# **The Remedy Lab – Personalized Treatment Plan**

## **Detailed Project Structure and Design**

1. **Architectural Overview**

* Frontend: Streamlit (Python)
* Backend Logic: Python (for data processing, AI model interaction)
* Database: SQLite (a single .db file for storing all data)
* AI Model: Pre-trained or custom-trained (e.g., using scikit-learn, TensorFlow, PyTorch, or an API like a medical AI service if available)
* Document Processing: Libraries like PyPDF2, python-docx, pandas for CSV/JSON, pdfplumber or Tesseract (for OCR if scanned PDFs are involved).

1. Database Schema Design – SQL Lite

Here's the schema with SQLite

**Table: users**

* user\_id (Primary Key, TEXT) - *Store UUIDs as text*
* username (Unique, TEXT)
* password\_hash (TEXT)
* user\_type (TEXT) - *e.g., 'patient', 'doctor'*
* first\_name (TEXT)
* last\_name (TEXT)
* email (Unique, TEXT)
* created\_at (TEXT) - *Store as ISO 8601 string*
* updated\_at (TEXT) - *Store as ISO 8601 string*

***Table: patients***

* patient\_id (Primary Key, TEXT) - *Foreign Key to users.user\_id*
* date\_of\_birth (TEXT) - *Store as ISO 8601 date string (YYYY-MM-DD)*
* gender (TEXT) - *e.g., 'Male', 'Female', 'Other'*
* contact\_number (TEXT)
* address (TEXT)

***Table: doctors***

* doctor\_id (Primary Key, TEXT) - *Foreign Key to users.user\_id*
* medical\_license\_number (Unique, TEXT)
* specialization (TEXT)
* contact\_number (TEXT)
* hospital\_affiliation (TEXT)

***Table: health\_reports***

* report\_id (Primary Key, TEXT) - *Store UUIDs as text*
* patient\_id (Foreign Key to patients.patient\_id, TEXT)
* uploaded\_by (Foreign Key to users.user\_id, TEXT)
* report\_type (TEXT)
* upload\_date (TEXT) - *Store as ISO 8601 string*
* file\_name (TEXT)
* file\_path (TEXT - path to stored file or cloud storage URL)
* extracted\_data\_json (TEXT) - *Store JSON as string (json.dumps)*
* processing\_status (TEXT) - *e.g., 'pending\_extraction', 'extracted', 'failed\_extraction'*

***Table: recommendations***

* recommendation\_id (Primary Key, TEXT) - *Store UUIDs as text*
* report\_id (Foreign Key to health\_reports.report\_id, TEXT)
* patient\_id (Foreign Key to patients.patient\_id, TEXT)
* ai\_generated\_treatment (TEXT)
* ai\_generated\_lifestyle (TEXT)
* ai\_generated\_priority (TEXT)
* doctor\_id (Foreign Key to doctors.doctor\_id, TEXT, Nullable)
* doctor\_notes (TEXT, Nullable)
* status (TEXT) - *e.g., 'AI\_generated', 'pending\_doctor\_review', 'approved\_by\_doctor', 'modified\_by\_doctor', 'rejected\_by\_doctor'*
* reviewed\_date (TEXT, Nullable) - *Store as ISO 8601 string*
* approved\_treatment (TEXT, Nullable)
* approved\_lifestyle (TEXT, Nullable)

***Table: patient\_doctor\_mapping***

* mapping\_id (Primary Key, TEXT) - *Store UUIDs as text*
* patient\_id (Foreign Key to patients.patient\_id, TEXT)
* doctor\_id (Foreign Key to doctors.doctor\_id, TEXT)
* assigned\_date (TEXT) - *Store as ISO 8601 string*
* is\_active (INTEGER) - *Store boolean as 0 or 1*

1. **Module Breakdown and UI Design (Streamlit)**

**Global Components (Across all pages)**

* Header: Project Title, Logo.
* Sidebar Navigation: Used for authenticated users.

**Page 1: Home Page (Initial Landing)**

* URL: /
* Design:
  + Prominent Project Title: "Personalized Treatment Plans"
  + Brief overview/tagline.
  + Two main action buttons: "Sign Up" and "Log In".
  + Optional: Brief explanation of the system's benefits.

**Page 2: Sign Up Page**

* **URL:** /signup
* **Design:**
  + **Header:** "Create Your Account"
  + **User Type Selection:** Radio buttons: "I am a Patient", "I am a Doctor".
  + **Patient Fields:** First Name, Last Name, Email, Password, Confirm Password, Date of Birth, Gender, Contact Number, Address.
  + **Doctor Fields:** First Name, Last Name, Email, Password, Confirm Password, Medical License Number, Specialization, Contact Number, Hospital Affiliation.
  + **Submit Button.**
  + **"Already have an account?" link** to Login Page.
* **Unique ID Generation:** Upon successful signup, generate a UUID for user\_id, patient\_id, or doctor\_id.

