

Database Management System

Mini Project Report

On

**HOSPITAL MANAGEMENT SYSTEM**

Submitted by

RAHUL RAMDAS CHINCHANKAR  
4SN21IS021

WILLIAM NARVIN FERNANDES  
4SN21IS031

INFORMATION SCIENCE AND ENGINEERING  
(Visvesvaraya Technological University)

Under the Guidance of  
Prof. Athmaranjan K  
Associate Professor



Department of Information Science and Engineering

**SRINIVAS INSTITUTE OF TECHNOLOGY**  
**(NAAC ACCREDITED)**

MANGALORE-574143, KARNATAKA

2023 – 2024

**SRINIVAS INSTITUTE OF TECHNOLOGY  
(NAAC ACCREDITED)  
MANGALORE -574143 – KARNATAKA**

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

**SRINIVAS GROUP  
CERTIFICATE**

*This is to certify that the project entitled “**STUDENT DATABASE MANAGEMENT**”, is an authentic record of the work carried out by*

**RAHUL RAMDAS CHINCHANKAR**

**4SN20IS021**

**WILLIAM NARVIN FERNANDES**

**4SN20IS031**

*as prescribed by Visvesvaraya Technological University, Belagavi, for V Semester B.E. in Artificial Intelligence & Machine Learning during the year 2022-2023.*

**SAMAGRA GNANA**

---

**Prof. Athmaranjan K**

**Project Guide**

---

**Prof. Sudharshan K**

**Head of the Department**

---

**Dr. Shrinivasa Mayya D**

**Principal**

**Name of the Examiners**

**1.**

**2.**

**Signature with Date**

# ABSTRACT

## **Project Description:**

The purpose of the project entitled as “Hospital Management System” is to computerize the Front Office Management of Hospital to develop software, which is user friendly simple, fast, and cost – effective. The main function of the system is to register and store patient details and doctor details and retrieve these details when required, and to manipulate these details meaningfully. The project is divided into three categories: Admin, Doctor, and Patient. In an overview of this web application, the patient can proceed with booking an appointment. For this, the user must select doctor, specialization, date, and time. After the selection of the doctor, the system displays the total fees for it. Besides, the patient can view his/her appointment history and prescription details. Administrator task includes managing doctors’ information, patient’s information. The complaints which are given by user will be referred by authorities. Here the doctor can schedule appointments and prescribe medication to the patient.

Organizations such as hospitals have to deal with a lot of patients regularly and hence a lot of data. Hence it is very important for a hospital to have a DBMS with a frontend that easily allows patients to book appointments and allows doctors or administrators to manage patient data.

# TABLE OF CONTENTS

CHAPTER No	TITLE	PAGE No
<b>1</b>	<b>INTRODUCTION</b>	
1.1	Problem Statement	1
1.2	Scope Of the Project	1
1.3	Organization of Report	2
<b>2</b>	<b>REQUIREMENT SPECIFICATION</b>	
2.1	Functional Requirements	3
2.2	Non-Functional Requirements	3
2.3	Hardware Requirements	4
2.4	Software Requirements	4
2.5	Software Tools Used	4
2.5.1	Front End Tool	4
2.5.2	Backend Database Used	5
<b>3</b>	<b>SYSTEM DESIGN</b>	
3.1	Flowchart	7
3.2	Database Table Design	8
3.3	Schema Diagram	12
3.4	E-R Diagram	12
3.5	Use-Case Diagram	13
3.6	Normalization	14
<b>4</b>	<b>IMPLEMENTATION</b>	
4.1	Creating Database Using XAMPP	16
4.2	Stored Procedure	18
4.3	Triggers	18
<b>5</b>	<b>SCREEN SHOTS</b>	<b>21</b>
<b>6</b>	<b>CONCLUSION AND SCOPE FOR FUTURE WORK</b>	<b>25</b>
	<b>REFERENCES</b>	<b>26</b>

## LIST OF FIGURES

FIGURE No	TITLE	PAGE No
3.1	Flowchart	14
3.2	Database Table Design	14
3.3	Schema Diagram	18
3.4	E-R Diagram	19
5.1	Home Page	27
5.2	Doctor Login Screen	27
5.3	Admin Login Page	28
5.4	Patient Login Screen	28
5.5	Patient's Portal Screen	29
5.6	Doctor's Portal Screen	29
5.7	Admin's Portal Screen	30

## LIST OF TABLES

Table No	TITLE	PAGE No
3.1	Admin	15
3.2	Patient Registration	15
3.3	Appointment	16
3.4	Contact	16
3.5	Doctor	16
3.6	Prescription	17

## ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any work would be incomplete without thanking the persons who made it perfect with their constant guidance and encouragement.

We take this opportunity to express my sincere thanks and indebtedness to our Project guide and mentor, **Prof. Athmaranjan K Associate Professor, Department of ISE**, for her support and guidance. Her vision and suggestions throughout the project period has been fundamental in the completion of the project.

We would like to extend my deep sense of acknowledgement to our **Project Coordinator Prof. Sowmya Assistant Professor, Department of ISE**, for his encouragement and timely advice provided to us during the project work.

We extend my warm gratitude to **Prof. Sudarshan K Head of the Department, Department of ISE**, for his constant support and advice that helped us to complete this project successfully.

We are extremely grateful to our beloved **Principal, Dr. Shrinivasa Mayya D** for his encouragement to come up with new ideas and advice to express them in a systematic manner.

We also like to thank all Teaching & Non-teaching staff of Srinivas Institute of Technology, Mangalore for their kind co-operation during the course of the work. Finally we are extremely thankful to our family and friends who helped us in our work and made the project a successful one.

**Rahul R Chinchankar (4SN21IS021)**

**William N Fernandes(4SN21IS031)**

## CHAPTER 1

### INTRODUCTION

Hospital Management System (HMS) is a web application that helps manage the day-to-day operations of a hospital. It enables efficient management of patient records, appointment schedules, prescription management. The main objective of this project is to develop a system that can automate and streamline the various processes in a hospital, making it more efficient and effective.

#### 1.1 Problem Statement

Manual hospital management systems are time-consuming and prone to errors, leading to reduced efficiency and patient satisfaction. There is a need for a computerized system that can automate and streamline various processes, such as patient registration, appointment scheduling, providing prescription. Additionally, there is a lack of effective communication between different departments and staff members, leading to delays and confusion in patient care.

