

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590018



**DBMS MINI PROJECT
REPORT ON**

“HOSPITAL MANAGEMENT SYSTEM”

Submitted in partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING**

Submitted by

Sahil Bassan (1KG19CS078)

Under the Guidance of

Mrs.Amitha S

Assistant Professor
Department of CSE
K.S.S.E.M



KSSEM
K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

**Department of Computer Science & Engineering
K. S. SCHOOL OF ENGINEERING AND MANAGEMENT
#15, Mallasandra, off. Kanakapura Road, Bengaluru – 560109
2021-2022**

**K. S. SCHOOL OF ENGINEERING AND MANAGEMENT
BENGALURU - 560109**

Department of Computer Science & Engineering



CERTIFICATE

This is to certify that the **DBMS MINI PROJECT** entitled “**HOSPITAL MANAGEMENT SYSTEM**” presented by **Mr. Sahil Bassan**, USN: **1KG19CS078** of **V semester** in partial fulfillment of the award of **Bachelor of Engineering** in **Computer Science & Engineering** in **Visvesvaraya Technological University**, Belagavi during the academic year **2021-2022**. The **DBMS MINI PROJECT** has been approved as it satisfies the academic requirements in respect of **DBMS Mini Project(18CSL58)** prescribed for the Bachelor of Engineering degree.

Mrs. Nita Meshram
Assistant Professor, CSE
K.S.S.E.M., Bengaluru

Dr. Venkata Rao
Profesor & Head, CSE
K.S.S.E.M., Bengaluru

Dr. K.RAMANARASIMHA
Principal / Director
K.S.S.E.M., Bengaluru

Name of the Student: Sahil Bassan

USN: 1KG19CS078

Signature of the Student:

Name of the examiners

Signature with date

1. _____

2. _____

CONTENTS

Particulars	Page No
Acknowledgment	I
Abstract	II
 1. INTRODUCTION	
1.1 OVERVIEW	1
1.2 PROBLEM STATEMENT	1
1.3 DATABASE MANAGEMENT SYSTEM	2
1.4 SQL	2
1.5 HTML /CSS/ BOOTSTRAP/ JAVASCRIPT	3
1.6 PHP	4
 2. REQUIREMENTS SPECIFICATION	
2.1 OVERALL DESCRIPTION	5
2.2 SPECIFIC REQUIREMENTS	5
2.3 HARDWARE REQUIREMENTS	5
2.4 TECHNOLOGY	6
 3. DETAILED DESIGN	
3.1 SYSTEM DESIGN	7
3.2 ENTITY RELATIONSHIP DIAGRAM	8
3.3 RELATIONAL SCHEMA	11
3.4 DESCRIPTION OF TABLES	12

4. IMPLEMENTATION

4.1 IMPLENTATION	16
4.2 PROGRAMMING LANGUAGE SELECTION	16
4.3 RESULT	16

5. TESTING

5.1 SOFTWARE TESTING	17
5.2 MODULE TESTING AND INTEGRATION	17
5.3 LIMITATIONS	18

6. SNAPSHOTS

6.1 MAIN PAGE	19
6.2 ADMIN	20
6.3 DOCTOR	24
6.4 PATIENT	26

CONCLUSION 29

FUTURE ENHANCEMENTS 30

REFERENCES 31

Chapter 1

INTRODUCTION

1.1 OVERVIEW

“Hospital Management System” is designed to enable hospital and staff to function systematically and manage information effectively to offer better patient care. It is an effective software specifically designed to fulfil various requirements in managing hospitals.

It also enables Admin, Doctors, and Patients to view their profile, records of doctors, patients, and pharmacy and can check the appointment history and doctors can also add their patient's prescription.

1.2 PROBLEM STATEMENT

The main aim of “Hospital Management System” is to make an easy interface for Admin, Doctors, and Patients to get information of each other like :

1. **ADMIN** : Admin can check about all the doctors, patients, appointments of patients, pharmacy and he can add new doctors, pharmacy, he can also update or delete any patient, doctor, or pharmacy part.
2. **DOCTOR** : Doctors can view their profile, patient info., and his patient appointments and he can also add the prescription for their patients.
3. **PATIENT** : Patient can view their profile, doctor's record, their appointments and can check the prescriptions prescribed by their doctor.

1.3 DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. The DBMS essentially serves as an interface between the database and end users application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified, and the database schema, which defines the database's logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.4 SQL

SQL is a standard language for storing, manipulating and retrieving data in databases.

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.[13] Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

1.5 HTML / CSS / Bootstrap / JavaScript

HTML(Hyper Text Markup Language) is a markup language used for structuring and presenting content on the web and the fifth and current major version of the HTML standard.

HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications.

CSS (Cascading Style Sheets) is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. CSS describes how HTML elements should be displayed.

Bootstrap is a potent front-end framework used to create modern websites and web apps. It's open-source and free to use, yet features numerous HTML and CSS templates for UI interface elements such as buttons and forms. Bootstrap also supports JavaScript extensions.

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

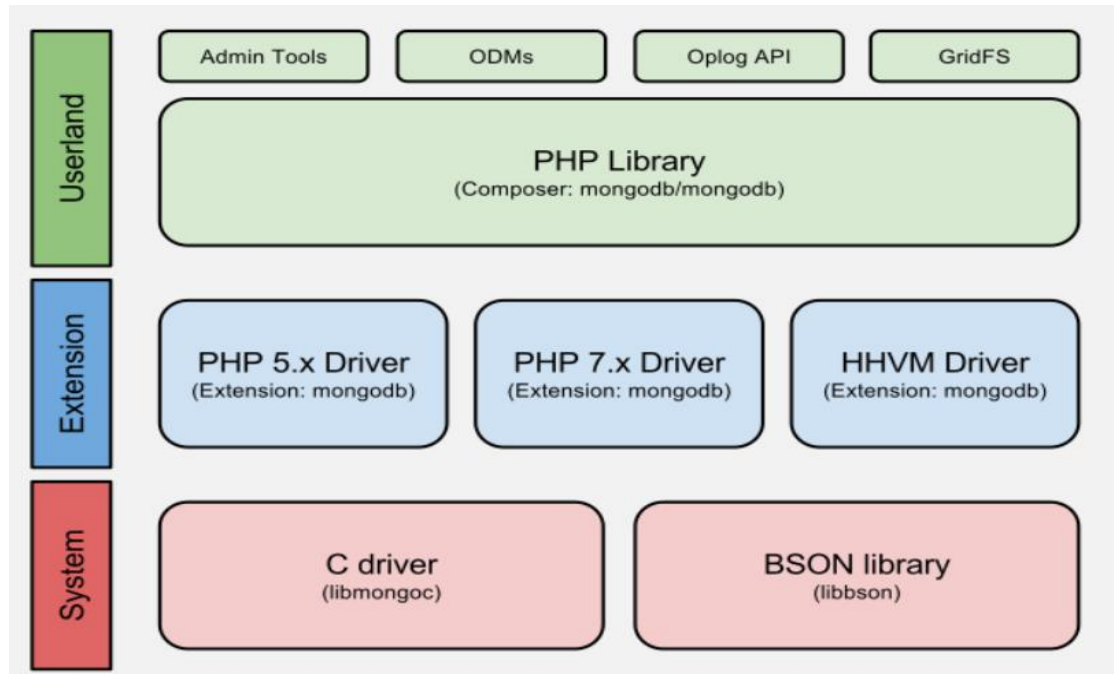
Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

1.5 PHP

PHP is a recursive acronym for "PHP: Hypertext Preprocessor". 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 pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time. PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time. PHP Syntax is C-Like.

To achieve connectivity we use PHP in this project. PHP `mysqli_connect()` function is used to connect with MySQL database. It returns resource if connection is established

or null. PHP mysqli_close() function is used to disconnect with MySQL database. It returns true if connection is closed or false.



