Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre

A Project Report
on
"KU Smart Meter Monitoring and Analytics System"

[Code No.: COMP 303]
(For partial fulfillment of Year III / Semester I in Computer Engineering)

Submitted by
Bijan Bakabal Thapa (Roll No. 7)
Manish Bhattarai (Roll No. 11)
Sadit Rasaili (Roll No. 40)
Sameep Kakshapati Shrestha (Roll No. 20)
Slok Pradhan (Roll No. 38)

Submitted to
Mr. Suman Shrestha
Department of Computer Science and Engineering

February 20, 2026

Bona fide Certificate

This project work on

“KU Smart Meter Monitoring and Analytics System”

is the bona fide work of

“

Bijan Bakabal Thapa (Roll No. 7),

Manish Bhattarai (Roll No. 11),

Sadit Rasaili (Roll No. 40),

Sameep Kakshapati Shrestha (Roll No. 20),

Slok Pradhan (Roll No. 38)

”

who carried out the project work under my supervision.

Project Supervisor

Mr. Suman Shrestha
Department of Computer Science and Engineering

Acknowledgements

This project would not have been possible without the continuous support, guidance, and encouragement of many individuals and institutions.

We would like to express our sincere gratitude to the Department of Computer Science and Engineering, Kathmandu University, for providing us with the academic environment, resources, and technical infrastructure necessary to carry out this project successfully. The department's support and encouragement played a crucial role in the completion of this work.

We are deeply indebted to our project supervisor for their valuable guidance, constructive feedback, and constant motivation throughout the development of the KU Smart Meter Analytics System. Their insights and supervision helped us refine our ideas, overcome technical challenges, and maintain the quality and direction of the project.

We would also like to extend our heartfelt appreciation to Dr. Samundra Gurung from the Department of Electrical and Electronics Engineering for his expert guidance and profound knowledge in the field of electrical systems and smart metering technologies. His technical insights, practical suggestions, and willingness to share his expertise greatly enriched our understanding and significantly contributed to the successful implementation of the project. We are especially grateful for the opportunity he provided us to explore real-world smart meter systems and apply interdisciplinary knowledge in this work.

Abstract

This report presents the design, implementation, and evaluation of a real-time Smart Meter Analytics System developed for Kathmandu University to enhance energy monitoring, reliability, and data-driven decision-making. Unlike traditional analog meters that provide only cumulative readings, the system processes high-resolution smart meter data, including energy consumption, voltage variations, load patterns, and outage events. It integrates directly with the Iammeter Smart Meter API for continuous IoT-based data acquisition using automated refresh cycles and secure JWT-based authentication.

An interactive web dashboard was developed using React for the frontend and FastAPI for the backend, enabling low-latency data handling and scalable API services. Real-time and historical data are visualized through intuitive graphs, trend analyses, and block-level comparisons. The backend uses PostgreSQL for scalable data storage, while Docker-based containerization supports modular deployment and future expansion.

In addition, AI/ML-based forecasting models analyze consumption trends, predict peak loads, assess voltage stability, and detect anomalies. Experimental results show improved energy transparency, timely anomaly detection, and actionable insights for campus energy planning and sustainability.

Keywords: *Smart Meter, Real-Time Monitoring, Energy Analytics, IoT, Forecasting, Data Visualization, Smart Grid*

Contents

Acknowledgements **ii**

Abstract **iii**

List of Figures **vi**

Abbreviations **vii**

1 Introduction **1**
- 1.1 Background 1
- 1.2 Objectives 1
- 1.3 Significance and Motivation 2
- 1.4 Expected Outcomes 2

2 Related Works **3**
- 2.1 ThingsBoard 3
- 2.2 Kaa IoT 4
- 2.3 OakMeter 5

3 Methodology **6**
- 3.1 Development Phases 6
  - 3.1.1 Planning and Analysis 6
  - 3.1.2 Research and Learning 6
  - 3.1.3 Development 7
  - 3.1.4 Documentation 9

4 System Walkthrough **10**
- 4.1 Application Overview 10
- 4.2 Dashboard Page 11
- 4.3 Analysis 12
- 4.4 Billing 13
- 4.5 Map 14
- 4.6 Individual Meter Details 15
- 4.7 Prediction 16
- 4.8 Login 17
- 4.9 Admin Dashboard 18
- 4.10 Super Admin Dashboard 19

5 Results **20**
- 5.1 Features 21

6 Conclusion and Recommendations **22**
6.1 Limitations 22
6.2 Future Enhancements and Recommendations 23

References **24**

Appendix A **25**
A.1 Additional Figures and Screenshots 25
A.2 Extended Model Evaluation Results 28
A.2.1 Random Forest Performance Metrics 28
A.2.2 Actual vs Predicted Power Plot 28
A.3 Project Planning and Implementation Schedule 29

List of Figures

2.1 Dashboard Interface of ThingsBoard IoT Platform 3
2.2 Kaa IoT Smart Meter Monitoring Dashboard 4
2.3 OakMeter UI 5

4.1 Context Diagram 10
4.2 System Flow Diagram of Dashboard Page 11
4.3 System Flow Diagram of Analysis 12
4.4 System Flow Diagram of Billing Page 13
4.5 System Flow Diagram of Map Page 14
4.6 System Flow Diagram of Individual Meter Details 15
4.7 System Flow Diagram of Prediction Page 16
4.8 System Flow Diagram of Login 17
4.9 System Flow Diagram of Admin Dashboard 18
4.10 System Flow Diagram of Super Admin Dashboard 19

6.1 Dashboard 25
6.2 Analysis Page 25
6.3 Prediction Page 26
6.4 Billing Page 26
6.5 Map Page 27
6.6 Super Admin Page 27
6.7 Meter Page 28
6.8 Actual vs Predicted Power Plot for Random Forest Model 29
6.9 Project Implementation Gantt Chart 30

Abbreviations

VR	Virtual Reality
RAM	Random Access Memory
DMA	Dynamic Memory Allocation
ER	Entity-Relationship
ML	Machine Learning
API	Application Programming Interface

Chapter 1: Introduction

1.1 Background

Modern energy systems increasingly rely on real-time data to improve efficiency, reliability, and operational transparency. Traditional analog energy meters are limited to measuring cumulative energy consumption and offer little insight into usage patterns, voltage stability, load variations, or outage occurrences. As energy demand grows and electrical networks become more interconnected, the need for intelligent, data-driven monitoring systems has become increasingly important.

Smart meters are advanced digital devices capable of recording high-resolution electrical parameters, including energy consumption, voltage levels, power balance, load variability, and outage duration. These measurements are transmitted through communication interfaces or application programming interfaces (APIs) to cloud-based platforms or local monitoring systems, enabling real-time analysis and visualization. Kathmandu University seeks to utilize this smart meter data to improve campus-wide energy management while also creating enhanced opportunities for academic research and learning in smart grid technologies.

In this project, a fully interactive web-enabled dashboard was developed to collect and visualize real-time smart meter data using secure API-based communication. The dashboard presents key metrics such as current energy consumption, monthly usage trends, voltage stability indicators, historical outage summaries, and projected energy demand. By providing comprehensive visual analytics and real-time monitoring, the system supports informed decision-making, improves energy efficiency, and serves as a practical platform for education and research in modern energy systems.

1.2 Objectives

The primary objectives of this project are:

i. Develop a scalable dynamic web-based platform for real-time visualization of smart meter data, including energy consumption, voltage range, outage events, and power balance / power ratio.

ii. Integrate secure APIs and cloud-based storage to enable automatic data refresh, reliable data management, and support for multiple smart meters.

iii. Provide advanced analytics with monthly summaries, detailed reports, and interactive graphical representations for effective energy monitoring and decision-making.

iv. Implement predictive energy forecasting models to deliver insights into future energy consumption patterns and load trends.

1.3 Significance and Motivation

Efficient energy management systems are becoming increasingly important to support modern infrastructure, especially among educational and research institutions. Kathmandu University operates a wide spectrum of laboratories, student hostels, offices, and facilities where the role of electrical energy consumption is vital to operations. When energy usage and conditions are not continuously witnessed, it can be difficult to detect issues that include but are not limited to line voltage drop, wasted energy, a power line draw attached to a 'ghost load,' or system component failure occurring from a local or distributed outage. This project is motivated to address the feasibility requirement of a transparent, datadriven energy monitoring platform that is beneficial to administrators on university and students' learning opportunities. The monitoring system translates raw smart-meter data into unobtrusive visualizations for better improved decision making, reliability of power, and support for sustainability initiatives. Furthermore, this presents an academic asset where students can study real-world applications of an API, data visualization, smart technologies, and IoT-related monitoring based systems.

The capability of this system is not limited to Kathmandu University, and with future scaling considerations, it could lead to households, industries, or work related to smartgrid initiatives, making it a relevant, meaningful, and impactful technological contribution.

