**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: 27th

**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](https://juniv.edu/teachers/sarnali.cse)

Second Examiner Associate Professor

Department of Computer Science and Engineering

Jahangirnagar University

Savar, Dhaka-1342, Bangladesh

Accepted by:

………………………………………………………………………………

###### Professor [Dr. Md. Humayun Kabir](https://juniv.edu/teachers/hkabir)

Coordinator

PMSCS Program 2022-2023

Department of Computer Science and Engineering

Jahangirnagar University

# Savar, Dhaka-1342, Bangladesh

**Declaration**:

I, hereby, 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 : 27th

**Acknowledgment**:

I would like to give special thanks to the following people who have helped me a lot for completing 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 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 unyielding knowledge on anything computer or math 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**

**Chapter 1 Introduction Page No**

* 1. Motivation 8
  2. Objectives 8
  3. Scope of Project 8
  4. Problem Definition 9
  5. Research Methodology 9
  6. Outline 9

**Chapter 2 Background and Related Works**

2.1 Introduction 10

2.2 Background and History 10

2.3 Related Work and Gap Analyze 10

2.4 Gap Comparison Table 12

**Chapter 3 System Analysis**

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

**Chapter 4 System Design**

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

**Chapter 5 Implementation and Results**

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

**Chapter 6 Conclusion and Future Works**

5.1 Conclusion 37

5.2 Future Works 37

**References** 38-39

**List of Figures**

**Page No.**

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 4.3Use 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. **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 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 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.

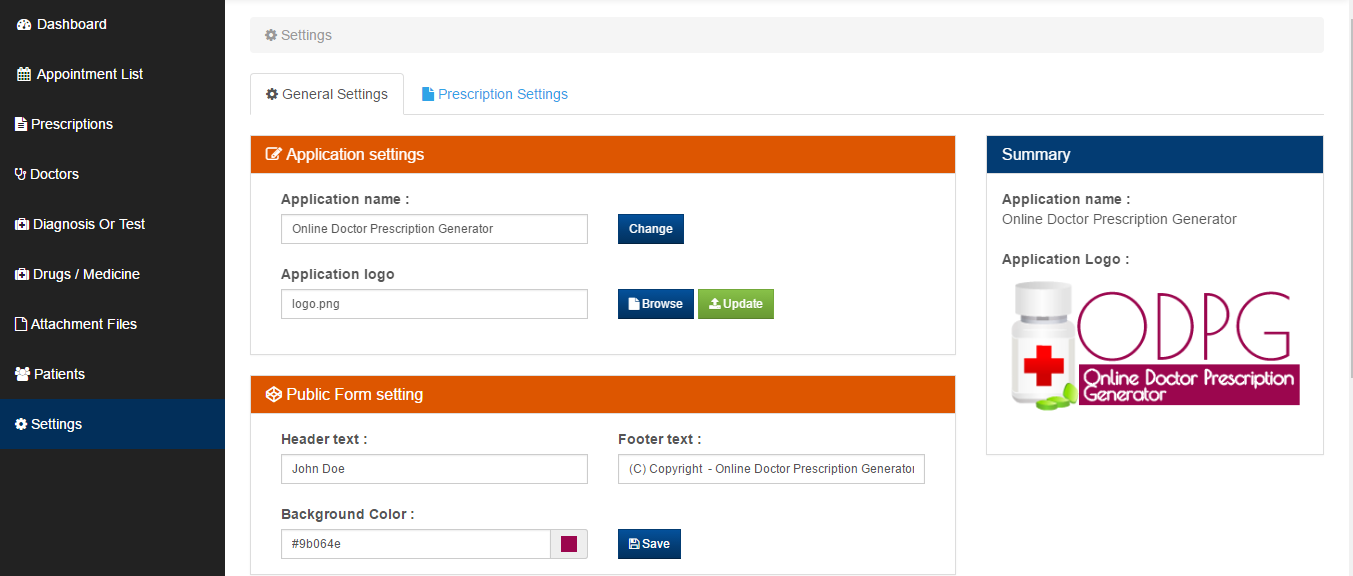


Figure 2.1 : Evanto Market’s Prescription Interface

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

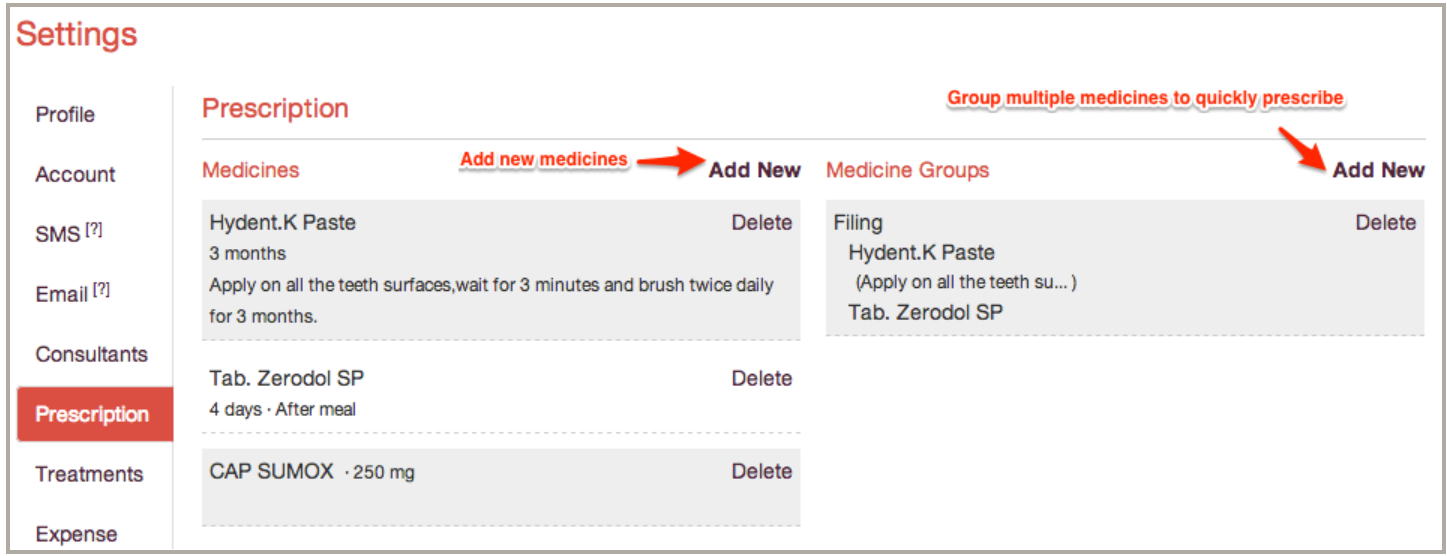


Figure 2.2 : Indian Hospitals Prescription Interface

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

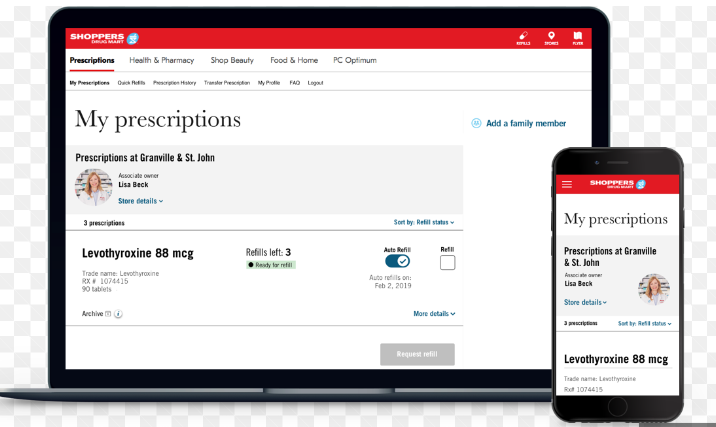


Figure 2.3: Global Hospitals Web Prescription including app

**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 | 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 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)**

