

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnana Sangama”, Belgaum-590018**



**A DBMS Laboratory Mini Project Report On**

**“DOCTOR’S APPOINTMENT SYSTEM”**

**SUBMITTED IN PARTIAL FULFILMENT FOR 5<sup>TH</sup> SEMESTER**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**SUBMITTED BY**

**SNEHA C M (1JB19CS131)**

**SIMRAN DUA (1JB19CS127)**

**UNDER THE GUIDANCE OF**

**MRS. RAJATHA**

**Assistant Professor**

**Dept. of CSE, SJBIT**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SJB INSTITUTE OF TECHNOLOGY**

**#67, BGS HEALTH AND EDUCATION CITY, DR VISHNUVARDHAN ROAD,  
Kengeri, BENGALURU-560060, KARNATAKA, INDIA.**

**2021 – 2022**

// Jai Sri Gurudev //

**Sri Adichunchanagiri Shikshana Trust ®**

**SJB INSTITUTE OF TECHNOLOGY**

#67, BGS HEALTH AND EDUCATION CITY, DR VISHNUVARDHAN ROAD, KENGERI,  
BENGALURU-560060, KARNATAKA, INDIA.

2021 – 2022

**Department Of Computer Science & Engineering**



**CERTIFICATE**

Certified that the DBMS Mini project work entitled **“Doctor’s Appointment System”** is a bonafide work carried out by **Ms. Sneha CM** and **Ms. Simran Dua** bearing USN **1JB18CS131, 1JB18CS127** respectively of **SJB Institute of Technology** in partial fulfilment for 5th semester in **COMPUTER SCIENCE AND ENGINEERING** of the **Visvesvaraya Technological University, Belagavi** during the academic year **2021-22** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Mini project report has been approved as it satisfies the academic requirements in respect of Mini Project prescribed by the institution.

---

**Mrs. Rajatha**  
Assistant Professor  
Dept. of CSE, SJB I.T.

---

**Dr. Krishna A.N**  
Professor and HOD  
Dept. of CSE, SJB I.T.

**1. Internal Examiner** \_\_\_\_\_

**2. External Examiner** \_\_\_\_\_



## ACKNOWLEDGEMENT

We would like to express our profound thanks to His Divine Soul **Padmabhushan Sri Sri Sri Dr. Balagangadharanatha MahaSwamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha MahaSwamiji** for providing us an opportunity to pursue my academics in this esteemed institution.

We would also like to express my profound thanks to Reverend **Sri Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this mini project in this admired institution.

We express our gratitude to **Dr. KV Mahendra Prasanth**, Principal, SJB Institute of Technology, for providing an excellent facilities and academic ambience; which have helped us in satisfactory completion of mini project work.

We extend our sincere thanks to **Dr. Krishna A.N**, Head of the Department, Computer Science and engineering; for providing an invaluable support throughout the period of mini project work.

We wish to express heartfelt gratitude to my guides, **Mrs. Rajatha**, Assistant Professor, Dept of CSE for their valuable guidance, suggestions and cheerful encouragement during the entire period of this work.

Finally, we take this opportunity to extend our earnest gratitude and respect to our parents, Teaching & Non-teaching staffs of the department, the library staff and all our friends, who have directly or indirectly supported us during the period of this mini project work.

Regards,  
**Sneha CM (1JB19CS131)**  
**Simran Dua (1JB19CS127)**

## **ABSTRACT**

As this is generic application it can be used by a wide variety of outlets (Survey and Maintenance) to automate the process of manually maintaining the records related to the subject of maintaining the details and bookings of doctor's appointment process. The main objective of this project is to design a webpage for patients to book appointment with the respective doctors.

The concerned person can directly book the appointment for with the doctor via internet. Users can use this website and there is no need to install any particular software.

The appointment booking process takes place through three steps. First the user needs to sign up to the website to be able to see the slots available for booking an appointment with a particular doctor after that he/she must login and check the slots and can book an appointment following which an email or telephonic confirmation will be reverted to the user according to the availability.

## TABLE OF CONTENTS

ACKNOWLEDGEMENT	II
ABSTRACT	III
CONTENTS	IV-V
LIST OF FIGURES	VI

CHAPTER	PARTICULARS	PAGE
1 INTRODUCTION	1.1 OVERVIEW 1.2 PROBLEM STATEMENT 1.3 DATABASE MANAGEMENT SYSTEM	1-2
2 LITERATURE SURVEY	2.1 TRADITIONAL FILE SYSTEMS 2.2 PROS AND CONS OF THE TRADITIONAL APPROACH 2.3 DOWNFALL OF TRADITIONAL MANAGEMENT SYSTEM 2.4 INTRODUCTION TO THE DATABASE MANAGEMENT SYSTEM 2.5 INDICATIVE AREAS FOR THE USE OF A DBMS 2.6 ADVANTAGES OF A DBMS 2.7 COMPONENTS OF A DBMS	3-7
3 SYSTEM REQUIREMENTS	3.1 REQUIREMENTS SPECIFICATION 3.2 SPECIFIC REQUIREMENTS 3.2.1 SOFTWARE REQUIREMENTS 3.2.2 HARDWARE REQUIREMENTS 3.2.3 TECHNOLOGY	8-9
4 SYSTEM DESIGN	4.1 SYSTEM DESIGN 4.2 FUNCTIONAL REQUIREMENTS 4.3 ENTITY RELATIONSHIP DIAGRAM	10-16

	4.4 RELATIONAL SCHEMA	
	4.5 DESCRIPTION OF TABLES	
<b>5</b>	<b>IMPLEMENTATION</b>	
	5.1 SQL	17-31
	5.2 HTML	
	5.3 XAMPP	
	5.4 CSS	
	5.5 PHP CONNECTIONS	
	5.6 MODULE AND THEIR ROLES	
	5.6.1 SIGN UP CODE SNIPPET	
	5.6.2 STYLE CSS CODE SNIPPET	
	5.6.3 HTML-PHP CODE SNIPPET	
	5.7 RESULT	
<b>6</b>	<b>TESTING</b>	
	6.1 SOFTWARE TESTING	32
	6.2 MODULE TESTING AND INTEGRATION	
	6.3 LIMITATIONS	
<b>7</b>	<b>SNAPSHOTS AND DISCUSSIONS</b>	
	7.1 SNAPSHOTS	33-40
	7.1 HOME PAGE	
	7.2 ABOUT PAGE	
	7.3 APPOINTMENT'S HOME PAGE	
	7.4 OUR DOCTOR'S PAGE	
	7.5 INSIGHTS PAGE	
	7.6 CONTACT PAGE	
	7.7 SIGN UP TAB	
	7.8 LOGIN TAB	
	7.9 BOOK AN APPOINTMENT TAB	
	7.10 SIGNUP PAGE	
	7.11 LOGIN PAGE	
	7.12 BOOK AN APPOINTMENT PAGE	
	7.13 NEWSLETTER PAGE	

## **LIST OF FIGURES**

<b>FIGURES</b>	<b>FIGURE NAME</b>	<b>PAGE</b>
1.1	DATABASE MANAGEMENT SYSTEM	2
2.1	INDICATIVE USERS OF DBMS	5
2.2	COMPONENTS OF DBMS	7
4.1	JSP ARCHITECTURE	10
4.2	SOFTWARE INTERFACES	12
4.3	ER DIAGRAM	14
4.4	SCHEMA DIAGRAM	15
7.1	HOME PAGE	33
7.2	ABOUT PAGE	33
7.3	APPOINTMENT'S HOME PAGE	34
7.4	OUR DOCTOR'S PAGE	34
7.5	INSIGHTS PAGE	35
7.6	CONTACT PAGE	35
7.7	SIGN UP TAB	36
7.8	LOGIN TAB	36
7.9	BOOK AN APPOINTMENT TAB	37
7.10	SIGN UP PAGE	37
7.11	LOGIN PAGE	38
7.12	BOOK AN APPOINTMENT PAGE	38

## CHAPTER 1

### INTRODUCTION

#### 1.1 Overview

Databases and database technology have had a major impact on the growing use of computers. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. Nowadays, this data is typically stored in mobile phones, which have their own simple database software. In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents. A database can be of any size and complexity. For example, the list of names and addresses referred to earlier may consist of only a few hundred records, each with a simple structure. On the other hand, the computerized catalogue of a large library may contain half a million entries organized under different categories. A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse. Changes to the mini world are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

#### 1.2 Problem Statement

In order to build the system, all the processes involved in managing a hospital system should be studied; System study Helps us under the problem and needs of the application. System study aims at establishing Requests for the system to be acquired, development and installed. It involves studying and analyzing the ways of an organization currently processing the data to produce information. Analyzing the problem thoroughly forms the vital part of the system study. In system analysis, prevailing situation of problem is carefully examined by breaking them into sub problems.

