**Team 7 - Machas United  
Problem Statement: Doctor’s assistant.**

**Prompt:** Design a Doctor’s Assistant AI that interacts with physicians to describe incoming patients. Integrate seamlessly with hospital databases (FHIR, HL7) to access patient history, conditions, medications, and recent test results. Enable real-time conversational AI for doctor queries, ensuring accurate and efficient decision-making.  
  
**Result:** I'm thinking of creating an elegant, minimalist Doctor's Assistant AI interface inspired by medical design principles and Apple's design philosophy. This will be a powerful but visually calm tool that helps physicians quickly access and understand patient information.

For this first version, I'll implement:

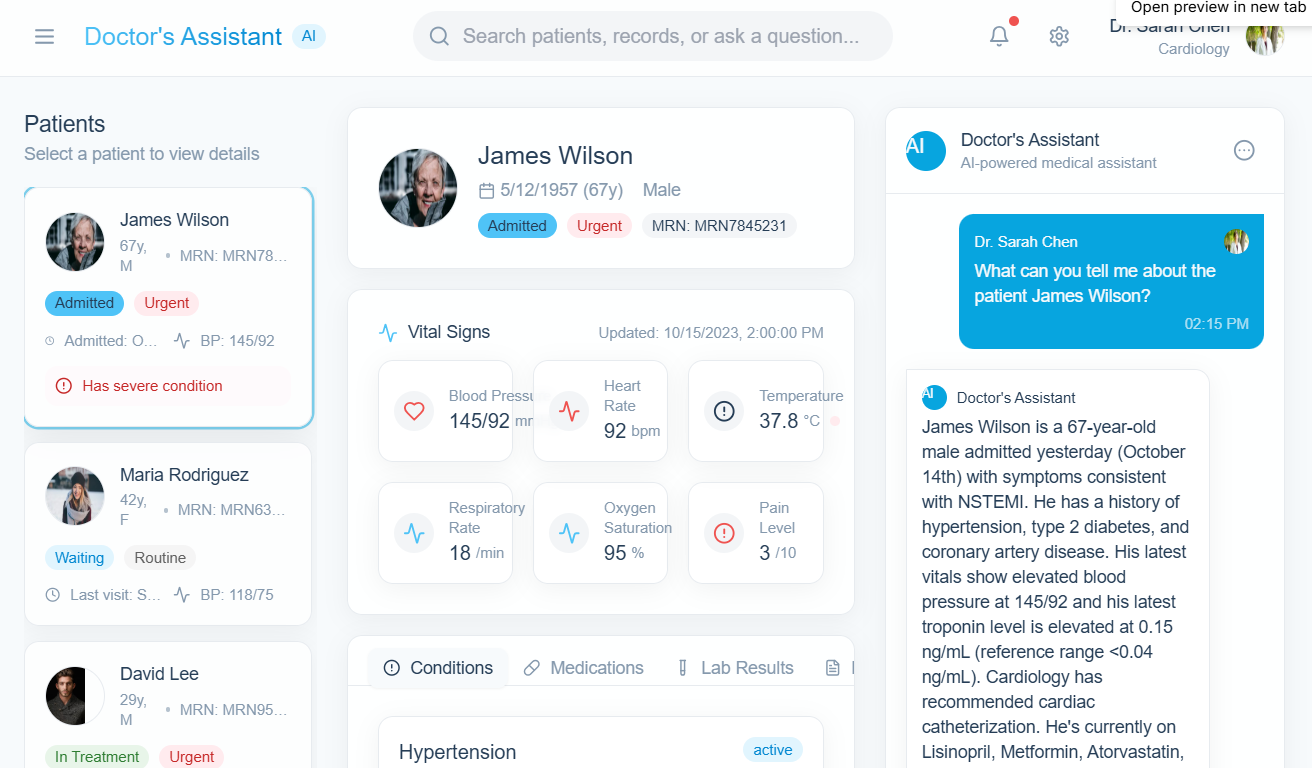
* A clean, minimalist interface with subtle animations
* A conversational AI chat interface
* Patient information display with key vitals and history
* Mock integration points for hospital data (simulated for now)
* Responsive design for desktop and mobile use