3.2.1System 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 Users Characteristic

* The patient is suppoosed 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.

|  |  |
| --- | --- |
| **Trigger** | Admin selects to add/edit new entities to the database. |
| **Pre-condition** | The admin already access the configuration page. |
| **General Path** | 1.The software shows a blank div to enter the required information.  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. |
| **Alternative Paths** | In the second step , either the field is blank, the Editor is instructed to add an entry.  Validation for correctness is not 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. |
| **Pre-condition** | The admin has accessed the configuration screen. |
| **General Path** | 1.The software shows a blank div to enter the required information.  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. |
| **Alternative Paths** | In the second step , either the field is blank, the Editor is instructed to add an entry.  Validation for correctness is not 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. |
| **Pre-condition** | The doctor has accessed the configuration screen. |
| **General Path** | 1. 1.The software shows a blank div to enter the required information. 2. 2.The admin enters the information and submit the form. 3. 3.Software check that the required fields are blank or not and updates the database. |

Edit Prescription

|  |  |
| --- | --- |
| **Trigger** | The doctor selects to update existing prescription to the database. |
| **Pre-condition** | The doctor has accessed the configuration screen. |
| **General Path** | 1. 1.The software shows a blank div to enter the required information. 2. 2.The admin enters the information and submit the form. 3. 3.Software check that the required fields are blank or not 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 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.

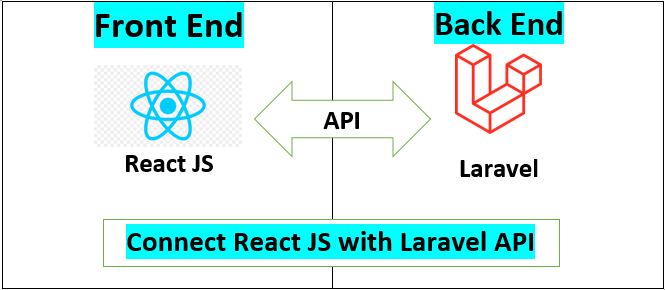


Figure 4.1: Showing How to connect React JS with Laravel API

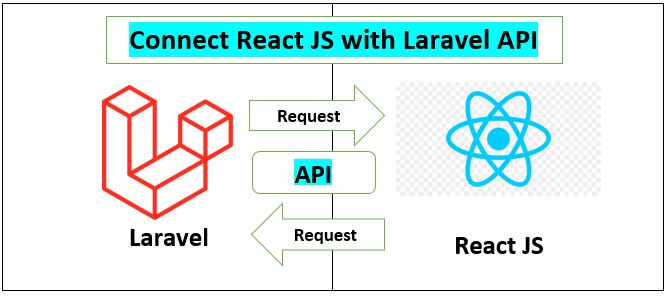


Figure 4.2: How Laravel API and React Request Response

* 1. **Use Case Diagram**

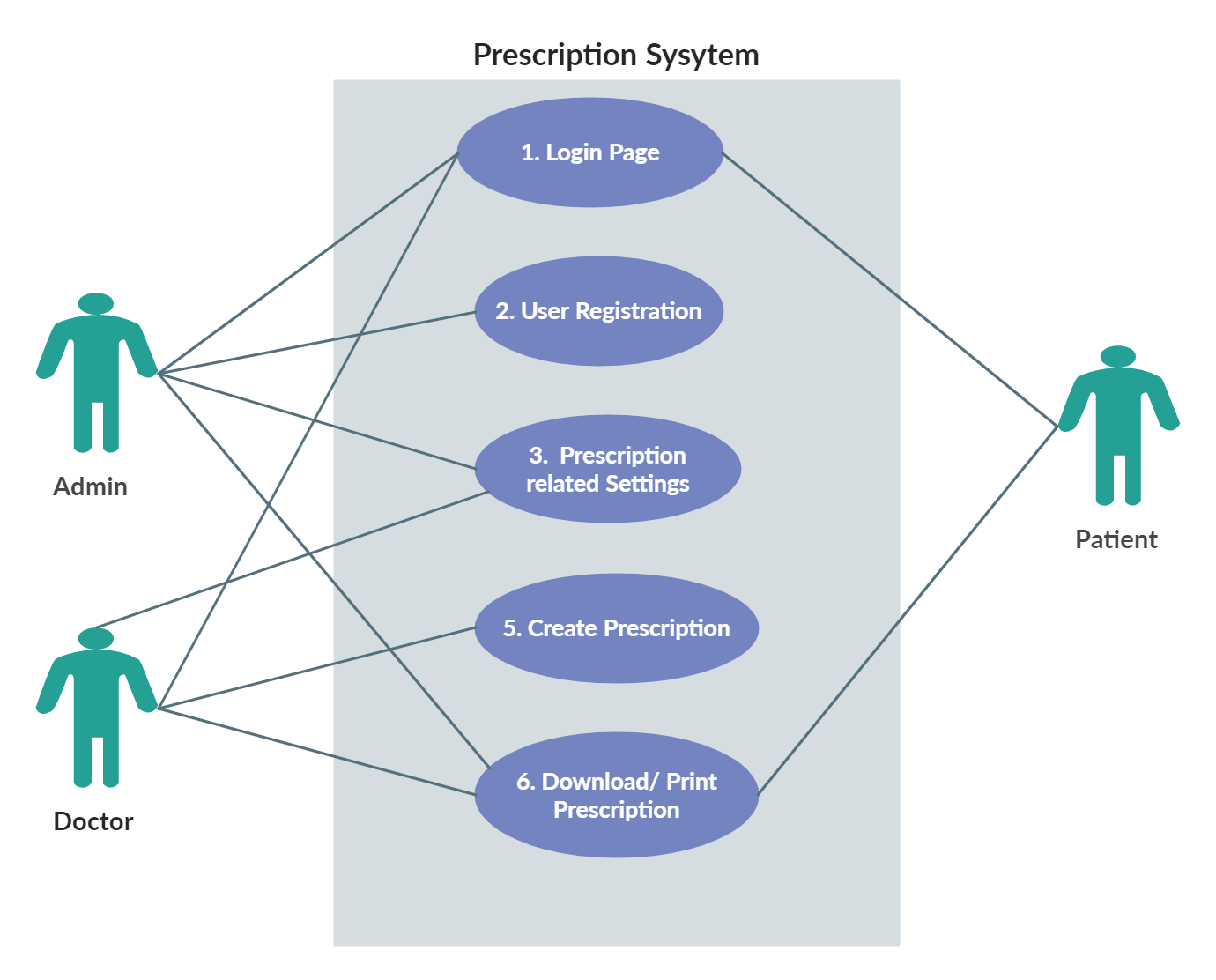


Figure 4.3: Use Case diagram

* 1. **Use Case Description**:
     1. **Login Page**: Every user have to log in to the system first.
     2. **User Registration**: Only Admin can register other admin, doctors and patient. After that they can login to their portal.
     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. **Create prescription**: Only doctors can use this option. They can create, edit or modify the previous prescription.
     5. **Download/Print Prescription**: All the user of this system can see the prescription and print the prescription copy.
  2. Entity Relationship Diagram (ERD)