#### 1.2 Scope of the Project

The scope of Hospital Management System includes:

- Create distinct product users based on their roles and permissions.
- Authenticate users at their login.
- Patient Record Management: It allows for the creation, storage, and retrieval of patient records, including personal information, medical history, and treatment details.
- Facility to book appointment with available doctor.
- A status page for all users to view their appointment.
- Facility to check the prescription and appointment history.
- Facility to cancel the appointment.
- Providing interface to add or delete doctor to admin.
- Providing interface to view patients, doctors, and queries to admin



### **1.3 Organization of Report**

The contents of the project report are organized in a well-structured manner where chapter 1 contains the Introduction, Problem statement that gives the introduction to the problem domain on which the project is based. Chapter 2 will give the detailed description on the requirement specification that are Functional Requirements, Non-Functional Requirements, Hardware Requirements, Software Requirements, Software tools used in Front-end and Back-end. Chapter 3 gives the System Design that consists of flow chart which represents workflow, database table designs, schema diagram that represents structure behind data organization, E-R diagram shows the relationships of entity sets stored in a database, use-case diagram that is a representation of a user's interaction with the system and normalization which is the process of structuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. Chapter 4 shows the Implementation in three sections. Section 1 is Creating database using Xampp which shows to create and use database, to create tables, insert values and view table. Section 2 includes Stored procedures that is a prepared SQL code that can be reused over and over again. Section 3 includes Triggers which is a special type of stored procedure that automatically executes when an event occurs in the database server. Chapter 5 displays all the screenshots of the developed project with description. Every page working screenshot is displayed. Chapter 6 gives the conclusion by solving the problem faced by manual systems therefore by developing HMS and future scope of the developed project.

## CHAPTER 2

### REQUIREMENT SPECIFICATION

#### 2.1 Functional Requirements

A functional specification (functional specifications document (FSD), functional requirements specification) in systems engineering and software development is a document that specifies the functions that a system or component must perform. A functional specification is the more technical response to a matching requirements document.

A functional specification does not define the inner workings of the proposed system; it does not include the specification of how the system function will be implemented. Instead, it focuses on what various outside agents (people using the program, computer peripherals, or other computers, for example) might "observe" when interacting with the system.

Functional requirement for HMS includes:

- Admin: for adding doctors
- Doctor: for accepting/cancelling appointments and for giving the prescription
- Patient: for booking appointment

#### 2.2 Non-Functional Requirements

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

**Efficiency Requirement:** When a hospital management system will be implemented, patient will easily access prescription and booking appointment will be very faster.

**Reliability Requirement:** The system should accurately perform patient registration, adding/deleting doctor.

**Usability Requirement:** The system is designed for a user-friendly environment so that patient can perform the various tasks easily and in an effective way.

**Performance Requirements:** Performance of the system should be fast and accurate.

#### 2.3 Hardware Requirements

The minimum hardware configuration required for developing the proposed software is given below:

- RAM: 256 MB RAM
- Hard disk: 40 GB

- Processor: 1.2GHz Processor
- Operating System: Windows 10/8/7

## 2.4 Software Requirements

- Front End
  - Programming language: HTML, CSS, JS
  - IDE: Visual Studio codes
- Back End
  - XAMPP Server
  - PHP
  - MySQL

## 2.5 Software Tools Used

Hospital management system is designed using visual studio as front-end user interface design tool and MySQLServer 8.0 at backend for creating tables and storing related information.

### 2.5.1 Front End Tool

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer). Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic

.NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

JavaScript is a programming language that is primarily used to create interactive and dynamic websites. It is a client-side scripting language, which means that it is executed by the web browser on the user's computer rather than on a server. JavaScript can be used to create complex interactions, animations, and dynamic data updates on websites, and it is also used for server-side programming with technologies like Node.js. It is often used in combination with HTML and CSS to create web pages and web applications.

## **2.5.2 Back End Database Used**

XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP and Perl, and it allows you to build Word Press site offline, on a local web server on your computer. This simple and lightweight solution works on Windows, Linux, and Mac – hence the “cross-platform” part. XAMPP's ease of deployment means a stack can be installed quickly and simply on an operating system by a developer, with the advantage that common add-in applications such can also be installed with similar ease using.

MySQL is an open-source relational database management system. It is based on the structure query language (SQL), which issued for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL. MySQL can be used for a variety of applications, but is most commonly found on Web servers. A website that uses MySQL may include Web pages that access information from a database. Many database-driven websites that use MySQL also use a Web scripting language like PHP to access information from the database. MySQL commands can be incorporated into the PHP code, allowing part or all of a Web page to be generated from database information. Because both MySQL and PHP are both open source (meaning they are free to download and use), the PHP/MySQL combination has become a popular choice for database-driven websites.

