**Visvesvaraya Technological University**

Belagavi, Karnataka-590 018



# A MINI PROJECT REPORT

# On

**‘CLINIC APPOINTMENT MANAGEMENT SYSTEM’**

# Submitted

# In partial fulfilment requirements for the award of the Degree

# of

# BACHELOR OF ENGINEERING

# IN

# INFORMATION SCIENCE AND ENGINEERING

by

**Siddharth Kamath(4NM21IS169)**

# Vinay Kumar U(4NM21IS207)

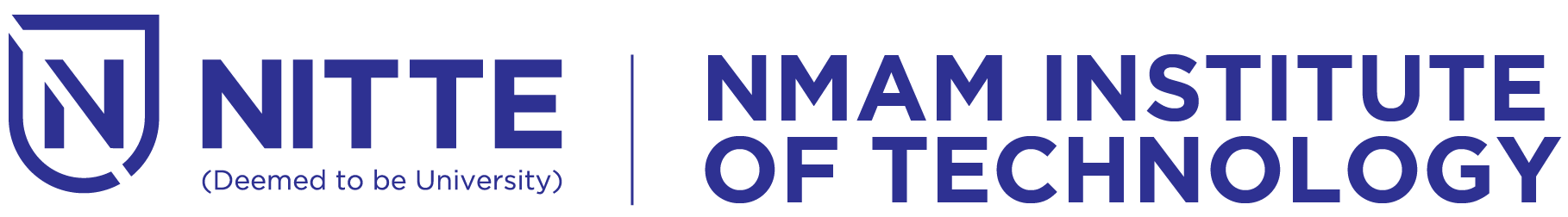
# 

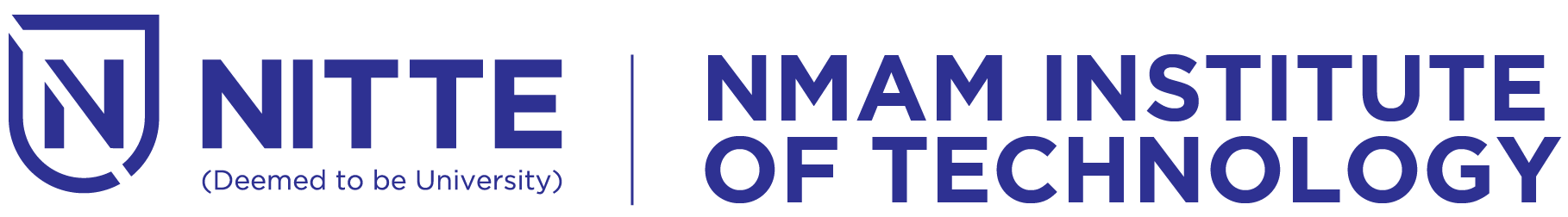
# Under the Guidance of

# Abhishek S. Rao

# Assistant Professor

**Department of Information Science and Engineering**





**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

**CERTIFICATE**

This is to certify that Mr. **Siddharth Kamath(**4NM21IS169) and Mr. **Vinay Kumar U(**4NM21IS207) has satisfactorily completed the Application Development Mini Project work entitled “**CLIENT APPOINTMENT MANAGEMENT SYSTEM**” of Third Year, Bachelor of Engineering in Information Science and Engineering at NMAMIT, Nitte in the academic year 2023 - 24.

**Project Guide Head, Dept. of ISE**

**Mr. Abhishek S. Rao Dr. Ashwini B Assistant Professor Associate Professor**

**ABSTRACT**

The "Online Appointment Management System" is a sophisticated computerized platform designed to streamline the process of scheduling and managing appointments. With an intuitive graphical user interface, the system enables customers to easily register appointments, which are subsequently verified by administrators. The software offers various user-friendly, menu-driven modules to efficiently store, update, and retrieve information.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CONTENTS** | **PAGE NO.** |
| **1. INTRODUCTION** | **5** |
| **2. PROJECT OVERVIEW** | **6** |
| **2.1. PURPOSE** | **6** |
| **2.2. SCOPE** | **6** |
| **3. DATABASE DESIGN** | **7** |
| **3.1. SOFTWARE REQUIREMENTS SPECIFICATION** | **7** |
| **3.1.1. SOFTWARE REQUIREMENTS** | **7** |
| **3.1.2. HARDWARE REQUIREMENTS** | **7** |
| **3.2. CONCEPTUAL DESIGN** | **8** |
| **3.2.1. E-R DIAGRAM** | **8** |
| **3.2.2. SCHEMA DIAGRAM** | **9** |
| **4. IMPLEMENTATION** | **10** |
| **4.1. FLOWCHART OF PROJECT WORK** | **10** |
| **4.2. WORKING METHODOLOGY** | **10** |
| **5. RESULTS** | **15** |
| **5.1-5.5. Sample Output (Snapshots)** | **18** |
| **6. CONCLUSION** | **18** |
| **7. REFERENCES** | **19** |

