**MEDS HELPING HANDS**

**SUBMITTED BY**

1. MUHAMMAD HAROON (SP21-BSCS-0067)

2. SYED WAJAHAT SHAH (SP21-BSCS-0025)

3. AHSAN ALI BHELLAR (SP21-BSCS-0013)

**SUPERVISED BY**

MISS ALEEZA RIAZ



FINAL YEAR PROJECT REPORT SUBMITTED TO THE FACULTY OF COMPUTING, MOHAMMAD ALI JINNAH UNIVERSITY, IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF COMPUTER SCIENCE/SOFTWARE ENGINEERING

SPRING 2025

**MEDS HELPING HANDS**

**Batch – 2025**



DEPARTMENT OF COMPUTER SCIENCE

Mohammad Ali Jinnah University. Karachi

September 2024

**Acknowledgment**Shape

First and foremost, we express our gratitude to the ALLAH ALMIGHTY for bestowing upon us the strength, courage, and intellectual capacity to engage in meaningful work. Additionally, we extend our heartfelt appreciation to our project supervisor, Miss Aleeza. We sincerely thank her for her continuous support, motivation, and guidance throughout the project.

Our profound gratitude is also extended to the members of the Final Year Project (FYP) Committee for their cooperative assistance during the culmination of our academic journey. We would like to acknowledge the invaluable contribution of all our teachers, who, knowingly or unknowingly, have played a vital role in shaping our studies and projects over the past four years, and continue to guide us.

Lastly, we extend our thanks to the entire departmental and university staff for serving as role models and providing assistance whenever we encountered challenges.

Shape

**Introduction to group members**



MUHAMMAD HAROON SP21-BSCS-0067

(Frontend, Backend of Project)

Immediate Contact: (sp21bscs0067@maju.edu.pk, 03341836101 )



SYED WAJAHAT SHAH SP21-BSCS-0025   
(Frontend, Backend and Documentation of Project)

Immediate Contact: (sp21bscs0025@maju.edu.pk , 03162361771)



AHSAN ALI BHELLAR SP21-BSCS-0013   
(Documentation)

Immediate Contact: (sp21bscs0013@maju.edu.pk , 03160157088)

**CERTIFICATE OF COMPLETION**

**This is to certify that the following students of BS (CS) of Batch 2021**

MUHAMMAD HAROON SP21-BSCS-0067

SYED WAJAHAT SHAH SP21-BSCS-0025

AHSAN ALI BHELLAR SP21-BSCS-0013

**have successfully completed their final year project titled**

**MEDS HELPING HANDS**

**in the partial fulfillment for the Degree of Bachelor of Science in Computer Science.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Miss Aleeza Riaz**

Mohammad Ali Jinnah University

**ABSTRACT**

The Meds Helping Hands project developed a comprehensive web portal aimed at simplifying and streamlining the process of medicine donation. This innovative platform effectively connects donors who wish to contribute their unused medicines with recipients in need, facilitating a direct and efficient exchange. In addition, the application provides robust administrative tools, allowing administrators to oversee and manage the entire donation lifecycle with ease and precision.

This medication donation portal is about the assortment of medication that is unused by the patient who recuperates totally and remaining prescriptions become squander, those medications can be gathered and utilized further. The aim and object of this project is to prepare an Online Portal for the collection of unused medicines so that they can be given to the people who are in need. What this project will do is list the unused medicines from the people who have completely recovered from the illness or do not require the medicines anymore. To ensure user security and data integrity, the application incorporates a secure login system and employs Redux for state management, which enhances the responsiveness and performance of the application by efficiently managing the app's data.

The design and features of the portal are tailored to accommodate the needs of various user groups, including donors and administrators, enabling them to perform their respective tasks with enhanced speed and less complexity. The user testing phase demonstrated that individuals could navigate the portal effortlessly, perform required actions without confusion, and achieve their objectives without encountering significant issues.

Overall, the Meds Helping Hands app does more than just simplify the medicine donation process. It significantly enhances the efficiency and user experience of medicine donation. By reducing the time and effort required to donate and receive medicines, the portal makes the process more accessible and manageable for all parties involved, so the impact of each donation contributes to greater health care access.

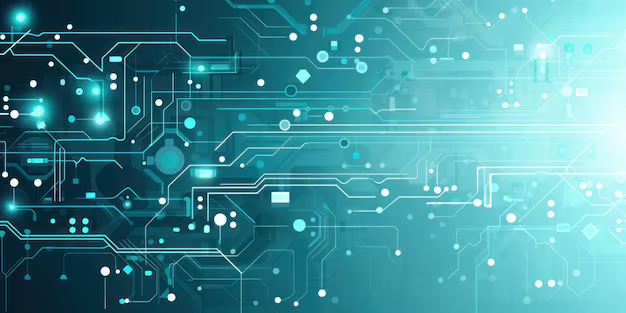


Table of Contents

[**CHAPTER 1:** 8](#_Toc186230617)

[**INTRODUCTION** 8](#_Toc186230618)

[**1.1 Overview:** 8](#_Toc186230619)

[**1.1.1 Objective:** 8](#_Toc186230620)

[**1.1.2 Abstract:** 8](#_Toc186230621)

[**1.2 Introduction to the Medicine Donation System:** 9](#_Toc186230622)

[**CHAPTER 2:** 10](#_Toc186230623)

[**BACKGROUND AND LITERATURE REVIEW** 10](#_Toc186230624)

[**2.1 The need for Medicine Donation Platform:** 10](#_Toc186230625)

[**2.2 Current System and Their Limitation:** 10](#_Toc186230626)

[**2.3 Reviews of Existing Technologies:** 10](#_Toc186230627)

[**2.4 Conclusion:** 11](#_Toc186230628)

[**CHAPTER 3:** 12](#_Toc186230629)

[**SYSTEM ANALYSIS** 12](#_Toc186230630)

[**3.1 System Requirements:** 12](#_Toc186230631)

[**3.1.1 Hardware Requirements:** 12](#_Toc186230632)

[**3.1.2 Software Requirements:** 12](#_Toc186230633)

[**3.1.3 Network Requirements:** 13](#_Toc186230634)

[**3.2 User Role Analysis** 13](#_Toc186230635)

[**3.2.1 Functional Requirements:** 13](#_Toc186230636)

[**3.2.2 Non Functional Requirement:** 14](#_Toc186230637)

[**CHAPTER 4:** 15](#_Toc186230638)

[**TOOLS AND TECHNOLOGIES** 15](#_Toc186230639)

[**4.1 Technologies:** 15](#_Toc186230640)

[**4.1.1 Backend:** 15](#_Toc186230641)

[**4.1.2 Frontend:** 15](#_Toc186230642)

[**4.2 Tools:** 16](#_Toc186230643)

[**4.2.1 Development Environment:** 16](#_Toc186230644)

[**CHAPTER 5:** 19](#_Toc186230645)

[**SYSTEM DESIGN** 19](#_Toc186230646)

[**5.1 System Architecture:** 19](#_Toc186230647)

[**5.2 Database Design:** 20](#_Toc186230648)

[**5.2.1 Database and Collection:** 20](#_Toc186230649)

[**5.2.2 User Profile Collection:** 20](#_Toc186230650)

[**5.2.3 Donation Listing Collection:** 20](#_Toc186230651)

[**5.2.4 Medication and Verification Collection:** 20](#_Toc186230652)

[**5.2.5 Administrator Log Collection:** 20](#_Toc186230653)

[**5.3 User Interface Design for Meds Helping Hand:** 21](#_Toc186230654)

[**5.4 Control Flow Diagram:** 22](#_Toc186230655)

[**CHAPTER 6:** 24](#_Toc186230656)

[**SYSTEM IMPLEMENTATION** 24](#_Toc186230657)

[**6.1 Backend Implementation:** 24](#_Toc186230658)

[**6.2 Frontend Implementation:** 25](#_Toc186230659)

[**6.3 Integration and Deployment:** 25](#_Toc186230660)

[**CHAPTER 7:** 27](#_Toc186230661)

[**TESTING PLAN** 27](#_Toc186230662)

[**7.1 Test Cases:** 27](#_Toc186230663)

[**CHAPTER 8:** 29](#_Toc186230664)

[**WORK BREAKDOWN** 29](#_Toc186230665)