Problematic areas are identified and information is collected. Data gathering is essential to any analysis of requests. It is necessary that this analysis familiarizes the designer with objectives, activities and the function of the organization in which the system is to be implemented.

#### 1.3 Database Management System

A database management system (DBMS) is a computerized system 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. Defining a database involves specifying the data types, structures, and



constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalogue or dictionary; it is called meta-data. Constructing 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. Sharing a database allows multiple users and programs to access the database simultaneously.

DBMS is a software tool to organize (create, retrieve, update, and manage) data in a database. A database management system (DBMS) refers to the technology for creating and managing databases. DBMS is a software tool to organize (create, retrieve, update, and manage) data in a database. DBMS is a software tool to organize (create, retrieve, update, and manage) data in a database. Database systems are meant to handle an extensive collection of information. Management of data involves both defining structures for storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

### **Database Management System**



**FIG 1.1**

## CHAPTER 2

### LITERATURE SURVEY

#### 2.1 Traditional File Systems

In the early days of computing, data management and storage were a very new concept for organizations. The traditional approach to data handling offered a lot of the convenience of the manual approach to business processes (e.g., handwritten invoices & account statements, etc.) as well as the benefits of storing data electronically.

The traditional approach usually consisted of custom-built data processes and computer information systems tailored for a specific business function. An accounting department would have their own information system tailored to their needs, where the sales department would have an entirely separate system for their needs.

Initially, these separate systems were very simple to set up as they mostly mirrored the business process that departments had been doing for years but allowed them to do things faster with less work. However, once the systems were in use for so long, they became very difficult for individual departments to manage and rely on their data because there was no reliable system in place to enforce data standards or management.

Separate information systems for each business function also led to conflicts of interest within the company. Departments felt a great deal of ownership for the data that they collected, processed, and managed which caused many issues among company-wide collaboration and data sharing. This separation of data also led to unnecessary redundancy and a high rate of unreliable and inconsistent data.

#### 2.2 Pros and Cons of the Traditional Approach

##### Pros

###### Simple

- Matched existing business processes and functions.
- Companies were not as interested in funding complicated information systems.

##### Cons

###### Separated ownership

- Business functions had a high sense of data ownership.
- Departments unwilling to share data for fear of minimizing their superiority.

###### Unmanaged Redundancy

- Multiple instances of the same data appeared throughout various files, systems.
- Information updated in one place was not replicated to the other locations.

###### Data inconsistency

- Redundant data stored in various locations was usually never stored the same way.
- Formatting was not centrally managed.

#### Lack of Data sharing

- Same data stored in multiple locations.
- Caused unnecessary doubling of efforts for data managing.
- Doubling of work is well as excessive maintenance costs.

### **2.3 Downfall of Traditional Management**

Conceived in a relatively centralized era when software was deployed in environments, legacy database architectures fail to support an increasingly mobile world where applications are accessed anytime, anywhere. today software users want consistent improvements in usability and expect SaaS vendors to deliver new features and functionalities needed to achieve their business objective.

- Inadequate failover capabilities
- Latency issues
- Insufficient provisions during peak demands
- Increasing operational costs
- Lack of high availability at all time
- Inability to meet the demands of global markets

For all of these reasons, traditional databases are unable to deliver results in a rapid growing environment where the workload is geographically distributed across heterogeneous data centers. Upgrading to a more distributed data model is costly and complicated and your DBAs can't just sit back and give up on this situation. Hence, due to these various reasons, the downfall of the traditional system was inevitable.

### **2.4 Introduction to Database Management System**

A database management system (DBMS) refers to the technology for creating and managing databases. Basically, a DBMS is a software tool to organize (create, retrieve, update and manage) data in a database. The main aim of a DBMS is to supply a way to store and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of database. A datum is a unit of data. Meaningful data combines to form information. Hence, information is interpreted data provided with semantics. MS ACCESS is one of the most common examples of database management software. Database systems are meant to handle large collection of information. Management of data involves both defining structures for storage of information and providing mechanism that can do manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

## 2.5 DBMS Methodology and Indicative Areas

- Conceptual database design - to build the conceptual representation of the database, which has the identification of the important entities, relationships, and attributes.
- Logical database design - to convert the conceptual representation to the logical structure of the database, which includes designing the relations.
- Physical database design - to decide how the logical structure is to be physically implemented (as base relations) in the target Database Management System (DBMS).

So, to summarize we could say DBMS allows users to create their own databases as per their requirement. The term “DBMS” includes the user of the database and other application programs. It provides an interface between the data and the software application.

### Indicative Areas for the use of DBMS

Sector	Use of DBMS
BANKING	For customer information, account activities, payments, deposits, loans, etc.
AIRLINES	For reservations and schedule information.
UNIVERSITIES	For student information, course registrations, colleges and grades.
TELECOMMUNICATION	It helps to keep call records, monthly bills, maintaining balances, etc.
FINANCE	For storing information about stock, sales, and purchases of financial instruments like stocks and bonds.
SALES	Use for storing customer, product & sales information.
MANUFACTURING	It is used for the management of supply chain and for tracking production of items. Inventories status in warehouses.
HR MANAGEMENT	For information about employees, salaries, payroll, deduction, generation of pay checks, etc.

**FIG 2.1**

## **2.6 Advantages of DBMS**

Compared to the File Based Data Management System, Database Management System has many advantages over the traditional file system used in the earlier days, such as:

### **1. Reducing Data Redundancy**

The file-based data management systems contained multiple files that were stored in many different locations in a system or even across multiple systems. Because of this, there were sometimes multiple copies of the same file which led to data redundancy. This is prevented in a database as there is a single database and any change in it is reflected immediately. Because of this, there is no chance of encountering duplicate data.

### **2. Data Integrity**

Data integrity means that the data is accurate and consistent in the database. Data Integrity is very important as there are multiple databases in a DBMS. All of these databases contain data that is visible to multiple users. So, it is necessary to ensure that the data is correct and consistent in all the databases and for all the users.

### **3. Data Security**

Data Security is a vital concept in a database. Only authorized users should be allowed to access the database and their identity should be authenticated using a username and password. Unauthorized users should not be allowed to access the database under any circumstances as it violates the integrity constraints.

### **4. Privacy**

The privacy rule in a database means only the authorized users can access a database according to its privacy constraints. There are levels of database access and a user can only view the data he is allowed to. For example - In social networking sites, access constraints are different for different accounts a user may want to access.

### **5. Backup and Recovery**

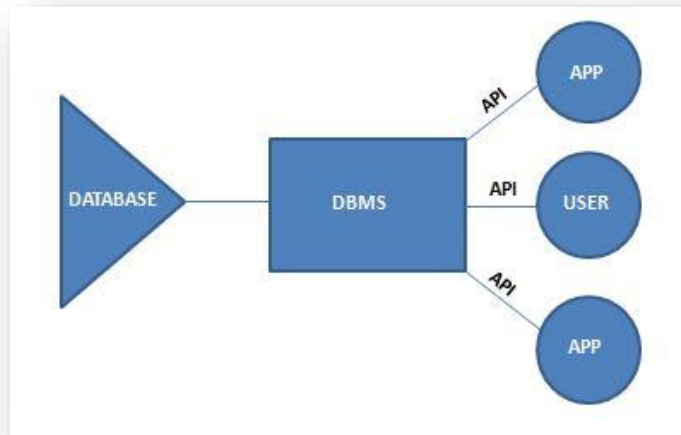
Database Management System automatically takes care of backup and recovery. The users don't need to backup data periodically because this is taken care of by the DBMS. Moreover, it also restores the database after a crash or system failure to its previous condition.