**1.INTRODUCTION**

1.1 OBJECTIVES:

* The main objective of the project is to design and develop a user friendly-system
* Easy to use and an efficient computerized system.
* To develop an accurate and flexible system, it will eliminate data redundancy.
* To study the functioning of the Appointments management System.
* To make a software fast in processing, with a good user interface.
* To make software with a good user interface so that users can change it and it should be used for a long time without error and maintenance.
* To provide a synchronized and centralized farmer and seller database. Computerization can be helpful as a means of saving time and money.
* To provide better Graphical User Interface (GUI).
* Less chances of information leakage.
* Provides Security to the data by using login and password methods.
* To provide immediate storage and retrieval of data and information.
* Improving arrangements for student coordination. Reducing paperwork.

1.2 LIMITATIONS:

* Time consumption in data entry as the records are to be manually maintained by faculties a lot of time.
* Lot of paperwork is involved as the records are maintained in the files and registers.
* Storage Requires as files and registers are used the storage space requirement is increased.
* Less Reliable use of papers for storing valuable data information is not at all reliable.
* Aadhar linkage with the official aadhar database has not been done.

**Chapter 2: Project Overview**

2.1. Purpose

Online Appointment Management System (OAMS) serves as a comprehensive replacement, leveraging MYSQL database for robust record storage and facilitating seamless report generation. A vital sub-module, the Appointment Audit Module, operates within the OAMS framework, allowing users to import and export appointment details. To access the web application, users undergo registration facilitated by the admin, requiring essential details such as name and username. Following admin verification, users receive an ID and password, granting them access to the OAMS.Once registered, users can efficiently manage appointments, and the system generates an appointment confirmation certificate.

2.2. Scope

1. Efficient Management: Centralize and organize data to facilitate easy access, update, and retrieval of information, thereby improving overall administrative efficiency.
2. Streamlined Academic Processing: Develop a user-friendly interface for Patients and Doctors to manage clinic allotments, doctor availability, and appointments efficiently, ensuring a smooth process.
3. Insightful Data Analysis: Implement features to capture and analyze academic and administrative data, providing valuable insights for informed decision-making and strategic planning.
4. Enhanced Operational Efficiency: Optimize administrative processes, reduce manual errors, and enhance productivity by implementing automation in tasks related to Clinic’s management and appointment processes.

