

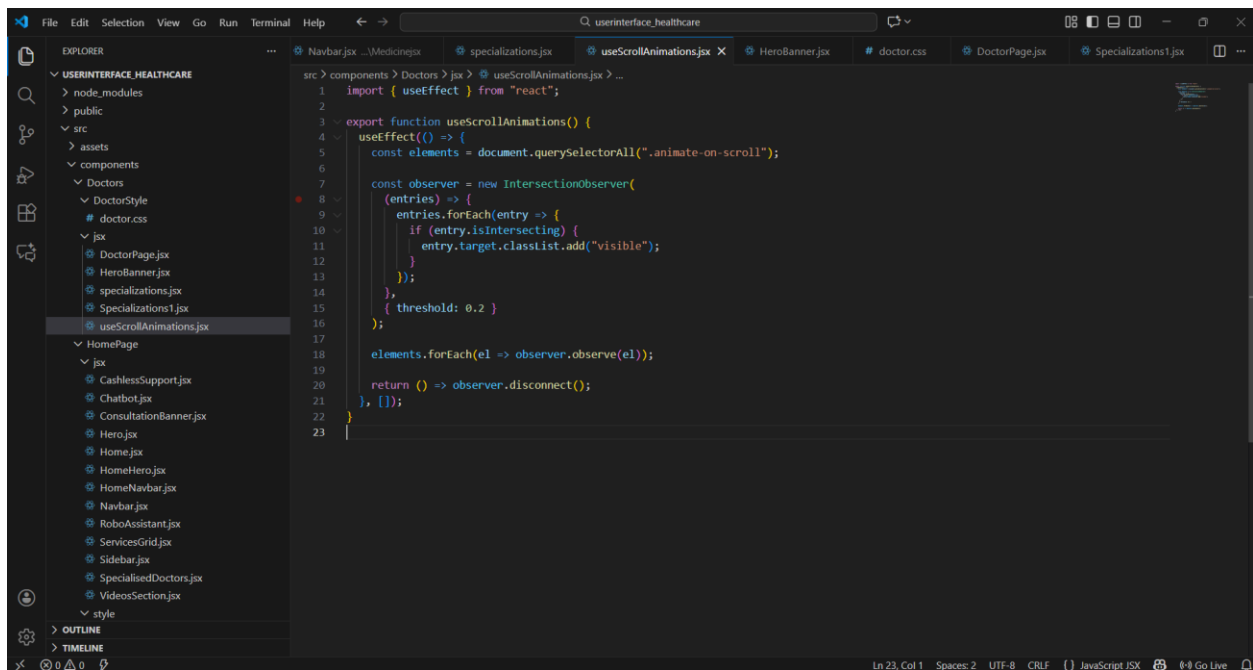
Daily Status Report

Name : M. Eswari

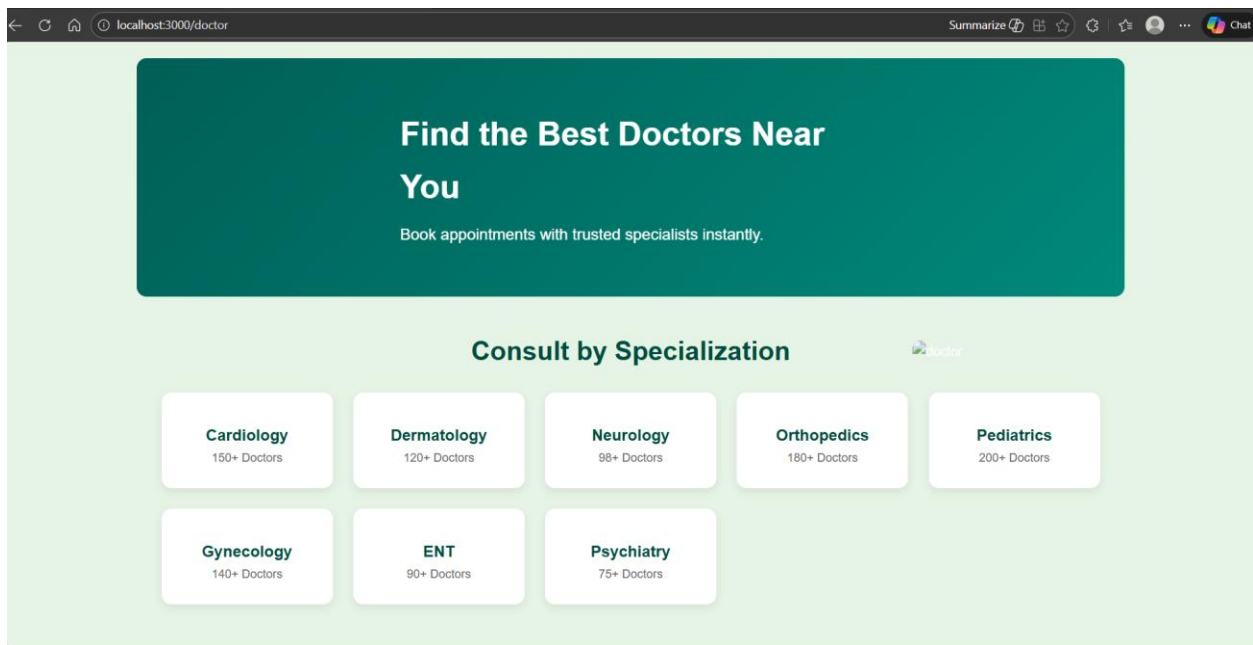
Date : 09-12-2025

Today I worked on the healthcare user interface, improving both the Doctor page and the Lab Test page for a smoother experience. The updates enhanced layout consistency, added better interactions, and improved overall usability across both modules.