1.4 Expected Outcomes

The successful completion of the project will deliver following key outcomes:

i. A Dynamic Web Dashboard A web-based interface that is responsive and fully functional to deliver progressive smart meter readings for energy consumption, voltage, power balance, and outage information in real-time.

ii. Automated Data Retrieval System An API-based data collection system built to reliably and accurately collect smart meter measurements dynamically.

iii. Comprehensive Visual Analytics Different visual representations of data for clear insights and decision making.

iv. Forecasting and Predictive Insights Forecasting models in the system to predict energy consumption, voltage trends and risk of outage for planning and optimization.

v. Scalable System Architecture An extensible and modular system with the capability to add other smart meters and expand the system over the Kathmandu University

Chapter 2: Related Works

In exploring the development of smart meter data visualization and energy monitoring platforms, reviewing existing systems is essential to understanding the current technological landscape and identifying gaps that the proposed project seeks to address. This chapter presents an overview of several relevant platforms used in energy monitoring and smart metering, analyzing their key features, strengths, and limitations.

2.1 ThingsBoard

ThingsBoard is a popular open-source IoT platform that supports real-time device monitoring, data ingestion, and customizable dashboards for energy consumption and voltage analytics. While it provides strong general-purpose IoT capabilities, including scalability and protocol support such as MQTT, it is not specifically designed for detailed smart meter analysis. Its dashboards require technical expertise and lack focused features like monthly outage visualization, power-balance tracking, and simplified user flows. In contrast, our project aims to deliver a more tailored, domain-specific solution optimized for smart meter data interpretation with intuitive, energy-centered analytics.

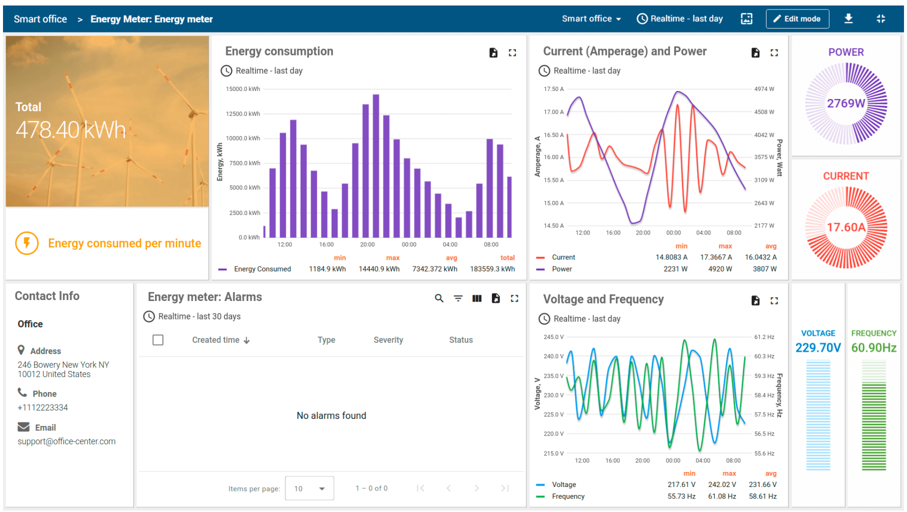

Figure 2.1: Dashboard Interface of ThingsBoard IoT Platform

2.2 Kaa IoT

Kaa IoT offers a commercial smart metering solution with real-time consumption tracking, device management, and enterprise-level dashboards. Although powerful, the platform is proprietary and limits customization, making it less suitable for academic or flexible development needs. Its analytics are broad and enterprise-oriented rather than focused on detailed technical insights like localized outage patterns or custom power-balance computations. Our project seeks to address this gap by developing an open and customizable energy visualization platform that empowers users to perform fine-grained analysis and adapt the system to various deployment environments, from small installations to academic research.

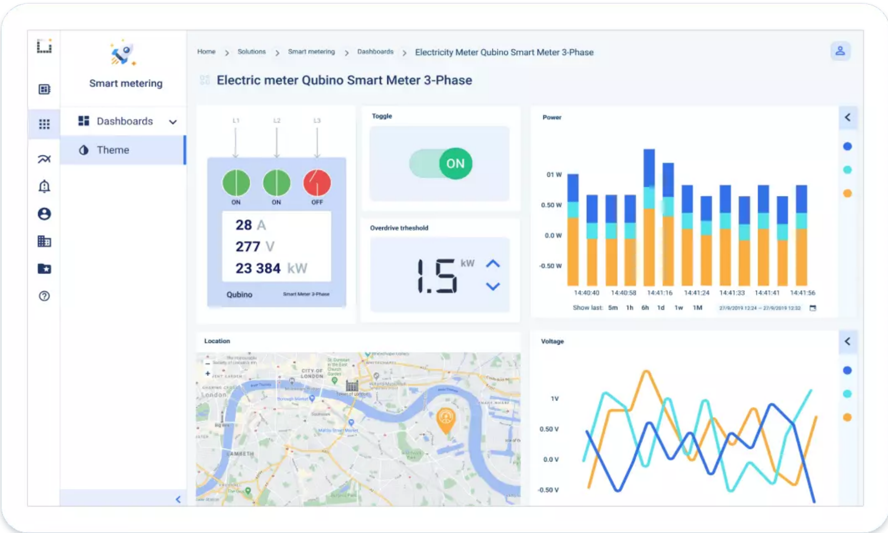

Figure 2.2: Kaa IoT Smart Meter Monitoring Dashboard

2.3 OakMeter

OakMeter provides accessible interfaces for viewing electricity usage, peak load alerts, and basic consumption graphs. However, these applications offer limited analytical detail and typically do not include advanced features such as voltage trend analysis, monthly outage summaries, or technical power-balance metrics. They also function as closed ecosystems with little room for developer-level extension. Our project seeks to enhance this space by combining user-friendly visualization with richer analytics and smart meter–specific insights, offering a more flexible and technically robust alternative.

Figure 2.3: OakMeter UI

Chapter 3: Methodology

The development of the KU Smart Meter Analytics System followed a structured and iterative approach to ensure that the final system was functional, user-friendly, and reliable. The system leveraged Immeter's smart meter API for real-time data acquisition and integrated an AI-based forecasting model to predict future electricity consumption patterns. An Agile development methodology was adopted throughout the project, enabling flexibility in design, continuous refinement of system features, and regular validation at each development stage.

3.1 Development Phases

3.1.1 Planning and Analysis

The initial phase of the project focused on identifying both functional and non-functional system requirements. This phase involved conducting surveys and interviews with key stakeholders to understand user priorities, expectations, and practical requirements for the system. In particular, surveys were conducted with faculty members and senior students of the Department of Electrical Engineering to gather insights into smart meter usage, data relevance, and desired analytical features.

In addition, a detailed study was carried out on the operational principles of smart meters, including their connectivity, data transmission mechanisms, and typical deployment in energy monitoring systems. Existing AI-based energy forecasting solutions were reviewed to inform the design of accurate electricity consumption prediction models. Furthermore, research was conducted on the Immeter smart meter API to understand how meter data could be accessed, processed, and leveraged to develop a custom real-time analytics dashboard.

The findings from this phase guided key design decisions, including the selection of data parameters to be visualized, refresh intervals for real-time updates, and the scope of prediction capabilities. The system scope was clearly defined to include outcomes such as kWh consumption forecasting and graphical visualization of energy usage across different Kathmandu University blocks.

3.1.2 Research and Learning

The research and learning phase focused on building a strong technical and theoretical foundation required for the successful implementation of the Smart Meter Analytics System. This phase involved an in-depth study of smart meter technologies, their internal architecture, and data communication mechanisms used for real-time energy monitoring. Particular emphasis was placed on understanding how electrical parameters such as energy consumption, voltage levels, load variations, and outage events are measured, recorded, and transmitted by modern smart meters.

Extensive research was conducted on application programming interfaces (APIs) used by utility providers, with specific focus on the Immeter smart meter API. This included studying authentication mechanisms, data access patterns, refresh cycles, and data security considerations. The learning outcomes from this research informed the design of a secure and efficient data acquisition pipeline that could reliably fetch and process real-time meter data.

The team also explored modern web-based data visualization techniques and dashboard design principles to effectively present complex electrical data in an intuitive and user-friendly manner. Research was carried out on frontend technologies for real-time data rendering, backend frameworks optimized for high-performance APIs, and scalable database architectures suitable for time-series energy data.

Additionally, foundational research was undertaken in the area of AI and machine learning for energy forecasting. This included studying existing load forecasting models, time-series analysis techniques, and anomaly detection approaches commonly used in smart grid applications. The acquired knowledge guided the selection and implementation of suitable forecasting models for predicting electricity consumption trends and identifying potential irregularities.

