**Web Based Online Prescription Systems**

**A Project Report**

Submitted to theDepartment of Computer Science and Engineering, Jahangirnagar University in partial fulfillment of the requirements for the degree of Master of science in Computer Science in PMSCS Program.

**Submitted By:**

Md. Mineuddin Ahmed Dipu

Roll: CSE 202103099

Batch: 27th

**Supervised By:**

Professor Dr. Md. Ezharul Islam



**Department of Computer Science and Engineering**

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

**Bangladesh**

**November 2022**

**Declaration**:

I, hereby, declare that the work 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 : 27th

**Acknowledgment**:

I would like to give special thanks to following people who have made the completion of this project possible:

My parents, for their continued support and generous 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 as well as for his support and guidance.

I would like to express 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 unyielding knowledge on anything computer or math related and his willingness to always help.

Arob Hossain (Younger Brother), for his unrested help to make the error vanished and also to the coding.

**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. 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**

**Chapter 1 Introduction Page No**

* 1. Motivation 7
  2. Objectives 7
  3. Scope of Project 7
  4. Problem Definition 8
  5. Research Methodology 8
  6. Outline 8

**Chapter 2 Background and Related Works**

2.1 Introduction 9

2.2 Background and History 9

2.3 Related Work and Gap Analyze 9

2.4 Gap Comparison Table 10-11

**Chapter 3 System Analysis**

3.1 Introduction 12

3.2 Software Requirements Specification 12

3.3 User Characteristics 12-13

3.4 Functional Requirement 13-14

3.5 Non-Functional Requirement 15

**Chapter 4 System Design**

4.1 Technology Used 16

4.2 PHP Analysis and Design 16

4.3 Use Case Diagram 17

4.4 ERD and Database Design 18

4.5 Table Structure 19-21

**Chapter 5 Implementation and Results**

5.1 GUI Design and Test Case 22-26

5.2 Testing 27-31

5.3 Coding 31-34

5.4 Hardware Requirement 35

**Chapter 6 Conclusion and Future Works**

5.1 Conclusion 36

5.2 Future Works 36

**References** 37

**List of Figures**

**Page No.**

Figure 2.1: Evanto Market’s Prescription 10

Figure 2.2: Indian Hospitals Prescription 10

Figure 2.3: Global Hospitals Web Prescription including app 11

Figure 3.1: - Logical Structure of the Prescription Data 15

Figure 4.1: Technology used in our project 16

Figure 4.2: How Laravel API and React work 17

Figure 4.3Use Case diagram 17

Figure 4.4: ER Diagram 18

Figure 4.5: Prescription Database Table 19

Figure 4.6: Prescription User Table 19

Figure 4.7: Prescription Patient Table 20

Figure 4.8: Prescription Doctor Table 20

Figure 4.9: Prescription Medicine Table 21

Figure 5.1: Dashboard design 22

Figure 5.2: Sidebar menu page design 23

Figure 5.3: Doctor add page design 24

Figure 5.4: Patient add page design 25

Figure 5.5: Medicine Add page design 25

Figure 5.6: Investigation insert page design 26

Figure 5.7 Prescription page design 26

# Chapter 1

**Introduction**

* 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. **Objectives**

The main objective is to obtain doctor’s 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 Scope of Project:**

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. **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. **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. **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**

* 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.

