



Design and Development of Patient Data Management & Digital Prescription System: Smart Rx

by

Mohsin Islam

ID: CSE1902017062

Shikha Biswas

ID: CSE1902017089

Md Mizanur Rahman

ID: CSE192017055

Mst. Sharmin Akther

ID: CSE1901016159

Samia Rahman Chaity

ID: CSE1902017054

Supervised by

Bulbul Ahamed

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)**

January 2023



Design and Development of Patient Data Management & Digital Prescription System: Smart Rx

by

Mohsin Islam

ID: CSE1902017062

Shikha Biswas

ID: CSE1902017089

Md Mizanur Rahman

ID: CSE192017055

Mst. Sharmin Akther

ID: CSE1901016159

Samia Rahman Chaity

ID: CSE1902017054

Supervised by

Bulbul Ahamed

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)**

January 2023

APPROVAL

The project titled “**Design and Development of Patient Data Management & Digital Prescription System: Smart Rx**” submitted by Mohsin Islam (CSE1902017062), Shikha Biswas (CSE1902017089), Md Mizanur Rahman (CSE192017055), Mst. Sharmin Akther (CSE1901016159) and Samia Rahman Chaity (CSE1902017054) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Bulbul Ahamed

Associate Professor & Head,
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name & Signature)

Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name & Signature)

Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name & Signature)

Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Bulbul Ahamed, Associate Professor & Head,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

Signature

(Bulbul Ahamed)
Supervisor

Mohsin Islam
ID: CSE1902017062

Shikha Biswas
ID: CSE1902017089

Md Mizanur Rahman
ID: CSE192017055

Mst. Sharmin Akther
ID: CSE1901016159

Samia Rahman Chaity
ID: CSE1902017054

ABSTRACT

We have developed software called Smart Rx. It is a prescription writing and patient data management system. This is mainly for doctors. Through this software a doctor will be able to manage his chamber digitally. We have provided prebuilt data in this software which includes Medicine, Chief Complaint, On Examination, Investigation Advice, Dose, Duration, Instruction, Short Advice etc. Also there is a facility to bookmark the data which the doctor uses regularly for which one uses it. The doctor can create a prescription in less time. Also, all information and medical history of the patient will be preserved. A doctor can refer a patient to another doctor and if any patient is referred to him then the doctor can see all the information of that patient. And through this you will be able to provide the printed prescription to the patient; this will benefit the patient a lot. Also, patients can book doctor's appointments at home if they want. And if patient books an appointment online and the doctor prescribes him, then the patient can collect the prescription online.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the scheduled time.

We are auspicious that we had the kind association as well as supervision of **Bulbul Ahamed**, Associate Professor & Head, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude to **Prof. Dr Md Alamgir Hossain**, Dean, Faculty of Science and Engineering for his kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

ADEs	Adverse Drug Events
API	Application Programming Interface
CSS	Cascading Style Sheets
CVM	Cytomegalovirus
DBMS	Database Management System
DOM	Document Object Model
E-Prescription	Electronic Prescription
FDA	Food and Drug Administration
FTP	File Transfer Protocol
HER	Electronic Health Records
HTML	Hyper Text Markup Language
HTTPS	Hypertext Transfer Protocol Secure
ID	Identification
IP	Internet Protocol
IV	Investigation Advice
JSON	JavaScript Object Notation
MDN	Mozilla Developer Network
OE	On Examination
PE	Physical Examination
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSH	Secure Shell
SU	Sonargaon University
WWW	World Wide Web

TABLE OF CONTENTS

Title	Page No.
DECLARATION	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
LIST OF ABBREVIATION	vi
CHAPTER 1	1 – 4
INTRODUCTION OF THE PROJECT	
1.1 Introduction	1
1.2 Objectives	1
1.3 Motivation	1
1.4 Why Should Doctors Use Prescription Writing Software	1
1.5 Benefits of e-prescribing (Doctors and Patients).....	2
1.6 Importance of Using a Computerized description System	2
1.7 Why is Smart Rx Prescription a better choice?	3
1.8 Major Features	3
CHAPTER 2	5– 6
BACKGROUND OF THE PROJECT	
2.1 Introduction	5
2.2 Related works	5
2.3 Comparative Studies	5
2.4 Scope of the Problem	6
2.5 Challenges.....	6
CHAPTER 3	7-11
REQUIREMENT SPECIFICATION	
3.1 Requirement Collection and Analysis.....	7
3.2 Hardware and Software Requirement for our system	7
3.3 Design Requirements	7
3.6 Tools & Technology	10
3.7 Supporting Tools	11
CHAPTER 4	12– 13
DESIGN SPECIFICATION	
4.1 Front-end Design	12
4.2 Back End Design	12
4.4 Implementation Requirements	13
CHAPTER 5	14 – 22
IMPLEMENTATION AND TESTING	
5.1 How does it work	14
5.2 Software Frontend Interface with Feature Details	14

5.14	Expected Outcome	20
5.15	Time saving	20
5.16	Monetary saving	20
5.17	Sustain tranquility.....	21
5.19	Implementation of Interaction.....	21
5.20	Testing Implementation.....	21
CHAPTER 6		23
CONCLUSION AND FUTURE WORKS		
6.1	Conclusion	23
6.2	Limitation of the project.....	23
6.3	Future Works	23
REFERENCES	24
APPENDIX	25-53

LIST OF TABLES

<u>Table No.</u>	<u>Title</u>	<u>Page No.</u>
Table 5.21	Registration and Login Test Case (Patients and Doctors)	22
Table 5.21	Input User Information Test Case	22

LIST OF FIGURES

<u>Figure No.</u>	<u>Title</u>	<u>Page No.</u>
Fig.3.4	Database diagram	8
Fig.3.5	ER Diagram of projects	9
Fig.4.3	UX	12
Fig.5.3	Patient Appointment	14
Fig.5.4	Make Prescription	15
Fig.5.5	Prescription Print Preview	15
Fig.5.6	Prescription Print	16
Fig.5.7	Prescription List	16
Fig.5.8	Appointment List	17
Fig.5.9	Referral Patient List	17
Fig.5.10	Prescription Header Set up with Profile	18
Fig.5.11	Doctors Portal Landing Page	19
Fig.5.12	Patient Login/Registration	19
Fig.5.13	Patient Dashboard	20
Fig.5.18	Database Implementation	21

CHAPTER 1

INTRODUCTION OF THE PROJECT

1.1 Introduction

In our daily life we face a lot of problems. Disease is one of the most common issues for a person's life. If anybody is ill and wants to visit a doctor for checkup, he or she needs to visit the hospital and wait until the doctor is available. The patient also waits in a queue while getting an appointment. If the doctor cancels the appointment for some emergency reasons, then the patient is not able to know about the cancellation of the appointment unless or until he or she visits the hospital. So, it's necessary to get a consultation with Doctors whenever we get affected with various diseases. As the internet is now available for everyone therefore anyone can use the online appointment system to overcome such problems and inconvenience for the patients.

1.2 Objectives

Imagine a personalized and safe prescribing system that allows us to easily create and print prescriptions quickly and efficiently that makes life more productive for doctors & patients.

The electronic prescription writing software will also craft relationships between physicians, patients, and pharmacies with the help of automated prescription writing and an efficient patient management system for doctors.

1.3 Motivation

As a patient we face many difficulties when we want to get an appointment for a doctor in their chambers or places. When people get affected by illness they need to visit a doctor for checkup but they have to visit their chambers or hospital to get an appointment. It is a lengthy process and wasting people's time. Sometimes people do visit the doctor's chamber for a health check but the doctor is not available for various reasons. It's the only way to get to know when people just visited their places. It harasses people a lot, besides people need an ambulance service to carry patients to hospitals. Merely, people need to visit hospitals or clinics to hire ambulances, it is a time consuming process. Our motivation is, if we have an option to get this appointment very easily, that can be more precious for us. Then we have planned to implement a Web-based doctor appointment system.

1.4 Why Should Doctor Use Prescription Writing Software

- Electronic Health Record keeping
- Excellent Doctor Friendly Interface
- Compatible with Govt. Regulations

- Chamber Manage Digitally
- Create a prescription within 2-4 minutes
- Print-out Options for Prescription

1.5 Benefits of e-prescribing (Doctors and Patients)

PREVENT PRESCRIPTION DRUG ERRORS

According to the FDA, as many as 95,000 prescription drug errors have been made since 2000. Contributing factors to these errors include drugs that sound similar or have similar spelling. For example, Valtrex (valACYclovir) and Valcyte (valGANciclovir), are both used for cytomegalovirus (CMV), but for different reasons. A heart transplant patient receiving Valtrex when they need Valcyte is a problem. By utilizing E-Prescribing, both the prescriber and those who handle the prescription along the way can feel assured they have given the patient the correct medication.

REDUCE LOST PRESCRIPTIONS

Paper prescriptions are small and easy to misplace which adds work for medical offices as they must rewrite or call the prescriptions into the pharmacy. E-Prescribing allows prescriptions to be sent directly to the pharmacy, so patients don't have to keep track of another piece of paper.

INCREASE PATIENT SAFETY

E-Prescribing improves patient safety by increasing prescription legibility and decreasing medication errors and adverse drug events (ADEs).

1.6 Importance of Using a Computerized Prescription System

- (One of the best Automated Prescription Software in the market) will increase doctors' efficiency dramatically.
- Doctors can now prepare prescriptions within 20-60 seconds using Smart-Rx Software. More patients can be prescribed within a specific time.
- Well formatted Prescription, no hassle of bad handwriting.
- No need for entry/writing the same things again and again.
- The patient's history will be recorded forever and can be checked anytime for any type of analysis.
- Patients Appointment Serial entry will be smoother than ever.

1.7 Why is Smart Rx Prescription a better choice?

- Shortest prescription writing time.
- Highly customizable to fulfill doctor's requirements.
- 25000+ Drugs incorporated in the database for quick writing prescription.
- 750 + Chief Complaint incorporated in the database for quick writing prescription.
- 500+ Investigation incorporated in the database for quick writing prescription.
- 300+ On Examination incorporated in the database for quick writing prescription.
- Privacy, Security and Reliability.
- Proven return on investment.
- Increase Doctors and office efficiency.
- Handouts, Referral Forms,
- Smart Rx prescription software allows doctors to archive cases for research purposes.

1.8 Major Features

Easy to generate prescriptions in Bangla and English: Quickly pinpoint the exact information both in Bangla and English help Doctors to increase prescription writing.

Drug database: Integrated DIMS drug database enables quick search of drugs and drug details, making it extremely convenient.

Patient History Documentation: Patient's complete demographic details are stored for quick viewing.

Provision of Medical Data analysis: Doctors can archive clinically important cases for research purposes.

Electronic Health Records: Capture all EHR data of each visit for future access.

Customizable Prescription Print: Fully customizable prescription template prints out options with headers. Advice & notes in Bangla.

Medical History: Complete medical history recording is the matter of a few clicks.

Patient appointment: To make a schedule with your patient. It ensures online appointments on 24/7, no need to engage an extra manpower, and the patient feels comfortable without the hazard of phone calling. Ensure patient presence on-time.

Patient PE sections: this will guide you for physical examination and allow you to keep records of each patient and easy to follow up visits, future reference and research.

Patient Ix form: Allow you to keep all investigation reports for follow up, future reference and research. There is provision to keep records as images (x-ray, USG, EKG) and view whatever you would like to see next.

Prescription writer: Doctors will be able to write prescriptions quickly even as short as handwriting time. No chance of error (double checked). You can print it for a patient to deliver in-hand or can send a copy to the patient's email or to a specific drug store or to a diagnostic laboratory or to a doctor that you want to refer to.

Why is the prescription writer quick? Because all brand medicines are generated from the DIMS app. You can see drugs that you want to recommend and dosage ready to input

Advice: Patient will get a handout of disease in non-professional language as you want to deliver in the mail or with the prescription.

Referral- You can refer a patient to a doctor or hospital with a courtesy letter and clinical findings by one click.