For the design, I'll use:

* Colors: Clean whites, soft blues, and minimal accent colors
* Typography: San Francisco or similar clean sans-serif fonts
* Animations: Subtle, purposeful transitions and microinteractions
* Layout: Generous whitespace with clear information hierarchy
* Glass morphism: For key information cards with soft shadows

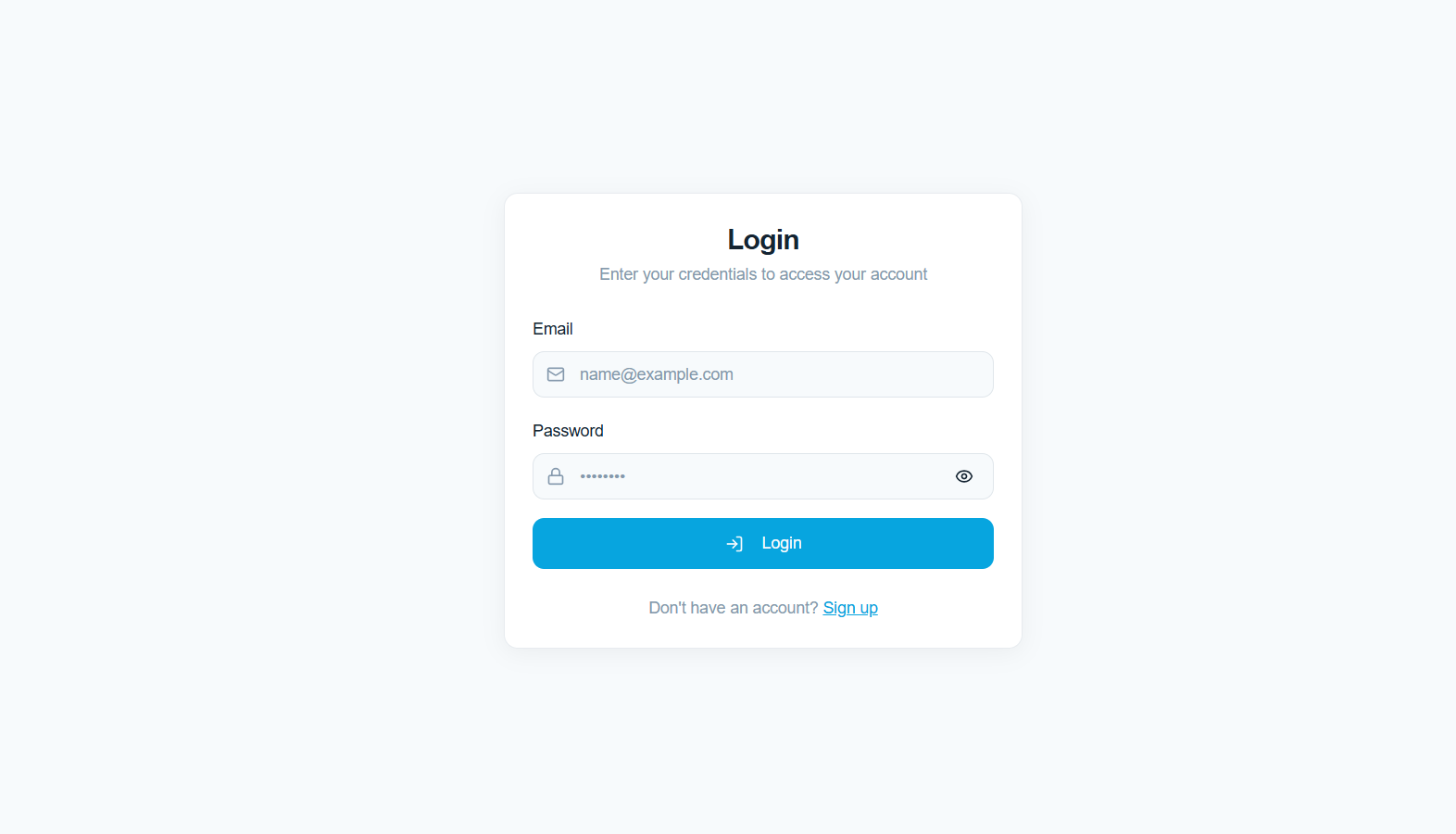
**Error:**Uncaught SyntaxError: The requested module '/node\_modules/.vite/deps/lucide-react.js?v=f8dde2ca' does not provide an export named 'Flask' { "colno": 59, "message": "Uncaught SyntaxError: The requested module '/node\_modules/.vite/deps/lucide-react.js?v=f8dde2ca' does not provide an export named 'Flask'", "error\_type": "runtime", "lineno": 14, "stack": "SyntaxError: The requested module '/node\_modules/.vite/deps/lucide-react.js?v=f8dde2ca' does not provide an export named 'Flask'", "blankScreen": true, "filename": "https://663a70de-93ab-4824-9799-781ce57d0444.lovableproject.com/src/components/PatientInfo.tsx" }