### **6. Data Consistency**

Data consistency is ensured in a database because there is no data redundancy. All data appears consistently across the database and the data is same for all the users viewing the database. Moreover, any changes made to the database are immediately reflected to all the users and there is no data inconsistency.

## 2.7 Components of DBMS

FIG 2.2



## CHAPTER 3

### SYSTEM REQUIREMENTS

#### 3.1 Requirements Specification

The production ready software is meant to run on a variety of verified hardware and software. As such, many of the required dependencies are available cross platform, both for the front end as well as the backend. Some of the verified software and hardware are specified below, along with software and hardware that are supposed to be compatible.

##### Overall Description

A reliable and scalable database driven web application with security features that is easy to use and maintain is the requisite.

#### 3.2 Specific Requirements

The specific requirements of the DOCTOR'S APPOINTMENT System are stated as follows:

##### 3.2.1 Software Requirements

- Technology Implemented: Xampp Server
- Language Used: PHP
- Database: My SQL
- User Interface Design: HTML, CSS

##### 3.2.2 Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

- Processor Speed - 500 MHz or above
- Hard Disk - 20GB (approx.)
- RAM - 64MB or above
- Storage Space - Approx. 2MB
- Processor - Intel 486/Pentium processor or better.

##### 3.2.3 Technology

HTML is used for the front-end design. It provides a means to structure text-based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.

CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document.

SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation.

We require a XAMPP connection between the front end and back-end components to write to the database and fetch required data. PHP is a general-purpose scripting language especially suited to web development and it can be embedded into HTML. A php page consists of HTML tags and php tags

PHP is highly flexible whether it is during an ongoing project or after completing the project. We require a PHP connection between the front end and back-end components to write to the database and fetch required data



## CHAPTER 4

### DETAILED DESIGN

#### 4.1 System Design

The web server needs a JSP engine, i.e., a container to process JSP pages. The JSP container is responsible for intercepting requests for JSP pages. A JSP container works with the Web server to provide the runtime environment and other services a JSP needs. It knows how to understand the special elements that are part of JSPs. This server will act as a mediator between the client browser and a database.

The following diagram shows the JSP architecture.

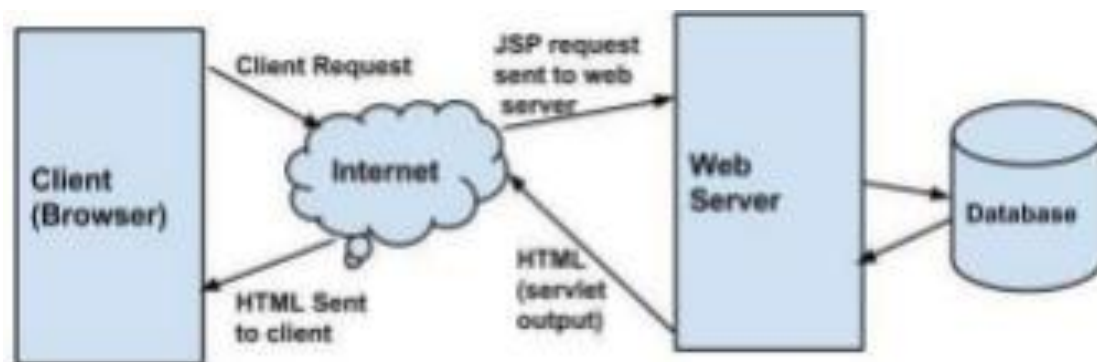


FIG 4.1

#### JSP Architecture

Three-tier Client / Server database architecture is commonly used architecture for web applications. Intermediate layer called Application server or Web Server stores the web connectivity software and the business logic (constraints) part of application used to access the right amount of data from the database server. This layer acts like medium for sending partially processed data between the database server and the client. Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users. Several types of databases, including relational or multimedia, may be created. Additionally, database architects may use one of several languages to create databases, such as structured query language.

#### 4.2 Functional Requirements

##### 1 -Distributed Database

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network

##### 2 -Client/Server System

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

### **3 -User Interfaces**

- Front-end software: Python 3.7
- Back-end software: Oracle SQL

### **4 -Hardware Interfaces**

- Windows.
- A browser which supports PHP

### **5 -Communication Interfaces**

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking etc.

### **6 -Security Requirements**

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

### **7 -Software Interfaces**

Following are the software used for the doctor's appointment system.

## Software Interfaces

Software used	Description
Operating system	We have chosen Windows operating system for its best support and user-friendliness.
Database	To save the customer records, tourist places record we have chosen oracle SQL database.
Python 3.7	To implement the project, we have chosen Python language and tkinter library for its more interactive support.

**FIG 4.2**

## Constraints

Mainly Constraints on the relational database are of 4 types:

- Domain constraints
- Key constraints
- Entity Integrity constraints
- Referential integrity constraints

### 1. Domain constraints

- Every domain must contain atomic values (smallest indivisible units) it means composite and multi-valued attributes are not allowed.
- We perform datatype check here, which means when we assign a data type to a column, we limit the values that it can contain. E.g., If we assign the datatype of attribute age as int, we can't give it values other than int datatype.

### 2. Key Constraints or Uniqueness Constraints

- These are called uniqueness constraints since it ensures that every tuple in the relation should be unique.
- A relation can have multiple keys or candidate keys (minimal super key), out of which we choose one of the keys as primary key, we don't have any restriction on choosing the primary key out of candidate keys, but it is suggested to go with the candidate key with a smaller number of attributes.
- Null values are not allowed in the primary key, hence Not Null constraint is also a part of key constraint.

### 3. Entity Integrity Constraints

- Entity Integrity constraints says that no primary key can take NULL value, since using primary key we identify each tuple uniquely in a relation.

### 4. Referential Integrity Constraints

- The Referential integrity constraints is specified between two relations or tables and used to maintain the consistency among the tuples in two relations.
- This constraint is enforced through foreign key, when an attribute in the foreign key of relation R1 have the same domain(s) as the primary key of relation R2, then the foreign key of R1 is said to reference or refer to the primary key of relation R2.
- The values of the foreign key in a tuple of relation R1 can either take the values of the primary key for some tuple in relation R2, or can take NULL values, but can't be empty.