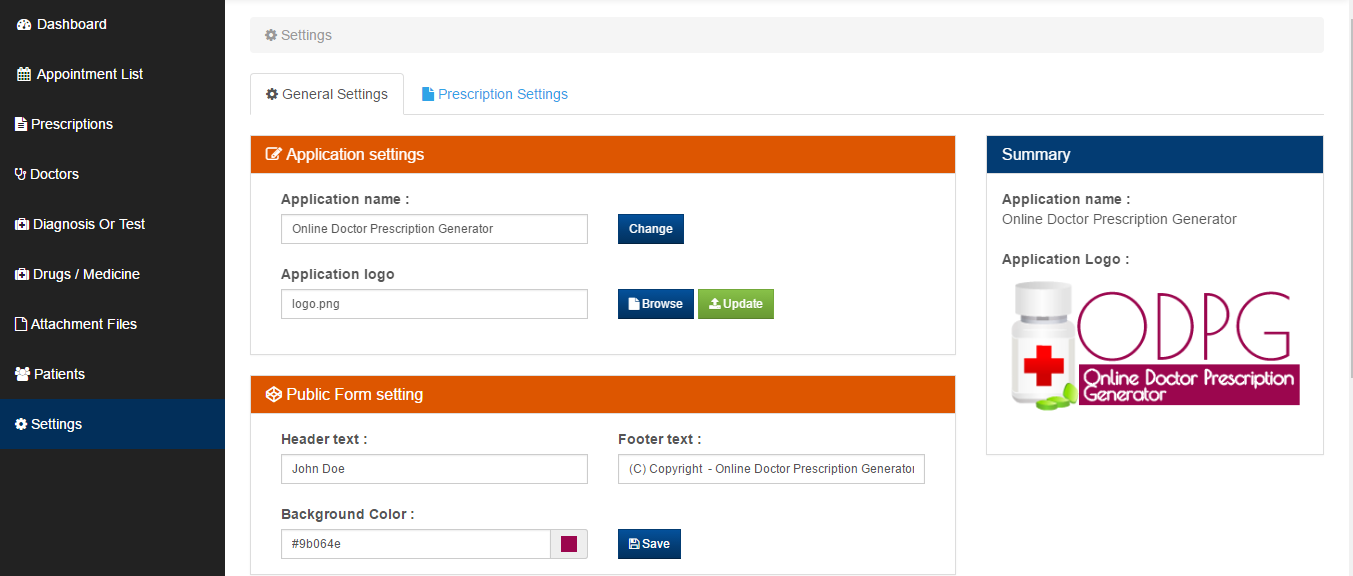


Figure 2.1 : Evanto Market’s Prescription

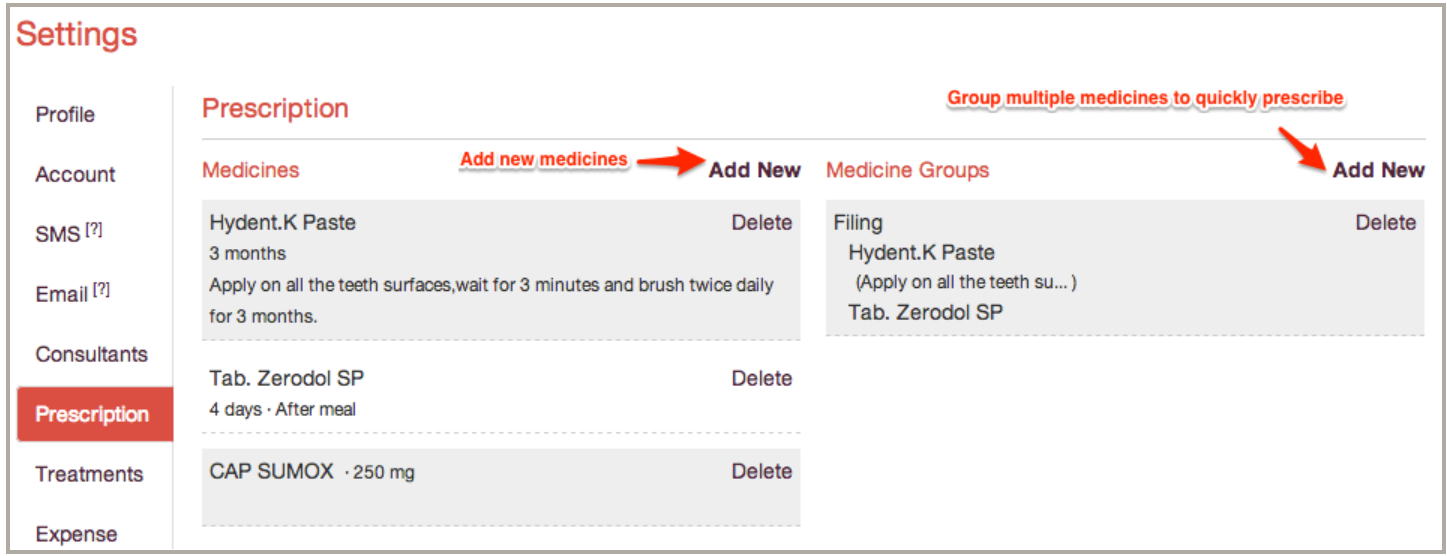


Figure 2.2 : Indian Hospitals Prescription

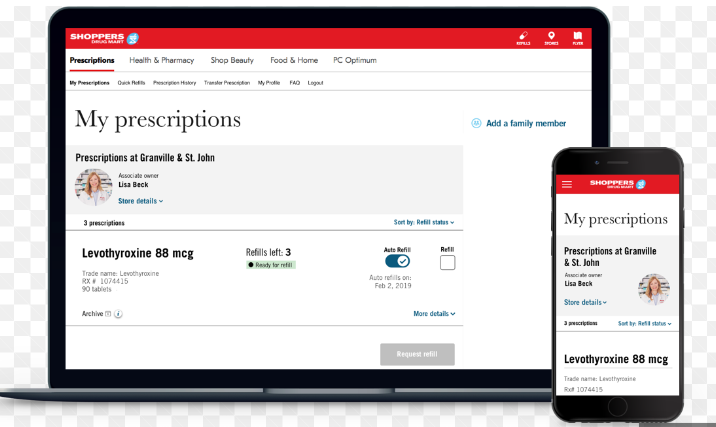


Figure 2.3 : Global Hospitals Web Prescription including app

**Comparison Table With Other System**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL | Feature Name | Description | Others | PMSCS |
| 1 | Home | This is Landing Page | Yes | Yes |
| 2 | Email Prescription | Prescription Soft Copy | No | Yes |
| 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 | No | Yes |
| 7 | Previous History | All previous history display | No | Yes |
| 8 | Admin panel login | Login admin panel | Yes | Yes |
| 9 | Doctor Dept. wise Config | Prescription interface as per doctor department | No | Yes |
| 10 | Patient Login | Patient login for download prescription | No | Yes |
| 11 | Other configuration | Medicine, advice, investigation etc config | Yes | Yes |
| 12 | Management Related all Report | Display all management related report | No | Yes |

**Chapter 3**

**System Analysis**

**3.1 Introduction**

Software systems analysis is a field in which analysts continually learn new tech- niques and approaches to properly capture, maintain, understand, and develop more efficient and effective software systems. So, there we will discuss how we analyze our system to achieve our goal. How we collect requirement, we analyze all those requirements and then finalize what actually should take as consideration to develop the system for a better experience.

**3.2 Software Requirement Specification (SRS)**

3.2.1System 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 User Characteristics

* The patient is expected to be Internet literate 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 Requirements**

The Logical Structure of the Data is contained in Section.

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

|  |  |
| --- | --- |
| **Trigger** | The admin selects to add new entities to the database. |
| **Precondition** | The admin has accessed the configuration screen. |
| **Basic Path** | 1. The system presents a blank grid to enter the required information. 2. The admin enters the information and submits the form. 3. The system checks that the required fields are not blank and updates the database. |
| **Alternative Paths** | 1. If in step 2, either field is blank, the Editor is instructed to add an entry. No validation for correctness is made. |
| **Post Condition** | The entity has been added to the database. |
| **Exception Paths** | The admin may abandon the operation at any time. |

Add Patient and Doctor

|  |  |
| --- | --- |
| **Trigger** | The admin selects to add new patient and doctor to the database. |
| **Precondition** | The admin has accessed the configuration screen. |
| **Basic Path** | 1 The system presents a blank grid to enter the required information.  2 The admin enters the information and submits the form.  3 The system checks that the required fields are not blank and updates the database. |
| **Alternative Paths** | 1. If in step 2, either field is blank, the Editor is instructed to add an entry. No validation for correctness is made. |
| **Post Condition** | The patient and doctor has been added to the database. |
| **Exception Paths** | The admin may abandon the operation at any time. |
| **Trigger** | The admin selects to add new entities to the database. |

Create Prescription

|  |  |
| --- | --- |
| **Trigger** | The doctor selects to create new prescription to the database. |
| **Precondition** | The doctor has accessed the configuration screen. |
| **Basic Path** | 1. The system presents a blank grid to enter the required information. 2. Doctor enters the information and submits the form 3. The system checks that the required fields are not blank and updates the database. |

Edit Prescription

|  |  |
| --- | --- |
| **Trigger** | The doctor selects to update existing prescription to the database. |
| **Precondition** | The doctor has accessed the configuration screen. |
| **Basic Path** | 1. The system presents a editable grid to update the required information. 2. Doctor edit the information and submits the form 3. The system checks that the required fields are not blank and updates the database. |