[**CHAPTER 9:** 30](#_Toc186230666)

[**RISK MANAGEMENT** 30](#_Toc186230667)

[**9.1 Modular Development & Testing:** 30](#_Toc186230668)

[**9.2 Integrated Testing & Feedback Loops:** 30](#_Toc186230669)

[**CHAPTER 10:** 31](#_Toc186230670)

[**CLOSURE** 31](#_Toc186230671)

[**CHAPTER 11:** 32](#_Toc186230672)

[**REFRENCES** 32](#_Toc186230673)

[**CHAPTER 12:** 33](#_Toc186230674)

[**APPENDIX A: GLOSSARY** 33](#_Toc186230675)

[**CHAPTER 13:** 34](#_Toc186230676)

[**APPENDIX B** 34](#_Toc186230677)

[**CHAPTER 14:** 35](#_Toc186230678)

[**APPENDIX C: To Be Determine List** 35](#_Toc186230679)

[**CHAPTER 15:** 36](#_Toc186230680)

[**SCREENSHOTS** 36](#_Toc186230681)

# **CHAPTER 1:**

# **INTRODUCTION**

## **1.1 Overview:**

Pakistan's populace is around 220 million out of which in excess of 70 million individuals are living underneath the neediness line for example procuring under $2 per day. Around 78% of the populace is paying for wellbeing consumptions out of their pockets which is reasonable for upper or working class yet for the lower class, it's an alternate story. With the pay of just 5000 to 6000 rupees per month, they need to pay for the food, bills, transport and furthermore for the charges in the event that their kids are contemplating. Additionally, assuming that any of the relatives becomes ill, they are left with less or no cash to treat them. As indicated by the World Wellbeing Association's norm, a nation ought to spend no less than 5% of its Gross domestic product on the wellbeing area. The portion of Gross domestic product that the vast majority of the created countries designate for wellbeing is 8%. Notwithstanding, Pakistan designates under 3% of its Gross domestic product to the wellbeing area which moderately a tiny sum to meet the wellbeing needs of a rising populace. Central point for such a terrible general wellbeing circumstance in the nation, are notable, some of which are:

* Exorbitance of appropriate clinical offices because of neediness.
* Half of the populace in Pakistan don’t have approach to the essential medical care administrations, and roughly 42% of the populace don’t have health care coverage.
* Absence of admittance to safe drinking water.
* Unfortunate cleanliness condition, which incorporates immature disinfection foundation. High proportion of ignorance, prompting unfortunate practices.
* Detachment to clinics and facilities, because of dissipated populace.

### **1.1.1 Objective:**

The primary objective of the Meds Helping Hands application is to facilitate the efficient redistribution of unused medicines by connecting donors with recipients. This system aims to reduce medicinal waste and aid individuals who cannot afford essential medications which ultimately lead to enhancing healthcare access and sustainability.

### **1.1.2 Abstract:**

This project is an online platform to donate unused medicines for the people who cannot afford the expensive medicines in Pakistan. Medications are a fundamental component in lightning enduring, and donation s of clinical supplies with incredibly profit worldwide helpful aid projects. This medication donation application is about the assortment of medication that is unused by the patient who recuperates totally and remaining prescriptions become squander, those medications can be gathered and utilized further. The aim and object of this project is to prepare an Online Portal for the collection of unused medicines so that they can be given to the people who are in need. What this project will do is, to collect the unused medicines from the people who have completely recovered from the illness and do not require the medicines anymore. These medicines would be handed over after checking to the people directly or to the hospitals who will be giving out these medicines for free. This system can contribute to reduce the cost for national health services by making proper use of unused medicines, and to help people to get better health services who can’t afford these medications.

## **1.2 Introduction to the Medicine Donation System:**

The framework created in this undertaking incorporates an exhaustive arrangement of highlights, like secure client enlistment, itemized postings of accessible medications, and a powerful solicitation and endorsement process, all inside a protected and easy to use online platform.

* **Information Assortment:** This information is vital for refining client experience, advancing framework execution, and upgrading the adequacy of the donation cycle. The framework's plan guarantees that information assortment is both thorough and inconspicuous, regarding client protection while social occasion essential bits of knowledge.
* **Information Logging:** Information logging inside the stage is thoroughly kept up with to guarantee extensive following, everything being equal. Every exchange and client connection is signed progressively, giving an important review trail that upgrades security and responsibility.
* **Analysis:** Examination of the logged information gives basic bits of knowledge that drive non-stop improvement of the stage. By understanding client conduct, donatio patterns, and framework bottlenecks, designers can carry out designated upgrades that further develop usefulness and client fulfillment.
* **Reporting:** The revealing element of the stage is intended to give executives integral assets to produce nitty gritty and noteworthy reports on different parts of the framework's activity. These reports help in evaluating the viability of the stage, understanding client commitment and settling on informed conclusions about future turns of events and improvements.

This basic section makes way for a nitty gritty assessment of the framework's plan, execution, and functional results in ensuing parts, clarifying the vital and functional groundworks of the Medications Assistance application.

# **CHAPTER 2:**

# **BACKGROUND AND LITERATURE REVIEW**

## **2.1 The need for Medicine Donation Platform:**

Studies have shown that a lot of recommended drugs stay unused and are frequently disposed of inappropriately, adding to natural harm and addressing a significant monetary waste. This overflow of prescriptions could be diverted to help people who are in battle to manage the cost of their clinical medicines because of high medical services costs. Successful medication donation stages address this by establishing an organized and managed climate where these unused drugs can be reallocated to those out of luck.

## **2.2 Current System and Their Limitation:**

Existing medication donation frameworks frequently depend on manual assortment communities or divided networks that are not adaptable and need functional effectiveness. These frameworks are troubled with calculated challenges including huge and weighty strategies for logging, putting away, and circulating meds. Such shortcomings can prompt postponements, possible mistakes, and expanded functional expenses.

## **2.3 Reviews of Existing Technologies:**

The writing and industry surveys uncover that latest medication donation stages use essential innovations zeroed in principally on data set administration for following stock and client communications, close by basic UIs for working with exchanges. Be that as it may, these frameworks frequently miss the mark in regions like constant information reconciliation, extensive client commitment, and high level safety efforts.

These advances offer the possibility to change the scene of medication donation by presenting more noteworthy proficiency and security. Block chain innovation, for example, could give a decentralized and changeless record framework that guarantees each exchange is recorded safely and straightforwardly, making the donation cycle more dependable and obvious. Then again, Artificial Intelligence could be utilized to dissect verifiable information and client communications to conjecture future patterns in medication interest and supply, working with a more proactive way to deal with donation the board.

## **2.4 Conclusion:**

This survey highlights a basic hole between the capability of innovation and its ongoing application inside the circle of medication donation stages. For the Prescriptions Assistance project, embracing these advances couldn't smooth out tasks yet in addition hoist the security and viability of the medication donation process, setting another norm in the field and possibly affecting more extensive reception across the business.

# **CHAPTER 3:**

# **SYSTEM ANALYSIS**

This section digs into the framework examination of the Drugs Assistance entry, illustrating the framework necessities, client job investigation and utilitarian prerequisites essential for the fruitful organization and activity of this medical services centered stage.

## **3.1 System Requirements:**

The Prescriptions Assistance gateway is based on a hearty heap of innovations intended to work with the productive gift and circulation of drugs. These framework prerequisites are classified into equipment and programming needs to guarantee ideal execution and client openness.

### **3.1.1 Hardware Requirements:**

* **Client Devices:** This entryway is available on standard processing gadgets including work areas, PCs, tablets and cell phones. This guarantees that clients can get to the assistance from any area, gave they have web access.
* **Servers:** Vigorous server framework is utilized to deal with the web and application administrations, including a devoted information base server for MongoDB, which stores and oversees client and exchange information proficiently.

### **3.1.2 Software Requirements:**

* **Operating System:** This entryway is stage free and can be gotten to by means of any working framework with an internet browser, including Windows, Linux, iOS, and Android.
* **Internet Browsers:** The stage is upgraded for present day internet browsers, for example, Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge to guarantee it arrives at a wide client base.
* **Backend Advancements:** Node.js is utilized to deal with server-side prearranging and business rationale, coordinated with MongoDB for information the board, guaranteeing quick information access and superior execution.
* **Frontend Innovations:** The client-side application is constructed utilizing Respond, offering a responsive and dynamic UI that adjusts to different gadget evaluates for a consistent client experience.