#### 4.3 Entity Relationship Diagram

An entity-relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

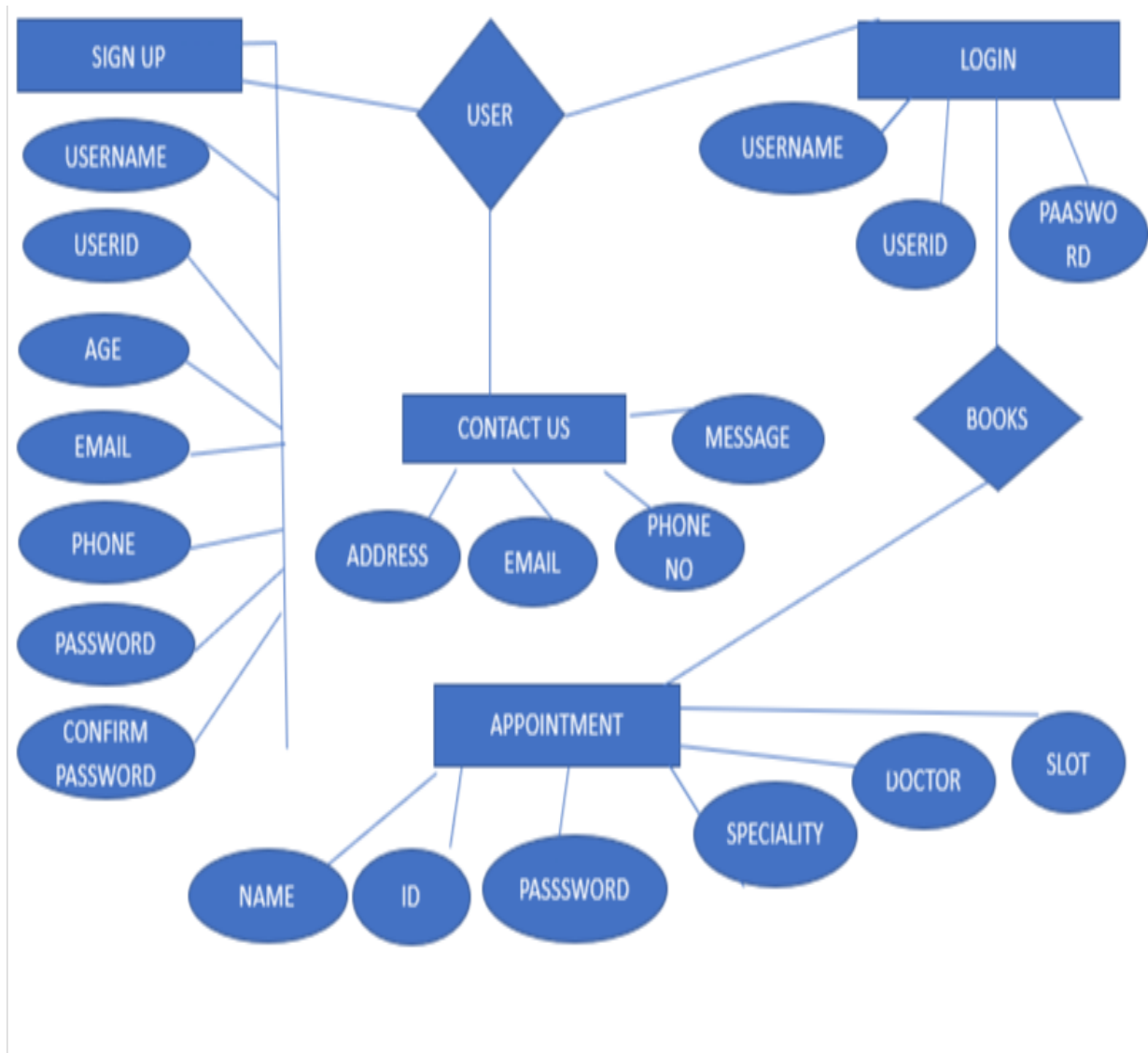
Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three-schema approach to software engineering. While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER Diagram is unlikely to be helpful on its own in integrating data into a pre-existing information system.

Cardinality notations define the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional.

## Entity Relationship Diagram



**FIG 4.3**

### 4.4 Relational Schema

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organize values in the database. The following diagram shows the schema diagram for the database.

**USER**

USERNAME	EMAIL	PHONE NUMBER
----------	-------	--------------

**SIGN UP**

USERNAME	USERID	AGE	EMAIL	PHONE	PASSWORD	CONFIRM PASSWORD
----------	--------	-----	-------	-------	----------	---------------------

**LOGIN**

USERNAME	USERID	PASSWORD
----------	--------	----------

**APPOINTMENT**

NAME	ID	PASSWORD	SPECIALITY	DOCTOR	SLOT
------	----	----------	------------	--------	------

**CONTACT US**

ADDRESS	EMAIL	PHONE.NO	MESSAGE
---------	-------	----------	---------

**FIG 4.4**

## 4.5 Description of the Tables

The database consists of four tables they are as described below

### 1. USER

A person approaching the website for the service.

**Username:** respected username is added.

**Email:** email is added.

**Phone Number:** phone number is added.

### 2. SIGN UP TABLE

Stores data of user trying to register on the website to book an appointment.

**Username:** It consists of the name given by the user.

**Userid:** The id taken is unique and is a numeric digit.

**Age:** It consists of the age of the user.

**Email:** gives the email of the user.

**Phone Number:** gives the phone number of the user.

**Password:** password that the user enters while signing up.

**Confirm password:** the user has to repeat the password entered.

### 3. LOGIN TABLE

Stores the data of the user who logs in to book an appointment.

**Username:** the username given by user during signup.

**Userid:** the userid corresponding to the given username.

**Password:** the password for the id and username.

### 4. APPOINTMENT TABLE

Stores the appointment details for the appointment to be booked.

**Name:** consist of the name of the patient for whom appointment is booked.

**Id:** gives the userid of the registered during booking an appointment.

**Password:** password of the user account.

**Specialty:** stores the data about the specialty of the doctor.

**Doctor:** stores the name of the doctor.

**Slot:** stores the slot the user chooses.

### 5. CONTACT US TABLE

For feedback/queries to be sent to the management system.

**Address:** store the address of the hospital.

**Email:** the registered email of the hospital.

**Phone Number:** gives the phone no. Of hospital for queries.

**Message:** the message or query the user needs to send.

## CHAPTER 5

### IMPLEMENTATION

#### 5.1 SQL

SQL is a database computer language designed for the retrieval and management of data in a relational database. SQL stands for Structured Query Language. SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database. SQL is the standard language for Relational Database System

##### **Applications of SQL**

As mentioned before, SQL is one of the most widely used query language over the databases. We are going to list few of them here:

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

#### 5.2 HTML

HTML stands for Hypertext Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g., HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text. HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and current major version of the FIT MI. standard.

It was published in October 2014 by the World Wide Web Consortium(W3C) to improve the language with support for the latest multimedia, while keeping it both easily readable by humans and consistently understood by computers and devices such as web browsers, parsers. etc.HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

HTML5 includes detailed processing models to encourage more interoperable implementations:it extends, improves and rationalizes the markup available for documents and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a candidate for cross-platform mobile applications because it includes features designed with low powered devices in mind.



Many new syntactic features are included. To natively include and handle multimedia and graphical content, the new <video>, <audio> and <canvas> elements were added, and support for scalable vector graphics (SVG) content and MathML for mathematical formulas. To enrich the semantic content of documents, new page structure elements such as <main>, <section>, <article>, <header>, <footer>, <aside>, <nav> and <figure>, are added. New attributes are introduced, some elements and attributes have been removed, and others such as <a>, <cite> and <menu> have been changed, redefined or standardized. The APIs and Document Object Model (DOM) are now fundamental parts of the HTML5 specification and HTML5 also better defines the processing for any invalid documents.

### **Features of HTML**

- It is easy to learn and easy to use.
- It is platform-independent.
- Images, videos, and audio can be added to a web page.
- Hypertext can be added to the text.
- It is a markup language.

## **5.3 XAMPP**

XAMPP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and MySQL. XAMPP is available as a free download in two specific packages: full and lite. While the full package download provides a wide array of development tools, XAMPP Lite contains the necessary technologies that meet the Ontario Skills Competition standard.

## **5.4 CSS**

CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.

What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

CSS Syntax

A CSS rule has two main parts: a selector, and one or more declarations:

The selector is normally the HTML element you want to style.

Each declaration consists of a property and a value

The property is the style attribute you want to change. Each property has a value

## 5.5 JAVA SCRIPT

**JavaScript** often abbreviated as **JS**, is a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. The majority of websites employ it and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine. Each of the many JavaScript engines represent a different implementation of JavaScript, all based on the ECMAScript specification, with some engines not supporting the spec fully, and with many engines supporting additional features beyond ECMA.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded. Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

## 5.5 PHP

PHP is a widely used open-source general purpose scripting language that is especially suited for web development and can be embedded into HTML basically, a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable.