* + 1. **Non-Functional Requirements**

3.1 Logical Structure of the Data

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

Prescription

Doctor

writes

sent to

Patient

Figure 3.1 - Logical Structure of the Prescription Data

**Chapter 4**

**System Design**

**4.1 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.
  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.

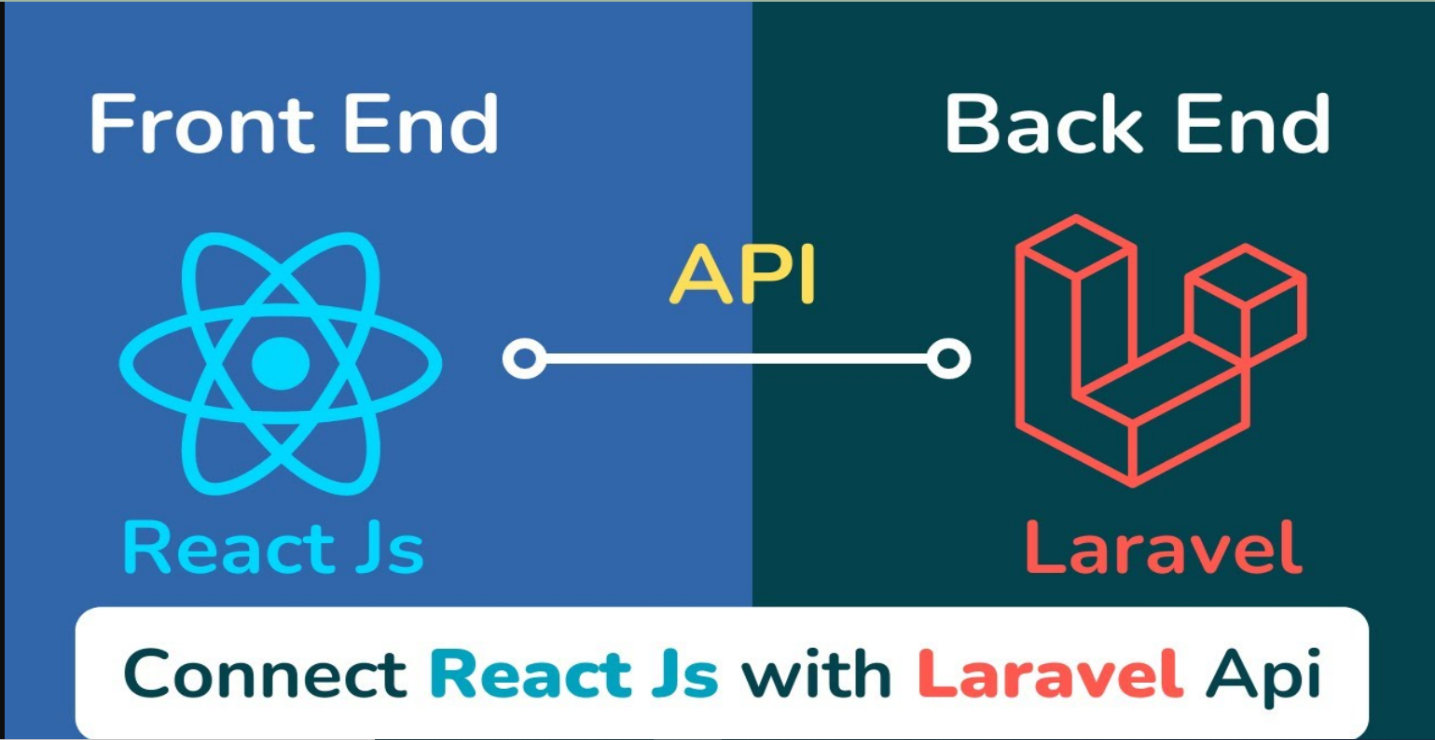


Figure 4.1: Technology used in our project

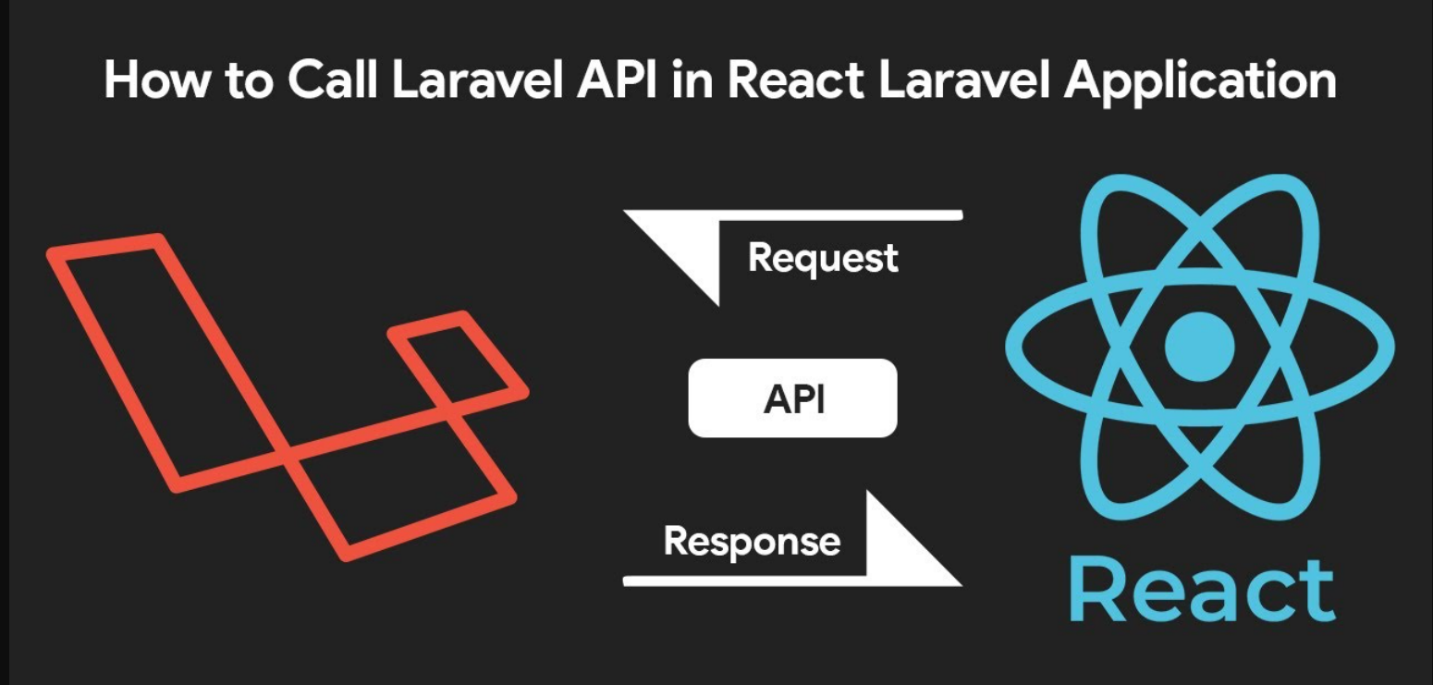


Figure 4.2: How Laravel API and React work

* 1. **Use Case Diagram**

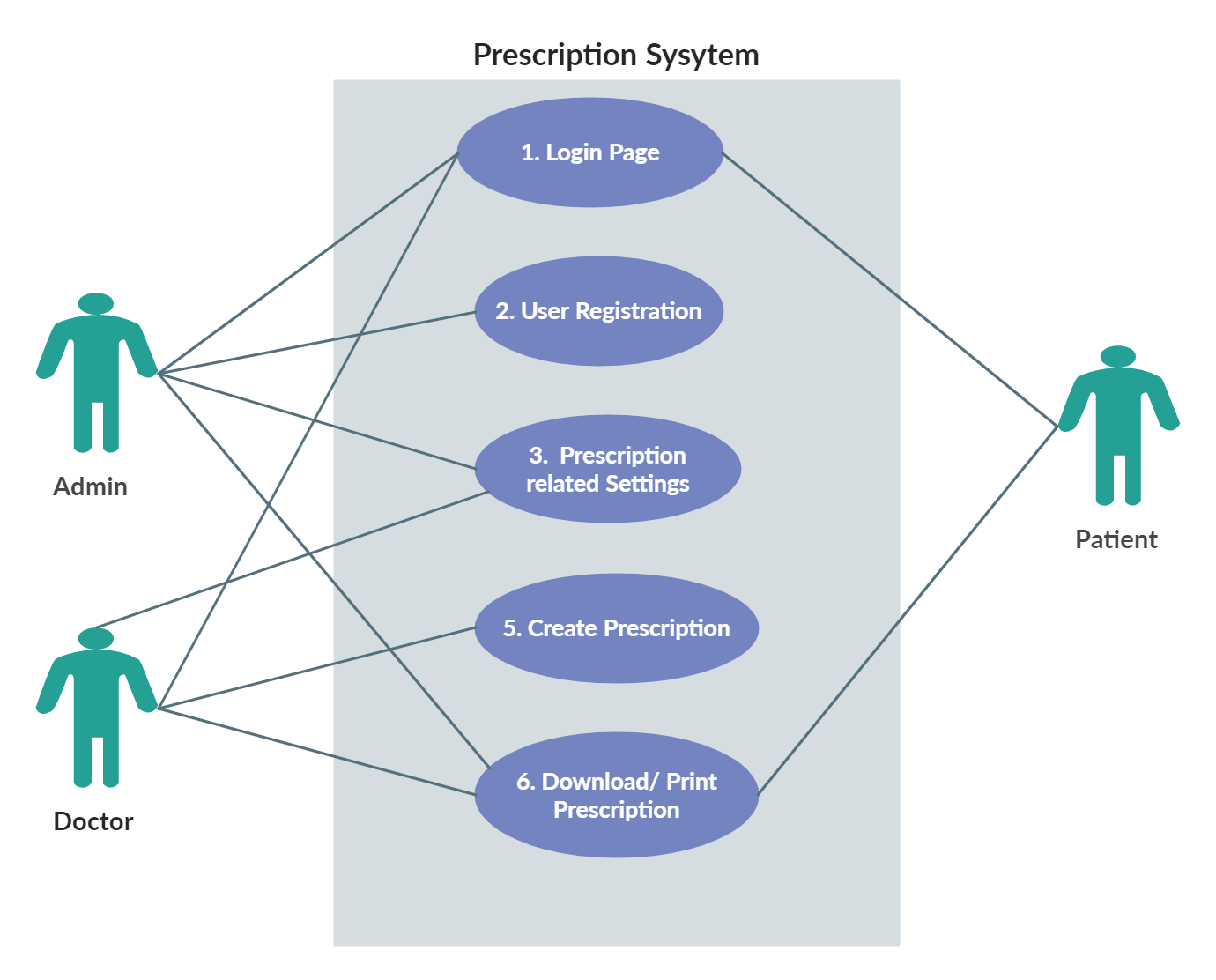


Figure 4.3 : Use Case diagram

* 1. Entity Relationship Diagram (ERD)