CHAPTER 2

BACKGROUND OF THE PROJECT

2.1 Introduction

For developing a system, we need to cover some initial study. It helps to investigate a system's background information. Also it helps to find the fault of this system so that necessary steps can be taken here to improve the project by adding and updating new features.

Our main goal of this project is to get the appointment within a less time & without getting any trouble from any patient. We developed this project for common people when they need to consult any doctors by getting their appointment within a short time by booking in online and getting a checkup at their desired time. Besides some blood donors profile is included there to get any purpose of donating also ambulance service is there for hiring and hospitalized immediately.

On the other hand, users can purchase their required medicine at a time from this system. So they won't be tense about the purchasing of any medicine whenever they need. If people get all the solutions at a time on the web it will be very helpful for them. So, people will now get interested in using the internet and get all medical solutions in a web-based system.

2.2 Related works

There are some related systems available right now but not a large number and no one is totally similar. Many of the systems have some limitations. From the study of this similar project we got interested in developing our system. Some of the related systems of our project are mentioned below.

We have explored many websites which are related to medical health consciousness, first our attention caught on 'Doctorola.com'. In their system users need to search for a doctor or hospital from different locations and get them for booking appointments. There is no user login option and personal profile so users are totally detached from getting extra facilities in future purpose. Another project is 'Doctorsbd.com'. This site provides only a doctor's list. Users are only able to know their service location from here.

2.3 Comparative Studies

Our modern age of technology is greatly dependent on the internet. An online system is also known as a Web-based system. As the world is going so fast, there is always a desire to communicate faster and effectively. There is no such thing that comes out without any limitations but we focused to overcome the best we could do. Our main focus was to determine the features for the patients. They are the large community of this system and they deserve to get the most out of this system. By using this system both doctors and patients become beneficial.

A lot of patient's daily visit to healthcare clinic or hospital and facing problem regarding have no knowledge about doctor specialty, wait for a long time to get doctor appointment and patients have no about doctor fee. There are several ways to book an appointment. A person can either go to the hospital directly for consultation or make an appointment from home through the internet by Web-based system. To solve this problem, we have developed an appointment system which improves the patient's satisfaction so the patient

can get reliable and timely access. So that we wish to offer such a system which will help them a lot. Our system will provide the best result and save their valuable time too.

2.4 Scope of the Problem

As we said, patients face some difficulties making appointments with doctors. Our present system isn't as the requirement is going. So, we can recapitulate some problem here;

- Using different numbers of platforms may not be enough for every patient to make an appointment with a doctor.
- There is some manual system for appointments, but this is fully online based.
- The people are not fully trusted on online system so that they will not get proper benefit from this system.

According to these problems, our system offers solutions that will help patients. users can easily access the system anytime and anywhere. This system is very simple and user friendly;

Few of them are:

For Patients, help

- Printed Prescription copy
- Can book an appointment from home.
- Smart way of appointment
- Don't worry if the prescription runs out.
- You will be able to collect from the doctor again.
- Reduce the appointment delay.

For Doctors, help

- No need any assistant for appointment
- Easily access to history of medication of a patient.
- Chamber can be managed digitally.
- Can refer patients to other doctors.
- A printed pad is not required to print the prescription.

2.5 Challenges

Every work has some challenges. So, we have to face a small number of challenges too.

1. As it's a Web-based system, so people need to appointments and reservations online through any web connected devices such as computer, laptop, smart phone, tablets etc. Then the system will not help them.

2. If a doctor doesn't check the notification for confirmation which is booking by patient then the main motto of this system will fail.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Requirement Collection and Analysis

Requirement collection and analysis is a very crucial part of any project. Without analysis, collecting data or good planning, a project will never complete properly. When we are developing on a project, a delivery time of the project has already been given. That's why project work must be planned and executed to the delivery time.

Our project is an online Web-based doctor appointment system. Analysis and requirement collection was our big challenge, when we started thinking about this project. After starting analysis we figure out some significant features that boost our project.

3.2 Hardware and Software Requirement for our system

Whenever you install software or hardware for your computer, you should first make sure that your computer supports the system requirements. We need internet connection based system hardware such as computers, laptops, tablets, smart phones and so on.

We have used some software tools and platform for development is described below:

- HTML
- CSS
- Bootstrap
- React
- PHP
- MySQL
- Postman

3.3 Design Requirements

When we design software or a system, we consider some requirements to make the project efficient. We have tried to build a user-friendly web application. Our system user interface is very simple and everyone can use it without facing any problem. For better knowing, we have drawn a business process model, use case diagram, activity diagram and logical data model. We have used MySQL for designing our database. For front end design we have used HTML, CSS. We have used OOP, Laravel framework and PHP for back end design.

3.4 Database diagram

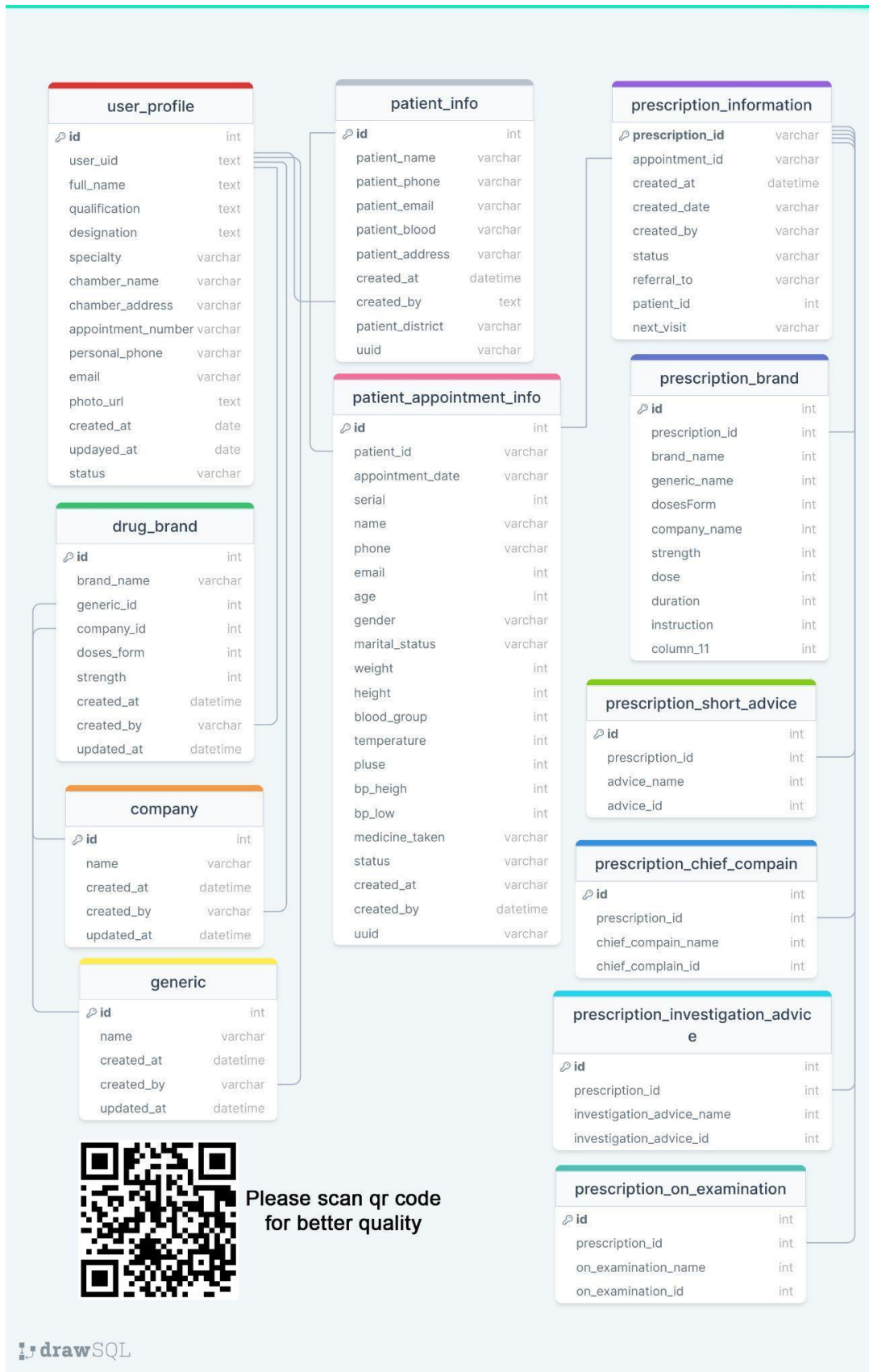


Fig 3.4: Database diagram

3.5 ER Diagram

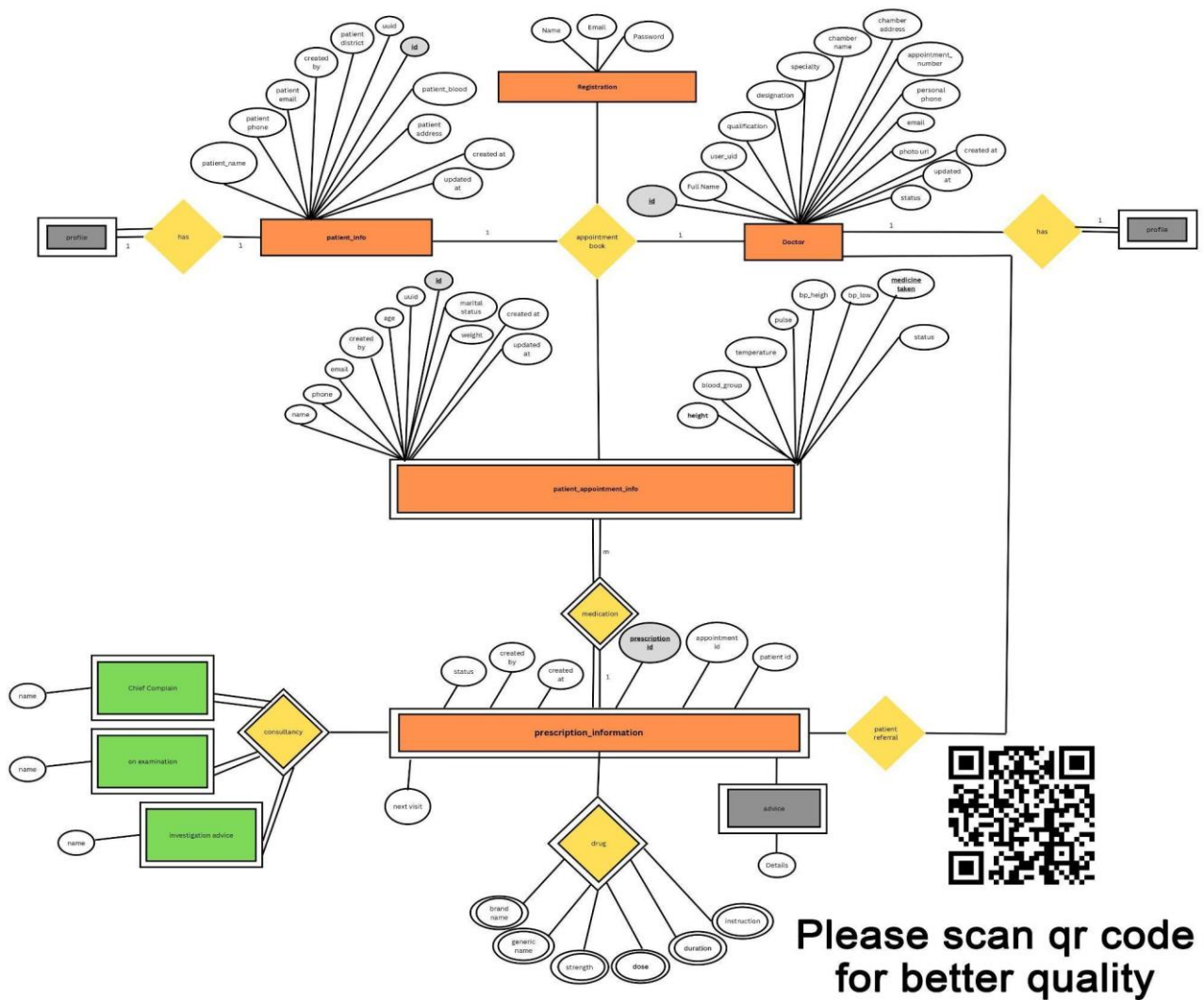


Fig 3.5: ER Diagram of projects

3.6 Tools & Technology

Application Type



Web-based applications (Online)

Browser Support



Chrome



Firefox



Chromium



Brave



Safari

A web browser is a software application that is used to locate, receive and display content received from the web servers. Popular web browsers include Mozilla Firefox, Google Chrome, Internet Explorer and Safari

Frontend



React
v18



React-Router-Dom
v6.6.2



Firebase
v10.1.0



Bootstrap
v5.2



Axios
v1.2.2

Database and API



PHP
V7.4



MySQL Database
v8.0



Json Data

3.7 Supporting Tools



Code Editor: Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications.