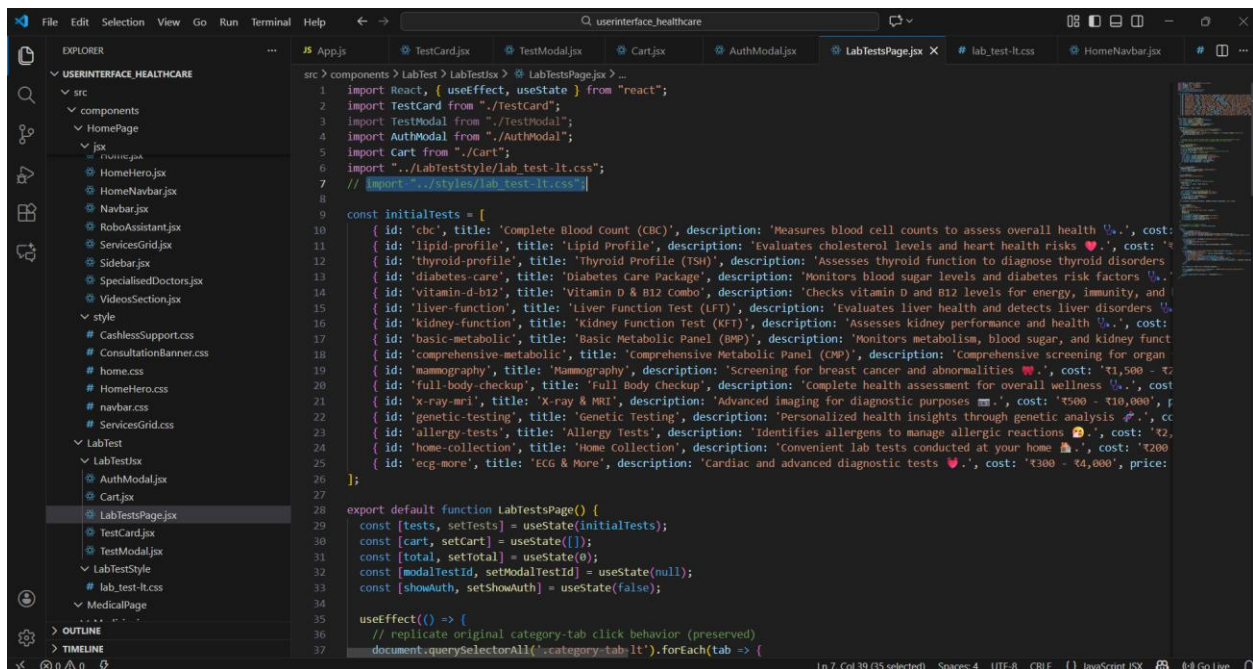
The Doctor module displays available doctors with their specialties, experience, and consultation details in a clean card layout. Users can quickly view profiles, book appointments, and access essential information for informed healthcare decisions.



