

# **Design and Implementation of a Web Based Prescription System**

## **A Project Report**

Submitted to the Department of Computer Science and Engineering,  
Jahangirnagar University in partial fulfillment of the requirements for the  
degree of Master of Science in Computer Science under PMSCS program.

### **Submitted By:**

Md. Mineuddin Ahmed Dipu

Roll: CSE 202103099

Batch: 27<sup>th</sup>

### **Supervised By:**

Professor Dr. Md. Ezharul Islam



**Department of Computer Science and Engineering**

**Jahangirnagar University, Savar, Dhaka-1342**

**Bangladesh**

**November 2022**

## Approval of Acceptance

This project report is written by Md. Mineuddin Ahmed Dipu (ID: CSE202103099) entitled “Design and Implementation of a web based prescription system” is submitted to the PMSCS Program, Department of Computer Science and Engineering, Jahangirnagar University in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The project is done under the supervision of Professor Dr. Md. Ezharul Islam, Department of Computer Science and Engineering, Jahangirnagar University.

We have examined this report and recommend its acceptance.

.....

Dr. Md. Ezharul Islam  
Supervisor and Professor  
Department of Computer Science and Engineering  
Jahangirnagar University  
Savar, Dhaka-1342, Bangladesh

.....

[Sarnali Basak](#)  
Second Examiner Associate Professor  
Department of Computer Science and Engineering  
Jahangirnagar University  
Savar, Dhaka-1342, Bangladesh

Accepted by:

.....

Professor Dr. Md. Humayun Kabir  
Coordinator  
PMSCS Program 2022-2023  
Department of Computer Science and Engineering  
Jahangirnagar University  
Savar, Dhaka-1342, Bangladesh

# Declaration:

There, I declare that the task presented in this report is the outcome of the investigation performed by me under the supervision of Professor Dr. Md. Ezharul Islam, Department of Computer Science and Engineering, Jahangirnagar University. The work was spread over one final semester course: Research Project, in accordance with the course curriculum of the department for the Professional Masters of Science in Computer Science.

.....

Md. Mineuddin Ahmed Dipu

Roll Number: CSE202103099

Batch : 27<sup>th</sup>

# Acknowledgment:

I would like to give thanks especially to the following people who have helped me a lot for completing this project possible:

My parents, for their regular support and continuous funding.

**Dr. Md. Ezharul Islam (My supervisor)** Department of Computer Science and Engineering, Jahangirnagar University, for giving me the opportunity to undertake this project with him and also for his support and guidance.

I also would like to show my sincere gratitude to Professor. **Dr. Md. Humayun Kabir**, Coordinator, PMSCS Program 2020-2021, Department of Computer Science and Engineering, for his kind help in completing my project and honorable faculty members and my department of staffs.

Md. Rahul (My friend), for his tremendous knowledge on anything programming or logic related and his willingness to always help.

# **Abstract.**

The key objective of this project is to provide a high tech online prescription with full of medicine list, investigation list and patient/doctor satisfaction. To develop this project I use Restful API for breezing with backend and frontend. For backend I use PHP 7.4 and Laravel framework and for frontend I use React 18.0 version. We will release the newer version day by day and always try to make it smooth, easy and fast loader. In the current version we try to provide very user friendly, easy to understand and a clean and smooth interface for the admin user, doctors and patient. Their admin can easily configure doctor, patient, medicine and other related settings for creating a prescription. The prescription will automatically send through email to the patients and also a patient can download the prescription after log into their own dashboard.

This document is the final report for the project online prescription system for doctors. Besides we will create more opportunities for new doctors so they can connect, share and learn more from our prescription.

# Table of Contents

<b>Chapter 1 Introduction</b>	<b>Page No</b>
1.1 Motivation	8
1.2 Objectives	8
1.3 Scope of Project	8
1.4 Problem Definition	9
1.5 Research Methodology	9
1.6 Outline	9
<b>Chapter 2 Background and Related Works</b>	
2.1 Introduction	10
2.2 Background and History	10
2.3 Related Work and Gap Analyze	10
2.4 Gap Comparison Table	12
<b>Chapter 3 System Analysis</b>	
3.1 Introduction	13
3.2 Software Requirements Specification	13
3.3 User Characteristics	13
3.4 Functional Requirement	14-15
3.5 Non-Functional Requirement	16
<b>Chapter 4 System Design</b>	
4.1 PHP Analysis and Design	17
4.2 Use Case Diagram	18
4.3 ERD and Database Design	19
4.4 Table Structure	20-23
<b>Chapter 5 Implementation and Results</b>	
5.1 Technology Used	24
5.2 Hardware Requirement	24
5.3 GUI Design and Test Case	24-28
5.2 Testing	27-31
5.3 Coding	33-36
<b>Chapter 6 Conclusion and Future Works</b>	
5.1 Conclusion	37
5.2 Future Works	37
<b>References</b>	38-39

# List of Figures

	<b>Page No.</b>
Figure 2.1: Evanto Market's Prescription	11
Figure 2.2: Indian Hospitals Prescription	11
Figure 2.3: Global Hospitals Web Prescription including app	12
Figure 3.1: - Logical Structure of the Prescription Data	16
Figure 4.1: Showing How to connect React JS with Laravel API	17
Figure 4.2: How Laravel API and React Request Response	17
Figure 4.3 Use Case diagram	18
Figure 4.4: ER Diagram	19
Figure 4.5: Prescription Database Table	20
Figure 4.6: Prescription User Table	20
Figure 4.7: Prescription Patient Table	21
Figure 4.8: Prescription Doctor Table	22
Figure 4.9: Prescription Medicine Table	22
Figure 4.10: Prescription Table	23
Figure 5.1: Dashboard Design	25
Figure5.2: Sidebar menu Page Design	25
Figure5.3: Doctor add Page Design	26
Figure5.4: Patient add Page Design	27
Figure5.5: Medicine Add Page Design	27
Figure5.6: Investigation insert Page Design	28
Figure5.7 Prescription Page Design	28

# Chapter 1

## Introduction

### 1.1 Motivation :

The core motivation behind this system is to reduce doctor's effort and patient hassle to understand the writing of doctors and the name of the medicines. Often patients are complaining that they don't understand the writing of doctor's. On the other hand a doctor has write a lots of prescription daily. They also has to write the same things repeatedly. Which is really annoying. We also face the same issue that sometimes we don't understand the writing of doctors. So, to help both doctors and patients we want to introduce this solution. So it will helpful for both doctors and patients.

### 1.2 Objectives

The main objective is to obtain doctors and patient satisfaction on health related sector. To reduce doctor's effort and patient hassle to understand the writing of doctors and the name of the medicines. Often patients are complaining that they don't understand the writing of doctor's. On the other hand a doctor has write a lots of prescription daily. They also has to write the same things repeatedly. Which is really annoying. We also face the same issue that sometimes we don't understand the writing of doctors. So, to help both doctors and patients we want to introduce this solution. So it will helpful for both doctors and patients. By this system all new comers doctor will be benefited. They will get a huge data set for research purpose. They can study the previous patient's history and what the prescription has provided to the patients for their betterment.

### 1.3 Project Scope:

By this project doctors and patient both can be benefited. Doctors can easily prescribe patient. They can keep record of their patient and patient also get email of the prescription. So they also can track their health history. They don't need to preserve any piece of paper at their home. Moreover a piece of paper may destroy at any time. On the other hand if the patient will come after a long time then the doctor can easily track this patient previous record. So it will easy to find the previous health history of any patient. A doctor also can research on the data he has. He also can share his experience with other practitioners and medical student for their study purpose. Overall it can play an incredible role in our health care sector.



## **1.4 Problem Definition**

It is a very difficult task to collect the information of all the patient variation and their diseases and create a database. Moreover a prescription system is the only way to collect information from anywhere in the world through the prescription system.

## **1.5 Research Methodology**

I took an initiative to development a web based prescription system. Through which a solution to for both doctors and patients. In creation this system, I will use HTML5, CSS 3, React Bootstrap4 , REACT, PHP 7.4 framework Laravel (MVC) ,MySQL

## **1.6 Outline**

This Prescription System is developed in such a way where a Doctor can be able to write prescription for their patient easily. Simple and easy admin panel to manage all Process. This system can store huge data as an archive. Can build report if need in future. Most popular and common language are used to develop this web system. Every user can see basic information but restricted to access permission.

# Chapter 2

## Background and Related Work

### 2.1 Introduction :

The interface helps the doctor to login through the system and add the details regarding the prescription. There are two type of user's. Admin user and Doctor Users.

The Admin users will login the system and make the basic configuration for the doctors to write prescription and see the previous prescription and analysis the input.