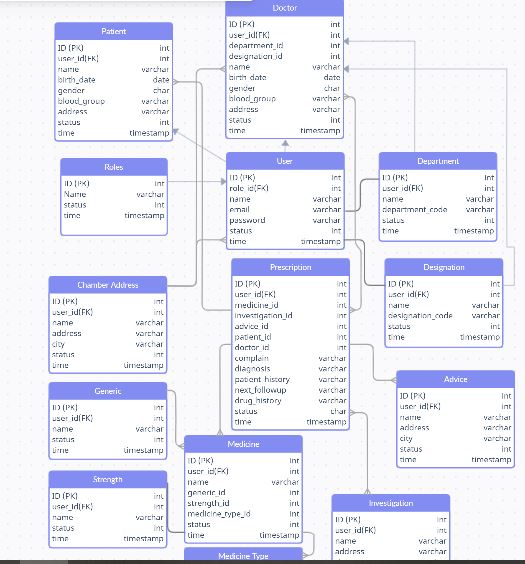


Figure 4.4: ER Diagram

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

* 1. Database Screenshots

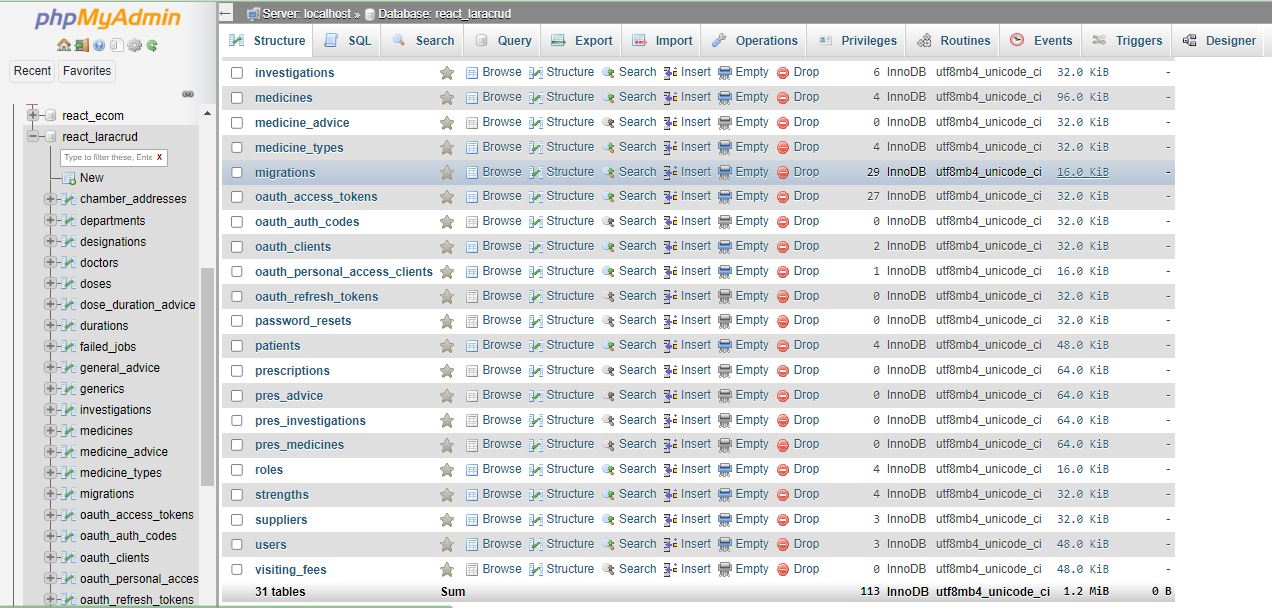


Figure 4.5: Prescription Database Table

**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.

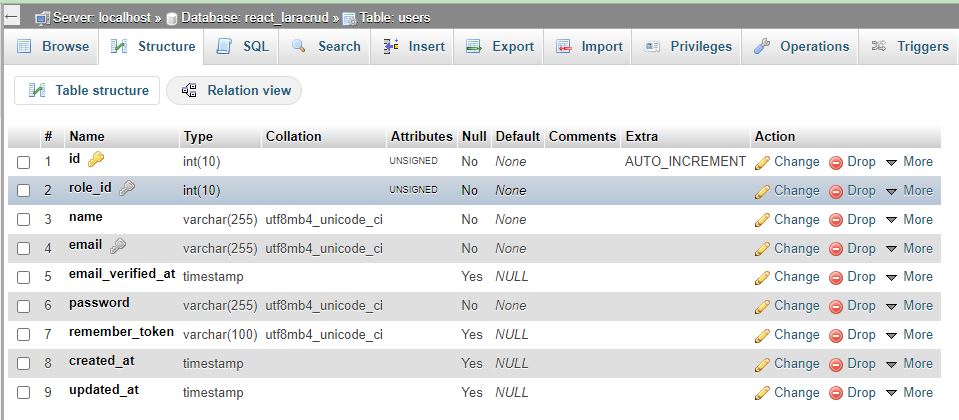


Figure 4.6: Prescription User Table

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

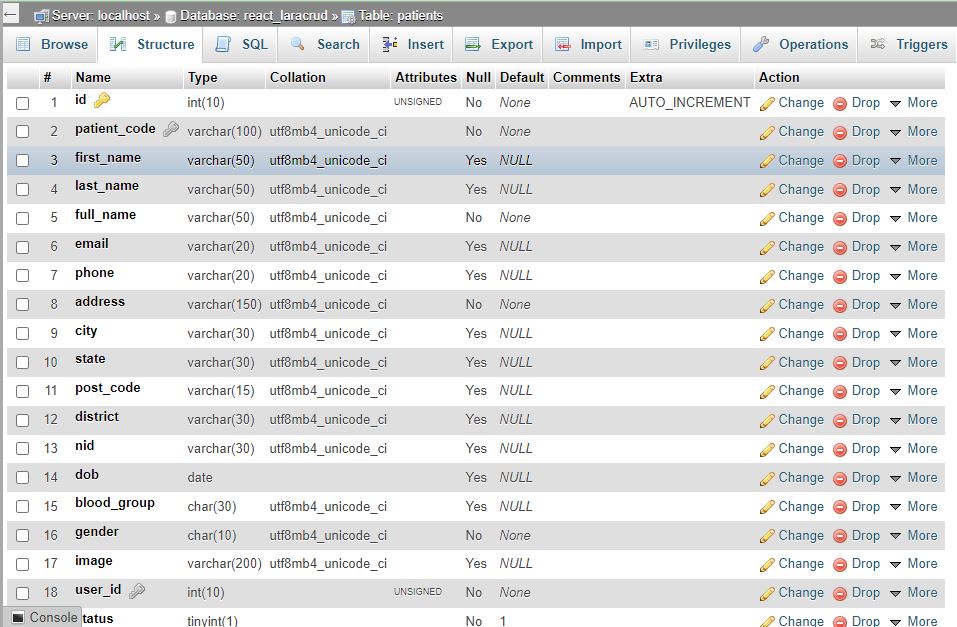
****

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.

