

“WeHeal”

an all-in-one healthcare web application

Software Requirements Specification

Prepared by

Student ID	Name
22201594	Reshad Ul Karim
22201628	Sammam Mahdi
22201574	Syeda Maliha Tabassum
22201679	Shararah Kibria

1. Introduction.....	3
1.1 Purpose.....	3
1.2 Scope.....	3
1.3 Definitions, Acronyms, and Abbreviations.....	3
1.4 References.....	3
1.5 Overview.....	4
2. Overall Description.....	4
2.1 Product Perspective.....	4
2.2 Product Features.....	4
2.3 User Classes and Characteristics.....	4
2.4 Operating Environment.....	5
2.5 Constraints.....	5
2.6 Assumptions and Dependencies.....	5
3. System Requirements.....	5
3.1 Functional Requirements.....	5
3.1.1 Authentication & Authorization.....	5
3.1.2 Patient & Doctor Interaction.....	6
3.1.3 Patient Care & Medical Services.....	6
3.1.4 Additional Services.....	6
3.2 Non-Functional Requirements.....	7
3.2.1 Performance Requirements.....	7
3.2.2 Security Requirements.....	7
3.2.3 Reliability & Availability.....	7
3.2.4 Maintainability.....	7
3.2.5 Scalability.....	7
3.3 External Interface Requirements.....	8
3.3.1 User Interfaces.....	8
3.3.2 Hardware Interfaces.....	8
3.3.3 Software Interfaces.....	8
3.3.4 Communication Interfaces.....	8
4. Technology Stack & Architectural Overview.....	8
4.1 MERN Stack Components.....	8
4.2 High-Level Architecture.....	8
5. Tentative Development Plan (Agile Methodology).....	9
Sprint 1 (Weeks 1–2): Core Setup & Authentication.....	9
Sprint 2 (Weeks 3–4): Appointment & Telemedicine System.....	9
Sprint 3 (Weeks 5–6): Patient Care & Emergency Services.....	10
Sprint 4 (Week 7): Payment & Additional Services.....	10
Sprint 5 (Weeks 8–9): Service Quality & Final Touches.....	11
6. Acceptance Criteria.....	11
7. Conclusion.....	11

1. Introduction

1.1 Purpose

This Software Requirement Specification (SRS) document outlines the requirements for developing “**WeHeal**” - **an all-in-one healthcare web application** using the MERN stack (MongoDB, Express.js, React.js, Node.js). The primary goal of this application is to facilitate a secure and efficient healthcare ecosystem, enabling patients to find nearby clinics/hospitals, schedule appointments, receive telemedicine services, manage medical records, and process payments—all while ensuring compliance with relevant healthcare regulations .

1.2 Scope

The scope of this project includes the design, development, testing, and deployment of the healthcare web application, catering to:

- **Patients** who can register, book appointments, access electronic medical records (EMR), and receive telemedicine services.
- **Doctors** who can manage appointments, conduct telemedicine sessions, maintain patient records, and handle prescriptions.
- **Clinic Staff** who can manage appointments, generate reports, and assist with administrative tasks.
- **Administrators** who oversee system operations, manage user roles, and maintain compliance standards.

This SRS covers the functional and non-functional requirements for the project, the constraints, assumptions, high-level design, and the development plan following Agile methodology.

1.3 Definitions, Acronyms, and Abbreviations

- **MERN**: MongoDB, Express.js, React.js, Node.js.
- **EMR**: Electronic Medical Records.
- **HIPAA**: Health Insurance Portability and Accountability Act (U.S. standard for healthcare data privacy).
- **GDPR**: General Data Protection Regulation (European Union data privacy and security law).
- **OTP**: One-Time Password.
- **RBAC**: Role-Based Access Control.

1.4 References

- MERN Stack documentation:
 - [MongoDB](#)
 - [Express.js](#)
 - [React.js](#)
 - [Node.js](#)

1.5 Overview

Section 2 provides an overall description of the product, including its user classes and environment. Section 3 details the functional and non-functional requirements. Section 4 outlines the technology stack and architectural overview. Section 5 presents the development plan, including sprint-wise breakdown and deliverables.

2. Overall Description

2.1 Product Perspective

The healthcare web application is a standalone system built on the MERN stack. It integrates with external APIs or services where necessary (e.g., payment gateways, SMS/email gateways, telemedicine integrations). The application must handle high levels of security and maintain data privacy in accordance with HIPAA/GDPR.

2.2 Product Features

1. **Location Services:** Real-time display of nearby clinics, hospitals, and doctors based on a patient's location.
2. **OTP-Based Authentication:** Secure login and registration for both patients and doctors.
3. **Role Management:** Differentiated dashboards for patients, doctors, staff, and admin users.
4. **Appointment & Telemedicine:** Scheduling, rescheduling, and cancellation of appointments; real-time doctor availability; video consultations via WebRTC or a third-party provider (e.g., Twilio).
5. **Notifications:** Automated SMS/email/push reminders for upcoming appointments, prescription pickups, and follow-up.
6. **EMR & Prescription Management:** Securely store medical records; generate and share e-prescriptions.
7. **Emergency Services:** Quick ambulance booking, emergency contact lists, and nurse support for urgent care.
8. **Payment & Insurance Integration:** Support for online payments, insurance claims, and reimbursements.
9. **Medicine & Lab Services:** Online medicine ordering and lab test bookings, with delivery options.
10. **Feedback & Ratings:** Patients can rate doctors, share feedback, and view ratings to facilitate service quality.

2.3 User Classes and Characteristics

1. **Patients:** Individuals seeking medical consultations, lab tests, emergency services, etc. Typically non-technical, requiring a user-friendly interface and secure data handling.

2. **Doctors:** Healthcare providers who manage appointments, conduct telemedicine, update EMRs, and generate prescriptions. Require an intuitive interface to manage multiple patients efficiently.
3. **Clinic Staff:** Administrative staff that handles scheduling, reception, and patient coordination. They need access to appointment management features and limited patient data.
4. **Administrators:** System overseers responsible for user management, compliance checks, and application maintenance. Have full access to system settings and analytics dashboards.

2.4 Operating Environment

- **Web Application:** Accessible on modern web browsers (Chrome, Firefox, Edge, Safari).
- **Mobile Compatibility:** Responsive design for smartphones and tablets.
- **Server-Side:** Node.js and Express.js running in a containerized or cloud environment.
- **Database:** MongoDB cluster (preferably cloud-managed for scalability and backups).
- **Hosting:** Deployment could be on AWS, Azure, Google Cloud, or any equivalent production environment.

2.5 Constraints

- **Regulatory Compliance:** Must adhere to HIPAA/GDPR for handling patient data.
- **Security:** Must implement encryption at rest and in transit (HTTPS/SSL, database encryption).
- **Performance:** The system must support a large user base with minimal latency for telemedicine and real-time updates.
- **Scalability:** The system architecture should allow horizontal scaling to handle peak loads.
- **Time & Resource:** The project is to be delivered in approximately 8–9 weeks with a team of four developers.

2.6 Assumptions and Dependencies

- Users have a stable internet connection for telemedicine and streaming.
- Third-party services for payment, SMS/email, and video conferencing are available and reliable.
- Mobile users will grant access to GPS for location-based services when prompted.
- Clinic staff are knowledgeable in using web-based tools and can be trained if needed.

3. System Requirements

3.1 Functional Requirements

3.1.1 Authentication & Authorization

1. **OTP Authentication:**
 - **FR-1:** The system shall generate and send a time-bound OTP to the user's email or phone number for registration and login.
 - **FR-2:** The system shall invalidate OTPs after a configurable time period (e.g., 5 minutes).

2. Role-Based Access Control (RBAC):

- **FR-3:** The system shall provide different dashboards for patients, doctors, clinic staff, and admins.
- **FR-4:** The system shall restrict each user's access to features according to their role (e.g., staff cannot access full patient EMR).

3.1.2 Patient & Doctor Interaction

1. Appointment Management:

- **FR-5:** Patients shall be able to search for doctors by specialty, location, or availability.
- **FR-6:** Patients shall be able to schedule, reschedule, or cancel appointments.
- **FR-7:** Doctors shall be able to update their availability in real-time (e.g., "available," "busy," "offline").
- **FR-8:** The system shall display a virtual queue for walk-in patients with real-time status updates.

2. Telemedicine Integration:

- **FR-9:** The system shall support secure voice/video calls for remote consultations.
- **FR-10:** The system shall store basic call logs and conversation summaries without violating privacy regulations.
- **FR-11:** The system shall enable secure in-app messaging for doctor-patient communication.

3.1.3 Patient Care & Medical Services

1. Electronic Medical Records (EMR):

- **FR-12:** The system shall maintain a secure, encrypted record of all patient medical data, including prior consultations.
- **FR-13:** Doctors and authorized staff shall be able to view and update EMR details (e.g., diagnosis, treatment plan).

2. Prescription Management:

- **FR-14:** The system shall allow doctors to generate e-prescriptions and share them with patients or partner pharmacies.
- **FR-15:** Patients shall be able to view their prescription history in their dashboard.

3. Emergency Services:

- **FR-16:** Patients shall have a quick-access feature to emergency contacts or facilities.
- **FR-17:** The system shall allow ambulance booking and real-time tracking if integrated with external providers.
- **FR-18:** Nurses or specialized staff shall have an interface for emergency service requests (oxygen, wheelchair).

3.1.4 Additional Services

1. Payments & Insurance Integration:

- **FR-19:** The system shall integrate with at least one major payment gateway (Stripe, PayPal, etc.).

- **FR-20:** Users shall have the option to link health insurance for streamlined claims processing.
2. **Medicine & Lab Services:**
 - **FR-21:** Patients shall be able to order prescribed medicines and track delivery.
 - **FR-22:** Patients shall be able to book lab tests and view results through the app.
 3. **Reward System:**
 - **FR-23:** The system shall allow loyalty points, discounts, or referral bonuses for regular or referred users.
 4. **Feedback & Ratings:**
 - **FR-24:** Patients shall be able to provide feedback and ratings for doctors and services.
 - **FR-25:** Admin shall have access to an aggregated view of feedback for quality monitoring.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

- **NFR-1:** The system should handle multiple concurrent telemedicine sessions without significant degradation in performance.
- **NFR-2:** Average page load time shall not exceed a defined time limit under normal network conditions.

3.2.2 Security Requirements

- **NFR-3:** All communication between client and server shall occur over HTTPS (SSL/TLS).
- **NFR-4:** Sensitive fields in the database (e.g., patient records, payment details) shall be encrypted at rest.
- **NFR-5:** The system must comply with regulations regarding data privacy, consent, and breach notifications.

3.2.3 Reliability & Availability

- **NFR-6:** The system shall maintain 99.9% uptime, excluding scheduled maintenance.
- **NFR-7:** Data backups shall be taken daily to ensure minimal data loss in case of catastrophic failure.

3.2.4 Maintainability

- **NFR-8:** The codebase shall follow consistent naming conventions and standard design patterns to ease future maintenance.
- **NFR-9:** Documentation of APIs, data models, and architecture shall be kept current.

3.2.5 Scalability

- **NFR-10:** The system architecture shall accommodate horizontal scaling of the Node.js application and MongoDB cluster with minimal reconfiguration.

3.3 External Interface Requirements

3.3.1 User Interfaces

- **UI-1:** A responsive web interface (React.js) that adjusts to desktop, tablet, and mobile viewports.
- **UI-2:** Intuitive navigation with separate dashboards for each user role.

3.3.2 Hardware Interfaces

- **HI-1:** None specific; the system will run on common server hardware or cloud-based instances.

3.3.3 Software Interfaces

- **SI-1:** Payment Gateway APIs (Stripe, PayPal, or equivalent).
- **SI-2:** SMS/email services for OTP and notifications (Twilio, SendGrid, or equivalent).
- **SI-3:** Video conferencing services (WebRTC-based or third-party SDK).