**Chapter 3: Database Design**

3.1 SOFTWARE REQUIREMENTS SPECIFICATION

3.1.1 SOFTWARE REQUIREMENTS:

Frontend- HTML, CSS, Java Script, Bootstrap

Backend-JavaScript, PHP

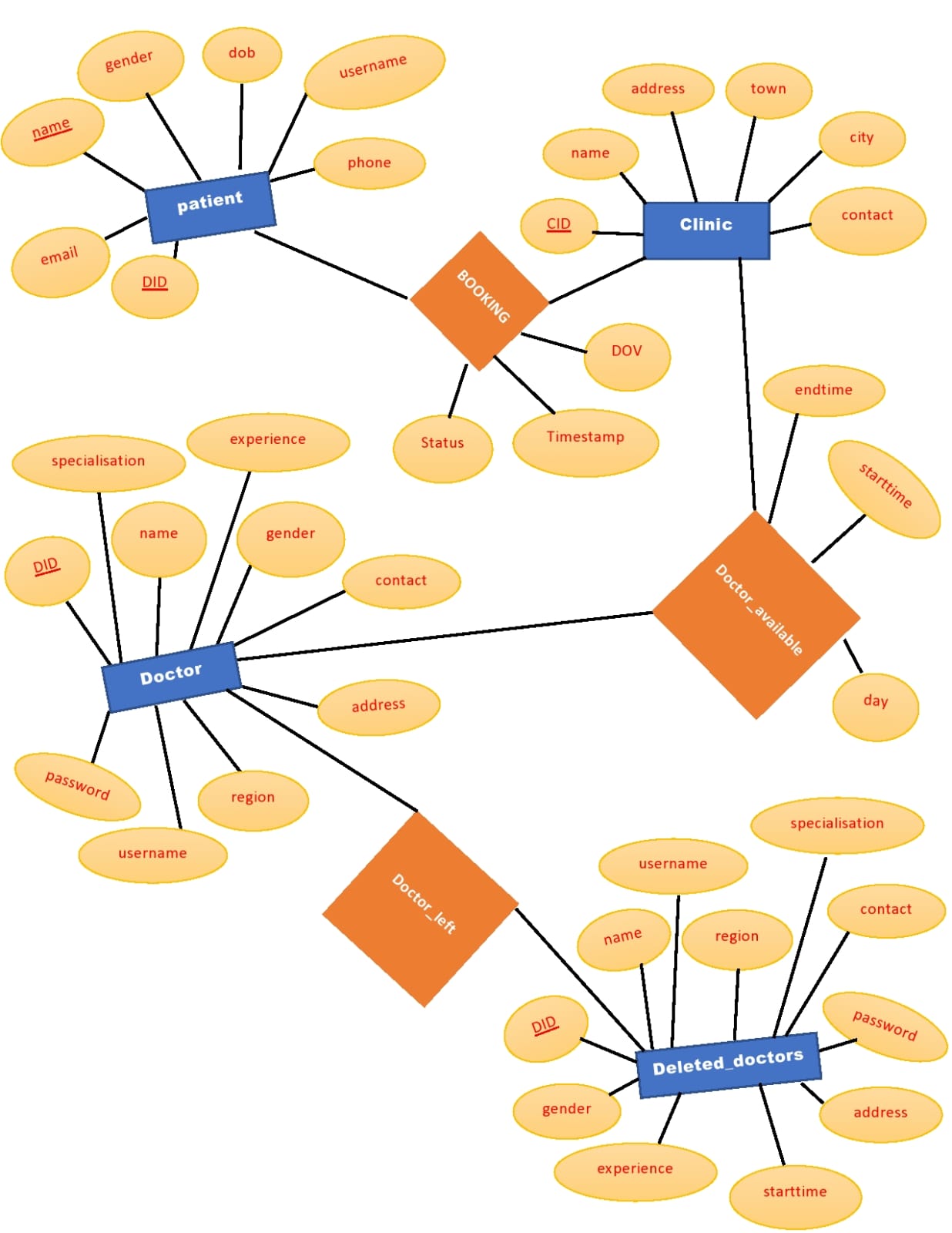
* Operating System: Windows 10 Google Chrome/Internet Explorer
* XAMPP (Version-3.7)
* Workspace editor: Sublime text 3

3.1.2. HARDWARE REQUIREMENTS:

* Computer with a 1.1 GHz or faster processor
* Minimum 2GB of RAM or more
* 2.5 GB of available hard-disk space
* 5400 RPM hard drive
* 1366 × 768 or higher-resolution display
* DVD-ROM drive

3.2 CONCEPTUAL DESIGN

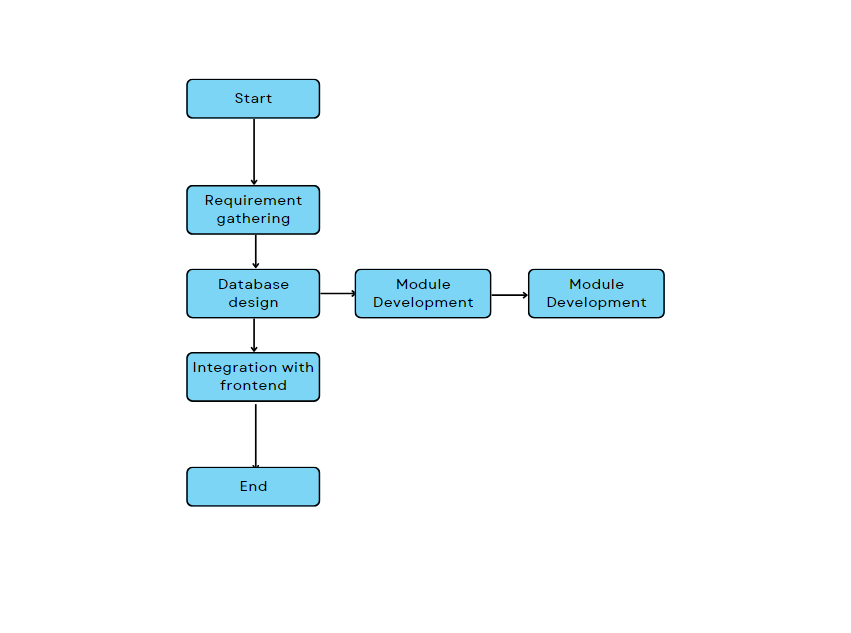
3.2.1 E-R DIAGRAM:

****

**Chapter 4: Implementation**

4.1. Flowchart of Project Work:

The flowchart for the Student Database Management System (DBMS) project outlines the sequential steps and processes involved in the system's development and functionality. Below is an illustration depicting the high-level flow of the project:



i.Flow chart for the project building Steps

4.2. Working Methodology:

An "implementation" of Python should be taken to mean a program or environment which provides support for the execution of programs written in the Python language, as represented by the CPython reference implementation.