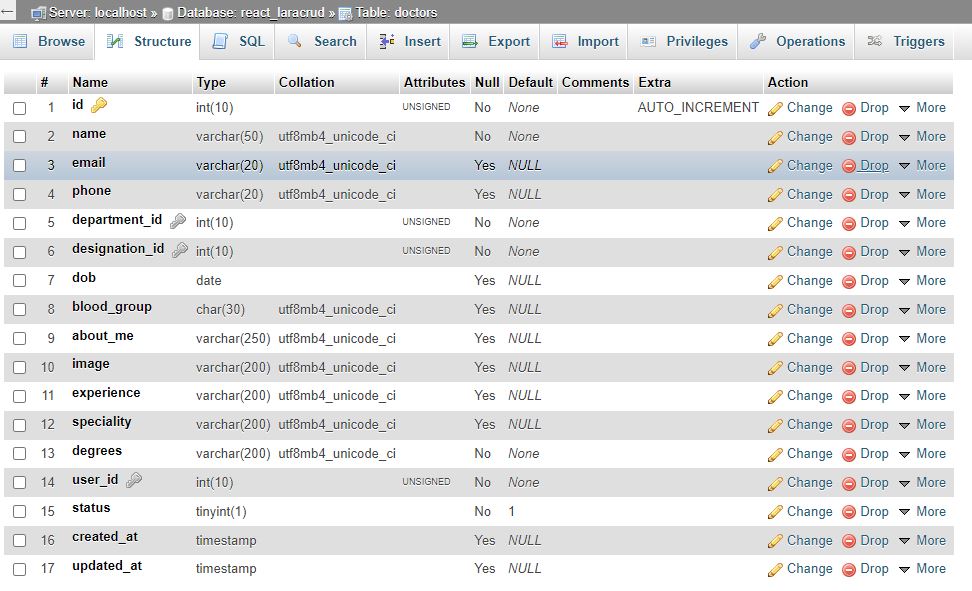


Figure 4.8: Prescription Doctor Table

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

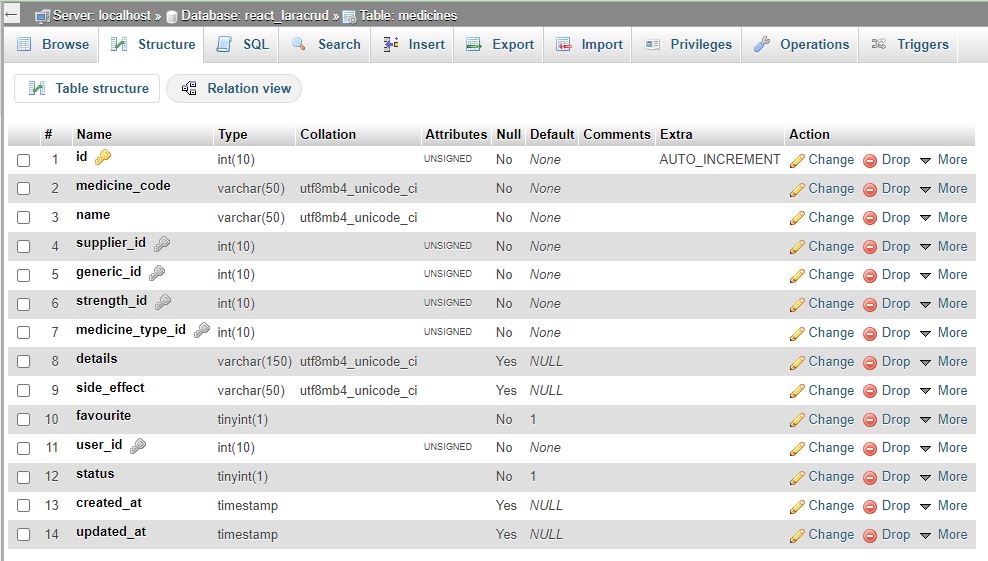


Figure 4.9: Prescription Medicine Table

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

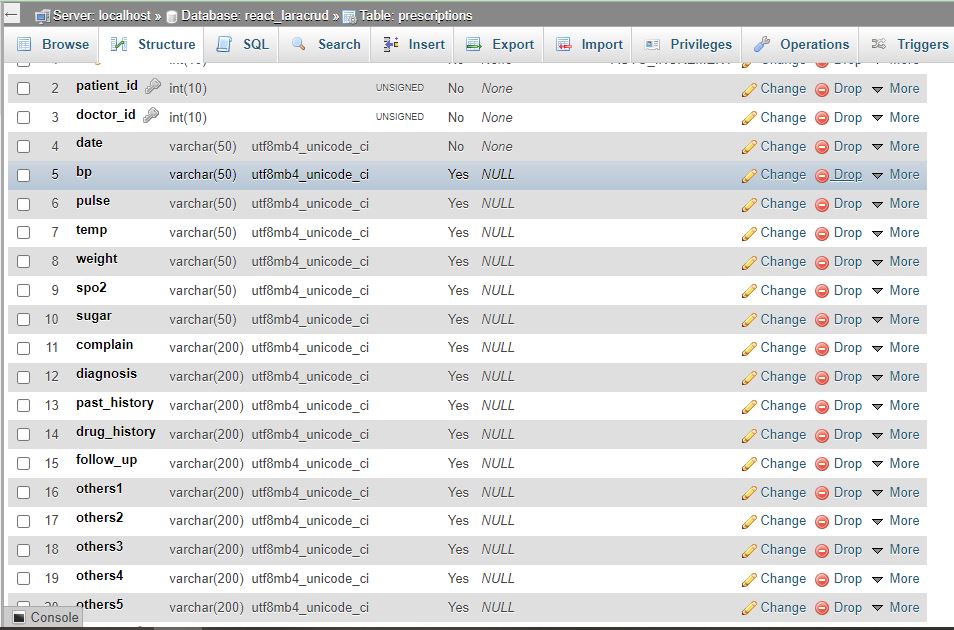


Figure 4.10: Prescription Table

**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 effective communication way between a user and a system. As people will the interface so it has to be clear and complete because it depict a users 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 before start to design the system.