### **3.1.3 Network Requirements:**

* **Connectivity:** Clients require a steady web association to access the stage, perform exchanges and get reports on their communications like medication gifts or demands.

## **3.2 User Role Analysis**

The application is designed to cater to three primary user roles each with specific interactions and permissions within the system:

* **Donors:** Users who have unused medicines and wish to donate them. They can register on the platform, submit information about the medicines and track the status of their donations.
* **Recipients:** Individuals or healthcare facilities who are in need of medications. Recipients can search for available medicines, request needed items and provide feedback on the received supplies.
* **Administrators:** Admins manage the overall functionality of the platform. They approve new users, validate medicine donations, manage the distribution process, and generate reports. Administrators also handle critical backend operations and maintain the integrity and security of the system.

Each user role has access to specific functionalities that align with their responsibilities, ensuring the system is secure and efficient.

### **3.2.1 Functional Requirements:**

The framework should have the option to enroll another client and on the off chance that the client is definitely not another client, he/she should have the option to login in the framework utilizing the login capability.

* The framework should acknowledge the name, amount and items in the prescriptions that the client needs to give.
* The framework should likewise demand the pictures of the meds that the benefactor needs to give.
* The framework should likewise show the situation with the gave medication to the contributor for their fulfillment.
* The framework should send a legitimate notice to the donator when the beneficiary will show up to gather the medication.
* The framework should store every one of the information of the gathered meds in a Data set and should sort the information as per the solutions for various sicknesses.
* The framework should incorporate a Direction and Instruction module for the clients to teach about appropriate techniques to give the prescriptions and data on different drugs.
* The framework should send warning to every one of the clients in the event of crisis medication necessities.
* The framework should incorporate a Gatherings Page for clients to examine their inquiries thus specialists and other experienced clients can direct them.

### **3.2.2 Non Functional Requirement:**

* The framework should be easy to understand and the GUI should be straightforward.
* The framework should meet Web Content Availability Rules.
* Every one of the items and parts should be shown appropriately and on their predefined put on the entryway.
* The product should have the option to deal with the predefined stacks and should proceed according to determine in the particulars.
* The framework should be versatile and viable with the predetermined frameworks.
* The product should be effectively viable, accessible for most extreme time and must be solid.
* The product should follow simplicity of ease of use and learnability.
* The framework should keep all the most recent security guidelines and should not be powerless against outer malware and assaults.

# **CHAPTER 4:**

# **TOOLS AND TECHNOLOGIES**

## **4.1 Technologies:**

### **4.1.1 Backend:**

* **Node.js:** Our server side application improvement is essentially subject to Node.js as it can un JavaScript on the server side making it conceivable to foster applications that are both versatile and productive. Significantly, Node.js is especially appropriate for our undertaking given the non-obstructing, occasion driven engineering which is incredibly reasonable for continuous applications that are circulated north of a few gadgets. Moreover, JavaScript's dependence intends that there is simpler correspondence with Mongoose, a framework that sits on top of our MongoDB and smoothest out improvement.
* **MongoDB:** MongoDB is our picked information base for its adaptability with report blueprints, superior execution and versatility. It stores information in BSON design, a double portrayal of JSON-like reports. This adaptability permits our entryway to store information structures that can shift in size and content, supporting the dynamic and differed information prerequisites of our gateway without the requirement for movements normally connected with social data sets. MongoDB's strong questioning capacities, joined with its adaptability both in an upward direction and on a level plane settle on it an ideal decision for web applications anticipating shifted and a lot of information.
* **Express:** A web application framework for node.js, used to create APIs and handle server-side routing

### **4.1.2 Frontend:**

* **React:** We chose Respond for the frontend to use definitive parts work with the production of intelligent UIs. Respond's effective update and delivering enhancements, like its virtual DOM, permit our application to perform well even with elevated degrees of client association and information changes. The particularity of parts in Respond not just aides in dealing with the application's state all the more typically yet additionally supports the reusability of code, upgrading both improvement speed and consistency across various pieces of the application.
* **Redux:** To deal with the application's state worldwide, we coordinated Revival with Respond. Revival gives a focal store that holds all conditions of the application, empowering every part to get to its expected state without requiring callbacks or props tying, hence improving on the engineering. This state the board design is especially valuable in complex applications where different parts need to associate and answer shared information, guaranteeing consistency and consistency of the application's way of behaving.

## **4.2 Tools:**

### **4.2.1 Development Environment:**

**Visual Studio Code (Versus Code):**

Visual Studio Code is our essential code manager, picked for its broad biological system, incorporated Git control, sentence structure featuring, shrewd code fruition, scraps, and code refactoring support. The lightweight idea of Versus Code, joined with its strong highlights, gives a productive coding experience. It upholds a large number of modules that upgrade its usefulness, catering not exclusively to JavaScript/Respond improvement yet in addition coordinating consistently with different advances utilized in the undertaking.

This innovation stack, including both turn of events and testing instruments, frames a strong and hearty starting point for building, testing, and conveying our web application. It guarantees that we are prepared to deal with the requests of present day web improvement, working with the making of a great, viable and versatile item.