The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications. PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the

PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

## 5.6 MODULES AND THEIR ROLES

### 5.6.1 SIGN UP CODE SNIPPET

```
<!DOCTYPE html>
<html>
<style>

body { font-family: 'Lato', sans-serif;}
* {box-sizing: border-box;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: lightgrey;
}

/* Add a background color when the inputs get focus */
input[type=text]:focus, input[type=password]:focus {
  background-color: lightgrey;
  outline: none;
}

/* Set a style for all buttons */
button {
  background-color: darkgrey;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}

button:hover {
  opacity: 1;
```

```
}

/* Extra styles for the cancel button */
.cancelbtn {
  padding: 14px 20px;
  background-color: darkgrey;
}

/* Float cancel and signup buttons and add an equal width */
.cancelbtn, .signupbtn {
  float: left;
  width: 50%;
}

/* Add padding to container elements */
.container {
  padding: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: darkgrey;
  padding-top: 50px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* Style the horizontal ruler */
hr {
```

```
border: 1px solid #f1f1f1;
margin-bottom: 25px;
}
```

```
/* The Close Button (x) */
```

```
.close {
  position: absolute;
  right: 35px;
  top: 15px;
  font-size: 40px;
  font-weight: bold;
  color: #f1f1f1;
}
```

```
.close:hover,
.close:focus {
  color: darkgray;
  cursor: pointer;
}
```

```
/* Clear floats */
```

```
.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
```

```
/* Change styles for cancel button and signup button on extra small screens */
```

```
@media screen and (max-width: 300px) {
  .cancelbtn, .signupbtn {
    width: 100%;
  }
}
```

```
</style>
```

```
<body>
```

```
<h2>SignUp Form</h2>
```

```
<button onclick="document.getElementById('id01').style.display='block'" style="width:
140px;;">Sign Up</button>
```

```
<div id="id01" class="modal">
```

```
  <span onclick="document.getElementById('id01').style.display='none'" class="close"
title="Close Modal">&times;</span>
```

```
<form class="modal-content" action="login.php" method="post">
  <div class="container">
    <h1>Sign Up</h1>
    <p>Please fill in this form to create an account.</p>
    <hr>
    <label for="Name"><b>Username</b></label>
    <input type="text" placeholder="Eg:Jon " name="username" required>

    <label for="Name"><b>Userid</b></label>
    <input type="text" placeholder="Eg:123 " name="userid" required>

    <label for="Age"><b>Age</b></label>
    <input type="text" placeholder="Eg:25yrs" name="age" required>

    <label for="Email"><b>Email</b></label>
    <input type="text" placeholder="Eg:abc@xyz.com" name="email" required>

    <label for="Phone"><b>Phone</b></label>
    <input type="text" placeholder="Eg:89779-27296" name="phone" required>

    <label for="psw"><b>Password</b></label>
    <input type="password" placeholder="" name="password" required>

    <label for="psw-repeat"><b>Repeat Password</b></label>
    <input type="password" placeholder="" name="rep_password" required>

    <label>
      <input type="checkbox" checked="checked" name="remember" style="margin-
bottom:15px"> Remember me
    </label>

    <p>By creating an account you agree to our <a href="#" style="color:darkgray">Terms
& Privacy</a>.</p>

    <div class="clearfix">
      <button type="button" onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
      <button type="submit" class="signupbtn">Sign Up</button>
    </div>
  </div>
</form>
</div>
```

```
<script>
// Get the modal
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
  }
}
</script>

<?php
$con = mysqli_connect("localhost","root","");
if (!$con)
{
  die('Could not connect: ' . mysqli_error());
}
mysqli_select_db($con, 'doctor');

if($_SERVER['REQUEST_METHOD'] === 'POST'){

  if(isset($_POST['submit'])){

    $sql="INSERT INTO signup1 (username, userid, age, email, phone, password,
rep_password)
VALUES"
($_POST['username'],$_POST['userid'],$_POST['age'],$_POST['email'],$_POST['phone'],$_
POST['password'],$_POST['rep_password']);

    if (!mysqli_query($con, $sql))
    {
      die('Error: ' . mysqli_error());
    }
    echo "<script>alert('Thank you for Signing up.')</script>";

  }
}
mysqli_close($con);
?>
</body>
</html>
```

### 5.6.2 STYLE CSS CODE SNIPPET

```
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Lato', sans-serif;
}

table {
  font-family: 'Lato', sans-serif;
  border-collapse: collapse;
  width: 100%;
}

td, th {
  border: 1px solid #dddddd;
  text-align: left;
  padding: 8px;
}

tr:nth-child(even) {
  background-color: #dddddd;
}

/* Bordered form */
form {
  border: 3px solid #f1f1f1;
}

/* Full-width inputs */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
```



```
color: white;
padding: 14px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 100%;
}

/* Add a hover effect for buttons */
button:hover {
  opacity: 0.8;
}

/* Extra style for the cancel button (red) */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: lightskyblue;
}

/* Center the avatar image inside this container */
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
}

/* Avatar image */
img.avatar {
  width: 40%;
  border-radius: 50%;
}

/* Add padding to containers */
.container {
  padding: 16px;
}

/* The "Forgot password" text */
span.psw {
  float: right;
  padding-top: 16px;
}
```

```
/* Change styles for span and cancel button on extra small screens */
```

```
@media screen and (max-width: 300px) {
```

```
  span.psw {  
    display: block;  
    float: none;  
  }
```

```
  .cancelbtn {  
    width: 100%;  
  }  
}
```

```
.column {  
  float: left;  
  width: 33.33%;  
}
```

```
/* Clear floats after the columns */
```

```
.row:after {  
  content: "";  
  display: table;  
  clear: both;  
}
```

```
/* Slideshow container */
```

```
.slideshow-container {  
  position: relative;  
  background: #f1f1f1;  
}
```

```
/* Slides */
```

```
.mySlides {  
  display: none;  
  padding: 80px;  
  text-align: center;  
}
```

```
/* Next & previous buttons */
```

```
.prev, .next {  
  cursor: pointer;  
  position: absolute;  
  top: 50%;  
  width: auto;  
  margin-top: -30px;
```

```
padding: 16px;
color: #888;
font-weight: bold;
font-size: 20px;
border-radius: 0 3px 3px 0;
user-select: none;
}

/* Position the "next button" to the right */
.next {
  position: absolute;
  right: 0;
  border-radius: 3px 0 0 3px;
}

/* On hover, add a black background color with a little bit see-through */
.prev:hover, .next:hover {
  background-color: rgba(0,0,0,0.8);
  color: white;
}

/* The dot/bullet/indicator container */
.dot-container {
  text-align: center;
  padding: 20px;
  background: #ddd;
}

/* The dots/bullets/indicators */
.dot {
  cursor: pointer;
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}

/* Add a background color to the active dot/circle */
.active, .dot:hover {
  background-color: #717171;
}
```

```
}

/* Add an italic font style to all quotes */
q {font-style: italic;}

/* Add a blue color to the author */
.author {color: cornflowerblue;}
```

### 5.6.3 PHP CONNECTION CODE SNIPPET

```
<?php
if(isset($_POST['submit'])){
    $connection = mysqli_connect("localhost","root","");
    $db = mysqli_select_db($connection,"doctor");
    $query = "select * from login1 where username= '$_POST[username]'";
    $query_run = mysqli_query($connection,$query);
    while($row = mysqli_fetch_assoc($query_run)){
        if($row['username'] == $_POST['username']){
            if($row['password'] == $_POST['password']){
                $_SESSION['username'] = $row['username'];
                $_SESSION['userid'] = $row['userid'];
                header("Location: index.php");
            }
            else{
                echo "<script>alert('Wrong Password...');</script>";
            }
        }
    }
}
?>
```

## 5.7 RESULT

The resulting system is able to:

Verify user credentials.

Register new user via the sign-up option

Allow users who have already registered to book an appointment

Allow users to contact the concerned management in case of any queries

## CHAPTER 6

### TESTING

#### 6.1 Software Testing

Testing is the process used to help identify correctness, completeness, security and quality of developed software. This includes executing a program with the intent of finding errors. It is important to distinguish between faults and failures. Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors. It can be conducted as soon as executable software (even if partially complete) exists. Most testing occurs after system requirements have been defined and then implemented in testable programs.

#### 6.2 Module Testing and Integration

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program. It is largely white box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows implementing of parallelism into the testing process by giving the opportunity to test multiple modules simultaneously. The final integrated system too has been tested for various test cases such as duplicate entries and type mismatch.

