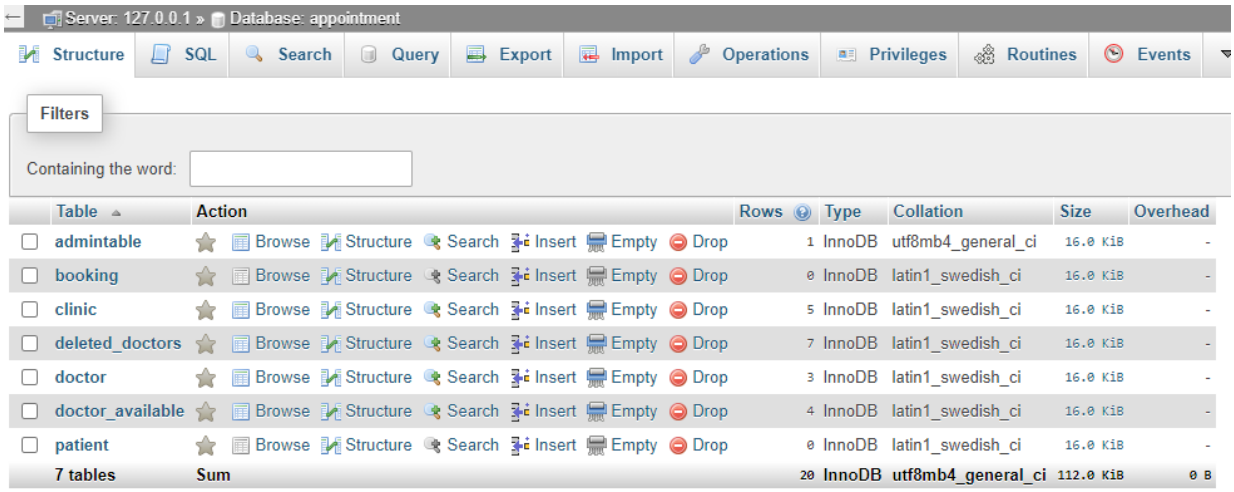


Fig 1: Appointment Database

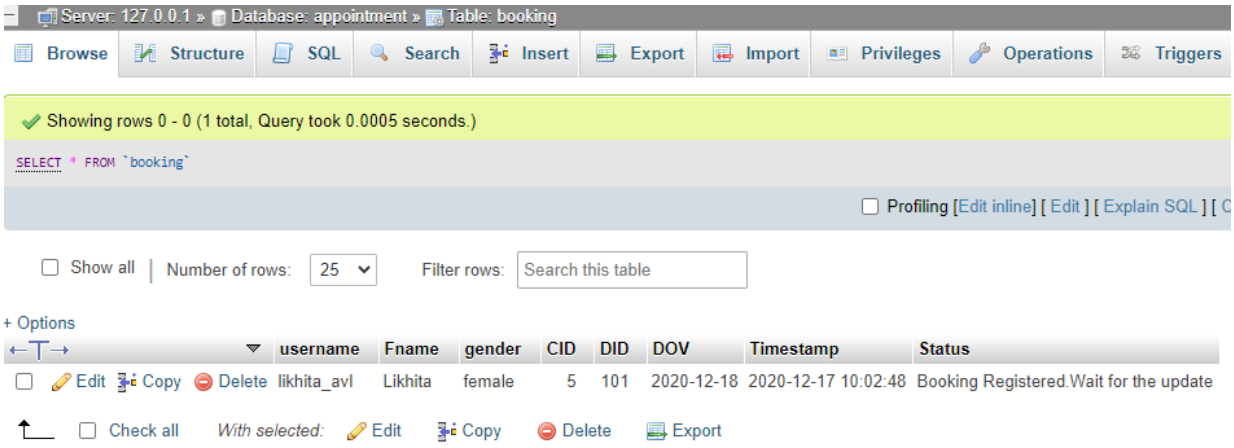


Fig 2: Booking table

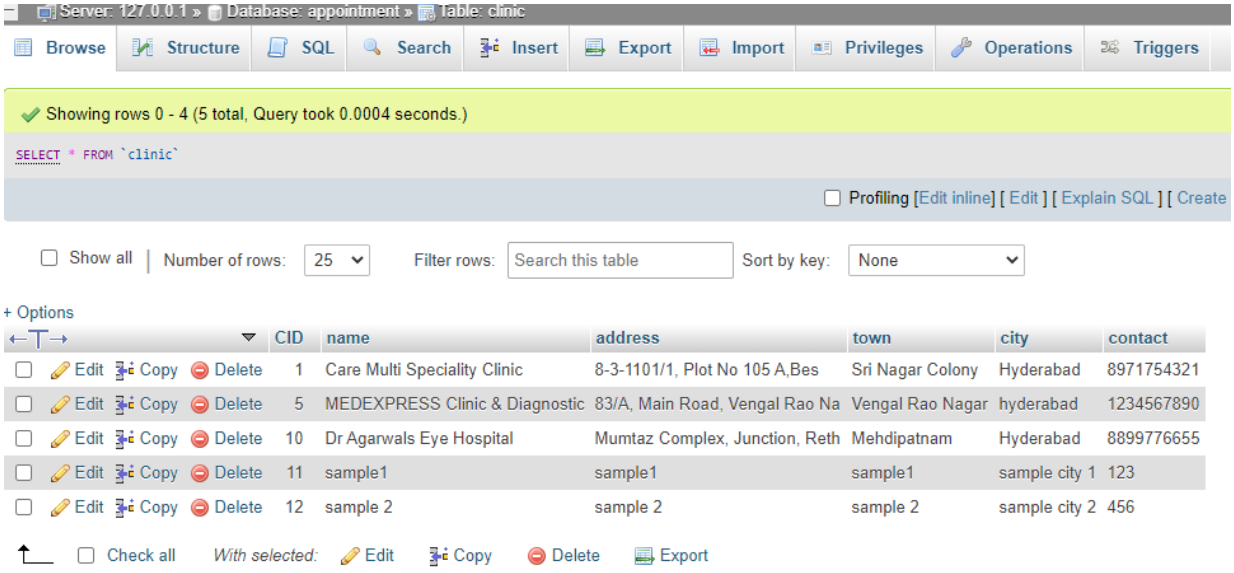


Fig 3: Clinic table

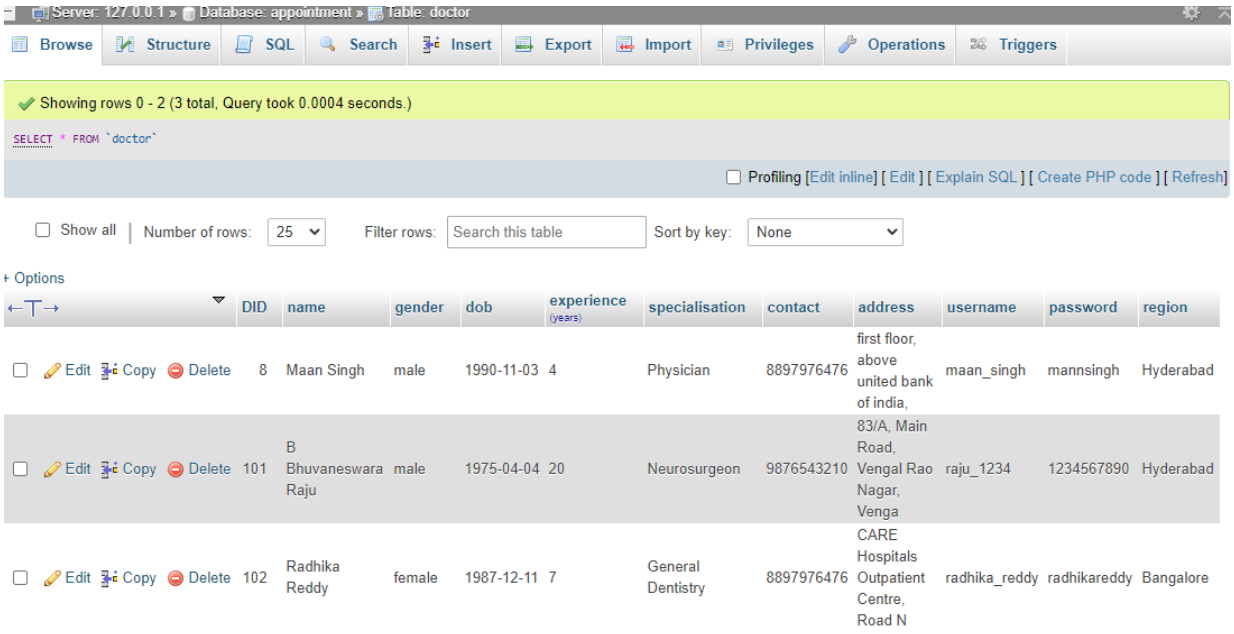


Fig 4: Doctor table

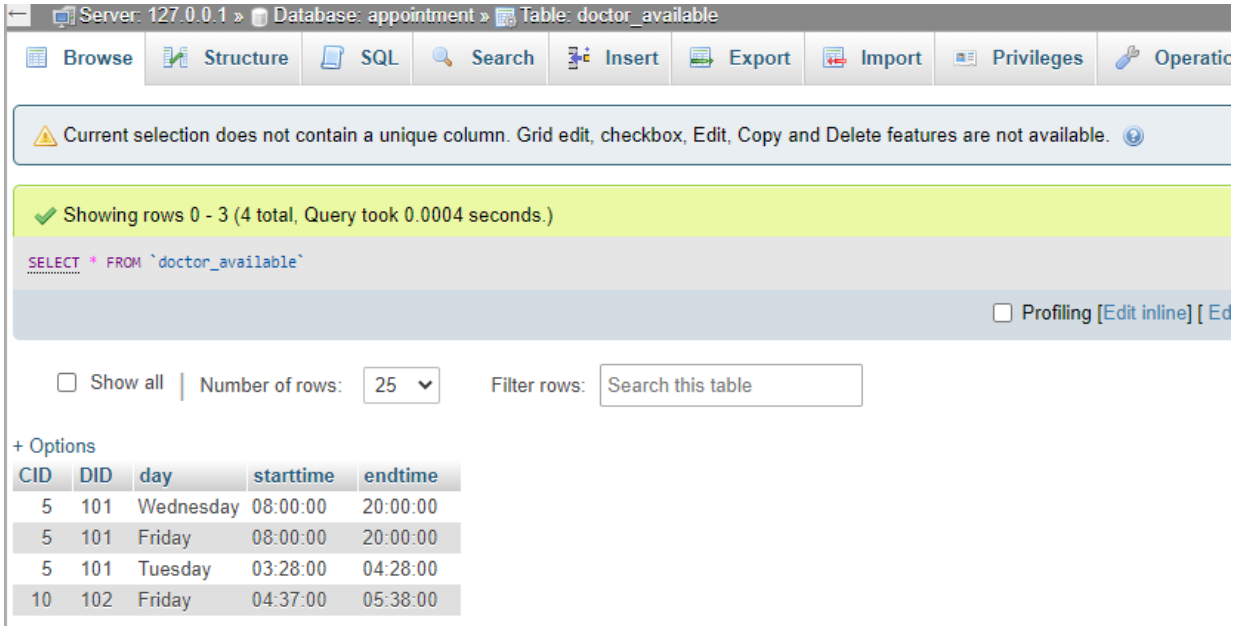