3.3.4 Communication Interfaces

- **CI-1:** RESTful APIs will be used for communication between front-end (React.js) and back-end (Node.js/Express.js).

4. Technology Stack & Architectural Overview

4.1 MERN Stack Components

1. **MongoDB:** Document-oriented database for storing user profiles, appointment data, EMRs, etc.
2. **Express.js:** Node.js framework for building the RESTful API and handling server-side logic.
3. **React.js:** Front-end library for building a responsive and interactive user interface.
4. **Node.js:** JavaScript runtime for executing the back-end application.

4.2 High-Level Architecture

1. **Presentation Layer (React.js):** Handles user interaction, form submissions, and displays data fetched from the API.
2. **Business Logic Layer (Express.js/Node.js):** Processes requests, enforces business rules, and orchestrates data flow.
3. **Data Layer (MongoDB):** Stores all persistent data, including user accounts, appointments, EMRs, etc.

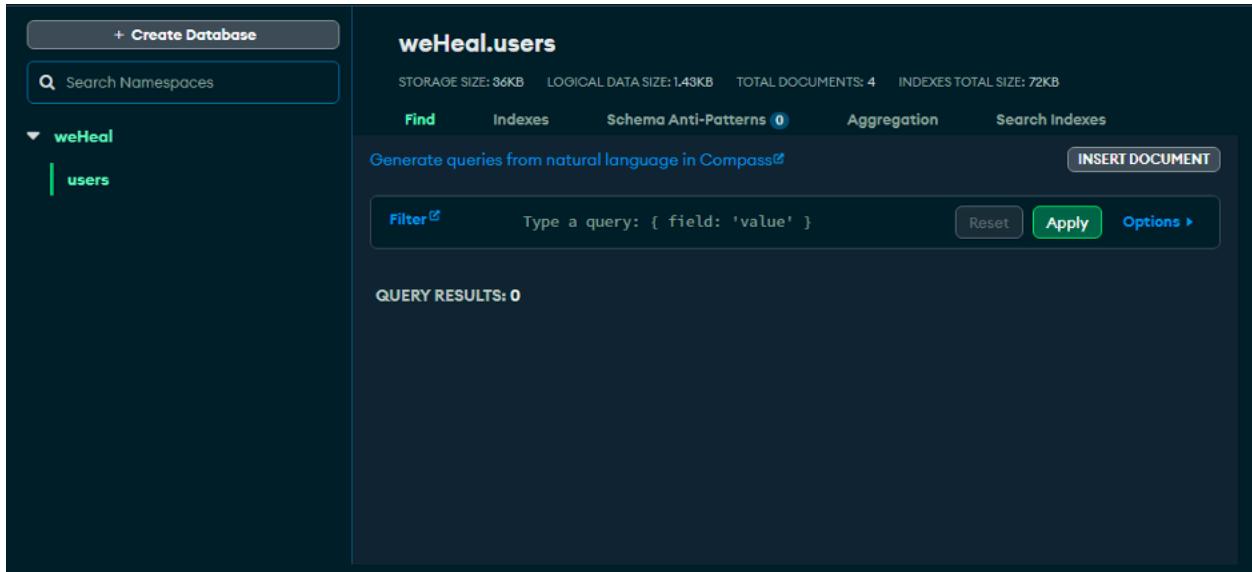
6. Conclusion

This SRS provides a detailed outline for the design and development of “WeHeal”—a comprehensive healthcare web application using the MERN stack. It ensures a patient-centric approach by integrating core healthcare functionalities—scheduling, telemedicine, EMR, and emergency services—while maintaining strict data privacy and security compliance. By following the outlined Agile sprints, the project aims to deliver a robust, scalable, and user-friendly solution within the designated timeline.

SPRINT 2

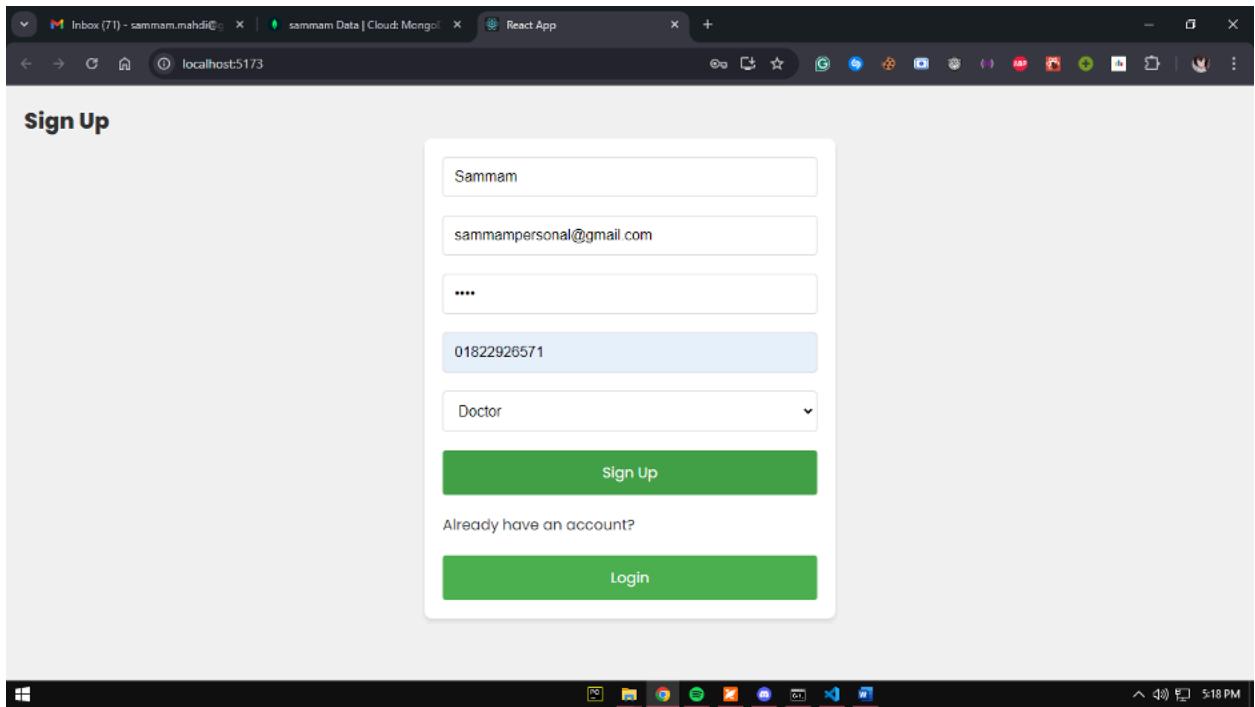
Feature 1 & 2: Registration and OTP Authentication

- Primarily the user table is empty before registering a user

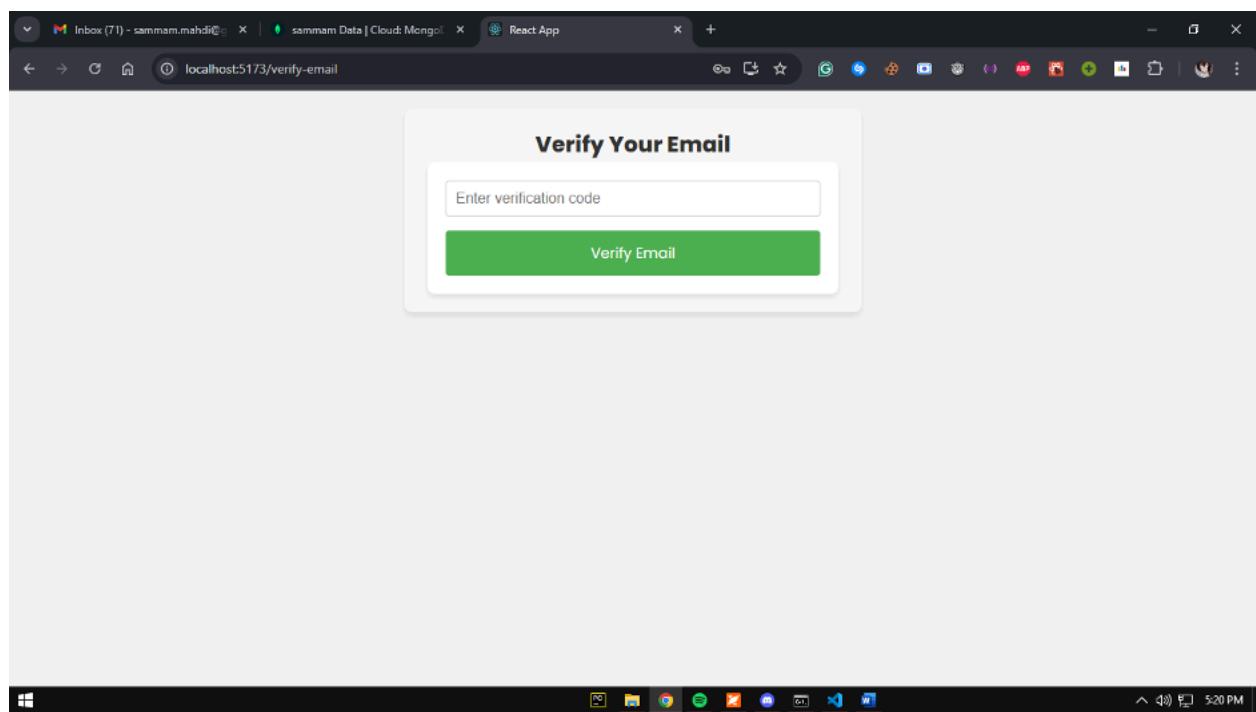


The screenshot shows the Compass interface for a MongoDB database named 'weHeal'. On the left, there's a sidebar with a '+ Create Database' button, a search bar for namespaces, and a tree view showing 'weHeal' expanded, with 'users' selected. The main panel is titled 'weHeal.users' and displays database statistics: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 1.43KB, TOTAL DOCUMENTS: 4, INDEXES TOTAL SIZE: 72KB. Below the stats are navigation tabs for 'Find', 'Indexes', 'Schema Anti-Patterns' (with 0 items), 'Aggregation', and 'Search Indexes'. A button labeled 'INSERT DOCUMENT' is on the right. A query input field says 'Type a query: { field: 'value' }' with 'Filter' and 'Options' buttons. At the bottom, it says 'QUERY RESULTS: 0'.