Local Server: XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache



API Tester: Postman is an API platform for developers to design, build, test and iterate their APIs

Networking

TCP (transfer control protocol) and IP (internet protocol) defines as how computers should

Connect with each other through the internet and how they share information. Common TCP/IP

Protocols are HTTPS, HTTP, and FTP. We use these protocols to access the website without

Even realizing it. FTP (file transfer protocol) used to download files from a browser.

HTTPS (where the “S” stands for security) responsible for the secure communication between a server and a browser like an account information or personal information

Ports

Ports allow one device to connect with the other through a different and unique IP address. A device can have more than one port. Port 25 (SMTP) is usually reserved for email. It is used to transmit data from remote email servers. If that port is blocked no emails can be sent. Some other common port numbers are 80(HTTP), 443 (HTTPS), 21 (FTP), 22 (SSH). Firewalls can often block a port to tighten the security.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-end Design

The front end is normally designed in two parts. The web design and front end web development. We use HTML, CSS and JavaScript for our front end design. These include things like fonts, drop-down menus, contract forms etc. In this section we include some home page front end design.

4.2 Back End Design

The backend is normally designed in two parts

- API
- A database

Users input the data that the application stores in a database that was created on a server. We basically used PHP, phpstorm etc.

4.3 Interaction and Design and UX

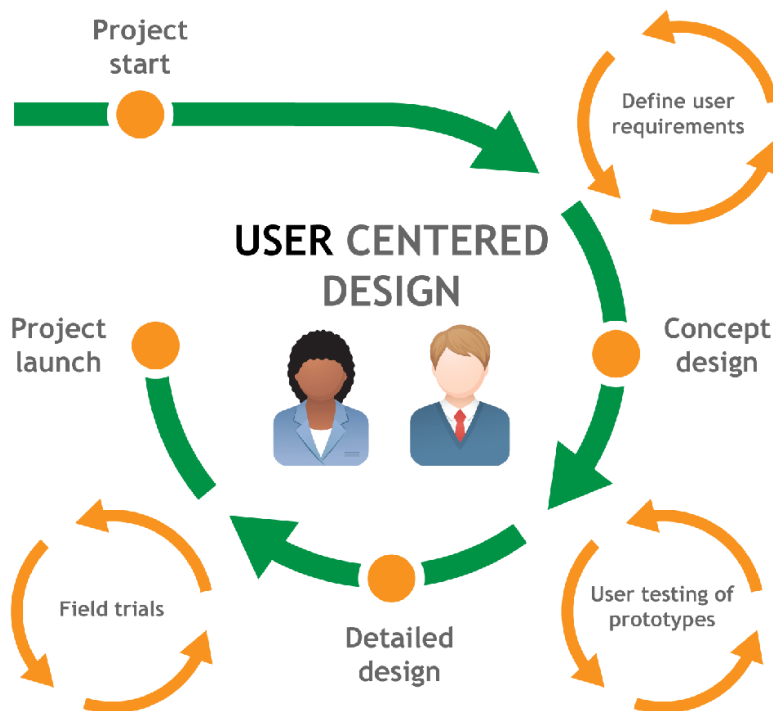


Figure 4.3: UX

4.4 Implementation Requirements

The main task of this part is to ensure or make all the things easier, useful and user friendly. The implementation Requirement gave us a very good idea. The list of implementation requirement is given below:

Easier to create

- Faster
- Clean Interface
- Easier to interact
- User-friendly
- Delightful
- Dynamic pages
- Easier to manage

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1: How does it work?

Appointment: Patients schedule an appointment from online, physically and over the phone

Confirmation: Appointment confirmed upon from doctor assistant

Consultancy: Doctor's assistant will call and confirm if the patient takes an appointment online. The patient can come directly and take an appointment, no confirmation is required.

Prescription: Doctors prescribe medications by Smart Rx online software and patients get prescription printed copy instantly.

5.2: Software Frontend Interface with Feature Details

Doctor Part

Patient Appointment

The first step in creating a prescription is to make an appointment. There are more than 12 fields to collect patient information to create an appointment. Patient name, mobile, appointment date, serial number must be given.

The screenshot shows the 'Make Appointment' form in the Smart Rx application. The form is divided into two main sections: 'Basic Information' and 'Add more info'. The 'Basic Information' section includes fields for Name, Phone, Email, Age, and Gender. The 'Add more info' section includes fields for Marital Status, Weight, Height, Blood Group, Temperature, and Pulse. A sidebar on the left contains navigation options like 'Create Prescription', 'Prescription List', 'Today Appointment', 'Appointment List', 'Settings', 'Clinical Options', and 'Referral Patients'. The 'Appointment List' section shows a list of appointments with columns for '1 | Rabbil Hasan' and 'Appointment List'. The 'Logout' and 'Profile' buttons are visible at the bottom of the sidebar.

Fig 5.3: Patient Appointment

Make Prescription

After the patient makes an appointment, the second step is the section on creating a prescription. The doctor will listen to the patient's chief complaints and perform an examination. After that, the doctor will advise the investigation if needed and if he thinks it is necessary to give medicine, he will give medicine to eat. Also, the doctor can specify the date for the next appointment and if he feels that he needs to be referred to another doctor, then he can select the doctor from the referral.

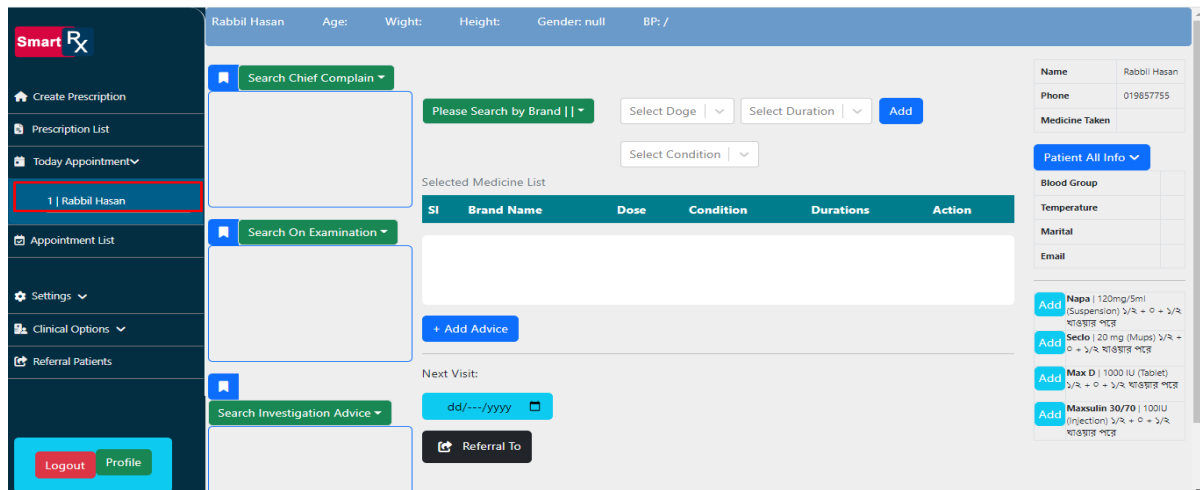


Fig 5.4: Make Prescription

Prescription Print Preview

After saving the prescription, the third step will show the prescription print preview. The top part shows the doctor's information, the lower part has the patient's information and the left side has the clinical history and the right side has the prescription medicine list.

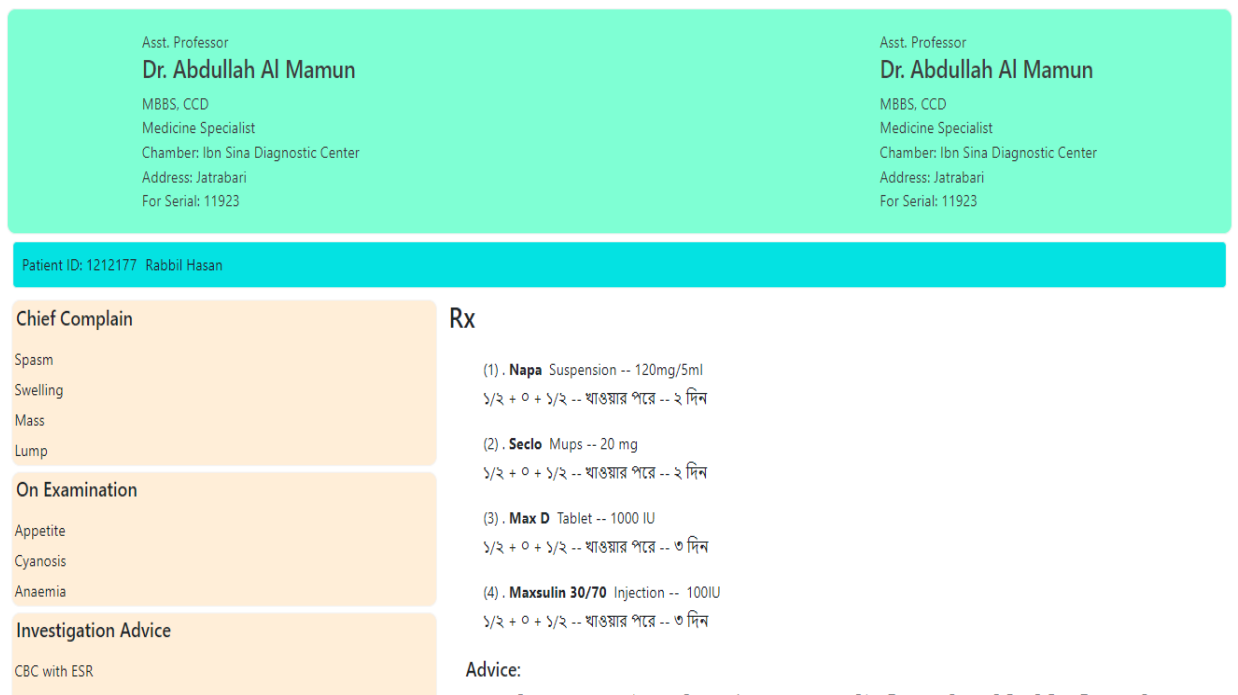


Fig 5.5: Prescription Print Preview

Prescription Print

Coming to the final step, the doctor will provide the printed copy of the prescription to his patient. Usually the printed copy is of A4 size, the doctor can change it if he wants.