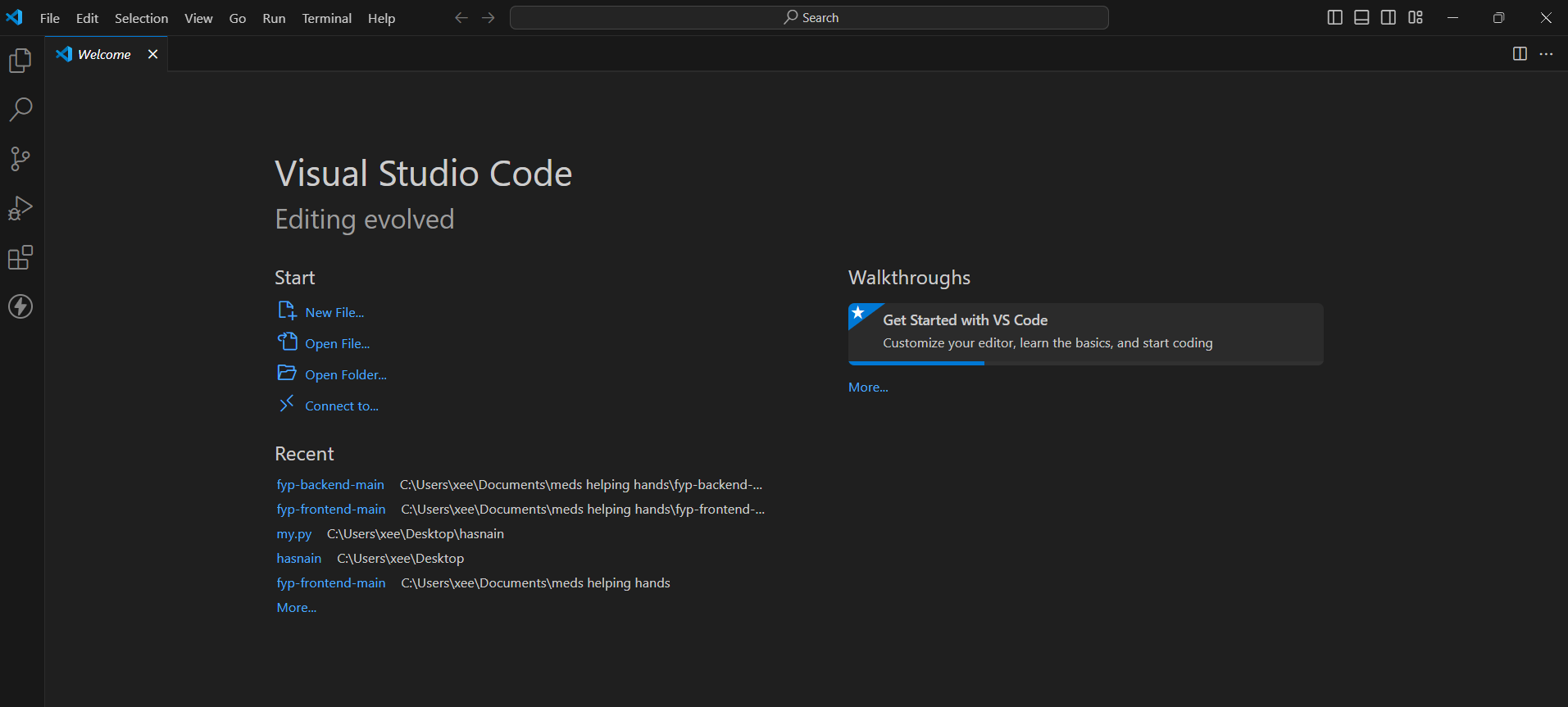


Fig 4.1 interface of VS code

**Coding Practices:**

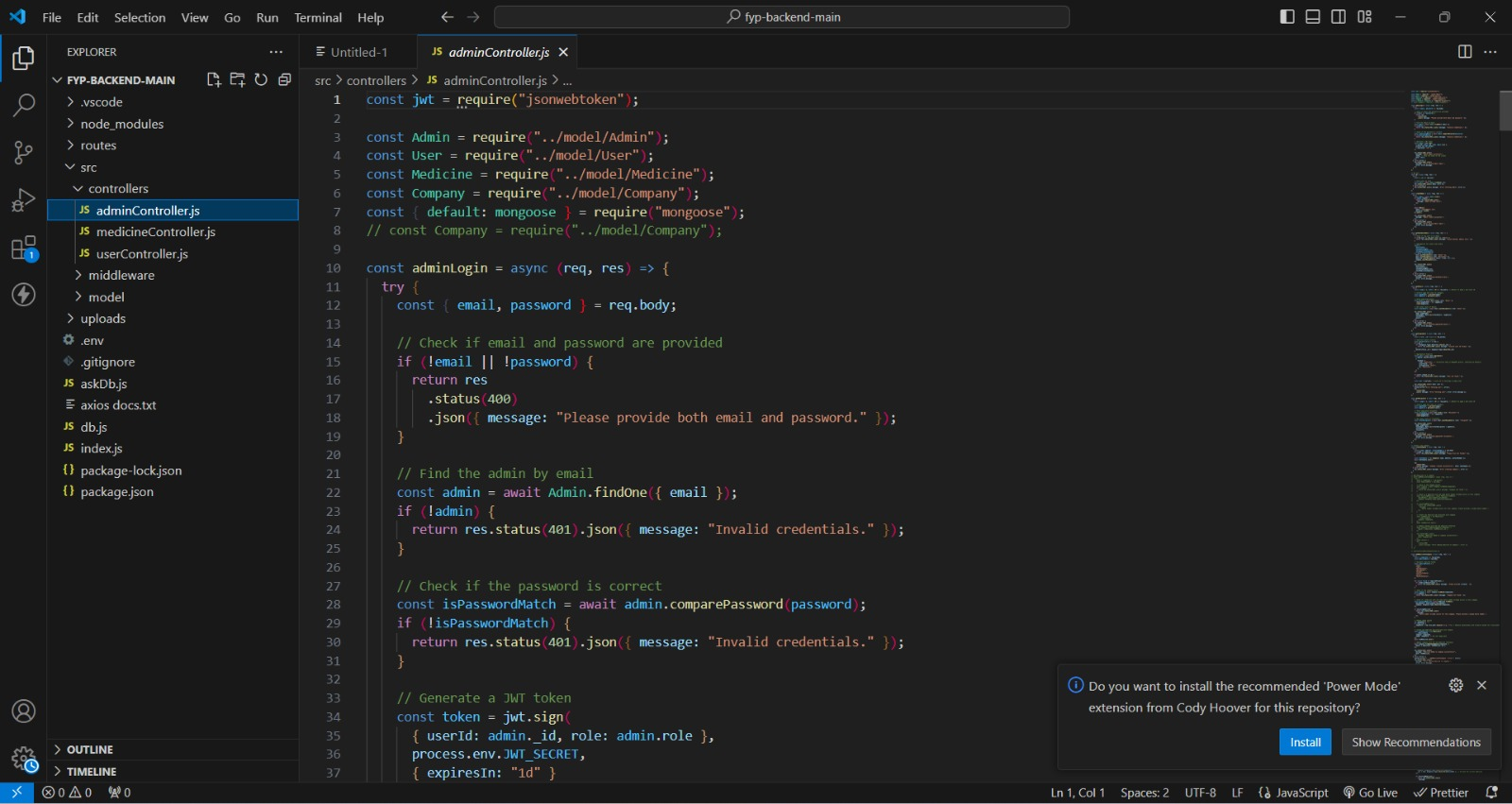


Fig 4.2 Coding Practice 1

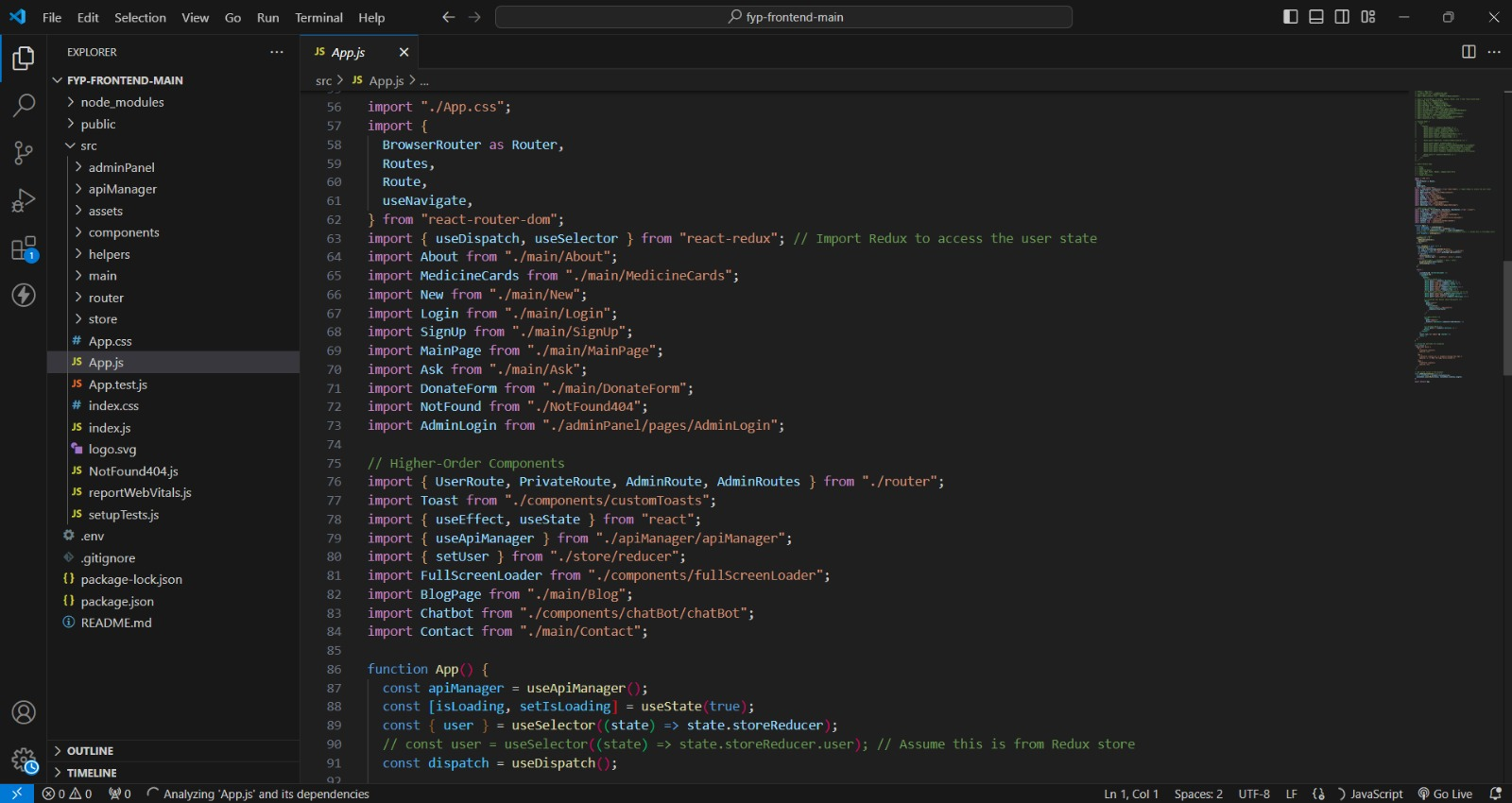


Fig 4.3 Coding Practice 2

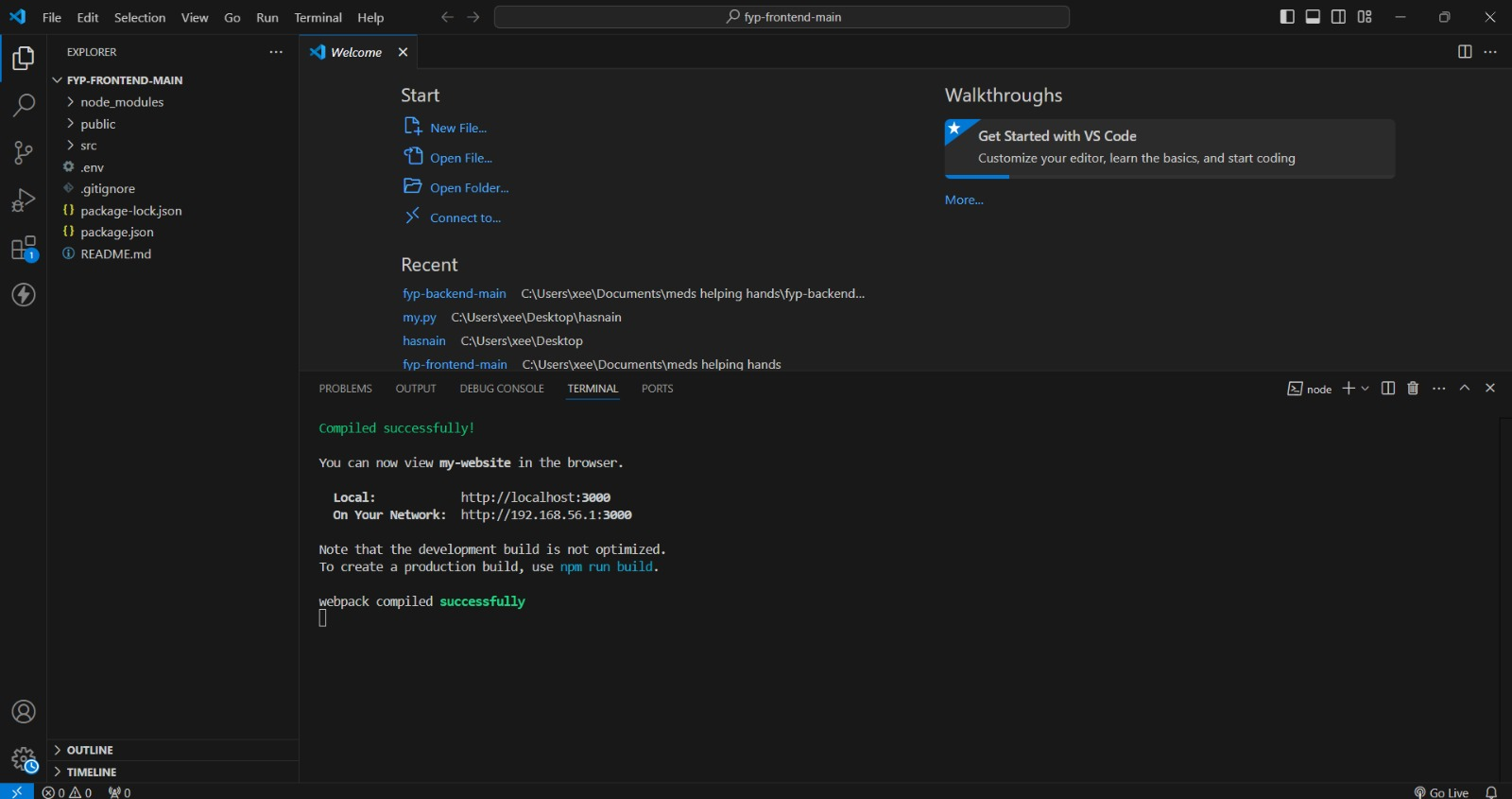


Fig 4.4 Coding Practice 3

# **CHAPTER 5:**

# **SYSTEM DESIGN**

This chapter outlines the system design of the Meds Helping Hands portal, detailing the system architecture, database design and user interface design. This system is structured to ensure efficient data flow, user friendliness and robust data management, reflecting the comprehensive needs of the platform as identified in the Software Requirements Specification and user role analysis.

## **5.1 System Architecture:**

The architecture of the Meds Helping Hands portal is designed to be scalable, secure, and efficient, supporting a multi approach that separates the user interface and data storage layers to enhance maintainability and scalability. The structure that underpins the Meds Helping Hands portal makes sure that the portal is reliable, safe, efficient, and most importantly scalable. It utilizes a multi-tiered approach architecture which enhances maintainability by having the user interface, the business logic, and the data storage in separate layers.

* **Client Layer:** Offering a responsible dynamic interface that can be accessed on a variety of devices, the front end which is constructed using the React framework addresses issues of responsive design. Through this layer, the server is reached through asynchronous calls to an API, thereby adding interactivity and fluidity to the user experience.
* **Server Layer:** All clients requests from the client layer and the database are processed in JavaScript code at the server layer aka back end. This includes business logic such as authenticating users, handling requests, and processing transactions, among others. As such it helps understand and formulate the necessary output while interacting with clients through the rewritten and incoming data.