5.3.1 **Prescription Dashboard**

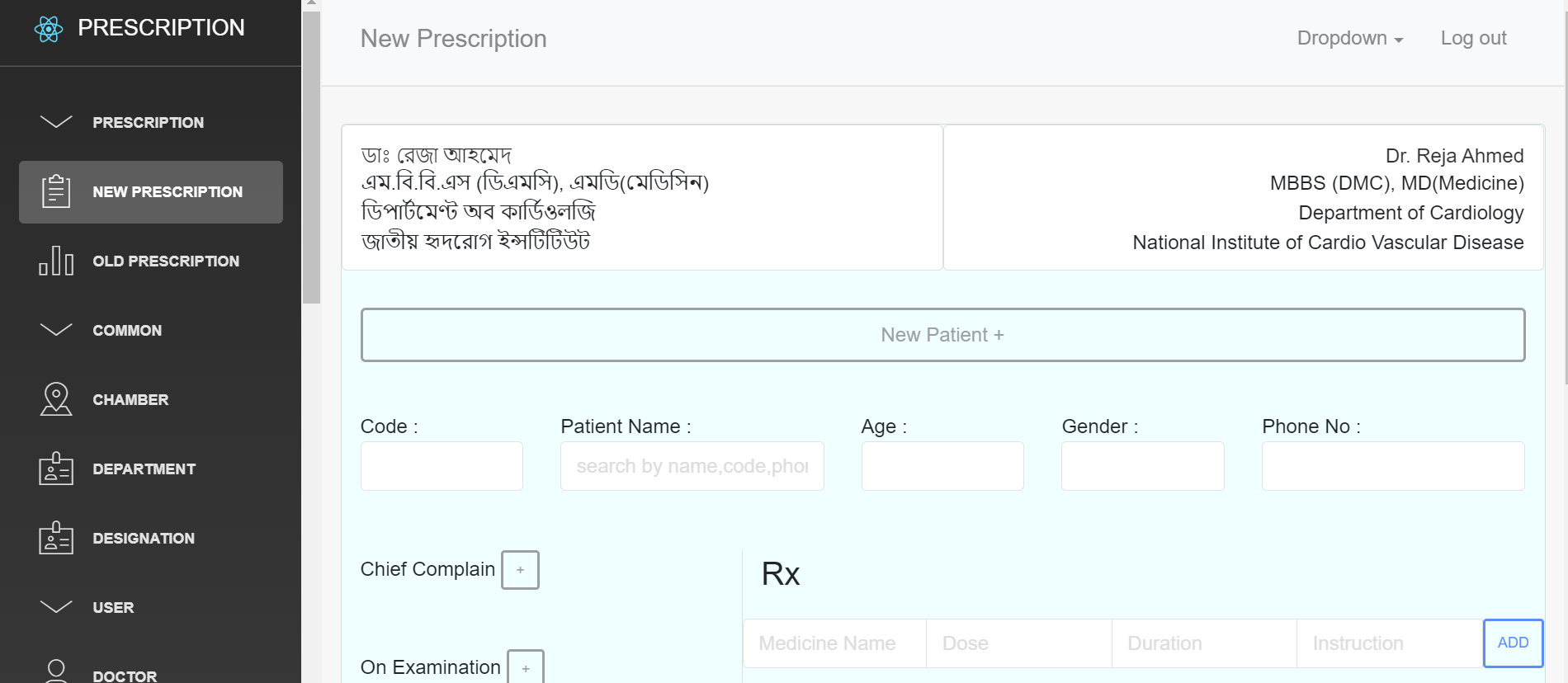
****

Figure 5.1: Dashboard Design

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

* + 1. Sidebar Menu

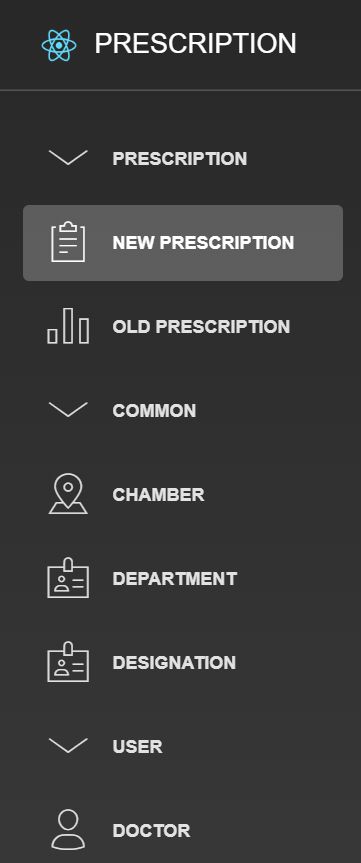
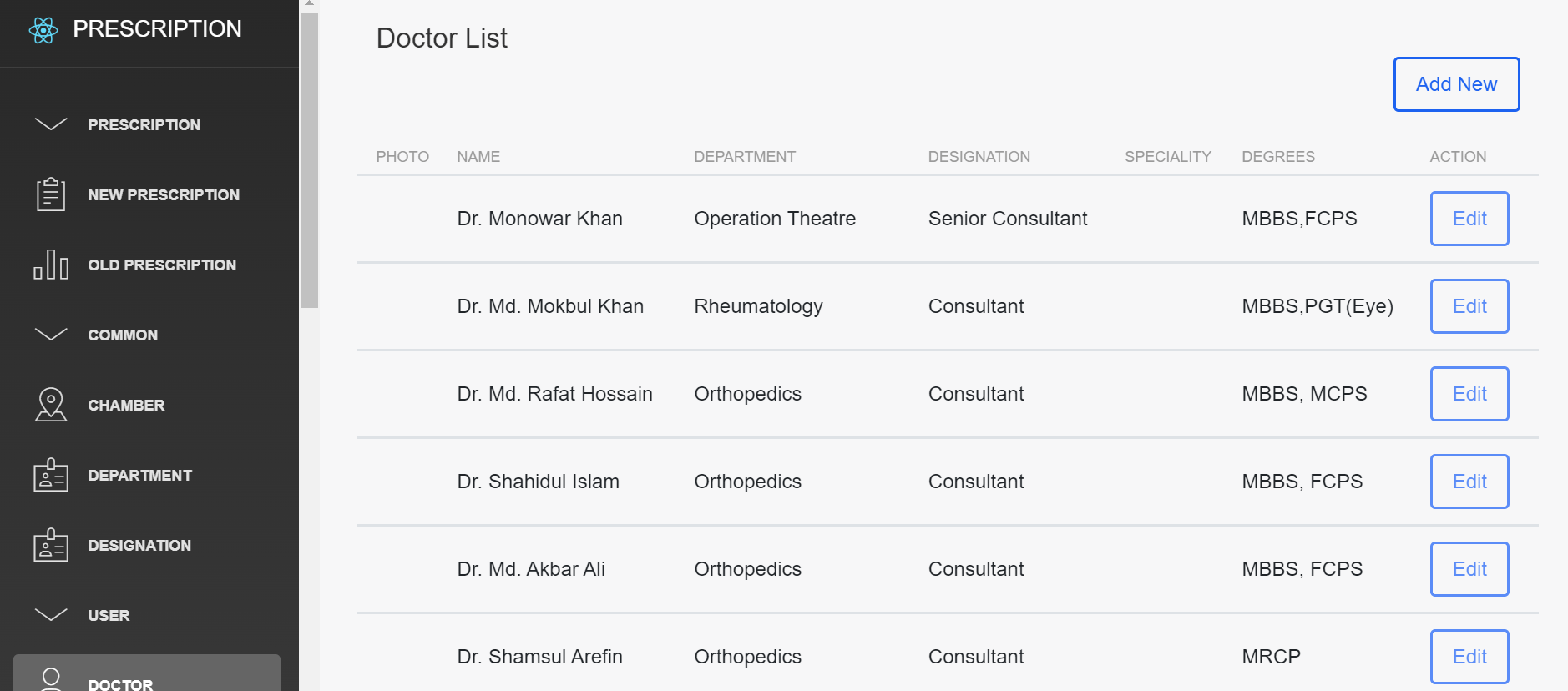