**Reolving prompt:** For the code present, I get the error below. Please think step-by-step in order to resolve it.  
{ERROR SPECIFIED HERE}  
  
**Result**: I'll help you resolve the error. The issue is that the Flask icon is not available in the lucide-react library. Let's fix this by replacing it with a compatible icon.  
I've replaced the unavailable Flask icon from lucide-react with TestTube, which is a better available alternative for lab-related UI elements. I've also updated the TransitionProps interface to include the className property and modified the AnimatedTransition component to accept and apply this className prop properly. These changes will fix all the reported type errors.



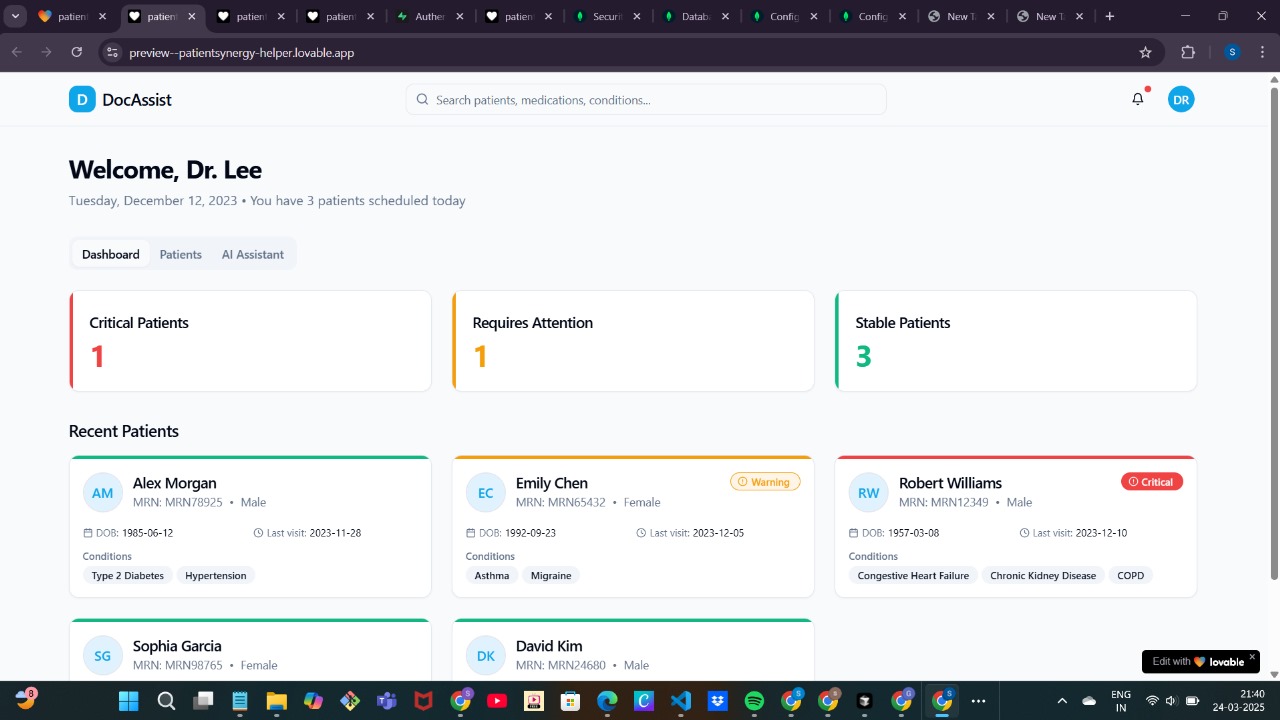
**Prompt:** Create a login and sign up page, and improve the UI keeping the current features intact. Make the UI minimalistic and simple