* **Database Layer:** On the other hand, all users’ information, medicines, administrative, and transactions records in the system are stored in a NoSQL database called MongoDB. Considering that most of the information is likely to grow as the expansion of the platform occurs, using MongoDB allows for more storage and database schema flexibility.
* **Communication Protocols:** In conclusion, the system establishes security between the client and server through HTTPS. In addition, data security is emphasized throughout the processes by employing RESTful APIs to allow communication between front end and back end systems.

This layered architecture not only facilitates effective data management and operational efficiency but also enhances security measures, crucial for protecting sensitive health-related information.

## **5.2 Database Design:**

Planning the MongoDB data set for the Internet based Unused Medication Donation Entryway includes characterizing the information construction and mapping that will store client profiles, donation postings, check records, and other pertinent data. MongoDB is a No SQL data set that utilizes an adaptable blueprint, making it reasonable for taking care of semi-organized and unstructured information. Underneath, I'll give a calculated outline of how MongoDB can be intended for the gateway:

### **5.2.1 Database and Collection:**

Make a MongoDB data set for the entrance, and inside it, characterize assortments (similar to tables in social data sets) to store various sorts of information.

### **5.2.2 User Profile Collection:**

This assortment stores client data, including benefactor and beneficiary profiles.

### **5.2.3 Donation Listing Collection:**

This assortment holds information connected with drug gift postings made by givers.

### **5.2.4 Medication and Verification Collection:**

Store records connected with the check of given drugs in this assortment.

### **5.2.5 Administrator Log Collection:**

For overseers, log significant framework exercises and activities in this assortment.