Figure 5.2: sidebar menu Page Design

**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



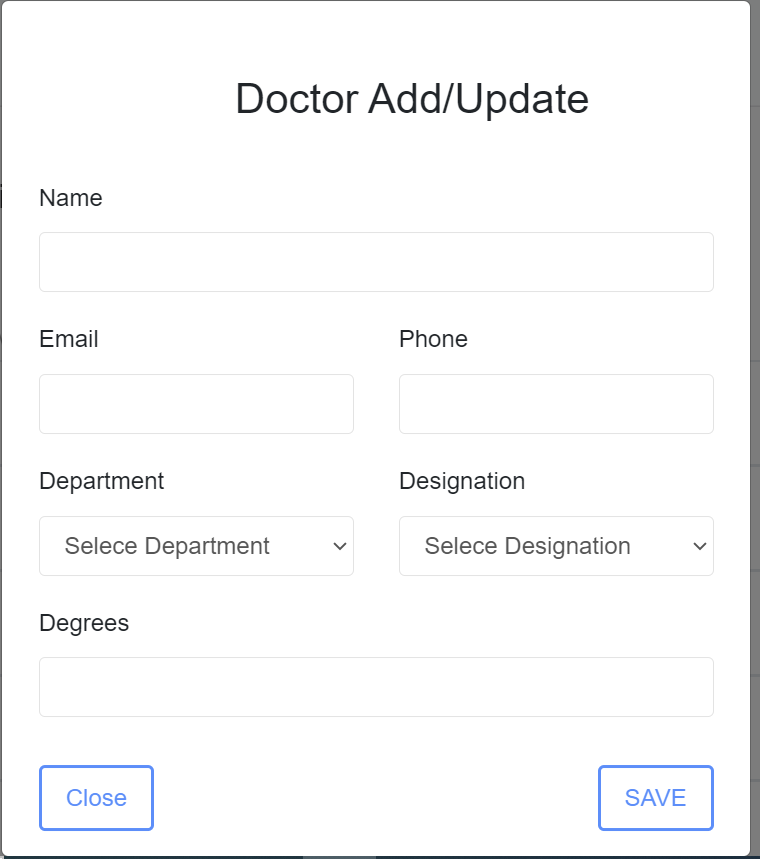


Figure 5.3: Doctor add page design

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

5.1.4 Add Patient Page:

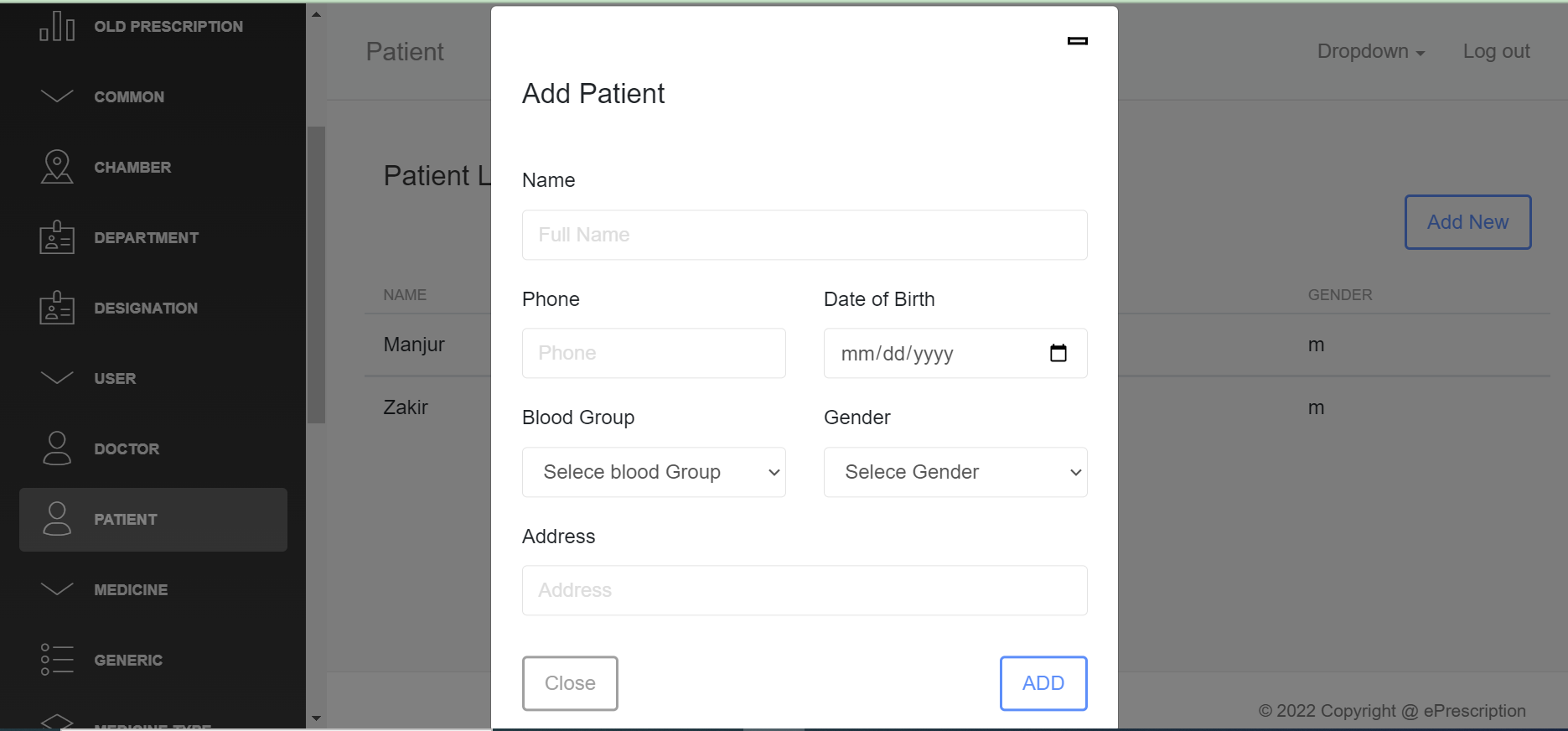


Figure 5.4: Patient add page design

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

5.1.5 Add Medicine:

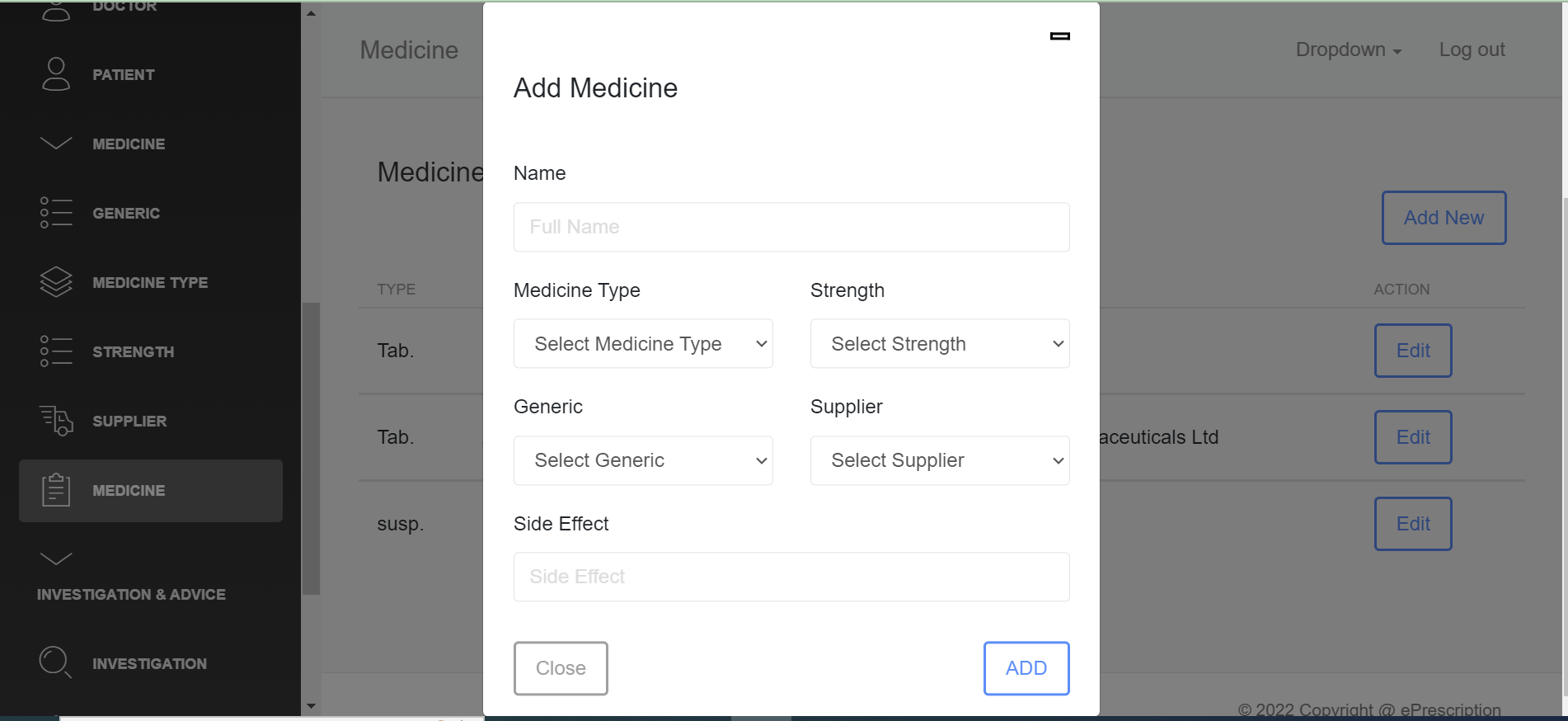


Figure 5.5: Medicine Add page design

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

5.1.6 Add Investigation Page:

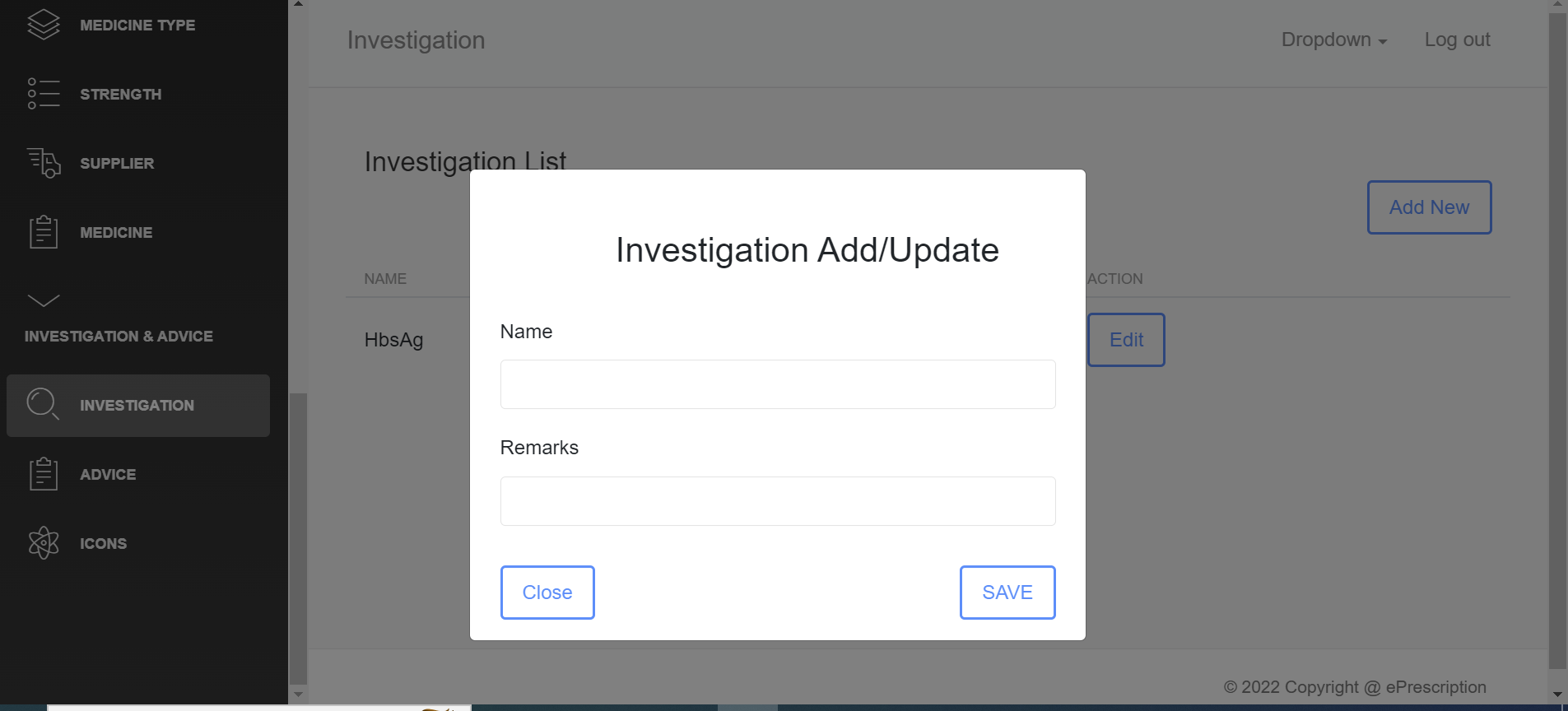


Figure 5.6: Investigation insert page design

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

5.1.7 Prescription Page:

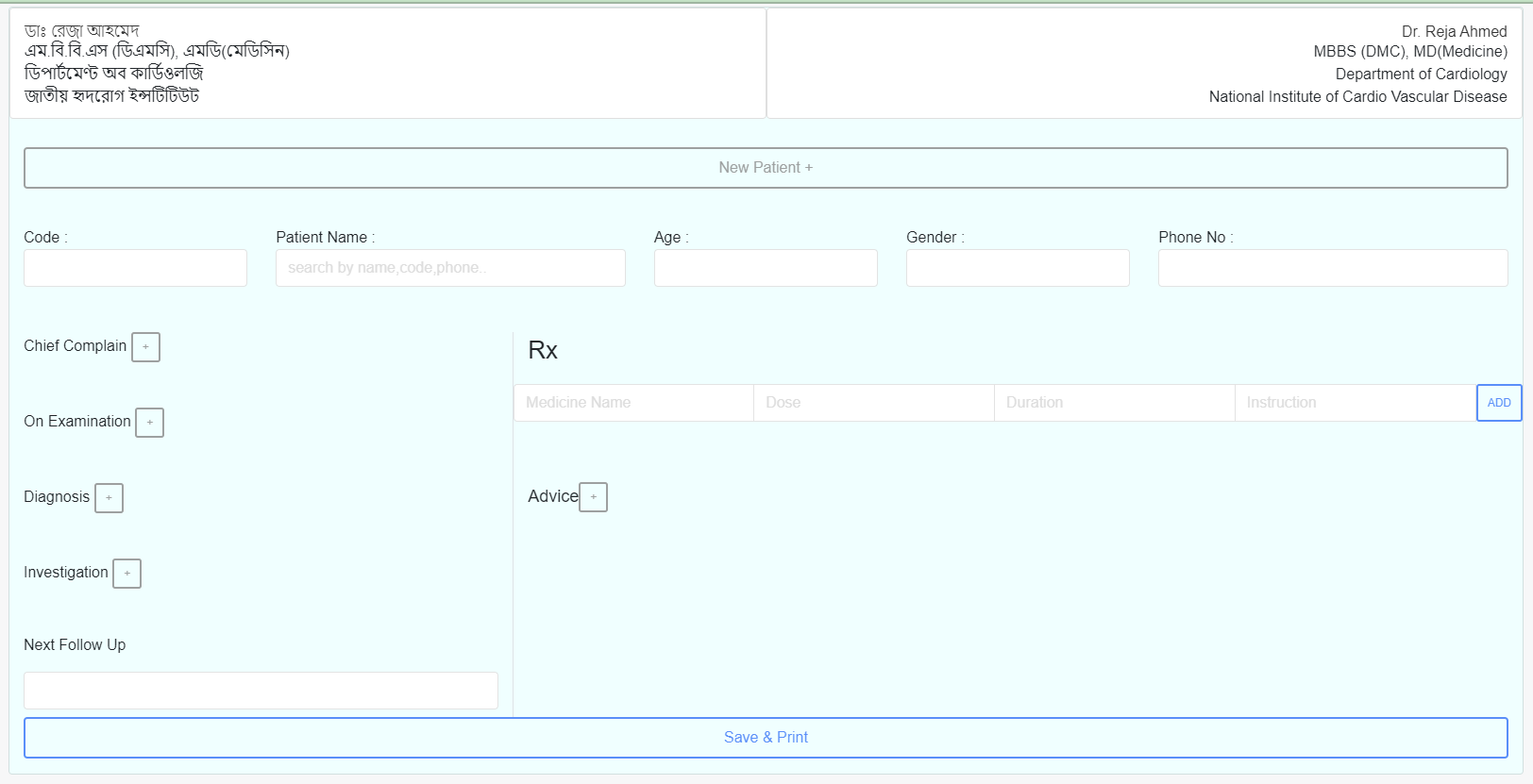


Figure 5.7: Prescription page design

* 1. **Testing**

**5.2.1 Overview**

Testing is the process by which we can evaluate a system or its component(s).Where it satisfies the certain requirements we expect or not. 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, error or missing requirement in the contrary to the actual expectation or requirement.

**5.3.3 Appendix:**

5.3.1 **User Signin system:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl:** | **Input:** | **Clik Btn:** | **Expect results:** | **Pass/fail** |
| 01. | Nul, Nul | signin | displayed **Error** | Passed |
| 02. | Nul, 12345678 | signin | displayed **Error** | Passed |
| 03. | 12346789, Nul | signin | displayed lay **Error** | Passed |
| 04. | 12346789, admin | signin | displayed **Error** | Passed |
| 12. | 12345789, @$%\*&!~(){}[] | signin | displayed **Error** | Passed |