**Result:** I'll create login and signup pages with a minimalistic UI while preserving the current features. Let's implement this.  
I've created login and signup pages with a clean, minimalistic design that maintains your current functionality. The pages include form validation, password visibility toggle, and smooth animations. I've also updated the Header to include login/signup links and made the MainLayout simpler with a fade animation. The UI now has a more modern and cohesive look while keeping all existing features intact.



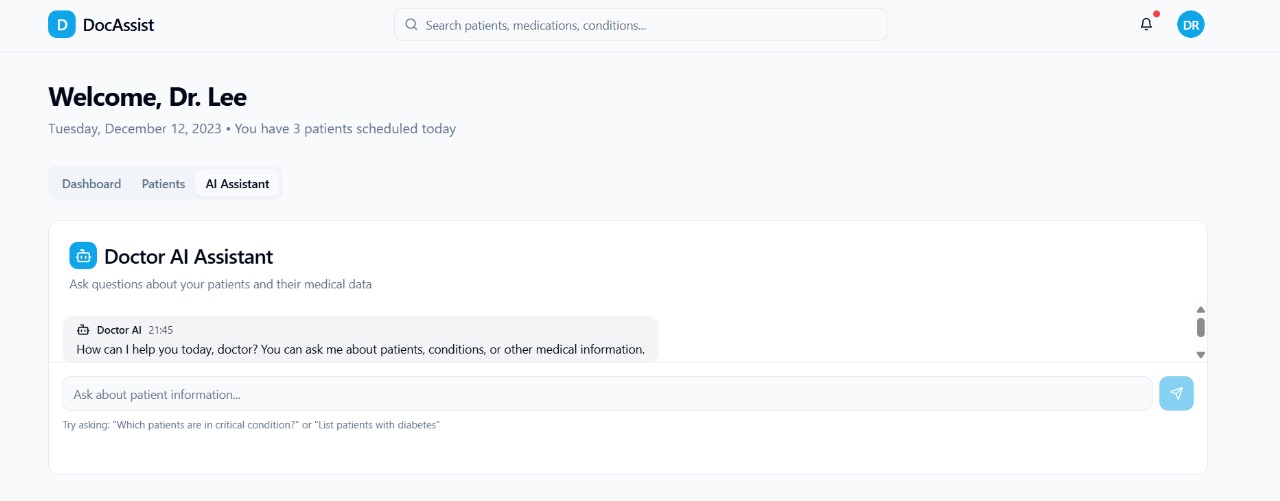
**Prompt:** For the current UI make some changes while keeping current basic functionalities intact: -Once the doctor logins, a dashboard should be displayed -SO there should be 3 sections, Dashboard, Patients and AI assistant. -The dashboard section automatically open once the user logs in -In the patients sections list of patients should be there, and once clicked on them, a detailed case history, their contact, summary, vitals, history, chat (personal for the particular patient which the doctor can enquire about) -And in the AI assistant section general questions about all patients can be asked. -- Make these upgrades