## CHAPTER 3

### SYSTEM DESIGN

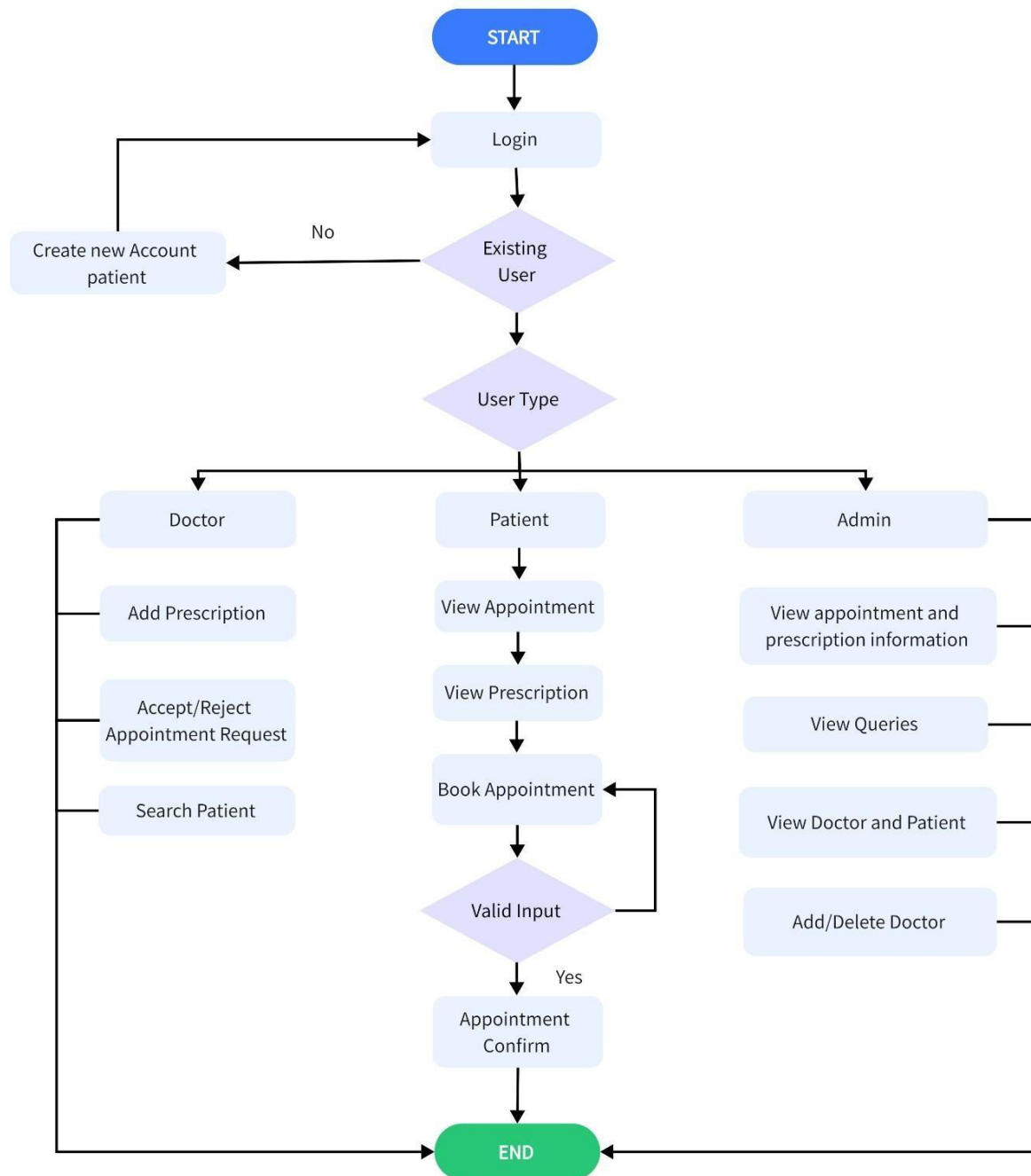
System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system. System analysis and design deal with planning the development of information systems through understanding and specifying in detail what a system should do and how the components of the system should be implemented and work together. System design solve business problems through analyzing the requirements of information systems and designing such systems by applying analysis and design techniques. System analysis and design is the most essential phase in the development of a system since the logical system design arrived at as a result of systems analysis which is in turn converted into physical system design.

#### 3.1 Flow Chart

A flowchart is a type of diagram that represents an algorithm, workflow or process. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Like other types of diagrams, they help visualize what is going on and thereby help understand a process, and perhaps also find less-obvious features within the process, like flaws and bottlenecks. With proper design and construction, it communicates the steps in a process very effectively and efficiently.

Figure 3.1 depicts the flow chart of HMS which shows various processing steps based on different events, actions, and conditions.

## Hospital Management System

*Fig 3.1: Flowchart of HMS***3.2 Database Table Design**

The basic database unit is the table. A table is a unit consisting of rows of related information. Each row consists of fields of information where data is stored. Field attributes include

information and rules that govern the data stored in the field. The field attributes and rules may limit the type of data stored in the field.

A field may be defined as a key or may be limited by rules requiring specific masks, such as a field may be limited to dates, formatted numbers like telephone numbers, or be limited to a specific number of characters. The database schema contains these rules. Database tables used in HMS are shown below.

**Table 3.1: Admin**

<b>Name</b>	<b>Data Type</b>	<b>Constraints</b>
Username	varchar(50)	Not Null
Password	varchar(30)	Not Null

The above admin table is used to store admin login details.

**Table 3.2: Patient Registration**

<b>Name</b>	<b>Datatype</b>	<b>Constraints</b>
pid	int(11)	Primary Key
fname	varchar(20)	Not Null
lname	varchar(20)	Not Null
gender	varchar(10)	Not Null
email	varchar(30)	Unique
contact	varchar(10)	Not Null
password	varchar(30)	Not Null
cpassword	varchar(30)	Not Null

The above patient table is used to store patient registration details.

**Table 3.3: Appointment**

<b>Name</b>	<b>Datatype</b>	<b>Constraints</b>
pid	int(11)	Foreign Key
ID	int(11)	Primary Key
fname	varchar(20)	Not Null
lname	varchar(20)	Not Null
gender	varchar(10)	Not Null
email	varchar(30)	Not Null
contact	varchar(10)	Not Null
doc_id	int	Foreign Key
doctor	int(5)	Not Null
docFees	varchar(30)	Not Null
appdate	date	Not Null
apptime	time	Not Null
userStatus	int(5)	Not Null
doctorStatus	int(5)	Not Null

The above appointment table is used to store appointment details.

**Table 3.4: Contact**

<b>Name</b>	<b>Datatype</b>	<b>Constraints</b>
name	varchar(30)	Not Null
email	text	Not Null
contact	varchar(10)	Not Null
message	varchar(200)	Not Null

The above contact table is used to store queries details.



**Table 3.5: Doctor**

<b>Name</b>	<b>Datatype</b>	<b>Constraints</b>
doctor_id	INT	Primary Key
username	varchar(50)	Unique
password	varchar(50)	Not Null
doctorname	varchar(225)	Not Null
email	varchar(50)	Not Null
spec	varchar(50)	Not Null
docfees	int(10)	Not Null

**Table 3.6: Prescription**

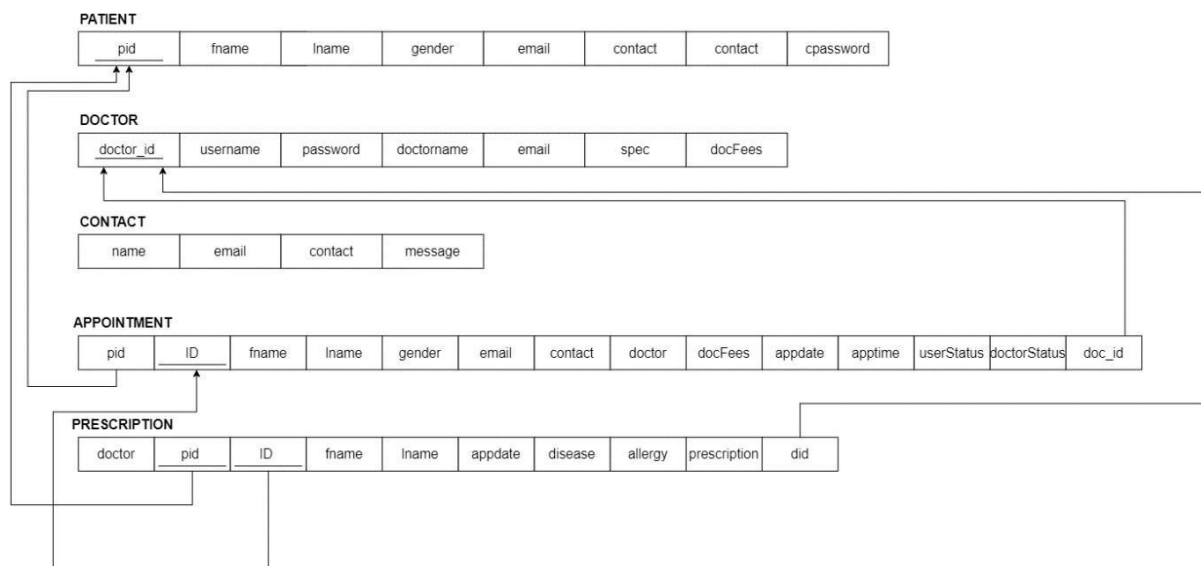
<b>Name</b>	<b>Datatype</b>	<b>Constraints</b>
did	int	Foreign Key
doctor	varchar(50)	Not Null
pid	int(11)	Primary Key
ID	int(11)	Primary Key
fname	varchar(50)	Not Null
lname	varchar(50)	Not Null
apppdate	date	Not Null
apptime	time	Not Null
disease	varchar(250)	Not Null
allergy	varchar(250)	Not Null
prescription	varchar(1000)	Not Null