#### 6.3 Limitations

- The screen time of the user is not recorded.
- The user cannot book an appointment without registering to the website the user needs to sign up first.
- The user cannot get the confirmation of the appointment booked immediately on the web page and the user is contacted after the appointment is booked from user end for confirmation
- There is no payment option to pay the fees of the appointment beforehand.

## CHAPTER 7

### SNAPSHOTS

#### HOME PAGE

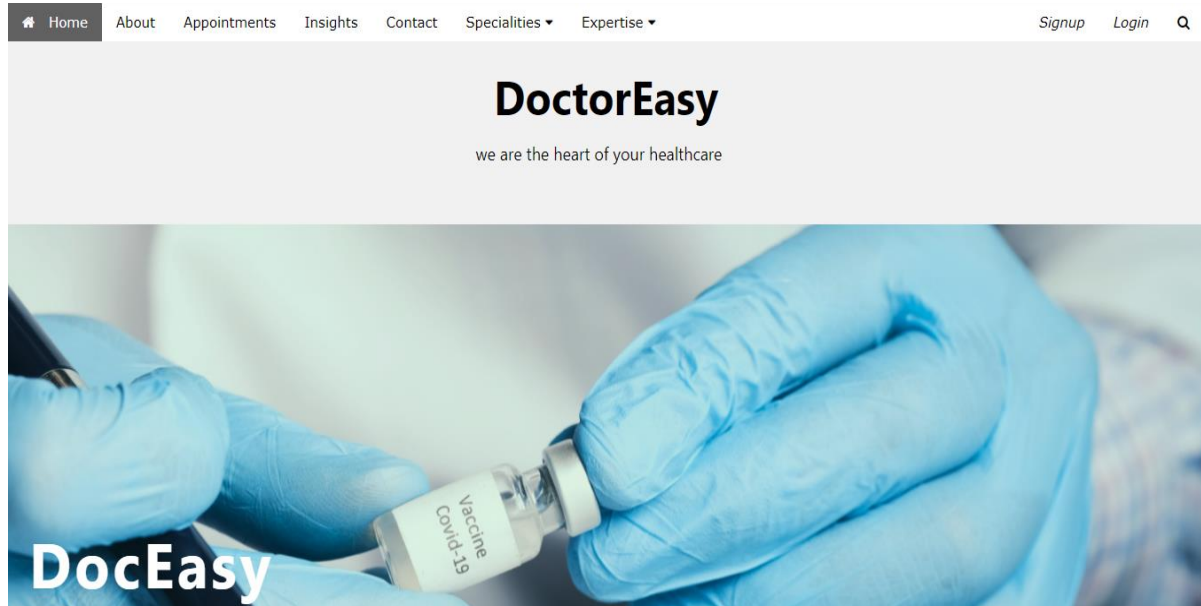


FIG 7.1

#### ABOUT PAGE

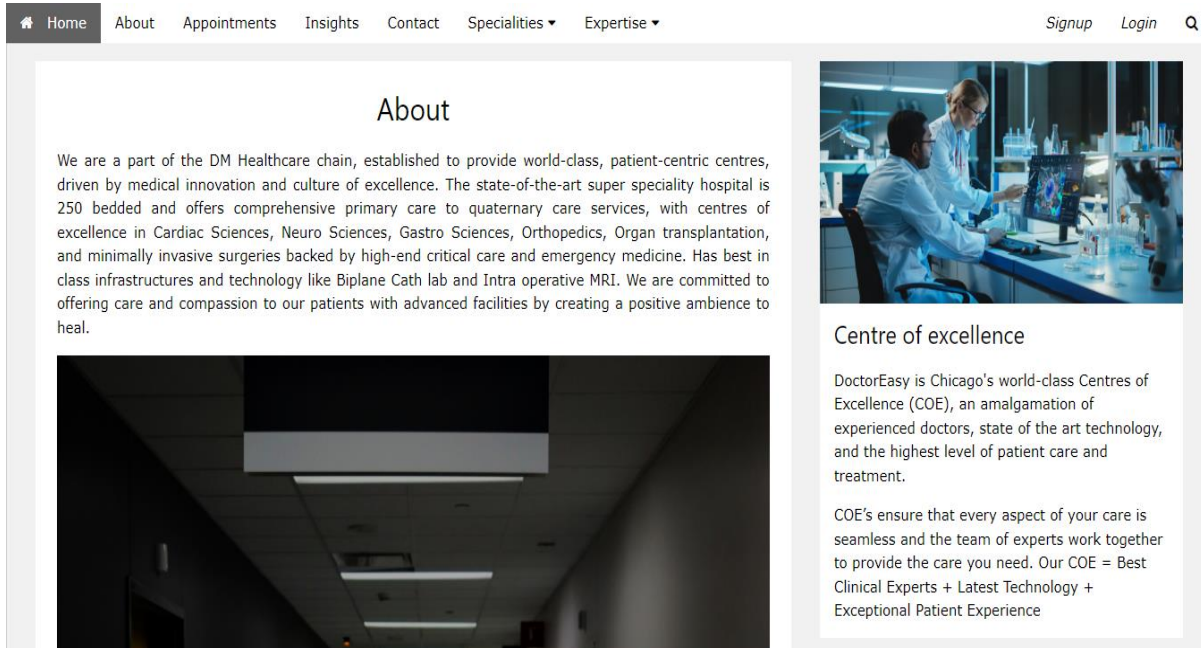


FIG 7.2

## APPOINTMENT'S HOME PAGE

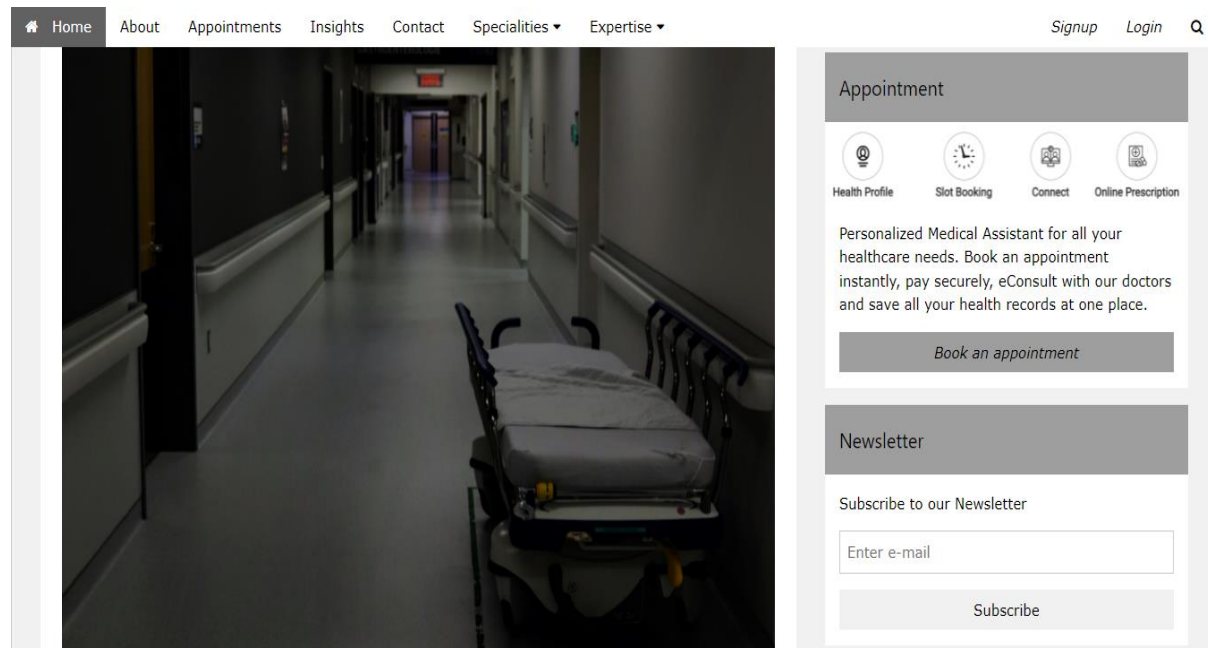


FIG 7.3

## OUR DOCTOR'S PAGE

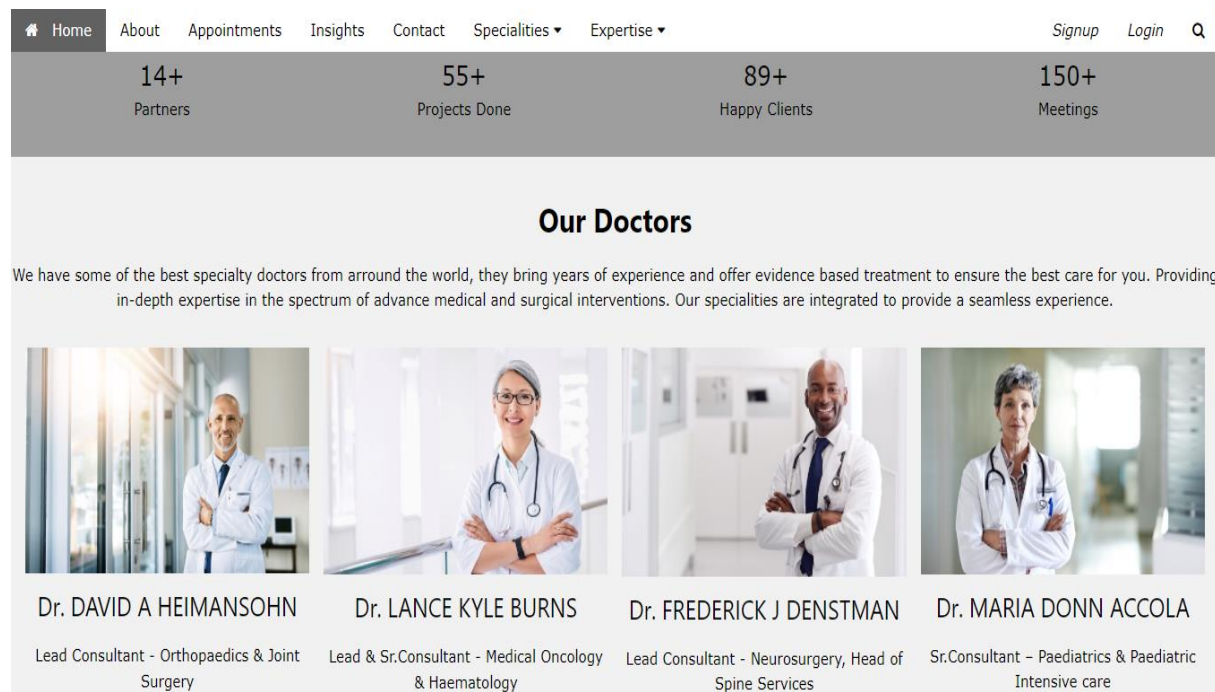


FIG 7.4




## INSIGHTS PAGE

[Home](#)
[About](#)
[Appointments](#)
[Insights](#)
[Contact](#)
[Specialities ▾](#)
[Expertise ▾](#)
[Signup](#)
[Login](#)
[Q](#)


### We provide the best care

#### Cancer Care




Specialised clinic manned by a multidisciplinary team of cancer surgeons, Plastic surgeon, and Oncology team. With state-of-the-art infrastructure and technology, we aim to deliver the best treatment options tailor made for your needs for all cancer-related ailments.

#### Insurance Partners




We are having one of the most trusted standalone health insurance partners. We have curated affordable and comprehensive healthcare policies catering to the different needs of our consumers aiming to give access to the best healthcare by providing insurance plans as per your needs.