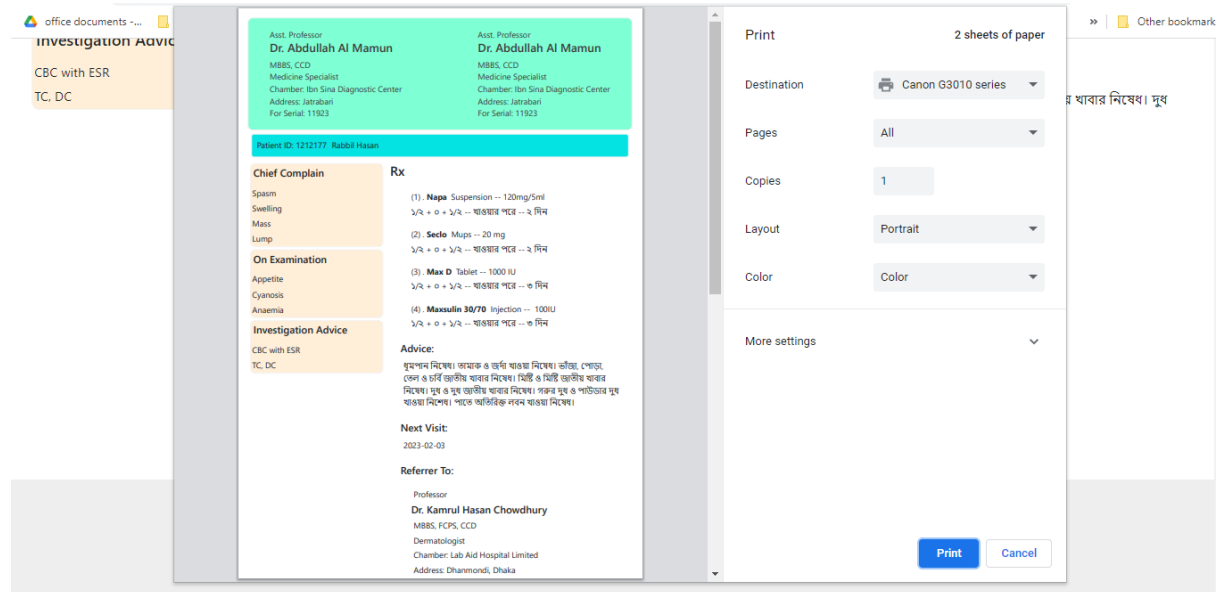


Fig 5.6: Prescription Print

Prescription List

The doctor can see the prescriptions made for his patients. You can also find and print specific prescriptions. To search the prescription, you need to search by patient's ID or phone number.

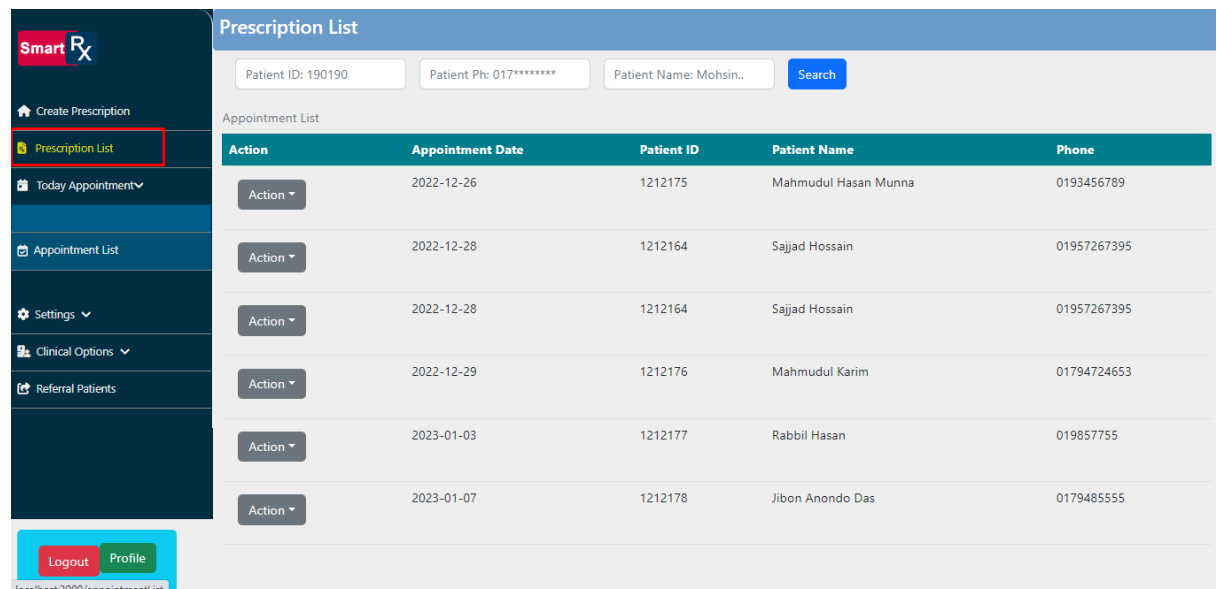


Fig 5.7: Prescription List

Appointment List

The doctor will see all the patients who have made an appointment with him in the appointment list. Neediest can see the patient's information by searching with the patient's ID or phone number. Prescription was made for the patient but can be understood by looking at the status.

Action	Appointment Date	Patient ID	Patient Name	Phone	Status
Action ▾	2022-12-26	121170	Mahmudul Hasan Munna	0193456789	Prescription Ready
Action ▾	2022-12-28	121171	Sajjad Hossain	01957267395	Prescription Ready
Action ▾	2022-12-29	121172	Mahmudul Karim	01794724653	Prescription Ready
Action ▾	2023-01-03	121173	Rabbil Hasan	019857755	Prescription Ready

Fig 5.8: Appointment List

Referral Patient List

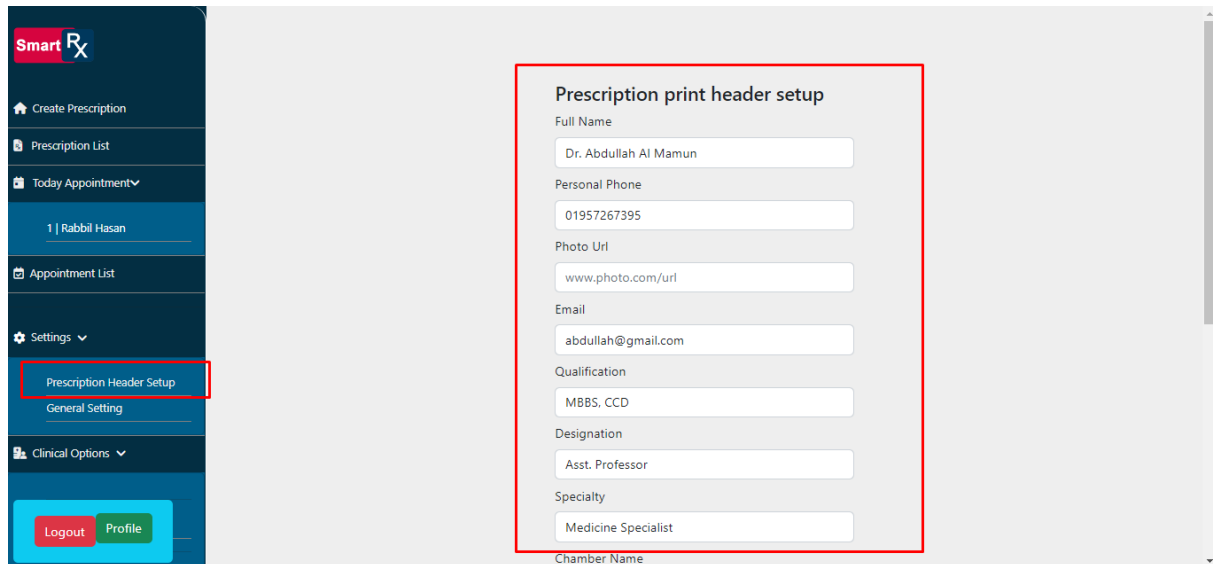
If the doctor who uses Smart Rx refers patients one after the other, then all the information of the referral patient can be seen from the referral patient list. You can use an ID or phone number to search for patients. The doctor can print the patient's prescription.

Action	Appointment Date	Patient ID	Patient Name	Phone
Action ▾	2022-12-28	1212164	Sajjad Hossain	01957267395
Action ▾	2022-12-28	1212164	Sajjad Hossain	01957267395
Action ▾	2023-01-07	1212178	Jibon Anondo Das	0179485555

Fig 5.9 Referral Patient List

Doctor Print Header Setting with Profile

Profile must be set up to show doctor information on prescription top.



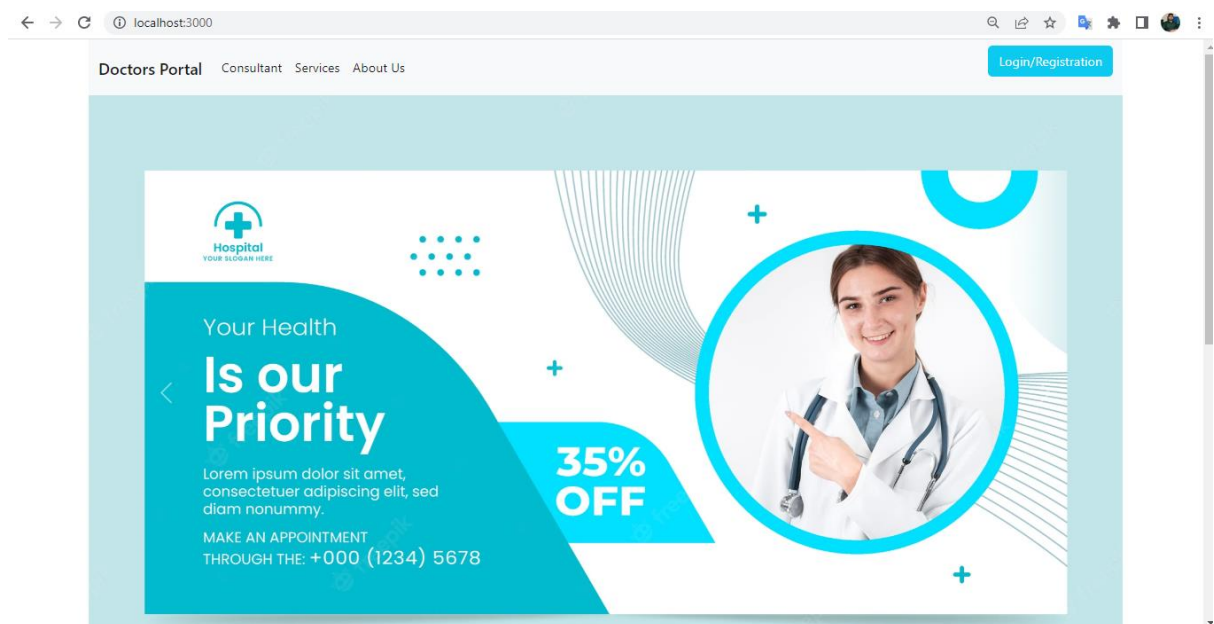
The screenshot displays the 'Smart Rx' application interface. On the left is a dark blue sidebar with navigation options: 'Create Prescription', 'Prescription List', 'Today Appointment' (with a dropdown arrow), '1 | Rabbil Hasan', 'Appointment List', 'Settings' (with a dropdown arrow), 'Prescription Header Setup' (highlighted with a red box), 'General Setting', 'Clinical Options' (with a dropdown arrow), and 'Logout' and 'Profile' buttons at the bottom. The main content area is titled 'Prescription print header setup' and contains several input fields, all highlighted with a red box: 'Full Name' (Dr. Abdullah Al Mamun), 'Personal Phone' (01957267395), 'Photo Uri' (www.photo.com/url), 'Email' (abdullah@gmail.com), 'Qualification' (MBBS, CCD), 'Designation' (Asst. Professor), 'Specialty' (Medicine Specialist), and 'Chamber Name' (empty).

Fig 5.10: Prescription Header Set up with Profile

Patient Part:

Landing Page

There is a website called Doctor Portal for patients to book doctor appointments at home.