The above prescription table is used to store prescription detail.

### 3.3 Schema Diagram

A schema is the structure behind data organization. It is a visual representation of how different table relationships enable the schema's underlying mission business rules for which the database is created. In a schema diagram, all database tables are designated with unique columns and special features, e.g., primary/foreign keys or not null, etc. Formats and symbols for expression are universally understood, eliminating the possibility of confusion. The table relationships also are expressed via a parent table's primary key lines when joined with the child table's corresponding foreign keys.

Schema diagrams have an important function because they force database developers to transpose ideas to paper. This provides an overview of the entire database, while facilitating future database administrator work. Fig 3.3 shows the schema diagram of Hospital Management System.

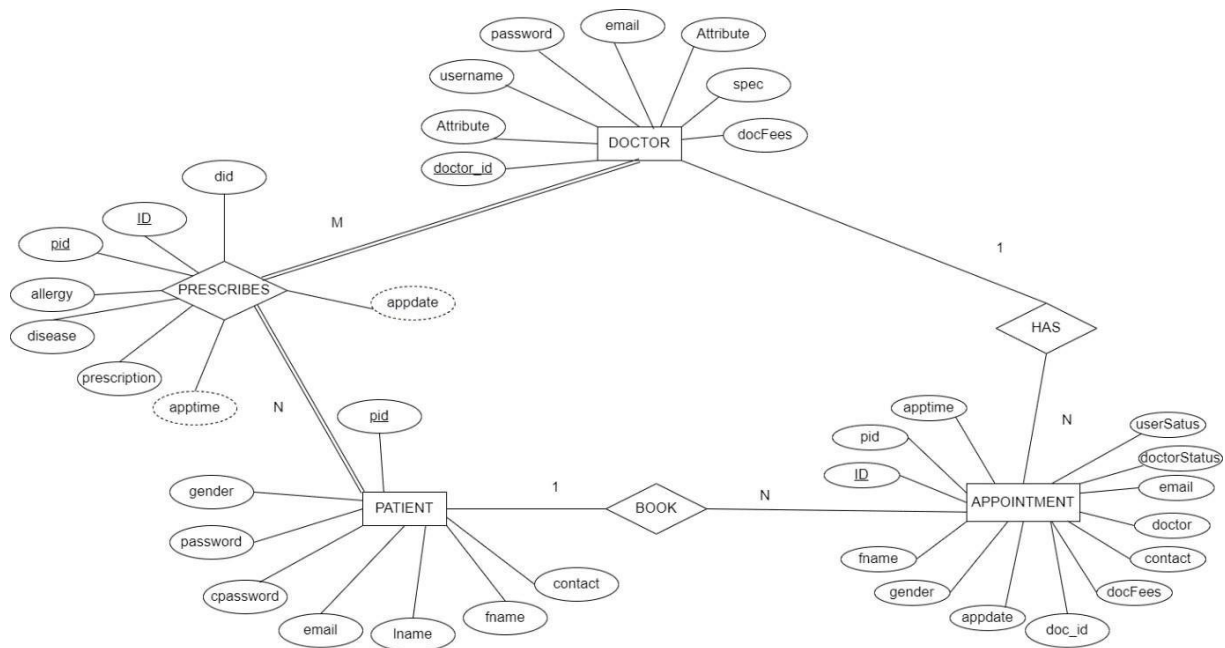


*Fig 3.3: Schema Diagram of HMS*

### 3.4 ER Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases. ER diagrams are used to sketch out the design of a database.

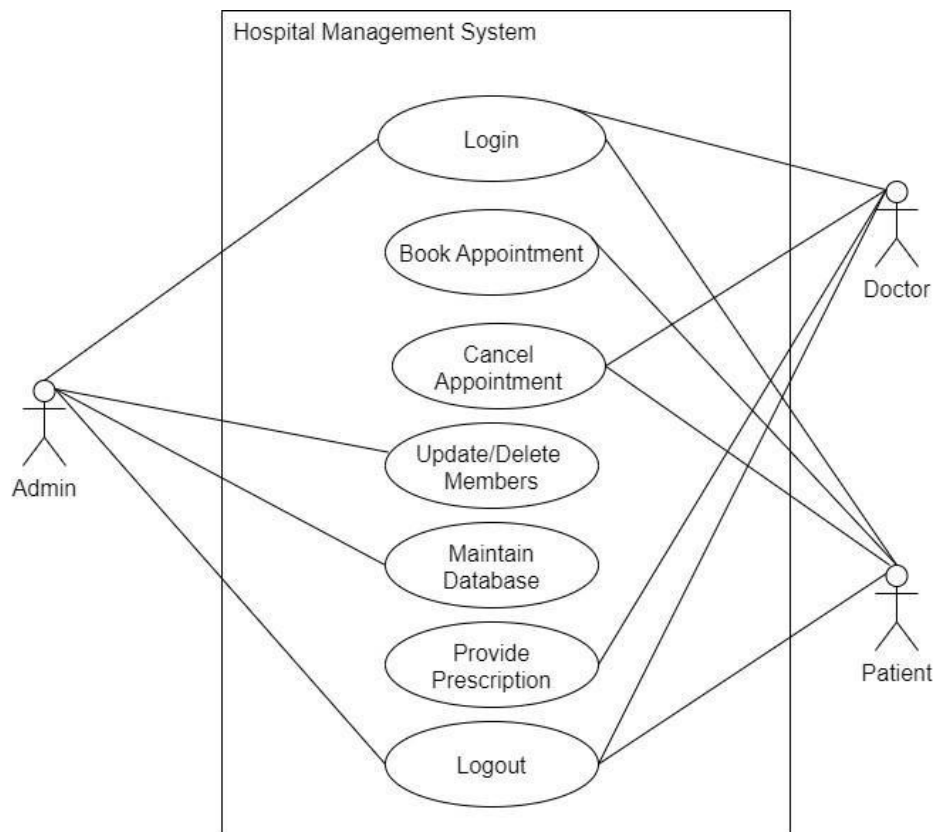
Fig 3.3 depicts the ER diagram of HMS. It shows various entities, their attributes, relationships with other entities, cardinality ratios between entities and participation constraints used in designing HMS database.