Chapter 2

REQUIREMENTS SPECIFICATION

A computerized way of handling information about property and users details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

2.1 OVERALL DESCRIPTION

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

2.2 SPECIFIC REQUIREMENTS

The specific requirements of the Hospital Management System are stated as follows:

- Server deployment - XAMPP v3.2.4
- Web Browser – Firefox 84.0.2 or later, Google Chrome – 87.0 or later
- Database support – phpMyAdmin
- Operating system – Windows 10 / Ubuntu 20.04
- IDE – Atom / VS Code

2.3 HARDWARE REQUIREMENTS

- Processor – intel i3 or above
- RAM – 2 GB or more
- Hard disk – 3 GB or more
- Monitor – VGA of 1024x768 screen resolution
- Keyboard and Mouse

2.4 TECHNOLOGY

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

- CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document.
- SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation.
- Bootstrap is a potent front-end framework used to create modern websites and web apps. It's open-source and free to use, yet features numerous HTML and CSS templates for UI interface elements such as buttons and forms. Bootstrap also supports JavaScript extensions.
- JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.
- 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.
- XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

Chapter 3

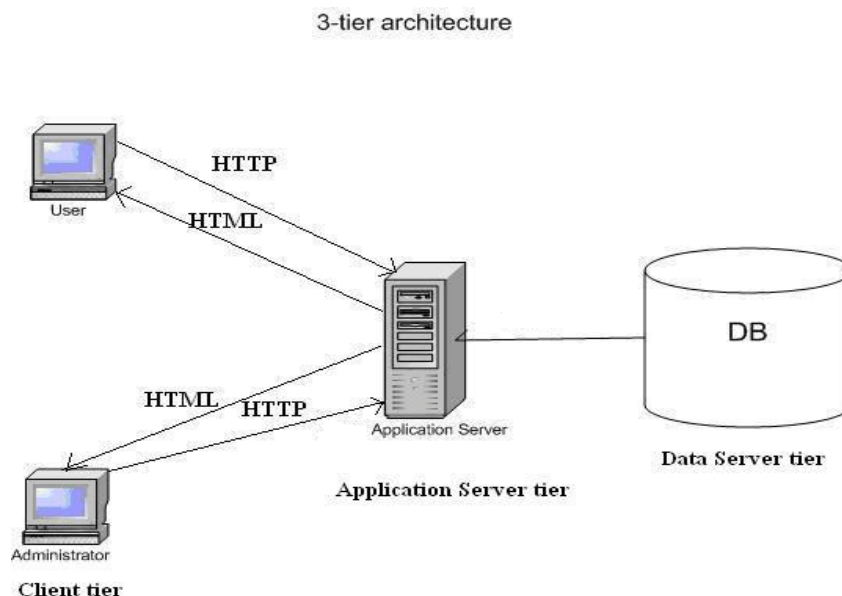
DETAILED DESIGN

3.1 SYSTEM DESIGN

The web server needs a XAMPP which 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. XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. This server will act as a mediator between the client browser and a database.

The following diagram shows the Three-tier architecture.

Fig. 3.1 Three-tier Architecture



Three-tier Client / Server database architecture is commonly used architecture for web applications. Intermediate layer called Application server or Web Server stores the web connectivity software and the business logic (constraints) part of application used to access the right amount of data from the database server. This layer acts like medium for sending partially processed data between the database server and the client.

Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users. Several types of databases, including relational or multimedia, may be created. Additionally, database architects may use one of several languages to create databases, such as structured query language.

3.2 ENTITY RELATIONSHIP DIAGRAM

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

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

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

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

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

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

DATABASE ER DIAGRAM (HOSPITAL MANAGEMENT SYSTEM)