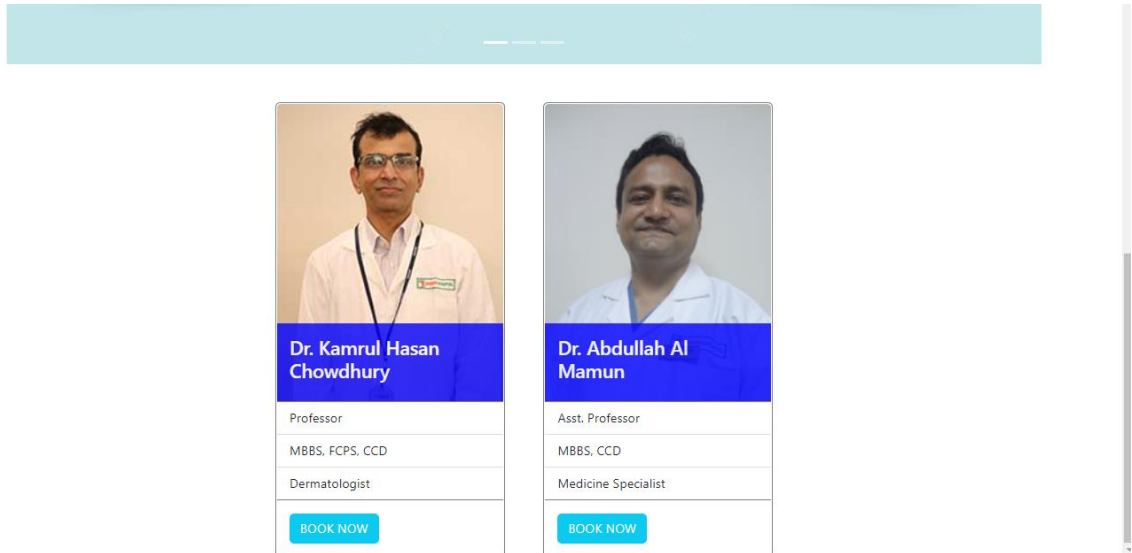


Fig 5.11 Doctors Portal Landing Page

Patient Login/Registration

Login/Registration is required to book doctor appointments.

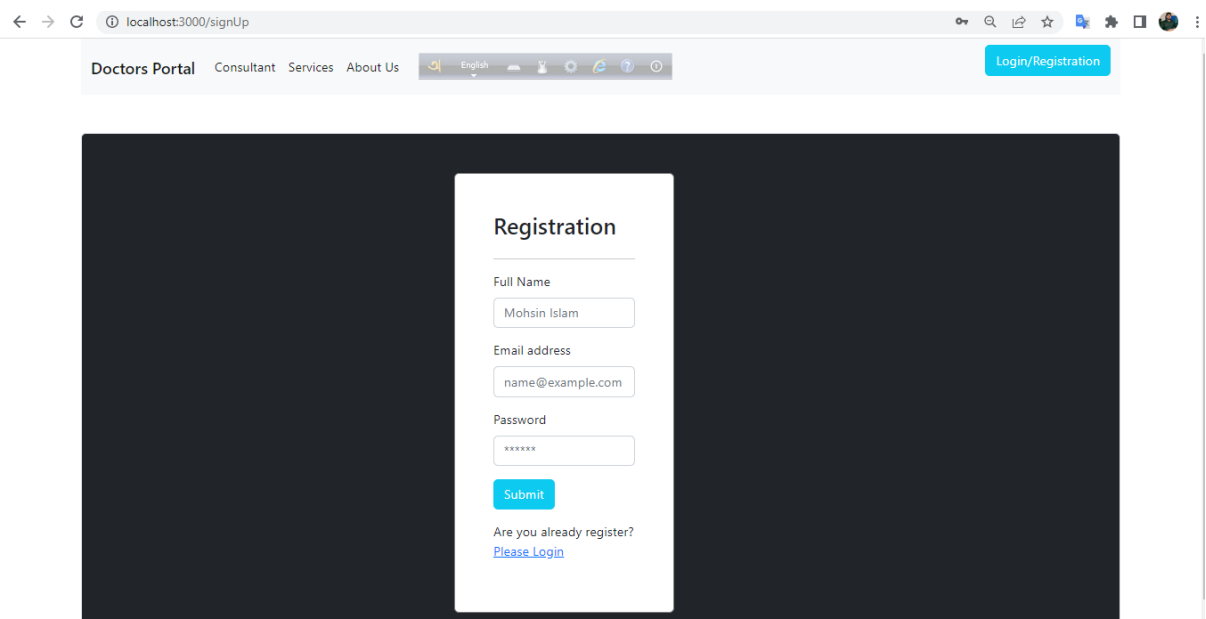


Fig 5.12.: Patient Login/Registration

Patient Dashboard

After taking the doctor's appointment, the patient should come directly and take the doctor's consultation. If the doctor prescribes, the patient can collect the prescription online.

Actin	Appointment Date	Appointment ID	Patient Name	Phone	status	Consultant Info
Actin ▾	05.12.2022	121311	Mizanur Rahman	0179374856	Prescription Ready	Dr. Kamrul Hasan Chowdhury

Fig 5.13: Patient Dashboard

5.14 Expected Outcome

There is an online scheduling system commonly referred to as a Web-based pattern that allows individuals to conveniently and securely book people appointments and reservations online through any web connected devices such as computer, laptop, smart phone, tablets etc. Once a date and time are selected the system will give booking confirmation and recorded documents for the next requirement.

The flexibility of our system enables it to be utilized for a variety of different services and activities for a patient and doctor, such as,

5.15 Time saving

Staffs spend a lot of time on the phone booking and can't maintain appointments properly so booking through online by individual's saves time as they no longer have to commit a part of their busy schedule to calling their medical, healthcare or wellness provider. As an example, a typical phone booking system spends an average of four minutes booking a hundred patients. Where our system is will take less time.

5.16 Monetary saving

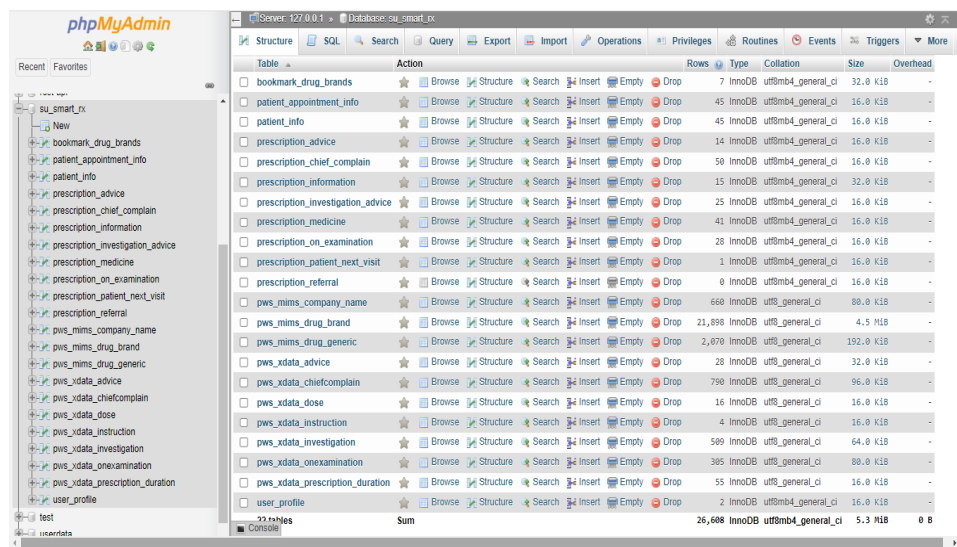
In Doctor's chambers the staffs are always ready to take money for giving appointments to patients. It is an unethical way to get a faster appointment. In our system people will be able to see the whole slots of any doctor so he/she can make an easy appointment for them whenever they need without paying extra money to the staff.

5.17 Sustain tranquility

If a person gets ill and wants to visit a doctor for checkup, he or she needs to visit the chambers and wait until the doctor is available. The patient also waits in a queue while getting an appointment. So a messy environment is possible. If the doctor cancels the appointment for some emergency reasons then the patients will try to make an uproar in those places. In this system, there is no need to wait for a while in the queue and patients will be able to see when doctors are available so that people will easily avoid the massing situation.

5.18 Implementation of Database

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.



The screenshot shows the phpMyAdmin interface for a database named 'su_smart_rx'. The left sidebar displays a tree view of the database structure, including tables like 'bookmark_drug_brands', 'patient_appointment_info', 'prescription_advice', etc. The main panel shows a table structure view for 'bookmark_drug_brands'.

Table	Action	Rows	Type	Collation	Size	Overhead
bookmark_drug_brands	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_general_ci	32.0 KiB	-
patient_appointment_info	Browse Structure Search Insert Empty Drop	45	InnoDB	utf8mb4_general_ci	16.0 KiB	-
patient_info	Browse Structure Search Insert Empty Drop	45	InnoDB	utf8mb4_general_ci	16.0 KiB	-
prescription_advice	Browse Structure Search Insert Empty Drop	14	InnoDB	utf8mb4_general_ci	16.0 KiB	-
prescription_chief_complain	Browse Structure Search Insert Empty Drop	59	InnoDB	utf8mb4_general_ci	16.0 KiB	-
prescription_information	Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_general_ci	32.0 KiB	-
prescription_investigation_advice	Browse Structure Search Insert Empty Drop	25	InnoDB	utf8mb4_general_ci	16.0 KiB	-
prescription_medicine	Browse Structure Search Insert Empty Drop	41	InnoDB	utf8mb4_general_ci	16.0 KiB	-
prescription_on_examination	Browse Structure Search Insert Empty Drop	28	InnoDB	utf8mb4_general_ci	16.0 KiB	-
prescription_patient_next_visit	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
prescription_referral	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
pws_mims_company_name	Browse Structure Search Insert Empty Drop	668	InnoDB	utf8_general_ci	88.0 KiB	-
pws_mims_drug_brand	Browse Structure Search Insert Empty Drop	21,858	InnoDB	utf8_general_ci	4.5 MiB	-
pws_mims_drug_generic	Browse Structure Search Insert Empty Drop	2,878	InnoDB	utf8_general_ci	192.0 KiB	-
pws_xdata_advice	Browse Structure Search Insert Empty Drop	28	InnoDB	utf8_general_ci	32.0 KiB	-
pws_xdata_chiefcomplain	Browse Structure Search Insert Empty Drop	798	InnoDB	utf8_general_ci	96.0 KiB	-
pws_xdata_dose	Browse Structure Search Insert Empty Drop	16	InnoDB	utf8_general_ci	16.0 KiB	-
pws_xdata_instruction	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16.0 KiB	-
pws_xdata_investigation	Browse Structure Search Insert Empty Drop	509	InnoDB	utf8_general_ci	64.0 KiB	-
pws_xdata_onexamination	Browse Structure Search Insert Empty Drop	385	InnoDB	utf8_general_ci	88.0 KiB	-
pws_xdata_prescription_duration	Browse Structure Search Insert Empty Drop	55	InnoDB	utf8_general_ci	16.0 KiB	-
user_profile	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
Sum		26,688	InnoDB	utf8mb4_general_ci	5.3 MiB	0 B

Fig 5.18: Database Implementation

5.19 Implementation of Interactions

Interactions are a common thing in today's system. It is a key to make a system dynamic and attractive to users. And it is very necessary to make a system interactive. Like we said before we tried to build a user friendly and an interactive platform. It should be done in such a way that attracts the user.

5.20 Testing Implementation

The purpose of this test is to evaluate our system's compliance with the specified requirement. We tried to make this system secure. Every single data that a user input on our system must pass the test.

5.21 Test Result and Reports

Table 1: Registration and Login Test Case (Patients and Doctors)

S1 No	Test case	Input	Expected outcome	Actual output	Result
1	Email check	Input without @	Invalid email	Invalid email. Must have @	Passed
2	Valid phone number	Input numeric value	correct	Correct	Passed
3	Invalid phone number	Input character	Incorrect	Numeric value required	Passed
4	Special character	Input special character	Special character not allowed	Not allowed	Passed

Table 2: Input User Information Test Case

S1 No	Test case	Input	Expected outcome	Actual output	Result
1	Display the webpage	Tested on us browser, chrome	Display successful	Display successful	Passed
2	User name	Wrong	Login failed	Login failed	Passed
3	Blank or Incorrect pass	Wrong password	Login failed	Login failed	Passed
4	Username	Input character	Special character not allowed	Letter and whitespace allowed	Passed

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

Doctor and patient appointment system is a very exciting topic to work on. After going through the work, we faced many challenging tasks. Day by day healthcare systems become an important part of our society. So we have decided to build this system.

