Phase-1 Submission Template

Student Name: THARANI V

Register Number: 822423104082

Institution: M R K Institute of Technology

Department: Computer Science and Engineering

Date of Submission: 26/04/2025

1.Problem Statement

- ✓ *This project aims to develop a predictive model using time series forecasting techniques to accurately predict short-term and long-term energy consumption patterns.*
- ✓ *Smart grids modern electricity networks that use digital technology to monitor and manage energy flows are being developed to overcome the challenges of urbanisation, population growth and expansion of digital infrastructure.*
- ✓ *One of the biggest challenges smart grids face is **predicting how much electricity will be needed at different times of the day, week, or year**.*
- ✓ *This is where time series forecasting becomes crucial.*
- ✓ *By using historical data, we can predict future energy needs, allowing utility companies to balance supply and demand in real time, reduce dependency on non-renewable resources, and improve customer satisfaction.*

2.Objectives of the Project

- ✓ *The primary objective of this project is to* ***develop an accurate and reliable time series forecasting model*** *to predict short-term and long-term energy consumption patterns for smart grids.*
- ✓ *The outcomes of this project will support better energy management, decision-making, and grid optimization.*
- ✓ *Goals of the project:*

Analyse Historical Energy Consumption Data

Develop Forecasting Models

Short-Term Energy Demand Prediction/ Long-Term Energy Consumption Forecasting

Visualization and Insights

Support for Smart Grid Management

3.Scope of the Project

- ✓ *This project focuses on developing a predictive system that utilizes time series forecasting techniques to estimate future energy consumption patterns within a smart grid environment.*
- ✓ *The scope includes data analysis, model development, evaluation, and visualization, all using historical energy usage data.*

4.Data Sources

Source: UCI Machine Learning Repository

Data Type: Time Series

Access Type: Public

Update Frequency: Static (downloaded once and not updated in real-time)

File Format: CSV

Sampling Frequency: 1-minute intervals

Time Span: December 2006 – November 2010

5.High-Level Methodology

Data Collection

- ***Source***: *UCI Machine Learning Repository*
- ***Dataset***: *Household Power Consumption (2006–2010)*
- ***Type***: *Public, static CSV file*
- ***Ingestion***: *Loaded via pandas for ETL pipeline*
- ***Granularity***: *1-minute resolution suitable for high-frequency energy forecasting.*

Data Cleaning

- *Missing Value Handling: Forward fill, linear interpolation*
- *Timestamp Engineering: Combine Date & Time into UTC-standard datetime index*
- *Data Type Casting: Enforce numeric types using astype()*
- *Outlier Detection: Z-score or IQR-based filtering*
- *Resampling: Downsample to hourly/daily using resample() API for smoother trends*

Exploratory Data Analysis (EDA)

- *Time Series Decomposition: Trend, seasonality, residuals via statsmodels*
- *Heatmaps: Correlation between power, voltage, and sub-metering features*
- *Rolling Statistics: Moving averages and standard deviation*
- *Autocorrelation: ACF and PACF plots for time dependencies*

Feature Engineering

- *Temporal Features: Hour of day, day of week, is_weekend, season*
- *Lag Features: Previous timestep values to model memory*
- *Window Features: Rolling mean/max over 3/6/12-hour windows*
- *Fourier Transform: Encode seasonality for models like Prophet.*

Model Building

- *Classical Models: ARIMA, SARIMA for univariate forecasting*
- *Hybrid Models: Facebook Prophet for additive seasonality and holiday effects*
- *Deep Learning: LSTM/GRU for capturing long-term dependencies in energy usage*
- *AutoML Tools: Consider using PyCaret or AutoTS for baseline benchmarks*

Model Evaluation

- *Time Series Split: Rolling-origin or walk-forward validation*
- *Evaluation Metrics:*
    - *MAE (Mean Absolute Error)*
    - *RMSE (Root Mean Square Error)*
    - *MAPE (Mean Absolute Percentage Error)*
- *Residual Analysis: Check model assumptions on errors*

Visualization & Interpretation

- *Interactive Dashboards: Use Plotly, Dash, or Streamlit for user interaction*
- *Forecast Plots: Overlay predicted vs actual with confidence intervals*
- *Anomaly Detection: Highlight energy spikes or dips*
- *Explainability: Use SHAP values for feature impact (especially with LSTM models)*

Deployment

- *Deployment Mode: Optional – local dashboard or web app*

- *Tools:*
  - *Streamlit/Dash for web interface*
  - *Flask API for backend prediction service*
- *Version Control: GitHub for source code and dataset tracking*
- *Documentation: README with setup instructions, dataset license info*

6.Tools and Technologies

- ***Programming Language**: Python*
- ***Notebook/IDE** : **Google Colab***

Libraries:

Data Processing:

- *pandas – Data manipulation and time series handling*
- *numpy – Numerical operations*
- *datetime – Date/time parsing and operations*

Visualization:

- *matplotlib & seaborn – Static plots for EDA*
- *plotly – Interactive visualizations*
- *statsmodels.graphics.tsaplots – For ACF and PACF plots*

Time Series Modeling:

- *statsmodels – ARIMA, SARIMA*
- *prophet – Time series forecasting with trend/seasonality*
- *scikit-learn – Preprocessing, evaluation, and ML models*
- *tensorflow.keras – LSTM/GRU-based deep learning models*
- *AutoTS or PyCaret – Optional AutoML tools for time series*

Optional Tools for Deployment :

- *Streamlit – To build an interactive web app or dashboard*
- *Flask or FastAPI – For API-based model deployment*
- *Gradio – Lightweight interface for showcasing model predictions*

7.Team Members and Roles

1.VAISHNAVI V – 822423104084 (Data collection)
2.VITHYASHASINI S – 822423104088 (Analysing tools and technology)
3.YOGALAKSHMI V – 822423104089 (Analysing methodologies)