Fig 5: Doctors_available table

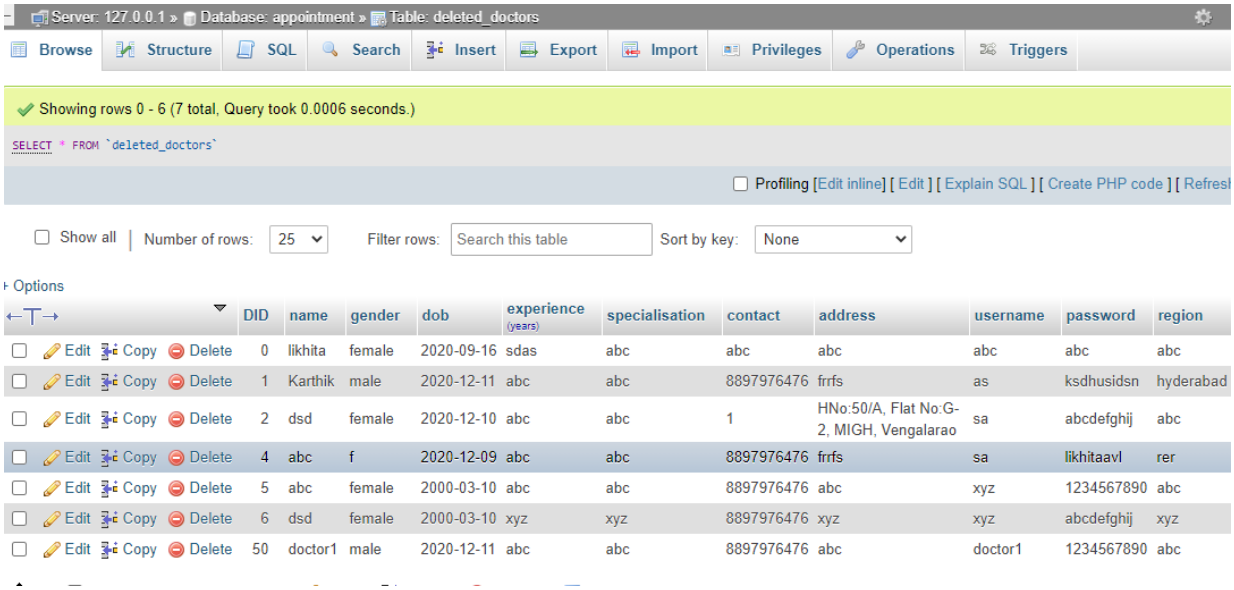


Fig 6: Deleted_doctors table

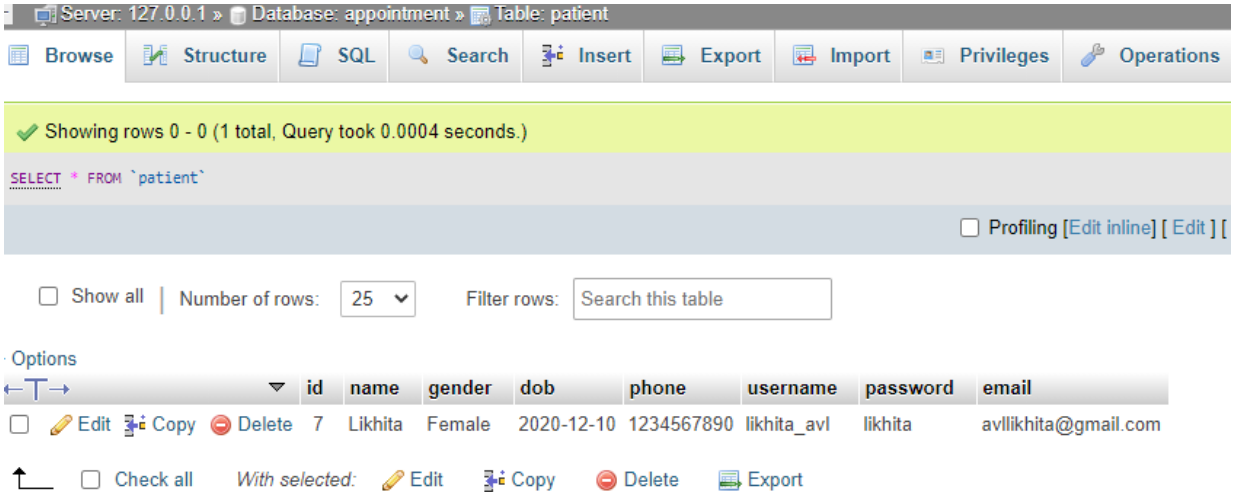


Fig 7: Patient table

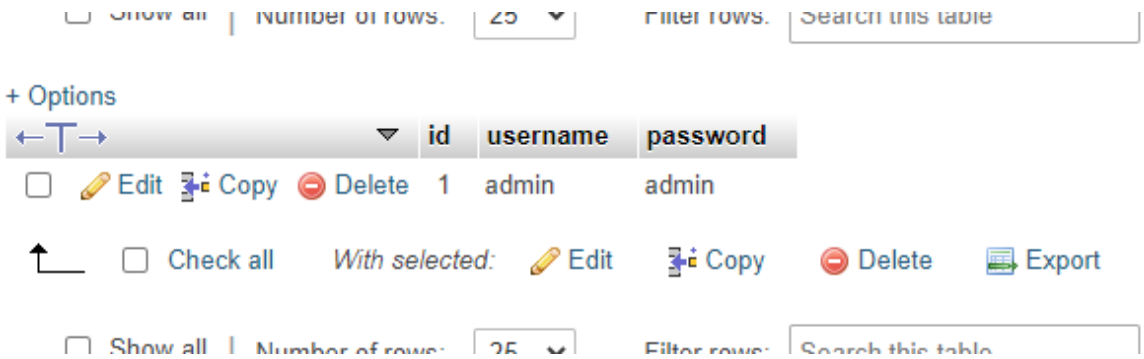


Fig 8: Admin table

6.3 SCREENSHOTS OF UI

User portal:

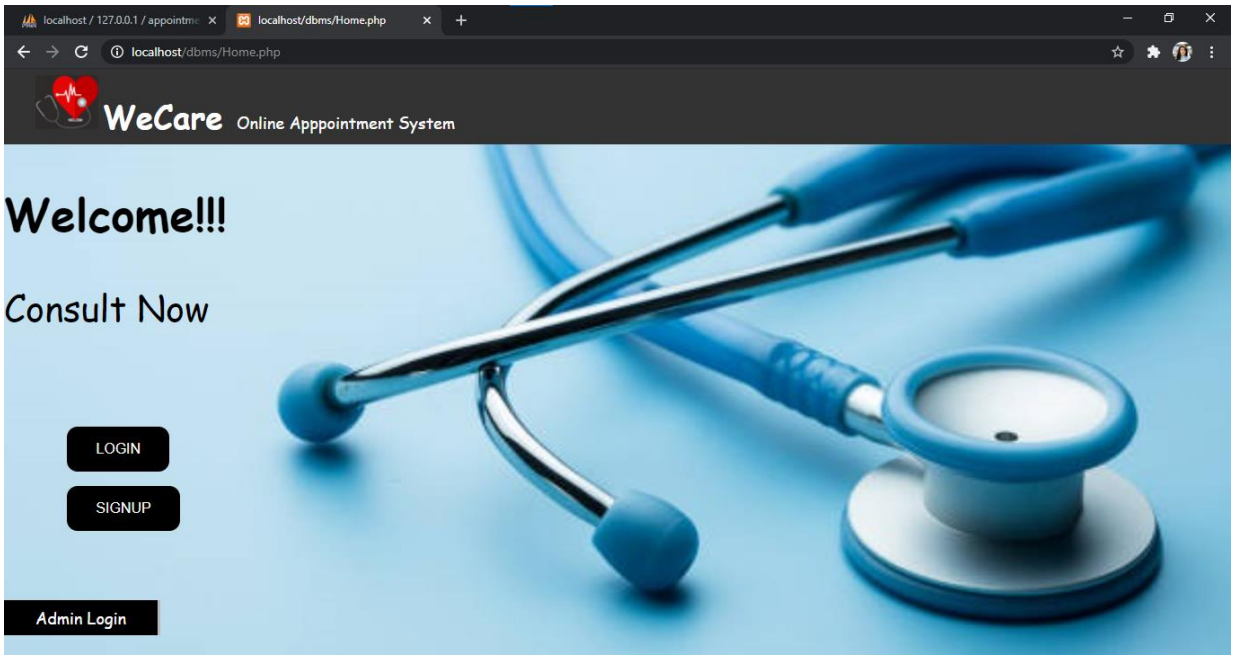


Fig 10: Home page

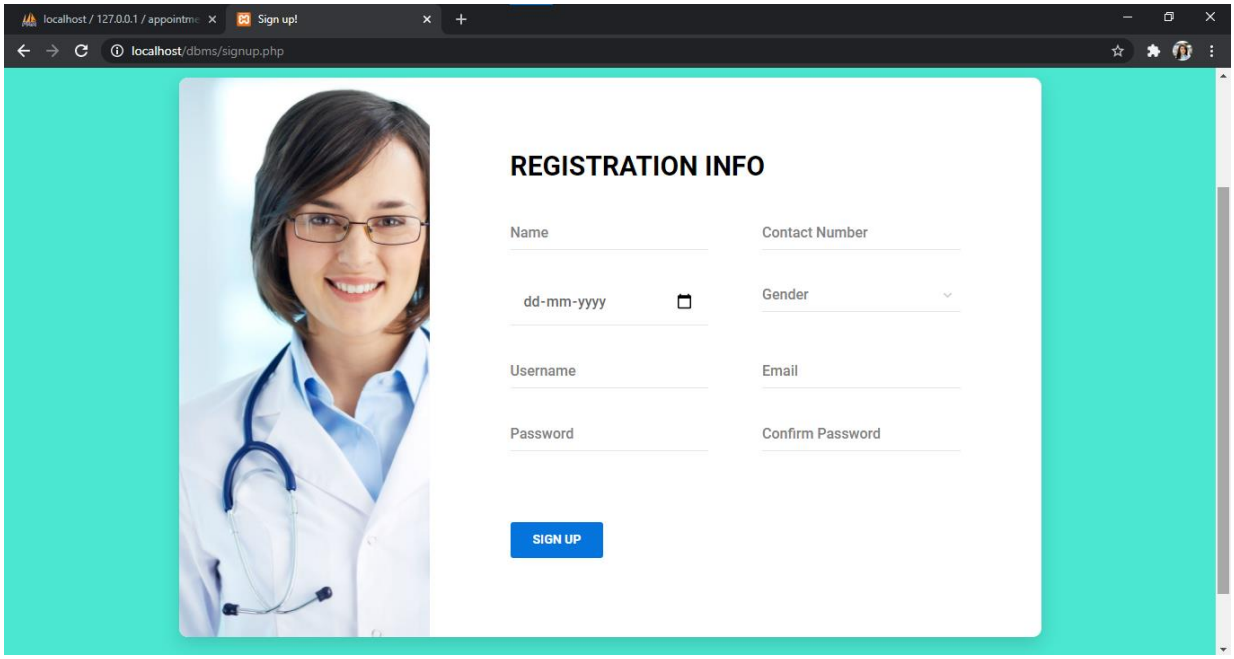


Fig 11: User Sign Up page

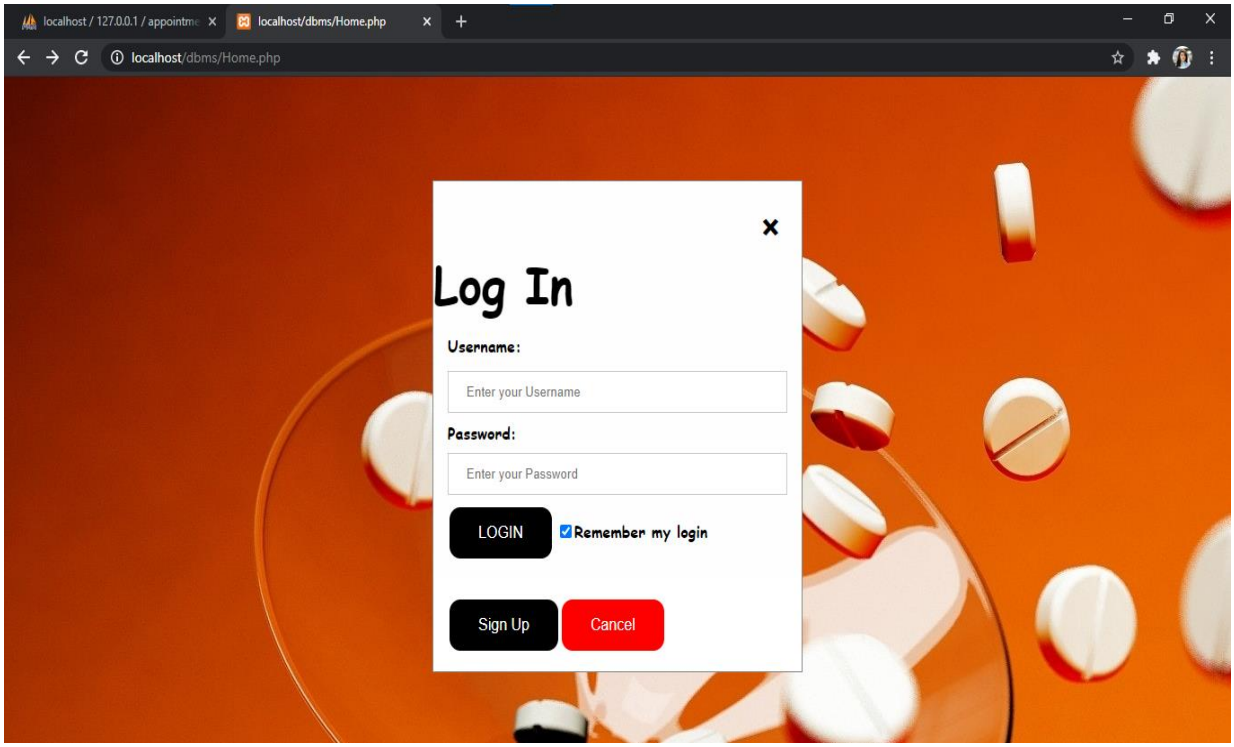


Fig 12: User login page