**Fig 3.4: ER diagram of HMS**

### 3.5 Use case Diagram

Use case diagrams referred as a Behavior model or diagram. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. All user describe in use case are actors and the functionality as action of system. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Users interacting with application are shown outside with stickman symbol. Fig 3.4 depicts the use case diagram of Hospital Management System.



**Fig 3.5:** Use case diagram of HMS

### 3.6 Normalization

Database normalization is the process of structuring a relational database [clarification needed] in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. It was first proposed by Edgar F. Codd as part of his relational model. Normalization entails organizing the columns (attributes) and tables (relations) of a database to ensure that their dependencies are properly enforced by database integrity constraints. It is accomplished by applying some formal rules either by a process of synthesis (creating a new database design) or decomposition (improving an existing database design).

#### First normal form (1NF):

First normal form (1NF) is a property of a relation in a relational database. A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain

**Second normal form (2NF):**

A relation is said to be in second normal form if it is already in first normal form and it has no partial dependency.

**Third normal form (3NF):**

It is a normal form that is used in normalizing a database design to reduce the duplication of data and ensure referential integrity by ensuring that:

- The entity is in second normal form.
- No non-prime (non-key) attribute is transitively dependent on any key i.e. no non-prime attribute depends on other non-prime attributes. All the non-prime attributes must depend only on the candidate keys.

## CHAPTER 4

### IMPLEMENTATION

System implementation is the important stage of project when the theoretical design is tuned into practical system. Systems implementation is the process of defining how the information system should be built (i.e., physical system design), ensuring that the information system is operational and used, ensuring that the information system meets quality standard (i.e., quality assurance).

A key difference between System Implementation and all other phases of the lifecycle is that all project activities up to this point have been performed in safe, protected, and secure environments, where project issues that arise have little or no impact on day-to-day business operations.

#### 4.1 Creating Database Using XAMPP

Creating database and different tables used in Hospital Management System are given below.

Different tables used in HMS are given below.

Create new database & use newly created database:

1. Create table patient;

				pid	fname	lname	gender	email	contact	password	cpassword
<input type="checkbox"/>				19	HARISH	KANNUR	Male	harishkannur@gmail.com	9876543211	12345678	12345678
<input type="checkbox"/>				20	RAJA	ADYAR	Male	raja@yahoo.com	7896543211	12345678	12345678
<input type="checkbox"/>				21	ALICE	VALANCIA	Female	alice121@gmail.com	6785949399	12345678	12345678
<input type="checkbox"/>				23	AZAR	VALACHIL	Male	azar21@gmail.com	8978675438	12345678	12345678
<input type="checkbox"/>				24	DEEPIKA	AGARWAL	Female	deepag@outlook.com	8796785678	12345678	12345678

*Fig 4.1: Patient Table*

The above *Fig 4.1* contains the patient details such as patient id, fname, gender, email, contact, password

## 2. Create table appointmenttb;

pid	ID	fname	lname	gender	email	contact	doctor	doc_id	docFees	apptime	apptime	userStatus	doctorStatus
19	23	HARISH	KANNUR	Male	harishkannur@gmail.com	9876543211	abraham	1	444	2023-01-27	14:00:00	1	0
20	24	RAJA	ADYAR	Male	raja@yahoo.com	7896543211	Saithejas	5	1000	2023-01-31	16:00:00	1	1
24	25	DEEPIKA	AGARWAL	Female	deepag@outlook.com	8796785678	sarala	6	550	2023-02-01	10:00:00	1	1
21	26	ALICE	VALANCIA	Female	alice121@gmail.com	6785949399	alicia	2	300	2023-01-29	08:00:00	1	1
20	27	RAJA	ADYAR	Male	raja@yahoo.com	7896543211	anuj	3	370	2023-01-28	14:00:00	1	1

Fig 4.2: Appointment Table

The above Fig 4.2 contains all the Appointment details such as fname, lname, gender, email, contact, doctor, docfees, apptime, apptime.

## 3. Create table contact;

name	email	contact	message
Nihal	nihal@gmail.com	7856453533	Please Can you give is there 3 units of blood in y...
Anu	anu@gmail.com	9999656448	Can you please provide the contact of general doct...

Fig 4.3: Contact Table

The above Fig 4.3 contains all the query details like name, email, contact and message.

## 4. Create table doctb;

	username	password	doctorname	email	spec	docFees
<input type="checkbox"/> Edit Copy Delete	abraham	12345678	Dr Abraham T	abrhm@sanjeevani.com	Gastroenterologist	444
<input type="checkbox"/> Edit Copy Delete	alicia	12345678	Dr Alicia D	alicia@sanjeevani.com	Oncologist	300
<input type="checkbox"/> Edit Copy Delete	anuj	12345678	Dr Anuj K	anuj@sanjeevani.com	Pediatrician	370
<input type="checkbox"/> Edit Copy Delete	pushparaj	12345678	Dr Pushparaj	pushparj@sanjeevani.com	General	671
<input type="checkbox"/> Edit Copy Delete	Saithejas	12345678	Dr Sai Thejas	saithejas@sanjeevani.com	Neurologist	1000
<input type="checkbox"/> Edit Copy Delete	sarala	12345678	Dr Sarala	sarala@sanjeevani.com	Gynecologist	550

Fig 4.4: Doctor Table

The above Fig 4.4 contains all the doctor details such as username, password, doctorname, email, spec, docFees.