We researched so many systems that showed us the direction how to develop our system. We interact with the people about what type of problem they are facing. They were very happy to take this system as it gives them some relief in the modern age.

Despite everything we achieved, we faced many challenges to finish this project. After all it's an online web-based system so in real life both doctor and patient need to follow the using rules otherwise its goal will be failed.

6.2 Limitations of the project

- Video call not available with the doctor
- No appointment verification system
- No medical certificate given

6.3 Future Works

Online system is always a changeable system. It develops day by day, getting better and better and easier for people. This could be a revolutionary web application that may help bonding between doctor and patient. We believe we can make this system more advanced in future. Advance features and User interface will be updated in future. Our system is already user friendly but we will try to make this system more users friendly in future

REFERENCES

- [1] Scott, Ian A., et al. "Using EMR-enabled computerized decision support systems to reduce prescribing of potentially inappropriate medications: a narrative review." *Therapeutic advances in drug safety* 9.9 (2018): 559-573.
- [2] Gerstle, Robert S., Christoph U. Lehmann, and Council on Clinical Information Technology. "Electronic prescribing systems in pediatrics: the rationale and functionality requirements." *Pediatrics* 119.6 (2007): e1413-e1422.
- [3] Stackoverflow. (n.d.). Documentation. <https://stackoverflow.com>
- [4] Firebase. (n.d.). Documentation. <https://firebase.google.com>
- [5] Medium. (n.d.). Blog. <https://medium.com>
- [6] MDN. (n.d.). Tutorial with Examples. <https://developer.mozilla.org/en-US>
- [7] React Router DOM. (n.d.). Tutorial. <https://reactrouter.com/en/main>
- [8] AXIOS. (n.d.). Tutorial. <https://axios-http.com/docs/intro>
- [9] Tutorial Republic. (n.d.). Live Examples. <https://www.tutorialrepublic.com/sql-tutorial>
- [10] Murach's PHP & MySQL – by Joel Murach & Ray Harris.
- [11] C in a Nutshell: The Definitive Reference Peter Prinz, 2015.

APPENDIX

Appendix A Project Reflection

In our daily life we face a lot of problems. Disease is one of the most common issues for a person's life. If anybody is ill and wants to visit a doctor for checkup, he or she needs to visit the hospital and wait until the doctor is available. When people get affected by illness they need to visit a doctor for checkup but they have to visit their chambers or hospital to get an appointment. It is a lengthy process and wasting people's time. Sometimes people do visit the doctor's chamber for a health check but the doctor is not available for various reasons. It's the only way to get to know when people just visited their places. It harasses people a lot. Besides, people need an ambulance service to carry patients to hospitals. Merely, people need to visit hospitals or clinics to hire ambulances, it is a time consuming process. Day by day the healthcare system becomes an important part of our society. So we have decided to build this system.

API

Appointment data

```
<?php

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");

include '../config/Config.php';
include '../classes/appointmentData.php';

$ClassAppointment = new Appointment();

$sapi = $_SERVER['REQUEST_METHOD'];

$appointment_info = json_decode(file_get_contents("php://input"), true);
$appointment_id = intval($_GET['id'] ?? "");

// update appointment status after saved the prescription
$appointment_status_update_id = intval($_GET['appointment_status_update_id'] ?? "");

// Insert data using post method
if ($sapi == 'POST') {
    $result = $ClassAppointment->insert($appointment_info);
    if ($result) {
```

```

                echo json_encode(['data'=>$result,'message'=>'Data fetch success',
'status'=>true]);
            } else {
                echo json_encode(['message'=>'Data fetch success', 'status'=>false]);
            }
        }

if($appointment_id){
    if($api == "GET"){
        $data = $ClassAppointment->fetchSingle($appointment_id);
        if($data){
            echo json_encode(['data'=>$data,'message'=>'Data fetch success',
'status'=>true]);
        }else{
            echo $ClassAppointment->message('Failed to fetch data!',false);
        }
    }
}

}else{
    if($api == "GET"){
        $data = $ClassAppointment->fetch();
        if($data){
            echo json_encode($data);

            } else {
                echo $ClassAppointment->message('Failed to fetch data!',false);
            }
        }
    }
}

if($appointment_status_update_id){
    if($api == "GET"){
        $result = $ClassAppointment-
>statusUpdate($appointment_status_update_id);
        if($result){
            echo json_encode(['message'=>'update success', 'status'=>true]);
        }
    }
}
}
}

```

Brand Search

```
<?php
```

```

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
X-Requested-With");

```

```

include '../config/Config.php';
include '../classes/instruction.php';

$ClassInstruction = new Instruction();

$api = $_SERVER['REQUEST_METHOD'];

$instruction_info = json_decode(file_get_contents("php://input"), true);
$instruction_id = intval($_GET['id'] ?? "");

if($instruction_id){
    if($api == "GET"){
        $data = $ClassInstruction->fetchSingle($instruction_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $ClassInstruction->message('Failed to fetch data!',false);
        }
    }
}

}else{
    if($api == "GET"){
        $data = $ClassInstruction->fetch();
        if($data){
            echo json_encode($data);
        } else {
            echo $ClassInstruction->message('Failed to fetch data!',false);
        }
    }
}
}

```

Dynamic_Clinical_Options_CRUD

```

<?php

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");

include '../config/Config.php';
include '../classes/DynamicClinicalOptionsCRUD.php';

$DynamicClinicalOptionsCRUD = new DynamicClinicalOptionsCRUD();

// received http request

```

```

$api = $_SERVER['REQUEST_METHOD'];

$body_info = json_decode(file_get_contents("php://input"), true);

// for dynamic search
$search_table = $_GET['table'] ?? "";
$search_column = $_GET['column'] ?? "";
$search_value = $_GET['value'] ?? "";

// get all data
$getAllData = $_GET['getAllData'] ?? "";

// get single data
$getSingleDataTable = $_GET['getSingleTab'] ?? "";
$getSingleDataId = intval($_GET['singleId'] ?? "");

// Delete data
$delTable = $_GET['delTable'] ?? "";
$delId = intval($_GET['delId'] ?? "");
//update data
$updateTable = $_GET['updateTable'] ?? "";
$updateColumn = $_GET['updateColumn'] ?? "";
$updateId = intval($_GET['updateId'] ?? "");
$updateVal = $_GET['value'] ?? "";

if($search_table && $search_column && $search_value){
    if($api == "GET"){
        $data = $DynamicClinicalOptionsCRUD->searchSingle($search_table,
$search_column, $search_value);
        if($data){
            echo json_encode($data);
        }else{
            echo $DynamicClinicalOptionsCRUD->message('Failed to fetch
data!',false);
        }
    }
}

// dynamically data insert
if($api == "POST"){
    if($body_info){
        $result = $DynamicClinicalOptionsCRUD->insert($body_info);
        if ($result) {
            echo json_encode(['message'=>"Data inserted", 'status'=> true,
'data'=>[$result]]);
            // echo $DynamicClinicalOptionsCRUD->message('Data Inserted',true);
        }
    }
}

```

```

        } else {
            echo json_encode(['message'=>"Data not inserted", 'status'=> false,
'data'=>[$result]]);
        }
    }
}

// Get All Data
if($api == "GET"){
    if($getAllData){
        $data = $DynamicClinicalOptionsCRUD->fetch($getAllData);
        if($data){
            echo json_encode($data);
        } else {
            echo $DynamicClinicalOptionsCRUD->message('Failed to fetch data!',false);
        }
    }
}

// get Single Data
if($getSingleDataId && $getSingleDataTable){
    if($api == "GET"){
        $data = $DynamicClinicalOptionsCRUD->fetchSingle($getSingleDataId,
$getSingleDataTable);
        if($data){
            echo json_encode($data);
        }else{
            echo $DynamicClinicalOptionsCRUD->message('Failed to fetch
data!',false);
        }
    }
}

// Delete item
if($api == "DELETE"){
    if($delTable && $delId){
        $result = $DynamicClinicalOptionsCRUD->delete($delTable, $delId);
        if ($result) {
            echo json_encode(['message'=>"Data deleted", 'status'=> true,
'data'=>[$result]]);
            // echo $DynamicClinicalOptionsCRUD->message('Data
Inserted',true);
        } else {
            echo json_encode(['message'=>"Data not deleted",
'status'=> false, 'data'=>[$result]]);
        }
    }
}
}

```



```

if($api == "PUT"){
    if($updateTable && $updateColumn && $updateId && $updateVal){

        $result = $DynamicClinicalOptionsCRUD->update($updateTable,
        $updateColumn, $updateId, $updateVal);
        if ($result) {
            echo json_encode(['message'=>"Data updated", 'status'=> true]);
            // echo $DynamicClinicalOptionsCRUD->message('Data
Inserted',true);
        } else {
            echo json_encode(['message'=>"Data not updated",
'status'=> false, ]);
        }
    }
}

```

Patient data

```
<?php
```

```

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
X-Requested-With");

```

```

// Include action.php file
include_once '../classes/patientData.php';

```

```

// Create object of
$classPatient = new PatientData();

```

```

// create a api variable to get HTTP method dynamically
$api = $_SERVER['REQUEST_METHOD'];
$data = json_decode(file_get_contents("php://input"), true);

```

```

// get id from url
$patient_id = intval($_GET['id'] ?? "");

```

```

$patientLastData;
if ($api == 'POST') {
    $result = $classPatient->insert($data);
    if ($result) {
        echo json_encode(['data'=>$result, 'message'=>'Appointment success',
'status'=>true]);
    } else {

```

```

        echo $ClassPatient->message('Failed to add an user!',false);
    }
}

if($patient_id){
    if($api == "GET"){
        $data = $ClassPatient->fetchSingle($patient_id);
        if($data){
            echo json_encode(['data'=>$data, 'message'=>'Fetch success',
'status'=>true]);
        }else{
            echo $ClassPatient->message('Failed to fetch data!',false);
        }
    }
} else{
    if($api == "GET"){
        $data = $ClassPatient->fetch();
        if($data){
            echo json_encode($data);
        } else {
            echo $ClassPatient->message('Failed to fetch data!',false);
        }
    }
}
?>

```

Prescription data

```

<?php

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
X-Requested-With");

// Include action.php file
include_once '../classes/prescriptionData.php';
// Create object of
$ClassPrescription = new Prescription();

// create a api variable to get HTTP method dynamically
$api = $_SERVER['REQUEST_METHOD'];

$Prescription_info = json_decode(file_get_contents("php://input"), true);

```

```

// get prescription id from url for print prescription
$Prescription_id = intval($_GET['id'] ?? "");

$authID = $_GET['authID'] ?? "";
// referral patient get
$refID = $_GET['refID'] ?? "";

// Insert prescription basic info
if ($api == 'POST') {
    if($Prescription_info){
        $result = $ClassPrescription->insert($Prescription_info);
        if ($result) {
            echo json_encode([ "data"=>$result,
"message"=>"Prescription Inserted", "status" => true ]);
        } else {
            echo json_encode(["message"=>"Prescription not
inserted", "status"=>false]);
        }
    }
}

// get single prescription data by id
// basically using for prescription print
if($Prescription_id){
    if($api == "GET"){
        $data = $ClassPrescription->fetchSingle($Prescription_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $ClassPrescription->message('Failed to fetch
data!',false);
        }
    }
}

// get all prescription data
if($api == "GET"){

    if($authID){
        $data = $ClassPrescription->fetch($authID);
        if($data){
            echo json_encode($data);
        } else {
            echo $ClassPrescription->message('Failed to
fetch data!',false);
        }
    }
}

```

```

        }

    }

if($api == "GET"){
    if($refID){
        $data = $ClassPrescription->fetchReferralPatient($refID);
        if($data){
            echo json_encode($data);

            } else {
                echo $ClassPrescription->message('Failed to fetch data!',false);
            }
        }
    }
}

```

?>

Prescription_data_get

<?php

// NOT USED API

// Replace api Prescription_data.php

header('Access-Control-Allow-Origin: *');

header('Content-Type: application/json');

header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");

header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");

include '../config/Config.php';