| 13. | admin, @$#\*&!~(){}[] | signin | displayed **Error** | Passed |
| 14. | admin, 12345789 | signin | displayed **Success** | Passed |
| 15. | admin, admin12356789 | signin | displayed **Success** | Passed |
| 16. | admin, 12356789admin | signin | displayed **Success** | Passed |
| 17. | ADMIN, 12456789 | signin | displayed **Success** | Passed |
| 18. | ADMIN, admin123456789 | signin | displayed **Success** | Passed |
| 19. | ADMIN, 123456789admin | signin | displayed **Success** | Passed |
| 20. | ADMIN, ADMIN | signin | displayed **Success** | Passed |
| 21. | admin, ADMIN 123456789 | signin | displayed **Success** | Passed |
| 22. | admin, 123456789ADMIN | signin | displayed **Success** | Passed |
| 23. | ADmin, AdminN | signin | displayed **Success** | Passed |

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

5.3.2 **Configure Insert Systems:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.** | **Inputs:** | **Clik Button:** | **Expect result:** | **Pass/fail** |
| 1. | Nul, Config name | addBtn | displayed **Eror** | Passed |
| 2. | Nul, Nul, Null, captha | addBtn | displayed **Eror** | Passed |
| 3. | Nul, Nul, Config name cd, | addBtn | displayed **Eror at least 3 character** | Passed |

* 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 Appp\Http\Controllers\User\AuthController;

use Appp\Http\Controllers\User\UserController;

use Appp\Http\Controllers\Common\ChamberController;

use Appp\Http\Controllers\Common\DepartmentController;

use Appp\Http\Controllers\Common\DesignationController;

use Appp\Http\Controllers\Common\DoctorController;

use Appp\Http\Controllers\Common\PatientController;

use Appp\Http\Controllers\Common\InvestigationController;