### 2.2 Background and History:

An online prescription system is software for doctors to create online prescription for patient for better experience of the patients so that understand well and keep track about health record. From the very beginning of the prescription practice doctors do it manually and give the prescription paper to the patient. Patient use it for doing prescribed investigation and taking medicine from shops. They need to keep it for further visit to the doctors so that doctor easily remember about the previous case history of that particular patient. There are some problems like patient may stole the paper or that paper may destroyed by water, fire or environmental case. So it is not possible for them to recollect the paper easily.

For this it is important to keep the prescription in online. It will helpful for both doctor and patient.

### 2.3 Related Work and Gap Analyze

There is already some same kinds of systems in the local and global market already. But still I did not get any open source platform where doctor can freely create account and doing prescription. Moreover everybody have some limitations in their system. So we want to make our systems open source so any one from anywhere in the world can use it. Moreover we also want to solve that limitations so that it becomes more user friendly and easy to use. Below I mention some others work. I am also going to show the difference between our system and others system. The comparison is listing below of the figure.

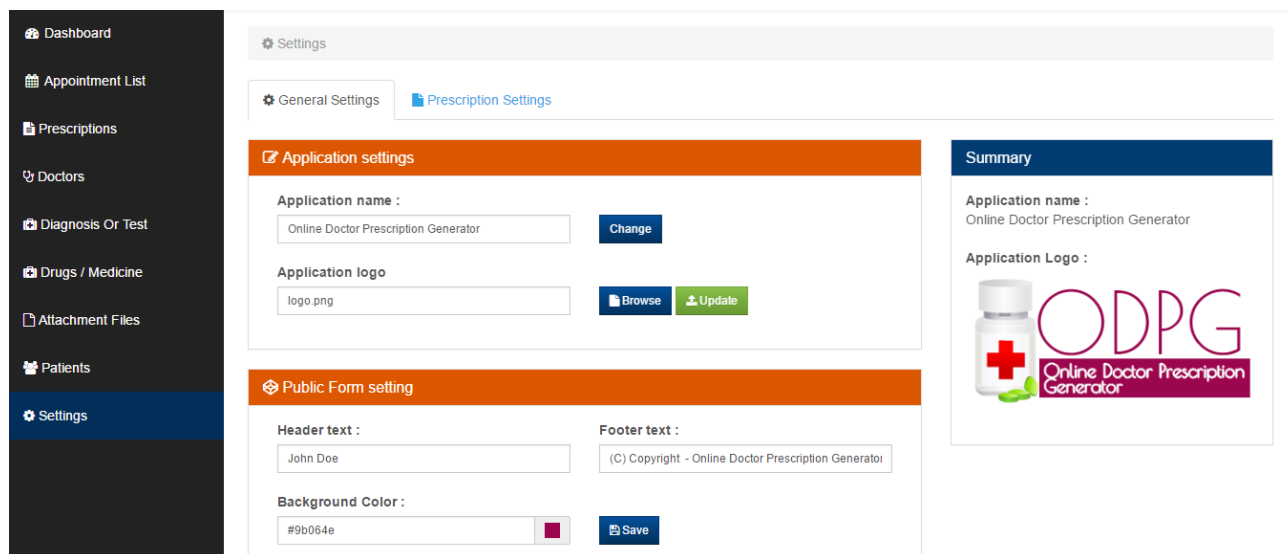


Figure 2.1 : Evanto Market's Prescription Interface

**Description:** Displaying the interface of a prescription system in evanto market place.

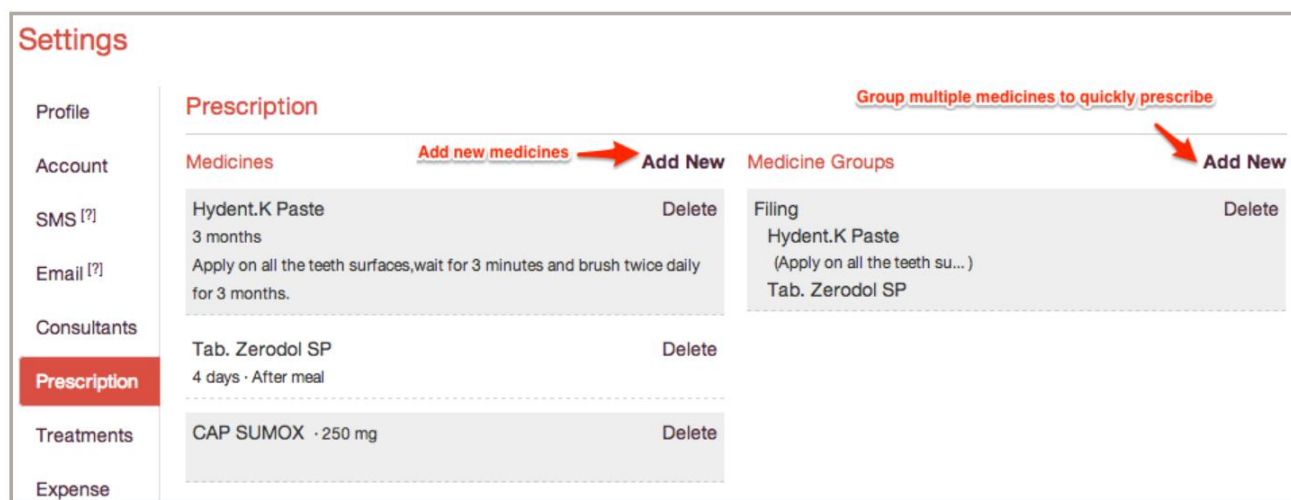


Figure 2.2 : Indian Hospitals Prescription Interface

**Description:** Displaying the interface of a prescription system of an indian hospital.

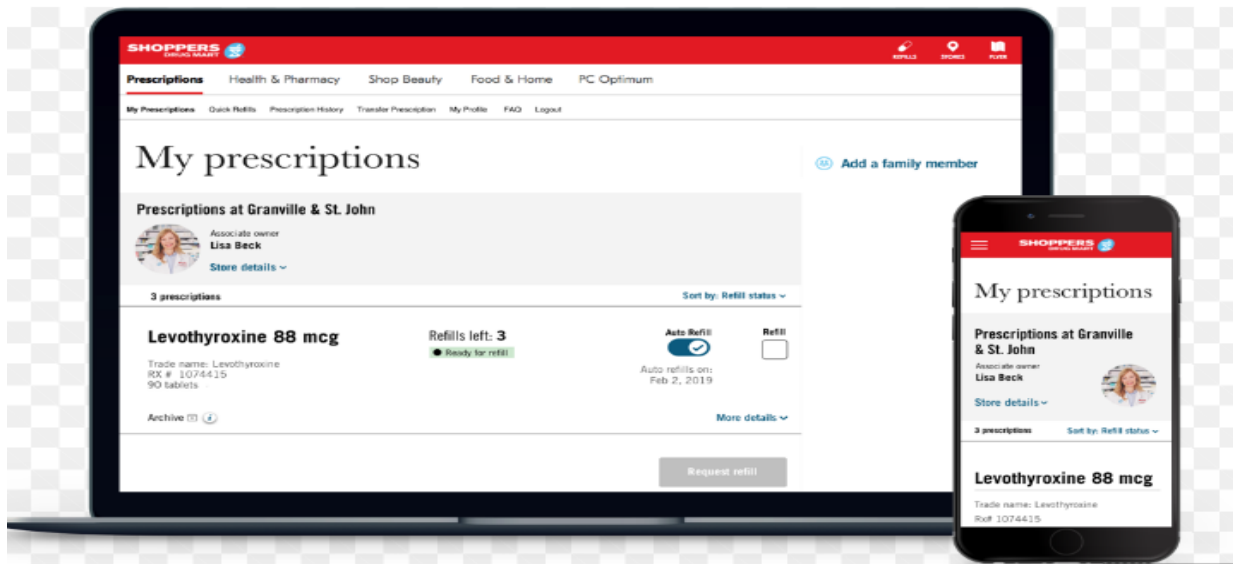


Figure 2.3: Global Hospitals Web Prescription including app

**Description:** Displaying the interface of a prescription system of a Global hospitals prescription.

### Comparison Table With Other System

SL	Feature Name	Description	Others	Prescription
1	Home	This is Landing Page	Yes	Yes
2	Email Prescription	<b>Prescription Soft Copy</b>	<b>No</b>	<b>Yes</b>
3	Registration	Applicant Registration Form	Yes	Yes
4	Login	Login admin/normal users	Yes	Yes
5	Doctor Registration	Doctor Registration	Yes	Yes
6	Doctor assistant Registration	Doctor assistant registration for helping doctor	<b>No</b>	<b>Yes</b>
7	Previous History	All previous history display	<b>No</b>	<b>Yes</b>
8	Admin panel login	Login admin panel	Yes	Yes
9	Doctor Dept. wise Config	Prescription interface as per doctor department	<b>No</b>	Yes
10	Patient Login	Patient login for download prescription	<b>No</b>	Yes
11	Other configuration	Medicine, advice, investigation etc config	Yes	Yes
12	Management Related all Report	Display all management related report	<b>No</b>	<b>Yes</b>

# Chapter 3

## System Analysis

### 3.1 Introduction

Software or system analysis is an area in which analysts are regularly learn new things and approaches to accurately gather, understand, maintain and create more effective and efficient software systems. So, here we discuss how we analyze my systems to achieve the goal. How we collect all of our requirement, analyze all those requirements and then finalize what actually should take for consideration to build the system for a better experience.