## 5. Create table prestb;

did	doctor	pid	ID	fname	lname	appdate	apptime	disease	allergy	prescription
1	abraham	19	23	HARISH	KANNUR	2023-01-27	14:00:00	Gastroesophageal reflux disease	No	Drug Name: Omeprazole Dosage: 20 mg Route of Admin...
6	sarala	24	25	DEEPIKA	AGARWAL	2023-02-01	10:00:00	Genital herpes	Penicillin	Drug Name: Acyclovir Dosage: 400 mg Route of Adm...

Fig 4.5: Prescription Table

The above Fig 4.5 contains all the prescription details such as disease, allergy, prescription.

## 6. Create table adminbt;

username	password
admin	admin123

Fig 4.6: Admin Table

The above Fig 4.6 contains admin credentials which as username and password.

## 4.2 Stored Procedure

A stored procedure is a prepared SQL code that can be reused over and over again. So, if an SQL query needs to be written over and over again, save it as a stored procedure, and then just call it to execute it. It is also possible to pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

Stored procedure used in HMS is as follows

pid	fname	lname	gender	email	contact	password	cpassword
19	HARISH	KANNUR	Male	harishkannur@gmail.com	9876543211	12345678	12345678
20	RAJA	ADYAR	Male	raja@yahoo.com	7896543211	12345678	12345678
21	ALICE	VALANCIA	Female	alice121@gmail.com	8765432109	12345678	12345678
23	AZAR	VALACHIL	Male	azar21@gmail.com	8978675438	12345678	12345678
24	DEEPIKA	AGARWAL	Female	deepag@outlook.com	8796785678	12345678	12345678
25	RAMANAN	IYYER	Male	raman@gmail.com	9876577556	12345678	12345678

Fig 4.2.1: Stored Procedure

The above Fig 4.2.1, displays stored procedure output in the back-end.



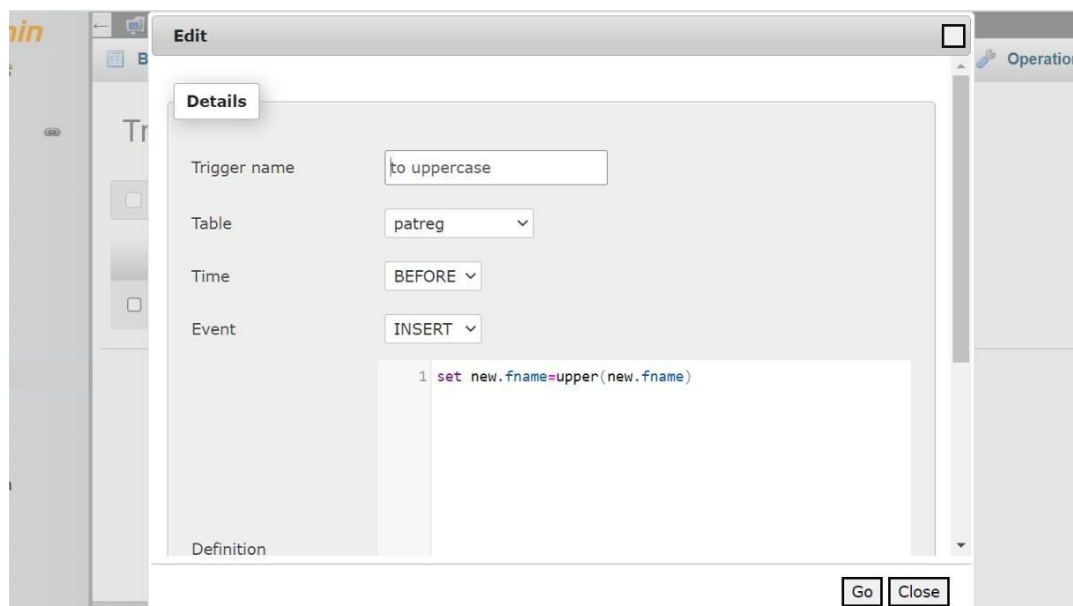
### 4.3 Triggers

A trigger is a special type of stored procedure that automatically executes when an event occurs in the database server. DML triggers execute when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view. These triggers fire when any valid event is fired, regardless of whether or not any table rows are affected.

#### Trigger used in HMS is as follows

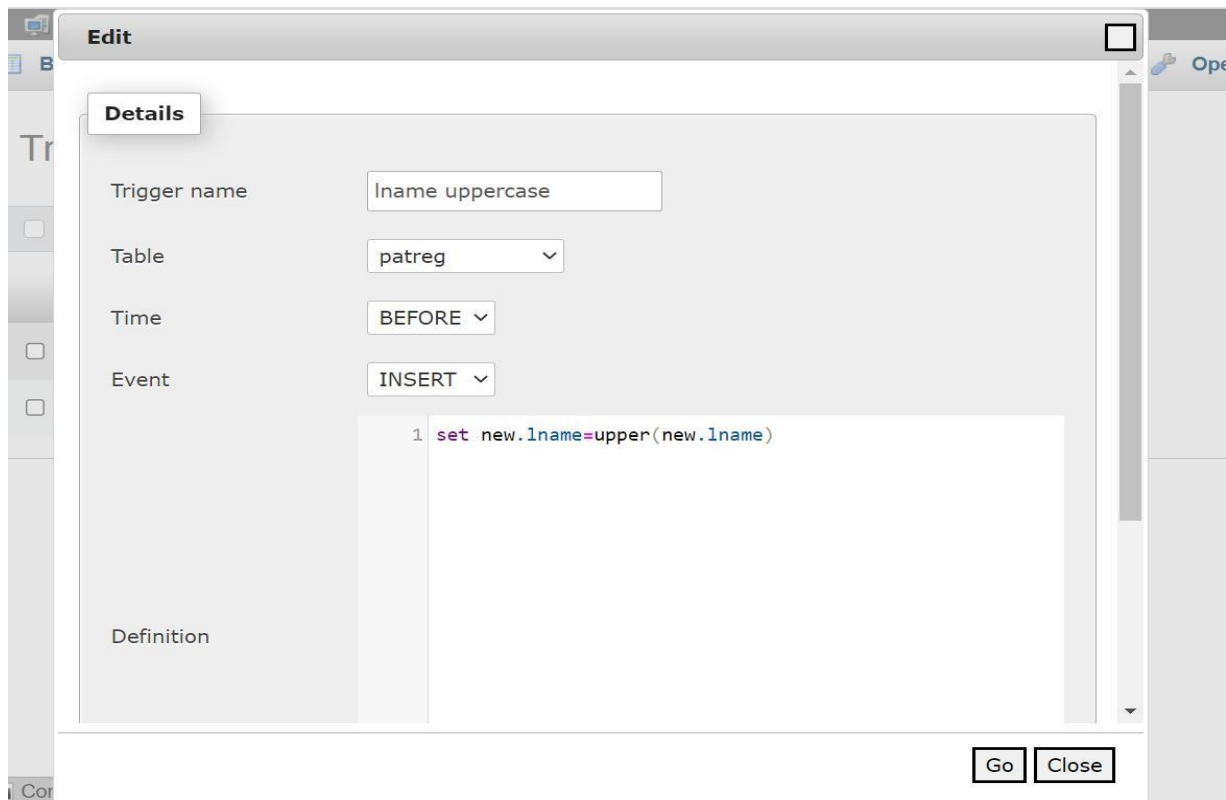
The trigger would be set to automatically convert any new first and last names entered into the database table to uppercase. This would be done by using the appropriate SQL command, such as the "UPPER" function, to convert the first and last name field to uppercase when a new record is inserted into the table.

