**Project plan**

1. Project Objectives

The main project goal consists of developing an industrial sensor monitoring system which tracks and analyses sensor data directly from the source in real-time. The system requires user access management for different roles while including machine learning models for anomaly prediction.

2. Project Scope

The system will make use of Flask to develop the backend which serves to ingest data and provide API services.

User interaction components will be designed through front-end elements built with HTML/CSS/JS plus PHP.

The project needs PHP development to establish user authorization functions next to role-based access systems.

The application will store data through SQLite at the local level.

Implementation of Prophet models should be integrated into the system for anomaly detection functions.

The development includes design aspects for dashboards that should function for both users and administrators.

The system accepts CSV files and reproduces live data interactions.

3. Deliverables

- Functional API for sensor data retrieval and simulation.

- User authentication system (register, login, logout).

- Admin panel for user management.

- Dashboard with live sensor data and prediction statuses.

- Machine learning integration with forecasting.

- Project documentation and diagrams.

Testing of CSV files alongside pre-trained models constitutes part of the workflow.

4. Timeline

Week 1-2: Requirements gathering and system design.

Week 3-4: Backend API development and database setup

Week 5: Frontend dashboard and UI integration

Users will gain authentication features and administrators will receive control systems throughout Week 6.

Week 7: Machine learning model integration

Week 8: Testing, optimization, and documentation

Week 9: Final review and deployment

5. Team Roles

The project manager oversees how the work progresses alongside achieving its predefined milestones.

The Backend Developer maintains responsibility for integrating the API alongside the database system along with the ML model.

A frontend developer implements the interface design together with UI/UX features while designing responsive visual components.

The tester executes three functions: quality assurance testing, validation checks and stress testing procedures.

The documentation lead creates project documents as well as prepares reports and diagrams.

6. Tools & Technologies

- Python 3.11

- PHP (for user management and session control)

- SQLite (lightweight database for sensor storage)

- HTML/CSS/JS (responsive UI design)

- Virtual Environment and Greenlet for compatibility

- Git for version control

7. Risks and Mitigation

Database locking can be achieved through using batch processing together with retry logic.

Model training alongside fallback systems should be properly executed to handle forecast errors.

User Interface performances should be improved through mobile-first CSS and testing.

User input must be validated before saving through any system because password strings should use cryptographic hashing methods.

- File format mismatch: Define strict CSV schema for ingestion