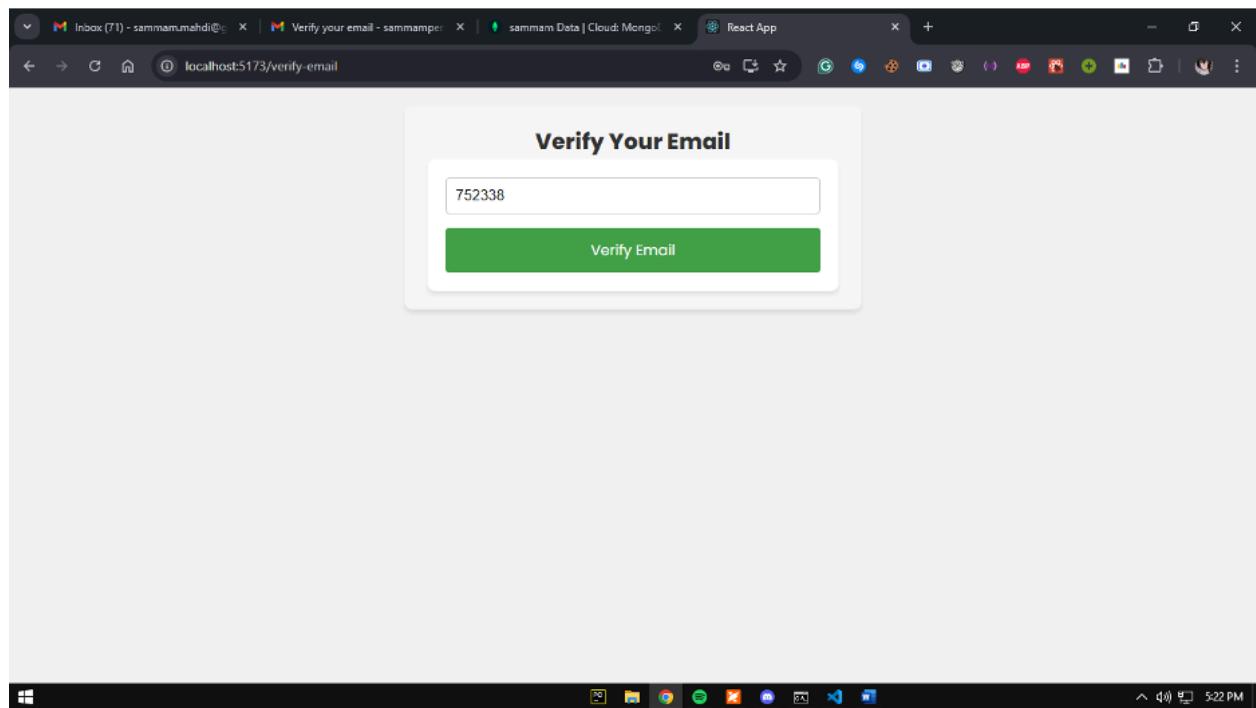
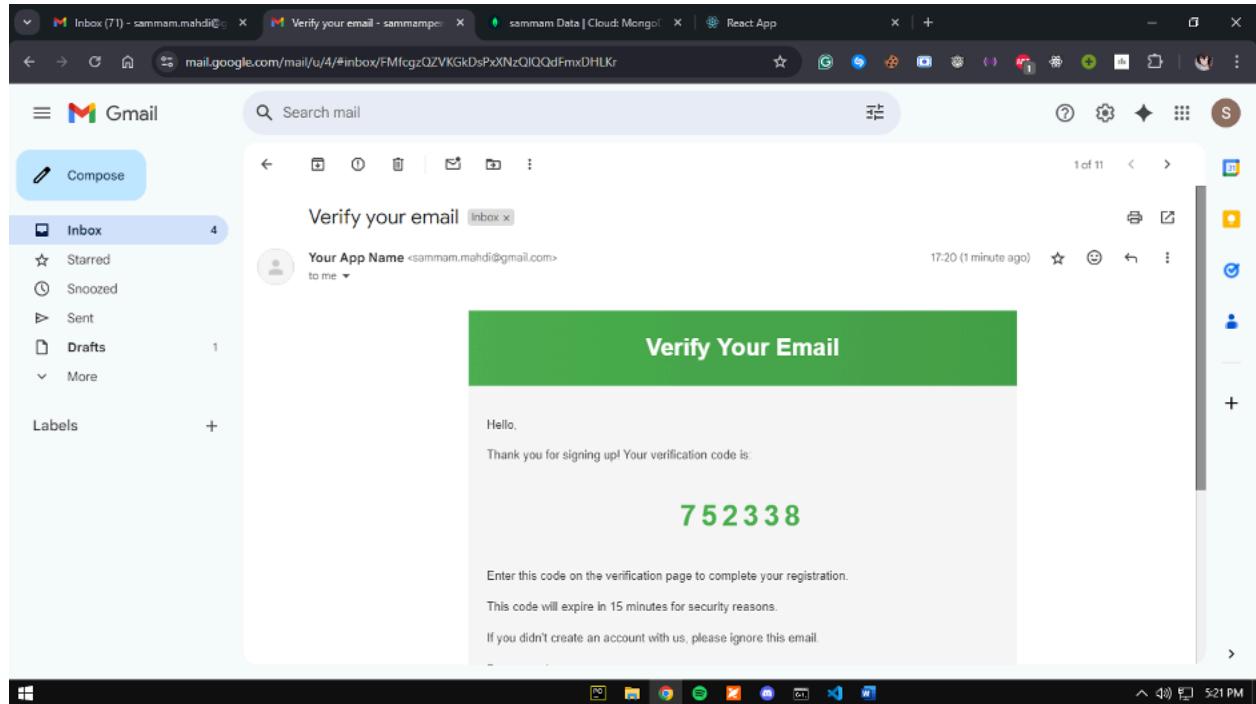
- The user registers

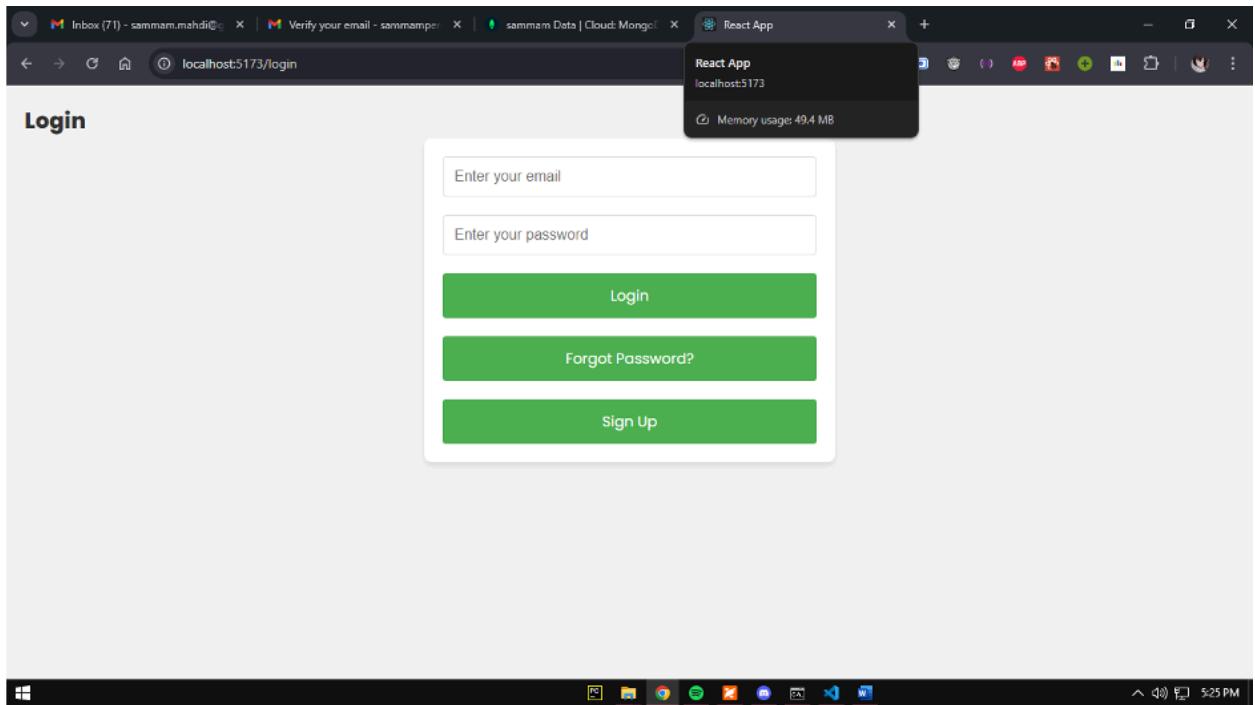


- After the user has given their information, they are redirected to the email verification page (An OTP is sent to the user email). The OTP expires in 1 hour.



- After entering the correct OTP, the user is redirected to the login page and the user information is successfully populated in the database.



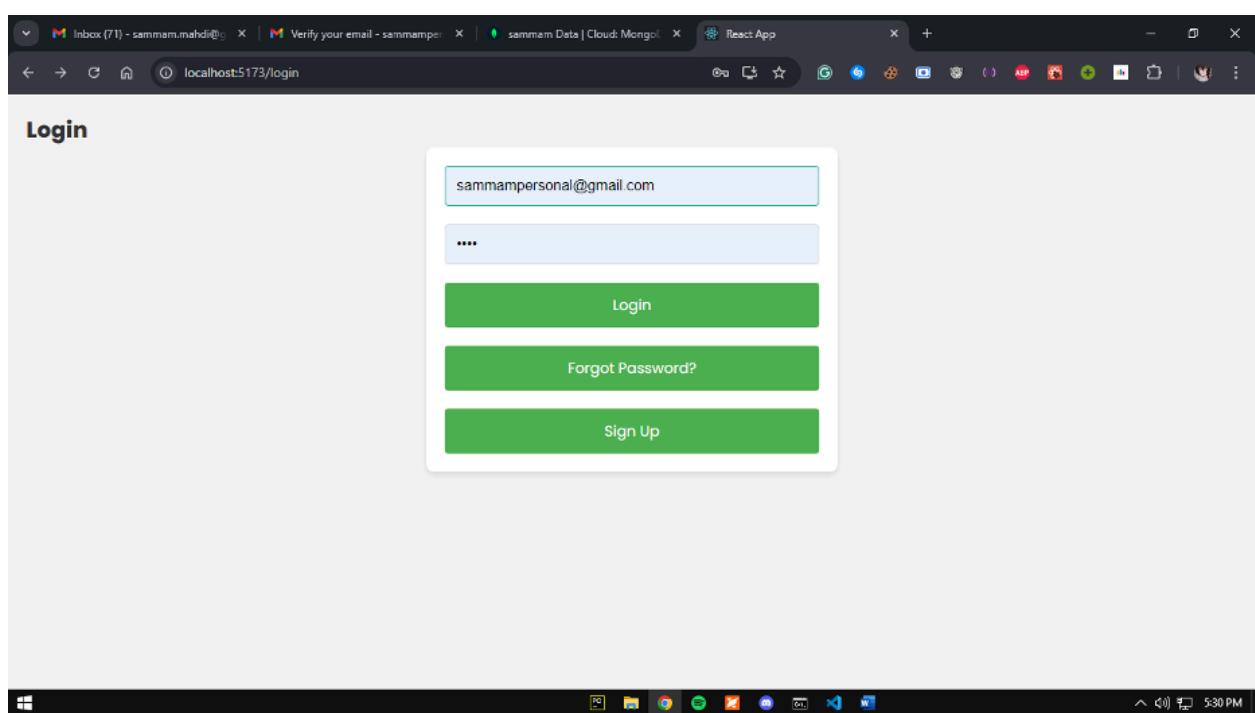
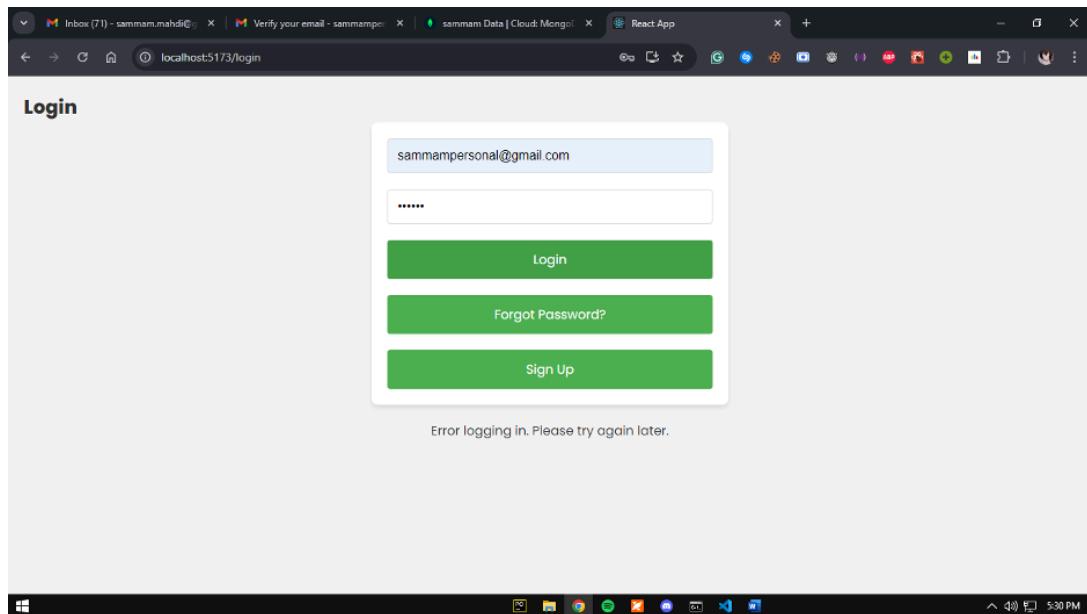