### 3.2 Software Requirement Specification (SRS)

#### 3.2.1 System Environment

The Admin, Doctor and Patient accesses the Prescription through the Internet and locally both. Admin make the configuration and doctors create the prescription for the patient. Patient can login to the system and download his/her prescription only.

#### 3.2.2 Users Characteristic

- The patient is supposed to have some knowledge about Internet and be able to use a search engine. He should search the prescription link and login with his credential and just download his prescription.
- The Admin is expected to be Internet literate and to be able to use email, software. He has to configure the whole system for the hospitals to make it usable for the doctors to write prescription.
- The Doctors are expected to use software. So they can make the prescription for patients. They just have to entry medicine name, investigation name and advice. They also see various report. They just search them. All will be pre-configured.

Moreover the specifications are....

Add Medicine, Update Medicine, Delete Medicine, Search Medicine, Add User, Add Doctor, Add Patient, Add/Edit Strength, Add/Edit Medicine Type, Add/Edit Generic, Add/Edit Chamber etc.

### 3.3 Detailed Requirement Specification

#### 3.3.1 Functional Requirement

The Functional requirement is what the software offer and its full description.

Pre-configuration for medicine and Prescription like medicine type, generic entry and some others.

<b>Triggering</b>	Admin selects to add/edit new entities to the database.
<b>Pre-condition</b>	Admin already access the configuration page.
<b>General Path</b>	1. The software shows a blank div to enter all the required data. 2. The admin enters the information and submit the form. 3. Software check that the required fields are blank or not and updates the database.
<b>Alternative Paths</b>	In the second step, either the field is blank, the Editor is instructed to add an entry. Validation for correctness is not made.
<b>Post-Condition</b>	All entities has been added to the database.
<b>Exceptions</b>	Admin may abandon the operations at any time.

#### Add Patient and Doctor

<b>Triggering</b>	Admin selects to add new patient and doctor to the database.
<b>Pre-condition</b>	Admin has accessed the configuration page.
<b>General Path</b>	1. The software shows a blank div to enter the required information. 2. The admin enters the informations and submit the form. 3. Software check that the required fields are blank or not and updates the database.
<b>Alternative-Paths</b>	In the second step , either the field is blank, the Editor is instructed to add an entry.

	Validation for correctness is not made.
<b>Post Conditions</b>	The patient and doctor has been added to the database.
<b>Exception Paths</b>	Admin may abandon the operations at any time.
<b>Trigger</b>	Admin selects to add new entities to the database.

## Create Prescription

<b>Triggering</b>	The doctor selects to create new prescription to the database.
<b>Pre-condition</b>	The doctor has accessed the configuration screen.
<b>General Path</b>	<ol style="list-style-type: none"> <li>1. The software shows a blank div to enter the required information.</li> <li>2. The admin enters the information and submit the form.</li> <li>3. Software check that the required fields are blank or not and updates the database.</li> </ol>

## Edit Prescription

<b>Triggering</b>	The doctor selects to update existing prescription to the database.
<b>Pre-condition</b>	The doctor has accessed the configuration screen.
<b>General Path</b>	<ol style="list-style-type: none"> <li>1. The software shows a blank div to enter the required information.</li> <li>2. The admin enters the information and submit the form.</li> <li>3. Software check that the required fields are blank or not and updates the database.</li> </ol>

### 1.3.2 Non Functional Requirement

#### 3.1 Logical Structure of the Data

The logical structure of the data is going to be stored in the internal database as given below.

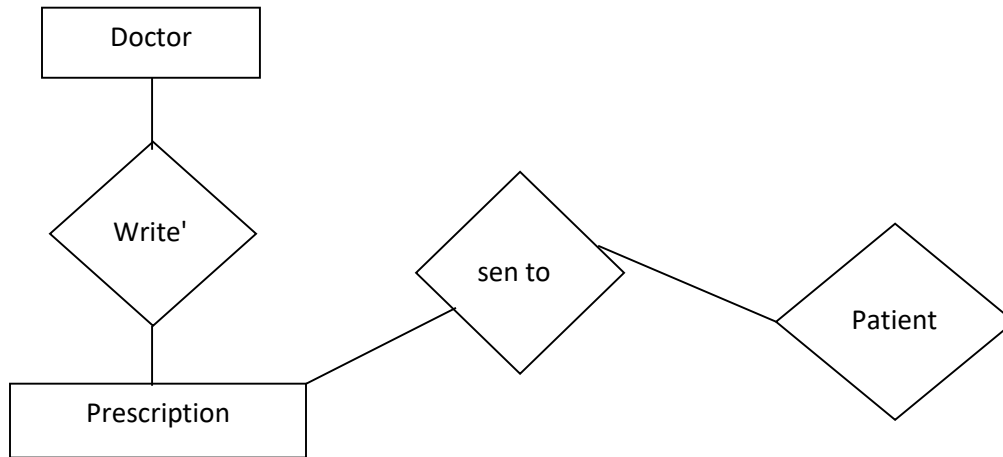


Figure 3.1 - Logical Structure of the Prescription Data



# Chapter 4

## System Design

### 4.1 PHP and React Analysis and Design

PHP is a High-level Programming language and web framework Laravel that motivates perfunctory development and clean, technical design. Now a days it is the most powerful and common web framework for web development. I have used PHP 7.4 and Laravel Framework (MVC) and GUI for HTML5, CSS3 Specially React to develop Online Prescription. There we use REST API.

