BYOWA Healthcare Data Portal (HDP)

A full-stack healthcare data management platform designed to handle patient records, clinical data, and FHIR-compliant medical information with role-based access control for patients, doctors, and administrators.

Project Overview

The Healthcare Data Portal provides a secure platform for managing healthcare data following the OMOP Common Data Model standards. It features:

- **User Authentication & Authorization** - JWT-based security with role-based access (Patient, Doctor, Admin)
- **FHIR Data Import** - Import and validate FHIR-compliant healthcare bundles
- **Patient Management** - View and manage patient records and medical history
- **Clinical Data Handling** - Support for observations, measurements, procedures, and drug exposures
- **PDF Export** - Generate patient data reports

Technologies Used

Backend

Technology	Version	Purpose
Java	21	Runtime
Spring Boot	3.1.12	Application framework
Spring Security	-	Authentication & authorization
Spring Data JPA	-	Database ORM
PostgreSQL	16	Database
JWT (jjwt)	0.11.5	Token-based authentication
Docker	-	Containerization

Frontend

Technology	Version	Purpose
Svelte	5.x	UI framework
SvelteKit	2.x	Application framework
TypeScript	5.x	Type safety
Tailwind CSS	4.x	Styling
Vite	7.x	Build tool

Getting Started

Prerequisites

- **Java 21** (for running backend without Docker)
- **Node.js 18+** and **pnpm** (for frontend)
- **Docker & Docker Compose** (recommended for backend)

Running the Backend

Option 1: Using Docker Compose (Recommended)

```
cd backend/api

# Start PostgreSQL database and Spring Boot application
docker compose up -d

# View logs
docker compose logs -f

# Stop services
docker compose down
```

The API will be available at `http://localhost:8080/api`

Option 2: Running Locally

1. Start a PostgreSQL database on port 5432 with:
   - Database: `hdp_auth`
   - Username: `postgres`
   - Password: `password`
2. Run the Spring Boot application:

```
cd backend/api

# Windows
./mvnw.cmd spring-boot:run

# Linux/Mac
./mvnw spring-boot:run
```

Running the Frontend

```
cd frontend/web-interface

# Install dependencies
```

```
pnpm install

# Start development server
pnpm dev
```

The frontend will be available at `http://localhost:5173`

Building for Production

```
cd frontend/web-interface

# Build the application
pnpm build

# Preview the production build
pnpm preview
```

Project Structure

```
byowa-hdp/
  backend/
      api/
          src/main/java/dev/byowa/hdp/
              config/          # Security & app configuration
              controller/      # REST API endpoints
              dto/             # Data transfer objects
              model/           # JPA entities (OMOP CDM)
              repository/      # Data access layer
              service/         # Business logic
          src/main/resources/
              application.properties
              schema.sql       # Database schema
              fhir-examples/   # Sample FHIR data
          docker-compose.yml
          pom.xml

  frontend/
      web-interface/
          src/
              lib/
                  components/  # Reusable Svelte components
                  pages/       # Page components
              routes/          # SvelteKit routes
          package.json
          vite.config.ts
```

API Endpoints

The backend exposes REST endpoints at `http://localhost:8080/api`:

- `/auth/*` - Authentication (login, register)
- `/users/*` - User management
- `/admin/*` - Admin operations
- `/fhir/*` - FHIR data import
- `/measurements/*` - Clinical measurements
- `/providers/*` - Healthcare provider data

License

This project is part of the BYOWA (Bring Your Own Web App) initiative.