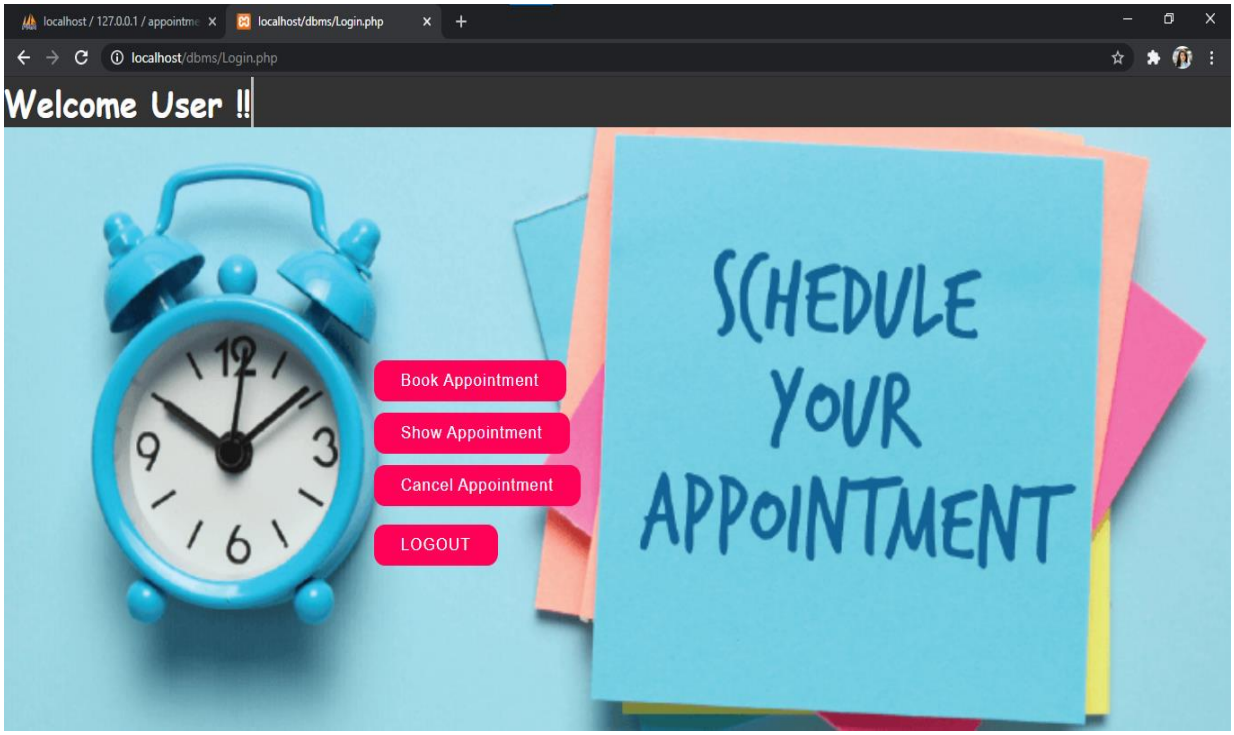


Fig 13: User navigation page

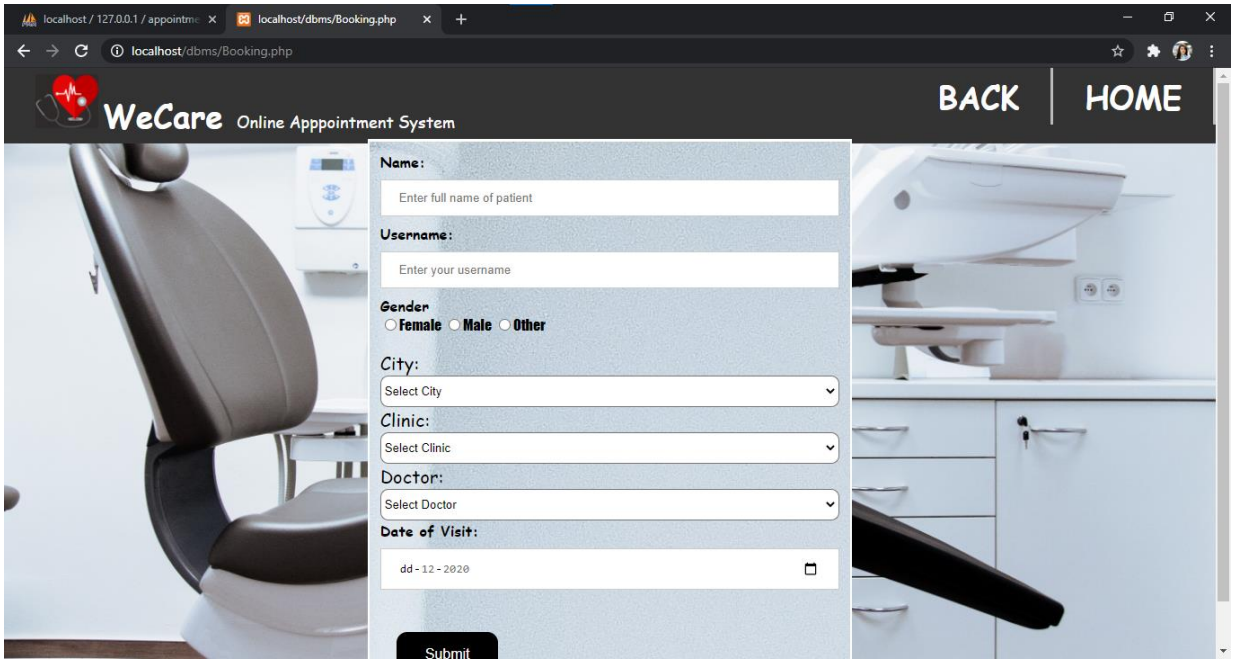


Fig 14: Booking appointment page

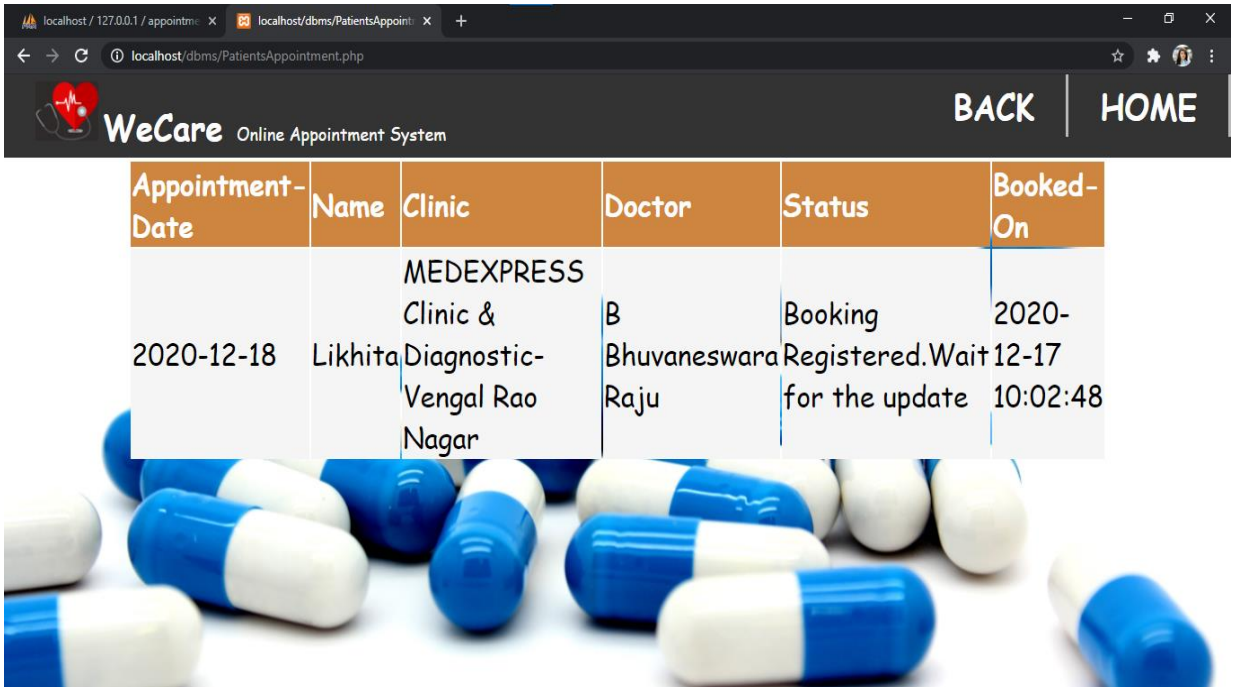


Fig 15: Display appointments page

