ABSTRACT

In this era of pandemic, hospitals are imposed strict guidelines on the number of person who can enter their facility. Consultations to doctors are also very limited. The online system for appointment is one of the solutions that can be used in order to give the people a way on how to contact their doctors and reserve an appointment for consultation.

Smart Surgeon Scheduling is aimed at developing a platform where a patient can book their own appointment based on their preferrable schedule, it was ease and comfort on the patient end. We are creating a web application for the hospital which manages doctors and patients. In this project, we use PHP and MySQL database. Our project is easy to use, very efficient and user friendly.

LIST OF CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1. Introduction

1.1 Introduction

The project Smart Surgeon Scheduling includes registration of patients, storing their details into the system. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of appointmnets. User can search availability of a doctor using the id.

The Smart Surgeon Scheduling can be entered using a username and password. It is accessible either by an administrator. Only admin can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

Smart Surgeon Scheduling is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes and it's easy for the users to book their own appointment.

1.2 Motivation

Smart Surgeon Scheduling is designed to improve the quality and management of hospital management in the areas of appointmnets and activity-based costing. Smart Surgeon Scheduling enables users to develop their own appointment bookings and improve its effectiveness and quality of work. And some of the advantages are

1.Detailed User's Information

This Smart Surgeon Scheduling booking platform can help the business to maintain the additional feature to capture the client's information at all the touchpoints they have had with the organization. This detail preserved over a longer duration serves as a store-house of context for any client, their preference, and behaviours.

2. Save our Staff Time

The customer service staff spends most of their time handling phone calls, explaining options to customers and managing schedules, all of which can be automated. The time-saving benefits work similarly for our prospects. The individuals who want to book an appointment with you do not necessarily have to commit part of their busy routine to call you for an appointment.They can do it on the go as they carry on with their respective duties. If there’s space available for a given date, our customer can choose their options, pick a date and time as per their preference. This frees up our staff to do more valuable things with their

3. User self-scheduling with 24-hour convenience

At a time when 75% of Millennials choose texting over talking on the phone, more and more people will appreciate the freedom that comes with an online appointment scheduling. Studies show that 35% of clients prefer scheduling appointments during non-business hours, and 40% of online bookings happen after hours. Our customers and can make a reservation whenever it fits into their schedule.

4.Avoid Overbooking

Have you ever had a scenario where two clients turned up at the same time? This is something not new for any organizations, and mistakes like these happen more frequently, to avoid such errors, it is mandatory that Smart Surgeon Scheduling system is put in place, which will remove all the redundancy in the booking.

5. Easier Booking Experience for Users

The traditional way of booking appointments can be quite cumbersome and usually involves multiple back-and-forth communication even when done via an email, instead of the phone. The process not only takes much time, but it also leaves room for human error and miscommunication. By using Smart Surgeon Scheduling,users can access on their own time.

1.3 Scope

This Smart Surgeon Scheduling suggests an actual solution where users can view different booking slots offered and select the favored date and time.

1.4 Project Outline

- The main objective of our project is patient able to book their own appointment based on their schedule time.
- If the scheduled time isn’t preferrable to doctor, he may cancel the appointment.
- And also patient can cancel their appointment.
- The availability and hours of staff can quickly be adjusted depending on the booking volume, which can be readily determined via online booking.

CHAPTER-2

LITERATURE SURVEY

2.Literature Survey

- **Title :** Patient Monitoring in the Hospital Management using Li-Fi.

Authors: Dr. S. Sudha, Ms . Indumathy , A . Lavanya, M. Nishanthi.

Abstract: This paper focuses on the patient monitoring in the hospitals using the Li-Fi technology. Li-Fi stands for LightFidelity. Li-Fi technology, proposed by the German physicist Harold Haas, provides transmission of data through illumination by sending data through an LED light bulb that varies in intensity faster than the human eye can follow. Li-Fi is a bidirectional, high speed and fully networked wireless optical communication and is a form of visible light communication. The proposed model helps in the Patient monitoring in the hospitals and can be done by using the concept of Li-Fi instead of the Wi-Fi technology to avoid the frequency interference with the human body. Sensors such as temperature, heartbeat, glucose and respiration used in this model perform its respective functions. The microcontroller that is used here is PIC16F877A. The output from the microcontroller is fed to the Li-Fi module which transmits the data in the form of light and the receiver end collects this data and then displays the graph for the different parameters using the PC. This report of the patient can be sent to the concerned person through e-mail.

- **Title :** Intelligent Hospital Management System (IHMS).

Authors: Baki Koyuncu , Hakan Koyuncu.

Abstract: An intelligent hospital information management system was developed to assist the patient at the front desk of a hospital. The patient will be able to learn about the doctors, appointment times, relevant departments, laboratory tests and the specific medicine about his/her medical situation. System will provide an intelligent front desk information service for the patients at the hospital entrance. It will also provide software assistance for the doctors to diagnose easily and rapidly by using the program's decision mechanism.

CHAPTER-3

SYSTEM STUDY AND ANALYSIS

3. System Study and Analysis

3.1 Problem Statement

This Smart Surgeon Scheduling offers patients or any user a best way of booking a doctor’s appointment online. This application overcomes the issue of handling and reserve appointments according to user’s choice. The job on occasion becomes very monotonous for the doctor himself in manually allotting appointments for the users as per their convenience.Hence this system offers an active resolution where patients can see different booking slots available and select the preferred date and time. This system also allows users to terminate their booking anytime.

3.2 Existing System

Hospitals use a manual system for booking a doctor’s appointment. And it’s paper oriented hence required more manpower. The data of patient and doctor in hospital is paper oriented hence we cannot share data remotely. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

Drawbacks:

- It’s a time consuming system.
- In this system user authentications are not provided.
- It leads to inconsistencies in data.

3.3 Proposed System

The Smart Surgeon Scheduling is designed for any hospital to replace their existing manual paper based appointment booking system. The proposed system is a smart appointment booking method. In this system user authentication is provided. The system is under administration of assistant who handles the bookings.

Advantages:

- It requires less manpower and also secure than offline appointment.
- Users can book appointment for their necessary date and time.
- This system is used to manage doctor’s time utilization and reduce the patient waiting time.

3.4 Requirements

3.4.1 Functional Requirements

In Software Engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. Generally, functional requirements are expressed in the form “ **system shall do** ”. The plan for implementing functional requirements is detailed in the system design.

In requirements engineering, functional requirements specify particular results of a system. Functional requirements drive the application architecture of a system. A requirements analyst generates use cases after gathering and validating a set of functional requirements.

The following are the Functional requirements of our system:

- Registration Process of SRS (Software Requirements Specification)
- Check Out of SRS
- Report Generation of SRS
- Database of SRS

3.4.2 Non-Functional Requirements

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. The project non functional requirements include the following.

- Performance
- Security
- Maintainability
- Reliability

Availability: A system's "availability" or "uptime" is the amount of time that is operational and available for use. It's related to is the server providing the service to the users. As our system will be used by hundreds of users at any time our system must be available always.

Efficiency: Specifies how well the software utilizes scarce resources: CPU cycles, disk space, memory, bandwidth etc. All of the above mentioned resources can be effectively used by performing most of the validations at client side and reducing the workload on server.

Flexibility: If the organization intends to increase or extend the functionality of the software after it is deployed, that should be planned from the beginning; it influences choices made during the design, development, testing and deployment of the system. New modules can be easily integrated to our system without disturbing the existing modules or modifying the logical database schema of the existing applications.

Portability: Portability specifies the ease with which the software can be installed on all necessary platforms, and the platforms on which it is expected to run. By using appropriate server versions released for different platforms our project can be easily operated on any operating system, hence can be said highly portable.

Scalability: Software that is scalable has the ability to handle a wide variety of system configuration sizes. The nonfunctional requirements should specify the ways in which the system may be expected to scale up (by increasing hardware capacity, adding machines etc.). Our system can be easily expandable. Any additional requirements such as hardware or software which increase the performance of the system can be easily added. An additional server would be useful to speed up the application.

Integrity: Integrity requirements define the security attributes of the system, restricting access to features or data to certain users and protecting the privacy of data entered into the software. Certain features access must be disabled to normal users such as adding the details of files, searching etc which is the sole responsibility of the server. Access can be disabled by providing appropriate logins to the users for only access.

Usability: Ease-of-use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its intended users. Hyper links will be provided for each and every service the system provides through which navigation will be easier. A system that has high usability coefficient makes the work of the user easier.

3.5 System Requirements

3.5.1 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. The following sub-sections discuss the various aspects of hardware requirements.

• Processor	:	Intel Dual Core , i3
• RAM	:	1GB
• Hard Disk	:	80GB

3.5.2 Software Requirements

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

• Operating System	:	Windows 7/ XP/8
• Front End	:	Html,css,java script
• Server Side Script	:	Php
• Server	:	XAMPP
• Database	:	Mysql

CHAPTER-4

SYSTEM DESIGN

4. System Design

4.1 System Architecture Design

A system Architecture is the conceptual model that defines the structure, behavior, and more views of the system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can consists of system components and the sub-systems developed, that will work together the overall system.

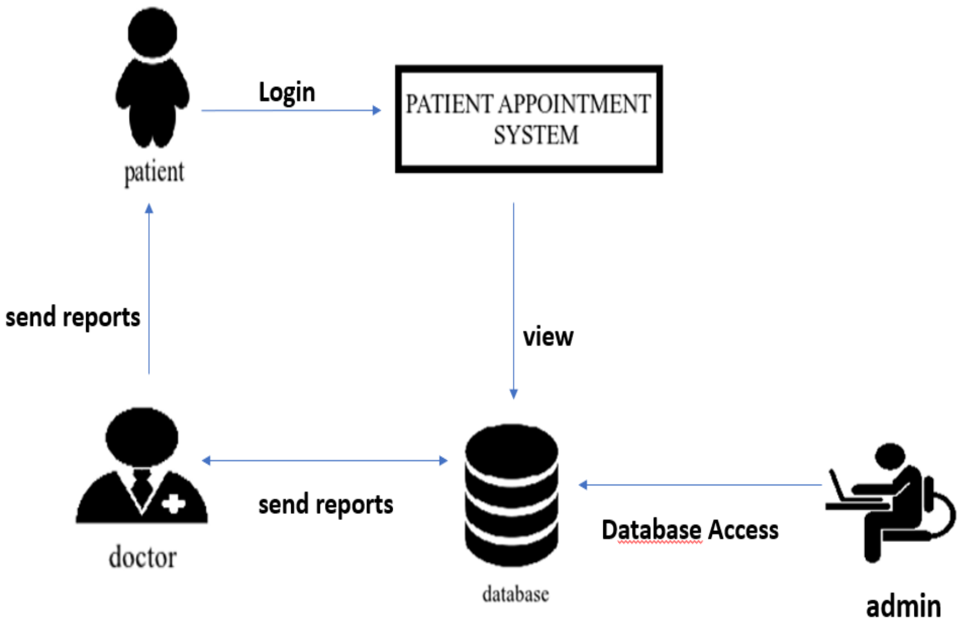


Figure 4.1 System Architecture

4.2 Design Diagrams using UML Approach

4.2.1 Introduction to UML

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language , which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for -

Visualizing

Through UML we see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize, how the components of the system communicate and interact with each other.

Specifying

Specifying means building, models that are precise, unambiguous and complete UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

Constructing

UML models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

Documenting

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating about a system during its developing requirements, architecture, desire, source code, project plans, tests, prototypes releasers, etc...

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine diagrams

1. Class diagram
2. Object diagram
3. Use case diagram
4. Sequence diagram
5. Collaboration diagram
6. State chart diagram
7. Activity diagram
8. Component diagram
9. Deployment diagram

4.2.2 UseCase Diagram

Use case diagram represents the functionality of the system. Use case focus on the behavior of the system from external point of view. Actors are external entities that interact with the system.

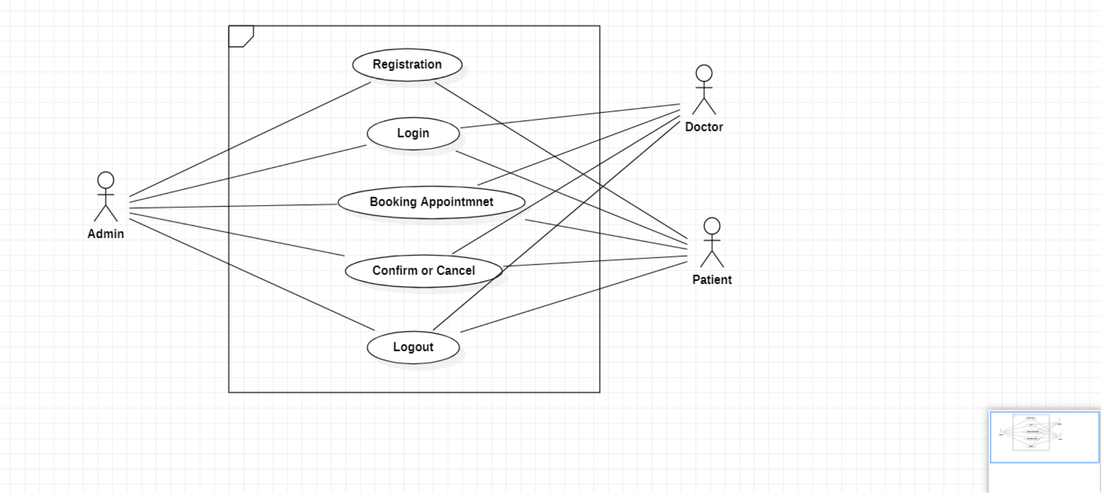


Figure 4.2.2 Usecase Diagram for Booking an Appointment

Usecases: A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

Actors: An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

Include: In one form of interaction, a given use case may include another. "Include is a Directed Relationship between two use cases, implying that the behavior of the included use case is inserted into the behavior of the including use case. The notation is a dashed arrow from the including to the included use case, with the label "«include»".There are no parameters or return values.

Extend: In another form of interaction, a given use case (the extension) may extend another. This relationship indicates that the behavior of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label "«extend»". Modelers use the «extend» relationship to indicate use cases that are "optional" to the base use case.

Generalization: In the third form of relationship among use cases, a generalization/specialization relationship exists. A given use case may have common behaviors, requirements, constraints, and assumptions with a more general use case. In this case, describe them once, and deal with it in the same way, describing any differences in the specialized cases. The notation is a solid line ending in a hollow triangle drawn from the specialized to the more general use case (following the standard generalization notation.

Association: Association between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case.

4.2.3 Class Diagram

Class-based Modeling, or more commonly class-orientation, refers to the style of object-oriented programming in which inheritance is achieved by defining classes of objects; as opposed to the objects themselves (compare Prototype-based programming).

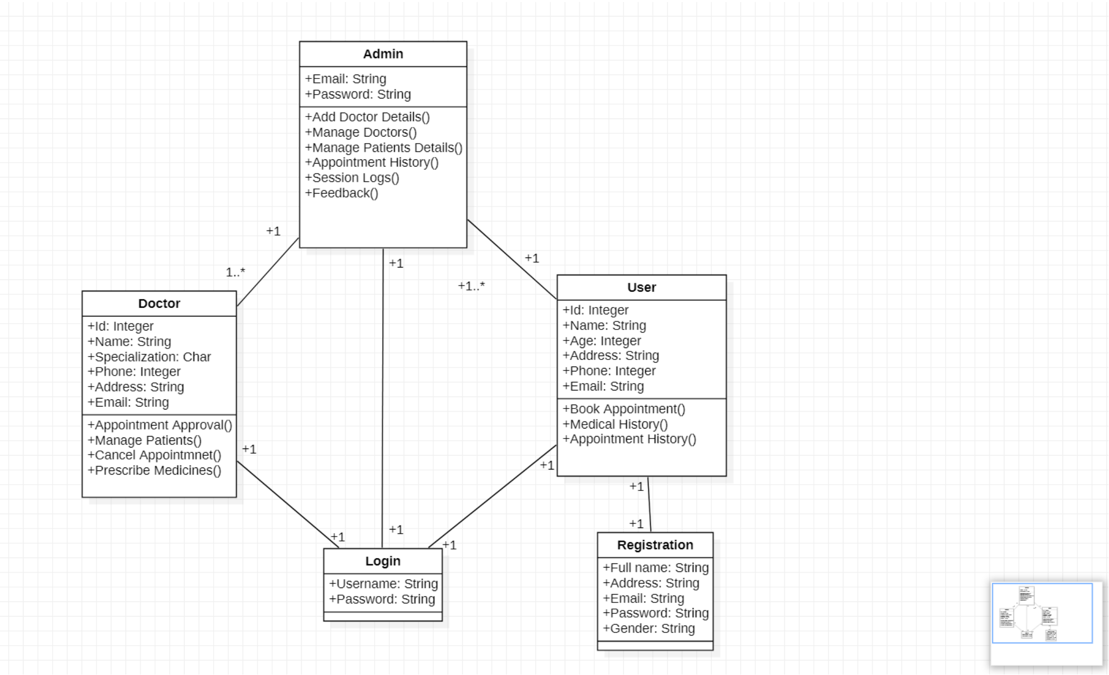


Figure 4.2.3 Class Diagram for Booking an Appointment

The most popular and developed model of OOP is a class-based model, as opposed to an objectbased model. In this model, objects are entities that combine state (i.e., data), behavior (i.e., procedures, or methods) and identity (unique existence among all other objects). The structure and behavior of an object are defined by a class, which is a definition, or blueprint, of all objects of a specific type. An object must be explicitly created based on a class and an object thus created is considered to be an instance of that class. An object is similar to a structure, with the addition of method pointers, member access control, and an implicit data member which locates instances of the class (i.e. actual objects of that class) in the class hierarchy (essential for runtime inheritance features).