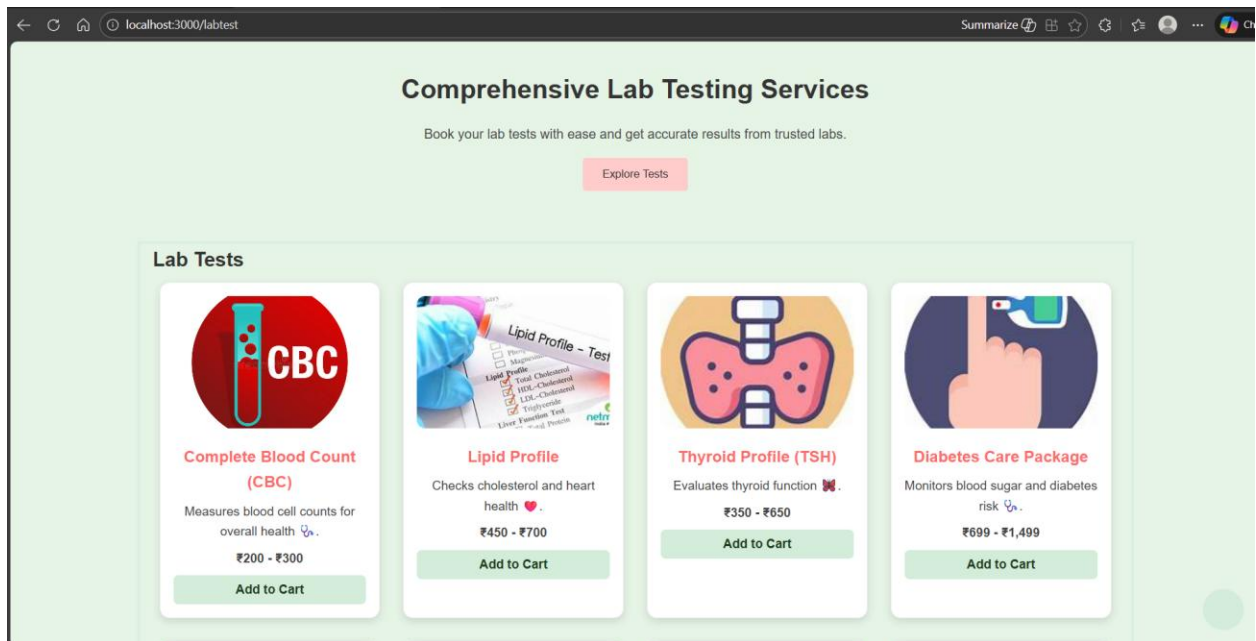
```
1 import { useEffect } from "react";
2
3 export function useScrollAnimations() {
4   useEffect(() => {
5     const elements = document.querySelectorAll(".animate-on-scroll");
6
7     const observer = new IntersectionObserver(
8       (entries) => {
9         entries.forEach(entry => {
10           if (entry.isIntersecting) {
11             entry.target.classList.add("visible");
12           }
13         });
14       },
15       { threshold: 0.2 }
16     );
17     elements.forEach(e1 => observer.observe(e1));
18
19     return () => observer.disconnect();
20   }, []);
21 }
22
23
```