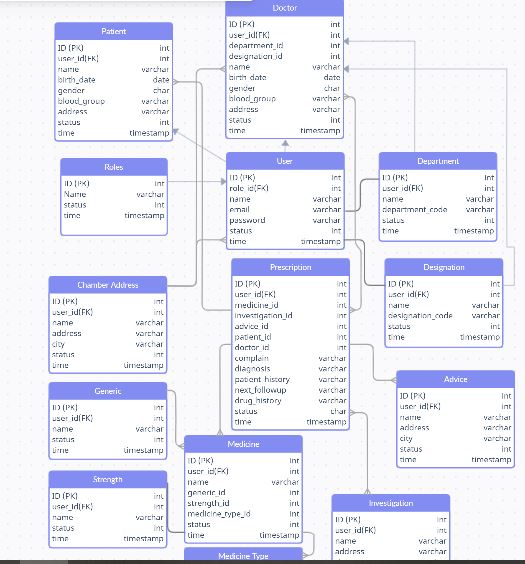


Figure 4.4 : ER Diagram

* 1. Database Screenshots

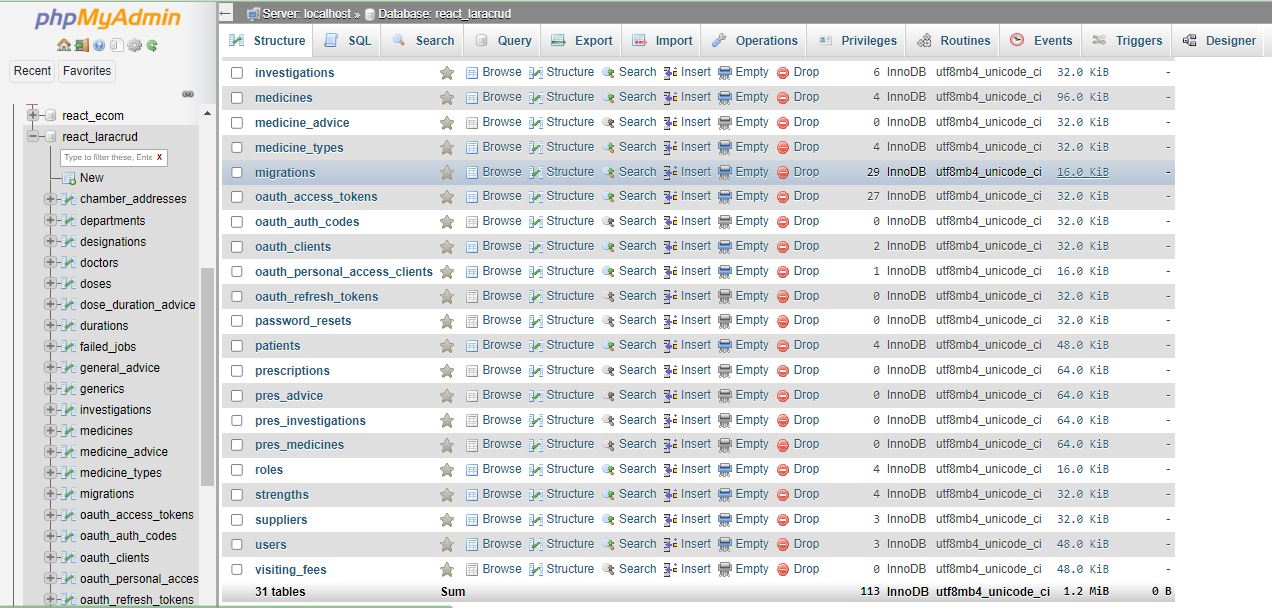


Figure 4.5 : Prescription Database Table

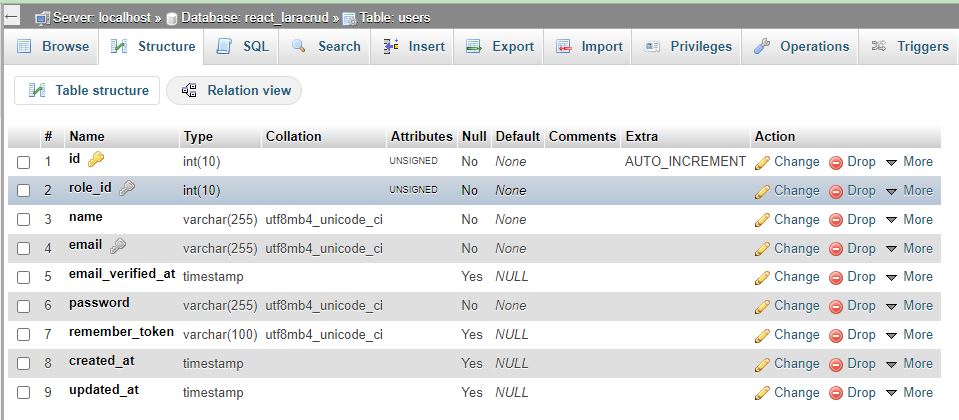


Figure 4.6 : Prescription User Table

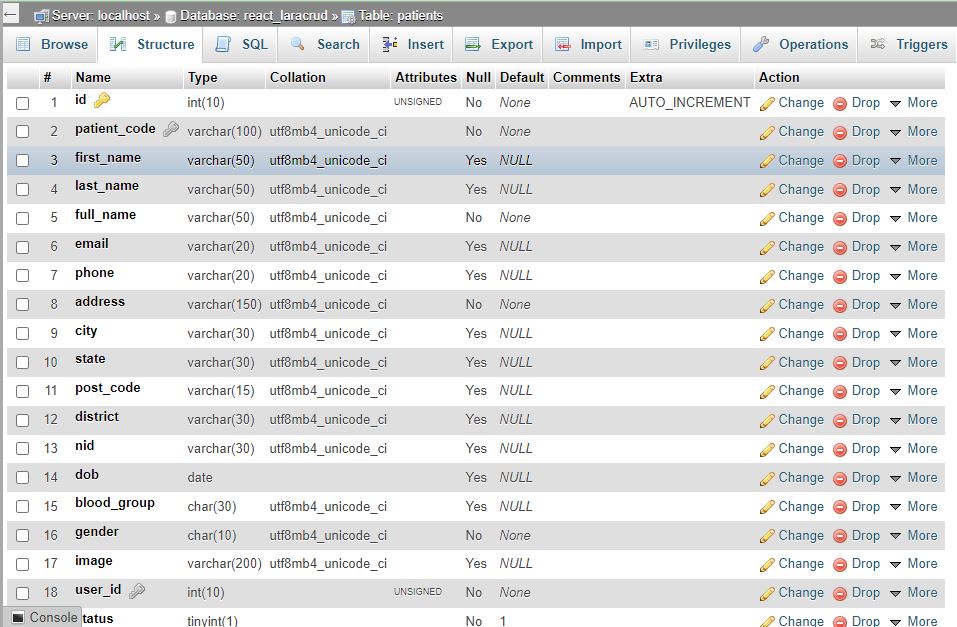
****

Figure 4.7 : Prescription Patient Table

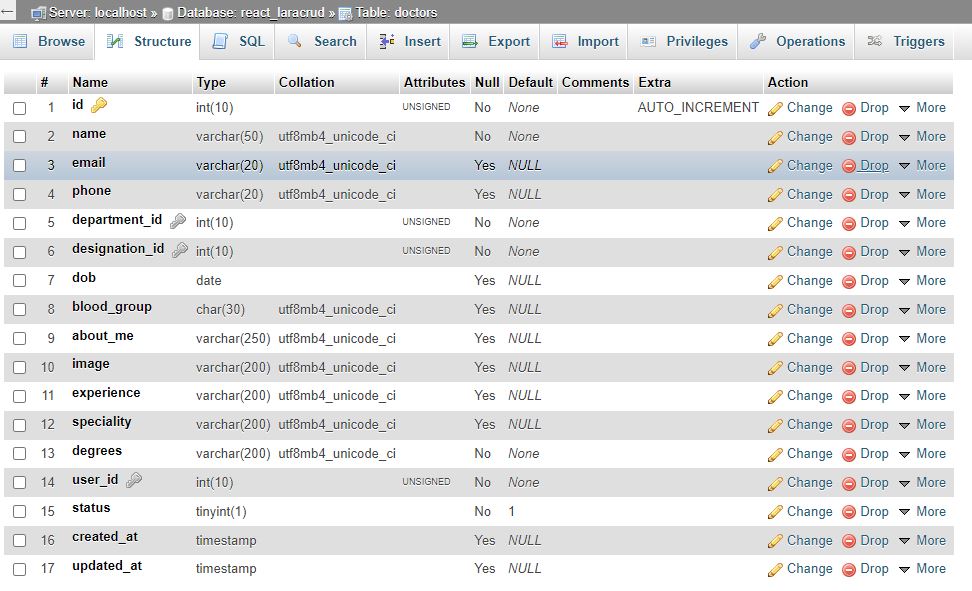


Figure 4.7: Prescription Doctor Table

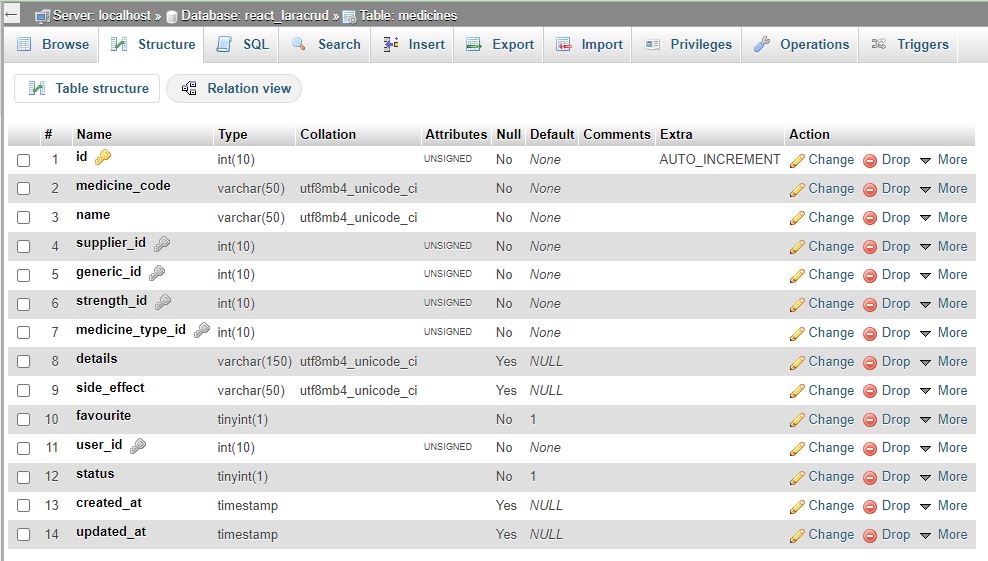


Figure 4.8 : Prescription Medicine Table

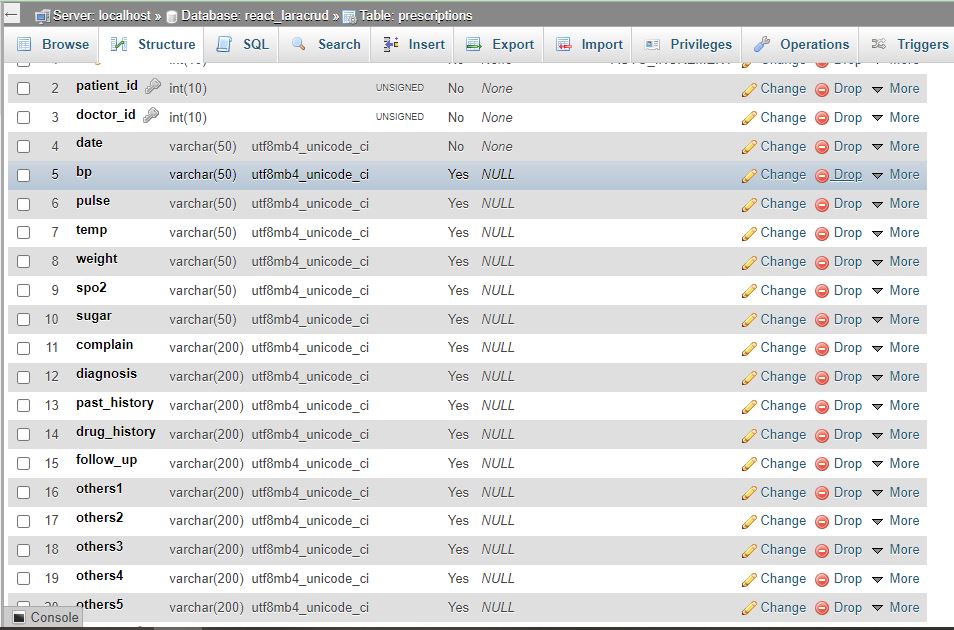


Figure 4.9 : Prescription Table

**Chapter 5**

**Implementation & Result**

5.1: GUI Design & Test Case