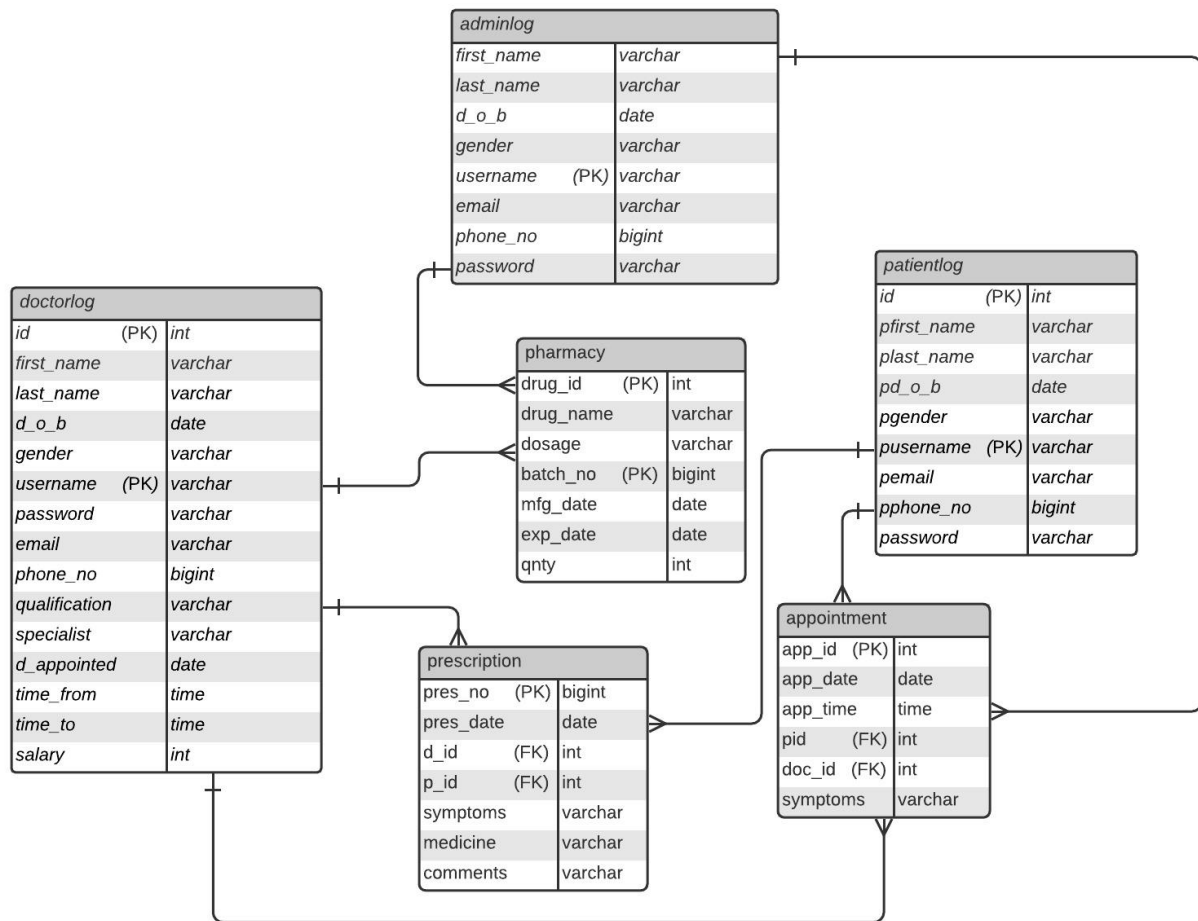


Fig. 3.2.1 Enhanced ER diagram of Hospital Management System

⌞ ← :- 1: n Identifying Relationship

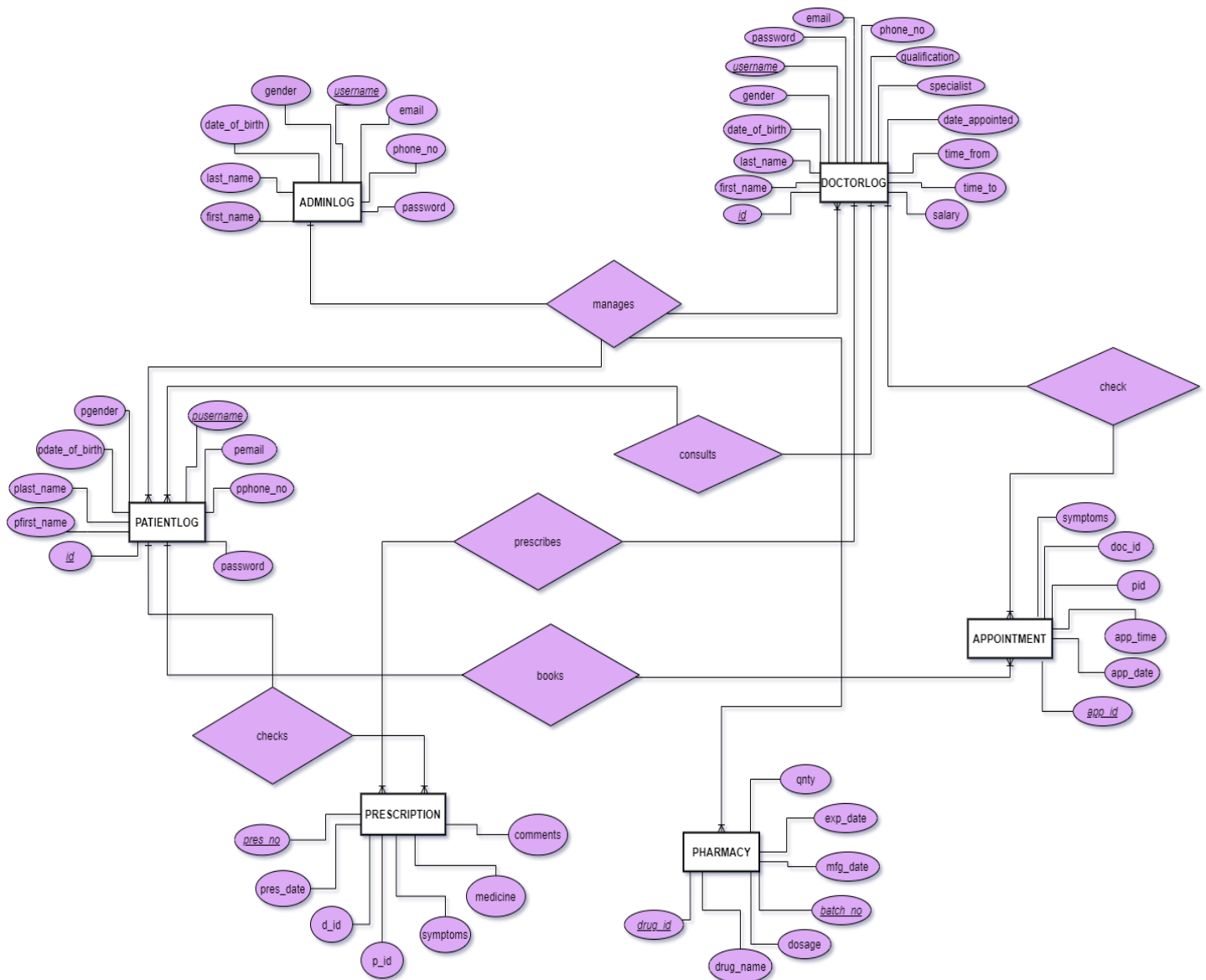


Fig. 3.2.2, ER diagram of Hospital Management System

⌞ ← :- 1: n Identifying Relationship

3.3 RELATIONAL SCHEMA

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

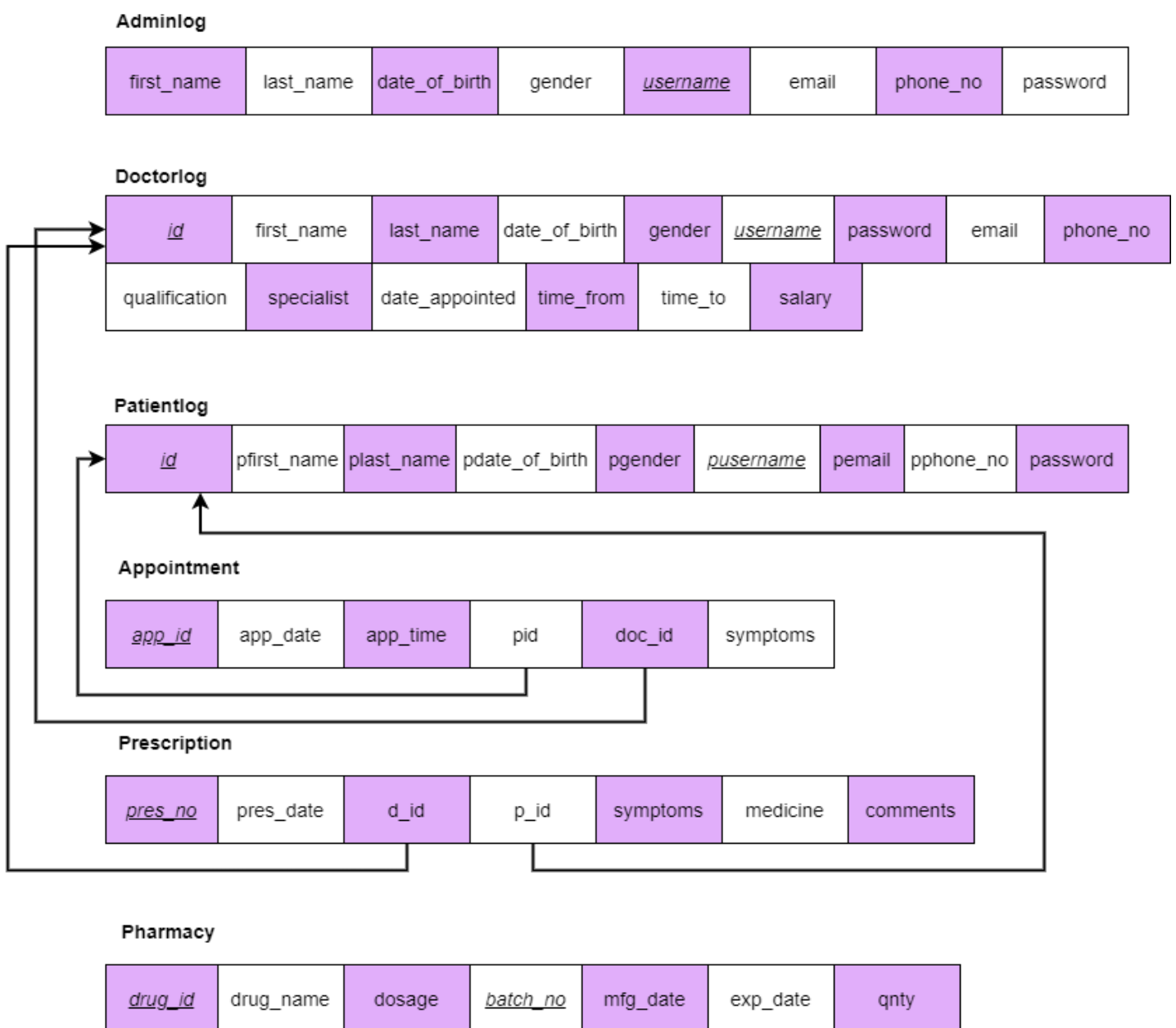




Fig. 3.3 Schema Diagram

3.4 DESCRIPTION OF TABLES

The database consists of six tables:

1. Adminlog: It stores the Admin details.
 - First_name: First Name of the Admin.
 - Last_Name: Last Name of the Admin.
 - Date_of_birth: Date of birth of the Admin.
 - Gender: Gender of the Admin.
 - Username: Unique Username for login
 - Email: Email id of the user.
 - Phone_No: Phone number of the user.
 - Password: Password associated with Admin to login into system.