A screenshot of the MongoDB Compass interface. The left sidebar shows a database named "weHeal" with a collection named "users". The main area is titled "weHeal.users" and displays document statistics: STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 346B, TOTAL DOCUMENTS: 1, INDEXES TOTAL SIZE: 72KB. It includes tabs for "Find", "Indexes", "Schema Anti-Patterns", "Aggregation", and "Search Indexes". A search bar at the top says "Generate queries from natural language in Compass". Below it is a "Filter" section with a query input "Type a query: { field: 'value' }" and buttons for "Reset", "Apply", and "Options". The results section is titled "QUERY RESULTS: 1-1 OF 1" and shows a single document with the following data:

```
_id: ObjectId('6802355df29cb392ed1feefe')
name: "Samman "
email: "sammampersonal@gmail.com"
phone: "01822926571"
isVerified: true
verificationExpires: 2025-04-19T11:19:57.053+00:00
role: "Doctor"
password: "$2b$10$IqtnhbArf9wq5WLNGuV45.u14uclCpBc3FTr793KgUlQhBcIOPfuG"
doctorDetails: Object
lastLogin: 2025-04-18T11:19:57.058+00:00
createdAt: 2025-04-18T11:19:57.062+00:00
```

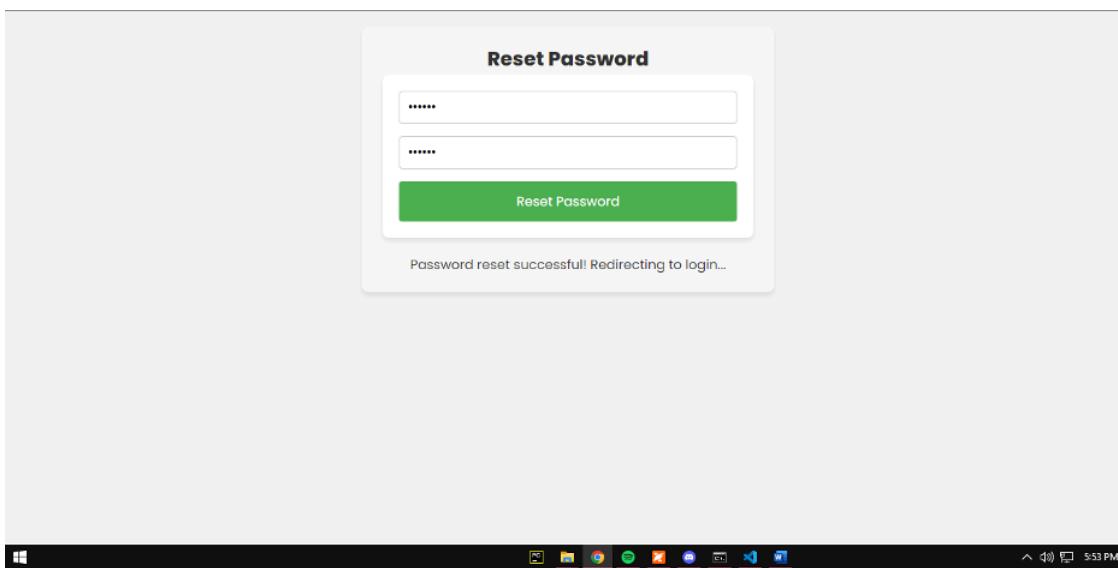
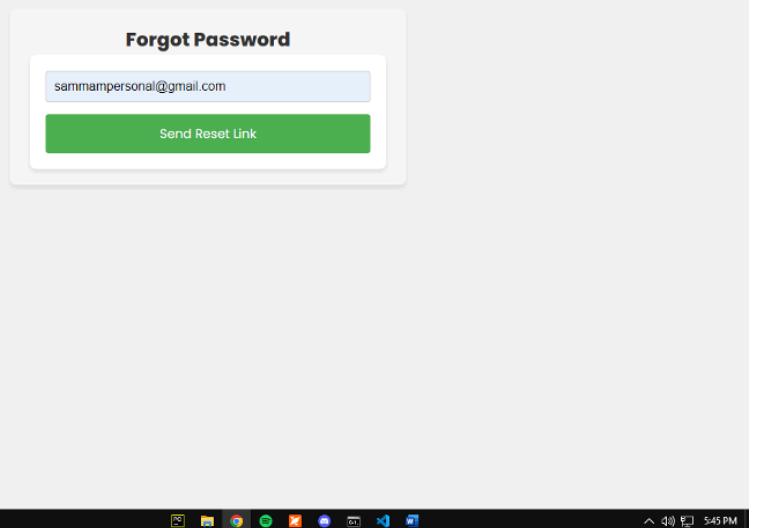
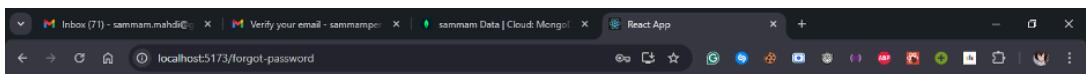
Feature 3: Login

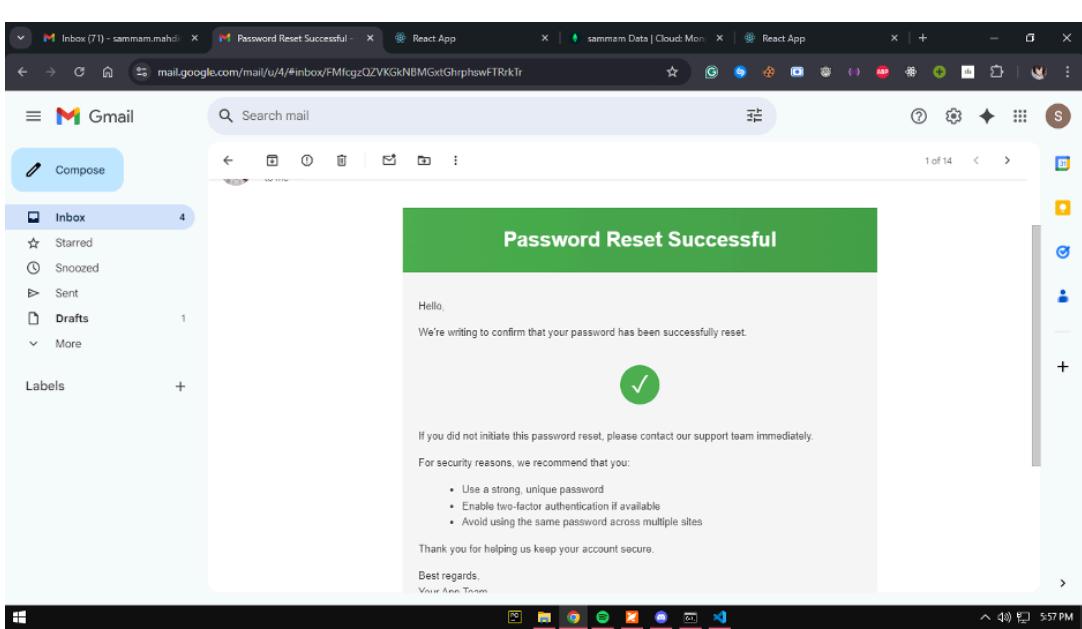
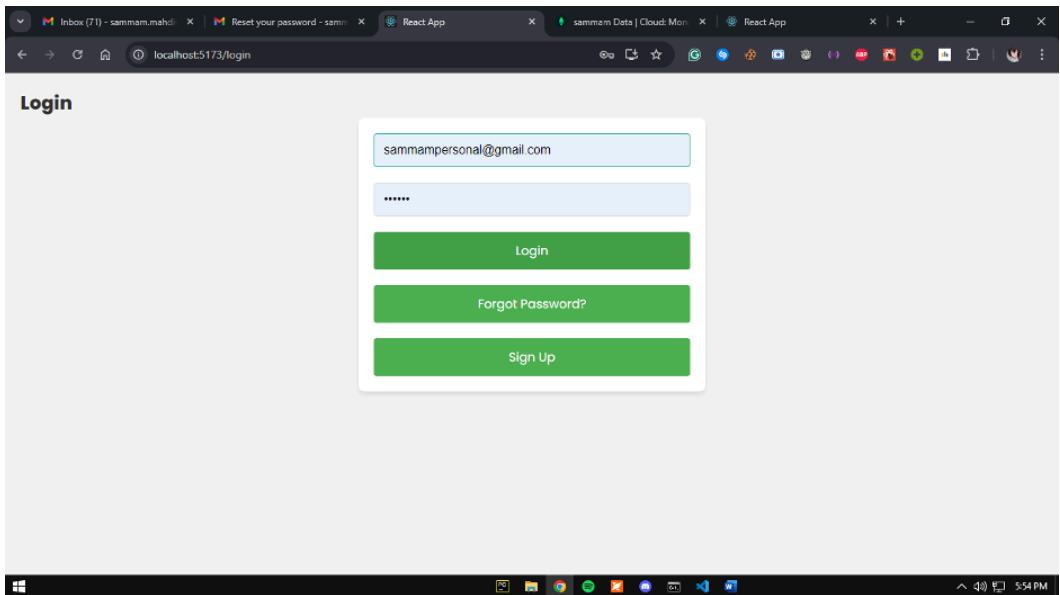
The user will successfully able to login to the website if they provide the correct credentials else, they will be denied access to the website.



Feature 4: Forgot Password and Password Reset

If the user forgets their password, then they can click on the forget password button and go to the forgot password page and if they have provided their email correctly then a password reset email will be sent to their email that has a password reset link with a specific token. If only the token is correct only then the user may successfully reset their password the token also has an expiration deadline of 1 hour.