**Result:** I'll create a new dashboard layout with three main sections: Dashboard, Patients, and AI Assistant. I'll maintain all current functionality while reorganizing the application structure to accommodate this new navigation flow.  
I've created a complete dashboard system with three main sections as requested. The new layout includes a Dashboard page that shows at login, a Patients section that maintains all current patient functionality, and an AI Assistant section for general questions. The navigation is intuitive with a sidebar for desktop and compact navigation for mobile devices.



**Prompt:** Integrate gemini AI into the AI assistant section which will go through the patient's data and answer doctor's questions in real time.

**Result:** I'll implement Gemini AI integration with our existing AI Assistant to provide more intelligent responses based on patient data. Let me set this up for you.  
The basic setup is done, please provide the API key to proceed further.  
  
**Follow up:** API KEY : AIzaSyDxTQKfSnBaoEQup51k-6tqUO6js6S7rCZ  
  
**Result:** The integration of Google Gemini AI is now complete.The AI assistant will use Gemini to generate more accurate and contextual responses based on patient data.



**Prompt:** Dashboard AI Assistant:

* + "AI Assistant (Dashboard) – Go beyond filtering; understand and respond to database-related questions."

**Result Generated:**

* Created a general-purpose AI assistant component that can:
  + Answer statistical queries about the patient database
  + Handle questions like "How many female patients aged 30-40?"
  + Provide aggregate data about patient demographics
  + Show trends in patient visits or conditions
* Implemented with proper TypeScript typing for all queries/responses
* Added error handling for invalid queries
* Connected to the patient database with appropriate data access layer

**Prompt:** Patient AI Chat:

* + "AI Chat (Patient Details) – Answer only the specific details asked (e.g., medicines, vitals)."

**Result Generated:**

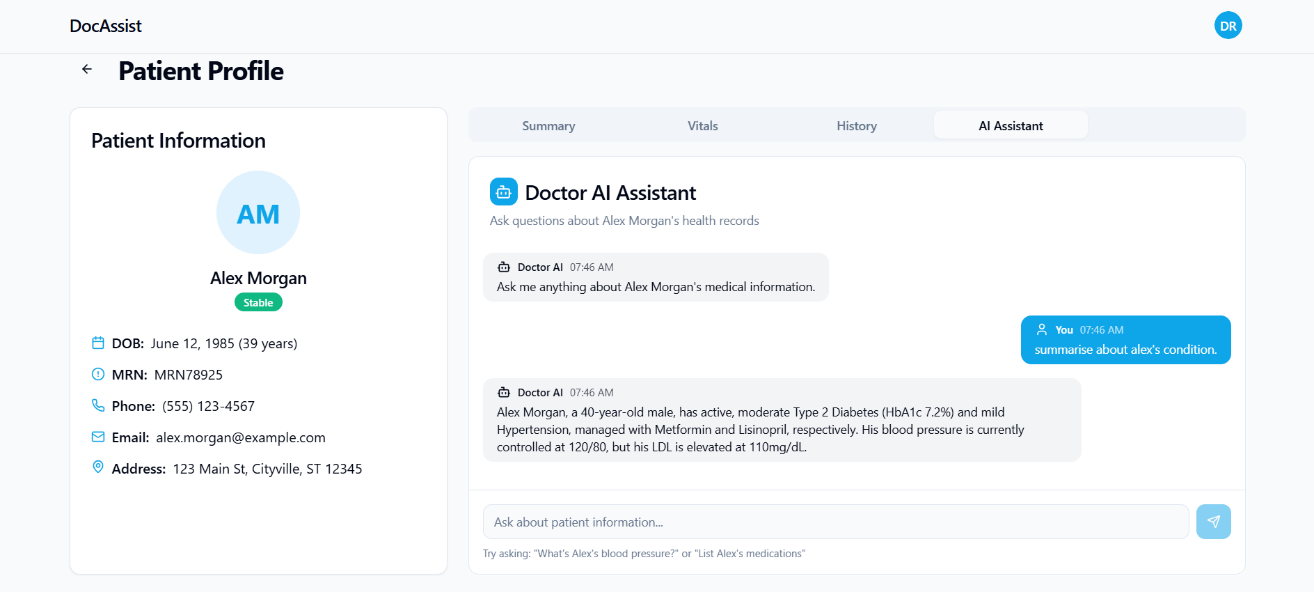
* Created a patient-specific AI chat component that:
  + Only responds to questions about the currently viewed patient
  + Answers medication-related queries ("What medicines is John taking?")
  + Handles vital sign questions ("Show me patient's last blood pressure readings")
  + Rejects queries about other patients
* Implemented with strict context boundaries
* Added typing for patient-specific data structures
* Includes error handling for out-of-context questions