desc adminlog;

#	Name	Type	Collation	Attributes	Null	Default
1	first_name	varchar(10)	utf8mb4_general_ci		No	None
2	last_name	varchar(10)	utf8mb4_general_ci		No	None
3	date_of_birth	date			No	None
4	gender	enum('Male', 'Female', 'Others')	utf8mb4_general_ci		No	None
5	username  	varchar(20)	utf8mb4_general_ci		No	None
6	email	varchar(30)	utf8mb4_general_ci		No	None
7	phone_no	bigint(20)			No	None
8	password	varchar(20)	utf8mb4_general_ci		No	None

2. Doctorlog: It stores the Doctor details.
 - Id: Unique Doctor id (primary key).
 - First_name: First Name of the Doctor.
 - Last_Name: Last Name of the Doctor.
 - Date_of_birth: Date of birth of the Doctor.
 - Gender: Gender of the Doctor.
 - Username: Unique Username for Doctor
 - Email: Email id of the Doctor.
 - Phone_No: Phone number of the Doctor.
 - Password: Password associated with Doctor to login into system.
 - Qualification: Qualification of the Doctor.
 - Specialist: Doctor specialization
 - Date_appointed: In Which Date Doctor is appointed.
 - Time_from: Doctor's availability from time.
 - Time_to: Doctor's availability to time.
 - Salary: Salary of the Doctor.

desc docorlog;

#	Name	Type	Collation	Attributes	Null	Default
1	id 🔑🔑	int(11)			No	None
2	first_name	varchar(20)	utf8mb4_general_ci		No	None
3	last_name	varchar(20)	utf8mb4_general_ci		No	None
4	date_of_birth	date			No	None
5	gender	varchar(10)	utf8mb4_general_ci		No	None
6	username 🔑	varchar(20)	utf8mb4_general_ci		No	None
7	password	varchar(20)	utf8mb4_general_ci		No	None
8	email	varchar(30)	utf8mb4_general_ci		No	None
9	phone_no	bigint(20)			No	None
10	qualification	varchar(20)	utf8mb4_general_ci		No	None
11	specialist	varchar(20)	utf8mb4_general_ci		No	None
12	date_appointed	date			No	None
13	time_from	time			No	None
14	time_to	time			No	None
15	salary	bigint(20)			No	None

3. Patientlog: It Stores the Patients details.




- Id: Unique Patient id (primary key).
- PFirst_name: First Name of the Patient.
- PLast_Name: Last Name of the Patient.
- PDate_of_birth: Date of birth of the Patient.
- PGender: Gender of the Patient.
- Pusername: Unique Username for Patient
- Pemail: Email id of the Patient.
- Pphone_No: Phone number of the Patient.
- Password: Password associated with Patient to login into system.

desc patientlog;

#	Name	Type	Collation	Attributes	Null	Default
1	id 🔑🔑	int(11)			No	None
2	pfirst_name	varchar(20)	utf8mb4_general_ci		No	None
3	plast_name	varchar(20)	utf8mb4_general_ci		No	None
4	pdate_of_birth	date			No	None
5	pgender	varchar(10)	utf8mb4_general_ci		No	None
6	pusername 🔑🔑	varchar(20)	utf8mb4_general_ci		No	None
7	pemail	varchar(30)	utf8mb4_general_ci		No	None
8	pphone_no	bigint(20)			No	None
9	password	varchar(20)	utf8mb4_general_ci		No	None



4. Appointment: It stores the appointment of the patient.
- App_id: Unique Appointment Id of the Appointment (primary key).
 - App_Date: Date of the Appointment.
 - App_Time: Booking time of the Appointment.
 - Pid: Foreign Key references to the Patient Id.
 - Doc_id: Foreign Key references to the Doctor Id.
 - Symptoms: Symptoms of the patient.

desc appointment;

#	Name	Type	Collation	Attributes	Null	Default
1	app_id 	int(11)			No	None
2	app_date	date			No	None
3	app_time	time			No	None
4	pid 	int(11)			Yes	NULL
5	doc_id 	int(11)			Yes	NULL
6	symptoms	varchar(30)	utf8mb4_general_ci		Yes	NULL

5. Pharmacy: It stores the Pharmacy details of the hospital.
- Drug_id: Unique Id of the medicine.
 - Drug_name: Name of the medicine.
 - Dosage: Amount of medication taken at one time.
 - Batch_No: Unique designation no. printed on drug label.
 - Mfg_Date: Manufacturing date of the medicine.
 - Exp_Date: Expiry Date of the medicine.
 - Qnty: Stock of the Medicine in the hospital.





desc pharmacy;

#	Name	Type	Collation	Attributes	Null	Default
1	drug_id 	int(11)			No	None
2	drug_name	varchar(20)	utf8mb4_general_ci		Yes	NULL
3	dosage	varchar(20)	utf8mb4_general_ci		Yes	NULL
4	batch_no 	bigint(20)			Yes	NULL
5	mfg_date	date			Yes	NULL
6	exp_date	date			Yes	NULL
7	qnty	int(11)			Yes	NULL

6. Prescription: It stores the prescription details of patient prescribed by the Doctor.
- Pres_No: Unique Prescription number of the patient.
 - Pres_Date: Prescribed Date of the Prescription.

- P_id: Foreign Key references to the Patient Id.
- D_id: Foreign Key references to the Doctor Id.
- Symptoms: Symptoms of the patient.
- Medicine: Medicine prescribed by the doctor.
- Comments: Advice from the doctor to patient.

desc prescription;

#	Name	Type	Collation	Attributes	Null	Default
1	pres_no  	bigint(20)			No	None
2	pres_date	date			Yes	NULL
3	d_id 	int(11)			Yes	NULL
4	p_id 	int(11)			Yes	NULL
5	symptom	varchar(30)	utf8mb4_general_ci		Yes	NULL
6	medicine	varchar(30)	utf8mb4_general_ci		Yes	NULL
7	comments	varchar(50)	utf8mb4_general_ci		Yes	NULL

Chapter 4

IMPLEMENTATION

4.1 Implementation

The Project is implemented in HTML, CSS, JavaScript And Bootstrap for Front-End(website) and MySQL for Back-End (database). It has three-tier architecture with Front-End forming Application layer and Back-End forming middle layer and Database Input Validation is done in PHP.PHP communicate with localhost through XAMPP server. Then XAMPP server communicate with HTML and PHP code that is then displayed in the front-end. The Server communicate with MySQL using phpMyAdmin.

4.2 Programming Language Selection

PHP was used to communicate with MySQL because PHP is the most popular scripting language for web development. It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. MySQL is a Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). It is also free and open source. The combination of PHP and MySQL gives unmet options to create just about any kind of website - from small contact form to large corporate portal.HTML and CSS is used for the front-end as it versatile and easy to use.

4.3 Result

The resulting system is able to:

- Authenticate user credentials during login.
- Register the information of new Doctor appointed to Hospital.
- Register the information of new Patient to Hospital.
- Register the Pharmacy of the hospital.
- Allow Admin to check the record of doctors, patients and their appointment history.
- Allow patients to book the appointments.
- Allow Doctors to prescribe the prescription for patients.
- Allow patients to check their prescriptions and appointment history.
- Allow patients to contact to their respective doctors on having any queries.

Chapter 5

TESTING

5.1 SOFTWARE TESTING

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

Test Case:

- Input fields should be checked for the max field value. Input values greater than the specified max limit should not be accepted or stored in the database.
- Application crash or unavailable pages should be redirected to the error page.