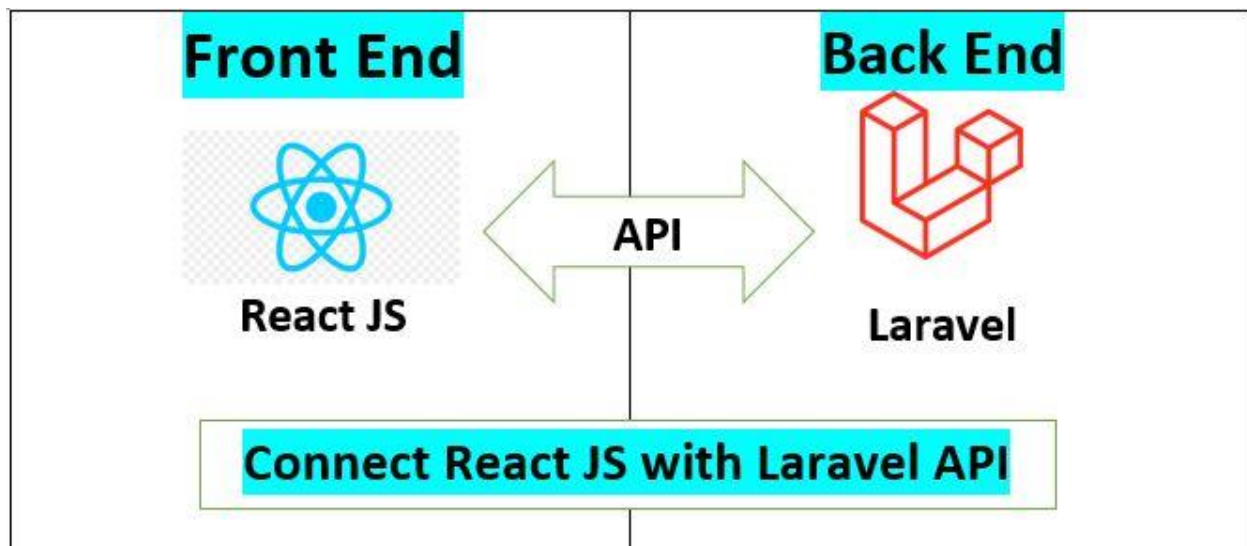


Figure 4.1: Showing How to connect React JS with Laravel API

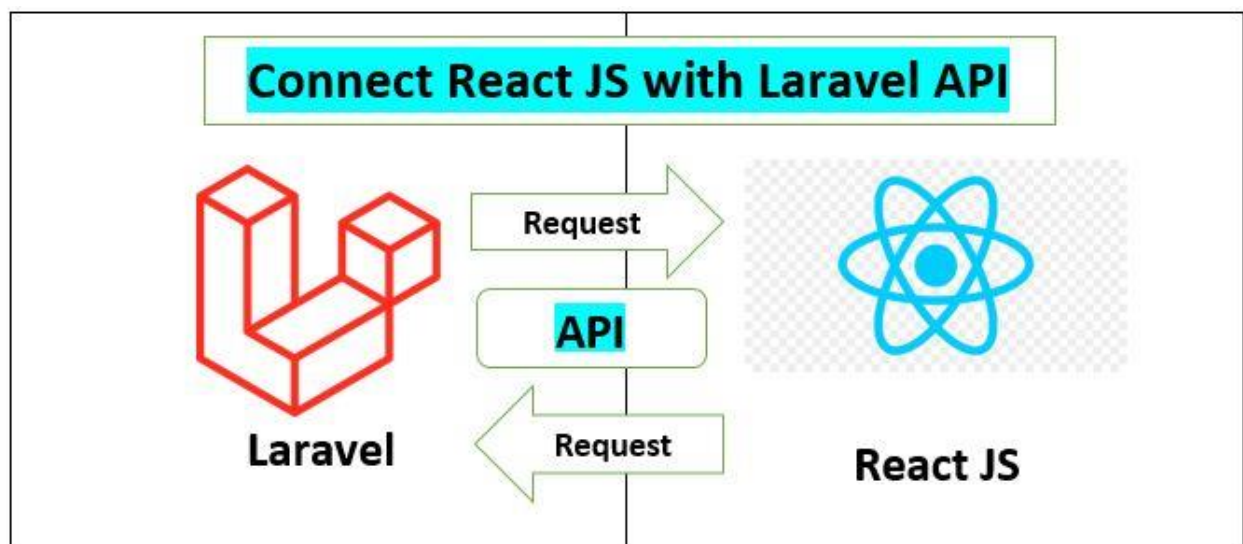


Figure 4.2: How Laravel API and React Request Response

## 4.2 Use Case Diagram

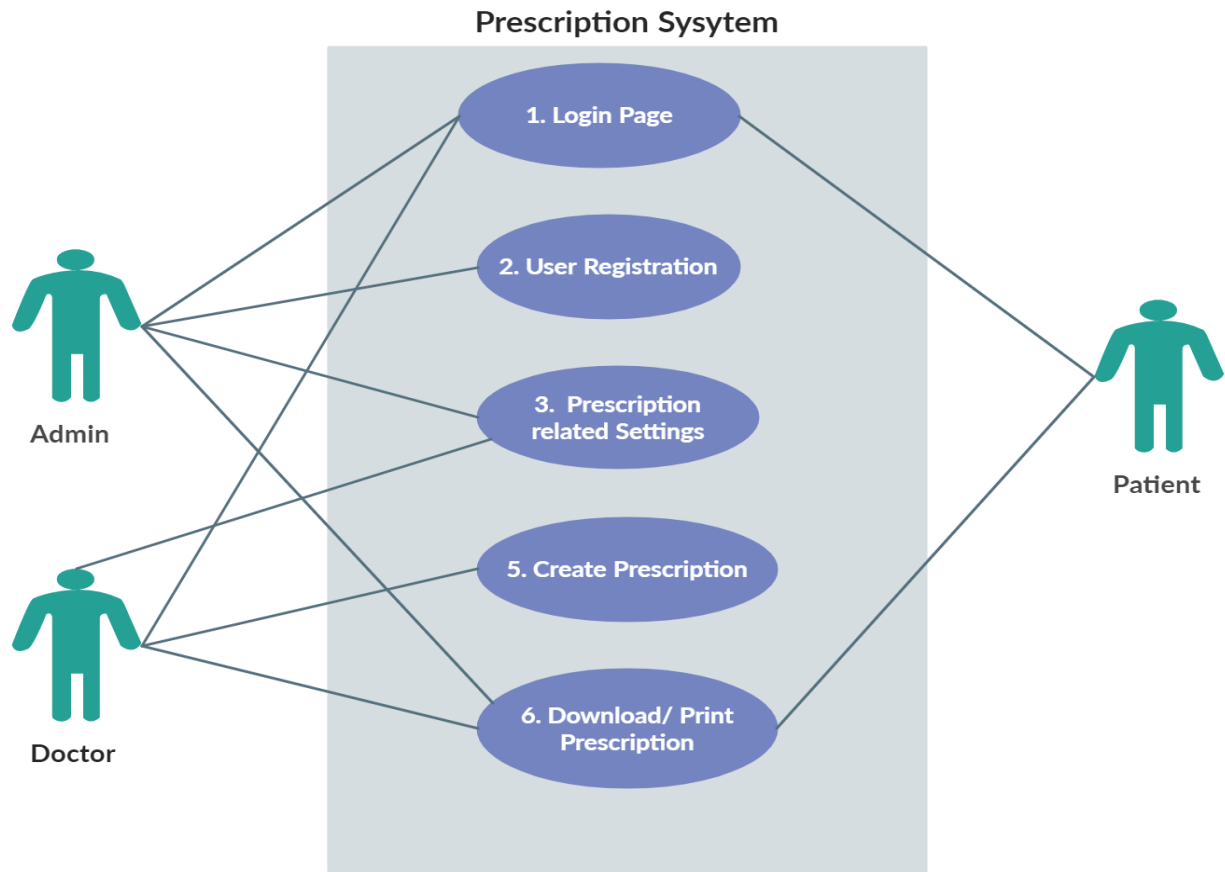


Figure 4.3: Use Case diagram

## 4.3 Use Case Description:

- 4.3.1 **Login Page:** Every user have to log in to the system first.
- 4.3.2 **User Registration:** Only Admin can register other admin, doctors and patient. After that they can login to their portal.
- 4.3.3 **Prescription related settings:** There are so many prescription related settings. Like medicine, investigation, advice, medicine supplier, medicine type, strength and so on. Both doctor and admin can configure all this sort of works.
- 4.3.4 **Create prescription:** Only doctors can use this option. They can create, edit or modify the previous prescription.
- 4.3.5 **Download/Print Prescription:** All the user of this system can see the prescription and print the prescription copy.