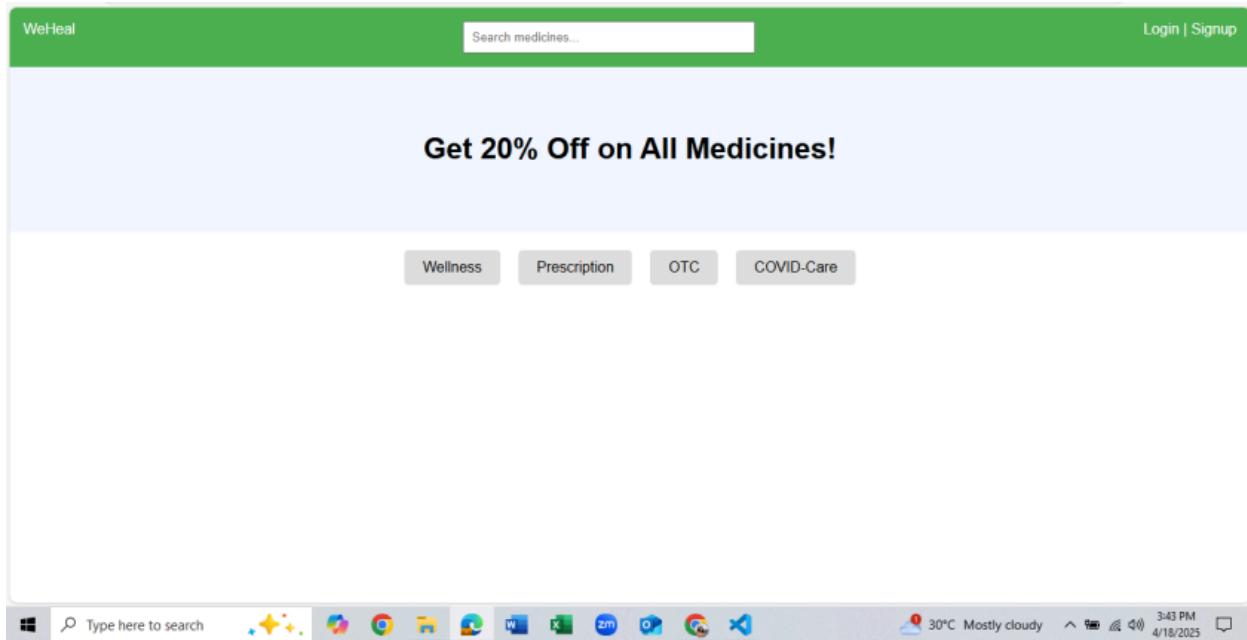
Feature 5: User-end dashboard

a) Patient Dashboard

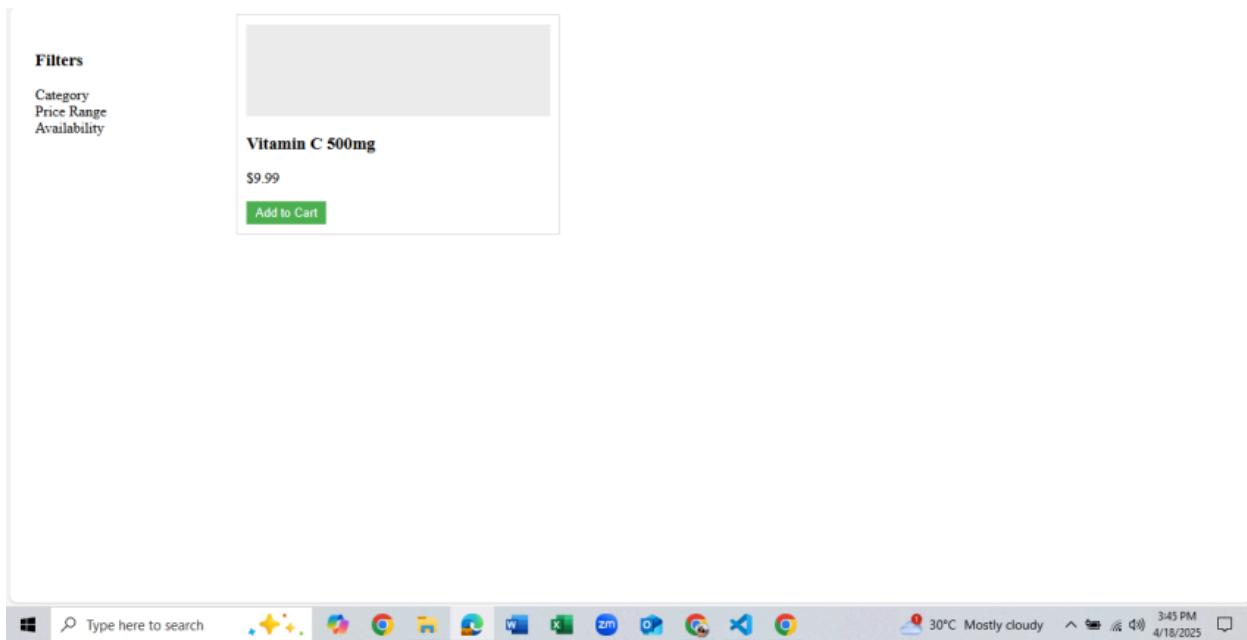
The screenshot shows the WeHeal Patient Dashboard. At the top, there's a search bar and a user profile for "Syeda Maliha Tabassum". On the left, a sidebar lists navigation options: Dashboard, Book Appointment, My Appointments, Tele-Consultations, My EMR, Prescriptions, Medicine & Labs, Emergency Services, Payments & Insurance, and Feedback & Ratings. The main content area starts with a "Welcome, Syeda Maliha Tabassum." message. It includes a "Health Snapshot" section with statistics: Total Visits: 12, Outstanding Bills: \$42.50, and Loyalty Points: 120. Below this are sections for "Latest Prescription" (2 prescriptions, View Details), "Notifications & Reminders" (Appointment with Dr. Smith tomorrow at 10:00, Refill Amoxicillin due in 2 days, Lab results ready for CBC test, View All), and "Upcoming Appointments". Two appointments are listed: one with Dr. Smith (In-Person, Apr 20 - 10:00-11:00) and one with Dr. Lee (Tele-Consult, Apr 25 - 14:30-15:00). Each appointment card has a "Join" button. A "Filter appointments..." search bar is above the appointment table. The "Appointment Details" section shows the appointment for Dr. Smith on Apr 20, 2025, from 10:00-11:00. It includes buttons for "Reschedule" and "Cancel". A large green "Join Consultation" button is at the bottom right.

SPRINT 3

Feature 6: Marketplace dashboard



Feature 7: Product list



Feature 8: Cart

Product: Vitamin C 500mg

Price	Quantity
\$9.99	1
Total: \$34.97	

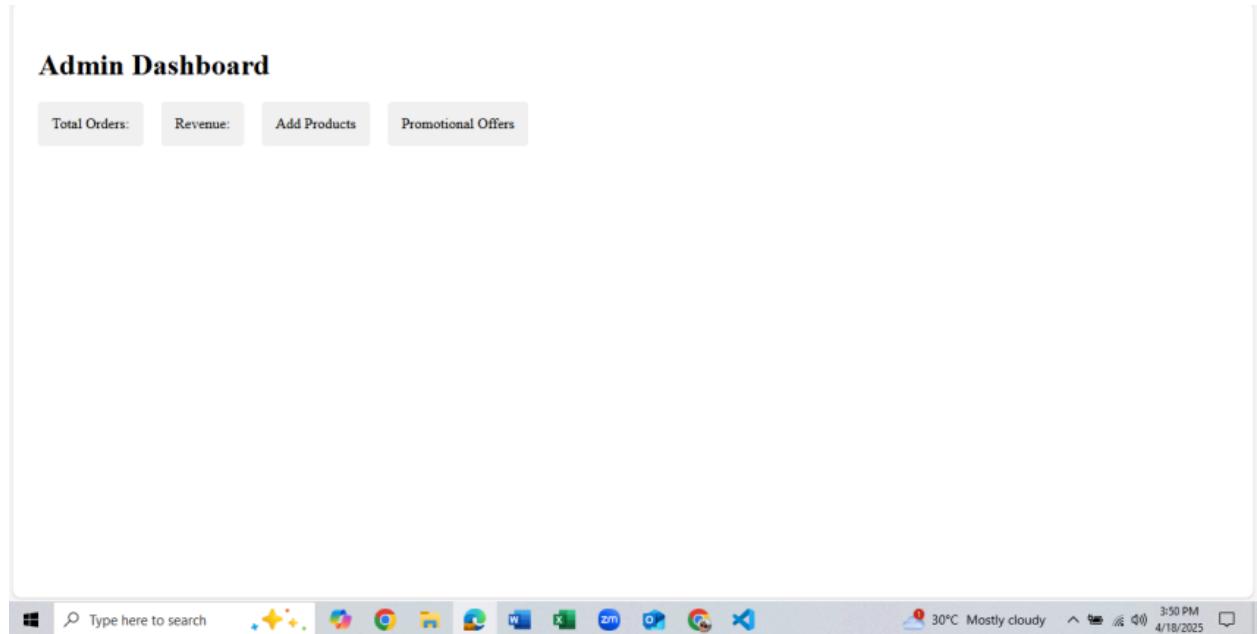
[Proceed to Checkout](#)

Feature 9: Delivery placement

Delivery Address
Please enter your exact address

Payment Method
 Credit Card PayPal

Feature: 10 Admin Dashboard



Sprint 4

Enhanced marketplace: Improvement of features from earlier sprints

The screenshot shows the homepage of the WeHeal marketplace. At the top, there's a navigation bar with links for Home, Medicines, Cart, and Contact. Below the navigation is a search bar with the placeholder "Search medicines...". Underneath the search bar, there are three product cards displayed in a grid:

- Paracetamol 500mg**
Brand: MedLife
Effective for fever and mild pain relief.
₹ 49.00
- Ibuprofen 400mg**
Brand: Healfast
Anti-inflammatory and pain reliever for body aches.
₹ 79.00
- Cetirizine 10mg**
Brand: AllergyFix
Relieves allergies, sneezing and itchy eyes.
₹ 29.00

Each product card has a blue "Add to Cart" button at the bottom.

The screenshot shows a modal window titled "Sort Medicines" on the WeHeal website. The modal includes a dropdown menu labeled "Sort by" with options: Name (selected), Price, and Popularity. Below the dropdown, there are five product cards arranged in two rows:

Medicine	Price	Popularity
Aspirin	₹40	★★★★☆
Paracetamol	₹49	★★★★★
Vitamin C	₹65	★★★★☆
Cetirizine	₹29	★★★★☆
Ibuprofen	₹59	★★★★☆

Your Cart

Paracetamol 500mg

Qty: 2

Price per unit: ₹49.00

₹98.00

Cetirizine 10mg

Qty: 1

Price per unit: ₹29.00

₹29.00

Vitamin C Tablets

Qty: 3

Price per unit: ₹65.00

₹195.00

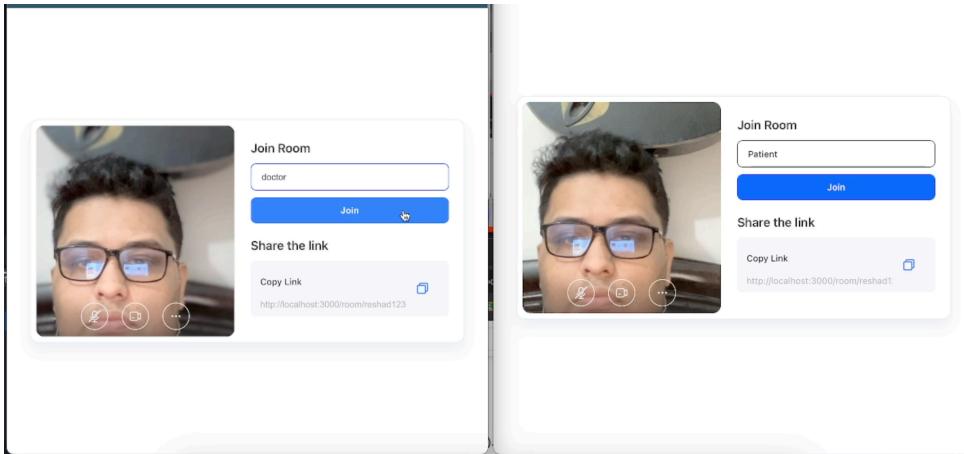
Total: ₹322.00**Proceed to Payment****Add New Medicine****Medicine Name****Price (₹)****Popularity (1-5)****Add Medicine****Medicine Inventory**