Overall, this phase played a crucial role in bridging theoretical concepts with practical system design, ensuring that implementation decisions were technically sound, scalable, and aligned with real-world smart grid practices.

3.1.3 Development

The development phase involved the implementation of a full-stack, real-time Smart Meter Analytics System using modern web and data processing technologies. The system architecture was designed to ensure scalability, responsiveness, and efficient handling of real-time energy data collected from smart meters.

The frontend of the system was developed using React, enabling the creation of a dynamic and interactive user interface. React's component-based architecture was used to design reusable UI components for visualizing energy consumption, voltage trends, historical usage patterns, and forecasted power demand. Real-time updates were handled through periodic API polling, ensuring that the dashboard reflected the most recent smart meter readings.

The backend was implemented using FastAPI, chosen for its high performance, asynchronous request handling, and ease of API development. FastAPI served as the core data processing layer, responsible for communicating with the smart meter APIs, managing authentication, validating incoming data, and exposing RESTful endpoints to the frontend. The backend also handled data preprocessing and integration with the prediction model.

For data storage, PostgreSQL was used as the primary database, hosted on NeonDB to provide a scalable and cloud-based storage solution. The database schema was designed to efficiently store time-series smart meter data, including timestamps, energy consumption values, voltage readings, and derived metrics. This structure enabled fast retrieval of both real-time and historical data for visualization and analysis.

Smart meter data was collected by utilizing official smart meter APIs, allowing automated and continuous retrieval of electrical measurements. The collected data was processed and stored in the database for further analysis and visualization.

To support predictive analytics, a Random Forest machine learning algorithm was implemented to forecast future power consumption. Historical energy usage data was used to train the model, enabling it to capture non-linear consumption patterns and seasonal variations. The trained model was integrated into the backend, allowing the system to generate power consumption predictions that were displayed on the dashboard alongside real-time data.

The Random Forest regression model was developed using historical smart meter data stored in the PostgreSQL database. Prior to training, the dataset underwent preprocessing steps including handling missing values, timestamp normalization, and feature extraction. Time-based features such as hour of the day, day of the week, and month were derived from timestamp data to capture temporal consumption patterns. Additionally, lag-based features representing previous energy usage values were incorporated to improve prediction accuracy.

The dataset was divided into training and testing subsets to evaluate model generalization performance. The Random Forest algorithm was selected due to its ability to model non-linear relationships between input variables and electricity consumption. Multiple decision trees were constructed using bootstrap sampling, and at each split, a random subset of features was considered to reduce correlation among trees and improve robustness.

For regression prediction, the final output was computed as the average of predictions from all individual trees. If n trees produce predictions $y_1, y_2, \ldots, y_n$, the final predicted consumption $\hat{y}$ is calculated as:

$$\hat{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{3.1}$$

Hyperparameters such as the number of trees, maximum tree depth, and minimum samples per leaf node were tuned experimentally to balance prediction accuracy and computational efficiency. Model performance was evaluated using metrics including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2 score). The optimized model was serialized and deployed within the FastAPI backend, enabling real-time prediction requests from the frontend dashboard.
Overall, the development phase successfully resulted in a modular and scalable sys-

tem that integrates real-time data collection, interactive visualization, and machine learning-based forecasting to support intelligent energy monitoring at Kathmandu University.

3.1.4 Documentation

The documentation phase was carried out alongside and after system development to ensure that all aspects of the Smart Meter Analytics System were clearly recorded, understandable, and reproducible. Comprehensive documentation was prepared to describe the system architecture, design decisions, implementation details, and operational workflow.

Technical documentation was developed for both the frontend and backend components of the system. This included descriptions of the React-based user interface structure, FastAPI backend endpoints, database schema design, and data flow between system components. API documentation was generated to detail request formats, response structures, authentication mechanisms, and error handling procedures, enabling future developers to easily understand and extend the system.

Detailed documentation was also maintained for the machine learning component, outlining the data preprocessing steps, feature selection, model training process, and evaluation approach used for the Random Forest power consumption forecasting model. This ensured transparency in how predictions were generated and allowed for future model refinement or replacement.

In addition to technical documentation, user-oriented documentation was prepared to explain system usage, dashboard features, and interpretation of visualized data and forecasts. This supports effective use of the system by administrative staff, faculty members, and students.

Finally, this project report consolidates all phases of the work, including planning and analysis, research and learning, development, testing, and outcomes. The documentation phase ensures long-term maintainability of the system and facilitates future research, system upgrades, and wider deployment across Kathmandu University facilities.

Chapter 4: System Walkthrough

The detailed walk-through of the developed platform's key features and user interaction is explained below.

4.1 Application Overview

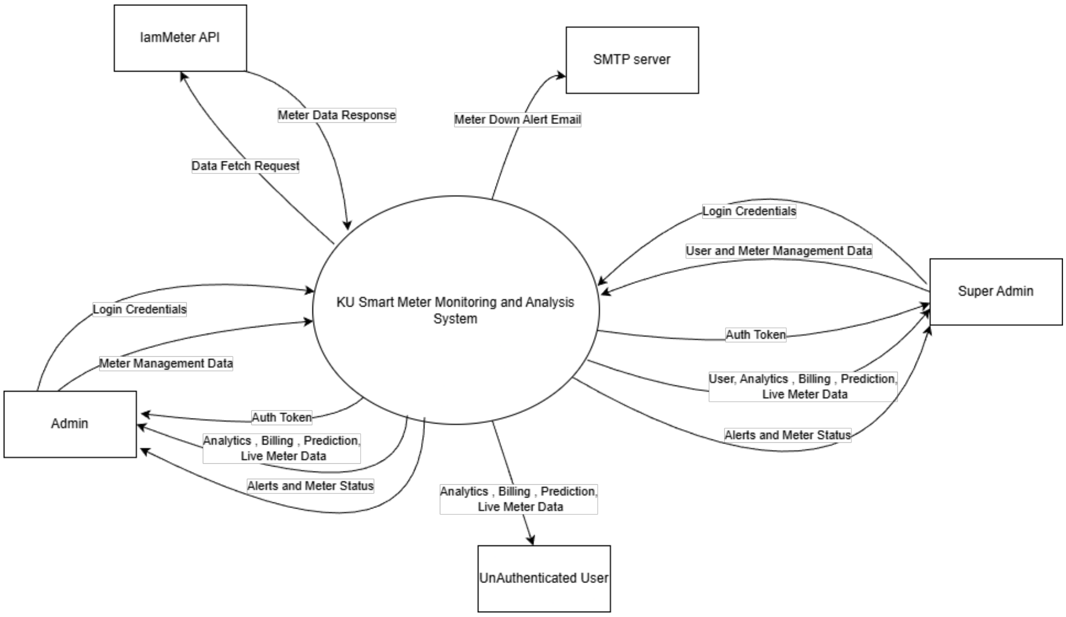

Figure 4.1: Context Diagram

The KU Smart Meter Monitoring and Analytics System is designed to provide real-time monitoring, analysis, and forecasting of energy consumption across Kathmandu University. It collects meter data (current, voltage, power, and energy) from the IamMeter API every five minutes and stores it for processing. The system offers dashboards for visualizing consumption trends, voltage and current patterns, billing reports, and machine learning based 24-hour power forecasts.

Role-based access ensures Super Admins can manage users and system configurations, Admins can manage meters and receive alerts, while unauthenticated users can access analytics and predictions. The system also includes a health monitoring feature that detects meter inactivity and automatically notifies administrators via SMTP email alerts, enabling efficient and proactive energy management.