User interface design creates an effective communication medium between a human and a computer. The interface has to be right because it models a user’s perception of the software. As we know that a key tenet of all software engineering process models is “understand the problem before you attempt to design a solution”, we analysis the interface before starting the design steps

5.1.1 Prescription Dashboard

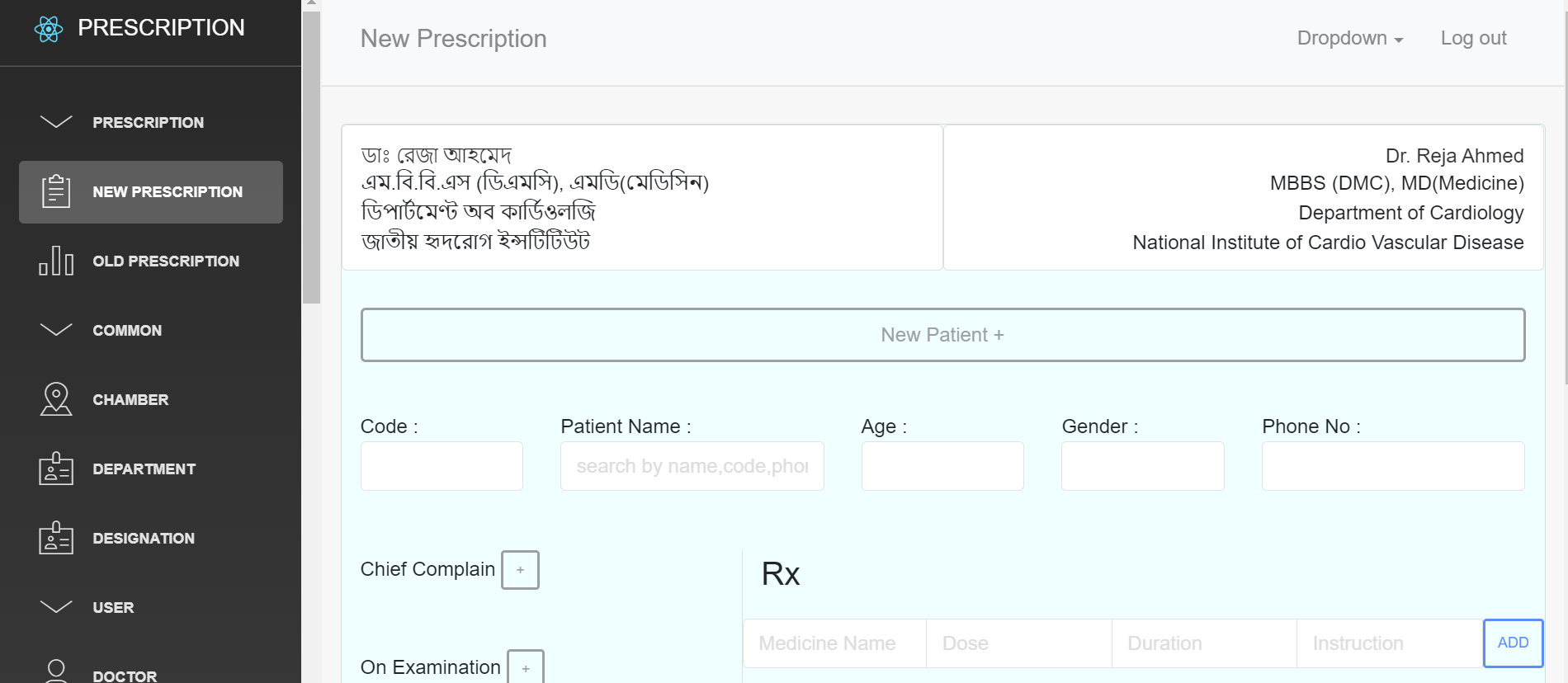
****

Figure 5.1: Dashboard design

* + 1. Sidebar Menu

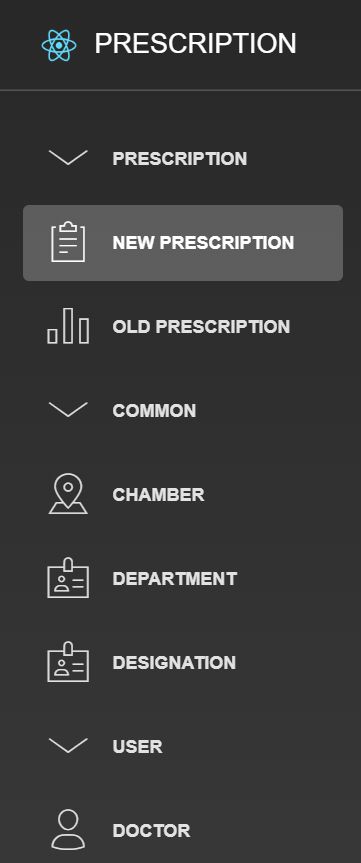
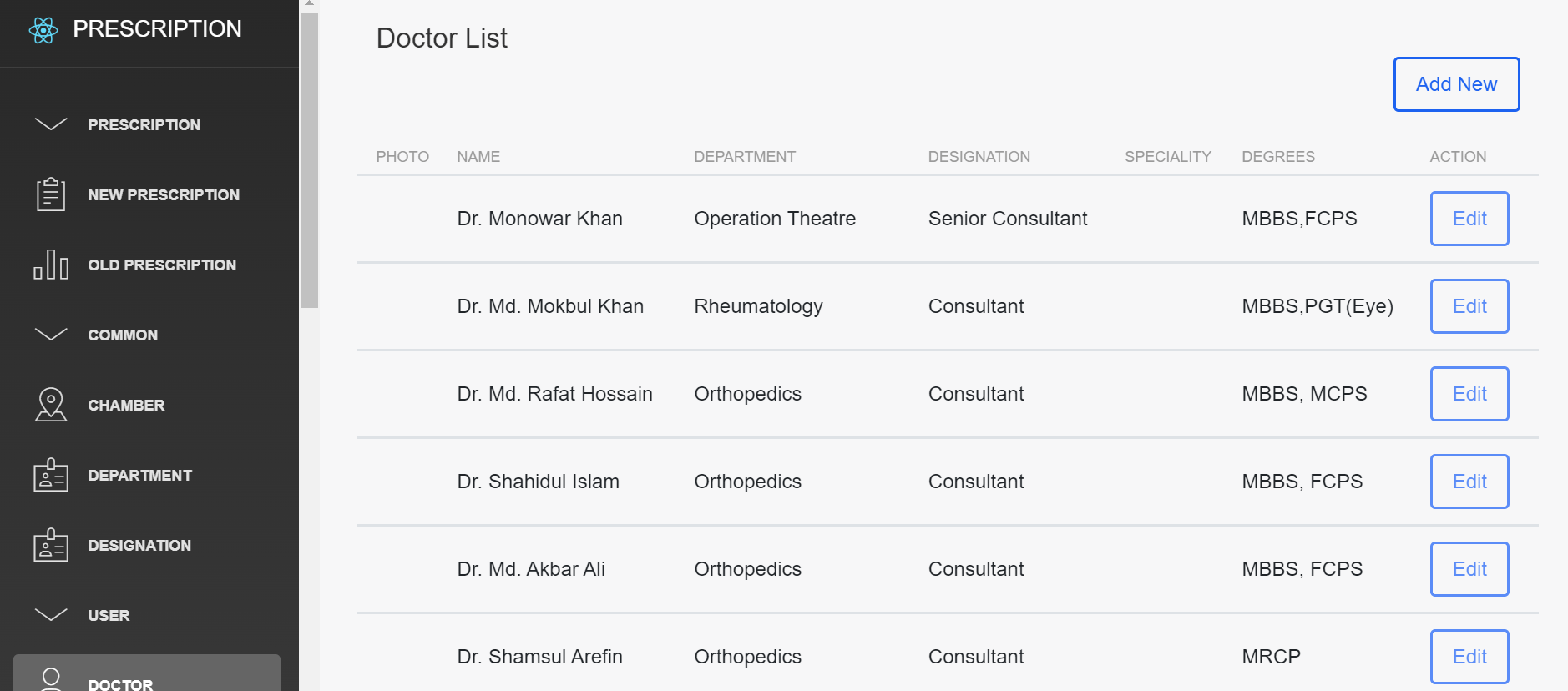


