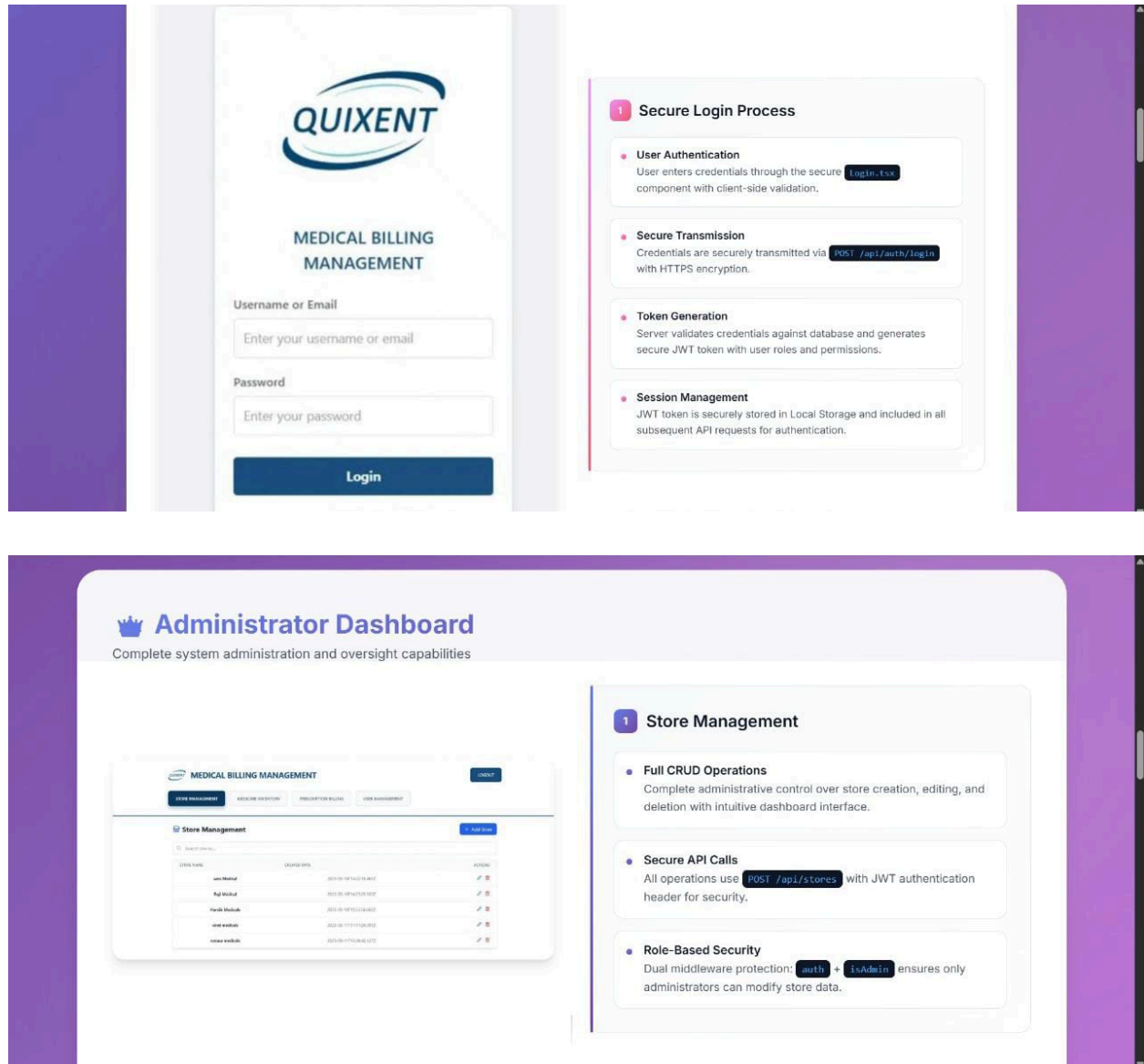# Medical Prescription Website – Complete Architecture Flow

2   Medicine Inventory Control

Comprehensive Inventory
Full access to medicine database with advanced search, filtering, and bulk operations capabilities.

Real-time Updates
Medicine modifications via `PUT /api/medicines/:id` with immediate database synchronization.

Administrative Security
Protected by `isAdmin` middleware ensuring inventory integrity and preventing unauthorized changes.