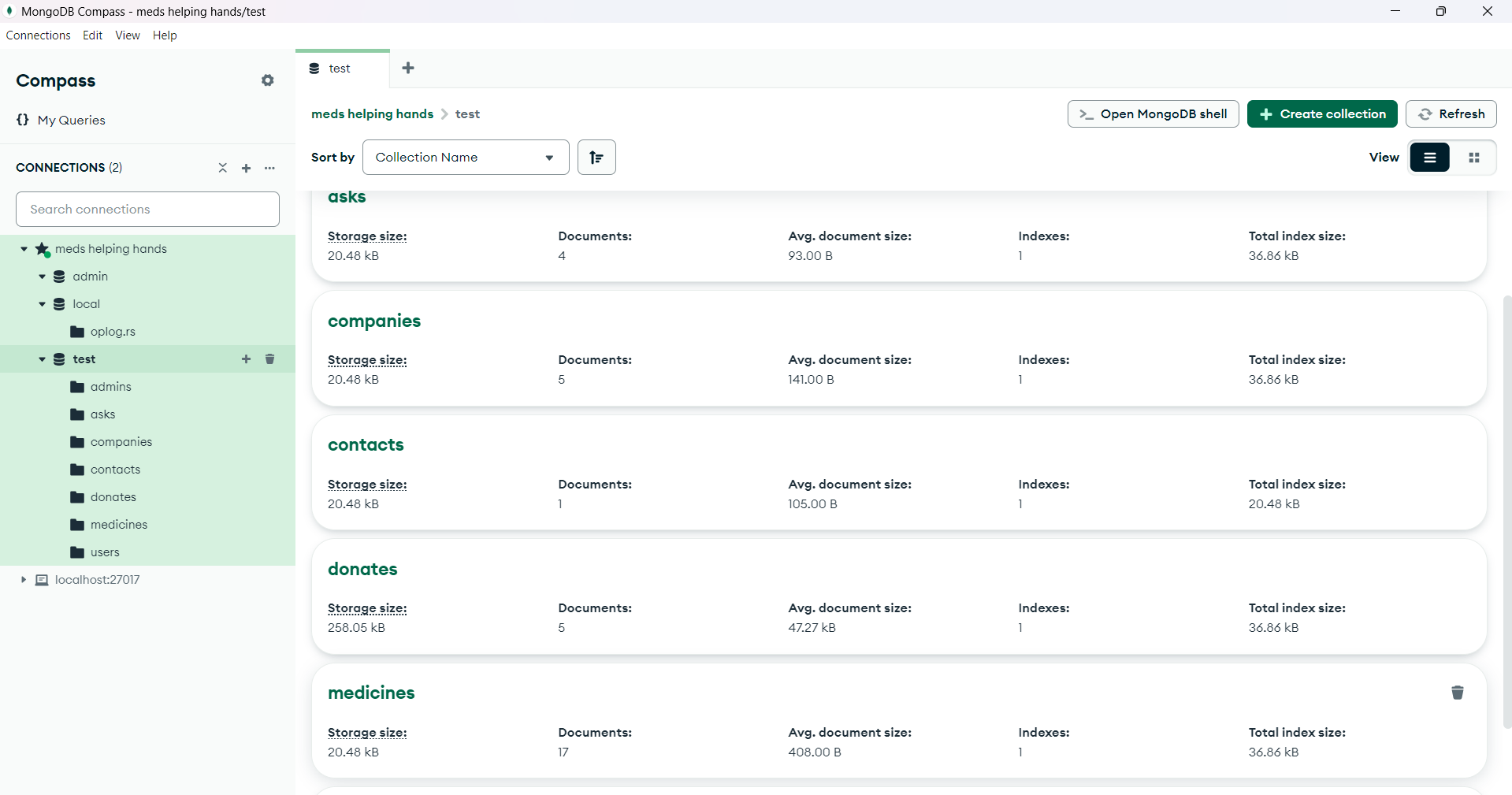


Fig 5.1 Database Interface

## **5.3 User Interface Design for Meds Helping Hand:**

UI arrangement may be essential piece of the Drugs Help application. It is made with client encounter at its center. The point of interaction is intuitive and open. It takes extraordinary care of clients with fluctuating degrees of specialized fitness. Key benchmarks of straightforwardness, accessibility and responsiveness are coordinates into its arrangement. A smooth and exceptionally organized plan is utilized. A course bar engages clients to switch between sections easily.

Ranges join domestic, medicine blessing, medicine requesting and record the board. Structures and sources of data are aiming for comfort. Structures, for case, Selection structures and blessing structures are simple to utilize .Clear names and endorsement messages are solidified. This makes a difference clients in wrapping up forms accurately. Arrange of UI is totally responsive.

This ensures Application ease of utilize on work range and cell phones. Bootstrap structures are utilized. CSS frameworks are utilized for energetic plan modification. Accessibility highlights Recalled for the application are tall distinction modes. Content resizing choices are given. Screen per user bolster is open. The objective is to enable clients with incapacities to investigate and utilize the application without any issue. Arrange components cooperate .They set up a simple to get it climate. They moreover develop communication. They increase commitment with the organization. For the most part talking, these components bolster the objective. That's to advance create induction to restorative care. Usually wrapped up by capably passing on unused solutions. The segment portrays a system. This system was exactingly organized to meet the specific necessities and commitments of different clients. The system guarantees that each component of building and UI maintains a capable, secure and charming involvement for each single included party.

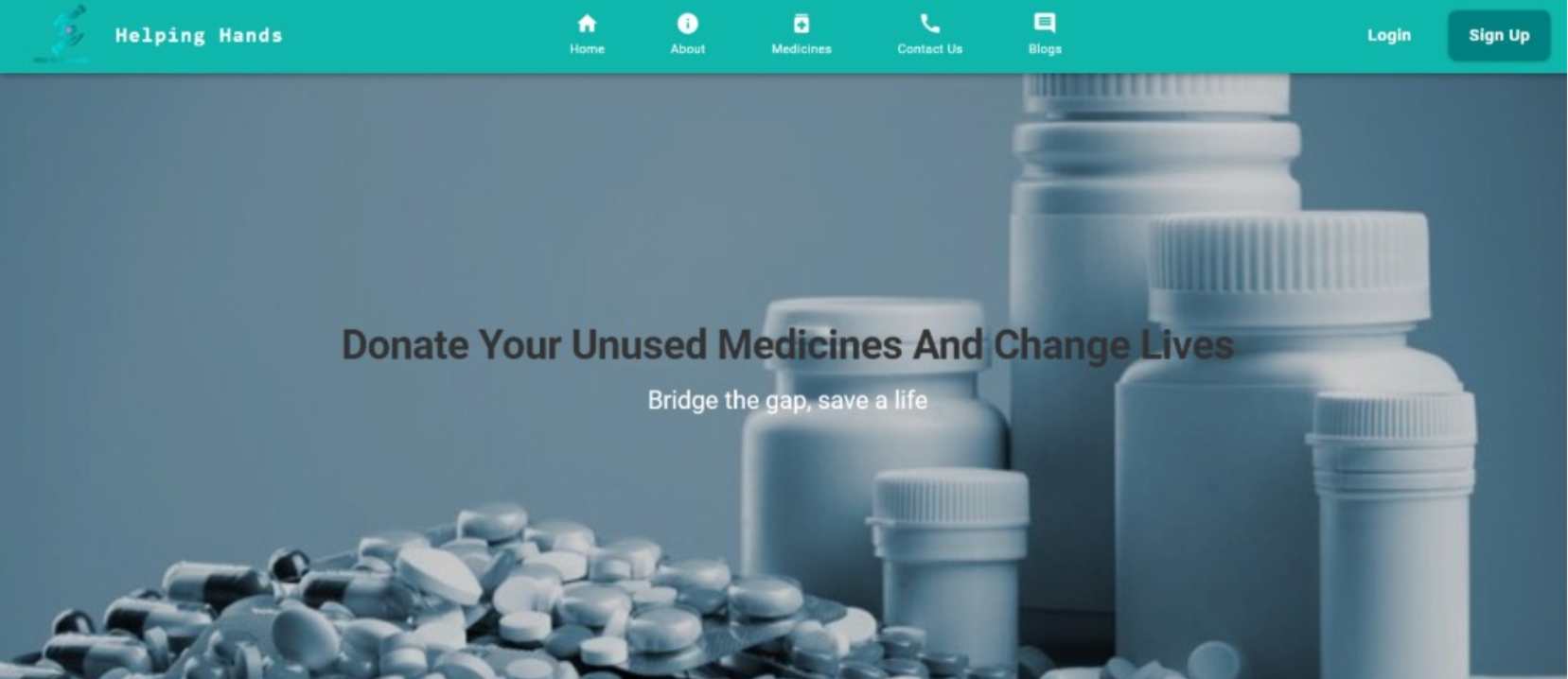


Fig 5.2 User Interface

## **5.4 Control Flow Diagram:**

As we can see in the block diagram below, there are three external entities, namely: the Admin, Member Users (Donors and Receivers), and the Medicines. The task and functionalities of each user is listed alongside them and the data flow between them is also specified. As listed, the functionalities of Admin are: Login and Management of all the users (donors and receivers), approvals from receivers and management of the database with the total medicine distribution; the functionalities of the Donor are: Login and Donate Medicines, see the medicines donated by other users, check the status of the donated medicines; the functionalities of the Receiver are: Register, Login, Add prescription, and receive medications; the functionalities of Medicines section are: Login and Accept the medicines from donors, collect and segregate the medicines, and updating of the database.