#### Emergency Services



The Department of Emergency Medicine is equipped with world-class facilities to provide the highest quality comprehensive emergency care. We manage all forms of emergency conditions like medical, surgical, cardiac, polytrauma, paediatric and neurological cases.

#### Bundle of Joy



It encompasses prenatal and post-natal services offering specially crafted delivery packages with comprehensive care. We have a multidisciplinary team of highly experienced Gynaecologists and Obstetricians who work together to achieve safe and healthy procedures for all.

FIG 7.5

## CONTACT PAGE

[Home](#)
[About](#)
[Appointments](#)
[Insights](#)
[Contact](#)
[Specialities ▾](#)
[Expertise ▾](#)
[Signup](#)
[Login](#)
[Q](#)

### CONTACT

Connect with us

📍 Chicago, US

☎ Phone: +00 151515

✉ Email: DocEasy@gmail.com

[f](#)
[@](#)
[O](#)
[v](#)
[t](#)
[in](#)

[↑ To the top](#)

© 2019 DoctorEasy, Inc. All rights reserved.

FIG 7.6



## **SIGN UP TAB**

### **SignUp Form**

Sign Up

**FIG 7.7**

## **LOGIN TAB**

### **Login Form**

Login

**FIG 7.8**

## **BOOK AN APPOINTMENT TAB**

### **Book an appointment**

Book

**FIG 7.9**

## SIGNUP PAGE



**Sign Up**

Please fill in this form to create an account.

**Username**  
Eg:Jen

**Age**  
Eg:25yrs

**Email**  
Eg:abc@xyz.com

**Phone**  
Eg:99779-27296

**Password**

**Repeat Password**

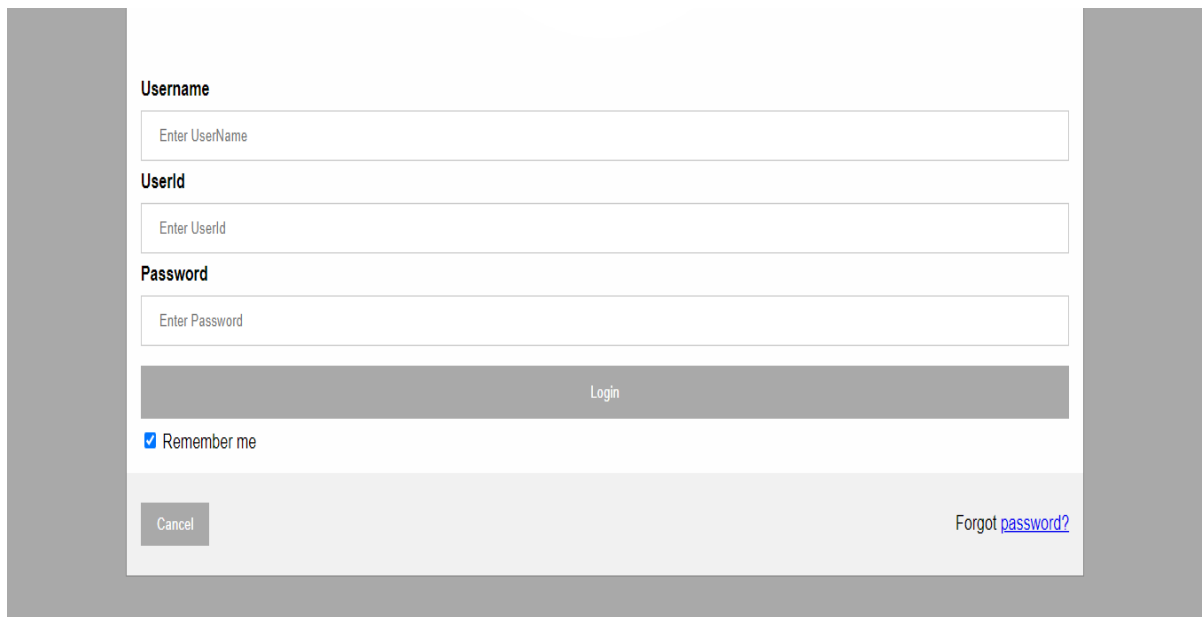
☒ Remember me

By creating an account you agree to our [Terms & Privacy](#).