The trigger is set to be activated when an INSERT statement is executed on the table, and it would automatically convert the first and last name field to uppercase before the data is inserted into the table. This would ensure that all first and last names in the table are in uppercase, and would prevent any issues that may arise from having mixed-case first and last names in the table.



**Fig 4.3.1:** Fname Trigger

The above Fig 4.3.1 contains the trigger to convert First name into upper case before adding it to the table.



**Fig 4.3.2:** *Lname Trigger*

The above Fig 4.3.2 contains the trigger to convert Last name into upper case before adding it to the table.

## CHAPTER 5

### SCREEN SHOTS

This chapter allows the students to understand how the Hospital Management System works. This entire chapter is focused to guide the users to make their experience of using the software in a better way.

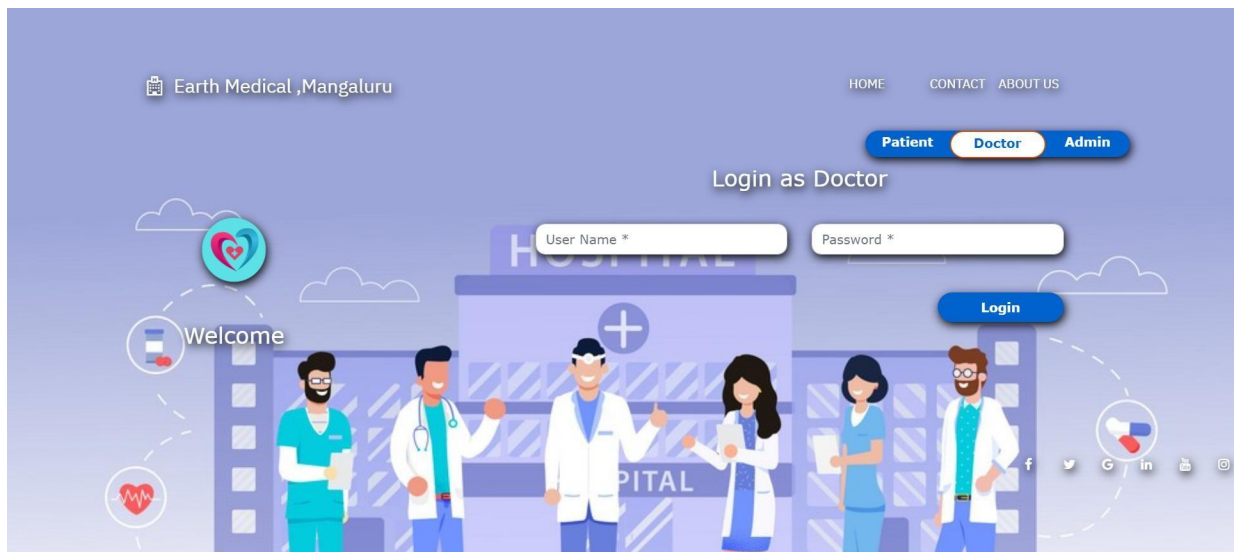
#### 5.1 Front Screen with Patient Registration

The screenshot displays the front screen of the Hospital Management System. At the top left, the text "Earth Medical, Mangaluru" is visible. To the right, there are navigation links: "HOME", "CONTACT", and "ABOUT US". Below these, there are three buttons: "Patient", "Doctor", and "Admin". The "Patient" button is highlighted. The main heading is "Register as Patient". Below this, there are several input fields: "First Name \*", "Last Name \*", "Your Email \*", "Contact \*", "Password \*", and "Confirm Password \*". There are also radio buttons for "Male" and "Female", with "Male" selected. A link "Already have an account? Login Now" is present. A "Register" button is at the bottom right. The background features a stylized illustration of a hospital building and medical professionals.

**Fig 5.1:** Front Screen with Patient Registration

The Fig 5.1, contains the front screen which includes Patient registration form and includes the button to navigate to doctor, admin and patient login page.

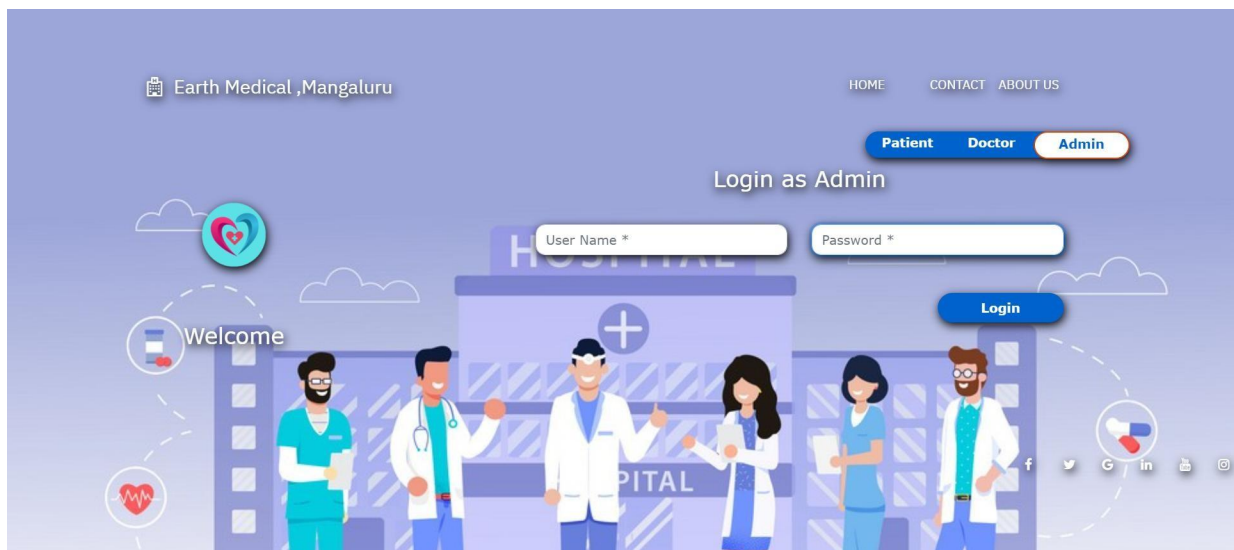
## 5.2 Doctor Login Screen



**Fig 5.2:** Doctor Login Screen

The Fig 5.2, Doctor's login page in which doctor can login through username and password assigned by admin.