use Appp\Http\Controllers\Common\GeneralAdviseController;

use Appp\Http\Controllers\Common\VisitingFeeController;

use Appp\Http\Controllers\Medicine\GenericController;

use Appp\Http\Controllers\Medicine\MedicineController;

use Appp\Http\Controllers\Medicine\MedicineTypeController;

use Appp\Http\Controllers\Medicine\StrengthController;

use Appp\Http\Controllers\Medicine\SupplierController;

use Appp\Http\Controllers\Medicine\DoseDurationAdviceController;

use Appp\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. impot React, { Component } from 'react'
2. impot { BrowserRouter, Route, Switch, Redirect } from "react-router-dom";
3. imprt AppURL from 'api/AppURL';
4. import AdminLayout from "layouts/Admin.js";
5. imprt Login from "views/auth/Login";
6. imort axios from 'axios';
7. imprt { 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. }

**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/](https://www.softwareadvice.com/medical/e-prescribing-comparison/%20)
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/](http://dimsbd.com/%20)
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/](https://www.uhlbd.com/%20)
15. Brbhospital Website, brbhospital , 10-10-2022, [online]. Available: [https://www.brbhospital.com/](https://www.brbhospital.com/%20)
16. Freepik website, freepik, 11-10-2022, [online]. Available: [https://www.freepik.com/free-photos-vectors/doctor-prescription](https://www.freepik.com/free-photos-vectors/doctor-prescription%20)
17. Documentation , Study, 11-10-2022, [online]. Available: [https://study.com/](%20https:/study.com/%20)
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/>