ENGINEERING DATA MONITORING SYSTEM

FINAL TECHNICAL DOCUMENTATION

PROJECT OVERVIEW

System Name: Engineering Data Monitoring System

Architecture: ASP.NET Core Blazor Server

Backend Database: SQLite (via Entity Framework Core)

Authentication: ASP.NET Core Identity

Frontend Styling: Tailwind CSS

Framework Version: .NET 9.0

Purpose:

This system was developed to enable secure, real-time entry, validation, analysis, and export of engineering data across multiple departments within an industrial or laboratory environment.

CORE FEATURES

1. Secure User Authentication and Role Management
2. Intuitive Data Entry Forms with Real-time Validation
3. Interactive Dashboards with Live Charts and Data Summaries
4. Filterable Data Management with Export Options
5. Multi-Unit Engineering Parameter Support
6. Parameter- and Department-Based Data Organization
7. CSV Data Export for External Analysis

DATABASE SCHEMA

- AspNetUsers – Handles authentication and user metadata
- EngineeringUnits – Department and unit information
- DataPoints – Stores recorded engineering parameters

ENGINEERING UNITS SUPPORTED

- Process Engineering (PROC-A)
- Quality Control Laboratory (QC-LAB)
- Maintenance Department (MAINT)
- Safety & Environmental (SAFE-ENV)

AUTHENTICATION SYSTEM

Default Admin Credentials:

Email: admin@engineering.com

Password: Admin123!

Capabilities:

- Role-ready architecture (Admin/User)
- Session-based secure login/logout
- Password security via Identity hashing and validation

TECHNICAL STACK

- Backend: .NET 9 (ASP.NET Core)
- Frontend: Blazor Server
- Styling: Tailwind CSS
- Database: SQLite (via EF Core)
- Realtime: SignalR-ready structure
- Charts: Chart.js
- Export: CSV via memory stream

SECURITY MEASURES

- Identity-based Authentication
- Role-based Access (Expandable)
- Input Sanitization and Data Validation
- CSRF Protection
- Secure Password Hashing
- SQL Injection & XSS Mitigation

DEPLOYMENT & PRODUCTION

- Production-Ready Configuration Files
- Environment-Specific AppSettings
- Database Migration Support
- Seed Data and Role Initialization
- Centralized Error Logging

FUTURE ENHANCEMENTS

- Transition to Firebase Realtime Database
- Automated Data Importing
- Terminology and Formatting Adjustments to ready for deployment
- Role-based permissions
- Advanced Reporting
- Data Analytics
- Mobile Responsiveness

LOCAL DEVELOPMENT INSTRUCTIONS

```
git clone https://your-repo-link

cd DataMonitoringSys

dotnet restore

dotnet run
```

Access via browser: http://localhost:5161

MAINTENANCE GUIDELINES

- Weekly database backups
- Monthly test run of critical features
- Apply security patches as needed
- Review and update third-party packages
- Collect and act on user feedback