Project Name: ML Analytics: Forecasting Fuel Consumption (IMS)

Nerie Ann S. Bravo | BSIT 3-1

Business Requirement	Steps	Input Fields and Files	Output Display and Report	Acceptance Criteria	Priority	Dependencies	Risks
View ML Forecast Dashboard	1. User accesses the Fuel Forecasting page from the system dashboard. 2. System loads the default forecast view.	None	Display: - Forecast method and period controls - Historical vs Predicted chart - KPI cards for consumption and confidence - Model insights and recommendations	1. Page loads with accurate forecast data. 2. Visuals (charts, KPIs) are clear. 3. User sees model insights and recommendations.	Must-have	Working /api/fuel endpoint Valid historical data	1. API unresponsive 2. Data structure errors
Select Forecast Method	1. User selects a forecasting model from the dropdown. 2. System fetches updated forecast.	Method dropdown: - ML Ensemble - ARIMA - Neural Network	Updated chart and KPIs based on method selected. Recommendations and metadata change.	1. Data updates correctly per method. 2. Accuracy and metadata reflect method.	Must-have	Forecast logic in AdvancedForecasting	Incorrect forecast logic or misaligned UI updates
Adjust Forecast Period	1. User selects desired number of future months. 2. System fetches forecast data for selected range.	Period dropdown: - 1, 3, 6, or 12 months	Chart and KPIs update to reflect the new forecast range.	1. Period change triggers forecast update. 2. Chart & statistics reflect the new period.	Must-have	Period-sensitive forecast data logic	Inconsistent forecast length or outdated data
Enable Method Comparison	1. User enables "Compare Methods" toggle. 2. System displays forecast results from all 3 methods.	Toggle button: Compare Methods	Display: - Method cards (ML Ensemble, ARIMA, Neural Network) - KPIs: confidence, forecast total, MAPE, reliability	1. All methods return forecast values. 2. Metrics shown clearly side by side.	Should-have	All forecast methods implemented & valid	Forecast logic fails for one method UI layout breaks
View Model Insights & Recommendations	1. System shows metadata and recommendations after forecast is generated.	Metadata: method, trend, seasonality, reliability	Insights panel: - Icons for trend & reliability - Text for seasonality & data points - List of recommendations	1. Forecast model data is visible and makes sense. 2. Recommendations are helpful.	Must-have	Forecast metadata generated by backend or mock	Incorrect or misleading advice
Assess Data Quality	1. System analyzes available data and displays badge.	Internal logic: stats.dataQuality based on data points	Badge shown: - Poor, Fair, or Good - Color-coded	1. Quality badge reflects data count. 2. Tooltip or label explains quality.	Must-have	Accurate filtering of isActual data	Poor badge despite enough data