3   Financial Overview

3   Financial Overview

Global Billing Access
Comprehensive view of all billing records across the entire system with advanced analytics and reporting.

Financial Controls
Administrative privileges for billing record deletion via `DELETE /api/billings/:id` with audit trails.

Authentication Required
Secured by `auth` middleware with comprehensive logging for financial compliance and audit requirements.

4   User Administration

4   User Administration

**User Lifecycle Management**
Complete user management system with dedicated admin tab for creating, editing, and managing user accounts and permissions.

Secure User Creation
New user accounts created through `POST /api/users` with password hashing and role assignment.

Maximum Security
Highest security tier with both `auth` and `isAdmin` middleware protecting sensitive user operations.

👤 Standard User Experience

Streamlined interface for daily operations and store management
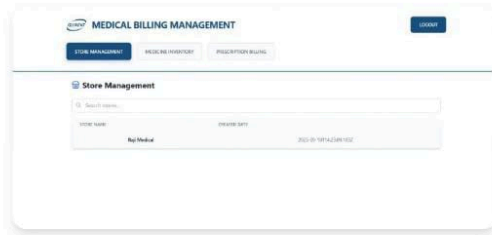
1 Store-Specific Access

Focused Interface
Clean, read-only view displaying only the user's assigned store information with hidden administrative controls.

Intelligent Data Filtering
Backend automatically filters `GET /api/stores` response based on user's assigned storeId from JWT token.

Security by Design
Data scoping ensures users can only access their designated store information, maintaining data isolation and privacy.

2 Medicine Inventory View

Read-Only Inventory
Comprehensive view of medicine inventory specific to user's store with search and filtering capabilities.

Store-Scoped Data
Automatic filtering via `GET /api/medicines` ensures users see only medicines relevant to their assigned store.

Simplified Interface
Clean UI with hidden add/edit/delete controls, focusing user attention on essential inventory information.
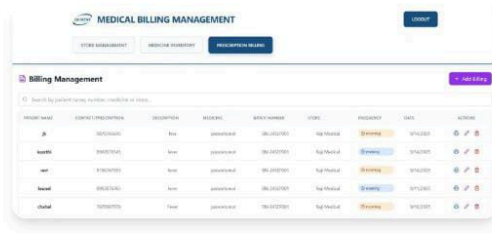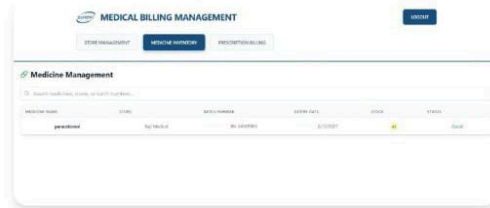
3 Billing Operations

Full Billing Control
Complete CRUD operations for billing records within their assigned store, enabling efficient daily operations.

Secure Transactions
New billing records created via `POST /api/billings` with automatic store association and validation.

Authentication & Scoping
Protected by `auth` middleware with backend logic ensuring all operations are scoped to user's store.

Flow:

Authentication Flow (Admin & User)

1. User Login
Action:
User submits email and password.

Component:
`Login.tsx`

2. API Request
Action:
Frontend sends credentials to the backend.

Endpoint:
`POST /api/auth/login`

3. Verification
Action:
Backend verifies credentials against the database.

DB Collection:
`Users`

4. Token Response
Action:
Backend returns a JWT. Frontend stores it and fetches user data to render the correct dashboard.

Admin Flow: Full CRUD Access

1. Admin Action
Action:
Clicks "Add Store", "Edit Medicine", etc.

UI:
Sees full controls.

2. API Request with JWT
Action:
Frontend sends request with JWT in header.

Example:
`POST /api/stores`

3. Middleware Check
Action:
`auth` middleware verifies JWT. `isAdmin` middleware checks if `user.type === 'admin'`.

4. Database Operation
Action:
If authorized, backend performs the Create, Update, or Delete operation in the database.

User Flow: Limited Access

1. User Action
Action:
Views stores/medicines or adds a new billing record.

UI:
CRUD buttons for stores/medicines are hidden.

2. API Request with JWT
Read Action:
`GET /api/stores`

Write Action:
`POST /api/billings`

3. Middleware Check
Action:
`auth` middleware verifies JWT. For billing, access is granted. For store/medicine writes, `isAdmin` would block it.

4. Scoped DB Operation
Read Action:
Backend logic filters data to show only records related to the user's `storeId`.

Write Action:
Billing record is created.