Figure 5.2: sidebar menu page design

5.1.3 Add Doctor Page



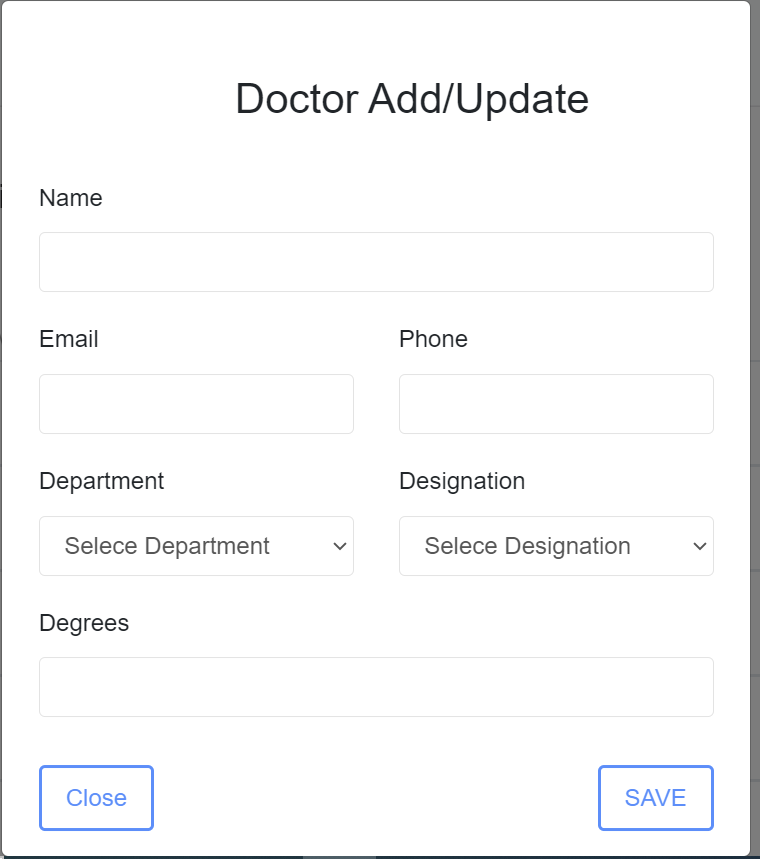


Figure 5.3: Doctor add page design

5.1.4 Add Patient Page:

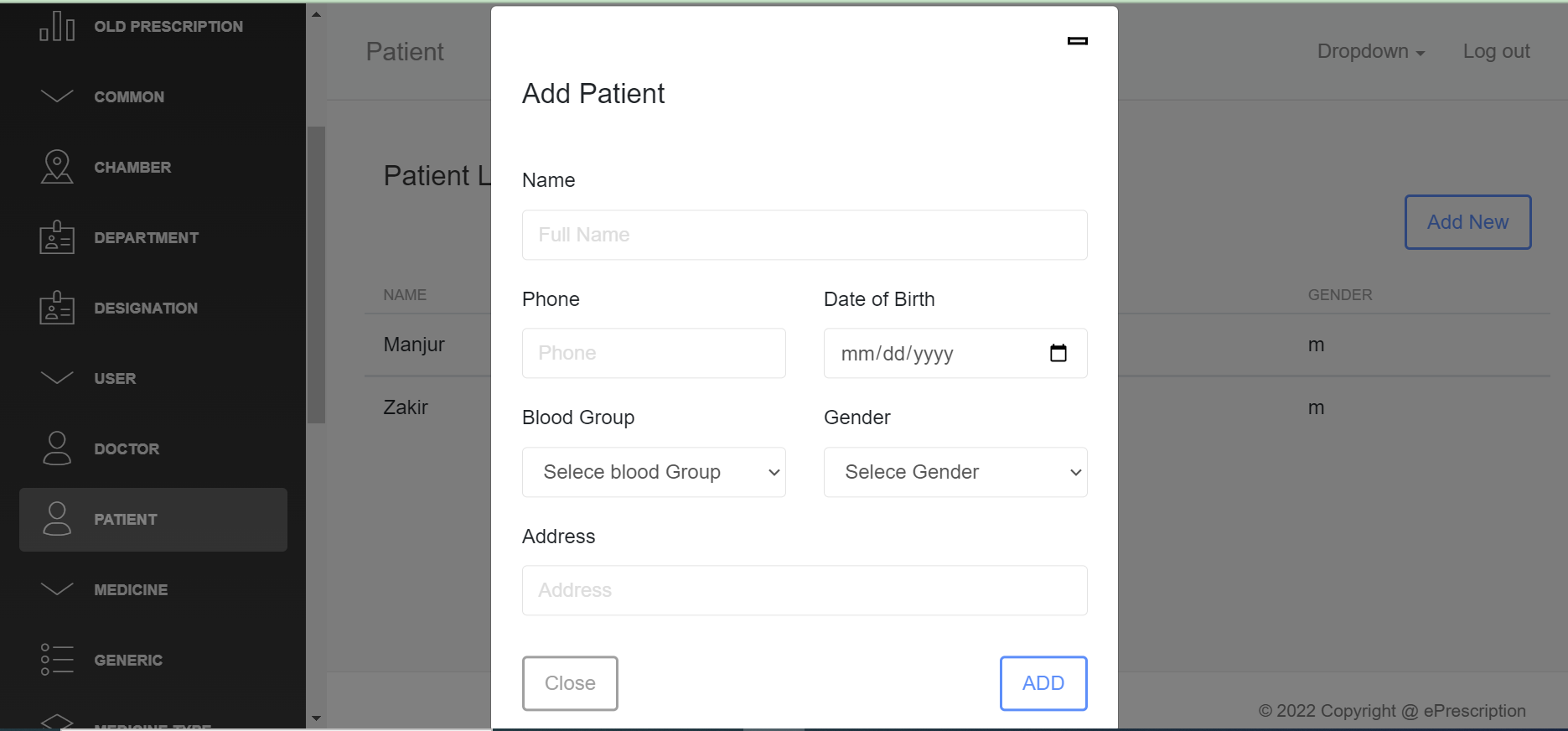


Figure 5.4: Patient add page design

5.1.5 Add Medicine:

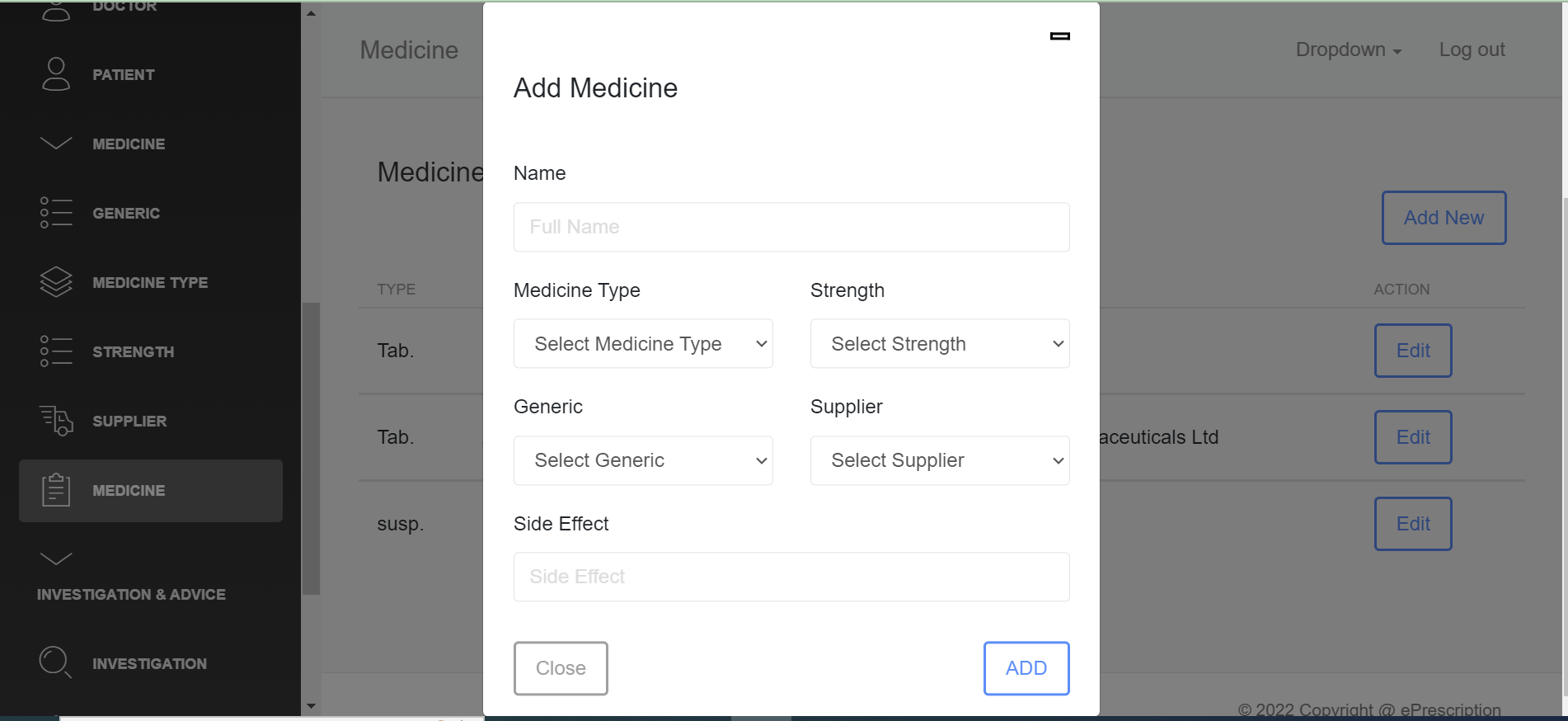


Figure 5.5: Medicine Add page design

5.1.6 Add Investigation Page:

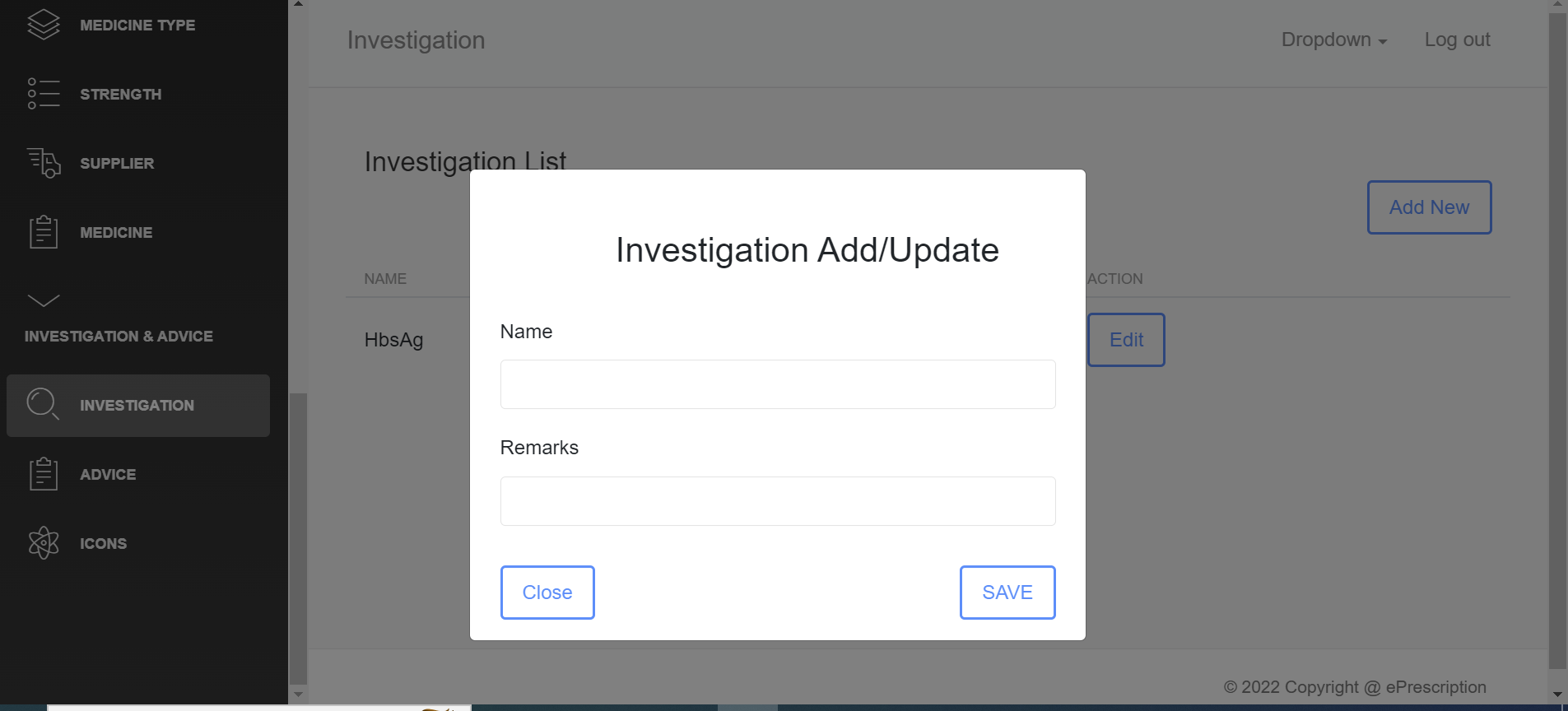


Figure 5.6: Investigation insert page design

5.1.7 Prescription Page:

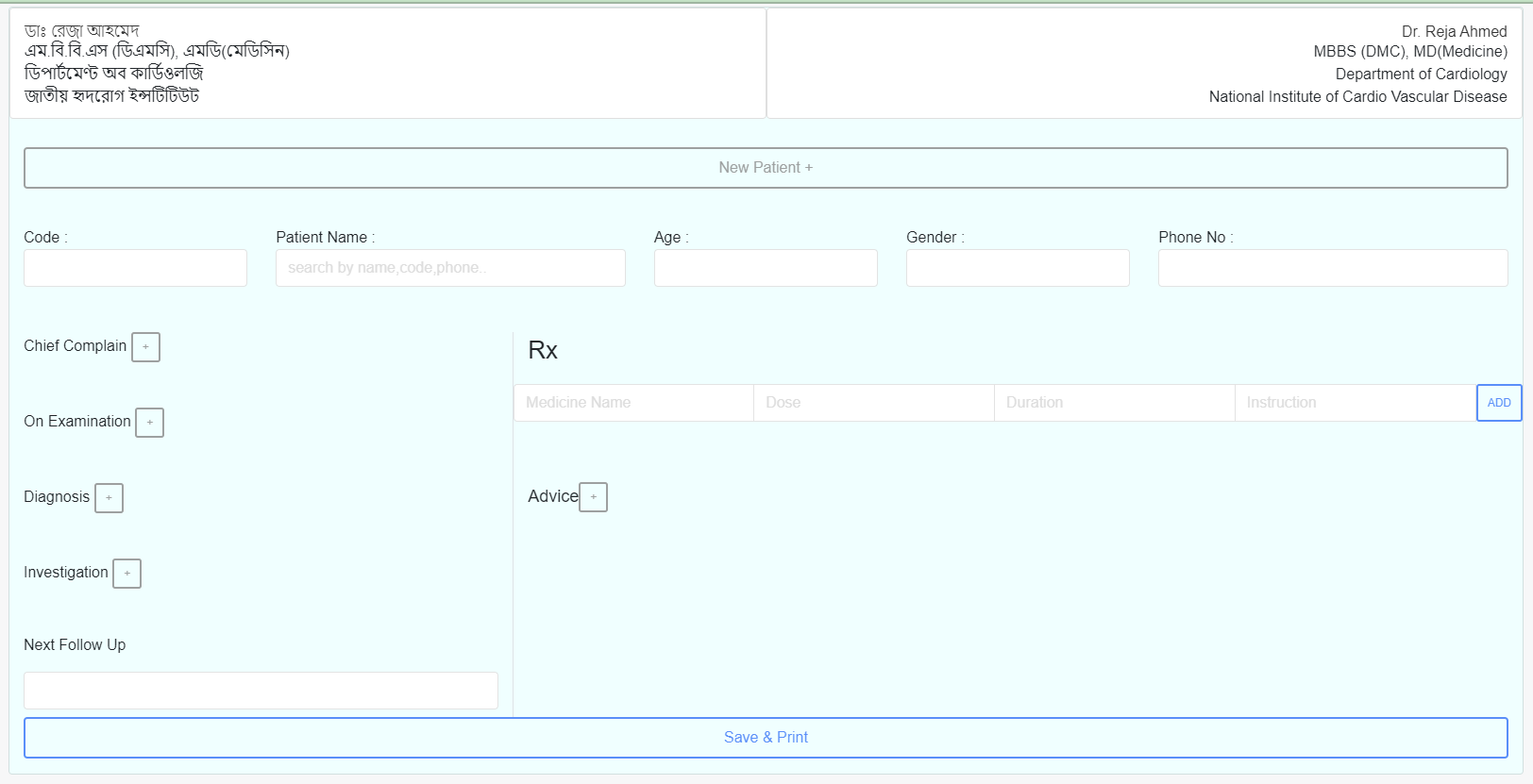


Figure 5.7: Prescription page design

* 1. **Testing**

**5.2.1 Overview**

Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements.

**5.2.2 Scope**

Testing depends on the source code but reviewing requirements and developing test cases is independent from the developed code. Finding bugs in the Software is the task of testers. Developers are only responsible for the specific component or area that is assigned to them but testers understand the overall workings of the software, what the dependencies are and what the impacts of one module on another module are. This section of the document describes the overall testing activities with test case.

**5.3.3 Appendix:**

**White box testing:**

5.3.1 **user login system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null, Null | Login | Message display **Error** | Pass |
| 02. | Null, 123456789 | Login | Message display **Error** | Pass |
| 03. | 123456789, Null | Login | Message display **Error** | Pass |
| 04. | 123456789, admin | Login | Message display **Error** | Pass |
| 05. | admin,Null | Login | Message display **Error** | Pass |
| 06. | Null, admin | Login | Message display **Error** | Pass |
| 07. | 123456789, 123456789 | Login | Message display **Error** | Pass |
| 08. | @$%#\*&!~(){}[], admin | Login | Message display **Error** | Pass |
| 09. | @$%#\*&!~(){}[], @$%#\*&!~(){}[] | Login | Message display **Error** | Pass |
| 10. | admin, @$%#\*&!~(){}[] | Login | Message display **Error** | Pass |
| 11. | @$%#\*&!~(){}[], 123456789 | Login | Message display **Error** | Pass |
| 12. | 123456789, @$%#\*&!~(){}[] | Login | Message display **Error** | Pass |
| 13. | admin, @$%#\*&!~(){}[] | Login | Message display **Error** | Pass |
| 14. | admin, 123456789 | Login | Message display **Success** | Pass |
| 15. | admin, admin123456789 | Login | Message display **Success** | Pass |
| 16. | admin, 123456789admin | Login | Message display **Success** | Pass |
| 17. | ADMIN, 123456789 | Login | Message display **Success** | Pass |
| 18. | ADMIN, admin123456789 | Login | Message display **Success** | Pass |