5.2 MODULE TESTING AND INTEGRATION

Module testing is a process of testing the individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommend testing the smaller building blocks of the program. It is largely white box oriented. The objective of doing Module testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module. Module testing allows implementing of parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

The final integrated system too has been tested for various test cases such as duplicate entries and type mismatch.

Test Case:

- Check the response time for any action under a light, normal, moderate and heavy load condition.
- Check the functionality of buttons available on all pages.

5.3 LIMITATIONS

- Patient can book the appointment but can't confirm whether the doctor will available or not.
- User's session timing is not recorded.
- Ambulance facility not available which can be implemented in future.
- Diagnosis lab Room's Record are not available which can also implemented in future version of this project.

Chapter 6

SNAPSHOTS

This chapter consists of working screenshots of the project.

6.1 Main Page



Fig. 6.1.a

Please Login from Here

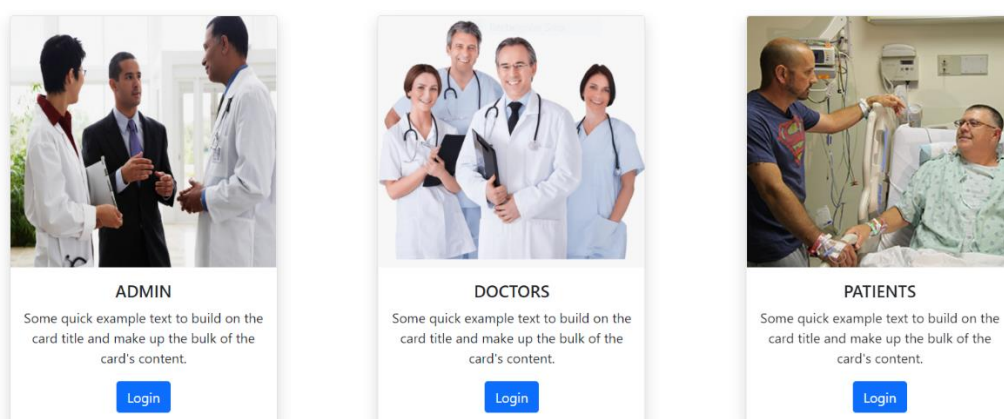


Fig.6.1.b

This is the Main page of Hospital Management System shown in fig. 6.1.a and 6.1.b to any User.

6.2 Admin

6.2.1 Login Page for Admin

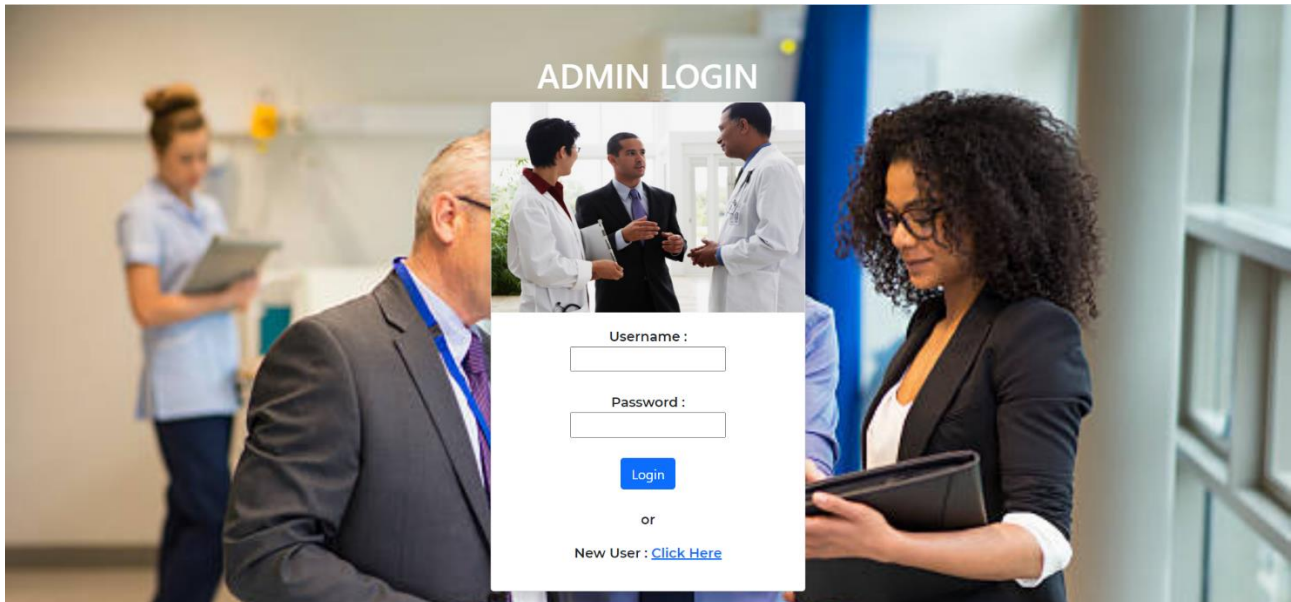


Fig.6.2.1

Fig.6.2.1 contains the login page of admin show that they can enter the dashboard and modify the doctor And patient details.

6.2.2. Sign Up Page for Admin

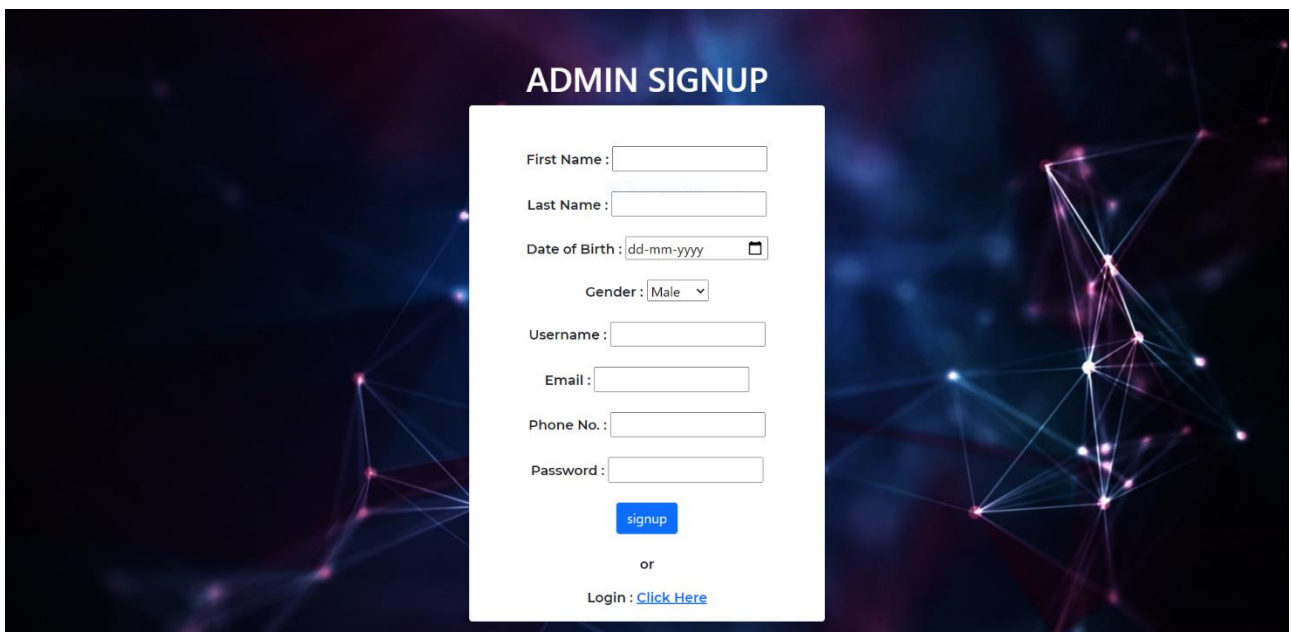


Fig. 6.2.2

Fig. 6.2.2 shows the Sign Up Page for New Admin.