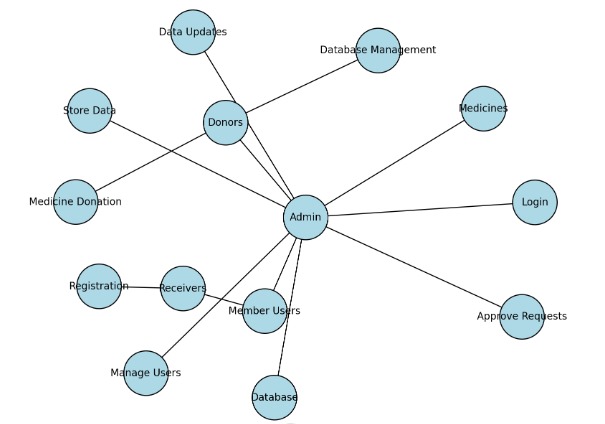


Fig 5.3 Control Flow

# **CHAPTER 6:**

# **SYSTEM IMPLEMENTATION**

This section talks about the execution of the Prescriptions Assistance application, enumerating the backend and frontend parts, as well as the cycles utilized for reconciliation and arrangement of the framework. This period of improvement is urgent for transforming plan and arranging into a utilitarian framework that meets the undertaking prerequisites.

## **6.1 Backend Implementation:**

The backend of the Prescriptions Assistance application is carried out utilizing Node.js, a JavaScript runtime climate that executes JavaScript code server-side, and MongoDB, a NoSQL information base, for putting away and overseeing information. This is the way these parts are carried out:

* **Server Arrangement:** Node.js makes a strong server that handles HTTP solicitations and reactions. Utilizing the Express structure improves on the steering and middleware designs, taking into account a coordinated and particular arrangement.
* **Data base Association:** MongoDB is coordinated utilizing Mongoose, an Item Information Demonstrating (ODM) library for MongoDB and Node.js. Mongoose works with the association of MongoDB information base and gives a straightforward answer for characterizing patterns for the information and overseeing connections between information.
* **Programming Interface Advancement:** Peaceful APIs are created to empower correspondence between the frontend and the backend. These APIs handle different functionalities like client verification, information recovery, information accommodation, and server-side rationale.
* **Verification:** The backend executes JWT (JSON Web Tokens) for overseeing client meetings and getting the Programming interface. This includes producing a token upon login and confirming the token on resulting solicitations to safeguard courses that require confirmation.
* **Blunder Dealing:** Appropriate mistake dealing with components are gotten up in a position get and answer different blunders, like data set association mistakes, information approval blunders, and course not found mistakes, guaranteeing that the waiter stays steady and solid.

## **6.2 Frontend Implementation:**

The frontend of the Prescriptions Assistance application is fabricated utilizing Respond, a famous JavaScript library for building UIs, alongside Revival for state the board. The execution subtleties include:

* **Component Architecture:** The application is organized into reusable Respond parts. Every part is liable for a particular piece of the UI, and together, they structure the whole application frontend.
* **State Management:** Revival is utilized to halfway deal with the application's state. This is especially valuable for taking care of the confirmation state, client information, and the situation with network demands across different parts.
* **Routing:** Respond Switch is executed to oversee route inside the application, empowering the formation of a solitary page application (SPA) where clients can explore between various pieces of the application without reloading the page.
* **Style:** CSS and Bootstrap are utilized to style the parts. The utilization of Bootstrap takes into considering responsive plans that work on an assortment of gadget sizes, it is versatile to guarantee the application.

Intelligent Components: Different intelligent components like structures, fastens, and connects are carried out to permit clients to interface with the application easily and effectively.

## **6.3 Integration and Deployment:**

Integration and deployment are critical steps in making the application accessible to users. The process involves:

* **Continuous Integration (CI):** Apparatus like Jenkins or GitHub Activities are utilized to robotize the testing and working of the application, guaranteeing that each converge into the primary branch is consequently tried and fabricated. This aides in distinguishing combination issues early and keeps the arrangement cycle smooth.
* **Testing:** Before arrangement, the application goes through different kinds of testing including unit testing, combination testing, and client acknowledgment testing (UAT) to guarantee all parts cooperate true to form.
* **Deployment:** The application is sent on a cloud specialist co-op like AWS, Sky blue, or Heroku. The backend and frontend are ordinarily conveyed independently, with the frontend being facilitated as a static site and the backend as a bunch of versatile server occasions.
* **Monitoring Maintenance:** When sent, the application's presentation is constantly checked utilizing devices like New Artifact or Data log. Logs are assessed consistently to catch and fix mistakes continuously.

This complete way to deal with execution, mix, and organization guarantees that the Medications Assistance application is strong, solid, and prepared for certifiable use, actually supporting the gift and dissemination of prescriptions to those out of luck.

# **CHAPTER 7:**

# **TESTING PLAN**

In the testing phase, we use different testing approaches to check the Functional and Nonfunctional part of our Project to ensure the quality of the project. The testing part includes these testing techniques:

* Smoke Testing was done to check and identify key issues and bugs before moving on to next step of the testing.
* Scenario Testing was done to ensure, the project is working as expected.
* Integration Testing was done to check that integrated modules interact efficiently with each other without any hazards.
* Performance Testing was done to check the system’s performance for best use.
* Load Testing was done to ensure the system's scalability and confirm it can handle the expected number of users.
* Usability Testing was done to find any usability issues and improve the application's ease of use

## **7.1 Test Cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Expected Result** | **Pass/Fail** | **Steps** |
| TC-001 | Verify Login with valid data | User successfully logs in | Pass | 1. Navigate to the login page. 2. Enter valid username and password. 3. Click the 'Login' button. |
| TC-002 | Verify Login with invalid data | Error message displayed | Pass | 1. Navigate to the login page. 2. Enter invalid username/password. 3. Click the 'Login' button. |
| TC-003 | Verify Signup functionality | User successfully registers | Pass | 1. Navigate to the signup page. 2. Fill in required details. 3. Click 'Signup' button. |
| TC-004 | Verify Home Page load | Home page loads with all elements visible | Pass | 1. Navigate to the website URL. 2. Verify visibility of header, navigation, content, and footer. |
| TC-005 | Verify Donate Medicine form | Donation form submits successfully | Pass | 1. Navigate to 'Donate Medicine' page. 2. Fill in the form. 3. Click the 'Submit' button. |
| TC-006 | Verify Contact Us form | Contact form submits successfully | Pass | 1. Navigate to 'Contact Us' page. 2. Fill in the form. 3. Click the 'Send' button. |
| TC-007 | Verify Login with empty fields | Error message displayed | Pass | 1. Navigate to the login page. 2. Leave username/password fields empty. 3. Click 'Login' button. |
| TC-008 | Verify Signup with invalid data | Error message displayed | Pass | 1. Navigate to the signup page. 2. Enter invalid details. 3. Click 'Signup' button. |
| TC-009 | Verify Home Page navigation | User navigates to correct pages from the menu | Pass | 1. Navigate to the home page. 2. Click each menu item. 3. Verify navigation to the correct page. |
| TC-010 | Verify Donate Medicine page load | All elements load correctly | Pass | 1. Navigate to 'Donate Medicine' page. 2. Verify form and content visibility. |
| TC-011 | Verify Contact Us form validation | Error shown for empty/invalid fields | Pass | 1. Navigate to 'Contact Us' page. 2. Leave fields empty or enter invalid data. 3. Click 'Send' button. |

# **CHAPTER 8:**

# **WORK BREAKDOWN**

The Work Breakdown Structure (WBS) is a tool that breaks down a project into smaller components, facilitating the analysis and prioritization of modules by team members. Presented below is the WBS for our project.