4.2.4 Sequence Diagram

A Sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

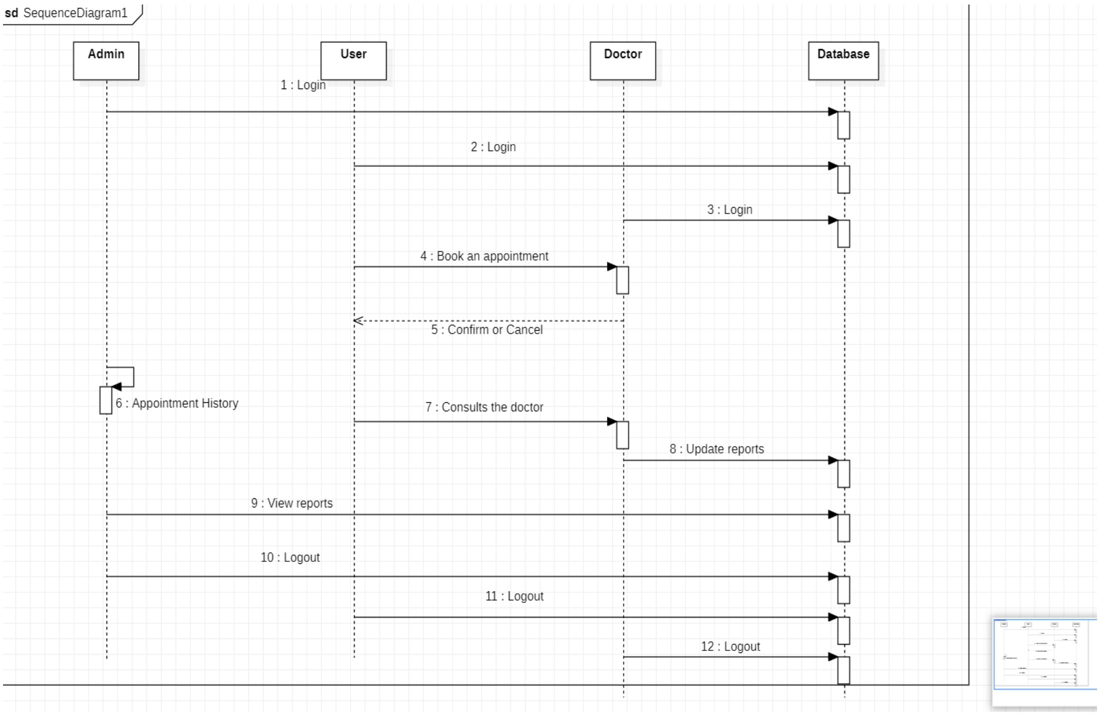


Figure 4.2.4 Sequence Diagram for Booking an Appointment

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. If the lifeline is that of an object, it demonstrates a role. Note that leaving the instance name blank can represent anonymous and unnamed instances. In order to display interaction, messages are used. These are horizontal arrows with the message name written above them. Solid arrows with full heads are synchronous calls, solid arrows with stick heads are asynchronous calls and dashed arrows with stick heads are return messages.

4.2.5 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.

An activity diagram shows the overall flow of control. Activity diagrams are constructed from a limited repertoire of shapes, connected with arrows.

The most important shape types:

- rounded rectangles represent activities;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial state) of the workflow;
- An encircled black circle represents the end (final state).

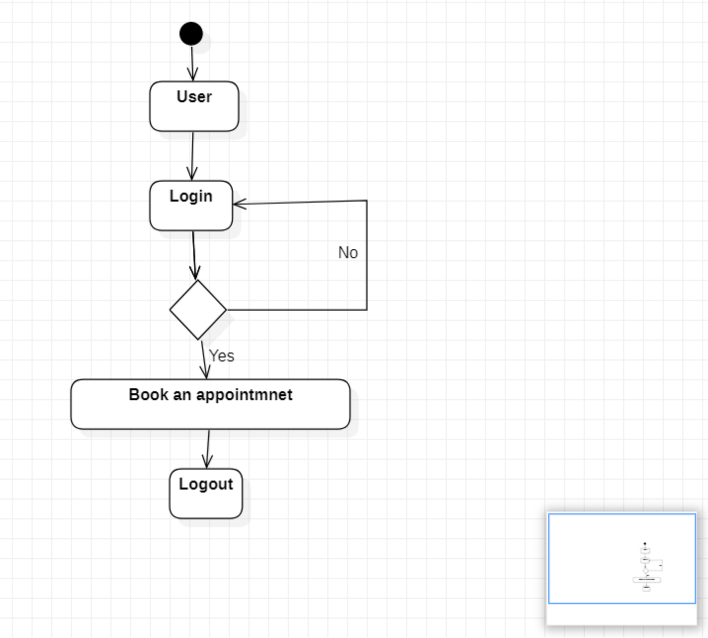


Figure 4.2.5.1 Activity Diagram for User

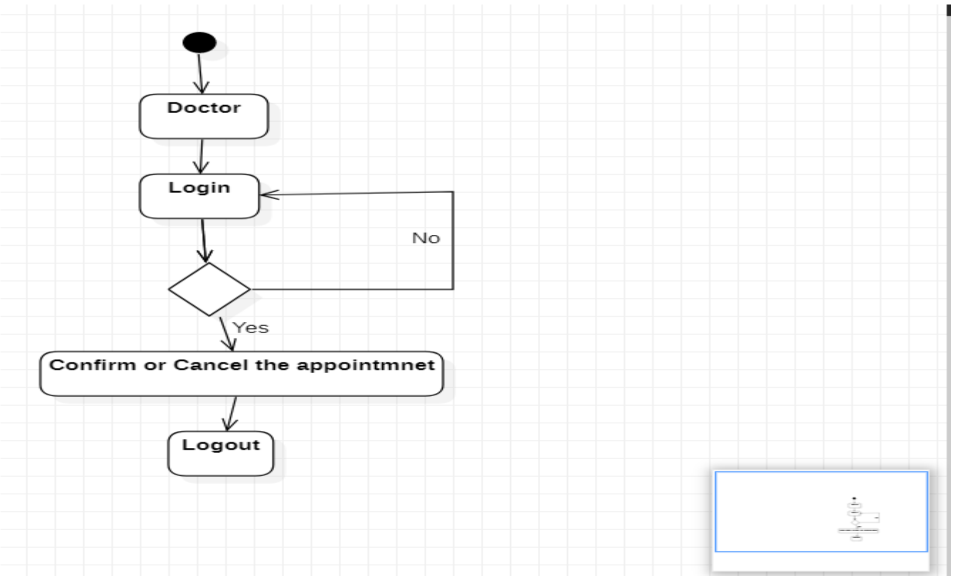


Figure 4.2.5.2 Activity Diagram for Doctor

4.2.6 Statechart Diagram

Objects have behaviors and states. The state of an object depends on its current activity or condition. A state chart diagram shows the possible states of the object and the transitions that cause a change in state.

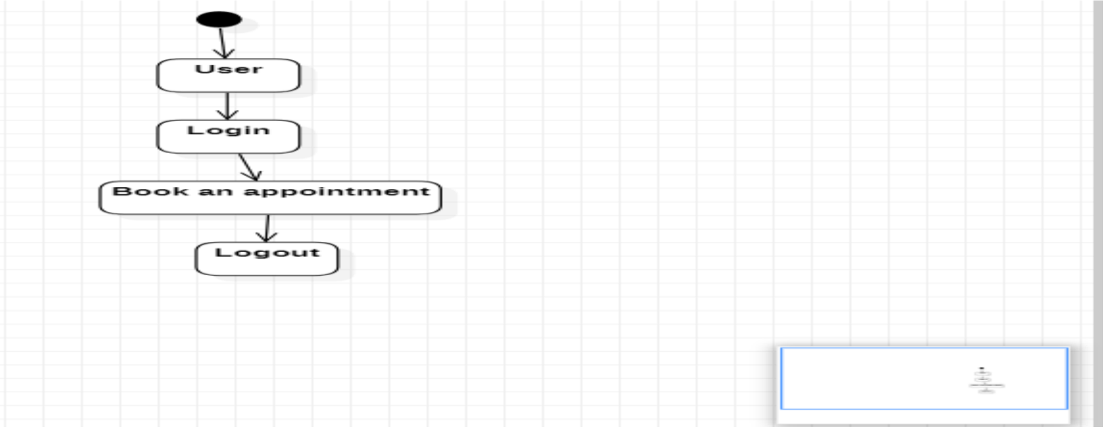


Figure 4.2.6.1 Statechart Diagram for User

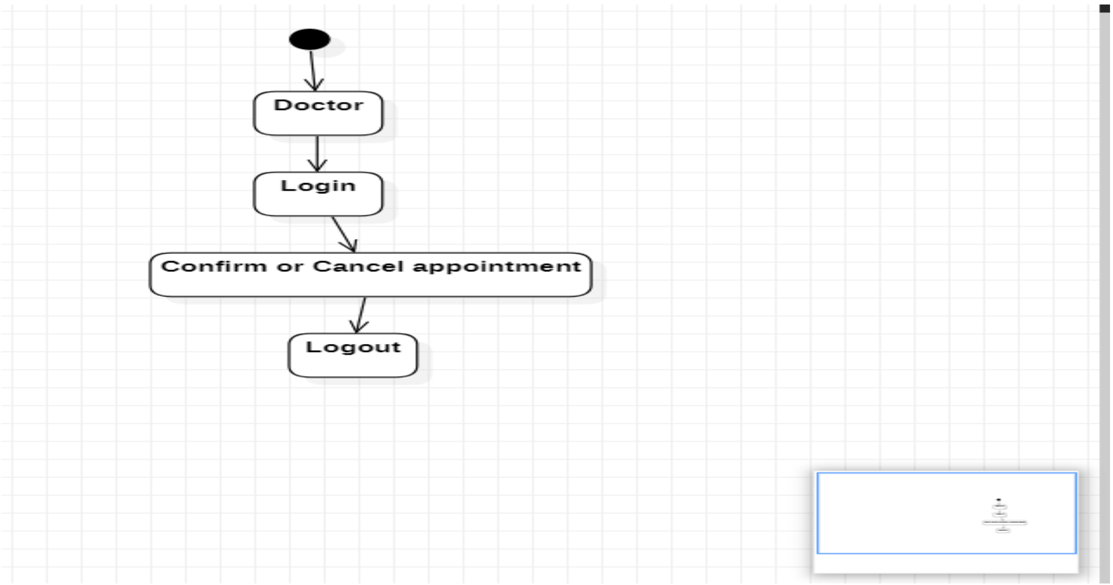


Figure 4.2.6.2 Statechart Diagram for Doctor

A state diagram, also called a state machine diagram or state chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modeling Language. A state diagram resembles a flowchart in which the initial state is represented by a large black dot and subsequent states are portrayed as boxes with rounded corners. There may be one or two horizontal lines through a box, dividing it into stacked sections. In that case, the upper section contains the name of the state, the middle section (if any) contains the state variables and the lower section contains the actions performed in that state. If there are no horizontal lines through a box, only the name of the state is written inside it. External straight lines, each with an arrow at one end, connect various pairs of boxes. These lines define the transitions between states. The final state is portrayed as a large black dot with a circle around it. Historical states are denoted as circles with the letter H inside.

4.2.7 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So, deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams show how they are deployed in hardware. UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components. So, most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers.

The purpose of deployment diagrams can be described as:

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe runtime processing nodes.

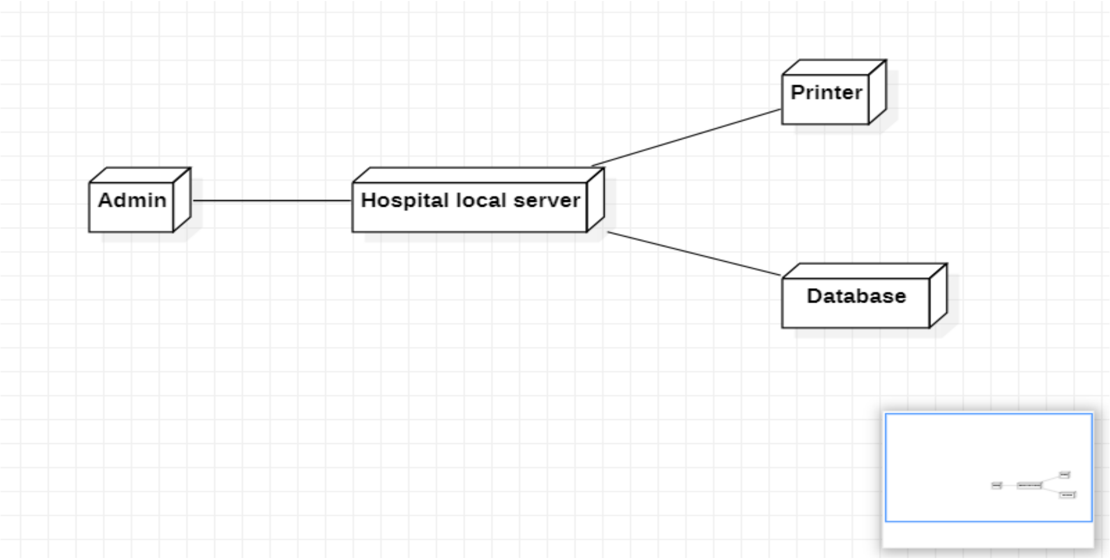


Figure 4.2.7 Deployment Diagram for Hospital

CHAPTER 5

SOFTWARE ENVIRONMENT

5. Software Environment

5.1 Technologies

Introduction to PHP

The term PHP is an acronym for PHP: **Hypertext Preprocessor**. PHP is a server-side scripting language designed specifically for web development. It is open-source which means it is free to download and use. It is very simple to learn and use. The files have the extension “.**php**”.

PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.

It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

PHP is forgiving: PHP language tries to be as forgiving as possible.

PHP Syntax is similar as C lamguage.

Features of PHP

As PHP can do anything related to server-side programming which contains the backend of any web page, it holds a lot of unique features within it. The main purpose of PHP design is web development.

1) Simplicity
2) Flexibility
3) Objective oriented
4) Interpreted language
5) Efficient
6) Fast Performance

7) Free and open-source

8) Case-sensitive

9) Security

10) Platform independent

11) Loosely typed language

12) Real-time access monitoring

13) Error reporting and handling

14) Memory and CPU use information.

Architecture Diagram of PHP based web applications

An Architecture Diagram is a logical diagram that shows how each of the components in a system is connected with each other and how are the data flowing between.