6.2.3. Admin Profile

Welcome to Admin kavita07

Log Out

Admin Profile

First Name : Kavita

Last Name : Chaudhary

Date of Birth : 2000-03-07

Gender : Female

Username : kavita07

Email : 831kavita73@gmail.com

Phone No. : 8318959667

Fig.6.2.3.

Fig.6.2.3 contains the dashboard of the admin database where admin can check his/her profile, records of doctor, patient ,appointment, and pharmacy and can update or add or delete the doctors record.

6.2.4. Doctors Records

Doctor's Records of LifeCare

ID	First_Name	Last_Name	Date_Of_Birth	Gender	Username	Email	Phone_No.	Qualifica
2	Avinash	Prasad	1998-09-26	Male	avinash26	avinashprasad184@gmail.com	7004806734	MBBS
3	Kavita	Chaudhary	2001-03-07	Male	kavita07	831kavita73@gmail.com	8318959667	MD-PhD

Fig.6.2.4

Fig.6.2.4 shows the records of the doctor in the hospital.

6.2.5. Add Doctors



Fig.6.2.5 Add doctor

Fig.6.2.5 Contains the form to the add doctors to the hospital.

6.2.6. Patient's Record



ID	First_Name	Last_Name	Date_Of_Birth	Gender	Username	Email	Phone_No.	Action
1	Dhruv	Shukla	2020-12-23	male	dhruv	dhruv@ksit.edu.in	9956620173	
2	pawan	patil	2000-05-15	male	pawan	pawan@yahoo.com	9988552211	
3	bhuvana	chandrika	2000-12-26	female	bhuvana	bhuvana@yahoo.com	7766443311	

Fig. 6.2.6 Patient's record

Fig.6.2.6. contains the patient records of the hospital.

6.2.7. Appointment's Record



The image shows a software interface titled "Appointment's Records of LifeCare" on a blue background. Below the title is a table with appointment records. To the right of the table, there is a small graphic of a potted plant, a calendar, and some keyboard keys.

Appointment_ID	Appointment_Date	Booking_Time	Patient_ID	P_First_Name	P_Last_Name	Date_Of_Birth	Gender
1	2021-01-10	21:43:00	2	pawan	patil	2000-05-15	male
5	2021-01-14	11:24:00	1	Dhruv	Shukla	2020-12-23	male
2	2021-01-10	21:45:00	1	Dhruv	Shukla	2020-12-23	male

Fig.6.2.7 Appointment's record

Fig.6.2.7 contains the appointment record of the patient recorded in the hospital.

6.2.8. Pharmacy's Record



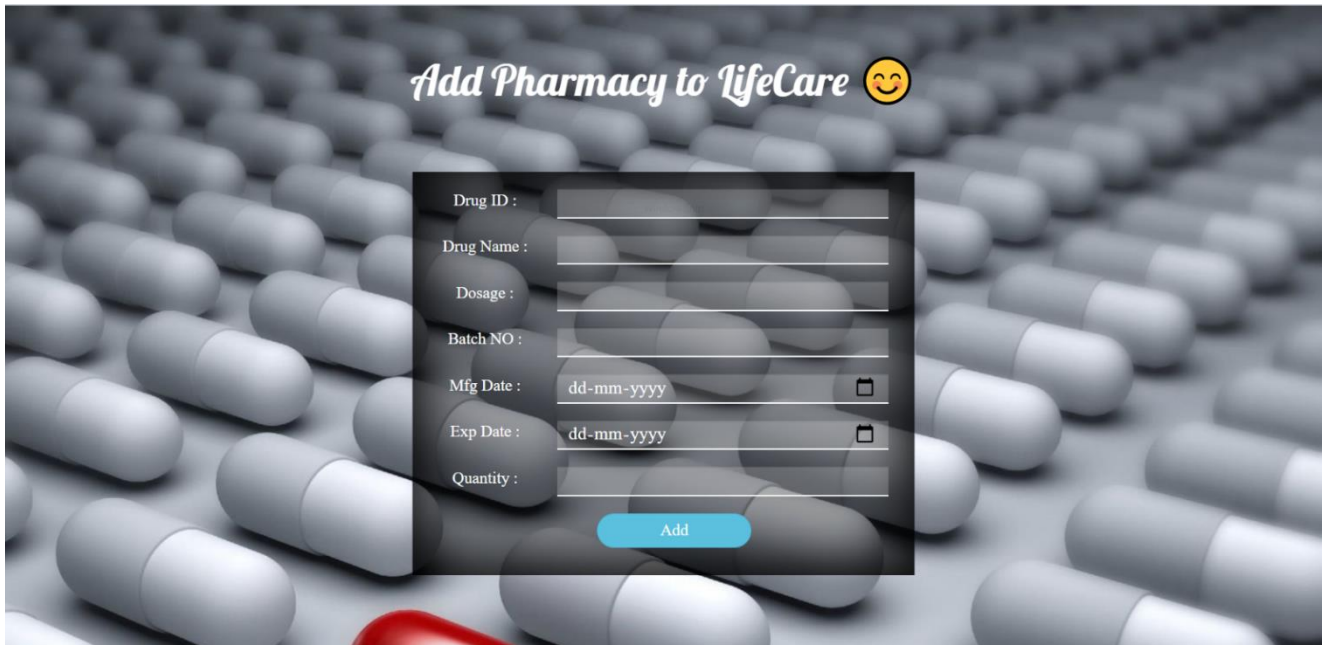
The image shows a software interface titled "Pharmacy Records of LifeCare" on a background of various colored capsules. Below the title is a table with pharmacy records. Each record in the table has edit and delete icons in the Action column.

Drug_ID	Drug_Name	Dosage	Batch_No	Mfg_Date	Exp_Date	Quantity	Action
120	paracetamol	100gm	789456	2020-12-15	2024-06-19	100	 
125	eno	10gm	123654	2020-12-29	2022-09-29	50	 
456	Combilame	500mg	78897	2016-06-28	2020-12-12	200	 

Fig. 6.2.8 pharmacy record

Fig. 6.2.8 contains the record of the pharmacy of the hospital which can be updated or deleted.

6.2.9. Add Pharmacy




Add Pharmacy to LifeCare 😊


Drug ID :

Drug Name :

Dosage :

Batch NO :

Mfg Date : 

Exp Date : 

Quantity :

Fig. 6.2.9 Add pharmacy

Fig. 6.2.9 contains the form to add the pharmacy to the hospital.

6.3. Doctor

6.3.1. Login Page



DOCTOR LOGIN



Username :

Password :

Fig. 6.3.1 Login page

Fig.6.3.1 shows the login page of the doctors.

6.3.2. Doctor's Profile

Welcome to Doctor kavita07

Log Out

Profile

Doctors Records

Patient

Appointment

Pharmacy

Add Prescription

Doctor Profile

First Name : Kavita

Last Name : Chaudhary

Date of Birth : 2000-03-07

Gender : Female

Username : kavita07

Email : 831kavita73@gmail.com

Phone No. : 8318959667

Fig. 6.3.2 Doctor's Profile

Fig.6.3.2 contains the doctor dashboard in which he/she can see his/her patients record and appointment of that patient and he /she can prescribed the prescription.

6.3.3 Add Prescription

Prescription For Patient

Prescription No :

Prescription Date : dd-mm-yyyy

Doctor ID :

Patient ID :

Symptom :

Medicine :

Comments :

Add

Fig.6.3.3 Add prescription

Fig.6.3.3 shows the form in which doctor can add the prescription for the patient.

