E-Rx Hub by DevXWorld

FULL DOCUMENTATION SUITE

DevXWorld e-Rx Hub - Technical & Setup Guide

Company: DevXWorld Inc. (Developer Shapes the Future)

1. System Overview

Purpose: The DevXWorld e-Rx Hub is a secure, legally compliant e-prescription portal designed for Indian Registered Medical Practitioners (RMPs) and Pharmacies. It facilitates the digital creation, signing, and dispensing of prescriptions while adhering to the DPDP Act 2023 and Telemedicine Practice Guidelines 2020.

Core Architecture:

- Architecture Pattern: Client-Side Single Page Application (SPA).
- Rendering: Client-side rendering (React 19).
- State Management: React Context / Local State + Supabase Realtime (simulated via polling/hooks).
- Data Persistence: Hybrid Strategy. Primary: Supabase (PostgreSQL). Fallback: LocalStorage (for offline/demo capabilities).

Tech Stack:

- Frontend Framework: React 19 (via ESM imports).
- Language: TypeScript (Strict typing).
- Styling: Tailwind CSS (Utility-first).
- Database & Auth: Supabase (PostgreSQL, GoTrue, Storage).
- AI Engine: Google Gemini API (v1.30.0 via `@google/genai`) - Used for drug interaction checks.
- Visualization: Recharts (Admin analytics).
- Icons: Lucide React.
- Build Tooling: Vite (implied by usage patterns).

2. Installation & Setup

Prerequisites

- Node.js: v18.0.0 or higher.
- npm/yarn: Latest version.
- Supabase Project: A standard Supabase project with `users`, `prescriptions`, `patients`, `audit_logs` tables.
- Google AI Studio Key: An API key with access to `gemini-2.5-flash`.

Local Installation

```
1. git clone https://github.com/devxworld/erx-hub.git
2. cd erx-hub
```

```
            3. npm install
```

Environment Configuration

Create a `.env` file in the root directory:

```
            VITE_SUPABASE_URL=your_supabase_project_url
            VITE_SUPABASE_ANON_KEY=your_supabase_anon_key
            API_KEY=your_google_gemini_api_key
```

Run Development Server

```
            npm run dev
```

Access the app at `http://localhost:5173`.

Deployment (Vercel/Netlify)

- Connect your Git repository to Vercel.
- Build Command: `npm run build`
- Output Directory: `dist`
- Environment Variables: Add the variables listed above in the Vercel Project Settings.

3. Codebase Documentation

Folder Structure

- `components/`: UI building blocks.
    - `auth/`: Login and Registration logic (`Login.tsx`).
    - `doctor/`: Doctor specific workflows (`DoctorDashboard`, `CreatePrescription`).
    - `pharmacy/`: Pharmacy workflows (`PharmacyDashboard`).
    - `admin/`: System administration (`AdminDashboard`).
    - `ui/`: Shared components (`Layout`, `PrintLayout`).
- `services/`: External API integrations.
    - `db.ts`: Abstracted database layer handling Supabase and LocalStorage fallback.
    - `geminiService.ts`: AI interaction logic for drug safety checks.
- `types.ts`: TypeScript interfaces for `User`, `Prescription`, `Patient`, `AuditLog`.
- `constants.ts`: Static data (`LOW_RISK_GENERIC_LIST`, `INDIAN_STATES`) and Regex patterns.

Key API Integration Points

- **`services/db.ts`**:
    - `loadData()`: Fetches initial state. Implements a robust merge strategy between SQL tables and Blob storage for audit logs.
    - `logSecurityAction(actorId, action, details)`: Critical function for forensic logging. Tries SQL insert first, falls back to JSON blob if permissions fail.
- **`services/geminiService.ts`**:
    - `analyzePrescriptionSafety()`: Sends diagnosis and medicine list to Gemini 2.5 Flash to check for interactions and contraindications. Returns JSON.