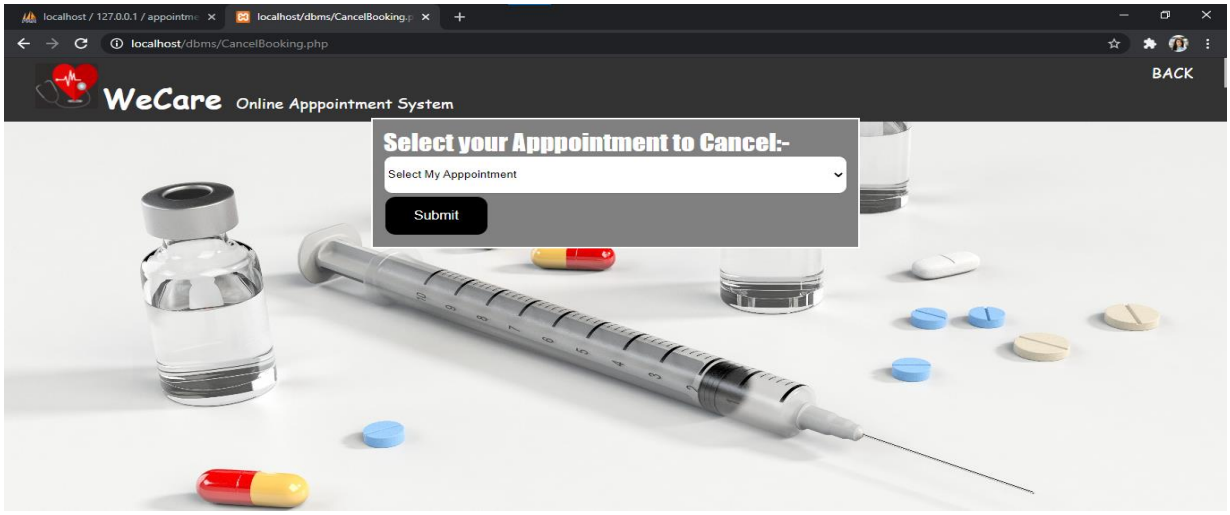


Fig 16: Cancel appointment page

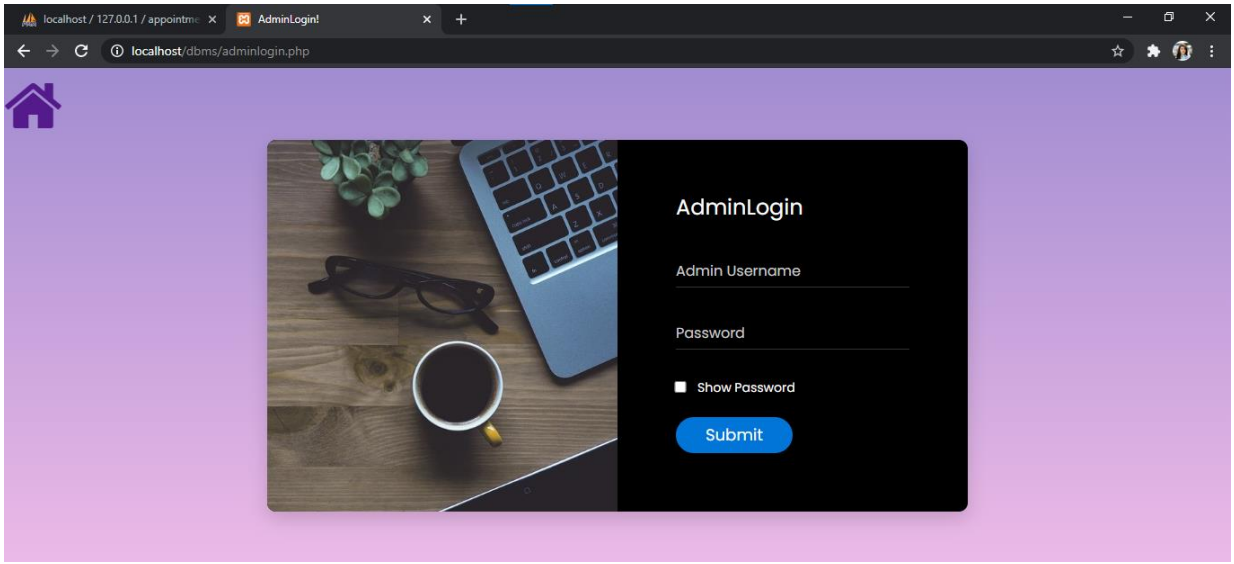


Fig 17: Admin login page

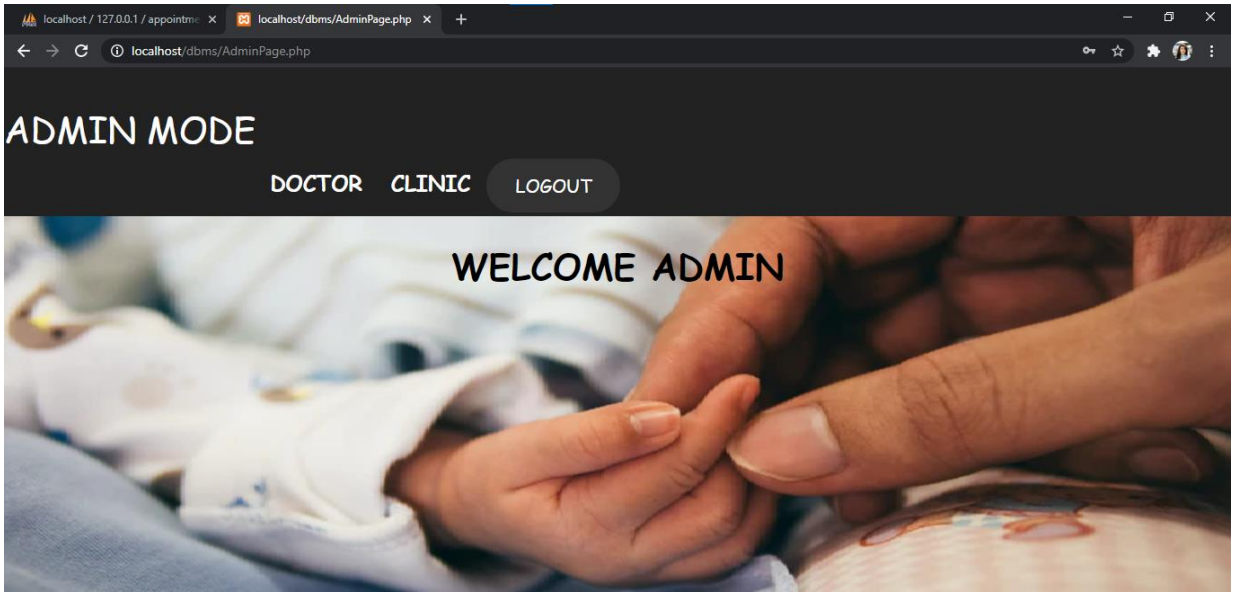


Fig 18: Admin home page (admin portal)