6.4. Patient

6.4.1. Login Page

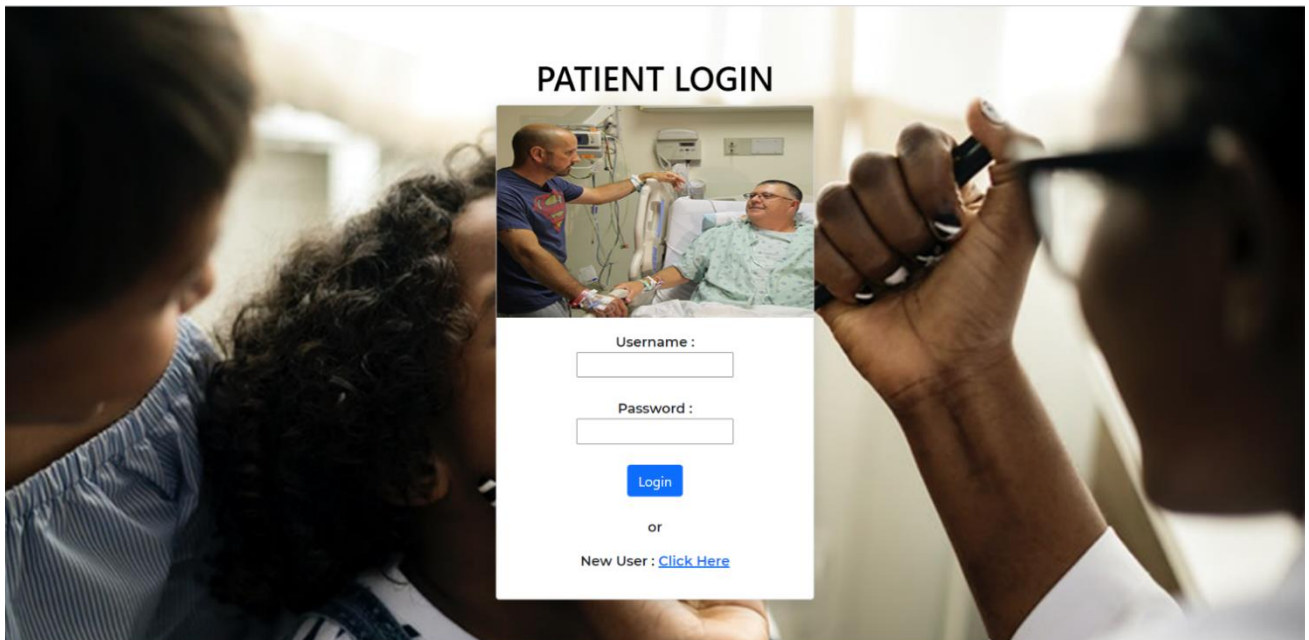


Fig. 6.4.1 Login Page

Fig.6.4.1 shows the login page of the patient.

6.4.2. Sign Up Page

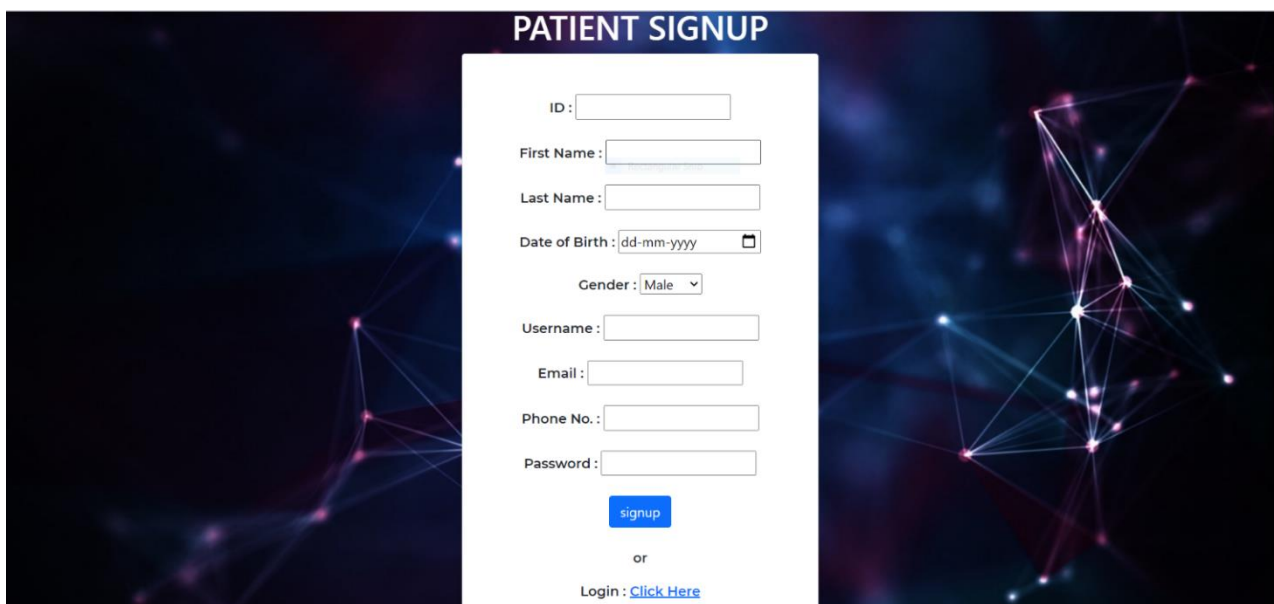
The image shows a 'PATIENT SIGNUP' form overlaid on a background with a dark blue and purple geometric pattern. The form has a title 'PATIENT SIGNUP' at the top. Below the title are several input fields: 'ID : ', 'First Name : ', 'Last Name : ', 'Date of Birth : ' (with a calendar icon), 'Gender : ' (with a dropdown menu showing 'Male'), 'Username : ', 'Email : ', 'Phone No. : ', and 'Password : '. Below these fields is a blue 'signup' button. Under the button is the word 'or'. At the bottom of the form is a link that says 'Login : [Click Here](#)'.

Fig.6.4.2 Sign Up Page

Fig.6.4.2 Contains the form to the add patient to the hospital.

6.4.3. Patient's Profile

The screenshot shows a web application interface for a patient's profile. At the top, a white header bar contains a hamburger menu icon, the text "Welcome dhruv to your LifeCare's Profile", a search icon, and a "Log Out" button. A blue sidebar on the left lists navigation options: "Profile", "Doctors Records", "Appointment", "Add Appointment", and "Prescription". The main content area features a "Patient Profile" form with the following fields: ID (1), First Name (Dhruv), Last Name (Shukla), Date of Birth (2020-12-23), Gender (male), Username (dhruv), Email (dhruv@ksit.edu.in), and Phone No. (9956620173). The background of the main area is a blurred image of a hand on a laptop.

Fig.6.4.3.Patient's Profile

Fig.6.4.3. contains the patient dashboard in which he/she can see their record, appointment and prescriptions.