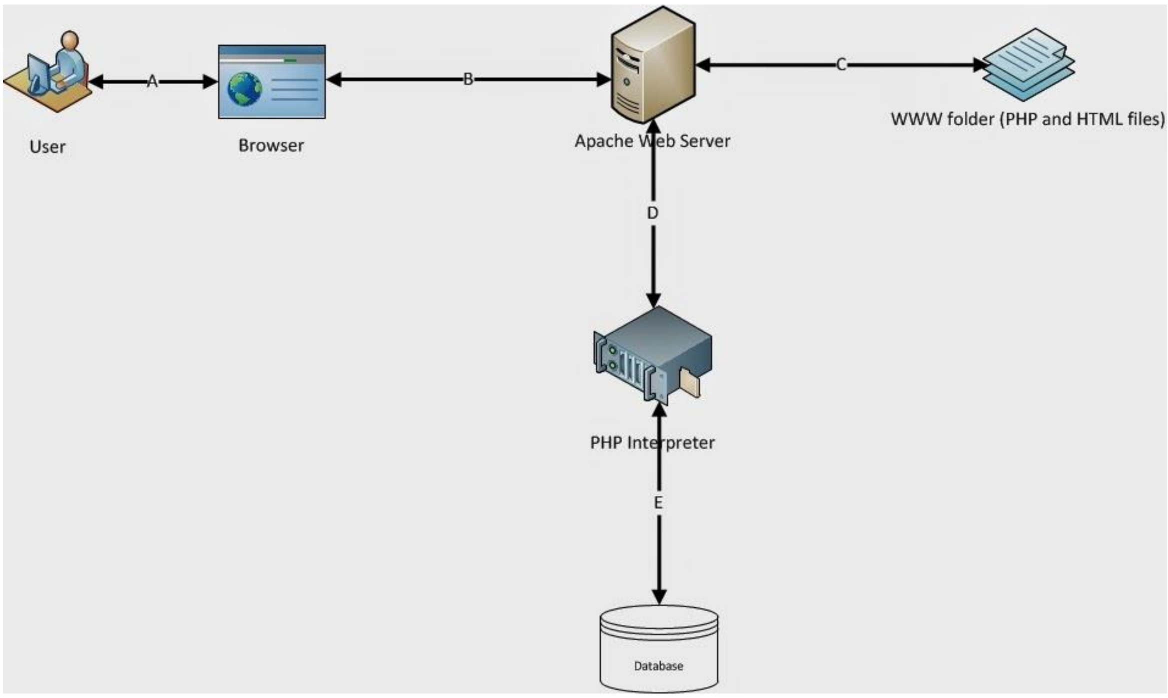


Figure 5.1 Architecture Diagram of PHP

The numbers used below beside the Label names following 1 for Request flow and 2 for response flow.

A1: First user accessed the website through browser. That means user types the URL of the websitein browser and hit go.

B1: The page request on browser will reach to the Web Server (Apache).

C: Web server will collect that requested page (HTML or PHP or Image file etc) from its document root. (In our example it will be www folder in WAMP. You will see it in next one)

B2: Now if it is a static element like HTML , CSS , image file or Java Script file then Apache will send it directly to browser.

A2: And Browser will render it to user on screen.

D1 : If it is a PHP file then Apache sends the content of the file to PHP Interpreter. PHP interpreter interprets the PHP code and executes it. if DB operation is required it performs the same (E).

D2: PHP Interpreter generates output (if the PHP code is to generate any output) and sends to Apache.

B2: Apache sends that content to browser.

A2: Browser renders it to users' screen.

XAMPP Server

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the **Apache Friends**, and its native source code can be revised or modified by the audience. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like **PHP** and **Perl**. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.

What is XAMPP?

XAMPP is an abbreviation where X stands for **Cross-Platform**, A stands for **Apache**, M stands for **MySQL** and the Ps stands for **PHP** and **Perl** respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP and Perl.

Components of XAMPP

As defined earlier, XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. XAMPP is an abbreviated form of each alphabet representing each of its major components. This collection of software contains a web server named **Apache**, a database management system named **MariaDB** and scripting/ programming languages such as **PHP** and **Perl**.
X denotes Cross-platform, which means that it can work on different platforms such as Windows, Linux and macOS.

Many other components are also part of this collection of software and are explained below.

1. **Cross-Platform:** Different local systems have different configurations of operating systems installed in it. The component of cross-platform has been included to increase the utility and audience for this package of Apache distributions. It supports various platforms such as packages of Windows, Linus, and MAC OS.
2. **Apache:** It is an HTTP a cross-platform web server. It is used worldwide for delivering web content. The server application has made free for installation and used for the community of developers under the aegis of Apache Software Foundation.

3. **MariaDB:** Originally, MySQL DBMS was a part of XAMPP, but now it has been replaced by MariaDB. It is one of the most widely used relational DBMS, developed by MySQL. It offers online services of data storage, manipulation, retrieval, arrangement, and deletion.
4. **PHP:** It is the backend scripting language primarily used for web development. PHP allows users to create dynamic websites and applications. It can be installed on every platform and supports a variety of database management systems. It was implemented using C language. PHP stands for **Hypertext Processor**. It is said to be derived from Personal Home Page tools, which explains its simplicity and functionality.
5. **Perl:** It is a combination of two high-level dynamic languages, namely Perl 5 and Perl 6. Perl can be applied for finding solutions for problems based on system administration, web development, and networking. Perl allows its users to program dynamic web applications. It is very flexible and robust.
6. **phpMyAdmin:** It is a tool used for dealing with MariaDB. Its version 4.0.4 is currently being used in XAMPP. Administration of DBMS is its main role.
7. **OpenSSL:** It is the open-source implementation of the Secure Socket Layer Protocol and Transport Layer Protocol. Presently version 0.9.8 is a part of XAMPP.
8. **XAMPP Control Panel:** It is a panel that helps to operate and regulate upon other components of the XAMPP. Version 3.2.1 is the most recent update. A detailed description of the control panel will be done in the next section of the tutorial.
9. **Webalizer:** It is a Web Analytics software solution used for User logs and provide details about the usage.
10. **Mercury:** It is a mail transport system, and its latest version is 4.62. It is a mail server, which helps to manage the mails across the web.
11. **Tomcat:** Version 7.0.42 is currently being used in XAMPP. It is a servlet based on JAVA to provide JAVA functionalities.
12. **Filezilla:** It is a File Transfer Protocol Server, which supports and eases the transfer operations performed on files. Its recently updated version is 0.9.41.

XAMPP Format Support

XAMPP is supported in three file formats:

- **.EXE**- It is an extension used to denote executable files making it accessible to install because an executable file can run on a computer as any normal program.
- **.7z - 7zip file**- This extension is used to denote compressed files that support multiple data compression and encryption algorithms. It is more favored by a formalist, although it requires working with more complex files.
- **.ZIP**- This extension supports lossless compression of files. A Zipped file may contain multiple compressed files. The **Deflate algorithm** is mainly used for compression of files supported by this format. The .ZIP files are quite tricky to install as compared to .EXE

MySQL Introduction

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as **Foreign Keys**.

A **Relational DataBase Management System (RDBMS)** is a software that

- Enables you to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.

RDBMS Terminology

Few definitions related to the database.

- **Database** − A database is a collection of tables, with related data.
- **Table** − A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- **Column** − One column (data element) contains data of one and the same kind, for example the column postcode.
- **Row** − A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
- **Redundancy** − Storing data twice, redundantly to make the system faster.
- **Primary Key** − A primary key is unique. A key value can not occur twice in one table. With a key, you can only find one row.
- **Foreign Key** − A foreign key is the linking pin between two tables.
- **Compound Key** − A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- **Index** − An index in a database resembles an index at the back of a book.

MySQL Database

- MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons .
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL is very friendly to PHP, the most appreciated language for web development.

- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

Creating MySQL Database with XAMPP

XAMPP stack of software is an open-source localhost server providing a number of functionalities through the package of software it contains. The software, which is part of XAMPP is started/stopped using the XAMPP Control Panel.
It is used for testing the projects and modifications offline before launching it on the global web. One such very important functionality provided by XAMPP is the creation of the MySQL database. This is done by using **phpMyAdmin**.

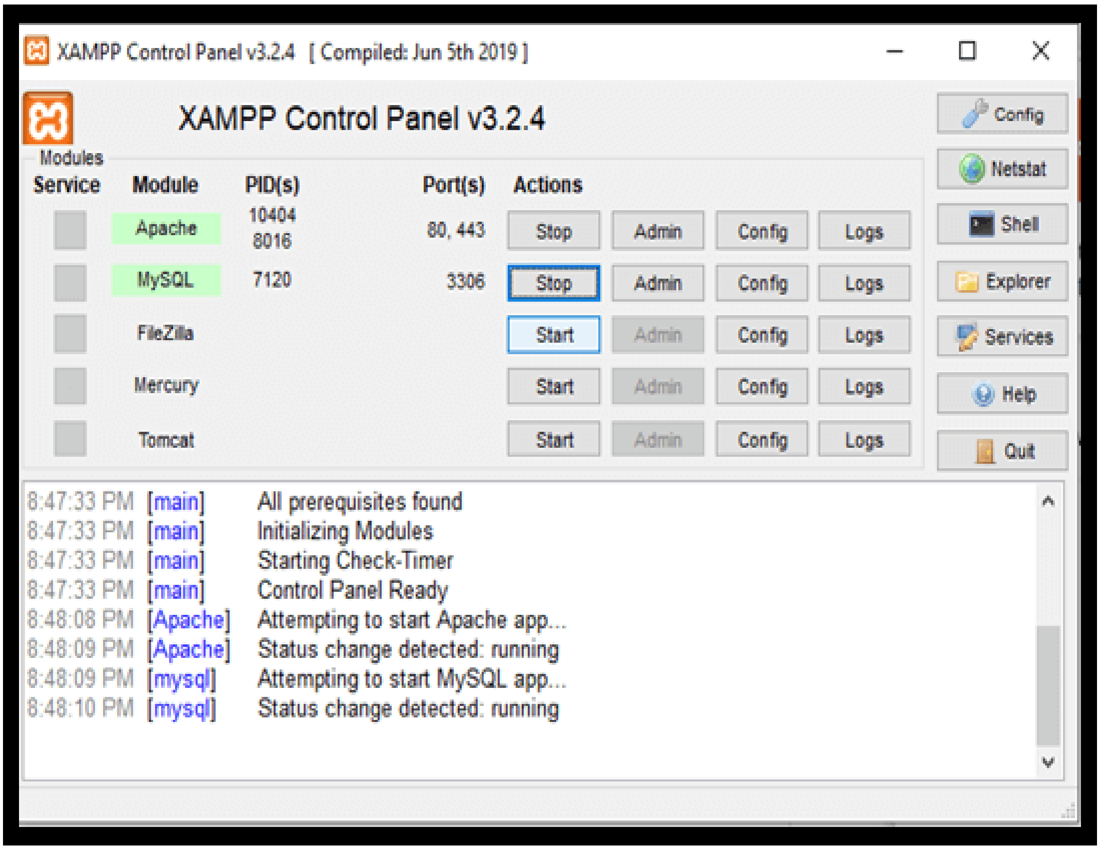
phpMyAdmin

phpMyAdmin is a costless and open source software that provides the functionality of operating and managing MySQL over the internet. It provides an ease to the user to control and supervise the database with the help of a graphic user interface known as phpMyAdmin.This GUI is written in PHP programming language. Over time it has gained a lot of trust and demand for the purpose of finding a web-based MySQL administration solution. The user can operate upon MySQL via phpMyAdmin user interface while still directly executing SQL queries. The GUI allows the host to carry a number of manipulation operations on the database, such as editing, creating, dropping, amending, alteration of fields, tables, indexes, etc. It can also be used to manage access control over the data by giving privileges and permissions. phpMyAdmin has thus a vital role to play in handling and creating a database.

Steps To Create MySQL Database Using XAMPP

STEP 1- Navigate to XAMPP in your system or simply launch it by clicking the XAMPP Icon. The Control Panel is now visible and can be used to initiate or halt the working of any module.

STEP 2- Click on the **"Start"** button corresponding to **Apache** and **MySQL** modules. Once it starts working, the user can see the following screen:



XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]
XAMPP Control Panel v3.2.4
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	10404 8016	80, 443	Stop	Admin	Config	Logs
	MySQL	7120	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
8:47:33 PM  [main]     All prerequisites found
8:47:33 PM  [main]     Initializing Modules
8:47:33 PM  [main]     Starting Check-Timer
8:47:33 PM  [main]     Control Panel Ready
8:48:08 PM  [Apache]   Attempting to start Apache app...
8:48:09 PM  [Apache]   Status change detected: running
8:48:09 PM  [mysql]    Attempting to start MySQL app...
8:48:10 PM  [mysql]    Status change detected: running
```

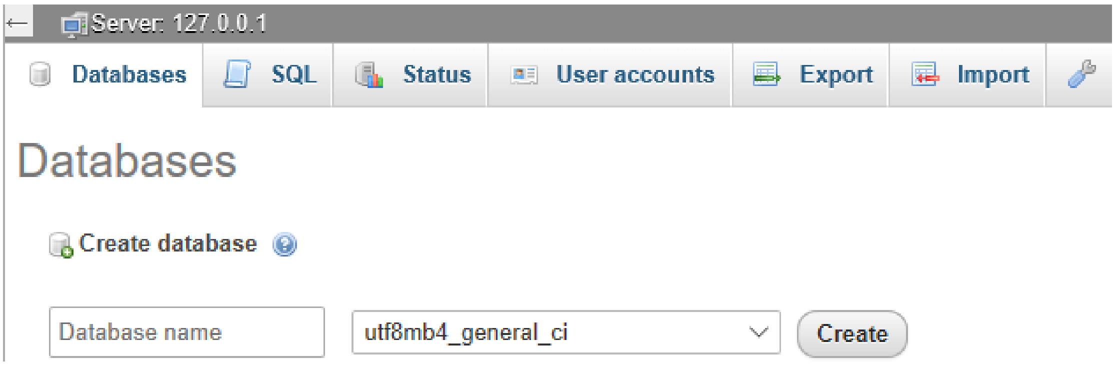
Figure 5.1.1 XAMPP Control Panel

STEP 3- Now click on the **"Admin"** button corresponding to the **MySQL** module. This automatically redirects the user to a web browser to the following address- **http://localhost/phpmyadmin**

STEP 4- One can see a number of tabs such as Database, SQL, User Accounts, Export, Import, Settings, etc. Click on the **"Database"** tab. Here we can see the **Create** option. Choose an appropriate name for the input field titled Database name.

Things to keep in mind while selecting the name for the database are-

- The number of characters used should be equal to or less than 64.
- The name should comprise of letters, numbers and underscore.
- The DB name should not start with a number.
- It should be relevant to the topic for which it is being created.



utf8mb4_general_ci

Figure 5.1.2 Creating a Database in phpMyAdmin

STEP 5- It is very important to create tables in order to store the information in a systematic manner. In this step, we will build tables for the created database. In the created Database (Login page in this case), click on the 'Structure' tab. Towards the end of the tables list, the user will see a **'Create Table'** option. Fill the input fields titled **"Name"** and **"Number of Columns"** and hit the **'Go'** button.

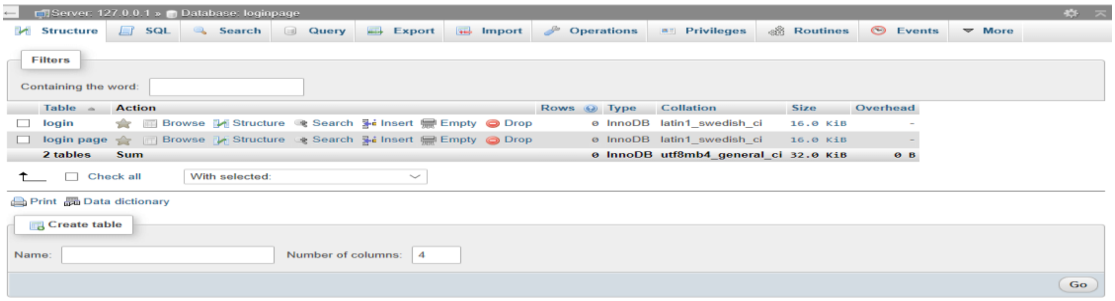


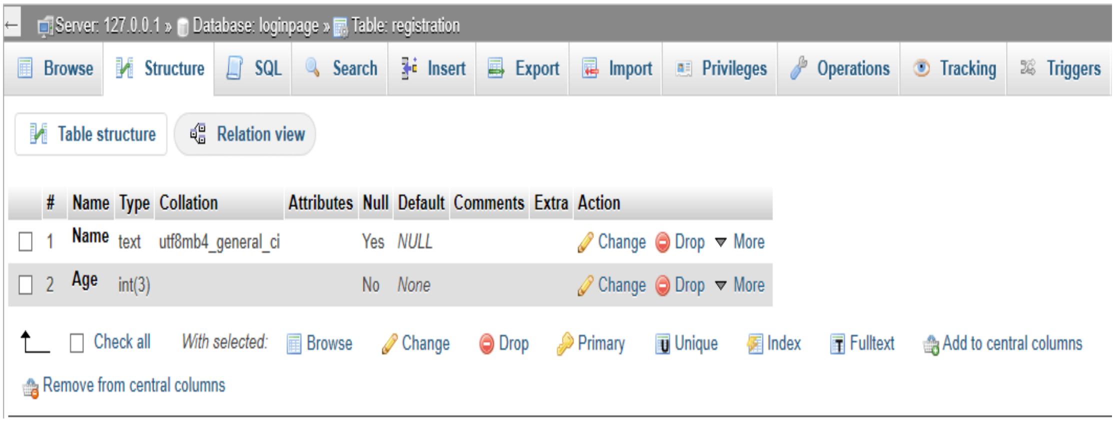
	Table	Action	Rows	Type	Collation	Size	Overhead
□	login	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	login page	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
	2 tables	Sum	0	InnoDB	utf8mb4_general_ci	32.0 KiB	0 B

Name:

Number of columns: 4

Figure 5.1.3 Creating a table

STEP 6- Now, we have to initialize our columns based on their type. Enter the names for each of your columns, select the type, and the maximum length allowed for the input field.



Server: 127.0.0.1 » Database: loginpage » Table: registration

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**Name**	text	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
□	2	**Age**	int(3)			No	*None*			Change Drop More

Figure 5.1.4 Insert data into the table

Click on **"Save"** in the bottom right corner. The table with the initialized columns has been created. You can create any number of tables for your database.

HTML

Hyper Text Markup Language Hypertext Markup Language (HTML), the language of the World Wide Web (WWW), allows users to produce Web pages that include text, graphics and pointers to other Web pages (Hyperlinks). HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags

<!-- -->	Specifies comments
<a> …..</a>	Creates hypertext links
<b>…..</b>	Formats text as bold
<big>….</big>	Formats text in large font
<body>….</body>	Contains all tags and text in the HTML document
<center>….</center>	Definition of a term
<dd>…..</dd>	Definition of a term
<dl>…..</dl>	Creates definition list
<font>….</font>	Formats text with a particular font
<form>….</form>	Encloses a fill-out form
<frame>….</frame>	Defines a particular frame in a set of frames
<h#>…..</h#>	Creates headings of different levels
<head>….</head>	Contains tags that specify information about a document
<table>…..</table>	Creates a table
<td>….</td>	Indicates table data in a table
<tr>…..</tr>	Designates a table row
<th>…..</th>	Creates a heading in a table
<html>….</html>	Contains all other HTML tags

Adavanatges

A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.

HTML is platform independent.

HTML tags are not case-sensitive.

JavaScript

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then updates the browser's display. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags <script>…</script>

We can place the **<script>** tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the **<head>** tags.

The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

Syntax:

```
<script>
      JavaScript code
</script>
```

There are following features of JavaScript:

1. All popular web browsers support JavaScript as they provide built-in execution environments. It provides good control to the users over the web browsers.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted, case- sensitive language.

CHAPTER-6

IMPLEMENTATION

6.Implementation

6.1 Modules

There are three modules that are implementing the Smart Surgeon Scheduling. They are as follows:

- Admin Module
- User Module
- Doctor Module

6.2 Module Description

6.2.1 Admin Module:

This admin module handles various enquires about the user(patient's) appointment and details about availability of doctors and manage both user and doctor details.

And also it handles all the entry details for the registration requirement such as

- Dashboard
- Doctors
- Users
- Patients
- Appointment History
- Doctor session log
- Patient session log
- Reports
- Patient Search

Dashboard: In this section, admin can view the Patients, Doctors, Appointments.

Doctors: In this section, admin can add doctor’s specialization and mange doctors(view/add/delete/update)

Users: In this section, admin can view users details (who take online appointment) and also have right to delete irrelevant user.

Patients: In this section, admin can view patient’s details.

Appointment History: In this section, admin can view appointment history.

Session Logs: In this section, admin can see login and logout time of doctor and patient.

Reports: In this section, admin can view reports of patients in particular periods.

Patient Search: In this section, admin can search patient with the help of patient name and mobile number.

6.2.2 User module:

This Smart Surgeon Scheduling mainy focus on user's facilities. The user can view the doctor’s report about his / her appointment and his / her prescription details. Thus, the patient can clearly know what has happened to him/her and how she or he should take care of herself or himself.

- Dashboard
- Book Appointment
- View Diagnosis report
- Appointment History

Dashboard: In this section, patients can view his/her profile, Appointments and Book Appointment.

Book Appointment: In this section, Patient can book his/her appointment.

Appointment History: In this section, Patients can see his/her their own appointment history.

Medical History: In this section, Patients can see his/her their medical history.

6.2.3 Doctor module:

The doctor is the user who will help the patients with their physical health. The doctor will be able to approve the appointments whether he or she wants to take them or not.

If the doctor wants, he or she can cancel the appointments.

The doctor can view the medication history of the patients.

And A doctor can manages

- Dashboard
- Appointment Approval
- Patients
- Search required patienst

Dashboard: In this section, doctor can view his/her own profile and online appointments.

Appointment History: In this section, Doctor can see patient’s appointments .

Patients: In this section, doctor can manage patients (Add/Update).

Search: In this section, doctor can search patient with the help of patient name or mobile number.

CHAPTER-7

TESTING

7.Testing

7.1 Introduction to Testing

The Purpose of Testing is to discover Errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of testing. Each test type addresses a specific testing requirement.

7.2 Testing Strategies

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results. Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test Strategy and Approach

Field testing will be performed manually and functional tests will be written in detail.

Test Objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed

Features to be Tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error. Integration tests are designed to test integrated software components to determine if they actually run as one program.

Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Test Results

All the test cases mentioned above passed successfully, no defects encountered.

Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

7.3 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing predriven process links and integration points.

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot see into it. The test provides inputs and responds to outputs without considering how the software works.

CHAPTER-8

SCREENSHOTS

8. Screenshots

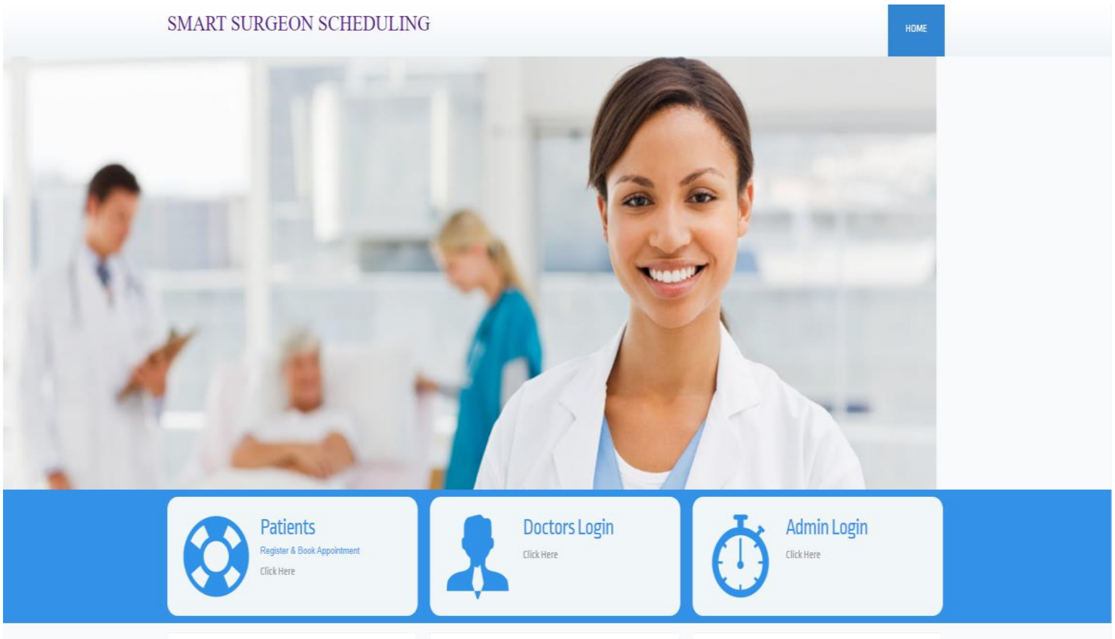
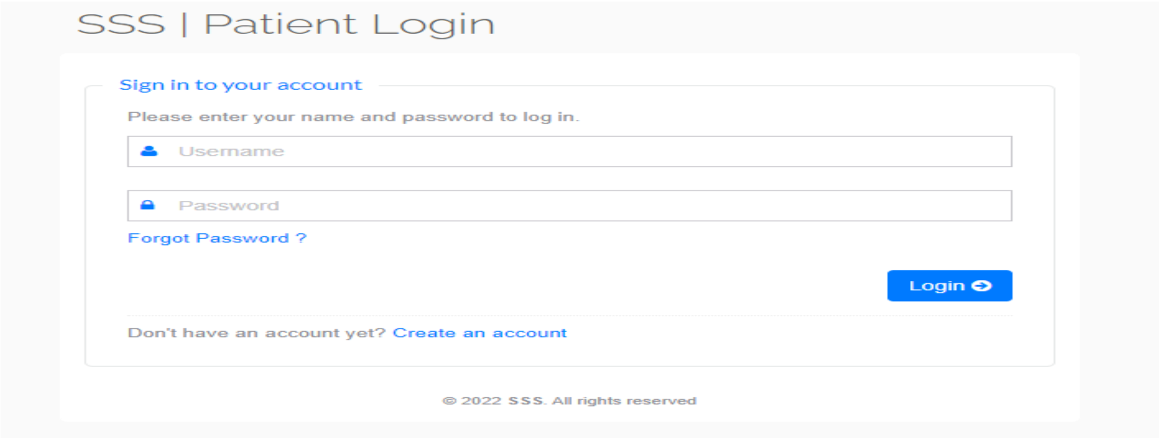


Figure 8.1 Home Page


SSS | Patient Login

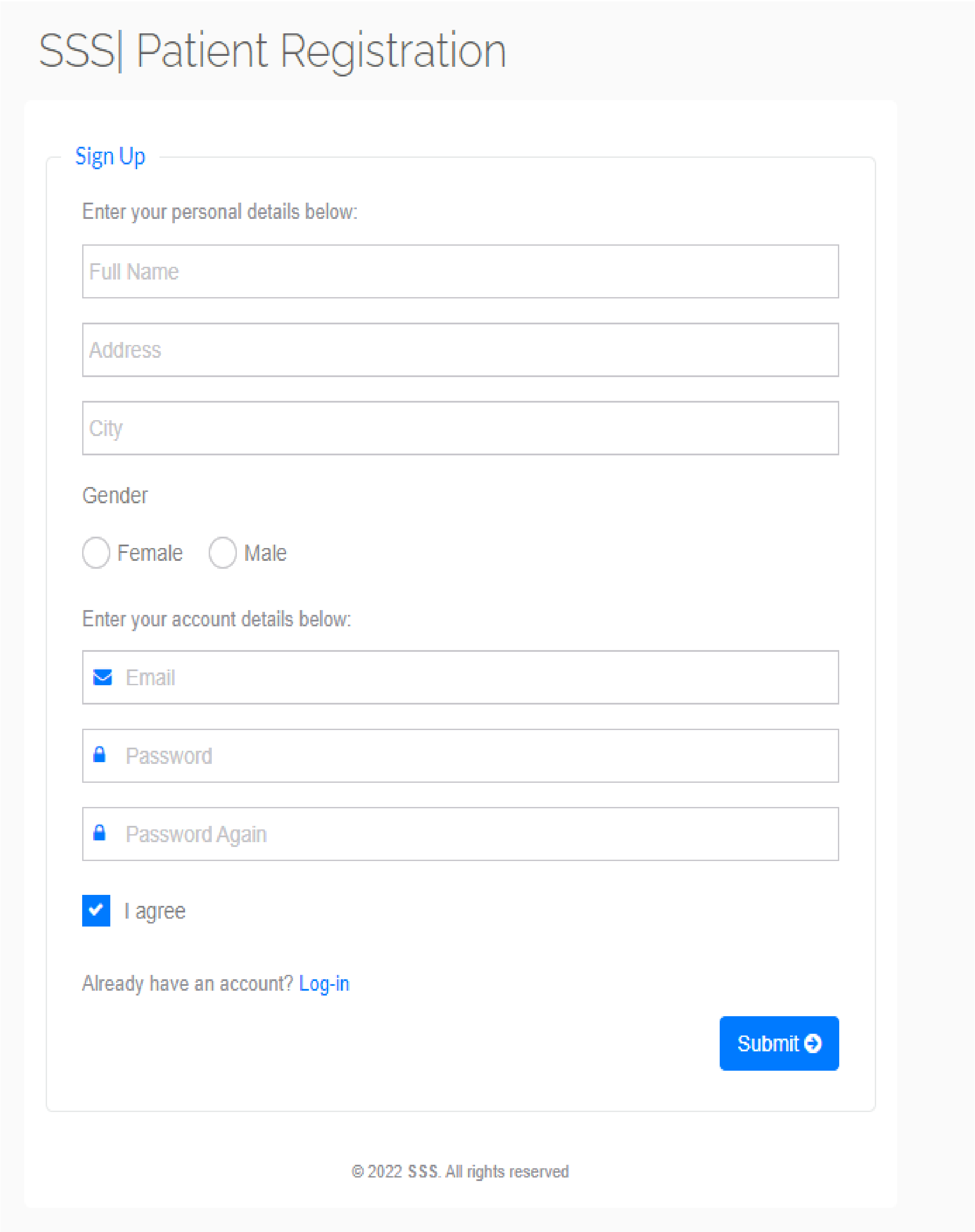
Sign in to your account

Please enter your name and password to log in.

Don't have an account yet? Create an account

© 2022 SSS. All rights reserved

Figure 8.2 Patient Login Page



SSS| Patient Registration

Sign Up

Enter your personal details below:

Full Name

Address

City

Gender

○ Female ○ Male

Enter your account details below:

Email

Password

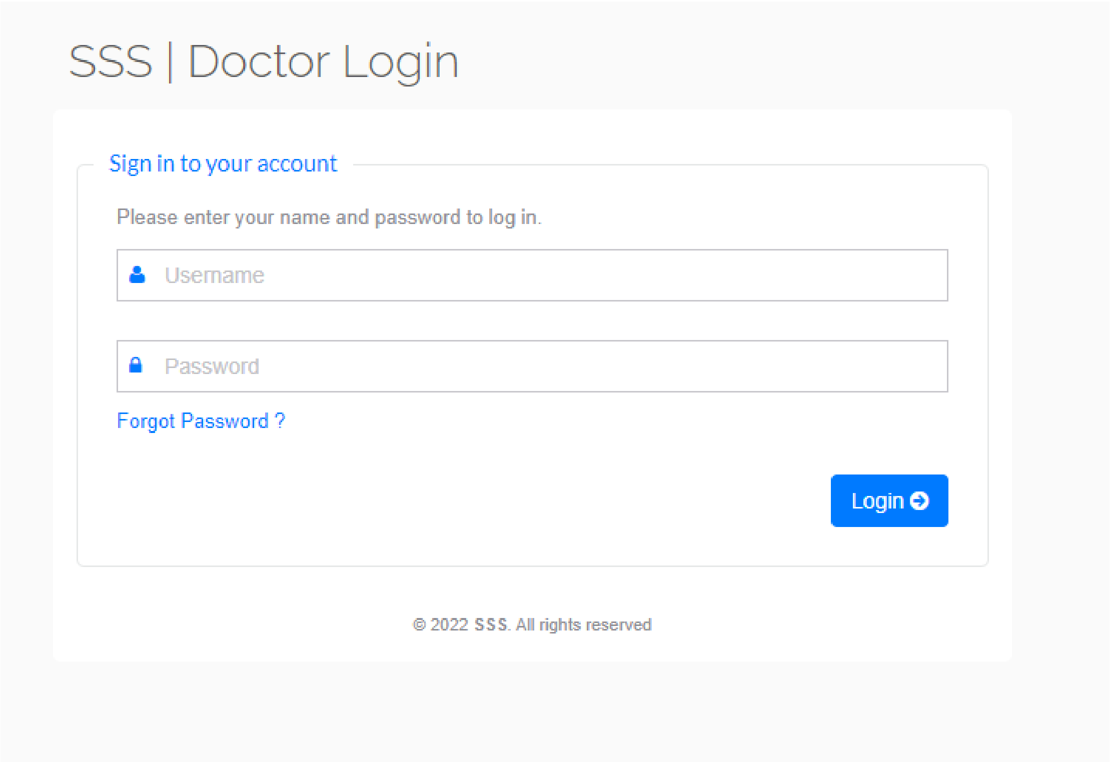
Password Again

☑ I agree

Already have an account? Log-in

© 2022 SSS. All rights reserved

Figure 8.3 Patient Registration Page



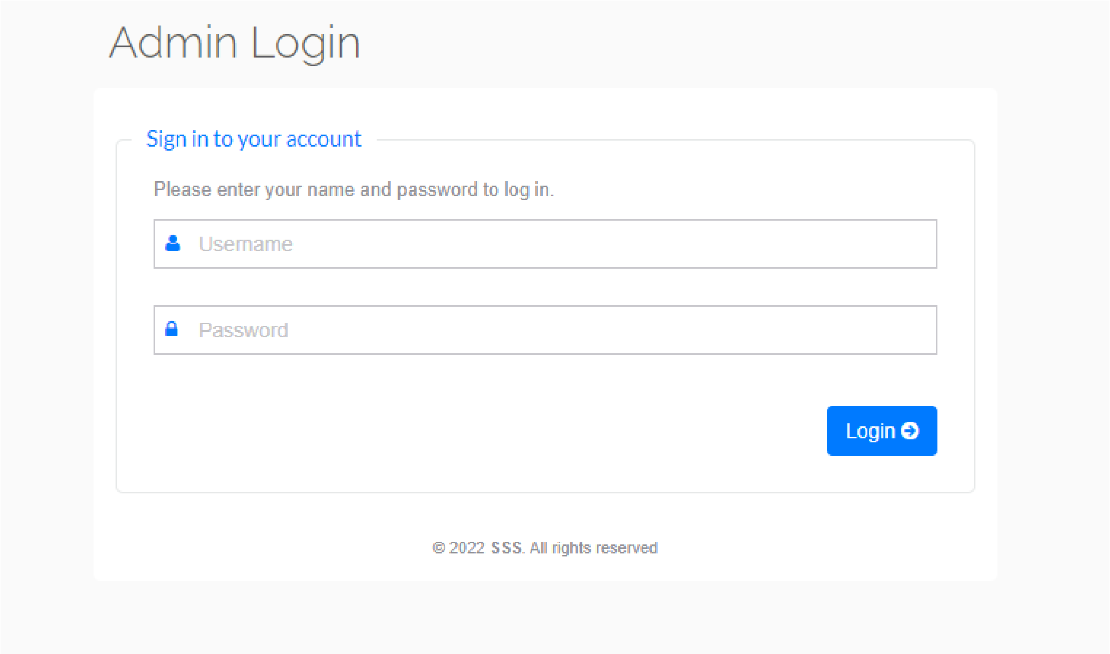
SSS | Doctor Login

Sign in to your account

Please enter your name and password to log in.

© 2022 SSS. All rights reserved

Figure 8.4 Doctor Login Page



Admin Login

Sign in to your account

Please enter your name and password to log in.

© 2022 SSS. All rights reserved

Figure 8.5 Admin Login Page

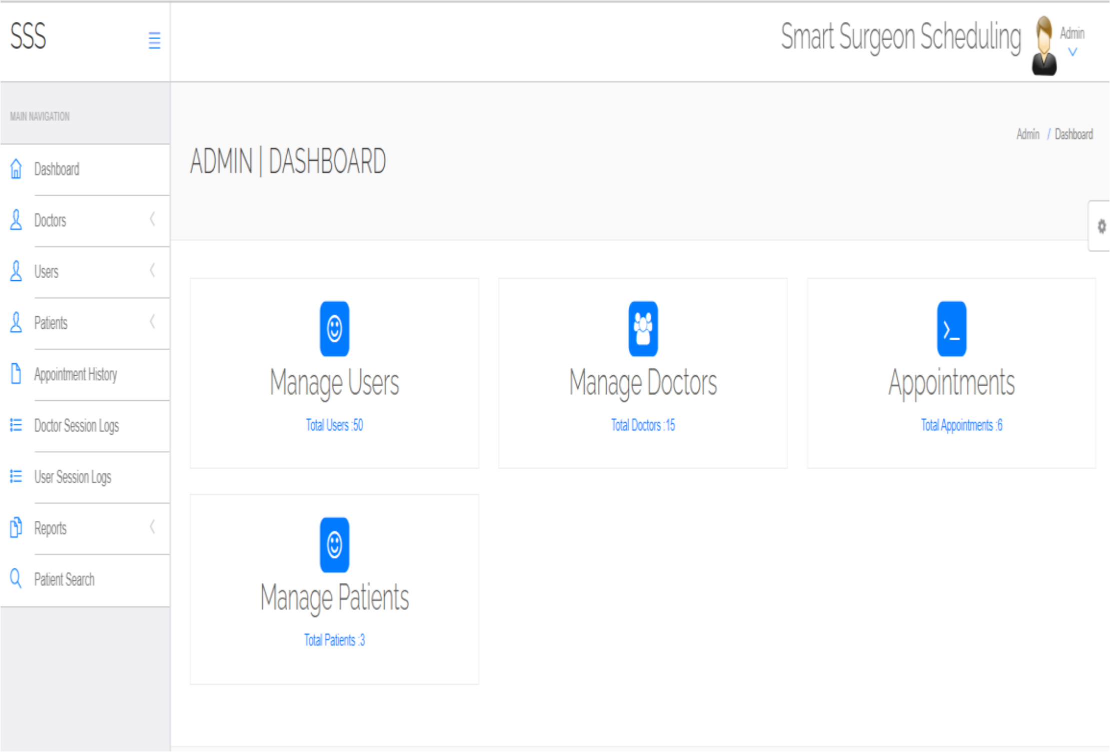
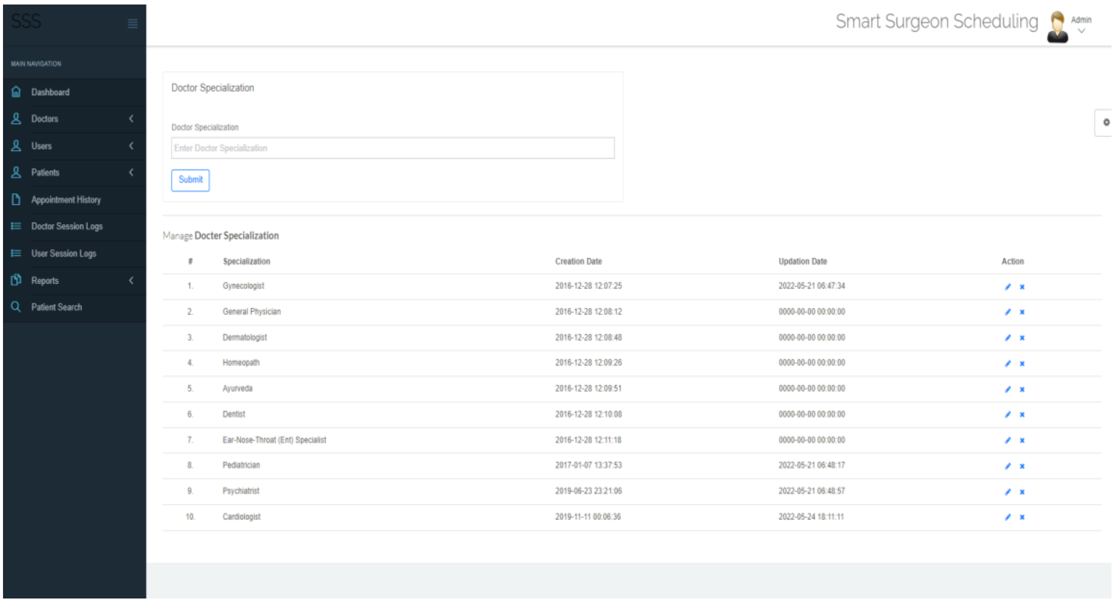


Figure 8.6 Admin Dashboard



Doctor Specialization

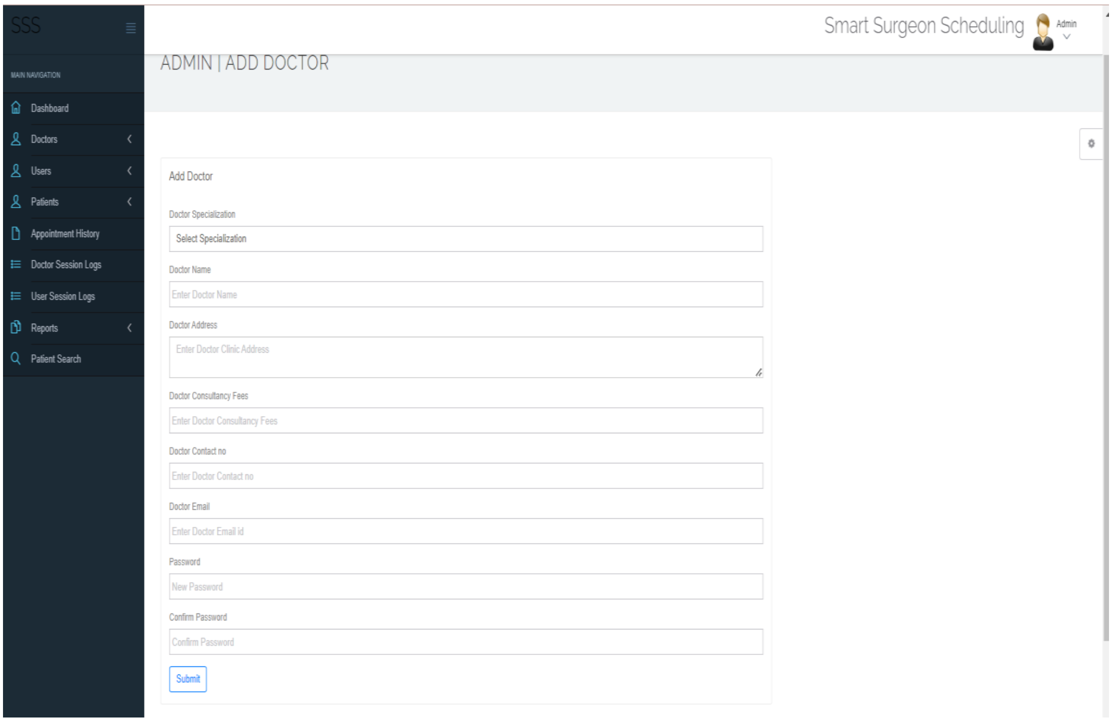
Doctor Specialization

Enter Doctor Specialization

Manage Docter Specialization

#	Specialization	Creation Date	Updation Date	Action
1.	Gynecologist	2016-12-28 12:07:25	2022-05-21 06:47:34	✎ ✕
2.	General Physician	2016-12-28 12:08:12	0000-00-00 00:00:00	✎ ✕
3.	Dermatologist	2016-12-28 12:08:48	0000-00-00 00:00:00	✎ ✕
4.	Homeopath	2016-12-28 12:09:26	0000-00-00 00:00:00	✎ ✕
5.	Ayurveda	2016-12-28 12:09:51	0000-00-00 00:00:00	✎ ✕
6.	Dentist	2016-12-28 12:10:08	0000-00-00 00:00:00	✎ ✕
7.	Ear-Nose-Throat (Ent) Specialist	2016-12-28 12:11:18	0000-00-00 00:00:00	✎ ✕
8.	Pediatrician	2017-01-07 13:37:53	2022-05-21 06:48:17	✎ ✕
9.	Psychiatrist	2019-06-23 23:21:06	2022-05-21 06:48:57	✎ ✕
10.	Cardiologist	2019-11-11 00:06:36	2022-05-24 18:11:11	✎ ✕

Figure 8.7 Doctor Specialization



ADMIN | ADD DOCTOR

Add Doctor

Doctor Specialization

Select Specialization

Doctor Name

Enter Doctor Name

Doctor Address

Enter Doctor Clinic Address

Doctor Consultancy Fees

Enter Doctor Consultancy Fees

Doctor Contact no

Enter Doctor Contact no

Doctor Email

Enter Doctor Email id

Password

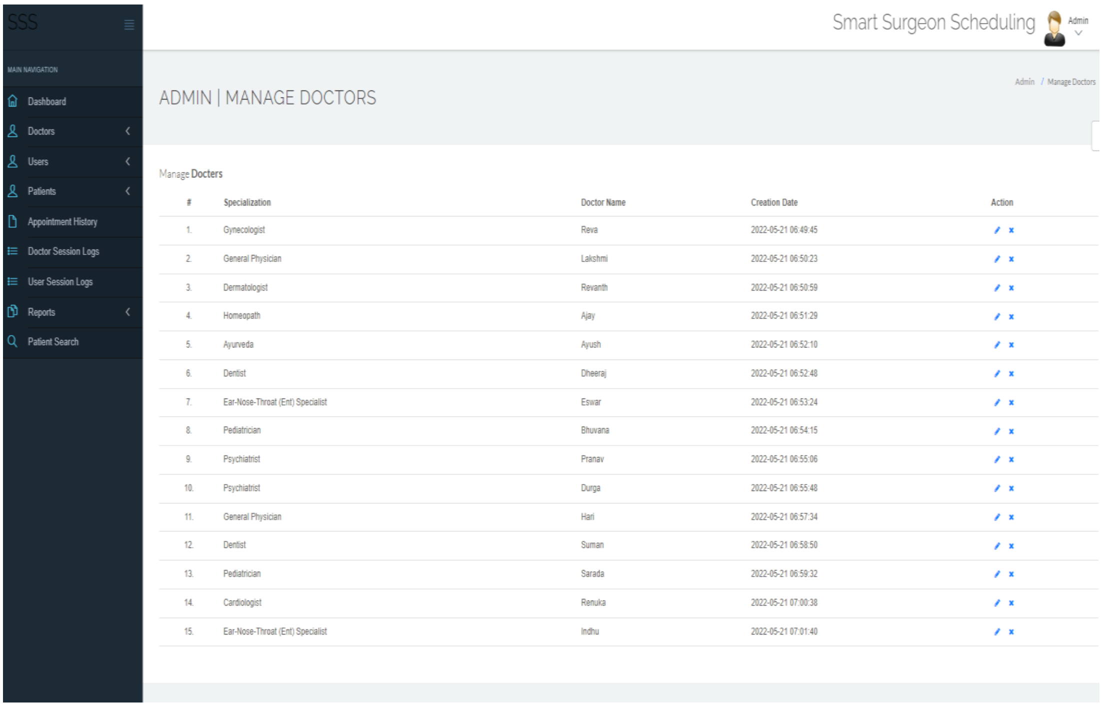
New Password

Confirm Password

Confirm Password

Submit

Figure 8.8 Add Doctor Details

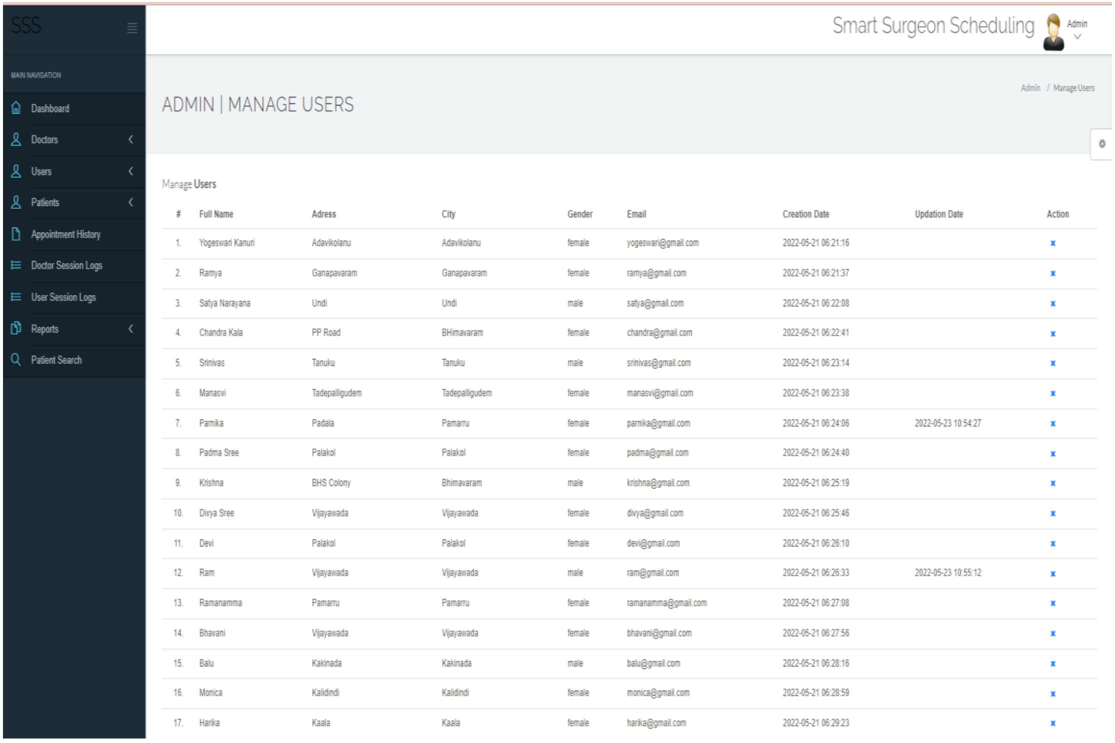


ADMIN | MANAGE DOCTORS

Manage Docters

#	Specialization	Doctor Name	Creation Date	Action
1.	Gynecologist	Reva	2022-05-21 06:49:45	✎ ✕
2.	General Physician	Lakshmi	2022-05-21 06:50:23	✎ ✕
3.	Dermatologist	Revanth	2022-05-21 06:50:59	✎ ✕
4.	Homeopath	Ajay	2022-05-21 06:51:29	✎ ✕
5.	Ayurveda	Ayush	2022-05-21 06:52:10	✎ ✕
6.	Dentist	Dheeraj	2022-05-21 06:52:48	✎ ✕
7.	Ear-Nose-Throat (Ent) Specialist	Eswar	2022-05-21 06:53:24	✎ ✕
8.	Pediatrician	Bhuvana	2022-05-21 06:54:15	✎ ✕
9.	Psychiatrist	Pranav	2022-05-21 06:55:06	✎ ✕
10.	Psychiatrist	Durga	2022-05-21 06:55:48	✎ ✕