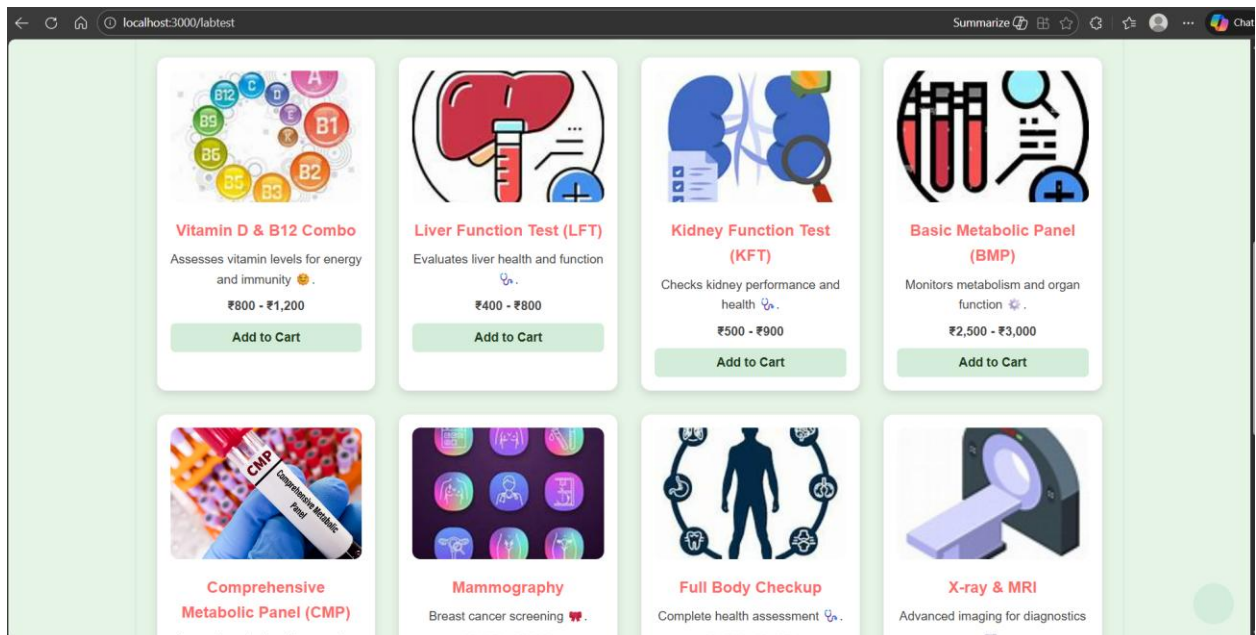
A reusable card component that displays each lab test with image, title, price, and short description. Handles clicking to open the modal and adding items to the cart without affecting parent layout.



Shows full details of a selected lab test inside a popup modal. Allows users to book the test and closes smoothly via the parent modal handler.



Handles Login, Signup, Forgot Password, OTP verification, and Captcha inside one modal. Uses React state and DOM elements to mimic your original flow while keeping all behaviors intact.



Main page that loads all test cards, handles modals, manages cart functionality, and user interactions. Connects all components into a complete, functional lab test booking screen.

```
File Edit Selection View Go Run Terminal Help
src > components > LabTest > LabTest.js > AuthModal.js > ...
1 import React, { useEffect, useRef, useState } from "react";
2
3 export default function AuthModal({ show, onClose }) {
4   const [isLoading, setIsLoading] = useState(true);
5   const [message, setMessage] = useState("");
6   const [otpVisible, setOtpVisible] = useState(false);
7   const captchaCanvasRef = useRef(null);
8
9   useEffect(() => {
10     if (show && !isLoading) generateCaptcha();
11     // eslint-disable-next-line
12   }, [show, !isLoading]);
13
14   useEffect(() => {
15     if (typeof window !== "undefined" && window.flatpickr) {
16       try {
17         window.flatpickr("#dob", { dateFormat: "Y-m-d", maxDate: "today" });
18       } catch (e) {}
19     }
20   }, []);
21
22   function generateCaptcha() {
23     const canvas = captchaCanvasRef.current || document.getElementById("captchaCanvas");
24     if (!canvas) return;
25     const ctx = canvas.getContext("2d");
26     ctx.clearRect(0, 0, canvas.width, canvas.height);
27     ctx.fillStyle = "#f0f0f0";
28     ctx.fillRect(0, 0, canvas.width, canvas.height);
29     ctx.font = "24px Arial";
30     ctx.fillStyle = "#000000";
31     const chars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
32     let captcha = "";
33     for (let i = 0; i < 6; i++) {
34       captcha += chars.charAt(Math.floor(Math.random() * chars.length));
35     }
36     ctx.fillText(captcha, 20, 40);
37     for (let i = 0; i < 5; i++) {
```