Name	Price (₹)	Popularity
Paracetamol	49	★★★★★
Vitamin C	65	★★★★☆



Feature 11: Telemedicine Integration

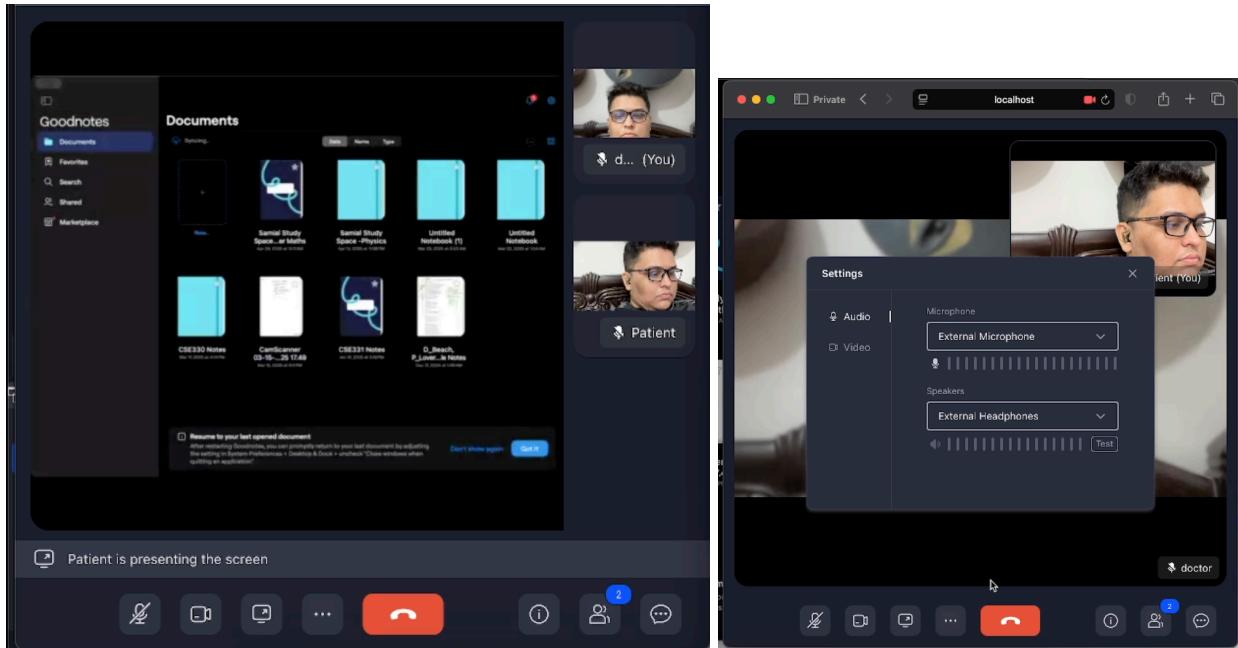
1. End-to-end encrypted video-conference doctor consultation.



2. Convenient UI with in-app chatbox



3. Audio/Video device change option with screen-sharing option for enhanced user experience



Feature 12: E-prescription generation

Symptoms

Enter each symptom on a new line:

Fever
Dry cough

Medicines

Medicine Name	Dosage	Frequency	Duration	Comments
Monas	10 mg	1+0+1	21 Days	Chew
e.g., Paracetamol	e.g., 500mg	e.g., 1+0+1	e.g., 5 days	e.g., After meals

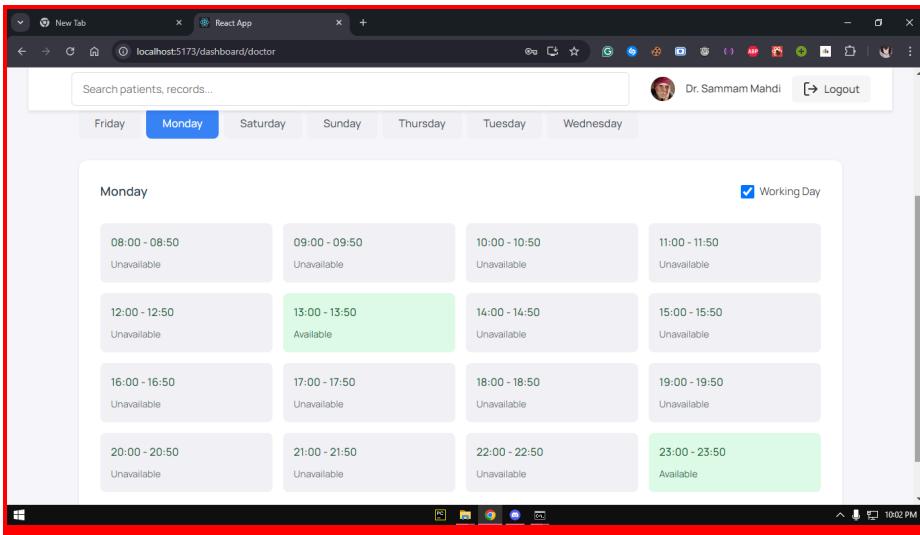
+ Add Another Medicine

Recommended Tests

Enter each test on a new line

Feature 13: Appointment management

- Doctors can set their desired consultation hours



2. Doctors can edit and update their profile

Two side-by-side screenshots of a doctor profile editing interface. Both are from the same browser window, showing different sections of the form. The left screenshot shows fields for 'Name' (Sammam Mahdi) and 'Phone' (01622926571). The right screenshot shows 'Professional Details' with 'Degree' (M.phil), 'Institution' (Bracu), and 'Year' (2024). There is also a 'Remove' button and an 'Add Education' button.

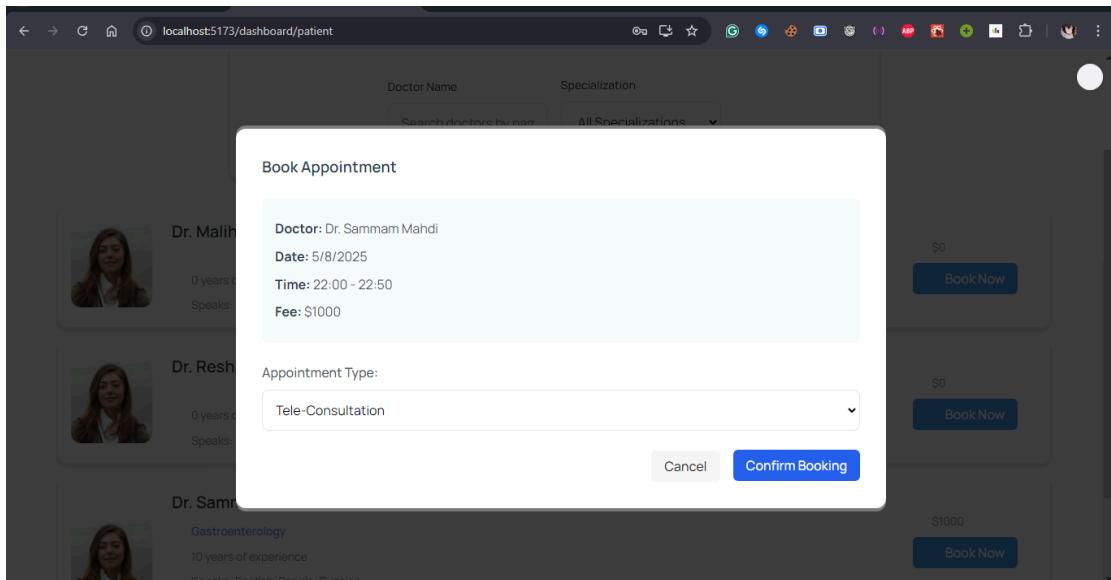
3. patients can find their preferred doctor and select time

The image displays three sequential screenshots of a web-based doctor booking application, likely built with React, showing the process of selecting a doctor and viewing their availability.

Screenshot 1: The main dashboard shows three doctors listed with their names, profiles, experience, and a "Book Now" button. Dr. Mallika Tabsum has 0 years of experience and speaks English. Dr. Reshadul has 0 years of experience and speaks English. Dr. Sammam Mahdi is a Gastroenterologist with 10 years of experience, speaking English, French, and Russian, and has a consultation fee of \$1000.

Screenshot 2: A modal window titled "Dr. Sammam Mahdi's Availability" is open. It shows a date picker set to May 2025. A message says "Please select a date to view availability". Below the calendar, it shows "Available slots for 5/8/2025" and "ConsultationFee: \$1000".

Screenshot 3: The same modal window is shown again, but now it displays two time slots: "22:00 - 22:50" and "23:00 - 23:50", each with a "Select" button.



A screenshot of a patient dashboard at localhost:5173/dashboard/patient. The page includes the following sections:

Find Doctor, **Medical Records**, **Payments**

Upcoming Appointments section:

- Appointment 1:** Dr. Sammam Mahdi (Gastroenterology)
 - Invalid Date
 - 08:00 - 08:50
 - In-person Visit
 - Scheduled
 - Cancel
- Appointment 2:** Dr. Sammam Mahdi (Gastroenterology)
 - Invalid Date
 - 22:00 - 22:50
 - Tele-consultation
 - Scheduled
 - Join Call
 - Cancel

4. Admins can edit, delete and search users (Enhanced Admin Dashboard)

	Name	Email	Role	Status	Actions
 easha	easha	syedamaliatabassum19@gmail.com	Patient	Verified	View Details Delete
 M	Maliha tabssum	syeda.maliha.tabassum1@g.bracu.ac.bd	Doctor	Verified	View Details Delete
 S	Sammam Mahdi	sammam.mahdi@gmail.com	Patient	Verified	View Details Delete
 S	Sammam Mahdi	sammampersonal@gmail.com	Doctor	Verified	View Details Delete
 A	Alisha Shenaz	shenaz.alisha02@gmail.com	ClinicStaff	Verified	View Details Delete
 S	Sammam Admin	sammam.mahdi@g.bracu.ac.bd	Admin	Verified	View Details Delete
 R	Reshadul	reshad.ul.karim@g.bracu.ac.bd	Doctor	Verified	View Details Delete
 R	Reshad Sazid	reshad.sazid@gmail.com	Patient	Verified	View Details Delete

New Tab React App localhost:5173/dashboard/admin

localhost:5173/dashboard/admin/user/68139060f243fc8cf8dd805d

Back

User Details		Patient Details	
Name:	easha	Patient ID:	None
Email:	syedamaliatabassum19@gmail.com	Date of Birth:	None
Role:	Patient	Address:	None
Status:	Verified	Insurance Details:	None
Phone:	012345678		
Created:	5/1/2025, 9:16:48 PM		
Last Login:	5/1/2025, 9:20:58 PM		

Edit [Delete User](#)

The screenshot shows a web browser window with a URL of `localhost:5173/dashboard/admin/user/68139060f243fc8cf8dd805d`. The main content area displays a 'User Details' form with fields for Name (easha69), Email (syedamalihabassum19@gmail.com), Role (Patient), Status (Verified), Phone (012345678), Created (5/1/2025, 9:16:48 PM), and Last Login (5/1/2025, 9:20:58 PM). A 'Save' and 'Cancel' button are at the top right. To the right, a sidebar titled 'Patient Details' contains fields for Patient ID, Date of Birth (mm/dd/yyyy), Address, and Insurance Details.

