eFiche

Take-Home Assignment: Patient Medical History Management System

Objective

Create a web-based application using React for the frontend, Node.js/Express for the backend, and PostgreSQL for data storage. The app will include two types of users: practitioners and patients, each with specific functionality:

1. Practitioners can log in, add patient records (e.g., allergies, lab orders, lab results, prescriptions).
2. Patients can log in to view their medical history in a meaningful and user-friendly way.

Requirements

1. User Authentication

- Practitioner Login:
- Practitioners can log in using a username and password.
- Use JWT (JSON Web Tokens) for session management.
- Seed the database with at least one practitioner account (e.g., username: doc1, password: pass123).
- Patient Login: Patients can log in using a username and password to view their medical history.
- Seed the database with at least one patient account (e.g., username: patient1, password: pass123).

2. Practitioner Features

Once logged in, practitioners can:

- Select a patient from a dropdown or search bar (populated from seeded patients).
- Add the following types of data to the selected patient's record:

    1. Allergies: A list of known allergies.

2. Lab Orders: Specify tests ordered (e.g., “CBC”, “X-ray”).
3. Lab Results: Record results for previous lab orders (e.g., “CBC: Normal”).
4. Prescriptions: Add medication names, dosages, and instructions.

- Backend: Create APIs to:
  - Retrieve a list of patients.
  - Save the above services for a selected patient.
  - Validate required fields for each type of data.
- Frontend:
  - Create a form where practitioners can input the above data.
  - Display confirmation or errors after data submission.

3. Patient Features

Once logged in, patients can:

- View their Medical History, organized in a meaningful way:
- Allergies: A simple list.
- Lab Orders & Results: Group lab results under the respective lab orders.
- Prescriptions: Show medication names, dosages, and instructions.
- Backend:
  - Create an API to fetch the logged-in patient’s medical history.
- Frontend:
  - Create a dashboard with tabs or sections for Allergies, Lab Orders & Results, and Prescriptions.
  - Present the data in a clean, user-friendly format (e.g., a timeline or grouped cards).

4. Bonus (Optional)

- Add a feature to upload and display lab result files (e.g., PDFs or images).
- Show a summary view of the patient’s health data on the patient dashboard (e.g., “3 active prescriptions, 2 pending lab results”).
- Add search functionality for practitioners to filter patients by name or ID.

eFiche

Tech Stack

- Frontend: React.js (with any state management library like Redux or Context API).
- Backend: Node.js with Express.js.
- Database: PostgreSQL.
- Authentication: JWT.

Deliverables

- A GitHub repository containing:
- Complete source code.
- A README file explaining:
    - How to set up and run the project locally.
    - Any assumptions or limitations in your implementation.
    - Clearly organized project structure for both frontend and backend.

Evaluation Criteria

1. Relevance: Does the app align with the healthtech context?
2. Functionality: Are the practitioner and patient workflows working correctly?
3. User Experience: Is the patient medical history presented meaningfully and cleanly?
4. Code Quality: Is the code modular, maintainable, and documented?
5. Scalability: Is the backend designed to handle additional features easily?

eFiche

Example Workflow

Practitioner:

1. Logs in to the app.
2. Selects a patient from a dropdown.
3. Adds details like allergies, lab orders, lab results, and prescriptions for the patient.

Patient:

1. Logs in to the app.
2. Views a clean, intuitive dashboard showing:
    - Allergies in a simple list.
    - Lab orders and results grouped together.
    - Prescriptions with names, dosages, and instructions.