Data Models

- **User:** Handles Doctors, Pharmacies, and Admins. Differentiated by `role` enum. Contains `verificationStatus`.
- **Prescription:** core transactional entity. Links `doctorId`, `patientId`, and `pharmacyId`. Statuses: `ISSUED`, `DISPENSED`, `REJECTED`, `REJECTED_STOCK`.
- **AuditLog:** Forensic trail. `actorId`, `action` (e.g., `USER_LOGIN_SUCCESS`), `timestamp`.

4. Development Guidelines

Coding Standards

- Type Safety: No `any`. Define interfaces in `types.ts`.
- Security First: All inputs validted via Regex (defined in `constants.ts`) before submission.
- Performance: Use `useMemo` for heavy filtering in Dashboards.
- Chunk Size: If build warnings occur, use dynamic imports (`React.lazy`) for the Dashboard components to split code chunks.

Debugging

- Local Storage Mode: If Supabase fails, the app falls back to Local Storage. Check `Application > Local Storage` in DevTools to inspect `devx_users` or `devx_prescriptions`.
- Console: Security logs are printed to console in Dev mode.

Known Limitations

- Offline Mode: While LocalStorage works, it does not sync across devices. Cloud connection is required for multi-device workflows.

e-Rx Hub: Compliance & Security Protocol

Version: 1.2 | Status: Audit Ready

1. Compliance Requirements

Data Privacy & Protection (DPDP Act 2023)

- Consent Architecture: Explicit Consent (Checkbox implemented in `Login.tsx`).
- Purpose Limitation: Data is collected solely for the purpose of medical prescription and dispensing.
- Data Minimization: Only essential clinical data (Diagnosis, Vitals) is required.

Telemedicine Compliance

- RMP Verification: Doctors must upload Medical Degree and State Council Registration. These are manually verified by Admins before account activation.
- Patient Identification: The `CreatePrescription` workflow forces the doctor to acknowledge they have verified the patient's identity via video/audio before prescribing.
- Record Maintenance: All prescriptions are stored indefinitely (or per statutory limits) in the database with timestamps.

Data Retention

- Prescriptions: Retained for legal duration (typically 3+ years).
- Audit Logs: Security logs are immutable and retained for forensic analysis.

2. Security Architecture

Authentication & Authorization

- Method: Supabase Auth (JWT).
- Two-Factor Authentication (2FA): Simulated 2FA (OTP) step implemented in `Login.tsx` for all roles.
- Role-Based Access Control (RBAC):
    - Doctors: Can only see their own patients and prescriptions.
    - Pharmacies: Can only see prescriptions assigned to them or processed by them.
    - Admins: Have global view but cannot alter clinical data.

Input Hardening & Integrity

- Regex Validation: Strict enforcement on registration.
    - Medical Reg No: `^[a-zA-Z0-9]{5,15}$` (Prevents SQLi/XSS via ID fields).
    - Phone: `^\d{10}$` (Strict length).
    - Pincode: `^\d{6}$`.
- Sanitization: React automatically escapes output, preventing XSS in rendered views.

Session Management

- Idle Timeout: Global activity listener in `App.tsx`.
    - Limit: 30 Minutes.

    - Warning: 30 seconds prior to termination.
    - Action: Immediate destruction of session tokens and redirect to login.

3. Risk Mitigation

Threat Model: Drug Abuse

- Mitigation: Restricted Drugs: The frontend filters out high-risk narcotics (e.g., Morphine, Fentanyl) from the autocomplete suggestions. Inventory Tracking: Pharmacies track stock of narcotic items via the `isNarcotic` flag.

4. Incident Response

Service Outage

- Failover: Application automatically falls back to LocalStorage if Supabase connection fails (`dbService.ts`).
- Recovery: Once connection restores, users must re-authenticate. Note: Offline data currently does not auto-sync to cloud to prevent conflict.