There have been and are several distinct software packages providing what we all recognize as Python, although some of those are more like distributions or variants of some existing implementation than a completely new implementation of the language.

Back End (MySQL)

Database:

A Database Management System (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes:

* A modeling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.
* The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.
* Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory).A database query language and report Students Management System Dept. Of CSE, SVCE 2020-2021 7 writer to allow users to interactively interrogate the database, analyze its data and update it according to the users privileges on data.
* Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called sub schemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and student data.
* If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization. These controls are only available when a set of application programs are customized for each data entry and updating function.

A transaction mechanism that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses (concurrency control), and faults (fault tolerance).

* It also maintains the integrity of the data in the database.
* The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers (key fields) can be entered into the database. See ACID properties for more information (Redundancy avoidance).

When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.

SQL:

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model.

* In the relational model, data is stored in structures called relations or tables.

SQL statements are issued for the purpose of:

* Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema objects such as tables and indexes)

Create Statements

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

)

CREATE TABLE `department` (

`cid` int(11) NOT NULL,

`branch` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

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

)

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

)

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

)

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

)

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

)

Triggers

It is the special kind of stored procedure that automatically executes when an event occurs in the database.

Triggers used :

1: Trigger name: on delete

Table: Deleted\_Doctors

Time: after

Event: insert

INSERT INTO trig VALUES(null,NEW.rid,DELETED\_DOCTORS Inserted',NOW())

2: Trigger name:

on Insert Table: Doctors

Time: after

Event: delete

Definition: INSERT INTO trig VALUES(null,OLD.rid,DOCTORS',NOW())

3: Trigger name: on update

Table: Deleted\_Doctors

Time: after

Event: update

Definition: INSERT INTO trig VALUES(null,NEW.rid,UPDATED',NOW())

Stored Procedure

CREATE DEFINER=`root`@`localhost` PROCEDURE `DeletePatient`(IN `rollnumber` VARCHAR(20))