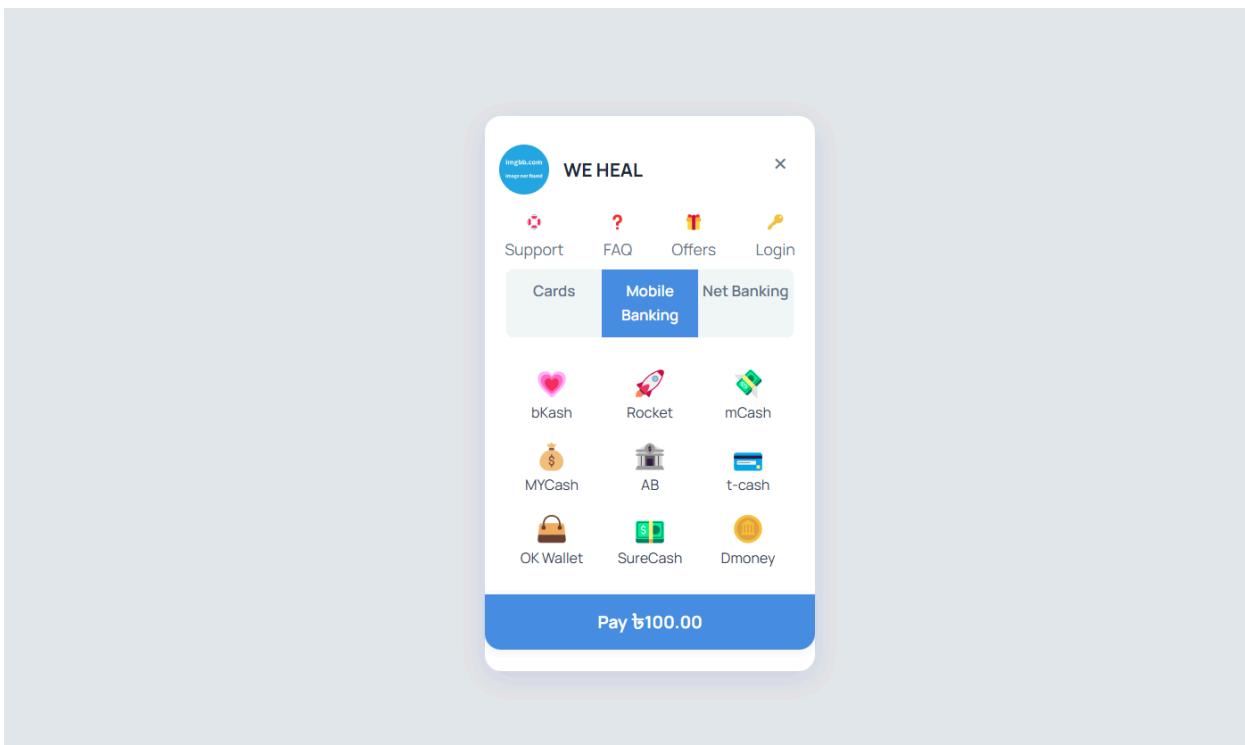
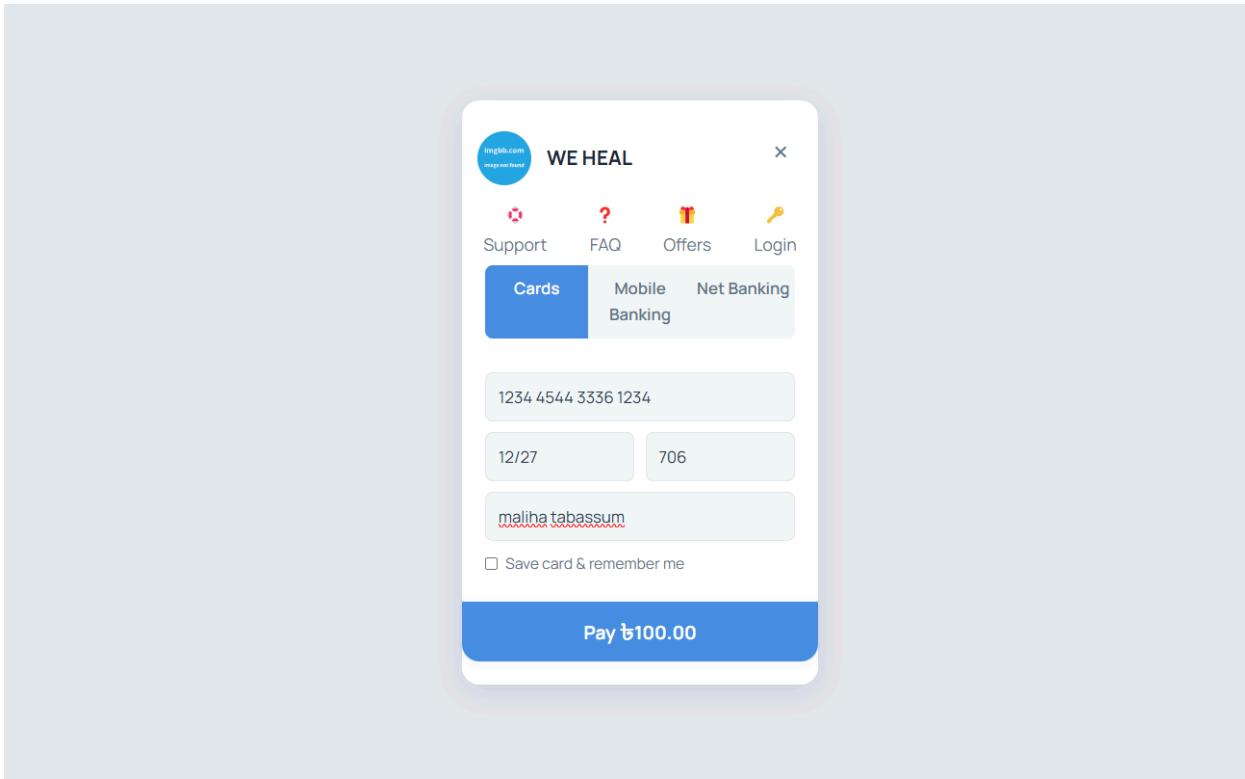
Feature 14: Track Order

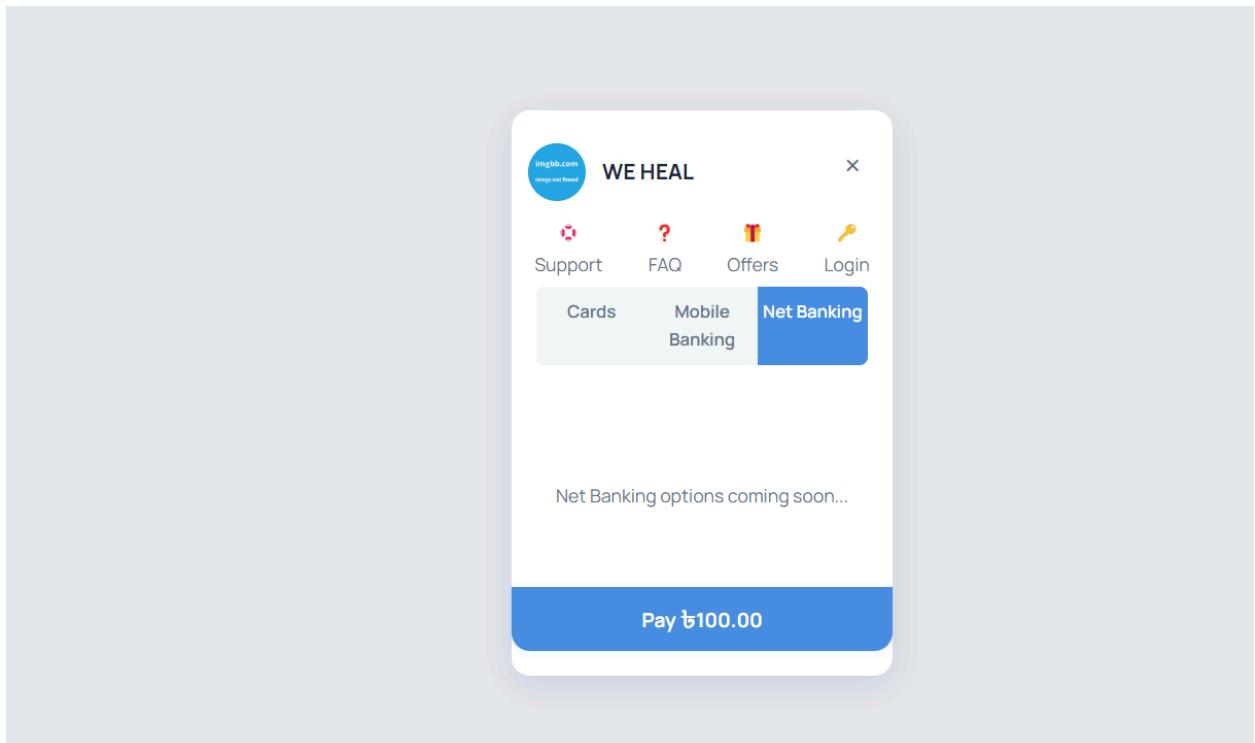
The screenshot shows the WeHeal mobile application interface. At the top, there are navigation links for 'Medicines', 'Cart', and 'Track'. The main content area is titled 'Track Your Order' and displays the following information:
Order ID: WH123456789
Items: Paracetamol 500mg (2), Vitamin C (3)
Expected Delivery: May 3, 2025
A vertical list of order status steps is shown:

- Order Placed - April 30, 2025
- Dispatched from Warehouse - April 30, 2025
- Out for Delivery - May 1, 2025**
- Delivered - Pending

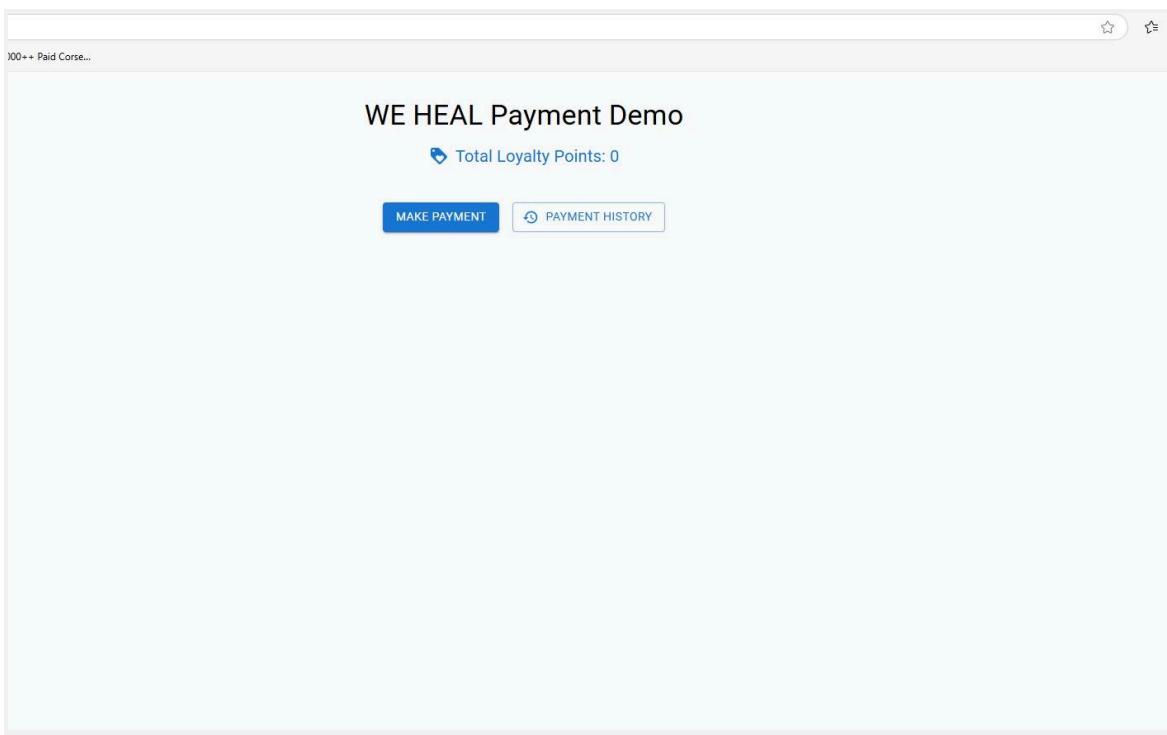
A note at the bottom states: "You will receive a notification once your order is delivered. For any assistance, contact our support team." The bottom of the screen shows the Windows taskbar with various pinned icons and the date/time as 5/1/2025, 2:57 PM.

Feature 15: Payment gateway initialization





Feature 16:Loyalty point addition after purchase



WE HEAL Payment Demo

>Total Loyalty Points: 0

WE HEAL Payment
X

Amount to Pay: ₩1000
Payment Type: medicine

MOBILE BANKING
CARD PAYMENT
ONLINE BANKING

Select Mobile Banking Method


bKash


Nagad


Q Cash

Nagad Number:

PIN:

You will earn 20 loyalty points for this purchase.