#### 4.4 Entity Relationship Diagram (ERD)

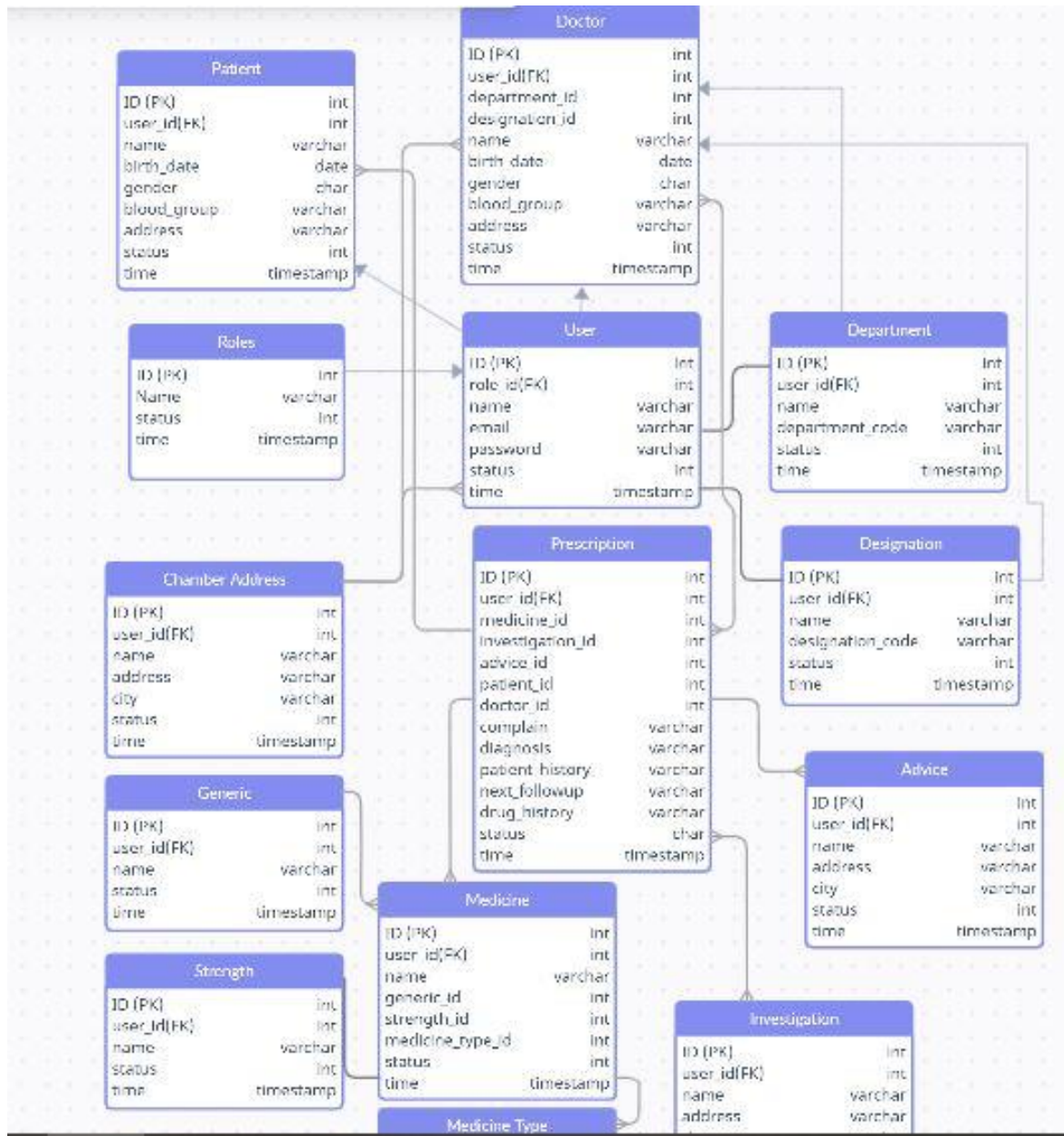


Figure 4.4: ER Diagram

**Description:** The above ER Diagram shows the relationships between the tables.

## 4.5 Database Screenshots

The screenshot shows the phpMyAdmin interface for a database named 'react\_laracrud'. The left sidebar lists various tables, and the main panel displays a list of 31 tables. Each table entry includes a checkbox, the table name, a star icon, and a set of icons for actions like Browse, Structure, Search, Insert, Empty, and Drop. The table names are: investigations, medicines, medicine\_advice, medicine\_types, migrations, oauth\_access\_tokens, oauth\_auth\_codes, oauth\_clients, oauth\_personal\_access\_clients, oauth\_refresh\_tokens, password\_resets, patients, prescriptions, pres\_advice, pres\_investigations, pres\_medicines, roles, strengths, suppliers, users, and visiting\_fees. The bottom row shows a 'Sum' of 113 tables with a total size of 1.2 MiB.

Table Name	Engine	Charset	Collation	Size	Index
investigations	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-	-
medicines	InnoDB	utf8mb4_unicode_ci	96.0 KiB	-	-
medicine_advice	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-	-
medicine_types	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-	-
migrations	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-	-
oauth_access_tokens	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-	-
oauth_auth_codes	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-	-
oauth_clients	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-	-
oauth_personal_access_clients	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-	-
oauth_refresh_tokens	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-	-
password_resets	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-	-
patients	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-	-
prescriptions	InnoDB	utf8mb4_unicode_ci	64.0 KiB	-	-
pres_advice	InnoDB	utf8mb4_unicode_ci	64.0 KiB	-	-
pres_investigations	InnoDB	utf8mb4_unicode_ci	64.0 KiB	-	-
pres_medicines	InnoDB	utf8mb4_unicode_ci	64.0 KiB	-	-
roles	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-	-
strengths	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-	-
suppliers	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-	-
users	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-	-
visiting_fees	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-	-
<b>Sum</b>	<b>113 InnoDB</b>	<b>utf8mb4_unicode_ci</b>	<b>1.2 MiB</b>	<b>0 B</b>	

Figure 4.5: Prescription Database Table

**Fig 4.5 Description:** In the above figure it shows table structures of my projects database. To complete this project I used 31 tables. Below I also showing some of the major tables screenshot.

The screenshot shows the 'Table structure' view for the 'users' table in the 'react\_laracrud' database. The table has 9 columns: id, role\_id, name, email, email\_verified\_at, password, remember\_token, created\_at, and updated\_at. Each column has a checkbox, a name, a type, a collation, attributes, nullability, default value, comments, extra options, and an action menu.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	role_id	int(10)		UNSIGNED	No	None			Change Drop More
3	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
4	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	email_verified_at	timestamp			Yes	NULL			Change Drop More
6	password	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
7	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
8	created_at	timestamp			Yes	NULL			Change Drop More
9	updated_at	timestamp			Yes	NULL			Change Drop More

Figure 4.6: Prescription User Table



**Fig 4.6 Description:** In the above figure it shows users table and the columns I have used to preserve the users data.

Server: localhost » Database: react\_laracrud » Table: patients

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/> 2	patient_code	varchar(100)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	first_name	varchar(50)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 4	last_name	varchar(50)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 5	full_name	varchar(50)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/> 6	email	varchar(20)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 7	phone	varchar(20)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 8	address	varchar(150)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/> 9	city	varchar(30)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 10	state	varchar(30)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 11	post_code	varchar(15)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 12	district	varchar(30)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 13	nid	varchar(30)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 14	dob	date			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 15	blood_group	char(30)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 16	gender	char(10)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/> 17	image	varchar(200)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 18	user_id	int(10)		UNSIGNED	No	None			Change  Drop  More
<input checked="" type="checkbox"/> Console	tatus	tinyint(1)			No	1			Change  Drop  More

Figure 4.7: Prescription Patient Table

**Fig 4.7 Description:** In the above figure it shows patients table and the columns I have used to preserve the patients data.

Server: localhost » Database: react_laracrud » Table: doctors										
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers										
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/> 1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT		Change  Drop  More
<input type="checkbox"/> 2	name	varchar(50)	utf8mb4_unicode_ci		No	None				Change  Drop  More
<input type="checkbox"/> 3	email	varchar(20)	utf8mb4_unicode_ci		Yes	NULL				Change  Drop  More
<input type="checkbox"/> 4	phone	varchar(20)	utf8mb4_unicode_ci		Yes	NULL				Change  Drop  More
<input type="checkbox"/> 5	department_id	int(10)		UNSIGNED	No	None				Change  Drop  More
<input type="checkbox"/> 6	designation_id	int(10)		UNSIGNED	No	None				Change  Drop  More
<input type="checkbox"/> 7	dob	date			Yes	NULL				Change  Drop  More
<input type="checkbox"/> 8	blood_group	char(30)	utf8mb4_unicode_ci		Yes	NULL				Change  Drop  More
<input type="checkbox"/> 9	about_me	varchar(250)	utf8mb4_unicode_ci		Yes	NULL				Change  Drop  More
<input type="checkbox"/> 10	image	varchar(200)	utf8mb4_unicode_ci		Yes	NULL				Change  Drop  More
<input type="checkbox"/> 11	experience	varchar(200)	utf8mb4_unicode_ci		Yes	NULL				Change  Drop  More
<input type="checkbox"/> 12	speciality	varchar(200)	utf8mb4_unicode_ci		Yes	NULL				Change  Drop  More
<input type="checkbox"/> 13	degrees	varchar(200)	utf8mb4_unicode_ci		No	None				Change  Drop  More
<input type="checkbox"/> 14	user_id	int(10)		UNSIGNED	No	None				Change  Drop  More
<input type="checkbox"/> 15	status	tinyint(1)			No	1				Change  Drop  More
<input type="checkbox"/> 16	created_at	timestamp			Yes	NULL				Change  Drop  More
<input type="checkbox"/> 17	updated_at	timestamp			Yes	NULL				Change  Drop  More

Figure 4.8: Prescription Doctor Table

**Fig 4.8 Description:** In the above figure it shows doctors table and the columns I have used to preserve the doctors data.

Server: localhost » Database: react_laracrud » Table: medicines										
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers										
Table structure		Relation view								
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/> 1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT		Change  Drop  More
<input type="checkbox"/> 2	medicine_code	varchar(50)	utf8mb4_unicode_ci		No	None				Change  Drop  More
<input type="checkbox"/> 3	name	varchar(50)	utf8mb4_unicode_ci		No	None				Change  Drop  More
<input type="checkbox"/> 4	supplier_id	int(10)		UNSIGNED	No	None				Change  Drop  More
<input type="checkbox"/> 5	generic_id	int(10)		UNSIGNED	No	None				Change  Drop  More
<input type="checkbox"/> 6	strength_id	int(10)		UNSIGNED	No	None				Change  Drop  More
<input type="checkbox"/> 7	medicine_type_id	int(10)		UNSIGNED	No	None				Change  Drop  More
<input type="checkbox"/> 8	details	varchar(150)	utf8mb4_unicode_ci		Yes	NULL				Change  Drop  More
<input type="checkbox"/> 9	side_effect	varchar(50)	utf8mb4_unicode_ci		Yes	NULL				Change  Drop  More
<input type="checkbox"/> 10	favourite	tinyint(1)			No	1				Change  Drop  More
<input type="checkbox"/> 11	user_id	int(10)		UNSIGNED	No	None				Change  Drop  More
<input type="checkbox"/> 12	status	tinyint(1)			No	1				Change  Drop  More
<input type="checkbox"/> 13	created_at	timestamp			Yes	NULL				Change  Drop  More
<input type="checkbox"/> 14	updated_at	timestamp			Yes	NULL				Change  Drop  More

Figure 4.9: Prescription Medicine Table

**Fig 4.9 Description:** In the above figure it shows medicine table and the columns I have used to preserve medicines data.