11.	General Physician	Hari	2022-05-21 06:57:34	✎ ✕
12.	Dentist	Suman	2022-05-21 06:58:50	✎ ✕
13.	Pediatrician	Sarada	2022-05-21 06:59:32	✎ ✕
14.	Cardiologist	Renuka	2022-05-21 07:00:38	✎ ✕
15.	Ear-Nose-Throat (Ent) Specialist	Indhu	2022-05-21 07:01:40	✎ ✕

Figure 8.9 Manage Doctor Details

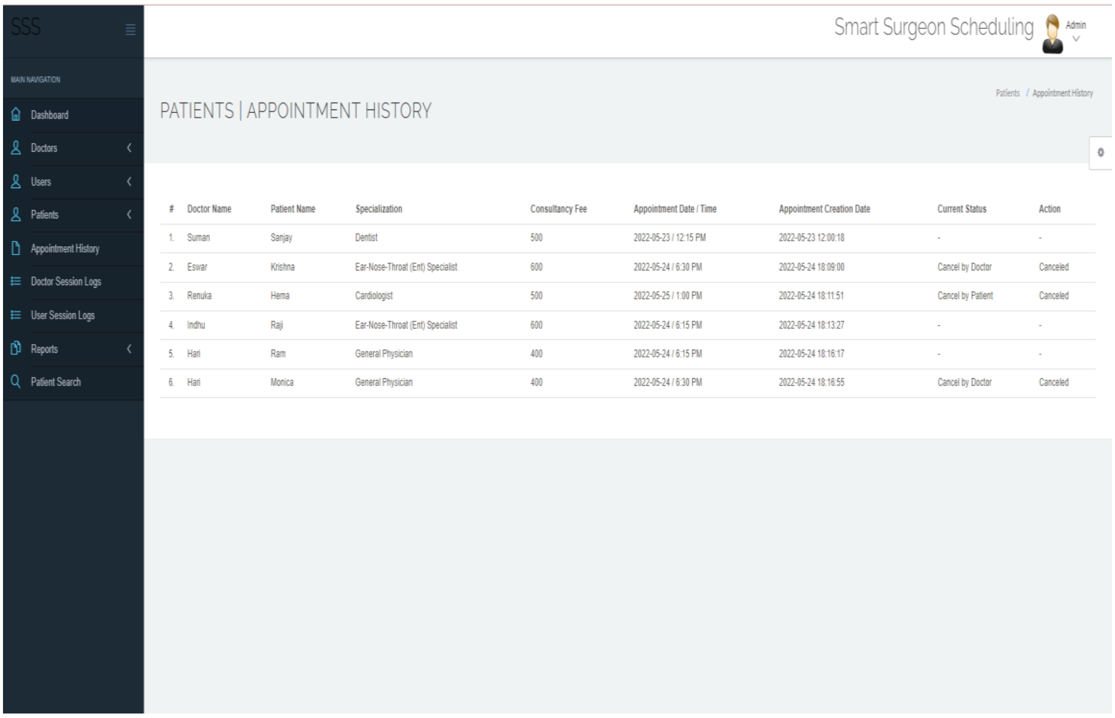


ADMIN | MANAGE USERS

Manage Users

#	Full Name	Adress	City	Gender	Email	Creation Date	Updation Date	Action
1.	Yogeswari Kanuri	Adavikolanu	Adavikolanu	female	yogeswari@gmail.com	2022-05-21 06:21:16		×
2.	Ramya	Ganapavaram	Ganapavaram	female	ramya@gmail.com	2022-05-21 06:21:37		×
3.	Satya Narayana	Undi	Undi	male	satya@gmail.com	2022-05-21 06:22:08		×
4.	Chandra Kala	PP Road	BHimavaram	female	chandra@gmail.com	2022-05-21 06:22:41		×
5.	Srinivas	Tanuku	Tanuku	male	srinivas@gmail.com	2022-05-21 06:23:14		×
6.	Manasvi	Tadepalligudem	Tadepalligudem	female	manasvi@gmail.com	2022-05-21 06:23:38		×
7.	Parnika	Padala	Pamarru	female	parnika@gmail.com	2022-05-21 06:24:06	2022-05-23 10:54:27	×
8.	Padma Sree	Palakol	Palakol	female	padma@gmail.com	2022-05-21 06:24:40		×
9.	Krishna	BHS Colony	Bhimavaram	male	krishna@gmail.com	2022-05-21 06:25:19		×
10.	Divya Sree	Vijayawada	Vijayawada	female	divya@gmail.com	2022-05-21 06:25:46		×
11.	Devi	Palakol	Palakol	female	devi@gmail.com	2022-05-21 06:26:10		×
12.	Ram	Vijayawada	Vijayawada	male	ram@gmail.com	2022-05-21 06:26:33	2022-05-23 10:55:12	×
13.	Ramanamma	Pamarru	Pamarru	female	ramanamma@gmail.com	2022-05-21 06:27:08		×
14.	Bhavani	Vijayawada	Vijayawada	female	bhavani@gmail.com	2022-05-21 06:27:56		×
15.	Balu	Kakinada	Kakinada	male	balu@gmail.com	2022-05-21 06:28:16		×
16.	Monica	Kalidindi	Kalidindi	female	monica@gmail.com	2022-05-21 06:28:59		×
17.	Harika	Kaala	Kaala	female	harika@gmail.com	2022-05-21 06:29:23		×

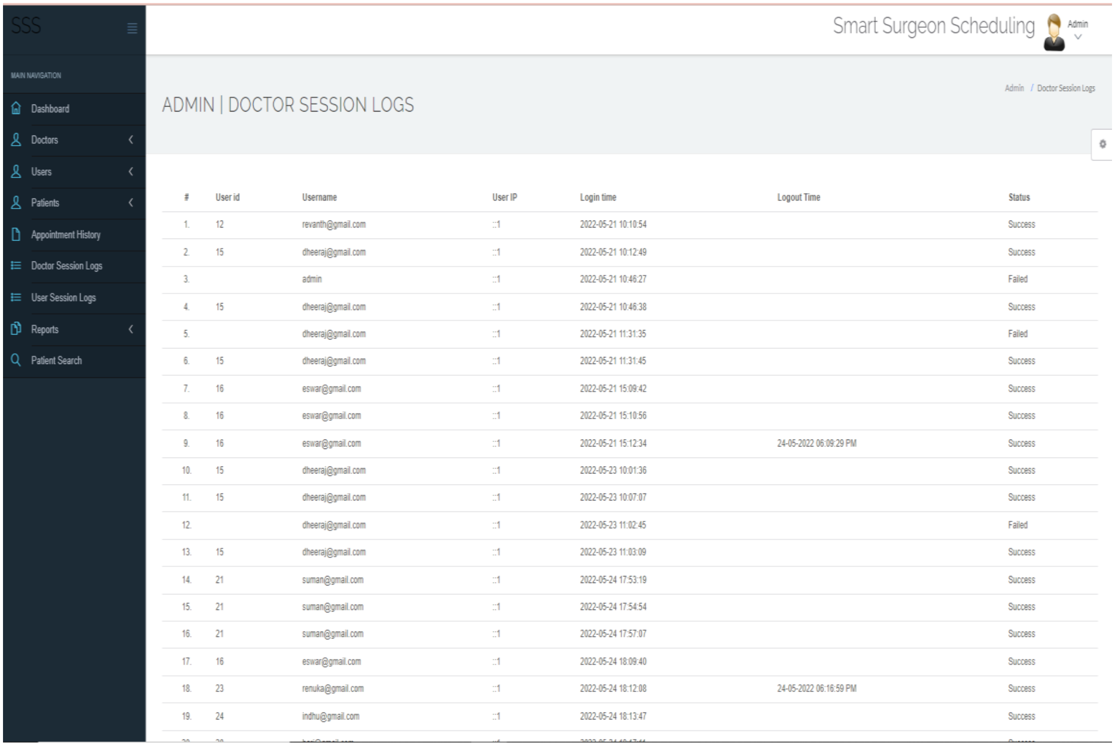
Figure 8.10 Manage User Details



PATIENTS | APPOINTMENT HISTORY

#	Doctor Name	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Suman	Sanjay	Dentist	500	2022-05-23 / 12:15 PM	2022-05-23 12:00:18	-	-
2.	Eswar	Krishna	Ear-Nose-Throat (Ent) Specialist	600	2022-05-24 / 6:30 PM	2022-05-24 18:09:00	Cancel by Doctor	Canceled
3.	Renuka	Hema	Cardiologist	500	2022-05-25 / 1:00 PM	2022-05-24 18:11:51	Cancel by Patient	Canceled
4.	Indhu	Raji	Ear-Nose-Throat (Ent) Specialist	600	2022-05-24 / 6:15 PM	2022-05-24 18:13:27	-	-
5.	Hari	Ram	General Physician	400	2022-05-24 / 6:15 PM	2022-05-24 18:16:17	-	-
6.	Hari	Monica	General Physician	400	2022-05-24 / 6:30 PM	2022-05-24 18:16:55	Cancel by Doctor	Canceled

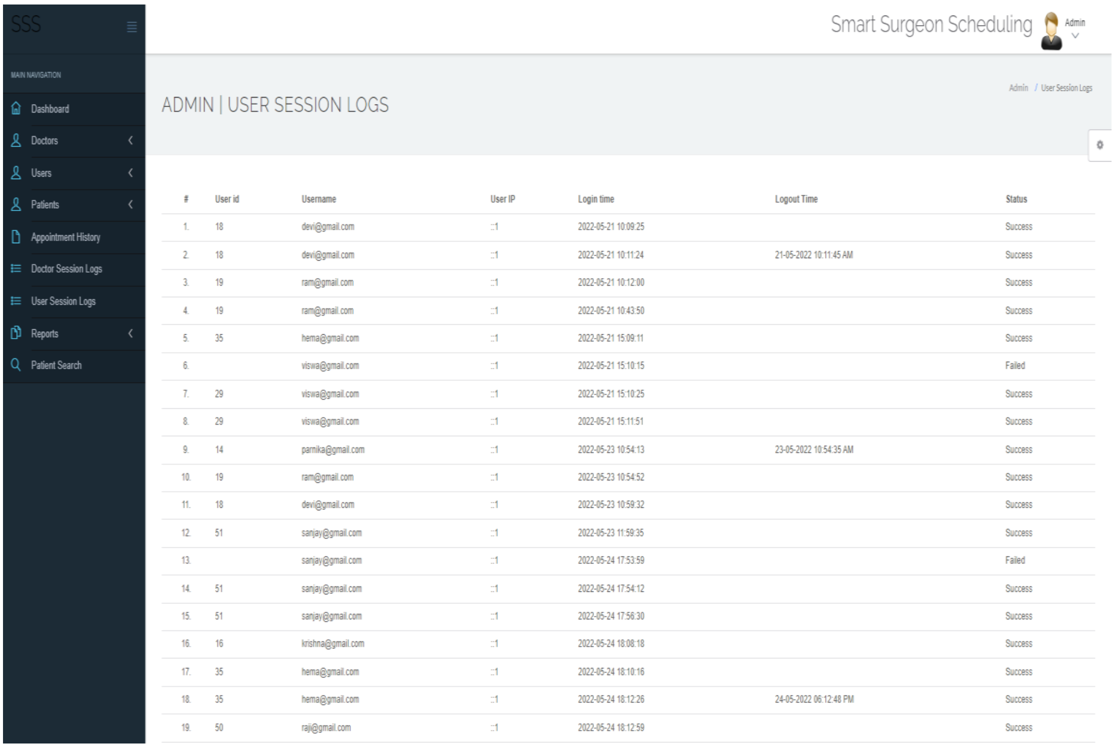
Figure 8.11 Appointment History



ADMIN | DOCTOR SESSION LOGS

#	User id	Username	User IP	Login time	Logout Time	Status
1.	12	revanth@gmail.com	::1	2022-05-21 10:10:54		Success
2.	15	dheeraj@gmail.com	::1	2022-05-21 10:12:49		Success
3.		admin	::1	2022-05-21 10:46:27		Failed
4.	15	dheeraj@gmail.com	::1	2022-05-21 10:46:38		Success
5.		dheeraj@gmail.com	::1	2022-05-21 11:31:35		Failed
6.	15	dheeraj@gmail.com	::1	2022-05-21 11:31:45		Success
7.	16	eswar@gmail.com	::1	2022-05-21 15:09:42		Success
8.	16	eswar@gmail.com	::1	2022-05-21 15:10:56		Success
9.	16	eswar@gmail.com	::1	2022-05-21 15:12:34	24-05-2022 06:09:29 PM	Success
10.	15	dheeraj@gmail.com	::1	2022-05-23 10:01:36		Success
11.	15	dheeraj@gmail.com	::1	2022-05-23 10:07:07		Success
12.		dheeraj@gmail.com	::1	2022-05-23 11:02:45		Failed
13.	15	dheeraj@gmail.com	::1	2022-05-23 11:03:09		Success
14.	21	suman@gmail.com	::1	2022-05-24 17:53:19		Success
15.	21	suman@gmail.com	::1	2022-05-24 17:54:54		Success
16.	21	suman@gmail.com	::1	2022-05-24 17:57:07		Success
17.	16	eswar@gmail.com	::1	2022-05-24 18:09:40		Success
18.	23	renuka@gmail.com	::1	2022-05-24 18:12:08	24-05-2022 06:16:59 PM	Success