4.2 Dashboard Page

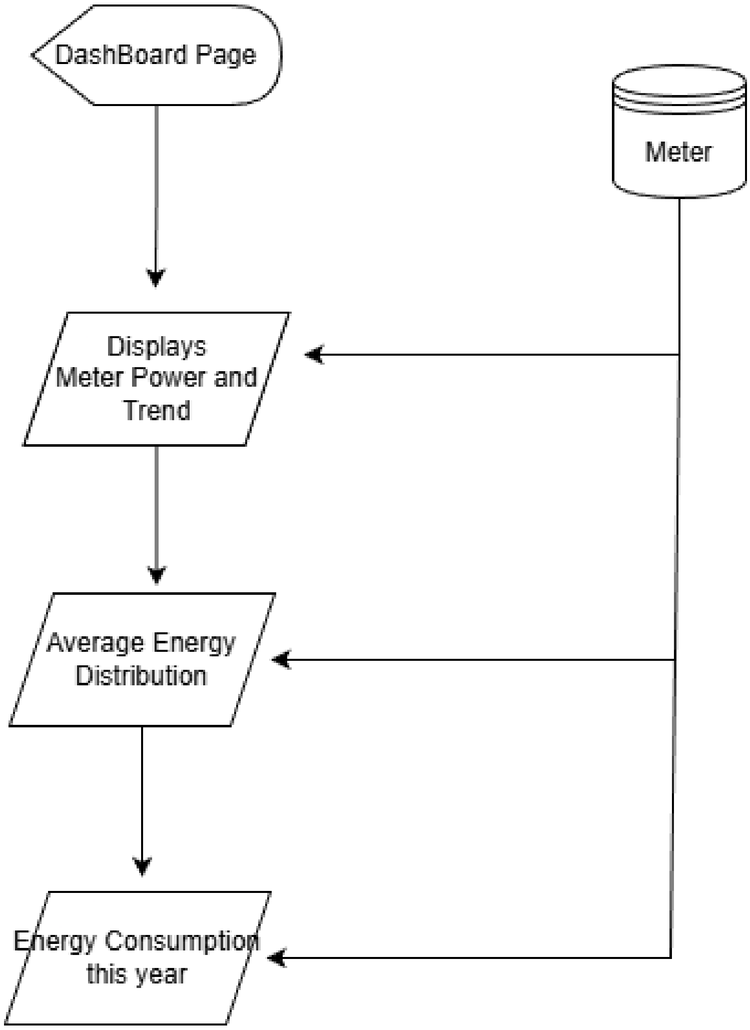

Figure 4.2: System Flow Diagram of Dashboard Page

The dashboard page retrieves and displays comprehensive meter data, including meter power consumption trends, average energy distribution across the system, and annual energy consumption statistics. It serves as the primary overview interface for monitoring system-wide energy metrics.

4.3 Analysis

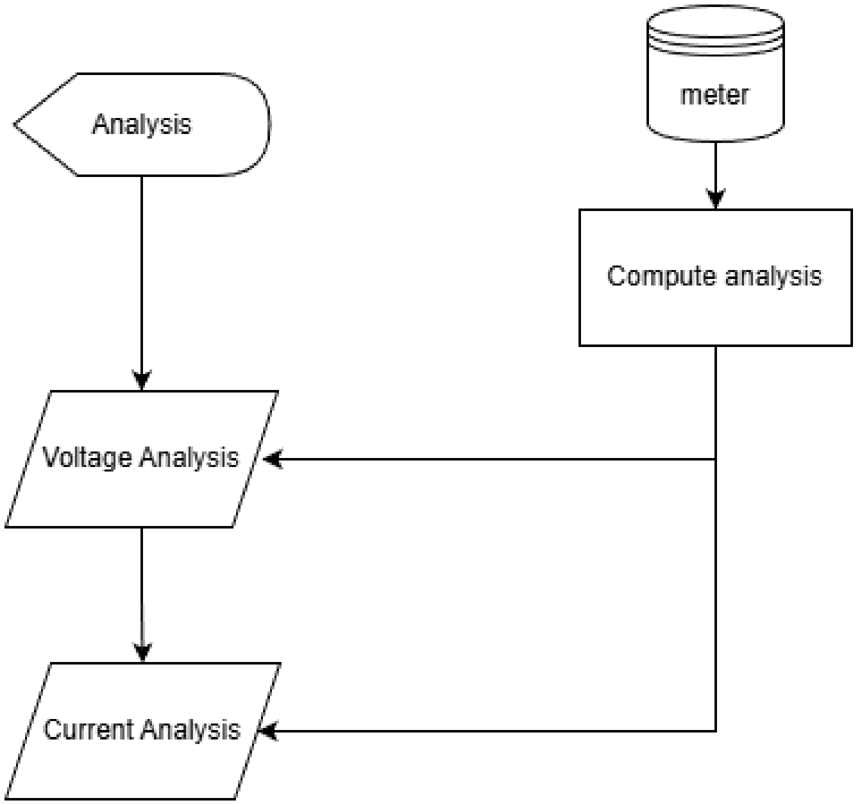

Figure 4.3: System Flow Diagram of Analysis

The analysis functionality computes and presents two distinct analytical views from the meter database: Voltage Analysis and Current Analysis, enabling users to examine electrical parameters and identify potential anomalies in the power distribution system.

4.4 Billing

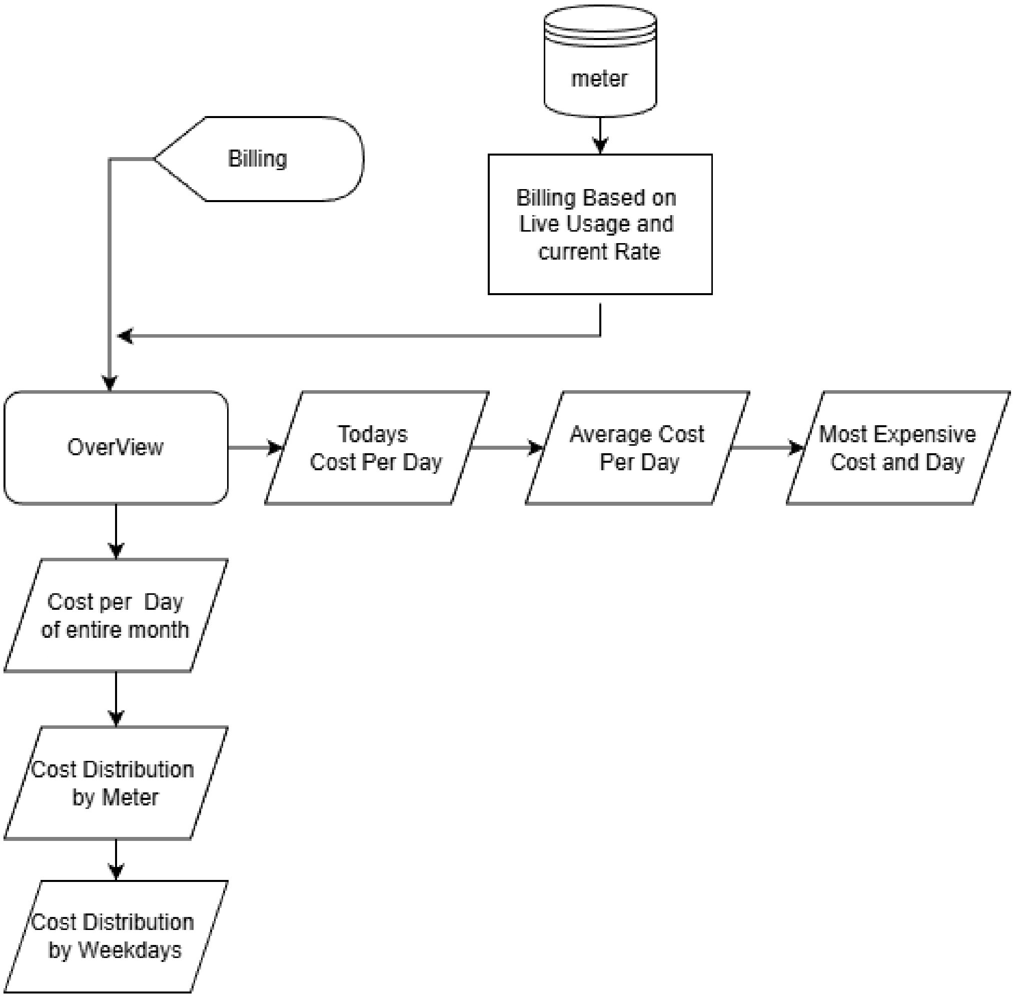

Figure 4.4: System Flow Diagram of Billing Page

This component calculates billing information based on real-time energy usage and applicable tariff rates. It provides multiple billing perspectives including an overview summary, current day cost, average daily cost, identification of peak expense periods, monthly cost distribution per day, cost distribution by individual meters, and weekday-based cost analysis.

4.5 Map

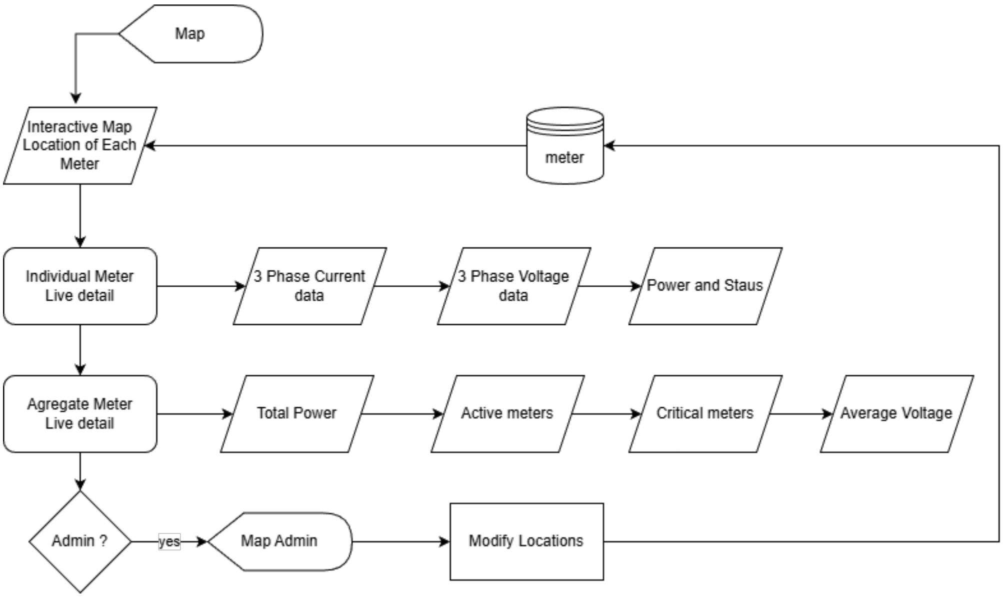

Figure 4.5: System Flow Diagram of Map Page

The map interface displays an interactive geographical representation of all meter locations. It provides real-time data visualization including individual and aggregate meter live details, three-phase current and voltage measurements, power consumption and operational status, total system power, count of active meters, identification of critical meters, and average voltage across the network. Administrative users can modify meter location data through the Map Admin interface.