CANCEL
PAY NOW

WE HEAL Payment Demo

Total Loyalty Points: 20

MAKE PAYMENT
PAYMENT HISTORY

Last Payment Details

Order ID: WH-hurvrw45p
Status: completed

Order ID	Date	Amount	Method	Type	Points	Status
WH-hurvrw45p	5/11/2025, 9:56:39 PM	₩1,000	mobile	medicine	20	completed

CLOSE

Feature 17: Prescription generation and download

Create Prescription

Patient & Doctor Info

Patient Name *

Doctor Name *

Date *

05/11/2025

Symptoms

Enter each symptom on a new line

e.g. Fever
Dry cough

Medicines

Medicine Name	Dosage	Frequency	Duration	Comments
e.g. Paracetamol	e.g. 500mg	e.g. 1+0+1	e.g. 5 days	e.g. After meals

+ Add Another Medicine

Recommended Tests

Enter each test on a new line

e.g. Chest X-Ray
Blood Test

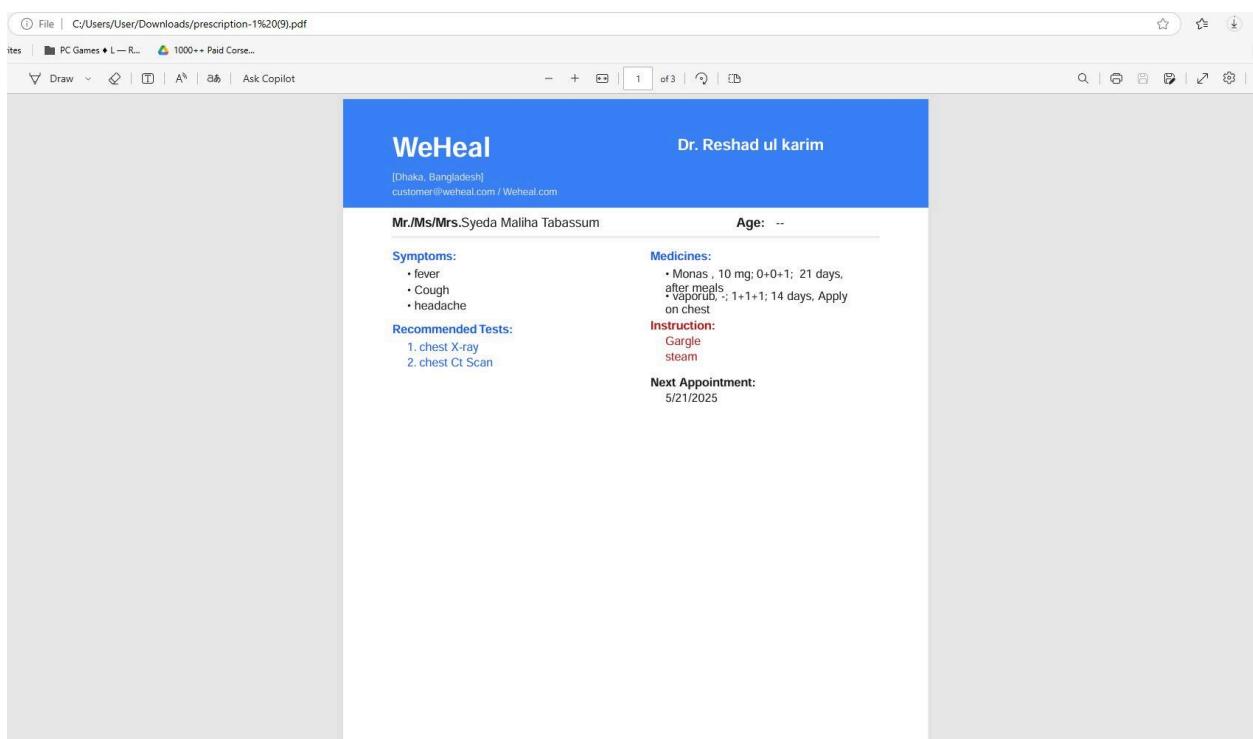
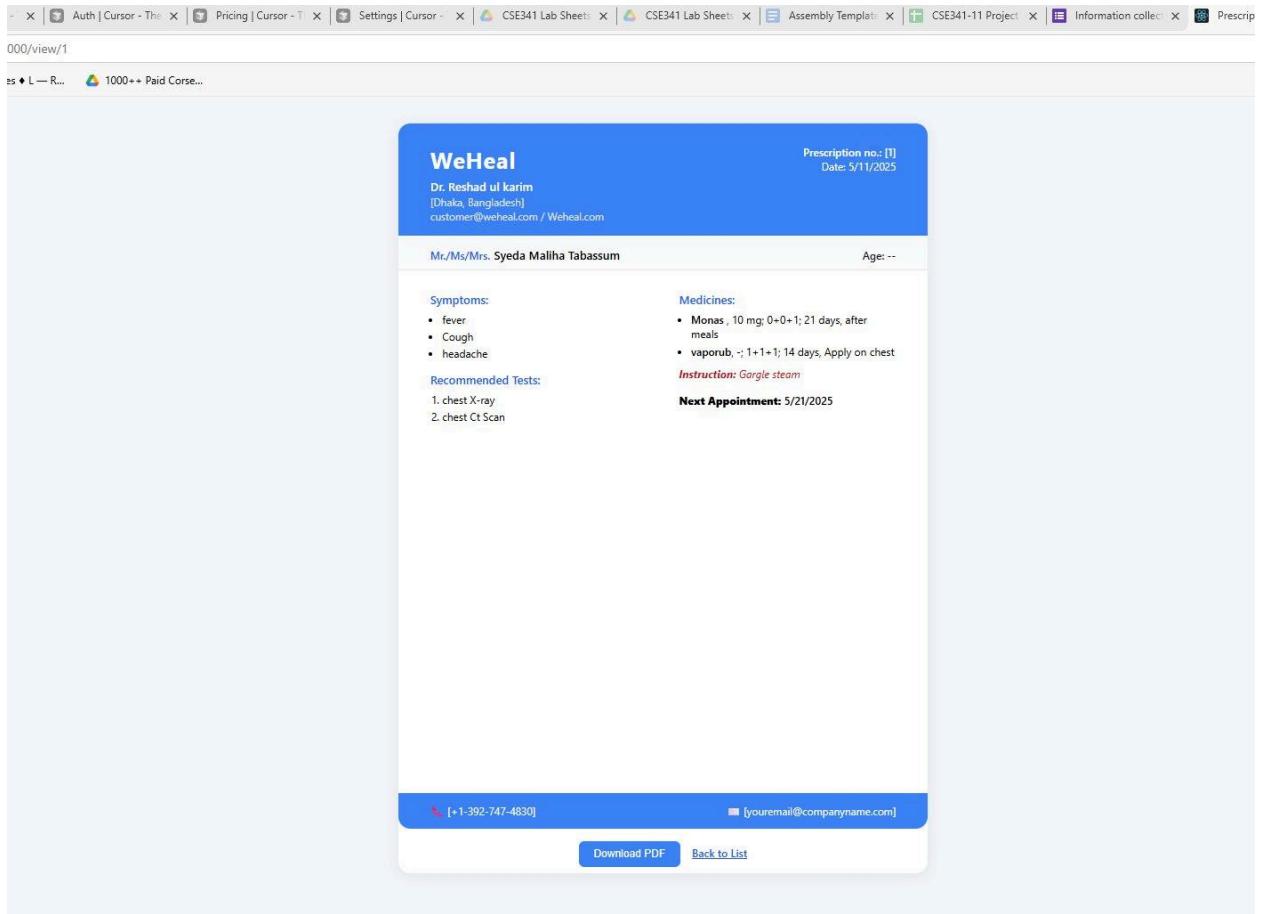
Next Appointment

Next Visit Date

mm/dd/yyyy

Extra Instructions

e.g. Drink plenty of water, rest, etc.



Feature 18: Prescription history

The screenshot shows a user interface titled "All Prescriptions". At the top left is a blue button labeled "Create New Prescription". To its right are two search fields: one for "Search by patient name, doctor" and another for "mm/dd/yyyy" with a calendar icon. Below these is a dropdown menu set to "Newest First". A large rectangular box contains patient information: "Syeda Maliha Tabassum", "Doctor: Reshad ul karim", "Date: 5/11/2025", and "Medications: 2 items".

Feature 19: Payment history

The screenshot shows a payment interface for "WE HEAL Payment Demo". It features a "Total Loyalty Points: 20" badge and two buttons: "MAKE PAYMENT" and "PAYMENT HISTORY". A modal window titled "Payment History" is displayed in the foreground. The modal header says "Last Payment Details" and shows "Order ID: WH-hurvrw45p" and "Status: completed". The main table in the modal lists a single payment row:

Order ID	Date	Amount	Method	Type	Points	Status
WH-hurvrw45p	5/11/2025, 9:56:39 PM	₹1,000	mobile	medicine	20	completed

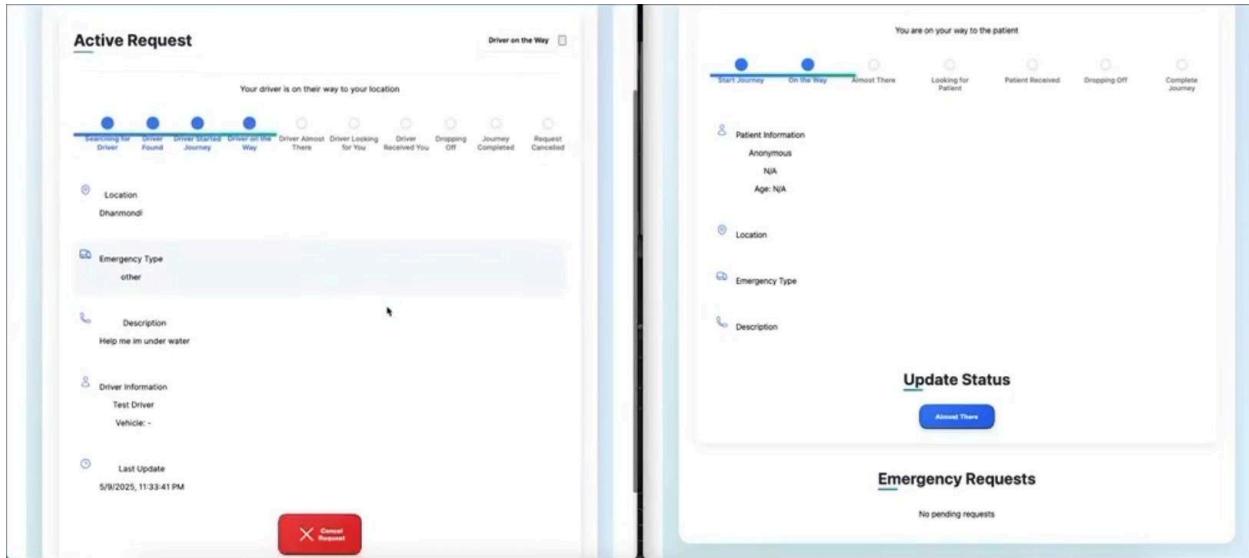
A "CLOSE" button is at the bottom right of the modal.

Feature 20: Ambulance request

The image shows two browser windows side-by-side. The left window is titled 'Emergency Service' and 'Active Request'. It displays a status message 'We are searching for a driver to assist you' and a timeline of events: 'Searching Driver for Driver Found', 'Driver Started Journey', 'Driver on the Way', 'Driver Almost There', 'Driver Looking for You', 'Driver Received You', 'Dropping Off', 'Journey Completed', and 'Request Cancelled'. Below this, it shows a location 'Dhanmondi' and an emergency type 'other'. The right window is titled 'Driver Dashboard' and 'Emergency Requests'. It shows a pending request from a patient named 'Anonymous' at a location 'Dhanmondi' with an emergency type 'other'. The status is 'pending' and the time is '11:33:22 PM'. There are 'Accept' and 'Reject' buttons.

This screenshot is similar to the one above but shows a cancellation. The Patient Dashboard now has a red 'Cancel Request' button at the bottom. The Driver Dashboard still shows the pending request from 'Anonymous' at 'Dhanmondi' with an emergency type 'other'.

Feature 21: Realtime ambulance status update and tracking



Feature 22: Hosting for realtime video conference