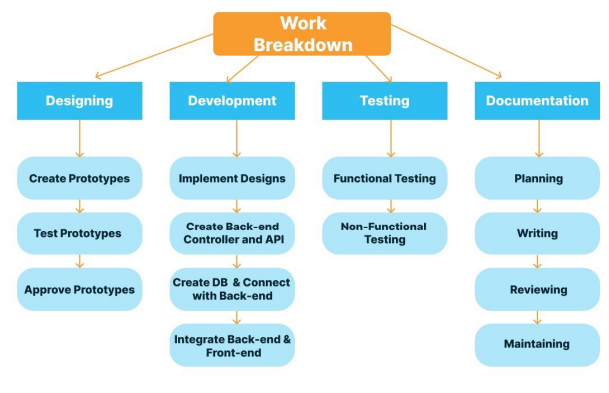


Fig 8.1 WBS

# **CHAPTER 9:**

# **RISK MANAGEMENT**

Ensuring the Project Success while managing the risk on the same time was the crucial part of the project, especially developing the technical part of the Web application, our main goal was to deliver the high-quality product/solution to the users. To achieve this we use these risk management plans:

## **9.1 Modular Development & Testing:**

• We divide the tasks into separate modules that could be developed and checked/tested independently before integration.

• Each module was test by each team member to maintain the quality of the project.

## **9.2 Integrated Testing & Feedback Loops:**

* We integrated modules and tested it to check the performance and to maintain the quality.
* We checked every loop hole and fix that on the spot.
* If we face any issues while integrating the modules, we again separated them and test them individually.

By implementing this structured plan, the aim was to minimize risks at each module of the project that could compromise the end goals. A modular process provided controls throughout development to successfully deliver a robust solution to meet user needs

# **CHAPTER 10:**

# **CLOSURE**

**CONCLUSION:**

Modern era has begun. Either privileged or unprivileged people, literate or illiterate are now conscious about their health. It is a material of sorrow that being conscious poor people cannot pay much attention to their health care routine because of their low income. Our portal takes great initiative by providing free medicines to needy people. But most of the cases, they get the treatment not the expensive remedies. Therefore, this great initiative become valuable to them. This paper provides a brief overview of the design and development of Online Unused Medicine Donation Portal, which will be very effective and will bear great contribution to get the health services for these needy or login come people. Due to this Portal even wastage of lots of medicines gets reduce.

**FUTURE SCOPE:**

This project Meds Helping Hands Portal in the future has the ability to become a full fledge application wherein all the facilities will be provided on this portal. In future, it can be expanded to provide treatment to user via video consultancy with doctors providing prescription upload feature. It can be very beneficial to user because as per the current situation of different diseases and it is very important to follow governments rules and regulations. The feature of video consultancy with doctor is very best option of future scope for the portal. For better suggestions, we are merging all the medication facility like consultation facilities as a part of the portal in future. So, in future, it can also become a commercial portal and app which will be unique in its nature and availability.

# **CHAPTER 11:**

# **REFRENCES**

**ARTICLES:**

[1] L. K. Pulasthi Dhananjaya Gunawardhanaorcid, “Process of Requirement Analysis Link to Software Development”, Department of Information Communication Technology, Faculty of Technology, University of Sri Jayewardenapura, Nugegoda, Sri Lanka, October 2019.

[2] JD Herbsleb, D Moitra, “Global Software Development”, IEEE software, 2001

[3] S Apel, C Kästner - J. Object Technol, “An Overview on featured oriented software development”, 2009.

**BOOKS:**

[1] Robert Cecil Martin, “Clean Code”, Prentice Hall, 2008.

[2] Kathy Sierra, Eric Freeman, Bert Bates“Head First Design Pattern”, O’Realiy Media, 2004 [3] Andrew Hunt, David Thomsan, “The Pragmatric Programmer”, Addison-wesely Professionals, 1999.

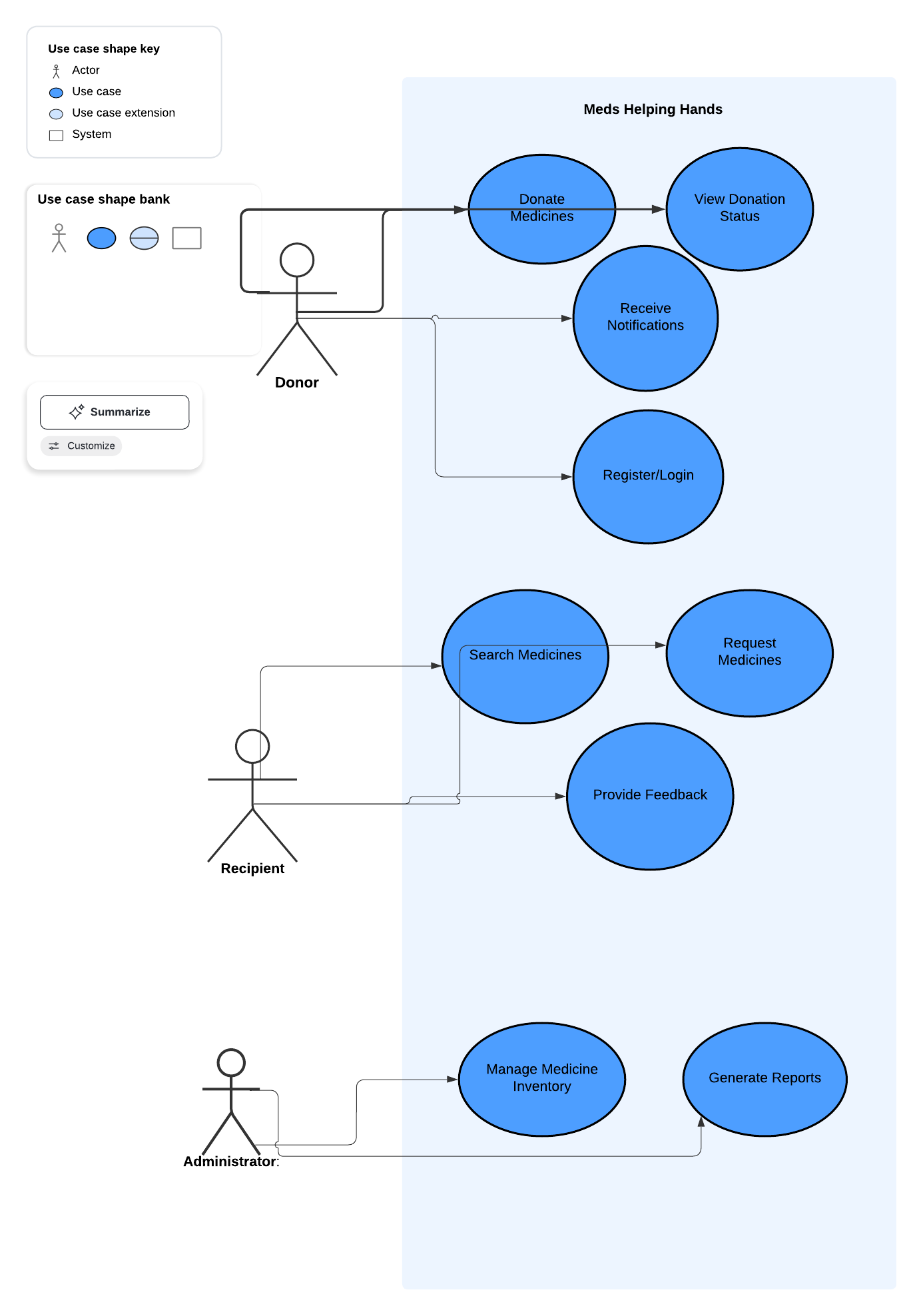
# **CHAPTER 12:**

# **APPENDIX A: GLOSSARY**

* **Node.js:** Node JavaScript
* **Express.js:** Express JavaScript
* **MongoDB:** Mongo Database
* **React.js:** React JavaScript
* **Mongoose:** MongoDB Object Data Modeling
* **API:** Application Programming Interface
* **CRUD:** Create, Read, Update, Delete
* **SQL:** Structured Query Language
* **PECA:** Prevention of Electronic Crimes Act
* **UI:** User Interface
* **UX:** User Experience

# **CHAPTER 13:**

# **APPENDIX B**

Fig 13.1 Use Case Diagram

# **CHAPTER 14:**

# **APPENDIX C: To Be Determine List**

• Specification algorithm for personalized recommendation.

• Third party APIs integration.

• Detail user role and permission.

• Data backup frequency and retention policies.

• Security protocols for data encryption.

• User interface design standard.

• Detailed risk management plan

• Localization requirement.

# **CHAPTER 15:**

# **SCREENSHOTS**