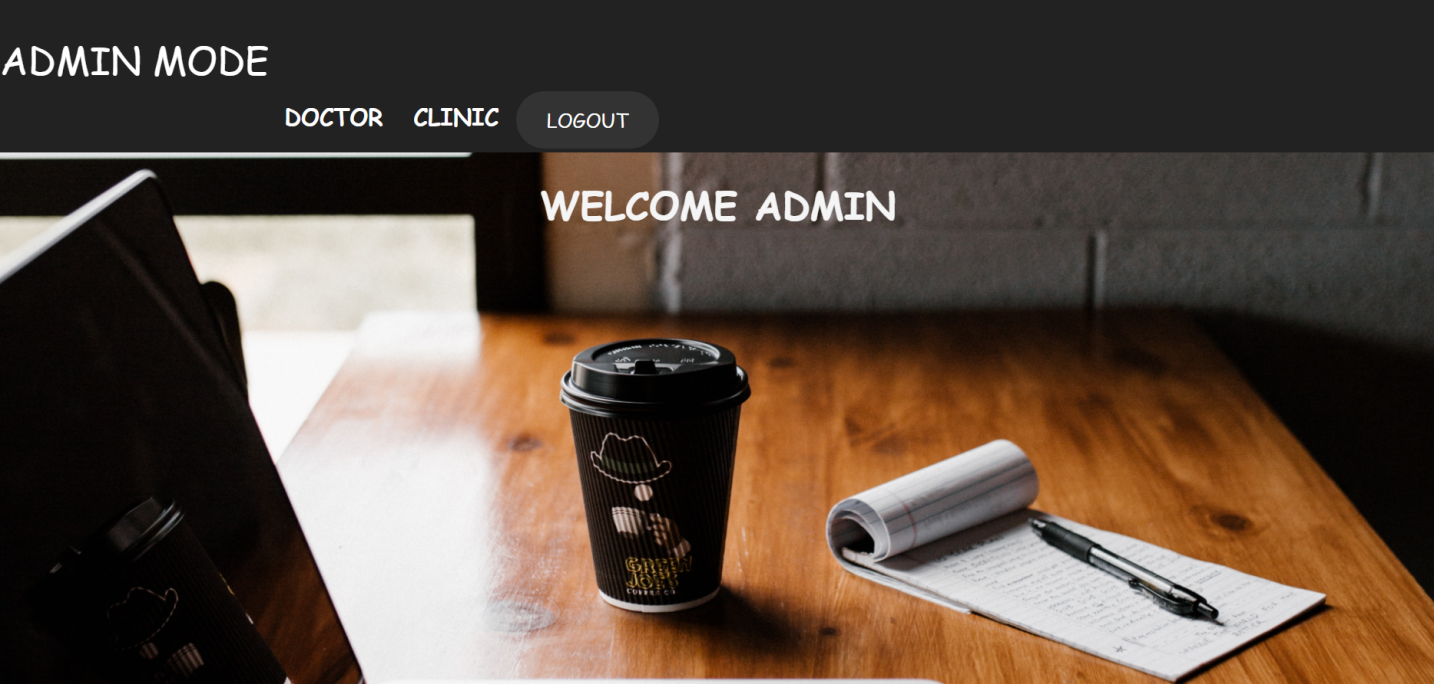
BEGIN

INSERT INTO trig VALUES (NULL, P\_ID, PATIENT DELETED', NOW());

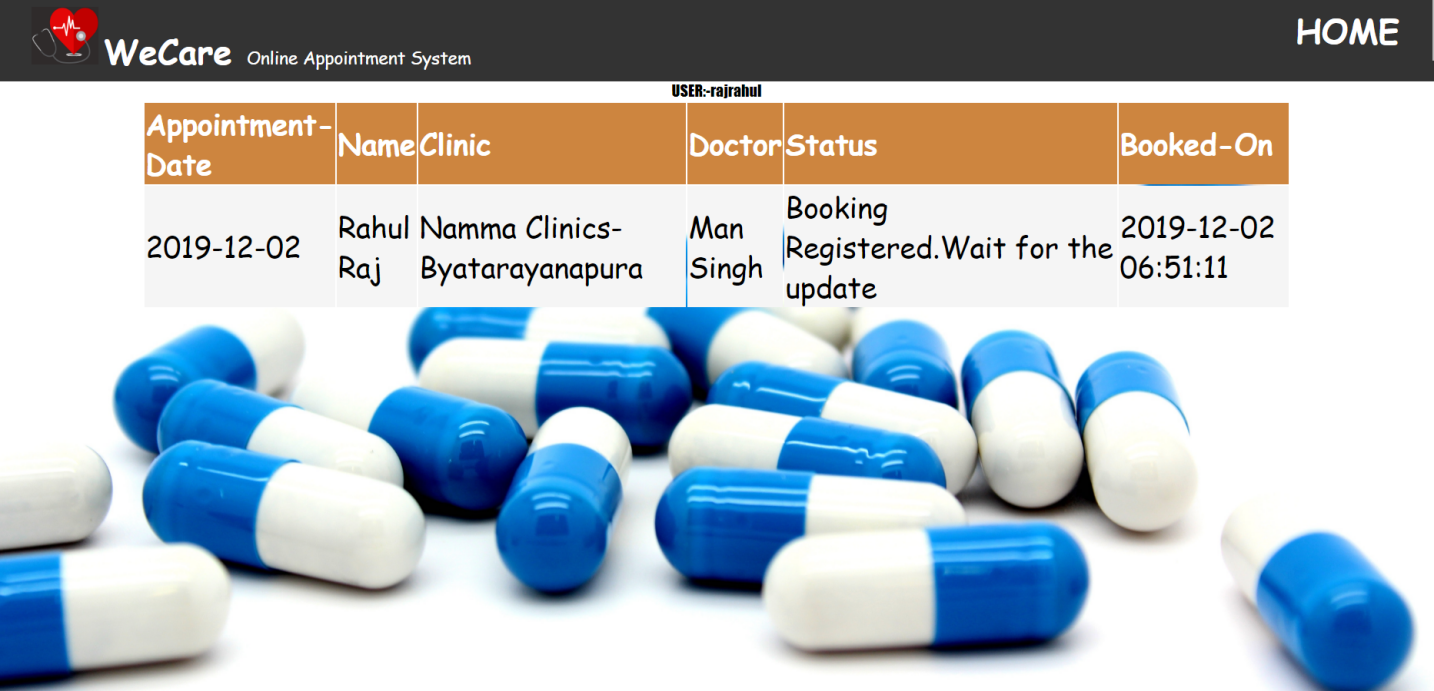
DELETE FROM student WHERE P\_ID =Patient\_ID;