4.6 Individual Meter Details

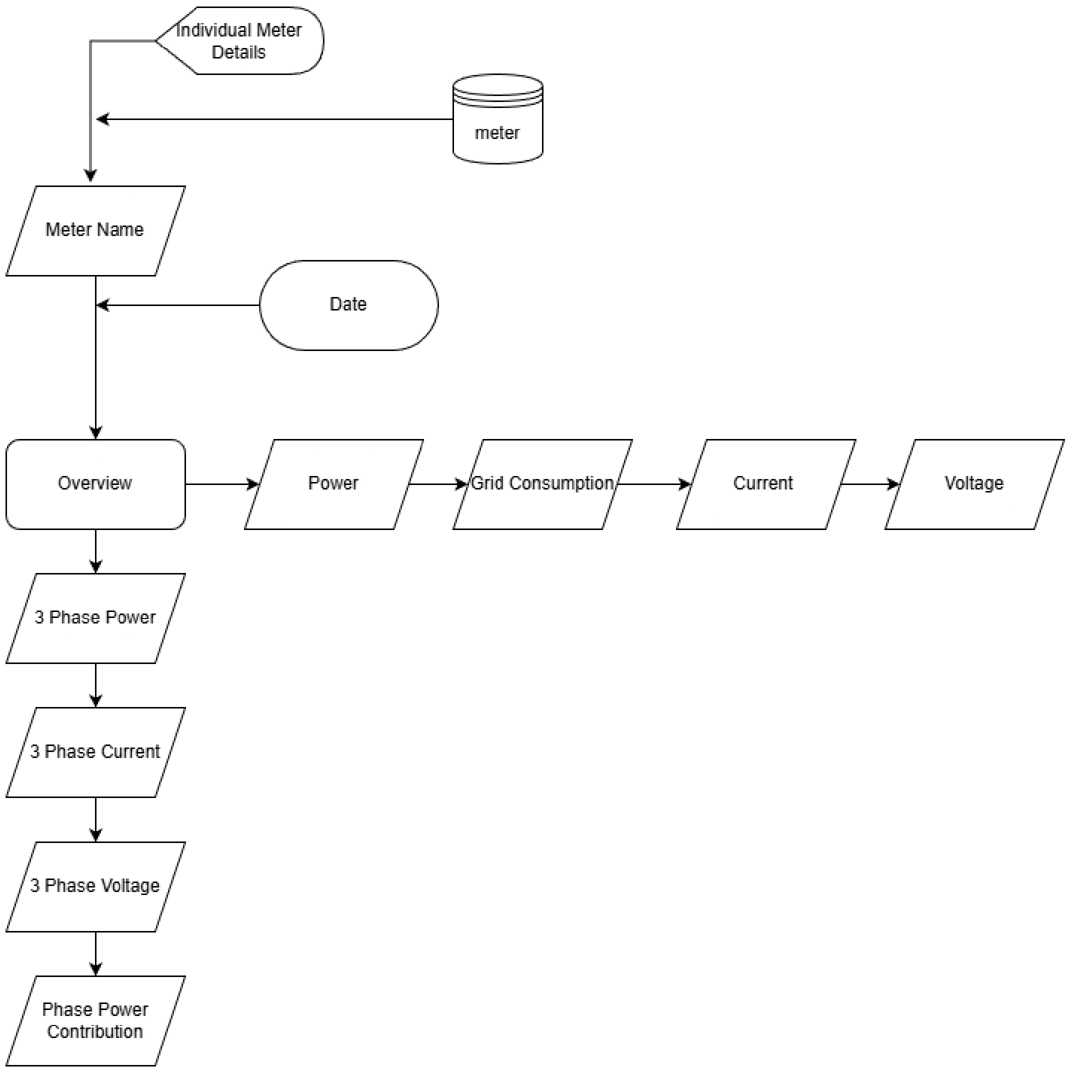

Figure 4.6: System Flow Diagram of Individual Meter Details

This allows users to select specific meters by name and date to access detailed performance metrics, including overview statistics, power consumption, grid consumption, current and voltage measurements, three-phase power, current and voltage data, and phase power contribution analysis.

4.7 Prediction

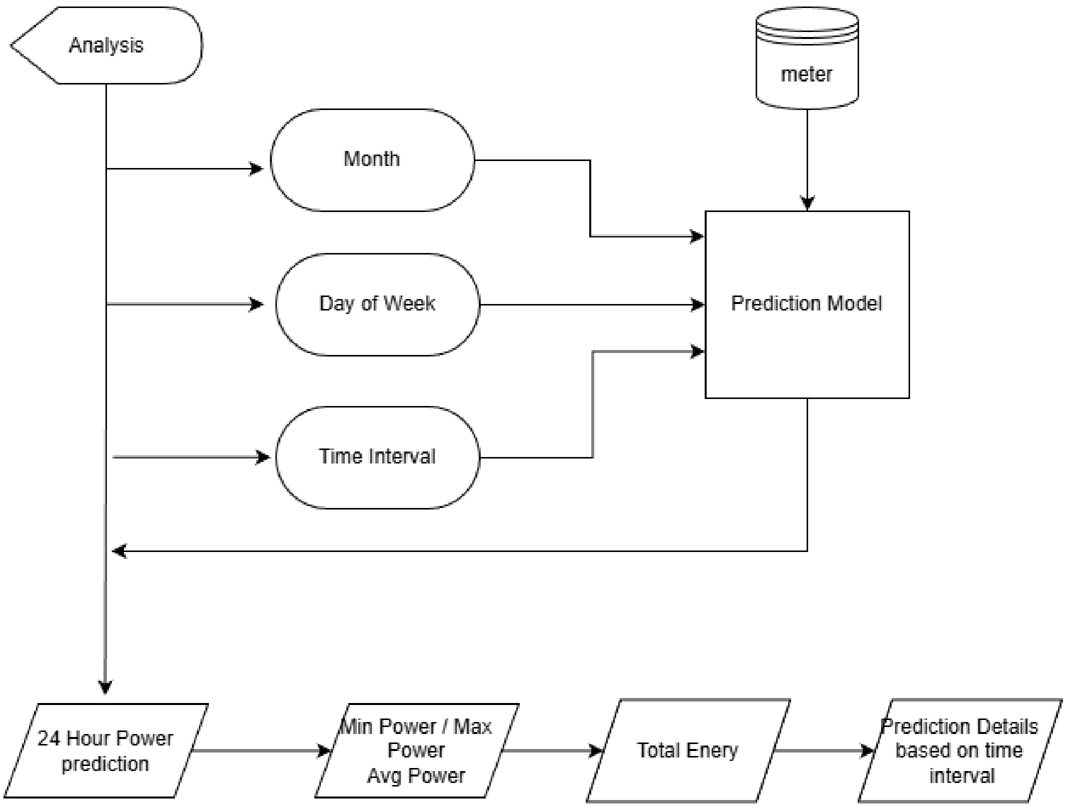

Figure 4.7: System Flow Diagram of Prediction Page

The prediction system utilizes a prediction model trained on historical meter data to forecast energy consumption patterns. Users can specify parameters including month, day of week, and time interval to generate predictions for 24-hour power consumption, minimum/maximum/average power values, total energy projections, and time-interval-based detailed predictions.