**Prompt:** General AI Assistant Features:

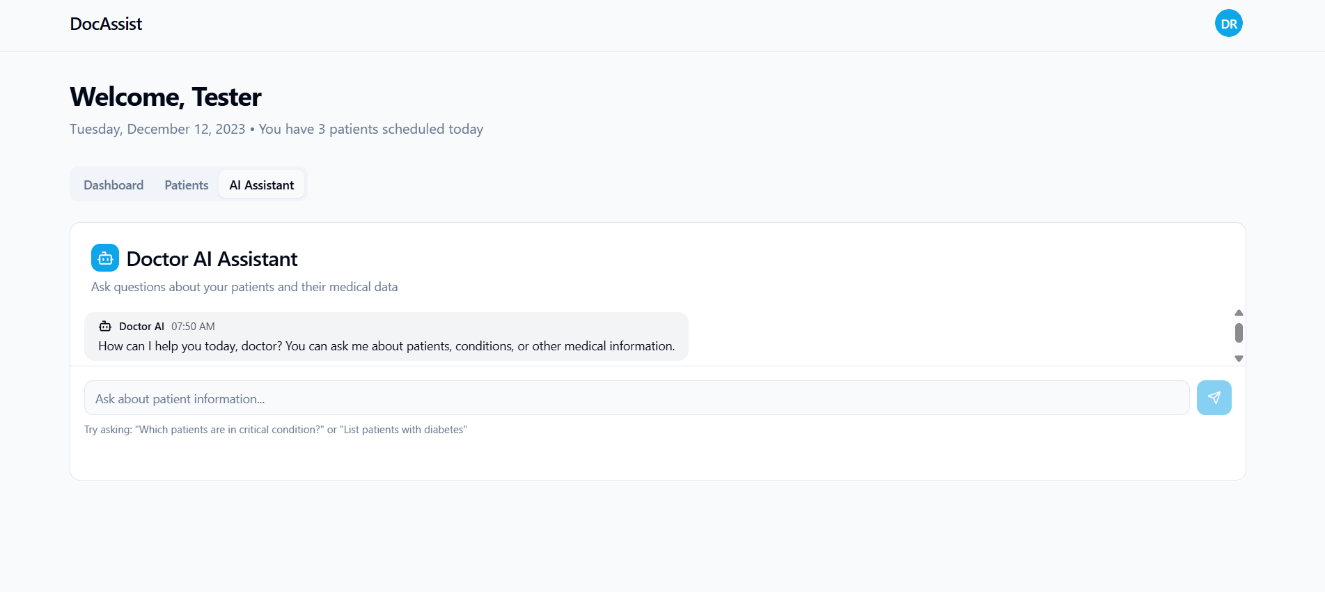
* + The dashboard AI should handle database-related questions about patient statistics
  + The patient-specific AI should only answer questions about the currently viewed patient

