Product Requirements Document (PRD)

1. Executive Summary

This project delivers an AI-based electrical load forecasting and decision-support application designed to assist power system planners and operators in generation scheduling, maintenance planning, and future capacity expansion. The system forecasts electrical load for user-defined horizons using historical load data and AI-based reasoning, and presents actionable insights through an interactive web dashboard. The MVP focuses on clarity, explainability, and academic demonstrability, with a roadmap toward enterprise-grade deployment.

---

2. Problem Statement

Electrical load demand varies continuously due to time-of-day effects, seasonal changes, consumer behavior, and industrial activity. Inaccurate forecasting leads to: - Over-generation, increasing fuel costs and inefficiency - Under-generation, causing grid instability and load shedding - Poorly planned maintenance and delayed infrastructure upgrades

Traditional forecasting methods often fail to capture nonlinear temporal patterns or provide actionable planning insights. There is a need for a flexible, explainable forecasting system that supports real-world power system planning decisions.

---

3. Goals & Objectives

Primary Objective

To develop an AI-based short-term to medium-term electrical load forecasting system that predicts future demand for user-defined horizons and supports power system planning decisions.

Key Goals

- Enable forecasting for hours, days, or years ahead
- Support generation planning and unit ON/OFF decisions
- Identify low-load periods for maintenance scheduling
- Assist in future plant upgradation and capacity planning
- Provide clear visual comparison between historical and predicted load

---

4. Target Users & Personas

Primary Users

- **Grid Operators:** Monitor demand trends and ensure grid stability
- **Power Plant Managers:** Plan generator operation and maintenance
- **Energy Analysts:** Analyze demand patterns and forecast accuracy

Secondary Users (Future)

- Utility planners
- Policy and strategy teams

---

5. User Journey & Workflow

1. User accesses the web dashboard
2. Uploads historical load data (CSV) or selects an existing dataset
3. Enters forecasting parameters:
4. Forecast horizon (hours/days/years)
5. Generator capacities and constraints
6. Runs the forecast
7. Reviews outputs:
8. Predicted load profile
9. Comparison with historical data
10. Planning recommendations
11. Uses insights for operational or strategic planning

---

6. Functional Requirements

6.1 Data Input

- CSV upload with format: `timestamp, load`
- Manual input of forecasting parameters

6.2 Data Preprocessing

- Data cleaning and validation
- Signal smoothing (Savitzky–Golay filter)
- Normalization for AI processing

6.3 Load Forecasting

- AI-based forecasting using Google Gemini API
- Support for multi-horizon forecasting
- Batch, on-demand prediction

6.4 Decision Support Logic

- Compare predicted load with available generation capacity
- Recommend generator ON/OFF status
- Identify maintenance windows during low-load periods
- Highlight trends indicating need for future capacity expansion

6.5 Visualization

- Interactive graphs (actual vs predicted load)
- Dynamic scaling based on data range

- Clear indication of forecast horizon

---

7. Non-Functional Requirements

- Forecast generation time < 5 seconds
- Page load time < 3 seconds
- Explainable and transparent logic
- Scalable architecture for future extensions
- Suitable for academic and demo use

---

8. Technical Architecture

Components

- **Frontend:** Streamlit web application
- **Preprocessing & Control Logic:** Data cleaning, smoothing, parameter handling
- **AI Reasoning Engine:** Google Gemini API for forecasting
- **Prediction & Decision Module:** Planning and recommendation logic
- **Database:** Firebase for storing predicted and historical data
- **Visualization Layer:** Graphical comparison and summaries

---

9. Technology Stack

- **Programming Language:** Python
- **Frontend:** Streamlit
- **AI Engine:** Google Gemini API
- **Database:** Firebase
- **Visualization:** Matplotlib / Streamlit components

---

10. Success Metrics & KPIs

- Forecast accuracy (target ±10% for MVP)
- User adoption during demo
- Reduction in manual analysis time
- System responsiveness and reliability

---

11. Security & Compliance

- Basic authentication for MVP
- Secure handling of uploaded datasets
- Configurable data retention policy

---

12. Roadmap & Future Enhancements

Short-Term

- Multi-plant support
- Exportable reports (PDF/Excel)
- Alert and notification system

Long-Term

- Integration with SCADA/EMS systems
- Renewable-aware forecasting
- Anomaly detection and what-if analysis
- Carbon emission and sustainability metrics

---

13. Assumptions & Constraints

- Software-only system for MVP
- No real-time streaming in initial version
- Single-variable forecasting (load only)
- Decision support only; no automatic control actions

---

14. Conclusion

The proposed system bridges the gap between AI-based load forecasting and practical power system planning by transforming predictions into actionable insights through a clean, explainable, and scalable application.