

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	31 January 2025
Team ID	LTVIP2026TMIDS42870
Project Name	electric motor temperature prediction using machine learning
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User input Management	Input form for motor parameters (torque, current, RPM) * Input fields for temperature measurements (ambient, coolant) * Form validation with error messages for invalid inputs * Default values provided for quick testing * Support for both web interface and API input
FR-2	Temperature Prediction Engine	* Preprocess input data using StandardScaler from transform.save * Generate rotor temperature prediction in real-time * Return prediction with 2 decimal precision * Handle batch predictions for multiple data points
FR-3	Results Visualization	* Show input parameters alongside prediction for reference * Color-code temperature based on safety levels: - Green: Normal operation (< 80°C) - Yellow: Warning (80-95°C) - Red: Critical (> 95°C) * Display prediction timestamp
FR-4	API Endpoints	* GET /info - Model information (type, features, importance) * GET /api - API documentation endpoint * POST /score - JSON API for programmatic predictions * POST /predict - Alias for /score endpoint
FR-5	Model Management	* Model evaluation metrics (MAE, R ²) display * Feature importance analysis * Model versioning capability * Retraining with new data support
FR-6	Data Management	* CSV data import/export functionality * Data validation and cleaning pipeline * Train/test split automation (80/20) * Data persistence in pmsm_temperature_data.csv

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Web interface should be intuitive with clear labels and instructions * All input fields should have proper labels and placeholder text * Results should be displayed prominently within 2 seconds * Error messages should be descriptive and user-friendly * Interface should be responsive and work on mobile devices
NFR-2	Security	* Input validation on both client and server side * API endpoints protected against excessive requests (rate limiting) * No sensitive data stored in logs * CORS properly configured for API access * Environment variables for configuration (not hardcoded)
NFR-3	Reliability	* Model files should be validated before loading * Fallback mechanism if models fail to load * Consistent prediction accuracy (MAE < 5°C) * 99.9% uptime for critical prediction service * Automatic recovery from failures
NFR-4	Performance	* Web page load time < 2 seconds * Support for at least 100 concurrent users * Model training time < 30 seconds for 10,000 samples * Memory usage < 500MB under normal load * Efficient feature scaling using pre-fitted StandardScaler
NFR-5	Availability	* Health check endpoint for monitoring * Graceful degradation if models unavailable * Quick restart capability (< 10 seconds) * Port conflicts handled gracefully (fallback ports) * Comprehensive logging for troubleshooting
NFR-6	Scalability	* Docker containerization ready for orchestration * API can handle increased load with load balancing * Model files can be served from shared storage * Batch prediction support for high throughput * Caching layer can be added for frequent predictions