| 19. | ADMIN, 123456789admin | Login | Message display **Success** | Pass |
| 20. | ADMIN, ADMIN | Login | Message display **Success** | Pass |
| 21. | admin, ADMIN 123456789 | Login | Message display **Success** | Pass |
| 22. | admin, 123456789ADMIN | Login | Message display **Success** | Pass |
| 23. | ADmin, AdminN | Login | Message display **Success** | Pass |

Table 4: white box testing of "**login system**"

5.3.2 **Configiration insert system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null, Config name | add | Message display **Error** | Pass |
| 02. | Null, Null, Null, captcha | add | Message display **Error** | Pass |
| 03. | Null, Null, Config name cd, | add | Message display **Error at least 3 character** | Pass |
|  |  |  |  |  |

5.3.3 **Doctor insert system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null, Doctor name | add | Message display **Error** | Pass |
| 02. | Null Doctor | add | Message display **Error** | Pass |
| 03. | Doctor name 3 character | add | Message display **Error at least 3 character** | Pass |

5.3.4 **patient insert system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null, patient | add | Message display **Error** | Pass |
| 02. | Null city | add | Message display **Error** | Pass |
| 03. | city and patient name less than 3 character | add | Message display **Error at least 3 character** | Pass |

5.3.5 **Insert prescription system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null doctor,patient | create | Message display **Error** | Pass |
| 02. | Null, Medicine | create | Message display **Error** | Pass |
| 03. | Null 1st page | create | Message display **Error** | Pass |
| 04. | Null O/E | create | Message display **Error** | Pass |

5.3.6 **user sign up:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No:** | **Inputs:** | **Click Button:** | **Expected result:** | **Pass/fail** |
| 01. | Null name | Submit | Message display **Error** | Pass |
| 02. | Null, Email | Submit | Message display **Error** | Pass |
| 03. | Null password or password at least 6 character | Submit | Message display **Error** | Pass |
| 04. | Null address | Submit | Message display **Error** | Pass |
| 05. | Null country select | Submit | Message display **Error** | Pass |
| 07. | Null Gender | Submit | Message display **Error** | Pass |
| 08. | Null contact | Submit | Message display **Error** | Pass |

* 1. **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\GeneralAdviseController;

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',[GeneralAdviseController::class, 'allGeneralAdvice']);

Route::post('/addgeneralAdvice',[GeneralAdviseController::class, 'addGeneralAdvice']);