Cancel Sign Up

FIG 7.10

## LOGIN PAGE



**Username**  
Enter UserName

**Userld**  
Enter Userld

**Password**  
Enter Password

Login

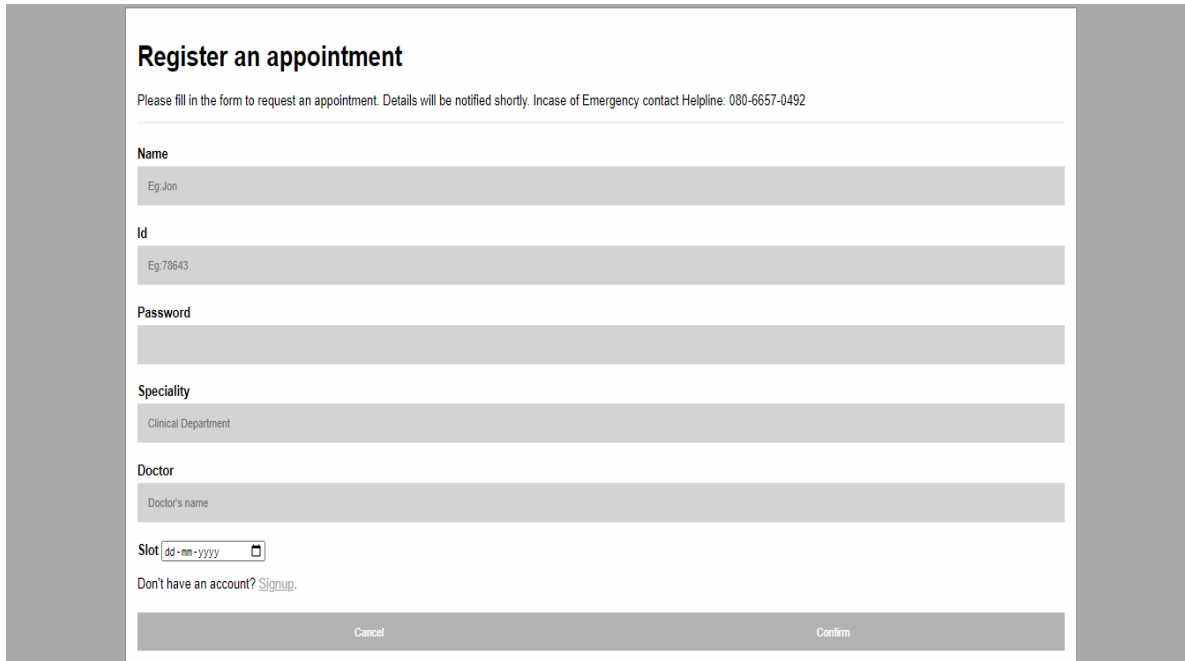
☒ Remember me

Cancel

Forgot [password?](#)

FIG 7.11

## BOOK AN APPOINTMENT PAGE



**Register an appointment**

Please fill in the form to request an appointment. Details will be notified shortly. In case of Emergency contact Helpline: 080-6657-0492

**Name**  
Eg: Jon

**Id**  
Eg: 78643

**Password**

**Speciality**  
Clinical Department

**Doctor**  
Doctor's name

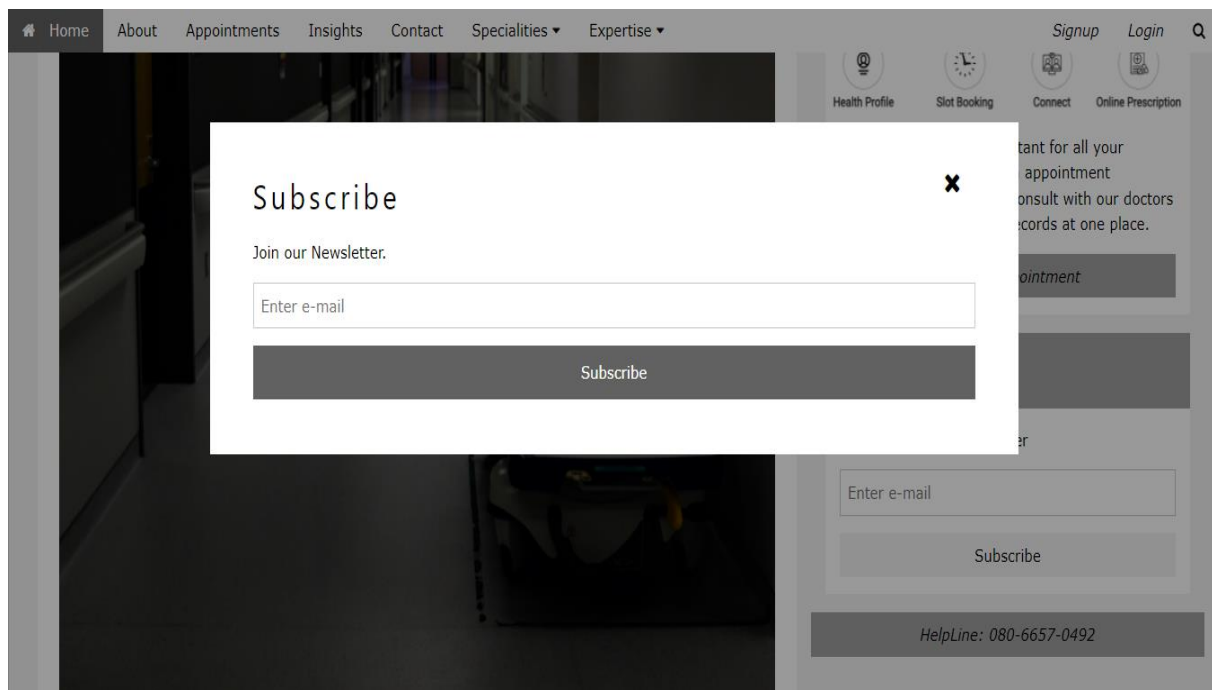
**Slot** dd-mm-yyyy

Don't have an account? [Signup.](#)

Cancel Confirm

FIG 7.12

## NEWSLETTER PAGE



Home About Appointments Insights Contact Specialities Expertise

Signup Login

Health Profile Slot Booking Connect Online Prescription

**Subscribe**

Join our Newsletter.

Enter e-mail

Subscribe

HelpLine: 080-6657-0492

FIG 7.13

## CONCLUSION

This project is created and the aims off this project is accomplished. This project is designed to meet the requirements of assigning jobs. It has been developed in PHP and the database has been built in My SQL server keeping in mind the specifications of the system. The user will be able to book an appointment with the doctor using this website. We have designed the project to provide the user with easy retrieval of data, details of theatre and necessary feedback as much as possible. To implement this as a web application we used php as the technology. Php has advantages such as enhanced performance, scalability, built- in security and simplicity. We have used XAMPP to connect the front end and backend code and make everything perfectly operational. To build any web application using PHP we need a programming language such as PHP and so on. MySQL was used as back-end database since it is one of the most popular open-source databases, and it provides fast data access, easy installation and simplicity. For front end we used HTML and CSS.

## Future Enhancements

Future upgrades to this project will implement:

Option for the user to book an appointment without signing up.

Facility for paying for the appointment beforehand.

Getting immediate confirmation of the appointment according to the real time scenario and schedule of the doctor.

To add user's session timing.

## REFERENCES

1. Ramakrishnan, R., & Gehrke, J. (2011). Database management systems. Boston: McGraw-Hill.
2. [http://www.w3schools.com/html/html\\_css.asp](http://www.w3schools.com/html/html_css.asp)
3. Database systems concepts. Estados Unidos: McGraw-Hill Companies, Inc.
4. Hanna P. (2002): JSP 2.0 The Complete Reference, Second Edition McGraw Hill Education.
5. David F. (2011). JavaScript: The Definitive Guide Sixth edition.
6. <http://stackoverflow.com/questions/tagged/php+mysql>
7. [http://www.w3schools.com/php/php\\_mysql\\_connect.asp](http://www.w3schools.com/php/php_mysql_connect.asp)
8. <https://www.javatpoint.com/xampp>
9. <https://www.php.net/manual/en/>
10. <https://www.javatpoint.com/install-php>