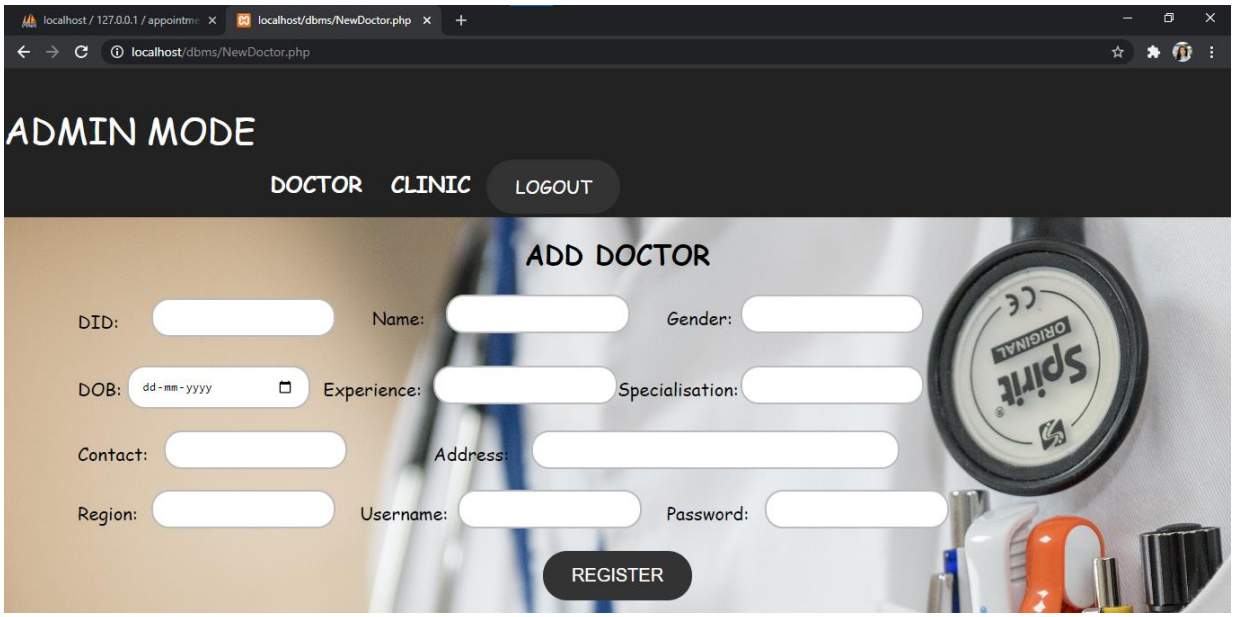


Fig 19: Add new doctor

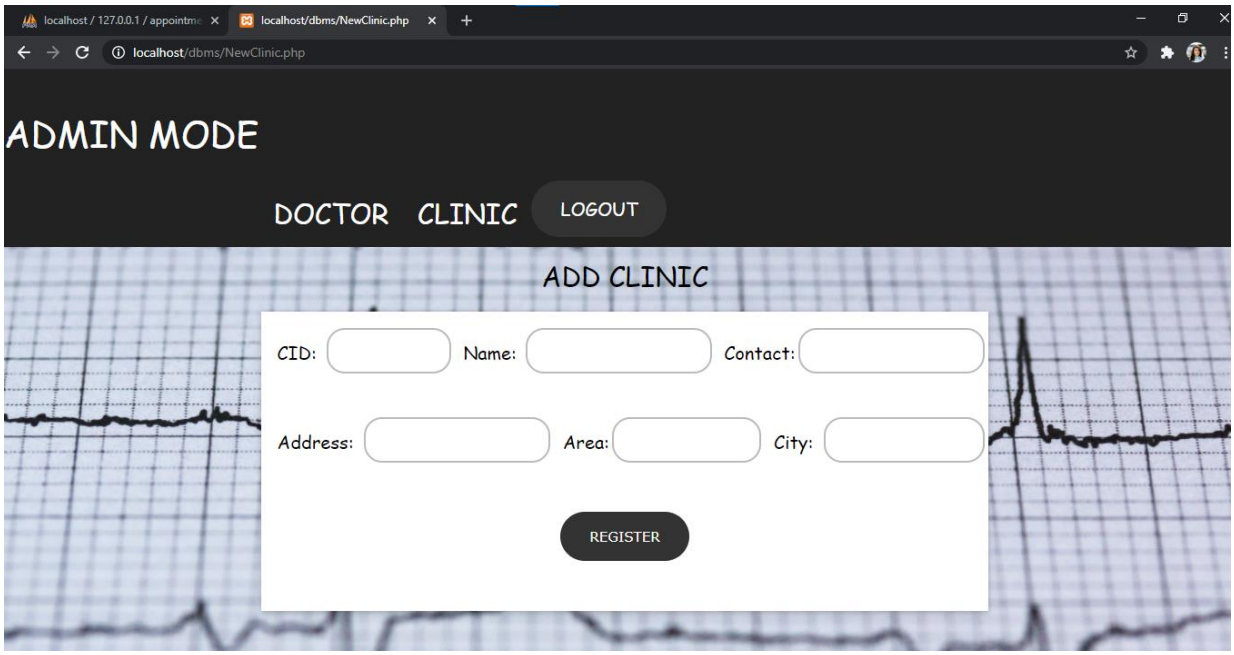


Fig 20: Add new clinic