19.	24	indhu@gmail.com	::1	2022-05-24 18:13:47		Success

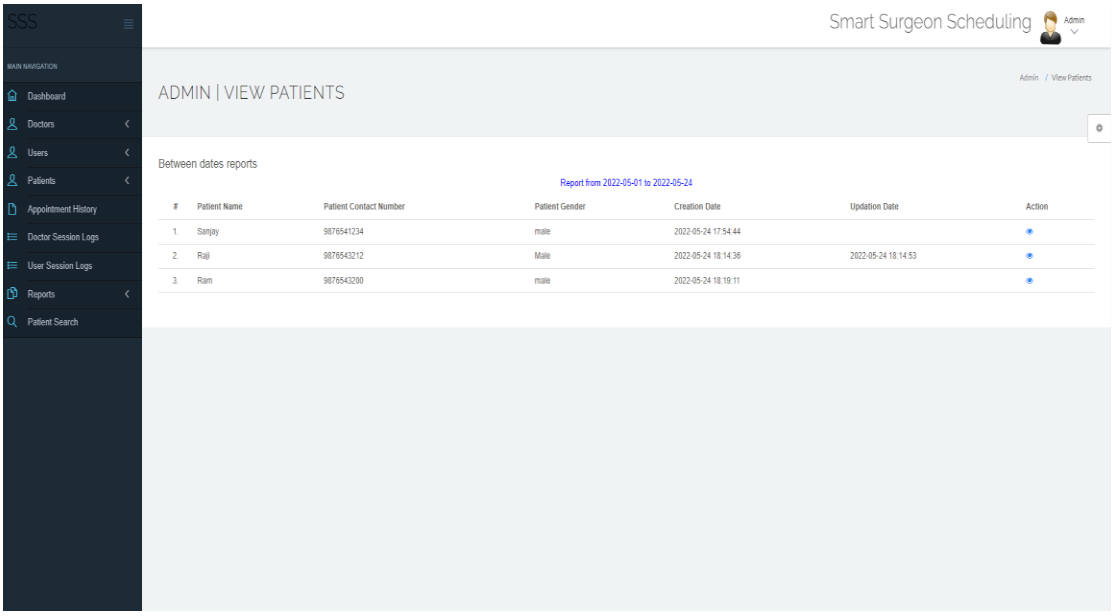
Figure 8.12 Doctor Session Logs



ADMIN | USER SESSION LOGS

#	User id	Username	User IP	Login time	Logout Time	Status
1.	18	devi@gmail.com	::1	2022-05-21 10:09:25		Success
2.	18	devi@gmail.com	::1	2022-05-21 10:11:24	21-05-2022 10:11:45 AM	Success
3.	19	ram@gmail.com	::1	2022-05-21 10:12:00		Success
4.	19	ram@gmail.com	::1	2022-05-21 10:43:50		Success
5.	35	hema@gmail.com	::1	2022-05-21 15:09:11		Success
6.		viswa@gmail.com	::1	2022-05-21 15:10:15		Failed
7.	29	viswa@gmail.com	::1	2022-05-21 15:10:25		Success
8.	29	viswa@gmail.com	::1	2022-05-21 15:11:51		Success
9.	14	parnika@gmail.com	::1	2022-05-23 10:54:13	23-05-2022 10:54:35 AM	Success
10.	19	ram@gmail.com	::1	2022-05-23 10:54:52		Success
11.	18	devi@gmail.com	::1	2022-05-23 10:59:32		Success
12.	51	sanjay@gmail.com	::1	2022-05-23 11:59:35		Success
13.		sanjay@gmail.com	::1	2022-05-24 17:53:59		Failed
14.	51	sanjay@gmail.com	::1	2022-05-24 17:54:12		Success
15.	51	sanjay@gmail.com	::1	2022-05-24 17:56:30		Success
16.	16	krishna@gmail.com	::1	2022-05-24 18:08:18		Success
17.	35	hema@gmail.com	::1	2022-05-24 18:10:16		Success
18.	35	hema@gmail.com	::1	2022-05-24 18:12:26	24-05-2022 06:12:48 PM	Success
19.	50	raji@gmail.com	::1	2022-05-24 18:12:59		Success

Figure 8.13 User Session Logs



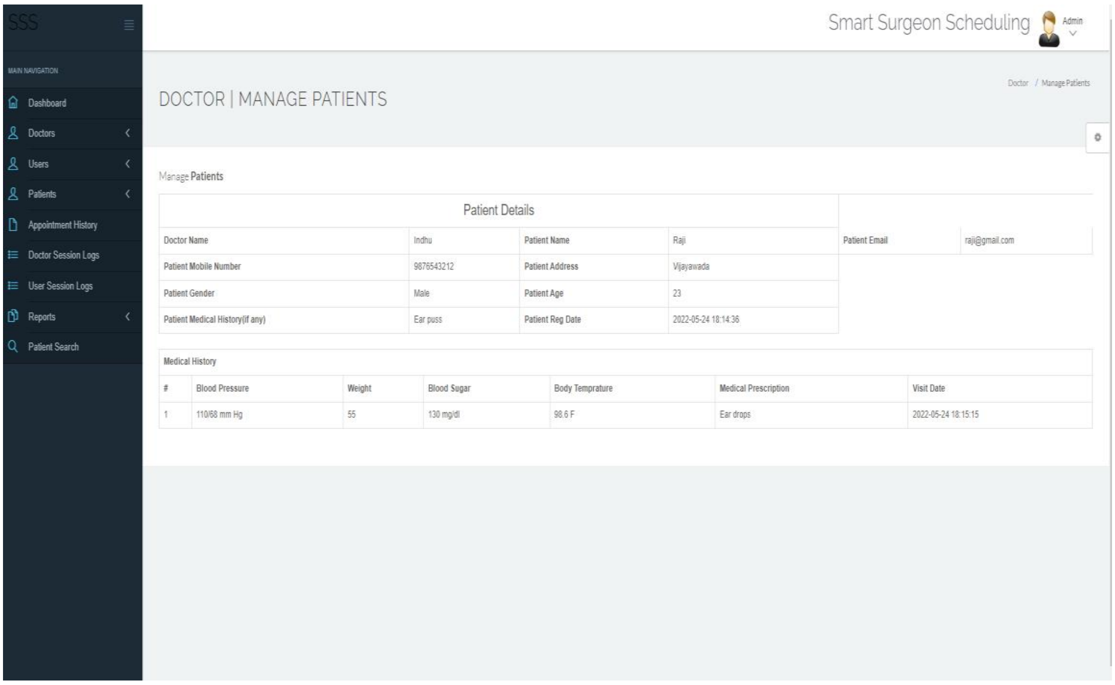
ADMIN | VIEW PATIENTS

Between dates reports

Report from 2022-05-01 to 2022-05-24

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Sanjay	9876541234	male	2022-05-24 17:54:44		●
2.	Raji	9876543212	Male	2022-05-24 18:14:36	2022-05-24 18:14:53	●
3.	Ram	9876543200	male	2022-05-24 18:19:11		●

Figure 8.14 View Patients



DOCTOR | MANAGE PATIENTS

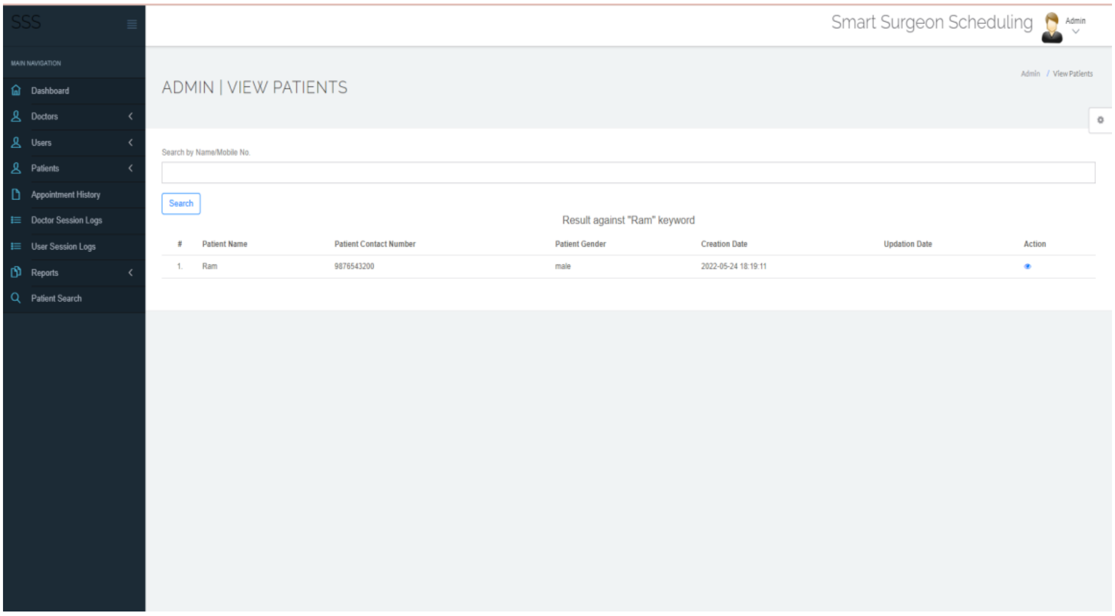
Manage Patients

Patient Details					
Doctor Name	Indhu	Patient Name	Raji	Patient Email	raji@gmail.com
Patient Mobile Number	9876543212	Patient Address	Vijayawada		
Patient Gender	Male	Patient Age	23		
Patient Medical History(if any)	Ear puss	Patient Reg Date	2022-05-24 18:14:36		

Medical History

#	Blood Pressure	Weight	Blood Sugar	Body Temprature	Medical Prescription	Visit Date
1	110/68 mm Hg	55	130 mg/dl	98.6 F	Ear drops	2022-05-24 18:15:15

Figure 8.15 Patient Report



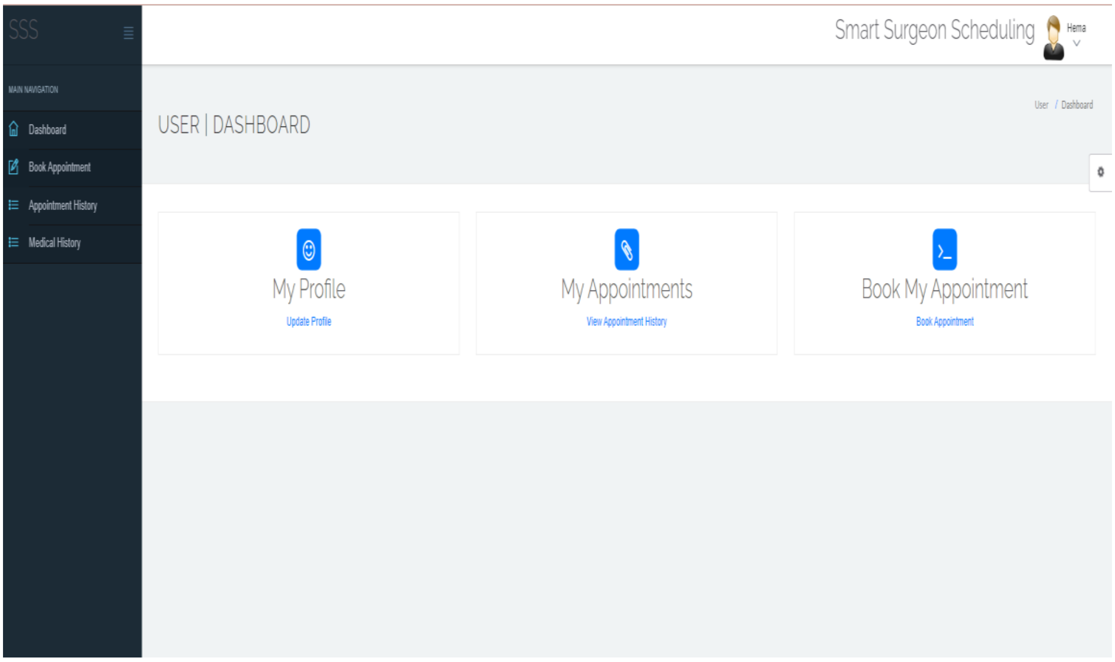
ADMIN | VIEW PATIENTS

Search by Name/Mobile No.

Result against "Ram" keyword

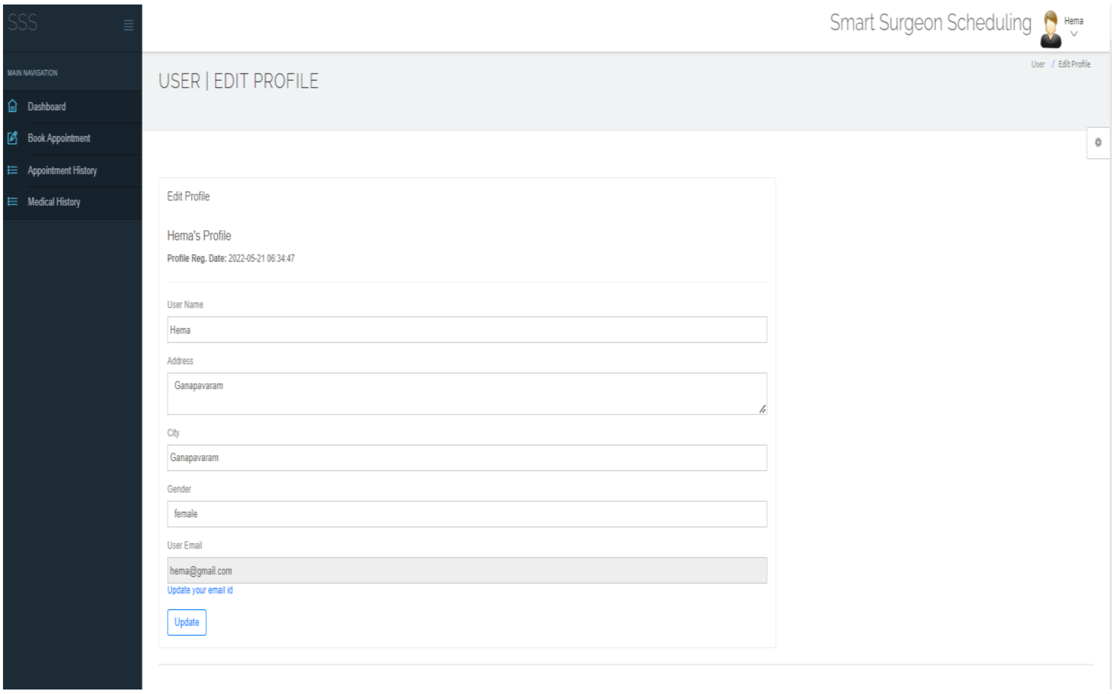
#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Ram	9876543200	male	2022-05-24 18:19:11		👁

Figure 8.16 Search Patients



USER | DASHBOARD

Figure 8.17 User Dashboard



USER | EDIT PROFILE

Edit Profile

Hema's Profile

Profile Reg. Date: 2022-05-21 06:34:47

User Name: Hema

Address: Ganapavaram

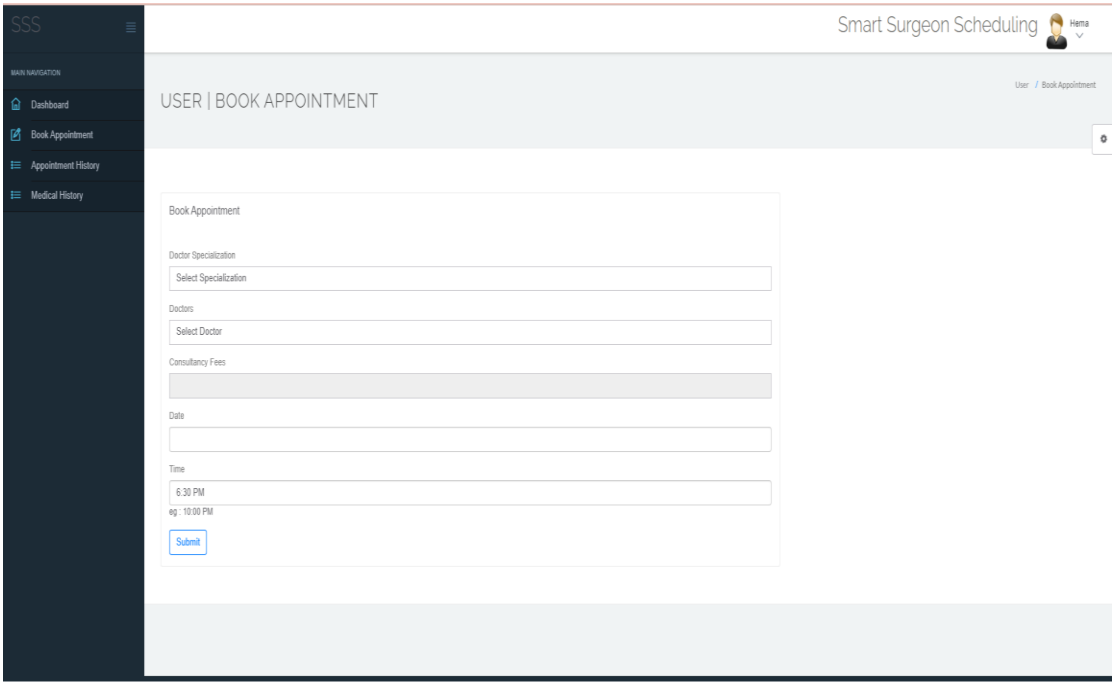
City: Ganapavaram

Gender: female

User Email: hema@gmail.com

Update your email id

Figure 8.18 Edit User Profile



USER | BOOK APPOINTMENT

Book Appointment

Doctor Specialization: Select Specialization

Doctors: Select Doctor

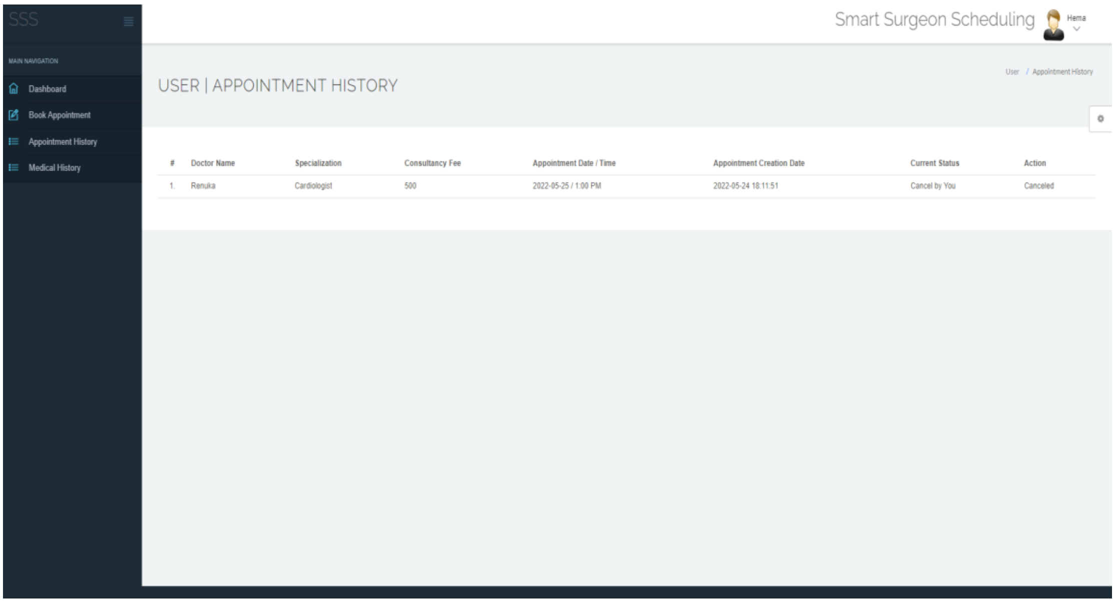
Consultancy Fees:

Date:

Time: 6:30 PM

eg : 10:00 PM

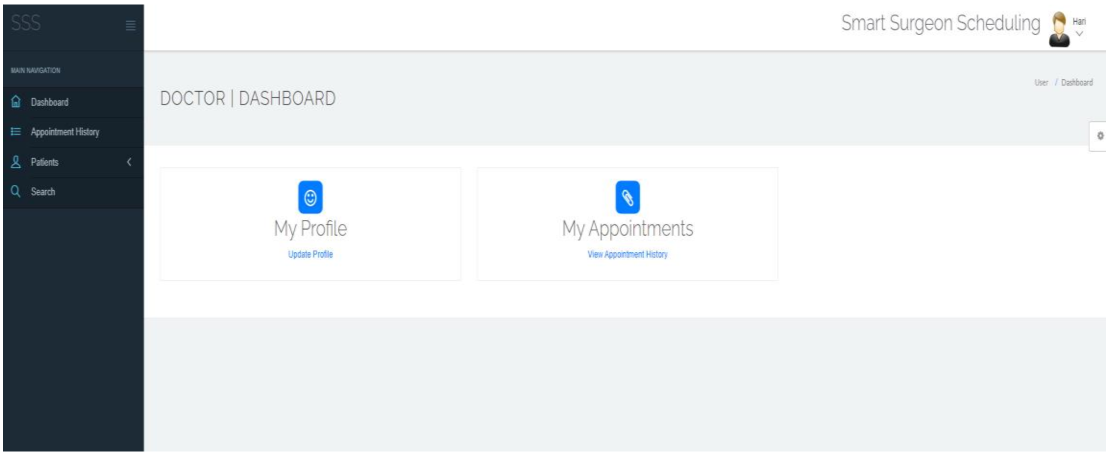
Figure 8.19 Book Appointment



USER | APPOINTMENT HISTORY

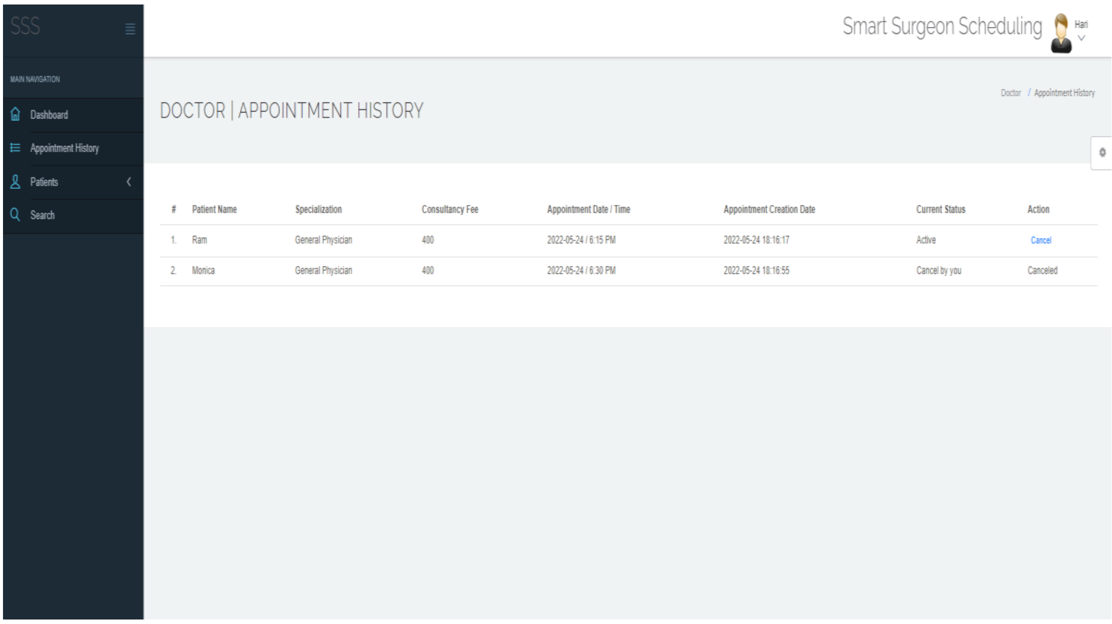
#	Doctor Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Renuka	Cardiologist	500	2022-05-25 / 1:00 PM	2022-05-24 18:11:51	Cancel by You	Canceled

Figure 8.20 Appointment Cancelled By Patient



DOCTOR | DASHBOARD

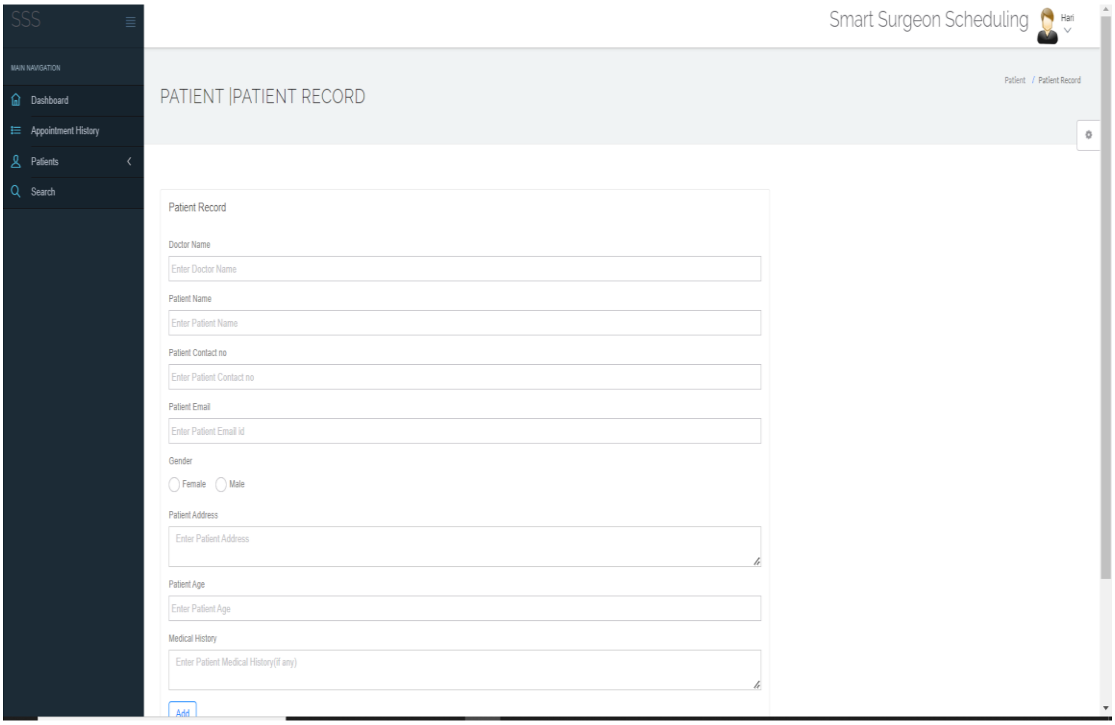
Figure 8.21 Doctor Dashboard



DOCTOR | APPOINTMENT HISTORY

#	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Ram	General Physician	400	2022-05-24 / 6:15 PM	2022-05-24 18:16:17	Active	Cancel
2.	Monica	General Physician	400	2022-05-24 / 6:30 PM	2022-05-24 18:16:55	Cancel by you	Canceled