**Results Generated:**

* Type-safe interfaces for both AI components
* Shared error handling system with:
  + "I can only answer about the current patient" responses for Patient Chat
* Context-aware response systems that understand their scope
* Proper state management for both assistant types



Patient’s AI Assistant



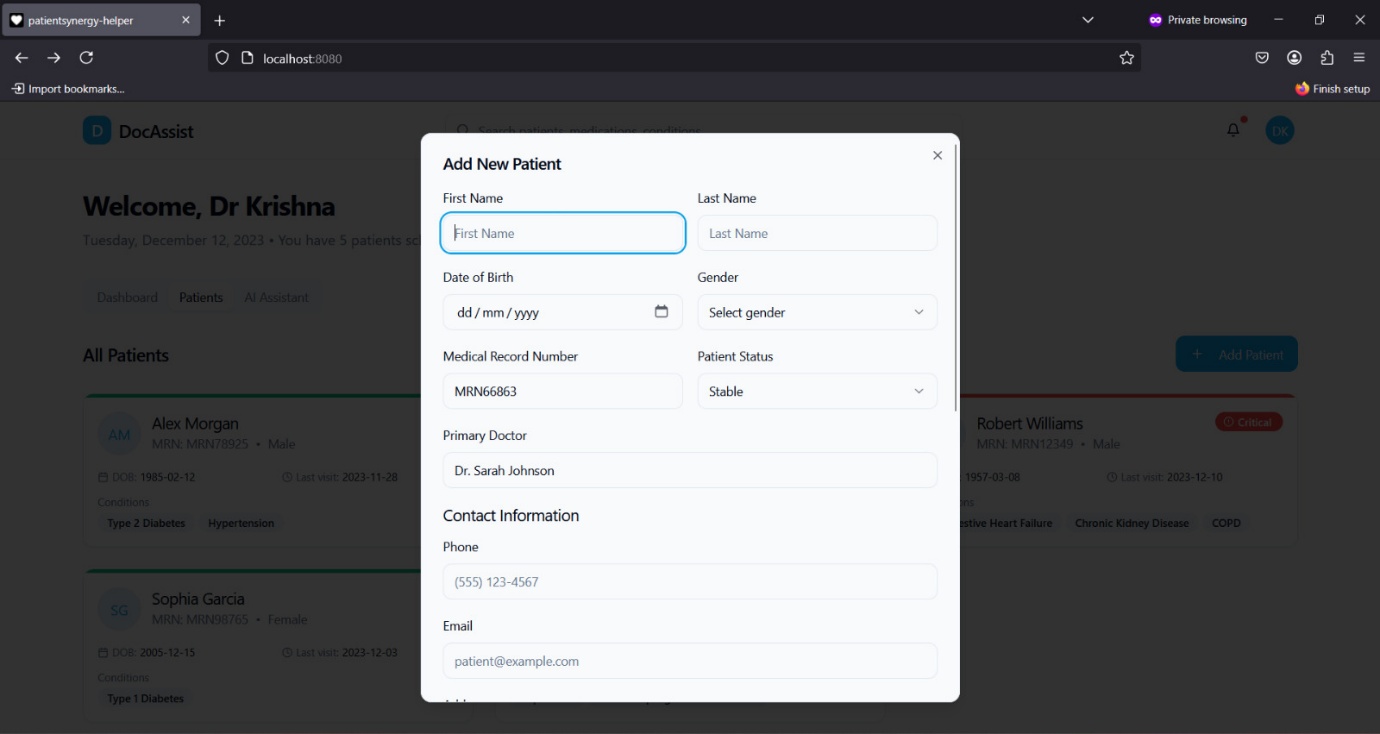
General AI Assistant

**Prompt:** Patients page add a Add Patient Button to add new patient details which will enable doctor to add new patient details which should take input through specific fields similar to existing patients.

**Result:** I'll create an "Add Patient" button in the patients tab that opens a form for adding new patient details. This will allow doctors to create new patient records with fields matching the existing patient data structure.