Column	Data Type	Collation	Unsigned	Zero	Null	Actions
2 patient_id	int(10)		UNSIGNED	No	None	Change Drop More
3 doctor_id	int(10)		UNSIGNED	No	None	Change Drop More
4 date	varchar(50)	utf8mb4_unicode_ci		No	None	Change Drop More
5 bp	varchar(50)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
6 pulse	varchar(50)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
7 temp	varchar(50)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
8 weight	varchar(50)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
9 spo2	varchar(50)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
10 sugar	varchar(50)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
11 complain	varchar(200)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
12 diagnosis	varchar(200)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
13 past_history	varchar(200)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
14 drug_history	varchar(200)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
15 follow_up	varchar(200)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
16 others1	varchar(200)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
17 others2	varchar(200)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
18 others3	varchar(200)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
19 others4	varchar(200)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More
20 others5	varchar(200)	utf8mb4_unicode_ci		Yes	NULL	Change Drop More

Figure 4.10: Prescription Table


**Fig 4.10 Description:** In the above figure it shows prescriptions table and the columns I have used to preserve the prescription data.

# Chapter 5


## Implementation & Result

### 5.1 Technology Used


#### 4.1.1 Admin Interface Design:

 HTML5.0, CSS3, React JS.

#### 4.1.2 Frontend and Backend Programming:

 Laravel 8.0 with Restful API.

#### 4.1.3 Backend Database

 Mysql database.

### 5.2 Hardware Requirement for php7.4, REST API, React and mysql database

- ✓ Required Memory: 5 GB.
- ✓ Required CPU: Intel Core i3-2340UE.
- ✓ Required File Size: 4 GB.
- ✓ Required OS: Windows ,Linux

This are the minimum require hardware information for running online prescription systems.

### 5.3: GUI Design & Test Case

UI/UX design makes an active communication way between a user and a computer system. As people will the interface so it has to be clear and complete because it depict a user's perception of the system. As we know that the main thing of all software engineering process model is understanding the requirements before we go to design the system. We analysis about the ui/ux design before start to design the system.



### 5.3.1 Prescription Dashboard

The screenshot displays the 'New Prescription' form in a web application. The sidebar on the left contains the following menu items: PRESCRIPTION, NEW PRESCRIPTION (highlighted), OLD PRESCRIPTION, COMMON, CHAMBER, DEPARTMENT, DESIGNATION, USER, and DOCTOR. The main form area is titled 'New Prescription' and includes a dropdown menu and a 'Log out' link. The form contains the following fields and sections:

- Patient Information:** Fields for Code, Patient Name (with a search bar labeled 'search by name,code,phoi'), Age, Gender, and Phone No.
- Chief Complain:** A field with a '+' icon.
- On Examination:** A field with a '+' icon.
- Rx (Prescription):** A table with columns for Medicine Name, Dose, Duration, and Instruction, and an 'ADD' button.

Figure 5.1: Dashboard Design

**Fig 5.1 Description:** The above image shows the main prescription panel. Where doctor write prescription for patients.

### 5.3.2 Sidebar Menu

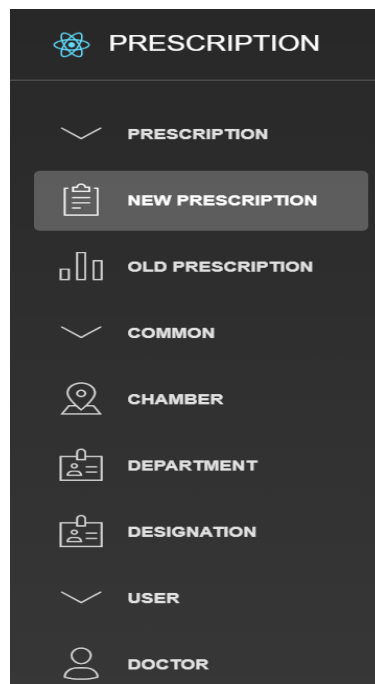


Figure 5.2: sidebar menu Page Design

**Fig 5.2 Description:** The above image shows the sidebar of the prescription panel. Where all menus are given.

### 5.1.3 Add Doctor Page

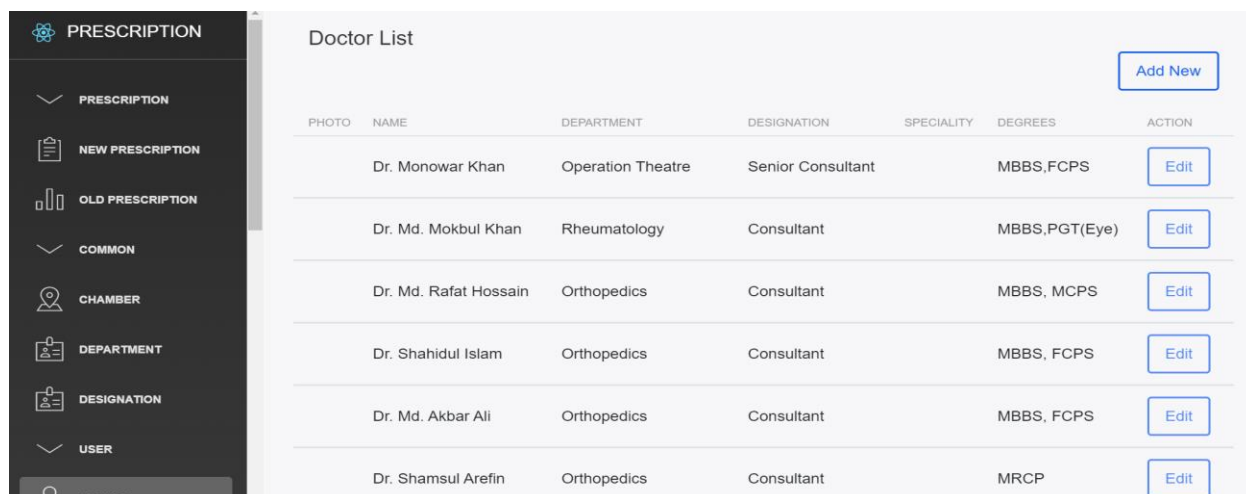
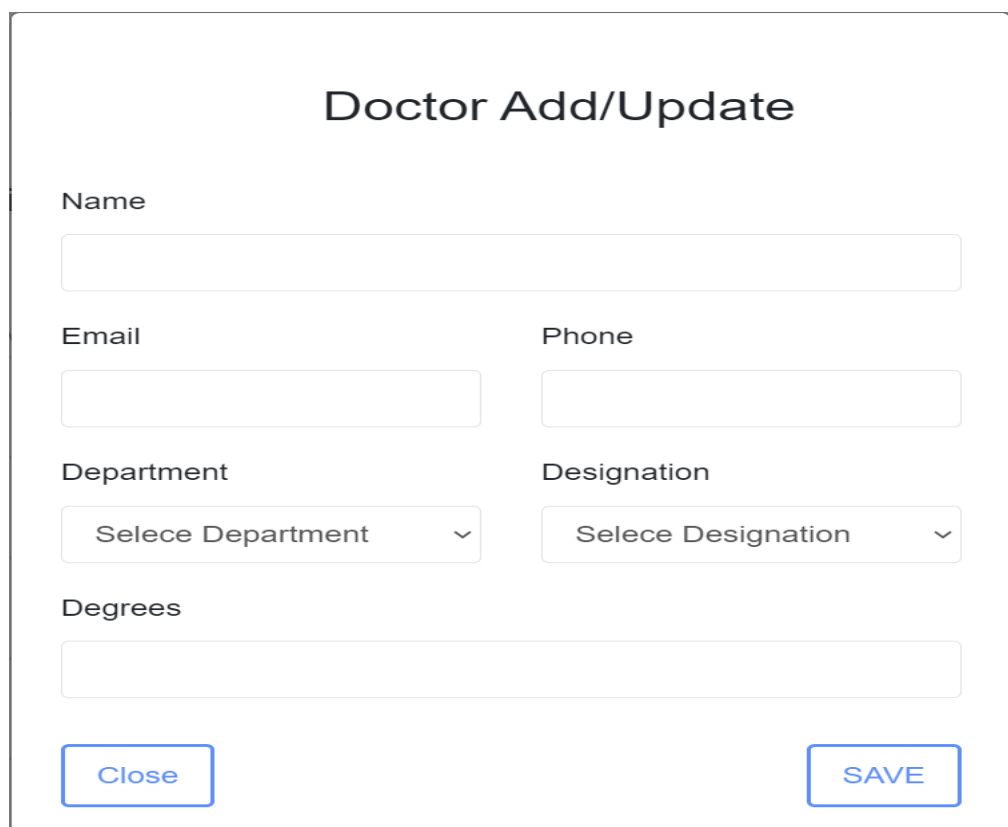