4.8 Login

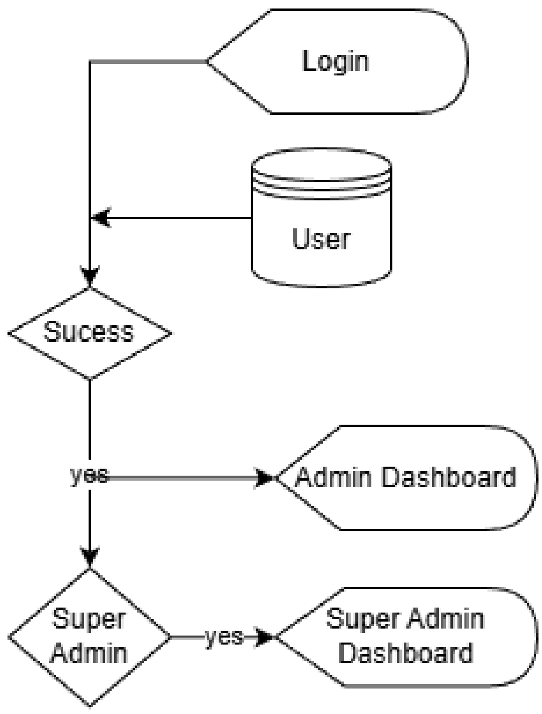

Figure 4.8: System Flow Diagram of Login

The login system authenticates users against the User database and implements role-based access control. Upon successful authentication, users are directed to their respective dashboards based on their authorization level: regular users access the standard Admin Dashboard, while Super Admin users are routed to the Super Admin Dashboard with elevated privileges.

4.9 Admin Dashboard

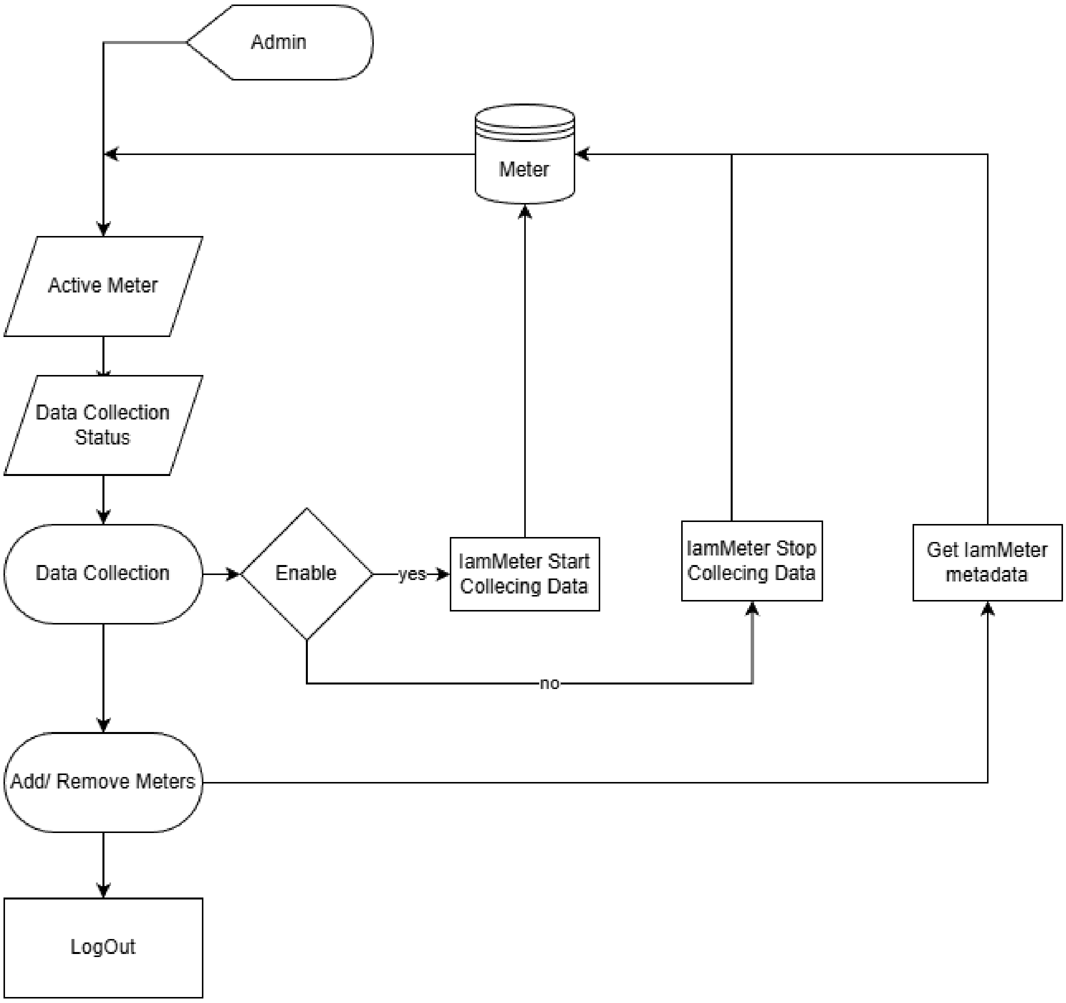

Figure 4.9: System Flow Diagram of Admin Dashboard

The administrative interface manages meter operations and data collection processes. Administrators can view active meters, monitor data collection status, and control the data collection process through enable/disable functionality. When enabled, the system initiates IamMeter data collection; when disabled, it stops data collection and retrieves IamMeter metadata. The also provides functionality to add or remove meters from the monitoring system.

4.10 Super Admin Dashboard

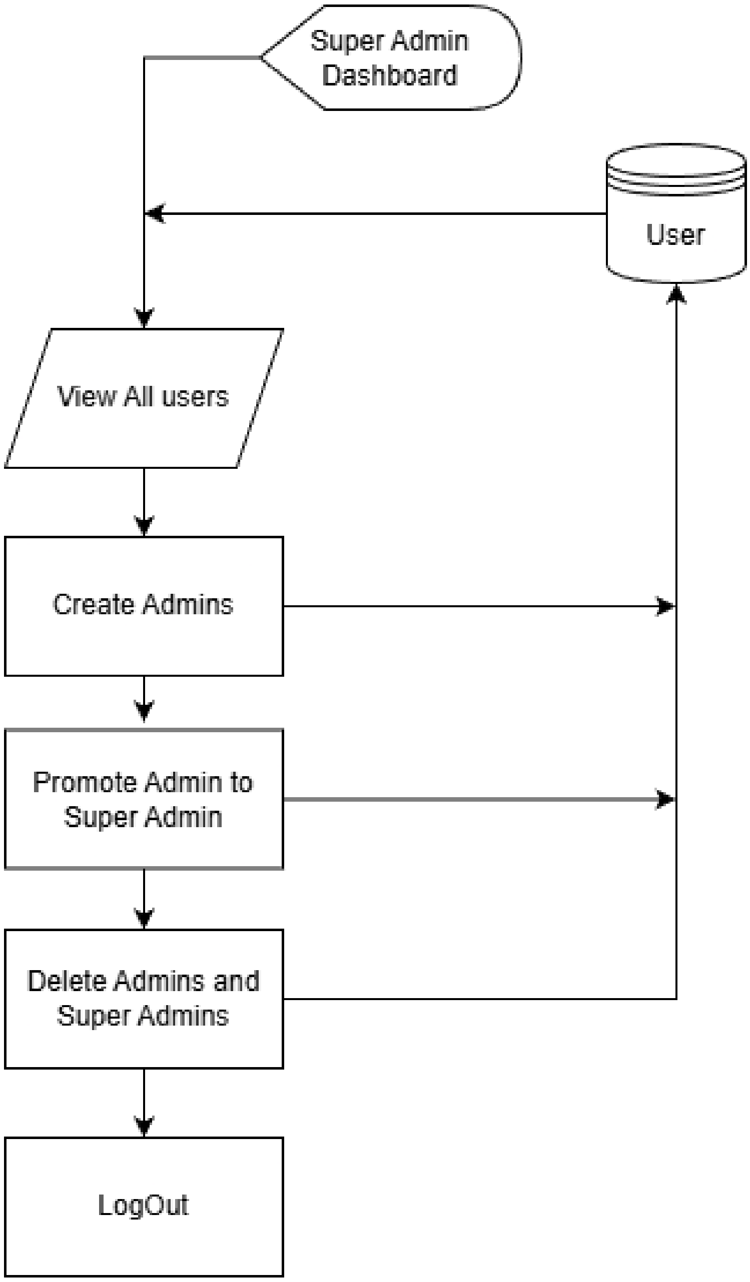

Figure 4.10: System Flow Diagram of Super Admin Dashboard

This elevated administrative manages user accounts and system administrators. Super Admin users can view all registered users, create new administrator accounts, promote existing administrators to Super Admin status, and delete administrator or Super Admin accounts, ensuring comprehensive system access control and user management capabilities.

Chapter 5: Results

A major accomplishment of the Kathmandu University Smart Meter Monitoring and Analysis System project was the successful design and implementation of a scalable, real-time energy monitoring and analytics platform tailored to an institutional environment. The system integrates directly with the Nepal Electricity Authority (NEA) Smart Meter API to retrieve high-resolution electrical data, including energy consumption, voltage levels, power balance, and outage events. Achieving reliable and secure data acquisition required careful handling of API authentication, automated refresh mechanisms, and robust error handling to ensure continuous operation despite network variability and external API constraints.