User Workflow Guides

Doctor Portal User Manual

Welcome, Doctor. The e-Rx Hub is your secure digital clinic. This guide will help you manage patients and issue legally compliant e-prescriptions in seconds.

1. Getting Started

- Sign Up: Select "Doctor" on the login screen and click "New Registration".
- Details: Enter your Full Name, Qualifications, and Clinic Address.
- Documents: You must upload your Medical Degree and State Council Registration Certificate.
- Verification: Your account will be in "Pending" status. Our Admin team will verify your credentials within 24 hours. You cannot issue prescriptions until verified.
- Login: Use your registered email and password.
- OTP: Enter the 6-digit OTP sent to your mobile (for demo: use any 6 digits).

2. Dashboard Overview

Once logged in, you will see your Doctor Dashboard:

- Create Prescription: The main workspace for writing Rxs.
- My Patients: A directory of all patients you have treated.
- Rx Logs: A history of all prescriptions issued, with their current status (Dispensed/Pending).

3. Creating a Prescription (Step-by-Step)

Step 1: Select Patient

- Existing Patient: Type the name or phone number in the search bar. Select the patient from the dropdown.
- New Patient: Click "Create New". Enter Name, Age, Gender, and Phone.
- History Autofill: Use the "Autofill" box at the top right to search previous prescriptions. Clicking one will copy the diagnosis and medicines from a past visit.

Step 2: Diagnosis & AI Check

- Enter the diagnosis (e.g., "Acute Bronchitis").
- Add Medicines using the "Add Drug" button.
- Autocomplete: Start typing a generic drug name (e.g., "Para") to see safe suggestions.
- AI Safety Check: Click "Check Interactions". The AI will analyze your prescription for drug-drug interactions and provide safety warnings.

Step 3: Select Pharmacy

- Choose a verified pharmacy from the dropdown list. The prescription will be sent directly to their dashboard.

Step 4: Sign & Send

- Check the "Telemedicine Compliance" box confirming you verified the patient.
- Click "E-Sign & Send".
- The prescription is now digitally signed and instantly available at the pharmacy.

4. Patient Management

- Navigate to the "My Patients" tab.
- View Profile: Click on a patient card to see their vitals, allergies, and Rx history.
- Edit: You can update phone numbers or add new chronic conditions.

5. Troubleshooting & Support

- Common Issues: "Account Pending" (Documents under review).
- Session Expiring: For security, the system logs you out after 30 minutes of inactivity. Move your mouse to reset the timer.
- "Out of Stock": If a prescription status shows "OUT OF STOCK", the pharmacy rejected it. You may need to prescribe an alternative.

Pharmacy Portal User Manual

Welcome, Pharmacist. The e-Rx Hub transforms your pharmacy into a digitally connected healthcare node. This guide covers compliant dispensing and record maintenance.

1. Getting Started

- Registration & Verification: You **must** upload a valid Pharmacy License. Access is granted only after Admin verification.
- Login: Use your registered email and password. Enter the 6-digit OTP sent to your registered mobile.

2. Dispensing Prescriptions (Workflow)

- Check the Queue: Navigate to ∗∗E-Rx Management** → ∗∗Pending Queue**.
- Patient Matching: Link the incoming prescription to the patient's existing profile or create a new one, verifying Phone Number and Address.
- Dispense: Once stock is verified, click ∗∗Confirm & Dispense**. This updates the status to "DISPENSED" and logs the transaction against your Pharmacist UID.
- Reject: Use ∗∗No Stock** or ∗∗Reject** buttons for issues, which notifies the doctor instantly.

3. Inventory Management

- Adding Stock: Enter Medicine Name, Batch No, Expiry, MRP, and Quantity. Flag controlled substances (NDPS) using the "Narcotic / Controlled?" checkbox.
- Reports: View financial reports based on dispensed prescriptions and monitor the ∗∗Activity Log** for audit purposes.

4. Support

Technical Lead: Devam Trivedi | Email: Devam9131@gmail.com