END$$

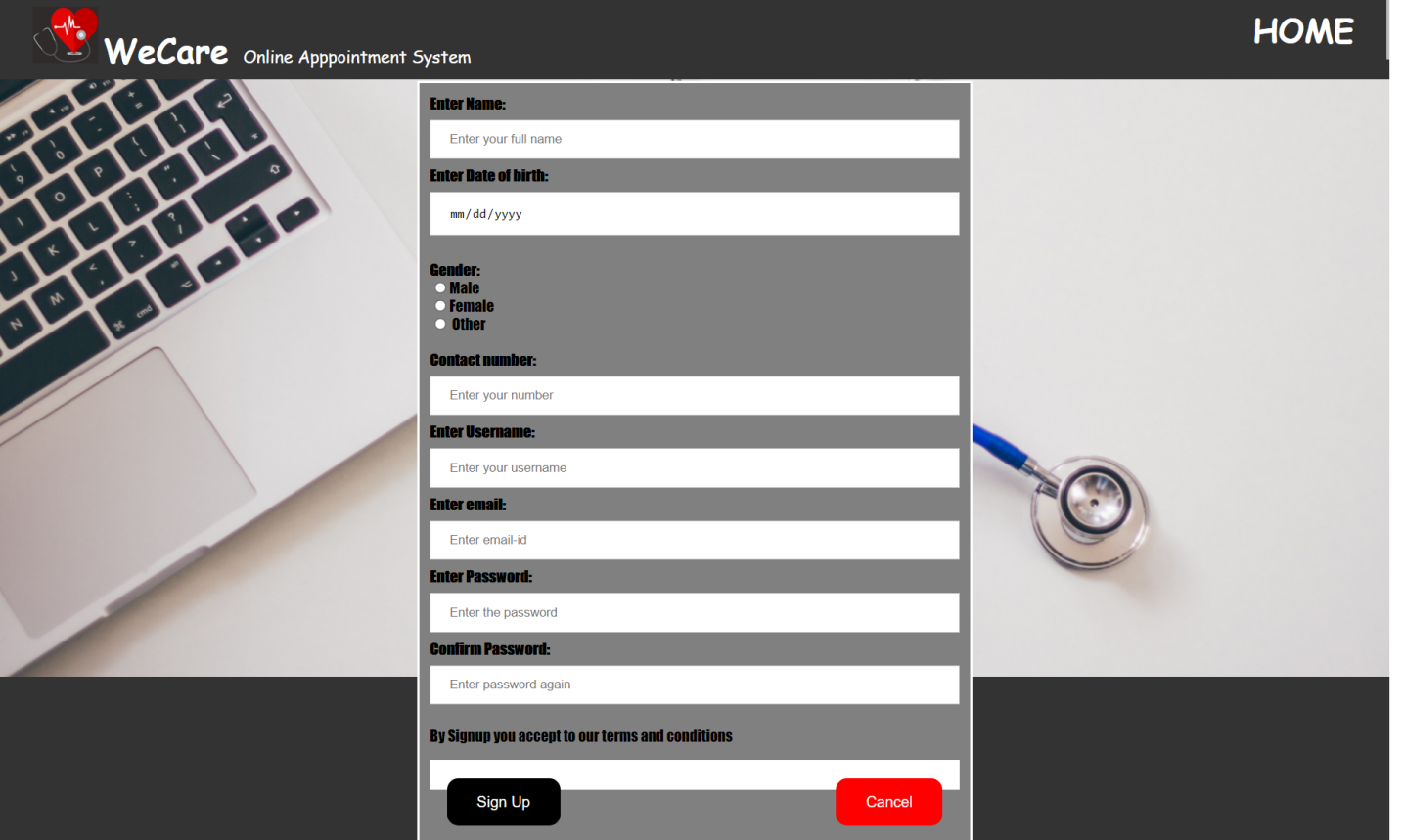
**Chapter 5: Result**

****

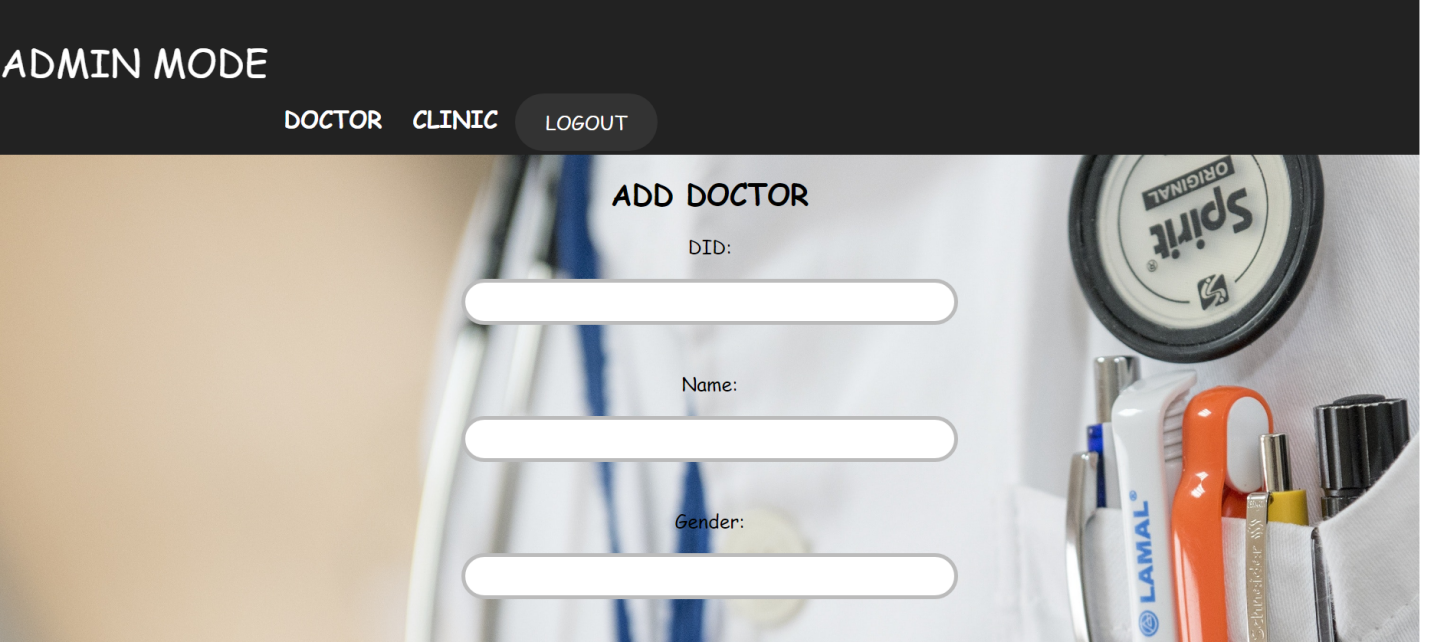
5.1. AdminPage

****

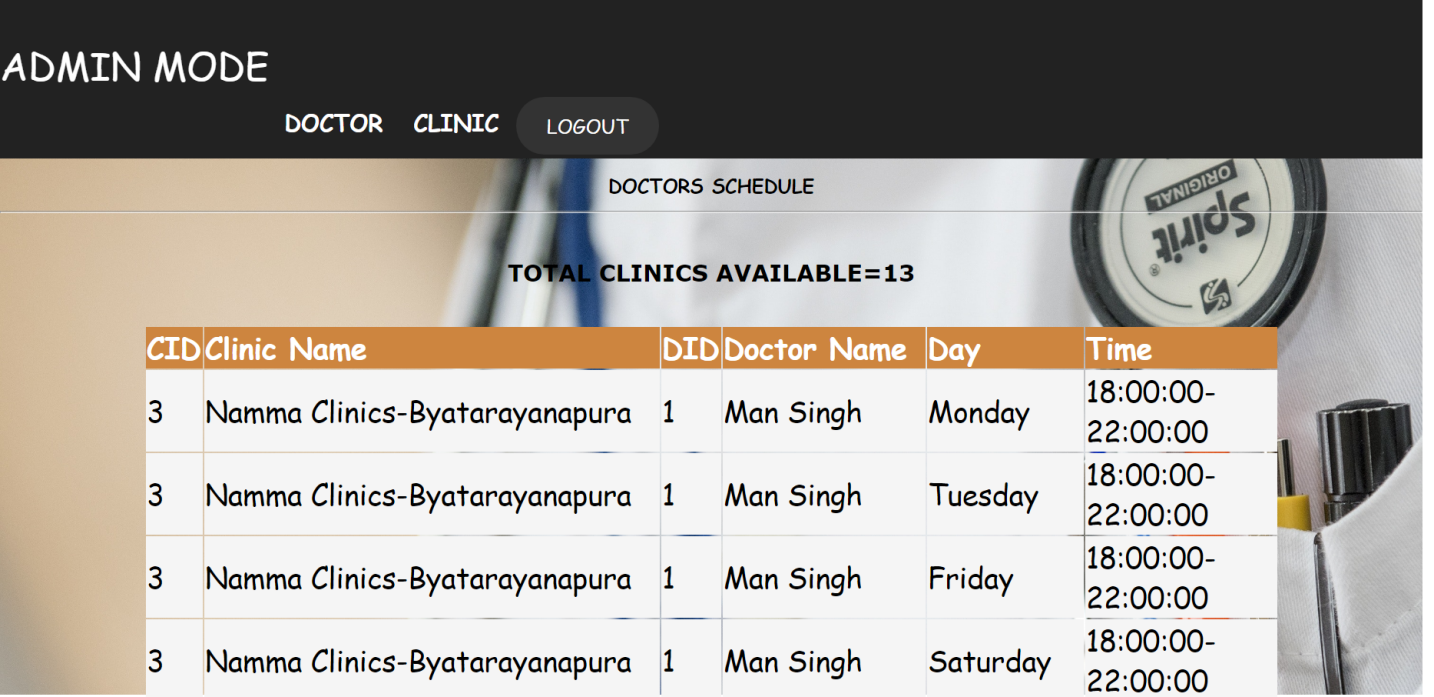
5.2. Home Page



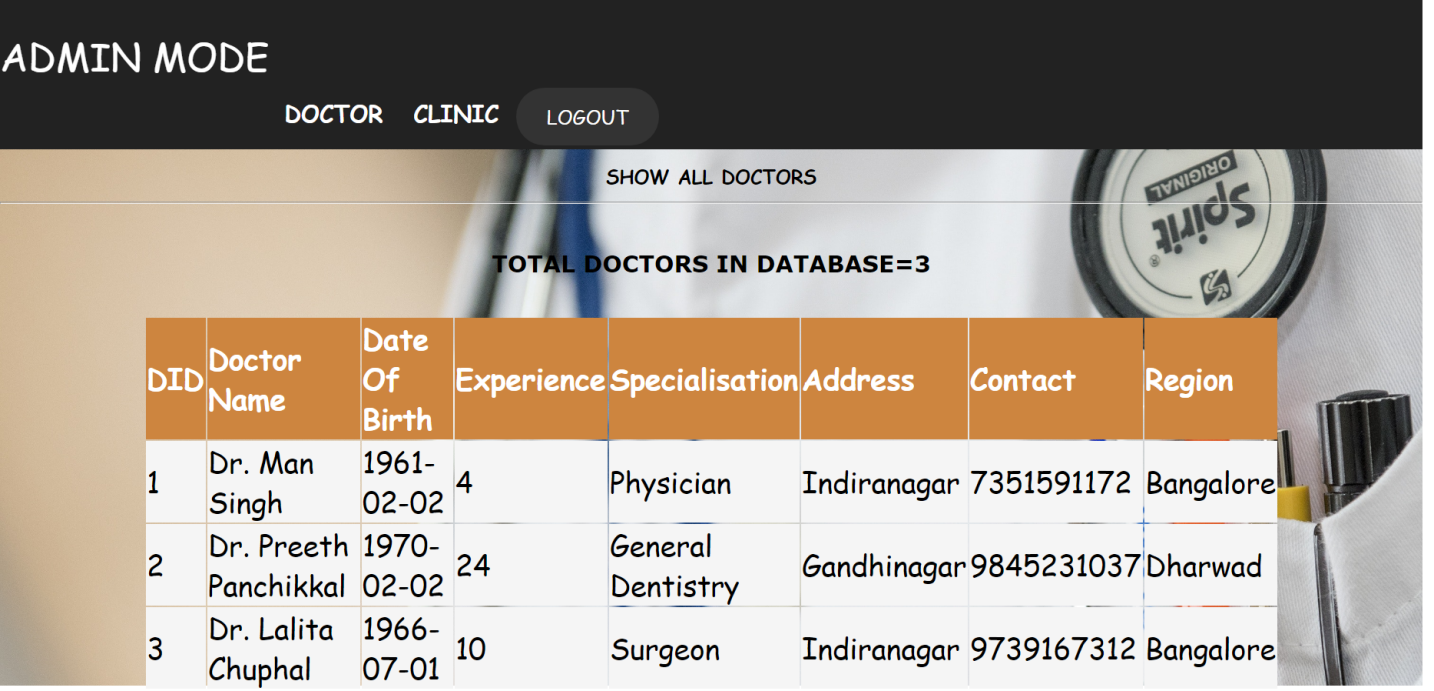
5.3. Adding Client Information



5.4. Adding Doctor’s Details



5.5. Clinic Details



5.6. DoctorsDetails

**Chapter 5: 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. During pandemic it was 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

FUTURE ENHANCEMENT

* Enhanced database storage facility
* Enhanced user friendly
* GUI more advanced results systems
* Online feedbacks forms

**References**

[1] [https://www.c-sharpcorner.com/article/using-stored-procedures-with-Asp-Net](https://www.c-sharpcorner.com/article/using-stored-procedures-with-Asp-Net/)

[2]https://www.youtube.com

[3] https://www.google.com

[4]http://www.getbootstrap.com