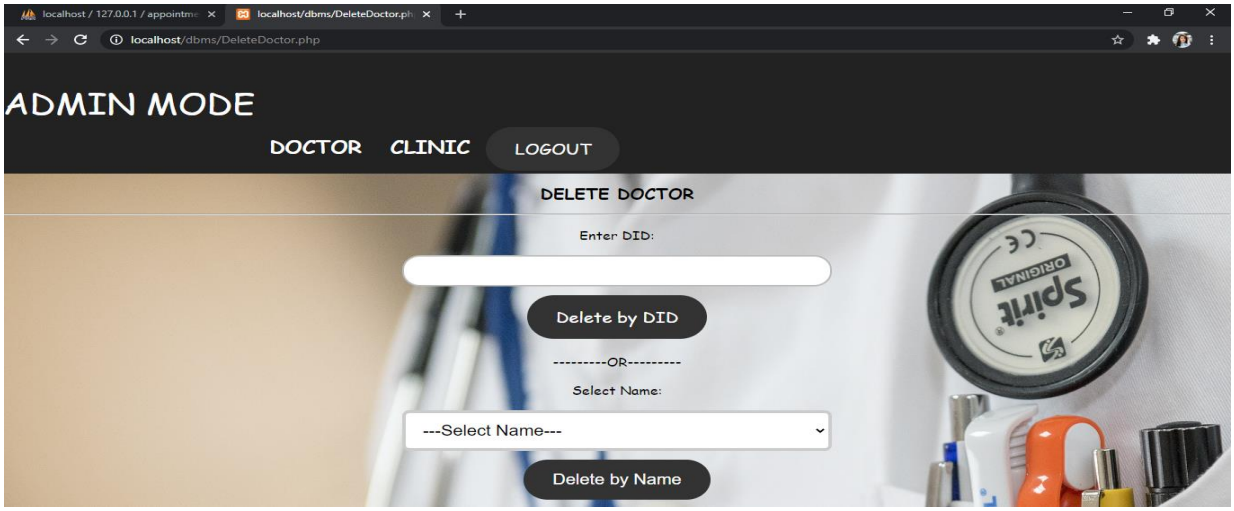
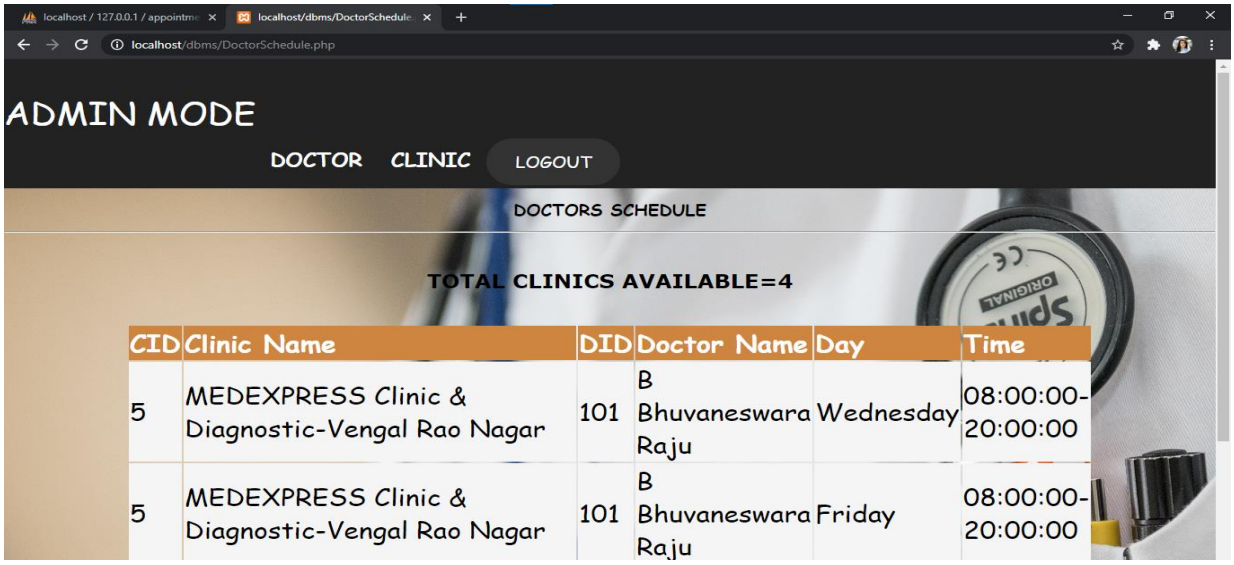
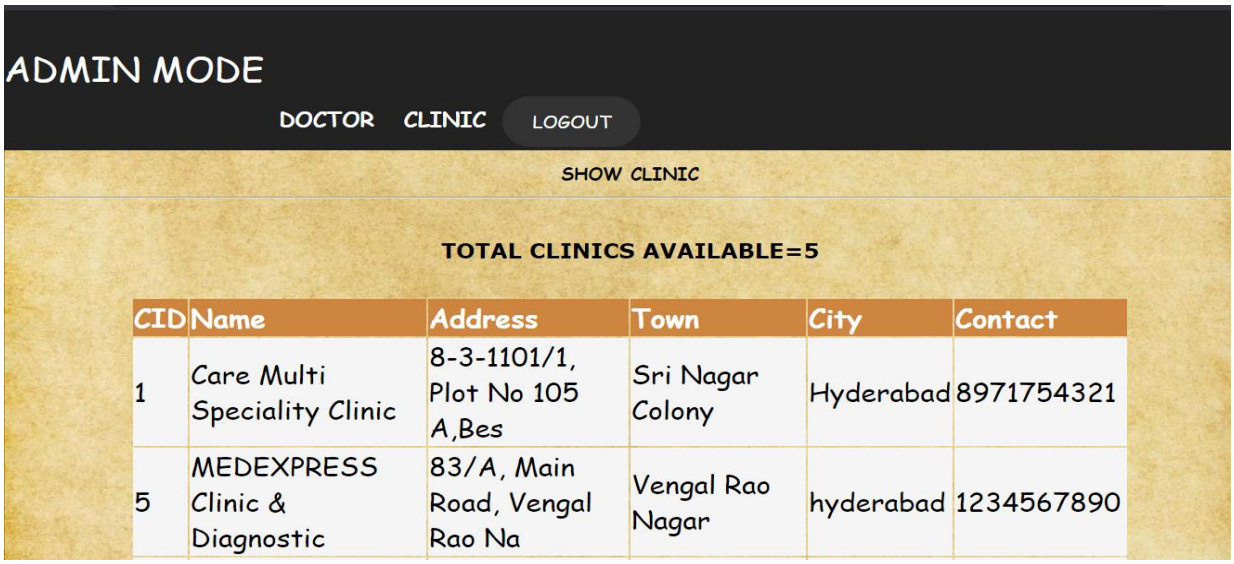