**Page 3: Log In Page**

* **URL:** /login
* **Design:**
  + **Header:** "Log In"
  + **User Type Selection:** Radio buttons: "I am a Patient", "I am a Doctor" (as you suggested).
  + **Login Fields:** Username/Email, Password.
  + **Submit Button.**
  + **"Don't have an account?" link** to Sign Up Page.

**Page 4: Patient Dashboard**

* **URL:** /patient\_dashboard
* **Design:**
  + **Sidebar:** Navigation: "Upload Report", "My Reports", "My Recommendations".
  + **Main Area (Default: "Upload Report"):**
    - **Header:** "Upload Health Report"
    - **File Uploader:** st.file\_uploader (accepts PDF, Docx, CSV, JSON).
    - **Report Type Selection:** Dropdown (e.g., "Blood Test", "Urine Test").
    - **Description (Optional):** st.text\_area.
    - **Upload Button.**
  + **"My Reports" View:**
    - **Table:** List of uploaded reports (Report Name, Upload Date, Status - e.g., "Pending AI Analysis", "Ready for Doctor Review", "Recommendations Available").
    - **Action Buttons per row:**
      * **"View Report":** Opens a modal or new tab to display the original uploaded report (PDF.js for PDFs, etc.).
      * **"View AI Analysis" (or "Raw Data"):** Shows the extracted structured data.
      * **"View Recommendation":** This button is initially disabled. It becomes **enabled and visible** only when recommendations.status for that report is approved\_by\_doctor or modified\_by\_doctor.
  + **"My Recommendations" View (Alternative to per-report button):**
    - A consolidated view of all *approved* recommendations for the patient.
    - Table: Report Name, Doctor Name, Approved Treatment, Approved Lifestyle, Notes.

**Page 5: Doctor Dashboard**

* **URL:** /doctor\_dashboard
* **Design:**
  + **Sidebar:** Navigation: "My Patients", "Pending Reviews".
  + **Main Area (Default: "My Patients"):**
    - **Header:** "My Patients"
    - **Table:** List of patients assigned to this doctor (from patient\_doctor\_mapping).
    - **Action Buttons per row:**
      * **"View Patient Profile":** Displays patient's basic info, medical history (if available/uploaded).
      * **"View Patient Reports":** Navigates to a view similar to Patient's "My Reports" but for that specific patient.
  + **"Pending Reviews" View:**
    - **Header:** "Reports Pending Your Review"
    - **Table:** List of health\_reports where recommendations.status is AI\_generated or pending\_doctor\_review AND doctor\_id is linked to the current doctor OR doctor\_id is NULL (if doctors can pick up unassigned cases, which adds complexity).
    - **Action Button per row:** "Review Recommendation"

**Page 6: Doctor's Patient Reports View**

* **URL:** /doctor\_patient\_reports\_view (Accessible from Doctor Dashboard)
* **Design:**
  + **Header:** "Reports for [Patient Name]"
  + **Table:** Similar to Patient's "My Reports" table, but with additional columns like "AI Analysis Status", "Doctor Review Status".
  + **Action Buttons per row:** "View Report", "View AI Analysis", "Review/Edit Recommendation" (this button is always visible if a recommendation exists, regardless of approval status, for the doctor to manage).

**Page 7: Doctor Review and Validation Interface**

* **URL:** /doctor\_review\_interface (Accessed via "Review Recommendation" button from Doctor Dashboard)
* **Design:**
  + **Header:** "Review and Validate Recommendation for [Patient Name] - [Report Name]"
  + **Section 1: Original Report View:**
    - Display the uploaded PDF/Docx/CSV/JSON content (or a rendered preview).
    - Maybe embed a PDF viewer or simply show extracted key metrics clearly.
  + **Section 2: Extracted Health Metrics:**
    - Display the standardized data from extracted\_data\_json in a readable format (e.g., st.dataframe).
  + **Section 3: AI-Generated Recommendations (Read-only initially):**
    - **"AI Treatment Suggestions":** Text area displaying ai\_generated\_treatment.
    - **"AI Lifestyle Recommendations":** Text area displaying ai\_generated\_lifestyle.
    - **"AI Priority":** Display ai\_generated\_priority.
  + **Section 4: Doctor's Review and Modification Area:**
    - **"Your Treatment Plan":** st.text\_area (pre-filled with AI suggestion, editable).
    - **"Your Lifestyle Plan":** st.text\_area (pre-filled with AI suggestion, editable).
    - **"Doctor's Notes":** st.text\_area for additional comments.
    - **Action Buttons:**
      * "Approve Recommendation"
      * "Modify and Approve"
      * "Reject Recommendation" (with a mandatory reason field)
      * "Save Draft" (if doctor wants to come back later)
* **Logic:**
  + When "Approve Recommendation" is clicked: Copy AI-generated text to approved\_treatment and approved\_lifestyle in recommendations table, set status to approved\_by\_doctor, doctor\_id to current doctor's ID, reviewed\_date to now.
  + When "Modify and Approve" is clicked: Save content of editable text areas to approved\_treatment and approved\_lifestyle, set status to modified\_by\_doctor.
  + When "Reject Recommendation" is clicked: Set status to rejected\_by\_doctor, save notes.
  + After any action, redirect back to Doctor Dashboard "Pending Reviews" or "My Patients" view.