## 5.3 Admin Login Screen



**Fig 5.3:** Admin Login Screen

The Fig 5.3, Admin can login to admin panel through this page using username and password.

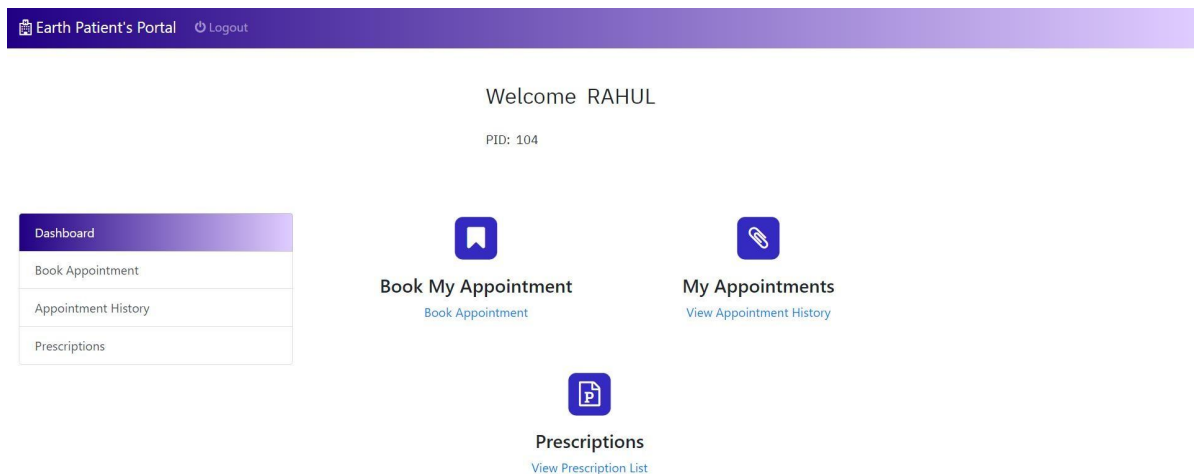
## 5.4 Doctor Login Screen



**Fig 5.4:** Patient Login Screen

The Fig 5.4, Shows the interface in which patient can login to admin panel through this page using email and password once they complete the registration in front screen.

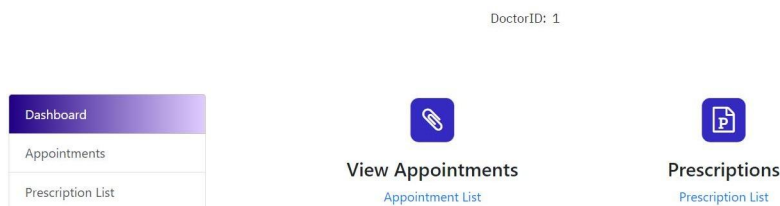
## 5.5 Doctor Login Screen



**Fig 5.5:** Patient's Portal Screen

The Fig 5.5, contains the options for patient to book appointment and to check their prescription prescribed by their doctor.

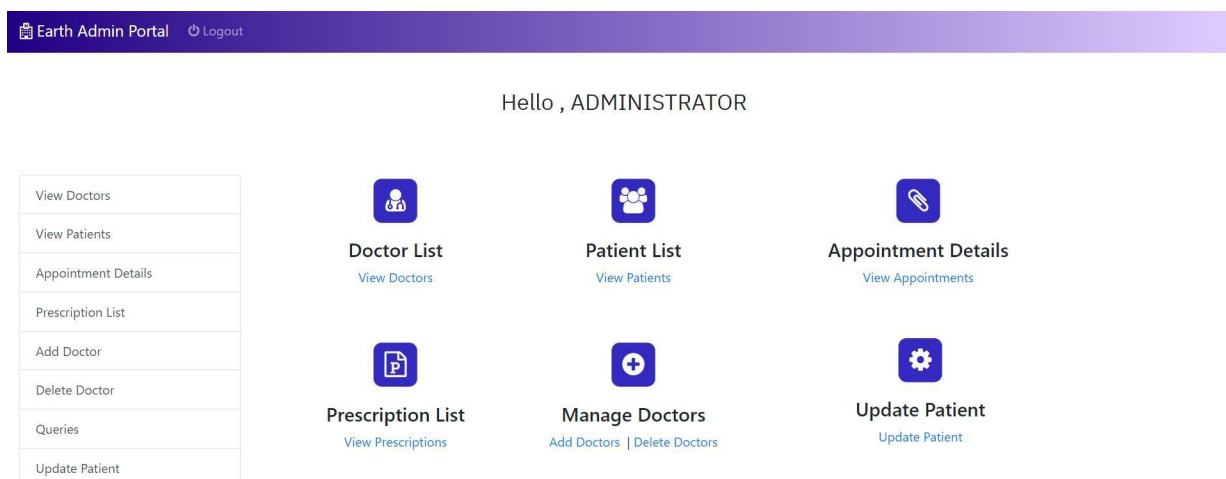
## 5.6 Doctor Login Screen



**Fig 5.6: Doctor's Portal Screen**

The Fig 5.6, contains the doctor panel in which doctors can check if there are any appointment and also can provide prescription for their patient.

## 5.7 Admin Login Screen



**Fig 5.7: Admin Portal Screen**

The Fig 5.7, contains the admin panel in which admin can check view doctor, patient, appointment, prescription information and can add/delete doctor. Admin also can update patient information and check the queries

## **CONCLUSION AND SCOPE FOR FUTURE WORK**

Hospital Management System provides a computerized version of Hospital management system which will benefit the patients well as the staff of the hospital. It makes entire process online where patients can make appointments, select doctors, date, time for their treatment; staffs can accept or reject appointments of the patients, add doctors, add patients, add services. It has another feature of showing transaction history. The future scope of this facility that many more features such as online transaction facilities, room allocation, pharmaceutical details, medicine prescription etc.

## REFERENCES

- [1] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
- [2] Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.
- [3] <https://stackoverflow.com>
- [4] <https://www.w3school.co>
- [5] <https://geeksforgeeks.org/>