// create a api variable to get HTTP method dynamically

\$api = \$_SERVER['REQUEST_METHOD'];

// get id from url

\$id = intval(\$_GET['id'] ?? "");

if (\$api == "GET") {

// Check prescription id is valid or invalid

\$sql = "SELECT * FROM prescription_information WHERE prescription_id='{ \$id }'";

\$prescriptionResult = mysqli_query(\$conn, \$sql) or die("Data fetch fail");

if (mysqli_num_rows(\$prescriptionResult) > 0) {

if (isset(\$id)) {

// Get Chief Complain Data

```

        $sql = "SELECT * FROM prescription_chief_complain WHERE
prescription_id='{ $id}";
        $chiefComplainResult = mysqli_query($conn, $sql) or die("Data fetch fail");
        if (mysqli_num_rows($chiefComplainResult) > 0) {
            $chiefComplainOutput = mysqli_fetch_all($chiefComplainResult,
MYSQLI_ASSOC);
        }

        // Get On Examination Data
        $sql = "SELECT * FROM prescription_on_examination WHERE
prescription_id='{ $id}";
        $onExaminationResult = mysqli_query($conn, $sql) or die("Data fetch fail");
        if (mysqli_num_rows($onExaminationResult) > 0) {
            $onExaminationOutput = mysqli_fetch_all($onExaminationResult,
MYSQLI_ASSOC);
        }

        // Get Investigation advice data
        $sql = "SELECT * FROM prescription_investigation_advice WHERE
prescription_id='{ $id}";
        $investigationAdviceResult = mysqli_query($conn, $sql) or die("Data fetch fail");
        if (mysqli_num_rows($investigationAdviceResult) > 0) {
            $investigationAdviceOutput = mysqli_fetch_all($investigationAdviceResult,
MYSQLI_ASSOC);
        }

        // Get Patient Advice data
        $sql = "SELECT * FROM prescription_advice WHERE prescription_id='{ $id}";
        $adviceResult = mysqli_query($conn, $sql) or die("Data fetch fail");
        if (mysqli_num_rows($adviceResult) > 0) {
            $adviceOutput = mysqli_fetch_all($adviceResult, MYSQLI_ASSOC);
        }

        // Get Next visit data
        $sql = "SELECT * FROM prescription_patient_next_visit WHERE
prescription_id='{ $id}";
        $nextVisitResult = mysqli_query($conn, $sql) or die("Data fetch fail");
        if (mysqli_num_rows($nextVisitResult) > 0) {
            $nextVisitOutput = mysqli_fetch_all($nextVisitResult, MYSQLI_ASSOC);
        }

        // Get prescription medicine
        $sql = "SELECT * FROM prescription_medicine WHERE
prescription_id='{ $id}";
        $prescriptionMedicineResult = mysqli_query($conn, $sql) or die("Data fetch
fail");
        if (mysqli_num_rows($prescriptionMedicineResult) > 0) {
            $prescriptionMedicineOutput = mysqli_fetch_all($prescriptionMedicineResult,
MYSQLI_ASSOC);
        }

```

```

        // create multidimensional array
        $output = [
            "ChiefComplainData"    => $chiefComplainOutput ? $chiefComplainOutput
: null,
            "OnExaminationData"    => $onExaminationOutput ?
$onExaminationOutput : null,
            "InvestigationAdviceData"    => $investigationAdviceOutput ?
$investigationAdviceOutput : null,
            "adviceData"            => $adviceOutput ? $adviceOutput : null,
            "nextVisitData"         => $nextVisitOutput ? $nextVisitOutput : null,
            "prescriptionMedicineData"  => $prescriptionMedicineOutput ?
$prescriptionMedicineOutput : null,
        ];

        // convert to json data
        echo json_encode($output);
    }
} else {
    echo json_encode(array('message' => "Prescription id invalid", "status" => false));
}
}
}

```

Prescription_data_insert

```
<?php
```

```

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
X-Requested-With");

```

```
include '../config/Config.php';
```

```
// create a api variable to get HTTP method dynamically
```

```
$sapi = $_SERVER['REQUEST_METHOD'];
```

```
$data = json_decode(file_get_contents("php://input"), true);
```

```
if ($sapi == "POST") {
```

```
    $prescription_id = $data['prescription_id'];
```

```
    $name = $data['name'];
```

```
    $created_by = $data['created_by'];
```

```
    if (count($data) > 0) {
```

```

    $sql = "INSERT INTO prescription_chief_complain (prescription_id, name,
created_by)VALUES({'$prescription_id'}, {'$name'}, {'$created_by'})";
    $chiefComplainInserted = mysqli_query($conn, $sql) or die("Data fetch fail");
    if ($chiefComplainInserted) {
        echo json_encode(array('message' => "Inserted", "status" => true));
    } else {
        echo json_encode(array('message' => "Not inserted", "status" => false));
    }
}
}
}

```

Prescription list

<?php

```

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
X-Requested-With");

```

```

// Include action.php file
include_once '../classes/prescription_list.php';
// Create object of
$classPrescription = new PrescriptionList();

// create a api variable to get HTTP method dynamically
$sapi = $_SERVER['REQUEST_METHOD'];

// get prescription id from url for print prescription
$patientID = intval($_GET['id'] ?? "");

```

```

$authID = $_GET['authID'] ?? "";

```

```

// get single prescription data by id
// basically using for prescription print
// get all prescription data

```

```

if($patientID){
    if($sapi == "GET"){
        $data = $classPrescription->fetchSingle($authID,$patientID);
        if($data){
            echo json_encode($data);
        }else{

```

```

        echo $ClassPrescription->message('Failed to fetch
data!',false);
    }
}
}else{
    if($authID){
        if($api == "GET"){
            $data = $ClassPrescription->fetch($authID);
            if($data){
                echo json_encode($data);
            } else {
                echo $ClassPrescription->message('Failed to fetch
data!',false);
            }
        }
    }
}
?>

```

Prescription_Multiple_Item_Insert

```

<?php

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
X-Requested-With");

include '../config/Config.php';
include '../classes/prescriptionMultipleItemInsert.php';

$ClassPrescriptionMultipleItemInsert = new PrescriptionMultipleItemInsert();

$api = $_SERVER['REQUEST_METHOD'];

$data_info = json_decode(file_get_contents("php://input"), true);
$data_id = intval($_GET['id'] ?? "");

$prescription_id = $data_info['prescriptionID'];
$chiefComplain = $data_info["chiefComplain"];
$investigationAdvice = $data_info["investigationAdvice"];

```



```

$onExamination = $data_info["onExamination"];
$prescriptionMedicine = $data_info["prescriptionMedicine"];
$patientAdvice = $data_info["patientAdvice"];

// chief complain insert
if ($api == 'POST') {
    if($prescription_id){
        $result = $ClassPrescriptionMultipleItemInsert->insertChiefComplain($chiefComplain,$prescription_id);
        if ($result ==true) {
            echo json_encode([ "data"=>$result,
"message"=>"Prescription chief complain inserted", "status" => true ]);
        } else {
            echo json_encode(["message"=>"Prescription chief complain not inserted", "status" => false ]);
        }
    }
}

// On Examination insert
if ($api == 'POST') {
    if($prescription_id){
        $result = $ClassPrescriptionMultipleItemInsert->insertOnExamination($onExamination,$prescription_id);
        if ($result ==true) {
            echo json_encode([ "data"=>$result, "message"=>"Prescription on examination inserted", "status" => true ]);
        } else {
            echo json_encode(["message"=>"Prescription on examination not inserted", "status" => false ]);
        }
    }
}

// Investigation advice
if ($api == 'POST') {
    if($prescription_id){
        $result = $ClassPrescriptionMultipleItemInsert->insertInvestigationAdvice($investigationAdvice,$prescription_id);
        if ($result ==true) {
            echo json_encode([ "data"=>$result, "message"=>"Prescription investigation advice inserted", "status" => true ]);
        } else {
            echo json_encode(["message"=>"Prescription investigation advice not inserted", "status" => true ]);
        }
    }
}

```

```

}
// prescription medicine
if ($api == 'POST') {
    if($prescription_id){
        $result = $ClassPrescriptionMultipleItemInsert->insertPrescriptionMedicine($prescriptionMedicine,$prescription_id);
        if ($result ==true) {
            echo json_encode([ "data"=>$result, "message"=>"Prescription
medicine inserted", "status" => true ]);
        } else {
            echo json_encode(["message"=>"Prescription medicine not
inserted", "status" => false ]);
        }
    }
}

// Patient advice
if ($api == 'POST') {
    if($prescription_id){
        $result = $ClassPrescriptionMultipleItemInsert->insertPatientAdvice($patientAdvice,$prescription_id);
        if ($result ==true) {
            echo json_encode([ "data"=>$result, "message"=>"Prescription
patient advice inserted", "status" => true ]);
        } else {
            echo json_encode(["message"=>"Prescription patient advice not
inserted", "status" => false ]);
        }
    }
}

// bookmark drug brands
if($api == "POST"){
    if($prescription_id){
        $result = $ClassPrescriptionMultipleItemInsert->insertBookmarkDrugBrands($prescriptionMedicine,$prescription_id);
        if ($result ==true) {
            echo json_encode([ "data"=>$result, "message"=>"Prescription
medicine inserted", "status" => true ]);
        } else {
            echo json_encode(["message"=>"Prescription medicine not
inserted", "status" => false ]);
        }
    }
}
}

```

Referral_patient_list

```
<?php
```

```
header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");
```

```
// Include action.php file
include_once './classes/referral_patient_list.php';
// Create object of
$classReferral = new ReferralPatientList();
```

```
// create a api variable to get HTTP method dynamically
$sapi = $_SERVER['REQUEST_METHOD'];
```

```
// get prescription id from url for print prescription
$patient_id = intval($_GET['id'] ?? "");
```

```
$authID = $_GET['authID'] ?? "";
```

```
// get single prescription data by id
// basically using for prescription print
if($patient_id){
    if($sapi == "GET"){
        $data = $classReferral->fetchSingle($authID,$patient_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $classReferral->message('Failed to fetch data!',false);
        }
    }
}
else{
    if($authID){
        if($sapi == "GET"){
            $data = $classReferral->fetch($authID);
            if($data){
                echo json_encode($data);
            } else {
                echo $classReferral->message('Failed to fetch
data!',false);
            }
        }
    }
}
```

```
}  
}
```

```
?>
```

Today_Appointment_data

```
<?php  
header('Access-Control-Allow-Origin: *');  
header('Access-Control-Allow-Methods: GET, POST, PUT, DELETE');  
header('Access-Control-Allow-Headers: X-Requested-With');  
header('Content-Type: application/json');  
  
include '../config/Config.php';  
include '../classes/todayAppointmentList.php';  
  
$ClassTodayAppointmentList = new TodayAppointment();  
  
$sapi = $_SERVER['REQUEST_METHOD'];  
  
$todayAppointmentList_info = json_decode(file_get_contents("php://input"), true);  
$onExamination_id = intval($_GET['id'] ?? "");  
$todayAppointmentList_id = $_GET['date'] ?? "";  
$authID = $_GET['authID'] ?? "";  
  
if($todayAppointmentList_id && $authID){  
    if($sapi == "GET"){  
        $data = $ClassTodayAppointmentList-  
>fetchSingle($todayAppointmentList_id, $authID);  
        if($data){  
            echo json_encode($data);  
        }else{  
            echo $ClassTodayAppointmentList->message('Failed to fetch  
data!',false);  
        }  
    }  
}  
  
} else{  
    if($sapi == "GET"){  
        if($authID){  
            $data = $ClassTodayAppointmentList->fetch($authID);  
            if($data){  
                echo json_encode($data);  
            } else {  

```

```

        echo $ClassTodayAppointmentList->message('Failed to fetch
data!',false);
    }
}
}
}