Another significant achievement was the development of a fully interactive web-based dashboard capable of visualizing both real-time and historical smart meter data. Using React for the frontend, FastAPI for the backend and PostgreSQL for the database, the project translated complex electrical measurements into intuitive charts, trend graphs, and block-level comparisons. Particular emphasis was placed on usability and clarity, allowing non-technical users such as administrators and faculty members to easily interpret energy usage patterns, voltage stability, and outage summaries without requiring prior expertise in power systems.

A key accomplishment of the project was the incorporation of AI-driven forecasting and predictive analytics into the monitoring platform. Instead of limiting the system to descriptive analytics, machine learning models were designed to forecast future electricity consumption, identify peak load periods, and analyze voltage trends. This predictive capability enables proactive decision-making, such as anticipating high-demand intervals or identifying early signs of abnormal behavior in the electrical network. Integrating these models into the backend required careful data preprocessing, aggregation, and model updating strategies to maintain accuracy over time.

The project also demonstrated strong system engineering practices through the implementation of a modular and scalable backend architecture. PostgreSQL was used to store raw, processed, and historical meter data efficiently, supporting fast querying and aggregation across multiple time scales. This ensured consistent deployment, simplified environment management, and prepared the system for future expansion across multiple buildings and smart meters within Kathmandu University. This architectural approach significantly enhances maintainability and long-term scalability.

In addition to technical implementation, the project achieved a comprehensive evaluation of existing smart metering platforms such as ThingsBoard, Kaa IoT, and OakMeter. By identifying their limitations in customization, academic flexibility, and smart-meter-specific analytics, the project clearly justified the need for a domain-focused solution. This comparative analysis strengthened the originality and relevance of the proposed system while guiding design decisions toward features most beneficial for an educational institution.

Overall, the accomplishments of this project reflect a successful integration of IoT data acquisition, web technologies, database systems, and machine learning techniques into a unified smart energy analytics platform. The system not only addresses operational energy management needs but also serves as a practical academic tool for students and researchers to explore real-world applications of smart grids, data visualization, and predictive analytics.

5.1 Features

i. **Real-Time Smart Meter Data Acquisition:**
The system fetches live energy consumption, voltage measurements, power balance metrics, and outage information from the NEA Smart Meter API. Automated polling and refresh mechanisms ensure that the dashboard remains up-to-date without manual intervention, enabling continuous monitoring of electrical conditions across university facilities.

ii. **Interactive Web Dashboard:**
A responsive web dashboard presents real-time and historical data using line charts, bar graphs, and comparative visualizations. Users can easily observe consumption trends, voltage stability, and block-wise energy usage, supporting informed operational decisions.

iii. **AI-Based Forecasting and Predictive Analytics:**
Machine learning models are integrated into the backend to forecast future electricity consumption and identify peak demand periods. These forecasts assist administrators in planning energy usage, load distribution, and potential infrastructure upgrades.

iv. **Historical Analytics and Trend Analysis:**
The system stores long-term meter data and provides aggregated views at hourly, daily, and monthly levels for analyzing historical trends. This feature enables users to study seasonal patterns, detect inefficiencies, and evaluate the impact of energy-saving initiatives.

v. **Secure Authentication and Access Control:**
JWT-based authentication and role-based access control mechanisms ensure that sensitive energy data is accessed only by authorized users. Administrators are granted elevated privileges to manage system settings, while standard users can view analytics and forecasting results.

vi. **Scalable and Modular System Architecture:**
The backend is designed with scalability in mind, allowing multiple smart meters to be integrated across different Kathmandu University blocks. The modular service-oriented architecture enables future system expansion, including advanced anomaly detection and campus-wide energy optimization.

Chapter 6: Conclusion and Recommendations

The Kathmandu University Smart Meter Monitoring and Analysis System represents a significant step toward modernizing energy management through data-driven intelligence. By integrating real-time data streams from the Immeter Smart Meter API with a scalable web-based analytics platform, the system transforms raw electrical measurements into actionable insights. Unlike traditional metering approaches that provide only cumulative readings, this system enables continuous visibility into energy consumption, voltage behavior, outage patterns, and load variations.

The project successfully demonstrates how IoT-based smart meter data, combined with modern web technologies and AI/ML forecasting models, can enhance operational transparency and decision-making. The React-based dashboard and FastAPI backend work cohesively to deliver real-time visualization, historical trend analysis, and predictive insights, supporting both administrative energy planning and academic research. Furthermore, the modular and containerized architecture ensures that the system can scale across multiple university blocks and adapt to future smart grid initiatives.

Overall, the system not only addresses Kathmandu University's immediate energy monitoring needs but also establishes a strong foundation for sustainable energy management, research, and innovation in smart grid technologies.

6.1 Limitations

Despite its strengths, the system has certain limitations that present opportunities for future enhancement:

i. **Dependence on External API Availability**
The system relies heavily on the NEA Smart Meter API for real-time data acquisition. Any downtime, increased latency, or changes in the API structure can directly affect system reliability, data availability, and the freshness of displayed information.

ii. **Forecasting Model Constraints**
The accuracy of AI/ML-based forecasting depends on the availability, consistency, and quality of historical data. Limited historical datasets or irregular smart meter readings can reduce prediction reliability, particularly for long-term forecasting scenarios.

iii. **Web-Based Access Only**
The system is accessible exclusively through a web-based dashboard and does not currently provide a dedicated mobile application or offline access. This may limit usability for stakeholders who require quick, on the go monitoring or access in low-connectivity environments.

6.2 Future Enhancements and Recommendations

To further improve system robustness, usability, and impact, the following enhancements are recommended:

i. **Advanced Anomaly Detection and Alerts**
   Incorporating real-time anomaly detection with automated notifications for voltage instability, unusual load spikes, or unexpected outages would enable faster response times and support proactive system maintenance.

ii. **Mobile Application Support**
   Developing a companion Android and iOS application would improve system accessibility and allow administrators and stakeholders to monitor energy metrics and receive alerts in real time while on the move.

iii. **Enhanced AI Models**
   Future iterations of the system could integrate more advanced forecasting techniques, such as deep learning architectures or seasonal time-series models, to improve prediction accuracy for peak load estimation and outage risk assessment.

iv. **Energy Optimization and Policy Insights**
   Beyond monitoring and forecasting, the system could provide data-driven optimization recommendations, including energy-saving strategies and load-shifting suggestions, to support long-term sustainability goals and institutional energy policy planning.

References

ThingsBoard. (n.d.). *ThingsBoard: Open-source IoT platform*. Retrieved from https://thingsboard.io/

KaaIoT Technologies. (n.d.). *Kaa IoT platform documentation and smart metering solutions*. Retrieved from https://www.kaaiot.com/

Oakter. (n.d.). *OakMeter smart energy monitoring system*. Retrieved from https://oakter.com/

Python Software Foundation. (2023). *Python documentation*. https://www.python.org/doc/

React. (2023). *React documentation*. Meta Platforms Inc. https://react.dev

IAMMETER. (n.d.). *How use IAMMETER-cloud more efficiently by the API*. IAMMETER. https://www.iammeter.com/docs/system-api