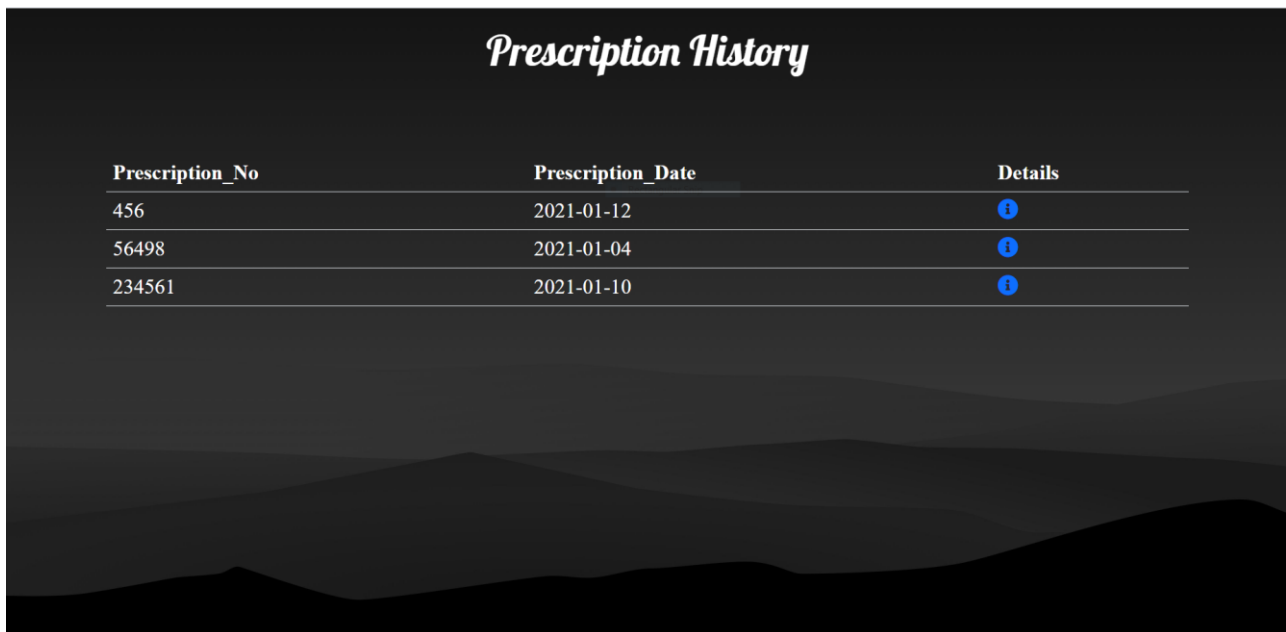
6.4.4. Add Appointment

The screenshot displays the "Add Appointment" form on a dark blue background with a starry constellation graphic on the right. The form is titled "Add Appointment" in a stylized font. It includes the following fields: Appointment ID, Appointment Date (with a calendar icon), Booking Time (with a clock icon), Patient ID, Doctor ID, and Symptoms. A blue "Add" button is positioned below the form. A note at the bottom states: "NOTE : If you wanna check Doctor's Record then [Click Here](#)".

Fig.6.4.4. Add appointment

Fig.6.4.4. shows the form for adding an appointment.

6.4.5. Prescription's History






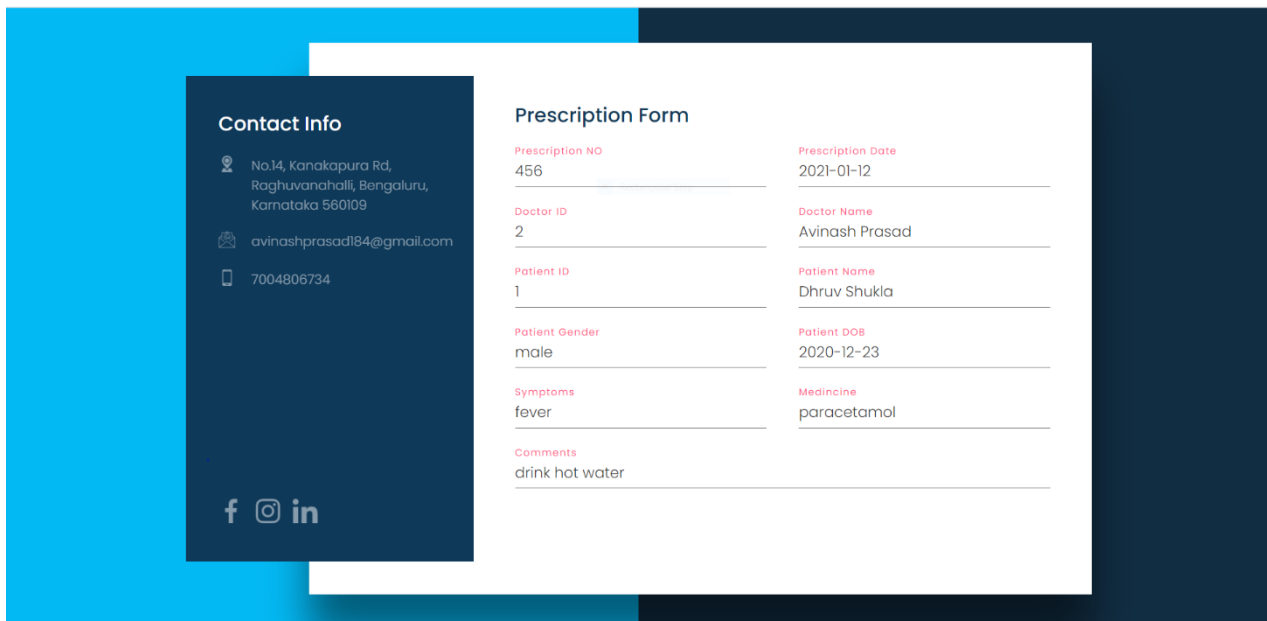
Prescription_No	Prescription_Date	Details
456	2021-01-12	
56498	2021-01-04	
234561	2021-01-10	


Fig.6.4.5.Prescription's history


Fig.6.4.5. contains the prescription history of the patient.


6.4.6. Prescription's Detail






Contact Info

 No.14, Kanakapura Rd,
Raghuvanahalli, Bengaluru,
Karnataka 560109

 avinashprasad184@gmail.com

 7004806734

Prescription Form

Prescription NO	Prescription Date
456	2021-01-12
Doctor ID	Doctor Name
2	Avinash Prasad
Patient ID	Patient Name
1	Dhruv Shukla
Patient Gender	Patient DOB
male	2020-12-23
Symptoms	Medicine
fever	paracetamol
Comments	
drink hot water	

Fig.6.4.6. Prescription's Detail

Fig.6.4.6.shows the details of particular prescription clicked by patient.

CONCLUSION

Hospital Management System is designed to enable hospital and staff to function systematically and manage information effectively to offer better patient care. It is an effective software specifically designed to fulfil various requirements in managing hospitals. It also enables Admin, Doctors, and Patients to view their profile, records of doctors, patients, and pharmacy and can check the appointment history and doctors can also add their patient's prescription. This is developed using HTML5, CSS, JavaScript, PHP, Bootstrap and XAMPP. The goals achieved by this project are:

- Centralized database.
- Easier way to update Doctor's data if any changes are to be made.
- Easier and efficient way to book an appointment and check prescription .
- User friendly environment.
- Efficient management of Storing the records of doctor and patient.
- Ability to view records of doctor, appointment and pharmacy.

FUTURE ENHANCEMENTS

Future upgrades to this project will implement:

- Better interfaces for the ability to book an appointment and add doctor.
- More facilities will be implemented for ambulance and diagnosis lab.
- Biometric attendance system should be added for the doctor.
- Aadhar number of the doctor and patient will be linked to the database.
- Notification system will be implemented so that patient will get the notification after 15 days for next visit.
- User interface will be updated time to time as per the requirements.

REFERENCES

- [1] Ramakrishnan, R., & Gehrke, J. (2011). Database management systems. Boston: McGraw-Hill.
- [2] PHP & MySQL Novice to Ninja – by Kevin Yank
- [3] Database systems concepts. Estados Unidos: McGraw-Hill Companies, Inc.
- [4] David F. (2011). JavaScript: The Definitive Guide Sixth edition.
- [5] <https://www.w3schools.com>
- [6] <https://www.canvasjs.com>
- [7] <https://bootstrap.com>
- [8] <https://fontawesome.com>
- [9] <https://stackoverflow.com>
- [10] <https://www.youtube.com/channel/UCwfaAHy4zQUb2APNOGXUCCA>
- [11] https://www.youtube.com/channel/UCJZZ7_zG_uu0K0nlnR5yz5Q
- [12] <https://www.youtube.com/channel/UCtVM2RthR4aC6o7dzySmExA>
- [13] <https://www.youtube.com/channel/UCKXDrRLolLRPTWudeC0wuPw>