```

User_Profile

```

<?php

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
X-Requested-With");

// Include action.php file
include_once '../classes/UserProfile.php';
// Create object of
$classUserProfile = new UserProfile();

// create a api variable to get HTTP method dynamically
$api = $_SERVER['REQUEST_METHOD'];
$userProfile_info = json_decode(file_get_contents("php://input"), true);

// get id from url for get single user
$userProfile_id = strval(($_GET['id'] ?? ""));

if ($api == 'POST') {
    if ($userProfile_info) {
        $result = $classUserProfile->insert($userProfile_info);
        if($result ==true){
            echo $classUserProfile->message('User added successfully!',true);
        } else {
            echo $classUserProfile->message('Failed to add an user!',false);
        }
    }
}

if($userProfile_id){
    if($api == "GET"){
        $result = $classUserProfile->fetchSingle($userProfile_id);
        if($result ==true){
            echo
json_encode(['data'=>$result,'message'=>'Success','status'=>true]);
        }else{

```

```

                echo
json_encode(['data'=>$result,'message'=>'Success','status'=>false]);
            }
        }
    }else{
        if($api == "GET"){
            $data = $ClassUserProfile->fetch();
            if($data){
                echo json_encode($data);

                } else {
                    echo $ClassUserProfile->message('Failed to fetch data!',false);
                }
            }
        }
    }
?>

```

X_advice

```

<?php

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");

include '../config/Config.php';
include '../classes/advice.php';

$ClassAdvice = new Advice();

$api = $_SERVER['REQUEST_METHOD'];

$advice_info = json_decode(file_get_contents("php://input"), true);
$advice_id = intval($_GET['id'] ?? "");

if($advice_id){
    if($api == "GET"){
        $data = $ClassAdvice->fetchSingle($advice_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $ClassAdvice->message('Failed to fetch data!',false);
        }
    }
}

```

```

}else{
    if($api == "GET"){
        $data = $ClassAdvice->fetch();
        if($data){
            echo json_encode($data);

            } else {
                echo $ClassAdvice->message('Failed to fetch data!',false);
            }
        }
    }
}

```

X_Chief_complain

```
<?php
```

```

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");

```

```

include '../config/Config.php';
include '../classes/ChiefComplain.php';

```

```
$ClassChiefComplain = new ChiefComplain();
```

```
$api = $_SERVER['REQUEST_METHOD'];
```

```

$ChiefComplain_info = json_decode(file_get_contents("php://input"), true);
$ChiefComplain_id = intval($_GET['id'] ?? "");

```

```

// For Edit
$ChiefComplain_edit = intval($_GET['edit'] ?? "");

```

```

// for search
$ChiefComplain_name = $_GET['name'] ?? "";

```

```

if($ChiefComplain_id){
    if($api == "GET"){
        $data = $ClassChiefComplain->fetchSingle($ChiefComplain_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $ClassChiefComplain->message('Failed to fetch data!',false);
        }
    }
}

```

```

}else if($ChiefComplain_name){
    if($ChiefComplain_name){
        if($api == "GET"){
            $data = $ClassChiefComplain-
>search($ChiefComplain_name);
            if($data){
                echo json_encode($data);
            }else{
                echo $ClassChiefComplain->message('Failed to fetch
data!',false);
            }
        }
    }
}
}else{
    if($api == "GET"){
        $data = $ClassChiefComplain->fetch();
        if($data){
            echo json_encode($data);
        } else {
            echo $ClassChiefComplain->message('Failed to fetch
data!',false);
        }
    }
}

if($api == "POST"){
    $data = $ClassChiefComplain->insert($ChiefComplain_info);
    if($data ==true){
        echo $ClassChiefComplain->message('Data Inserted!',true);
    }else{
        echo $ClassChiefComplain->message('Data Not Inserted!',false);
    }
}

if($api == "PUT"){
    if($ChiefComplain_edit){
        $data = $ClassChiefComplain-
>updateFav($ChiefComplain_info,$ChiefComplain_edit);
        if($data ==true){
            echo $ClassChiefComplain->message('Added to favorite list!',true);
        }else{
            echo $ClassChiefComplain->message('Sorry not added to favorite
list',false);
        }
    }
}
}

```


X_dose

```
<?php

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers,
X-Requested-With");

include '../config/Config.php';
include '../classes/instruction.php';

$classInstruction = new Instruction();

$sapi = $_SERVER['REQUEST_METHOD'];

$instruction_info = json_decode(file_get_contents("php://input"), true);
$instruction_id = intval($_GET['id'] ?? "");

if($instruction_id){
    if($sapi == "GET"){
        $data = $classInstruction->fetchSingle($instruction_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $classInstruction->message('Failed to fetch data!',false);
        }
    }
}

}else{
    if($sapi == "GET"){
        $data = $classInstruction->fetch();
        if($data){
            echo json_encode($data);
        } else {
            echo $classInstruction->message('Failed to fetch data!',false);
        }
    }
}
}
```

X_drug_brands

```
<?php
```

```

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");

include '../config/Config.php';
include '../classes/instruction.php';

$ClassInstruction = new Instruction();

$api = $_SERVER['REQUEST_METHOD'];

$instruction_info = json_decode(file_get_contents("php://input"), true);
$instruction_id = intval($_GET['id'] ?? "");

if($instruction_id){
    if($api == "GET"){
        $data = $ClassInstruction->fetchSingle($instruction_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $ClassInstruction->message('Failed to fetch data!',false);
        }
    }
}

}else{
    if($api == "GET"){
        $data = $ClassInstruction->fetch();
        if($data){
            echo json_encode($data);
        } else {
            echo $ClassInstruction->message('Failed to fetch data!',false);
        }
    }
}
}

```

X_duration

```
<?php
```

```

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");

```

```

include '../config/Config.php';
include '../classes/instruction.php';

$classInstruction = new Instruction();

$sapi = $_SERVER['REQUEST_METHOD'];

$instruction_info = json_decode(file_get_contents("php://input"), true);
$instruction_id = intval($_GET['id'] ?? "");

if($instruction_id){
    if($sapi == "GET"){
        $data = $classInstruction->fetchSingle($instruction_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $classInstruction->message('Failed to fetch data!',false);
        }
    }
}

}else{
    if($sapi == "GET"){
        $data = $classInstruction->fetch();
        if($data){
            echo json_encode($data);
        } else {
            echo $classInstruction->message('Failed to fetch data!',false);
        }
    }
}
}

```

X_instruction

```

<?php

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");

include '../config/Config.php';
include '../classes/instruction.php';

$classInstruction = new Instruction();

$sapi = $_SERVER['REQUEST_METHOD'];

```

```

$instruction_info = json_decode(file_get_contents("php://input"), true);
$instruction_id = intval($_GET['id'] ?? "");

if($instruction_id){
    if($api == "GET"){
        $data = $ClassInstruction->fetchSingle($instruction_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $ClassInstruction->message('Failed to fetch data!',false);
        }
    }
}

}else{
    if($api == "GET"){
        $data = $ClassInstruction->fetch();
        if($data){
            echo json_encode($data);
        } else {
            echo $ClassInstruction->message('Failed to fetch data!',false);
        }
    }
}
}

```

X_Investigation

```
<?php
```

```

header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");

```

```

include '../config/Config.php';
include '../classes/investigation.php';

```

```
$ClassInvestigation = new Investigation();
```

```
$api = $_SERVER['REQUEST_METHOD'];
```

```

$investigation_info = json_decode(file_get_contents("php://input"), true);
$investigation_id = intval($_GET['id'] ?? "");

```

```

if($investigation_id){
    if($api == "GET"){

```

```

        $data = $ClassInvestigation->fetchSingle($investigation_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $ClassInvestigation->message('Failed to fetch data!',false);
        }
    }
}
}else{
    if($api == "GET"){
        $data = $ClassInvestigation->fetch();
        if($data){
            echo json_encode($data);

            } else {
                echo $ClassInvestigation->message('Failed to fetch data!',false);
            }
        }
    }
}
}

```

X_onexamination

```
<?php
```

```
header('Access-Control-Allow-Origin: *');
header('Content-Type: application/json');
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, X-Requested-With");
```

```
include '../config/Config.php';
include '../classes/onExamination.php';
```

```
$ClassOnExamination = new onExamination();
```

```
$api = $_SERVER['REQUEST_METHOD'];
```

```
$onExamination_info = json_decode(file_get_contents("php://input"), true);
$onExamination_id = intval($_GET['id'] ?? "");
```

```
if($onExamination_id){
    if($api == "GET"){
        $data = $ClassOnExamination->fetchSingle($onExamination_id);
        if($data){
            echo json_encode($data);
        }else{
            echo $ClassOnExamination->message('Failed to fetch data!',false);
        }
    }
}

```

```

}else{
    if($api == "GET"){
        $data = $ClassOnExamination->fetch();
        if($data){
            echo json_encode($data);

            } else {
                echo $ClassOnExamination->message('Failed to fetch
data!',false);
            }
        }
    }
}

```

Configurations

```

<?php
// $conn = mysqli_connect("localhost", "root", "", "su_smart_rx") or die("Connection
failed");

```

```

class Config {
    // Database Details
    private const DBHOST = 'localhost';
    private const DBUSER = 'root';
    private const DBPASS = "";
    private const DBNAME = 'su_smart_rx';
    // Data Source Network
    private $dsn = 'mysql:host=' . self::DBHOST .
';dbname=' . self::DBNAME . ";

    // conn variable
    protected $conn = null;

    // Constructor Function
    public function __construct() {
        try {
            $this->conn = new PDO($this->dsn,
self::DBUSER, self::DBPASS);

            $this->conn-
>setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
        } catch (PDOException $e) {
            die('Connectionn Failed : ' . $e-
>getMessage());
        }
        return $this->conn;
    }
}

```

```

    }

    // Sanitize Inputs
    public function test_input($data) {
        $data = strip_tags($data);
        $data = htmlspecialchars($data);
        $data = stripslashes($data);
        $data = trim($data);
        return $data;
    }

    // JSON Format Converter Function
    public function message($content, $status) {
        return json_encode(['message' => $content,
'error' => $status]);
    }
}

?>

```

Database

```

<?php
    // Include config.php file
    include_once 'config.php';

    // Create a class Users
    class Database extends Config {

        // Fetch all or a single user from database
        public function fetch($id = 0) {
            $sql = 'SELECT * FROM users';
            if ($id != 0) {
                $sql .= ' WHERE id = :id';
            }
            $stmt = $this->conn->prepare($sql);
            $stmt->execute(['id' => $id]);
            $rows = $stmt->fetchAll();
            return $rows;
        }

        // Insert an user in the database
        public function insert($data) {
            $patient_name = $data['patient_name'];
            $patient_phone = $data['patient_phone'];
            $patient_email = $data['patient_email'];
            $patient_blood = $data['patient_blood'];
            $patient_address = $data['patient_address'];
            $created_by = $data['created_by'];
            $patient_district = $data['patient_district'];
            $suuid = $data['uuid'];

```

```

    $sql = 'INSERT INTO patient_info
        (patient_name,patient_phone,patient_email,patient_blood,patient_address,created
        _by,patient_district,uuid)
        VALUES
        (:patient_name,:patient_phone,:patient_email,:patient_blood,:patient_address,:created_by,
        :patient_district,:uuid)';
    $stmt = $this->conn->prepare($sql);
    $stmt->execute([
        'patient_name' => $patient_name,
        'patient_phone' => $patient_phone,
        'patient_email' => $patient_email,
        'patient_blood' => $patient_blood,
        'patient_address' => $patient_address,
        'created_by' => $created_by,
        'patient_district' => $patient_district,
        'uuid' => $uuid
    ]);
    return true;
}

// Update an user in the database
public function update($name, $email, $phone, $id) {
    $sql = 'UPDATE users SET name = :name, email = :email, phone = :phone
    WHERE id = :id';
    $stmt = $this->conn->prepare($sql);
    $stmt->execute(['name' => $name, 'email' => $email, 'phone' => $phone, 'id' =>
    $id]);
    return true;
}

// Delete an user from database
public function delete($id) {
    $sql = 'DELETE FROM users WHERE id = :id';
    $stmt = $this->conn->prepare($sql);
    $stmt->execute(['id' => $id]);
    return true;
}
}
?>

```