PHOTO	NAME	DEPARTMENT	DESIGNATION	SPECIALITY	DEGREES	ACTION
	Dr. Monowar Khan	Operation Theatre	Senior Consultant		MBBS,FCPS	<a href="#">Edit</a>
	Dr. Md. Mokbul Khan	Rheumatology	Consultant		MBBS,PGT(Eye)	<a href="#">Edit</a>
	Dr. Md. Rafat Hossain	Orthopedics	Consultant		MBBS, MCPS	<a href="#">Edit</a>
	Dr. Shahidul Islam	Orthopedics	Consultant		MBBS, FCPS	<a href="#">Edit</a>
	Dr. Md. Akbar Ali	Orthopedics	Consultant		MBBS, FCPS	<a href="#">Edit</a>
	Dr. Shamsul Arefin	Orthopedics	Consultant		MRCP	<a href="#">Edit</a>



### Doctor Add/Update

Name

Email

Phone

Department

Selece Department

Designation

Selece Designation

Degrees

Close

SAVE

Figure 5.3: Doctor add page design

**Fig 5.3 Description:** The above image shows the Doctor Add page of prescription.

### 5.1.4 Add Patient Page:

OLD PRESCRIPTION  
COMMON  
CHAMBER  
DEPARTMENT  
DESIGNATION  
USER  
DOCTOR  
PATIENT  
MEDICINE  
GENERIC

Patient

Patient L

NAME

Manjur

Zakir

Dropdown ▾ Log out

Add New

GENDER

m

m

© 2022 Copyright @ ePrescription

ADD

Close

Full Name

Phone

Date of Birth

mm/dd/yyyy

Blood Group

Selec blood Group ▾

Gender

Selec Gender ▾

Address

Address

Figure 5.4: Patient add page design

**Fig 5.3 Description:** The above image shows the Patient Add page of prescription.

### 5.1.5 Add Medicine:

DOCTOR  
PATIENT  
MEDICINE  
GENERIC  
MEDICINE TYPE  
STRENGTH  
SUPPLIER  
MEDICINE  
INVESTIGATION & ADVICE  
INVESTIGATION

Medicine

Medicine

TYPE

Tab.

Tab.

susp.

Dropdown ▾ Log out

Add New

ACTION

Edit

Edit

Edit

© 2022 Copyright @ ePrescription

ADD

Close

Full Name

Medicine Type

Select Medicine Type ▾

Strength

Select Strength ▾

Generic

Select Generic ▾

Supplier

Select Supplier ▾

Side Effect

Side Effect

Figure 5.5: Medicine Add page design

**Fig 5.3 Description:** The above image shows the Medicine Add page of prescription.

### 5.1.6 Add Investigation Page:

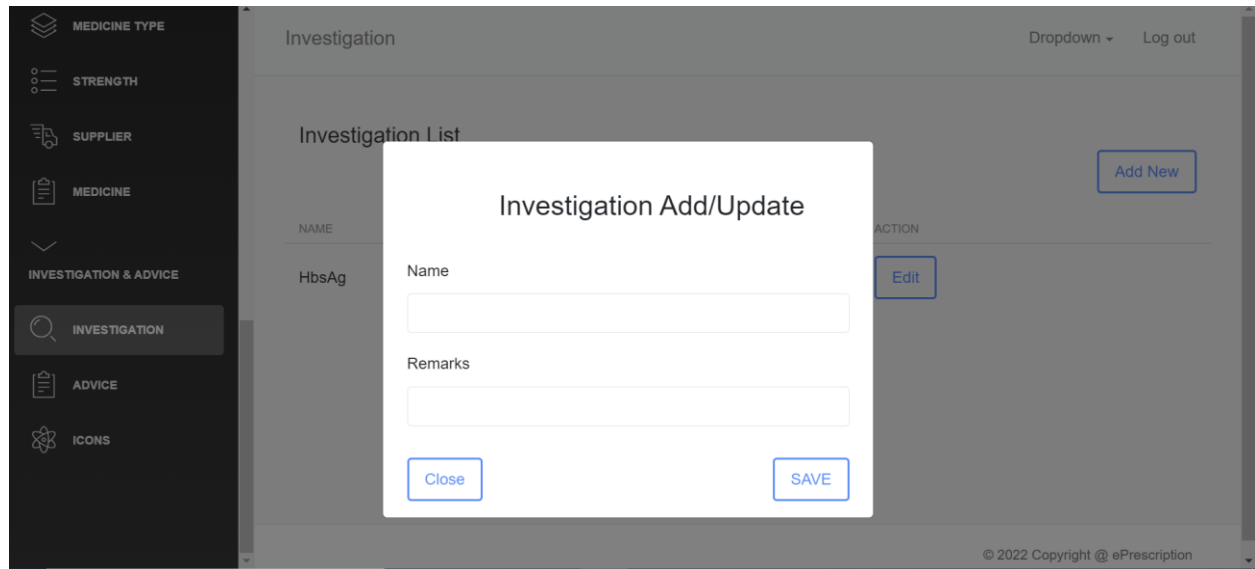


Figure 5.6: Investigation insert page design

**Fig 5.3 Description:** The above image shows the Investigation Add page of prescription.

### 5.1.7 Prescription Page:

ডাঃ রেজা আহমেদ এম.বি.বি.এস (ডি.এম.সি), এমডি(মেডিসিন) ডিপার্টমেন্ট অব কার্ডিওলজি জাতীয় হৃদরোগ ইনস্টিটিউট		Dr. Reza Ahmed MBBS (DMC), MD(Medicine) Department of Cardiology National Institute of Cardio Vascular Disease												
New Patient +														
Code :	Patient Name : search by name,code,phone..	Age :	Gender :	Phone No :										
Chief Complain +		Rx <table><thead><tr><th>Medicine Name</th><th>Dose</th><th>Duration</th><th>Instruction</th><th>ADD</th></tr></thead><tbody><tr><td colspan="5">Advice +</td></tr></tbody></table>			Medicine Name	Dose	Duration	Instruction	ADD	Advice +				
Medicine Name	Dose				Duration	Instruction	ADD							
Advice +														
On Examination +														
Diagnosis +														
Investigation +														
Next Follow Up		Save & Print												

Figure 5.7: Prescription page design

## 5.1 Testing

### 5.2.1 Overview

Testing is the process by which we can evaluate a system or its component(s). Where it meets the certain requirement we expect or reject. This steps outcomes in the actual expectation and difference between the outcomes. In easy to say testing execute a way in order to identify any kind of problems, errors or missing requirement. on the other hand to the actual expectation or requirement.

### 5.3.3 Appendix:

#### 5.3.1 User Sign in System:

Sl:	Input:	Click Btn:	Expectation results:	Pass or fail
1.	Nul, Nul	sign in	displayed <b>Error</b>	Passed
2.	Nul, 12345678	sign in	displayed <b>Error</b>	Passed
3.	12346789, Nul	sign in	displayed lay <b>Error</b>	Passed
4.	12346789, admin	sign in	displayed <b>Error</b>	Passed
5.	12345789, @\$%*&!~(){}[]	sign in	displayed <b>Error</b>	Passed
6.	Admi n, @\$%*&!~(){}[]	sign in	displayed <b>Error</b>	Passed
7.	Adm in, 12345789	sign in	displayed <b>Success</b>	Passed
8.	Adm in, admin1236789	sign in	displayed <b>Success</b>	Passed
9.	Admi n, 1235789admin	sign in	displayed <b>Success</b>	Passed
10.	DMIN, 12456789	sign in	displayed <b>Success</b>	Passed
11.	ADIN, admin123456789	sign in	displayed <b>Success</b>	Passed
12.	AMIN, 123456789admin	sign in	displayed <b>Success</b>	Passed
13.	ADIN, ADMIN	sign in	displayed <b>Success</b>	Passed
14.	admin, ADMIN 12345689	sign in	displayed <b>Success</b>	Passed
15.	admin, 12345678ADMIN	sign in	displayed <b>Success</b>	Passed
16.	ADmin, AdminN	sign in	displayed <b>Success</b>	Passed

Table 4: white box testing of "login system"

### 5.3.2 Configure Insert Systems:

SL.	Inputs:	Click Button:	Expect result:	Pass/fail
1.	Nul, Config name	addBtn	displayed <b>Error</b>	Passed
2.	Nul, Nul, Null, captha	addBtn	displayed <b>Error</b>	Passed
3.	Nul, Nul, Config name cd,	addBtn	displayed <b>Error at least 3 character</b>	Passed

## 5.2 Coding

### 5.3.1 Restful API for interacting with database and frontend

#### Controller List

<?php

```
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\User\AuthController;
use App\Http\Controllers\User\UserController;
use App\Http\Controllers\Common\ChamberController;
use App\Http\Controllers\Common\DepartmentController;
use App\Http\Controllers\Common\DesignationController;
use App\Http\Controllers\Common\DoctorController;
use App\Http\Controllers\Common\PatientController;
use App\Http\Controllers\Common\InvestigationController;
use App\Http\Controllers\Common\GeneralAdviceController;
use App\Http\Controllers\Common\VisitingFeeController;
use App\Http\Controllers\Medicine\GenericController;
use App\Http\Controllers\Medicine\MedicineController;
use App\Http\Controllers\Medicine\MedicineTypeController;
use App\Http\Controllers\Medicine\StrengthController;
use App\Http\Controllers\Medicine\SupplierController;
use App\Http\Controllers\Medicine\DoseDurationAdviceController;
use App\Http\Controllers\Prescription\PrescriptionController;
```