Route::post('/updateadvice/{id}',[GeneralAdviseController::class, 'updateAdvice']);

Route::get('/searchadvices/{key}',[GeneralAdviseController::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']);

* + 1. **React Code**

1. import React, { Component } from 'react'
2. import { BrowserRouter, Route, Switch, Redirect } from "react-router-dom";
3. import AppURL from 'api/AppURL';
4. import AdminLayout from "layouts/Admin.js";
5. import Login from "views/auth/Login";
6. import axios from 'axios';
7. import { useEffect , useState } from 'react';
8. export default *function* App() {
9. *const* [user,setUser] = useState([]);
10. useEffect(() *=>* {
11. axios.get(AppURL.UserData).then(*response* *=>* {
12. setUser(response.data);
13. }).catch(*error* *=>* {
14. });
15. }, [])
16. return (
17. <>
18. <*BrowserRouter*>
19. <*Switch*>
20. <*Route* exact path="/" render={(*props*) *=>* <*Login* user={user} setUser={setUser} />} />
21. <*Route* exact path="/login" render={(*props*) *=>* <*Login* user={user} setUser={setUser} />} />
22. <*Route* path="/admin" render={(*props*) *=>* <*AdminLayout* user={user} setUser={setUser} {...props} />} />
23. </*Switch*>
24. </*BrowserRouter*>
25. </>
26. )
27. }

**5.4 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.

**Chapter 6**

**Conclusion and Future Works**

**6.1** **Conclusion**

I am pleased to submit the final Software documentation report on Online prescription. From this, the patients will get a clear and easy view of medicine names. To improve prescriptions efficiency, prescription is very essential. An online prescription system is more effective than paper based manual system. This document can be used effectively to maintain software development cycle. It will be very easy to conduct the whole project using it. Hopefully, this document can also help our junior BSSE batch students. We tried our best to remove all dependencies and make effective and fully designed document. We believe that reader will find it in order.

**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. For creating usecase and ER diagram <https://app.creately.com/d/rWNGg4JUeCt/edit>
2. Laravel Related Documentation <https://laravel.com/docs/8.x>
3. React related documentation <https://reactjs.org/>
4. Refil Prescription systems **bit.ly/3GhGkDG**
5. Pro Clinic Prescription <https://www.konnectplugins.com/proclinic/Vertical/index.html>
6. Zilsoft Prescription <https://zilsoft.net/about/>
7. Amar chamber website landing page <https://amarchamber.com/>
8. For Multiple Prescription software https://www.softwareadvice.com/medical/e-prescribing-comparison/