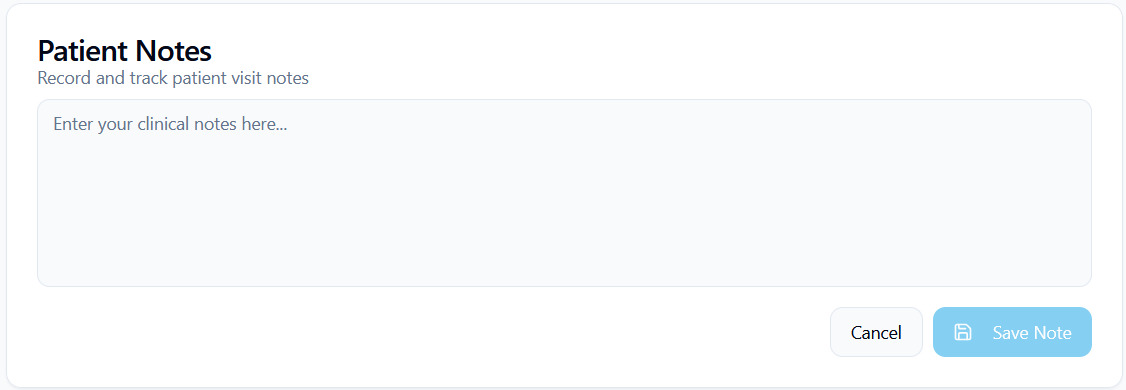
Added patient creation functionality

I've implemented an "Add Patient" functionality with a form that collects all necessary patient information. The button appears in the Patients tab and opens a dialog with fields for personal information, contact details, and medical data. When submitted, the new patient is added to the list and displayed alongside existing patients. I also made sure to update the patient count in the welcome message to reflect any newly added patients.



**Prompt:** Now in profile section of each patient remove chat feature and add replace it with Notes So that doctor can write in that note section whenever the patient visits and that data should be saved forever in his profile as notes  
**Result:** I'll modify the patient profile to replace the chat tab with a Notes section where doctors can add and save notes for each patient.

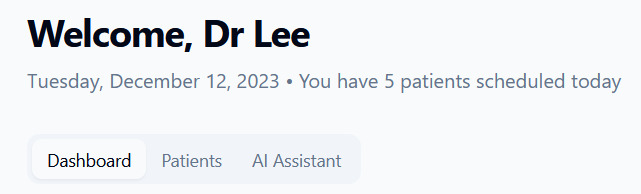
I've replaced the chat feature with a new Notes section in each patient's profile. Doctors can now add, view, and delete clinical notes for each patient. The notes are automatically saved in the browser's localStorage with timestamps, making them persistent between sessions. This allows doctors to keep a record of visits and important information for each patient.



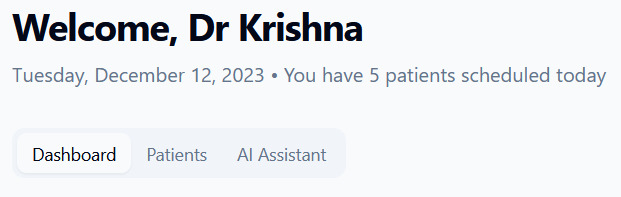
**Prompt:** In the dashboard when user signup they will you provide you with Fullname that should be displayed that is Welcome "Thier Full name" it is showing dr lee for all that should be fixed  
(prompt for minor data display issue in dashboard)

**Result:** I'll update the dashboard to display the user's full name from their signup info instead of the hard-coded "Dr. Lee". This will require modifications to use the authenticated user data in the Index page.

Now the dashboard will display the user's full name (entered during signup) in the welcome message instead of the hard-coded "Dr. Lee". The Navbar will also show the user's name and appropriate initials in the avatar. This change uses the authentication data that's stored in the user's metadata when they sign up.



Before (Default doctor name)



After ( Name of the user logged in )