## Route List

// Register Routes

```
Route::post('/register',[AuthController::class, 'Register']);
```

// Login Routes

```
Route::post('/login',[AuthController::class, 'Login']);
```

// Current User Route

```
Route::get('/user',[UserController::class, 'User']);
```

```
Route::get('/chamberaddress',[ChamberController::class, 'index']);
```

```
Route::post('/storechamberaddress',[ChamberController::class, 'storechamber']);
```

```
Route::get('/getdepartments',[DepartmentController::class, 'allDepartments']);
```

```
Route::post('/storedepartment',[DepartmentController::class, 'storedepartment']);
```

```
Route::get('/getdesignations',[DesignationController::class, 'allDesignations']);
```

```
Route::post('/storedesignation',[DesignationController::class, 'storedesignation']);
```

```
Route::get('/getalldoctors',[DoctorController::class, 'allDoctors']);
```

```
Route::post('/addnewdoctor',[DoctorController::class, 'addDoctor']);
```

```
Route::post('/updatedoctor/{id}',[DoctorController::class, 'updateDoctor']);
```

```
Route::get('/getallpatients',[PatientController::class, 'allPatients']);
```

```
Route::get('/searchpatients/{key}',[PatientController::class, 'searchPatients']);
```

```
Route::post('/addnewpatient',[PatientController::class, 'addPatient']);
```

```
Route::post('/updatepatient/{id}',[PatientController::class, 'updatePatient']);
```

```
Route::get('/getGenerics',[GenericController::class, 'allGenerics']);
```

```
Route::post('/addgeneric',[GenericController::class, 'addGeneric']);
```

```
Route::post('/updategeneric/{id}',[GenericController::class, 'updateGeneric']);
```

```
Route::get('/getMedicineTypes',[MedicineTypeController::class, 'allMedicineTypes']);
```

```
Route::post('/addMedicineTypes',[MedicineTypeController::class, 'addMedicineTypes']);
```

```
Route::post('/updateMedicineTypes/{id}',[MedicineTypeController::class, 'updateMedicineTypes']);
```

```

Route::get('/getStrength',[StrengthController::class, 'allStrength']);
Route::post('/addStrength',[StrengthController::class, 'addStrength']);
Route::post('/updateStrength/{id}',[StrengthController::class, 'updateStrength']);

Route::get('/getsupplier',[SupplierController::class, 'allSupplier']);
Route::post('/addsupplier',[SupplierController::class, 'addSupplier']);
Route::post('/updatesupplier/{id}',[SupplierController::class, 'updateSupplier']);

Route::get('/getMedicines',[MedicineController::class, 'allMedicine']);
Route::post('/AddMedicines',[MedicineController::class, 'addMedicine']);
Route::post('/updateMedicines/{id}',[MedicineController::class, 'updateMedicine']);
Route::get('/searchmedicines/{key}',[MedicineController::class, 'searchMedicines']);

Route::get('/alldoseDurationAdvice',[DoseDurationAdviceController::class, 'allDoseDuration']);
Route::post('/adddoseDurationAdvice',[DoseDurationAdviceController::class,
'addDoseDuration']);

Route::get('/allinvestigation',[InvestigationController::class, 'allInvestigation']);
Route::post('/addinvestigation',[InvestigationController::class, 'addInvestigation']);
Route::post('/updateinvestigation/{id}',[InvestigationController::class, 'updateInvestigation']);
Route::get('/searchinvestigations/{key}',[InvestigationController::class, 'searchInvestigations']);

Route::get('/allgeneralAdvice',[GeneralAdviceController::class, 'allGeneralAdvice']);
Route::post('/addgeneralAdvice',[GeneralAdviceController::class, 'addGeneralAdvice']);
Route::post('/updateadvice/{id}',[GeneralAdviceController::class, 'updateAdvice']);
Route::get('/searchadvices/{key}',[GeneralAdviceController::class, 'searchAdvice']);

Route::get('/allVisitingFees',[VisitingFeeController::class, 'allVisitingFees']);
Route::post('/addVisitingFees',[VisitingFeeController::class, 'addVisitingFees']);
Route::post('/updateVisitingFees/{id}',[VisitingFeeController::class, 'updateVisitingFees']);

Route::post('/createPrescription',[PrescriptionController::class, 'newPrescription']);
Route::get('/allPrescription',[PrescriptionController::class, 'allPrescription']);

```

### 5.2.2 React Code

```

6  import React, { Component } from 'react'
7  import { BrowserRouter, Route, Switch, Redirect } from "react-router-dom";
8  import AppURL from 'api/AppURL';
9  import AdminLayout from "layouts/Admin.js";
10 import Login from "views/auth/Login";
11 import axios from 'axios';

```



```

12  import { useEffect , useState } from 'react';
13
14  export default function App() {
15
16    const [user,setUser] = useState([]);
17
18    useEffect(() => {
19      axios.get(AppURL.UserData).then(response => {
20        setUser(response.data);
21      }).catch(error => {
22
23      });
24    }, [])
25
26    return (
27      <>
28        <BrowserRouter>
29          <Switch>
30            <Route exact path="/" render={ (props) => <Login user={user} setUser={setUser} /> }
31            />
32            <Route exact path="/login" render={ (props) => <Login user={user} setUser={setUser}
33            /> } />
34            <Route path="/admin" render={ (props) => <AdminLayout user={user}
35            setUser={setUser} {...props} /> } />
36          </Switch>
37        </BrowserRouter>
38      </>
39    )
40  }

```

# **Chapter 6**

## **Conclusion and Future Works**

### **6.1 Conclusion**

I am glad to present the final software documentation on web based prescription. By this, the patients will get a clear and easy view of medicine names. To improve the efficiency of the prescription, online prescription is very essential. Web based prescription is more effective than a written prescription. Moreover this document can be used by the doctors, patients and also the undergrad students. We have tried our best to make it effective and fully designed documentation. We hope the reader will find it according to.

### **6.2 Future Works**

As this system is developed only for web system and there still lots of features to complete. So there have easy option to include any model or update. Future I will update technology and develop Mobile apps also. Besides more features will release to make it more easy and helpful for doctors. Patient dashboard will also be reach so that they can discuss with doctors with their problems.

## References[]:

1. Creating use case, creately,15-11-2022, [online]. Available:  
<https://app.creately.com/d/rWNGg4JUeCt/edit>
2. Creating ER diagram , creately,15-11-2022, [online]. Available:  
<https://app.creately.com/d/rWNGg4JUeCt/edit>
3. Laravel Related Documentation, Laravel, 10-11-2022, [online]. Available:  
<https://laravel.com/docs/8.x>
4. React related documentation , ReactJS,09-11-2022, [online]. Available:  
<https://reactjs.org/>
5. Refil Prescription systems , Refil , 17-09-2022, [online]. Available:  
**bit.ly/3GhGkDG**
6. Pro Clinic Prescription, Pro clinic, 17-09-2022, [online]. Available:  
<https://www.konnectplugins.com/proclinic/Vertical/index.html>
7. Zilsoft Prescription , Zilsoft ,17-09-2022, [online]. Available:  
<https://zilsoft.net/about/>
8. Amar chamber website landing page , Amar Chamber,17-09-2022, [online].  
Available: <https://amarchamber.com/>
9. For Multiple Prescription software , various ,15-11-2022, [online]. Available:  
<https://www.softwareadvice.com/medical/e-prescribing-comparison/>
10. IEEE Reference Format, IEEE ,15-11-2022, [online]. Available:  
<https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf>
11. DIMS Website, DIMS,15-10-2022, [online]. Available: <http://dimsbd.com/>
12. Square hospital Website, Square, 10-10-2022, [online]. Available:  
<https://www.squarehospital.com/>
13. Apollo hospital chennai website, Apollo hospital, 10-10-2022, [online].  
Available: <https://www.bdhci.com/apollo-chennai-office-in-bangladesh>
14. United Hospital website , United,10-10-2022, [online]. Available:  
<https://www.uhlbd.com/>

15. Brbhospital Website, brbhospital , 10-10-2022, [online]. Available:  
<https://www.brbhospital.com/>
16. Freepik website, freepik, 11-10-2022, [online]. Available:  
<https://www.freepik.com/free-photos-vectors/doctor-prescription>
17. Documentation , Study, 11-10-2022, [online]. Available: <https://study.com/>
18. Bootstrap website, bootstrap, 26-10-2022, [online]. Available:  
<https://getbootstrap.com/>
19. Free Icon website , flaticon , 26-10-2022, [online]. Available:  
<https://www.flaticon.com/>
20. Stack overflow website , Stack overflow , 26-10-2022, [online]. Available:  
<https://stackoverflow.com/>