Fig 21: Delete doctor



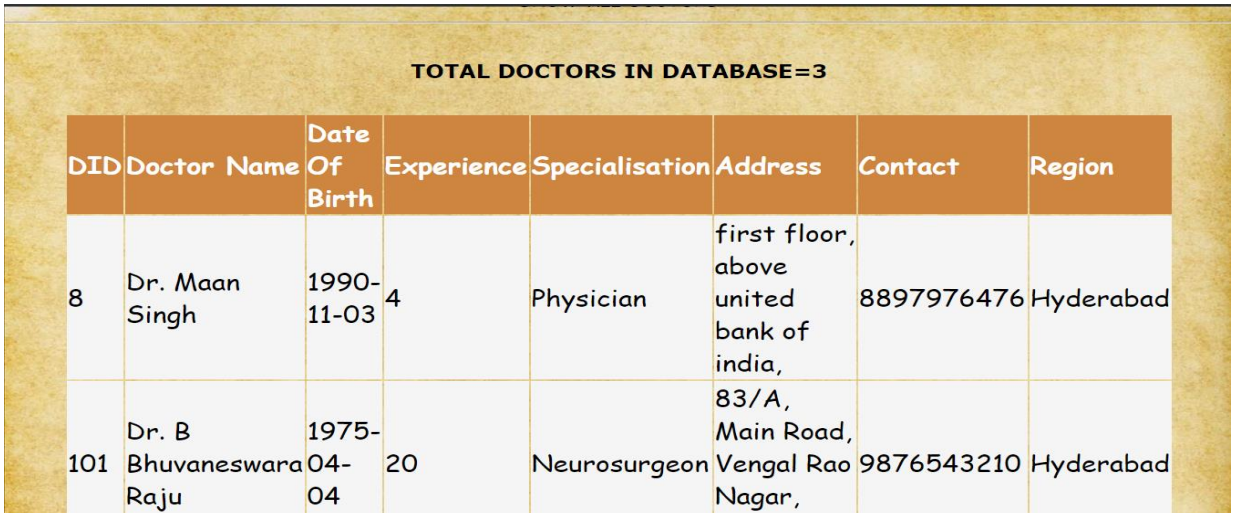
CID	Clinic Name	DID	Doctor Name	Day	Time
5	MEDEXPRESS Clinic & Diagnostic-Vengal Rao Nagar	101	B Bhuvaneswara Raju	Wednesday	08:00:00-20:00:00
5	MEDEXPRESS Clinic & Diagnostic-Vengal Rao Nagar	101	B Bhuvaneswara Raju	Friday	08:00:00-20:00:00

Fig 22: Doctor Schedules



CID	Name	Address	Town	City	Contact
1	Care Multi Speciality Clinic	8-3-1101/1, Plot No 105 A,Bes	Sri Nagar Colony	Hyderabad	8971754321
5	MEDEXPRESS Clinic & Diagnostic	83/A, Main Road, Vengal Rao Na	Vengal Rao Nagar	hyderabad	1234567890

Fig 23: Show all clinics



TOTAL DOCTORS IN DATABASE=3

DID	Doctor Name	Date Of Birth	Experience	Specialisation	Address	Contact	Region
8	Dr. Maan Singh	1990-11-03	4	Physician	first floor, above united bank of india,	8897976476	Hyderabad
101	Dr. B Bhuvaneswara Raju	1975-04-04	20	Neurosurgeon	83/A, Main Road, Vengal Rao Nagar,	9876543210	Hyderabad

Fig 24: Show all doctors

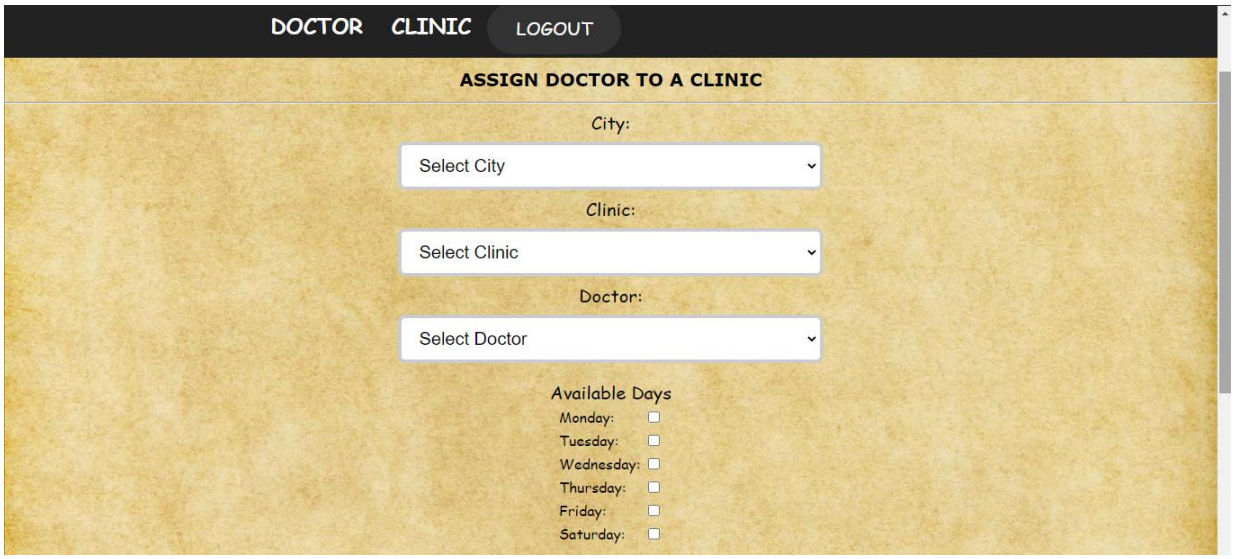


Fig 25: Assign doctor to a clinic

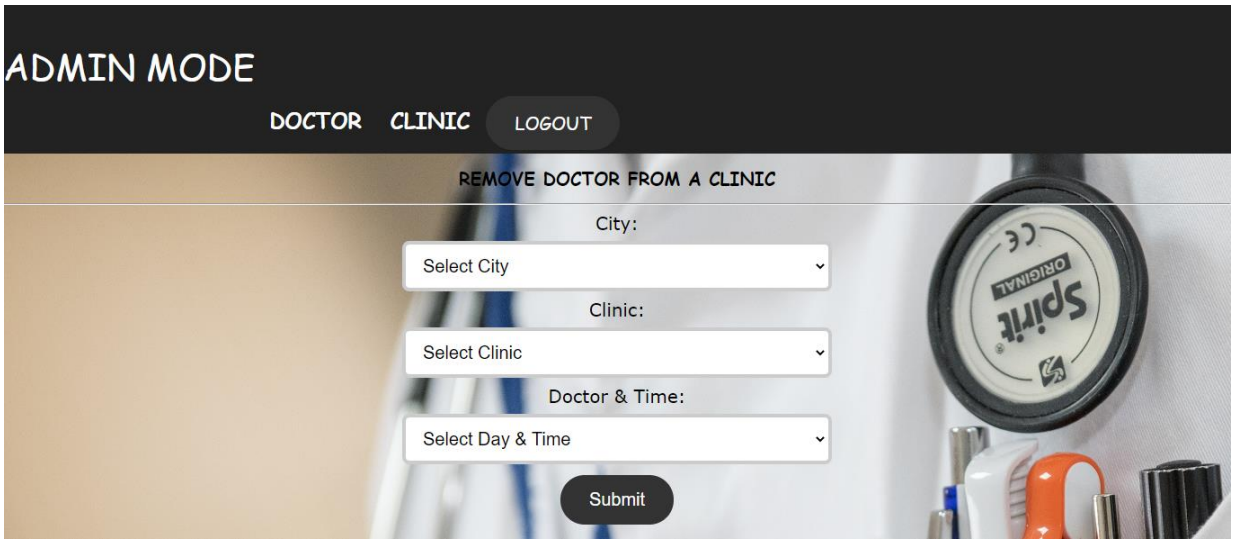


Fig 26: Delete doctor form clinic

7. CONCLUSION

One of the prime reasons that online appointment management system is gaining popularity in recent days is that the system provides an easier platform for receiving healthcare facilities to the general users. In this pandemic it is gaining popularity due to a smaller number of facilities. At the end it is concluded that we have made effort on following points:

- A description of doctor along with years of experience is provided to help the patient understand the circumstances
- Admin portal to manage appointments and to perform all the operations
- User portal to let the user book and cancel the appointments

8. FUTURE SCOPE

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can add print appointment details
- We can use more advanced languages like Angular and other good databases like postgreSQL
- We can host the platform on online servers to make it accessible worldwide
- Integrate multiple load balancers to distribute the loads of the system
- Create the master and slave database structure to reduce the overload of the database queries
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers

9. REFERENCES

- https://dev.mysql.com/doc/
- https://www.tutorialspoint.com/dbms/er_diagram_representation.html
- https://www.php.net/docs.php
- https://projectabstracts.com/6933/online-appointment-management-system.html