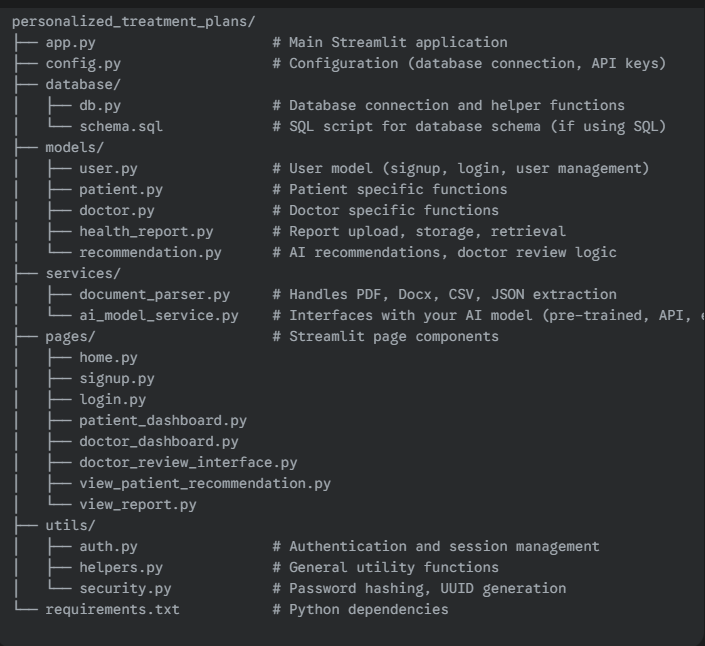
**Page 8: Patient Recommendation View (After Doctor Approval)**

* **URL:** /view\_patient\_recommendation (Accessed from Patient Dashboard "My Reports" -> "View Recommendation" button)
* **Design:**
  + **Header:** "Your Personalized Plan from Dr. [Doctor Name]"
  + **Sections:**
    - **"Your Health Report Summary":** Brief overview of key metrics.
    - **"Approved Treatment Plan":** Display approved\_treatment from recommendations table.
    - **"Approved Lifestyle Recommendations":** Display approved\_lifestyle from recommendations table.
    - **"Doctor's Notes":** Display doctor\_notes.
    - **"Review Date":** Show reviewed\_date.
  + No editing options for the patient.

1. **Summary - Relation Between Patient and Doctor**
   * + 1. **Patient Signup: A patient signs up and gets a patient\_id.**
       2. **Doctor Signup: A doctor signs up and gets a doctor\_id.**
       3. **Assigning Doctors to Patients (Crucial):**

* **(Doctor Initiated - Recommended for Simplicity):**
  + **On the doctor's dashboard, there could be a "Add Patient" feature. The doctor searches for a patient (e.g., by patient ID or email provided by the patient).**
  + **Once found, the doctor can "assign" themselves to that patient, creating an entry in the patient\_doctor\_mapping table.**
  + ***Benefit:* Doctors control their caseload. Patients don't need to explicitly "choose" a doctor in the system, relying on the doctor to initiate the relationship.**
    - 1. **Report Upload: When a patient uploads a report, the health\_reports entry is created with their patient\_id.**
      2. **AI Analysis: The AI processes the report, generates recommendations, and creates an entry in recommendations table, linking it to the report\_id and patient\_id. The status is AI\_generated. The doctor\_id is initially NULL.**
      3. **Doctor Review Queue:**
* **Doctors see recommendations where status is AI\_generated AND patient\_id is associated with *their* doctor\_id in the patient\_doctor\_mapping table.**
* **This ensures doctors only see recommendations for *their* assigned patients.**
* **If you implement a system where doctors "pick up" unassigned cases, then the query would also include doctor\_id IS NULL and upon picking up, the doctor\_id in recommendations gets updated.**
  + - 1. **Recommendation Visibility to Patients:**
* **The "Recommendation" button on the patient's "My Reports" page is enabled only when the recommendations.status for that specific report is approved\_by\_doctor or modified\_by\_doctor.**
* **The patient can then view the final, doctor-validated recommendation.**

1. **Structured Framework (Python/Streamlit)**

****