Figure 8.22 Appointment History



PATIENT |PATIENT RECORD

Patient Record

Doctor Name: Enter Doctor Name

Patient Name: Enter Patient Name

Patient Contact no: Enter Patient Contact no

Patient Email: Enter Patient Email id

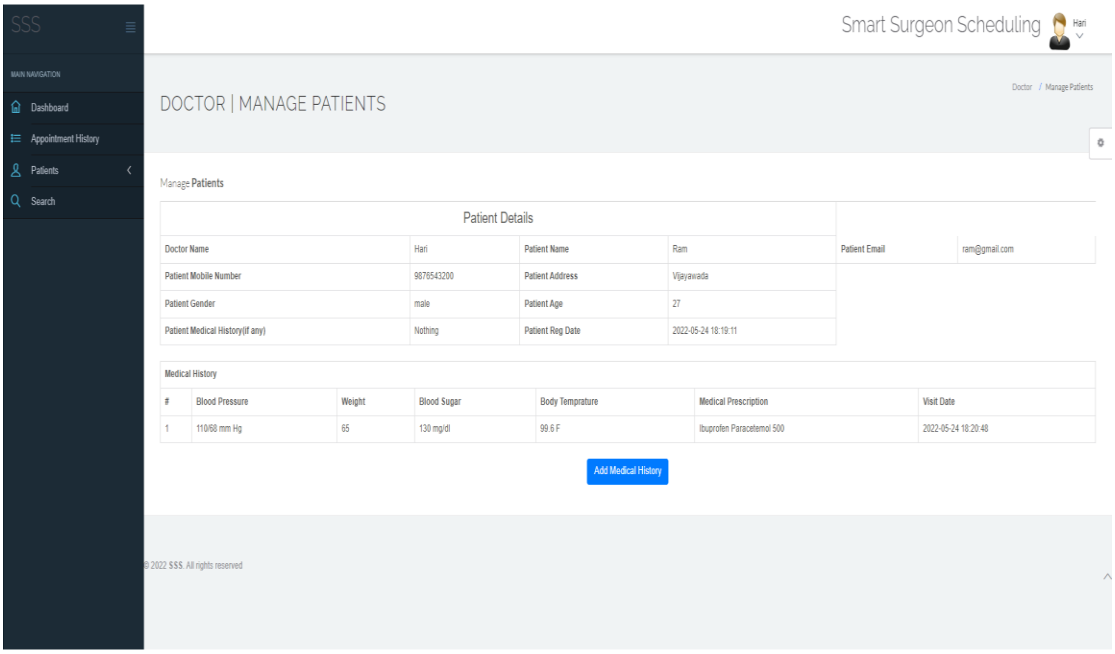
Gender: ○ Female ○ Male

Patient Address: Enter Patient Address

Patient Age: Enter Patient Age

Medical History: Enter Patient Medical History(if any)

Figure 8.23 Patient Record



DOCTOR | MANAGE PATIENTS

Manage Patients

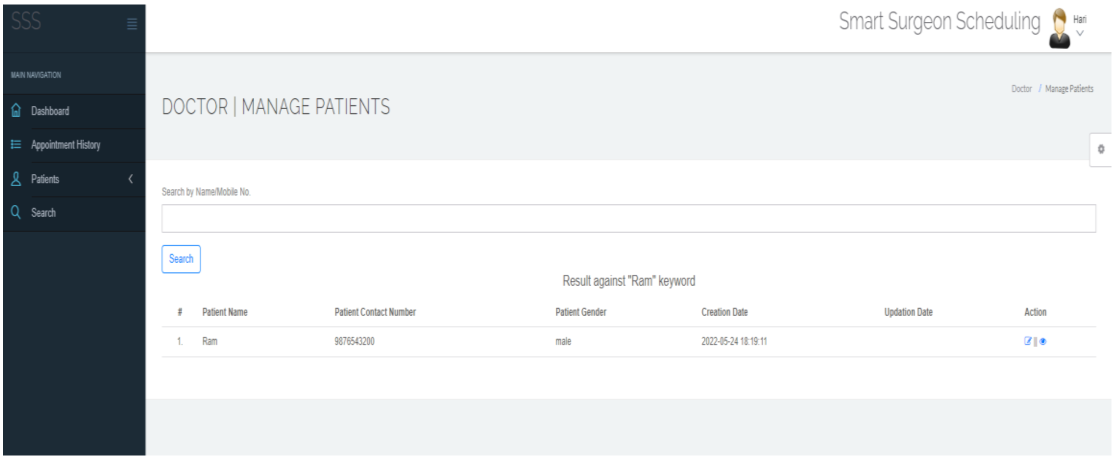
Patient Details

Doctor Name	Hari	Patient Name	Ram	Patient Email	ram@gmail.com
Patient Mobile Number	9876543200	Patient Address	Vijayawada		
Patient Gender	male	Patient Age	27		
Patient Medical History(if any)	Nothing	Patient Reg Date	2022-05-24 18:19:11		

Medical History

#	Blood Pressure	Weight	Blood Sugar	Body Temprature	Medical Prescription	Visit Date
1	110/68 mm Hg	65	130 mg/dl	99.6 F	Ibuprofen Paracetemol 500	2022-05-24 18:20:48

Figure 8.24 Manage Patient Details



DOCTOR | MANAGE PATIENTS

Result against "Ram" keyword

#	Patient Name	Patient Contact Number	Patient Gender	Creation Date	Updation Date	Action
1.	Ram	9876543200	male	2022-05-24 18:19:11		

Figure 8.25 Search Patients By Name

CHAPTER-9

CONCLUSION AND FUTURE SCOPE

9. Conclusion and Future Scope

Conclusion

This project has been a rewarding experience in more than one way. The entire project work has enlightened us in the following areas.

- We have gained an insight into the working of hospital. This represents a typical real world situation.
- Our understanding of database design has been strengthened this is because in order to generate the final reports of database designing has to be properly followed.
- Scheduling a project and adhering to that schedule creates a strong sense of time management.
- Sense of teamwork has developed and confidence of handling real life project has increased to a great extent.
- Initially, there were problem with the validation but with discussions, we were to implement validations.

Future Scope

- In future patients will be prompted of their appointments via SMS/email that will be reminded sent to them before their appointment date.
- Booked slots and vacant slots are display.

CHAPTER-10

REFERENCES

10. References

[1]. Buchbinder S. and Thompson J, 2010: An Overview of Healthcare Management.

[2]. Dr. Ariyaratne, 2010: “A private hospital management system”. Postgraduate Institute of Medicine, University of Colombo, Sri Lanka.

[3]. Farzandipour M, Sadoughi F and Meidani Z, 2010: Hospital Information Systems User Needs Analysis: A Vendor Survey.

[4]. Hosford, S.B. (2008). Hospital Progress in Reducing Error: The Impact of External Interventions. Hospital Topics, Winter 2008, 86, 1, 9-19.

[5]. Hospital Management and Information System. Quintegra Solutions (2006).

[6]. Hospital Management Information System, Gujarat Informatics System (2012)

[7]. Ndira SP, Rosenberger KD, Wetter T. (2008) Assessment of data quality of and Staff Satisfaction with an electronic health record system in a developing country (Uganda): a qualitative and quantitative comparative study.

[8]. Praveen K.A and Gomes L.A, 2006: “A Study of the Hospital Information System (HIS) In the Medical Records Department of A Tertiary Teaching Hospital” Journal of the Academy of Hospital Administration, Vol. 18, No. 1 (2006-01 - 2006-12).

[9]. Smith M. and Gert van der Pijl, 1999: “Developments in Hospital Management and Information Systems”. Tilburg University School of Economics, Netherlands.

[10]. Zwietering, M. H., I. Jongenburger, F. M. Rombouts, and K. van ’t Riet. 1990. "Modeling of the Bacterial Growth Curve". Applied and Environmental Microbiology.