DMH Role-Based Data Visualisation Platform

System Architecture, Delivered Modules & Comprehensive Project Roadmap

Target Organisation: Deenanath Mangeshkar Hospital (DMH), Pune	**Date:** 24 November 2026
Initial Dataset: admissionData-2025_v1.csv (86,099 rows, 19 fields)	**Status:** Phases 1–3 Complete (25/25 Tests Passing)
Tech Stack: FastAPI, SQLAlchemy 2.0 Core, React 18, Apache ECharts	**Lead:** AI Engineering Team

1. Executive Summary & Background

Healthcare organisations manage high-volume operational and clinical records. Historically, this data is either circulated via raw spreadsheets (causing privacy risks, version confusion, and non-compliance) or locked behind one-off manual reports built by developers (which are expensive and slow). This platform provides a governed, self-service visualization hub where a **Super Admin** configures exact permissions, and departmental users log in to see interactive charts limited strictly to their permitted data.

2. The One Rule That Matters Most (Zero Data Leakage)

> ***THE CRUCIAL SECURITY RULE:***
> *The browser must never receive data the user is not allowed to see. All permission checks, column masking, and aggregations happen on the backend. The frontend client sends a declarative JSON chart specification, never SQL. The backend validates it against a strict whitelist, applies user/role row-level rules, runs an aggregated query via SQLAlchemy Core with bound parameters, and returns only the mathematical summary result.*

3. Technology Stack & Architectural Decisions

Layer	Technology	Architectural Rationale & Implementation
Backend API	FastAPI (Python 3.11+)	Async endpoints, Pydantic v2 data validation, OpenAPI autodoc
Data Engine	SQLAlchemy 2.0 Core	Dynamic typed table reflections, parameter binding, whitelisting (Zero f-string SQL)
Migrations	Alembic	System schema versioning; dynamic data tables (ds_*) excluded cleanly
Auth & Crypto	Argon2-cffi + PyJWT	GPU-resistant password hashing; 8-hour stateless signed JWT tokens
Database	PostgreSQL 16 / SQLite	Postgres for production; zero-dependency SQLite for fast isolated unit tests
Frontend UI	React 18 + TypeScript + Vite	Type-safe SPA with strict compilation checks (no unused locals/params)
Visualisations	Apache ECharts (React)	High-performance canvas/SVG charts with responsive resize and custom DMH palette
Deployment	Docker Compose + Nginx	Reverse proxy, asset compression, multi-container orchestration

4. Modules Currently Developed & Verified (Phases 1, 2 & 3)

A. Authentication & User Management (Phase 1)

- Implemented Argon2id password hashing and PyJWT token generation.
- Established 4-role RBAC hierarchy: Super Admin, Viewer, Department Manager, and Finance User.
- Built Super Admin user management CRUD (/api/v1/users) with full 403 Forbidden enforcement for non-admin requests.

B. Data Ingestion & Schema Registration (Phase 1)

- Ingested all **86,099 rows and 19 fields** from admissionData-2025_v1.csv into ds_hospital_admissions_2025.
- Robust parsing of DD-MM-YYYY HH:MM timestamps into real database DATETIME types.
- Automatic sensitive column tagging: direct patient identifiers (mrdNumber) and doctor names flagged as sensitive by default; bedNumber handled as text.

C. Access Control Matrix & Rule Intersection (Phase 2)

• Three-tiered access control: Dataset grants, Column masking, and Row-level filter rules.
• Mathematical AND-rule intersection: users inherit all applicable role and user-specific constraints.
• Super Admin **'Preview As' Simulation Tool**: Allows administrators to simulate restricted roles (Medicine only, TPA only) with instant live UI re-rendering.

D. Declarative Query & Aggregation Engine (Phase 3)

• Accepts client JSON ChartSpec, verifies whitelist, injects parameterized WHERE clauses.
• Rejects restricted dimensions/measures with HTTP 403. Executes aggregations (COUNT, SUM, AVG) via SQLAlchemy Core.

E. Interactive Apache ECharts Dashboard (Phase 3)

• **4 Executive KPIs:** Total Admissions, Average Patient Age, Active Clinical Specialities, Hospital Wings.
• **5 Core Charts:** Admissions by Medical Speciality (Bar), Hospital Building Breakdown (Donut), Monthly Volume Trend (Area), Admission Categories (Pie), and Patient Payer Mix & Demographics (Stacked Bar).
• **Dynamic Filter Bar:** Permission-scoped dropdowns for Building, Speciality, and Admission Type.

5. Live Dataset Verification Metrics (86,099 Records)

Role / Persona	Enforced Access Boundaries	Verified Output Metric	Status
Super Admin	Unrestricted hospital-wide access	**86,099** admissions across all 19 columns	**PASSED**
Top Specialities	Hospital rankings (Unrestricted)	Oncology: 18,536 | Medicine: 9,770 | Eye: 5,366	**PASSED**
Department Manager	Row rule: Speciality = 'MEDICINE'	**9,770** admissions (Non-Medicine rows excluded)	**PASSED**
Finance User	Row rule: PatientCategory = 'TPA'	**32,056** claims (Cash/GNL rows excluded)	**PASSED**
Restricted Viewer	Column masking (mrdNumber, Doctor)	HTTP 403 Forbidden on sensitive fields	**PASSED**
Automated Tests	Pytest test suite (backend/tests)	**25/25 unit tests passing** (7.27 seconds)	**PASSED**

6. Remaining Roadmap & Planned Enhancements

Phase	Module Title	Scope & Planned Functionality	Status
Phase 3.5	Advanced Healthcare Analytics & Cross-Filtering	• **Average Length of Stay (ALOS):** Days calculation per department. • **24x7 Day-vs-Hour Surge Heatmap:** Emergency surge hours visualization. • **Geographic Breakdown:** In-state vs out-of-state and district spread. • **Cross-Filtering:** Click-to-filter across all dashboard widgets. • **Image Exports:** High-resolution PNG/SVG chart export via ECharts.	**READY TO BUILD**
Phase 4	Governance, Compliance & Export Engine	• **Tamper-Evident Audit Logging:** Log user logins, queries, permission edits. • **Admin Audit Viewer:** Searchable audit trail (/admin/audit-logs). • **India DPDP Act Compliance:** Small-group suppression (k < 5 patients). • **Role-Gated Data Exports:** Watermarked Excel/CSV export gated by can_export.	**SCHEDULED**
Phase 5	Live Enterprise Connectors & Deployment	• **External Database Connectors:** Direct PostgreSQL, MySQL, SQL Server. • **Encrypted Credentials:** Symmetric Fernet encryption for connection strings. • **Aggregation Caching:** Redis / In-memory caching for sub-50ms query responses. • **Docker Compose Hardening:** Production Nginx reverse proxy & containers.	**SCHEDULED**

7. Execution Guide & Demo Accounts

Role	Username	Password	Enforced Data Scope
Super Admin	admin	Admin@123456	Unrestricted hospital-wide access (86,099 rows)
Staff Viewer	viewer	Viewer@123456	General view; sensitive doctor and MRN masked
Dept Manager	dept_manager	Manager@123456	Medicine department admissions only (9,770 rows)
Finance Analyst	finance_user	Finance@123456	TPA insurance claims only (32,056 rows)