Appendix A

A.1 Additional Figures and Screenshots

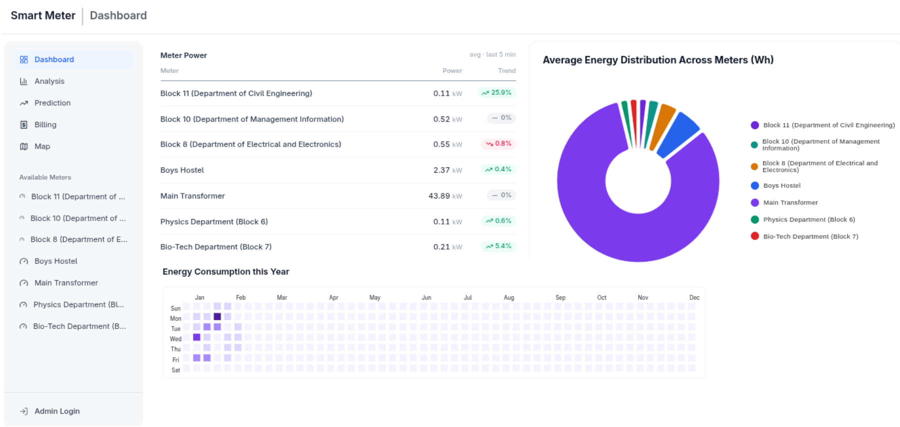

Figure 6.1: Dashboard

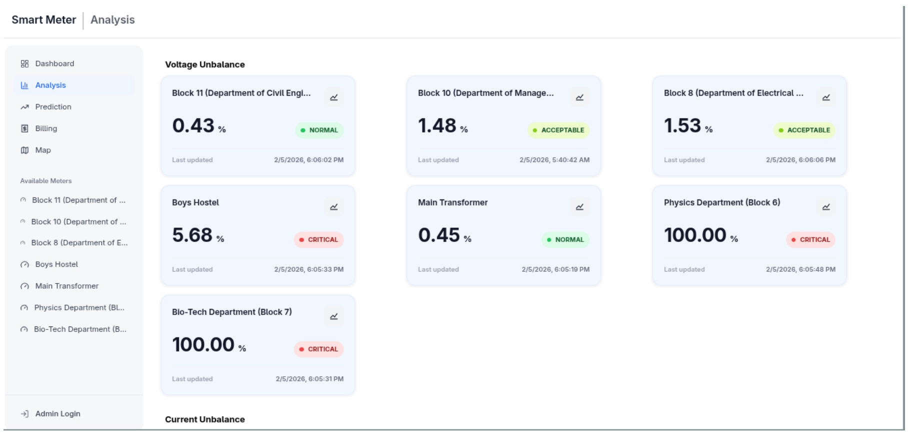

Figure 6.2: Analysis Page

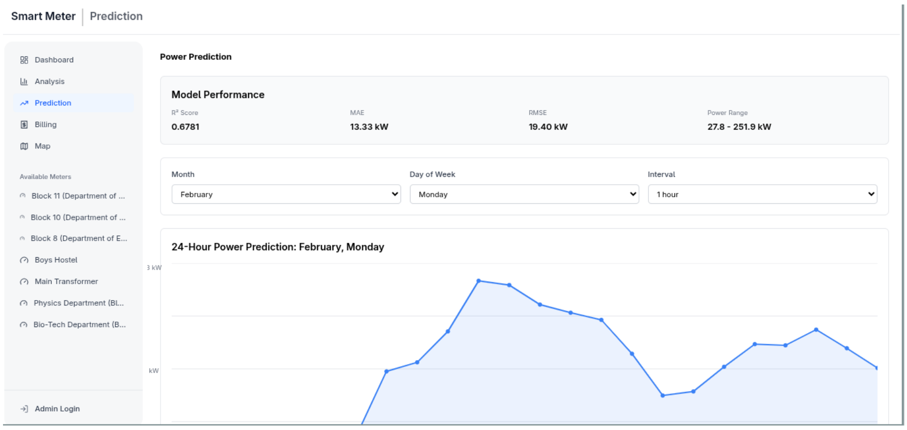

Figure 6.3: Prediction Page

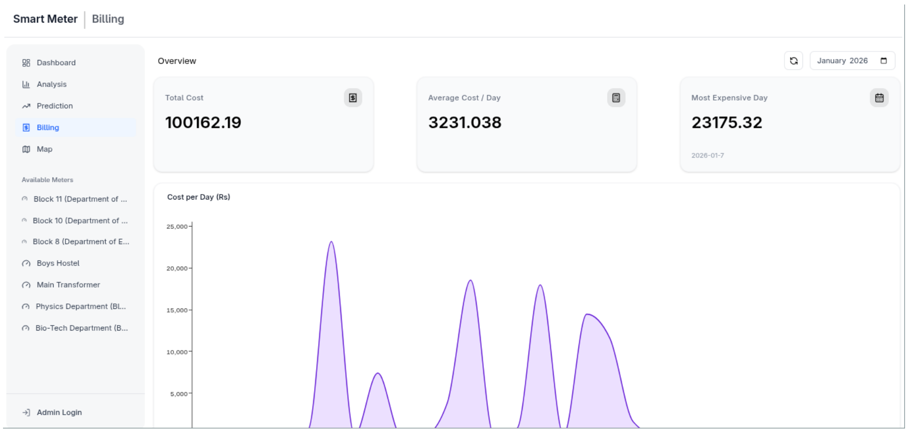

Figure 6.4: Billing Page

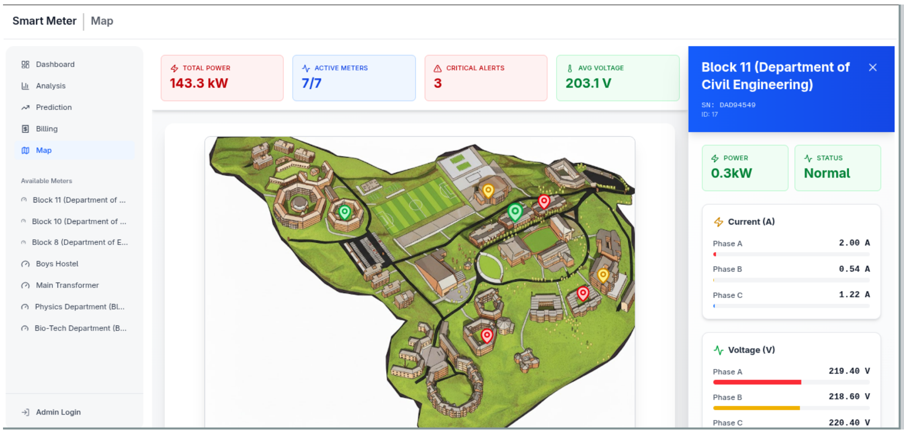

Figure 6.5: Map Page

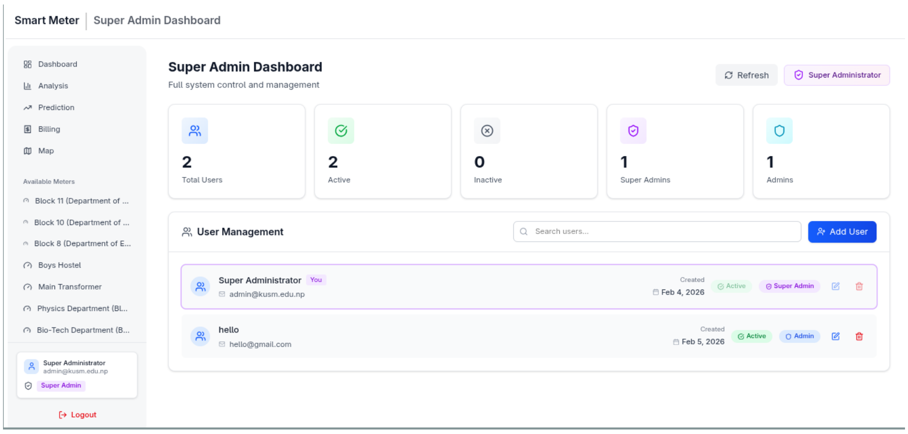

Figure 6.6: Super Admin Page

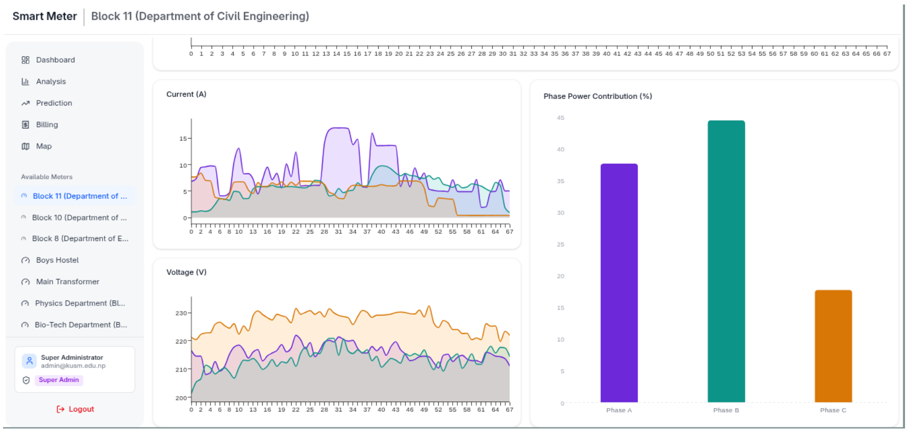

Figure 6.7: Meter Page

A.2 Extended Model Evaluation Results

This section presents detailed evaluation results of the Random Forest power consumption prediction model that supplement the main results discussed in Chapter 4.

A.2.1 Random Forest Performance Metrics

The following performance metrics were obtained from testing the trained Random Forest model on the unseen dataset:

- Total Dataset Size: 8355 samples
- Training Samples: 6684
- Testing Samples: 1671
- Mean Absolute Error (MAE): 13.26 kW
- Root Mean Squared Error (RMSE): 19.65 kW
- Coefficient of Determination (R^2 Score): 0.6777

A.2.2 Actual vs Predicted Power Plot

Figure 6.8 illustrates the relationship between actual and predicted power values. The red dashed line represents the ideal prediction line ($y = x$). The clustering of points around this line indicates the predictive performance of the model.

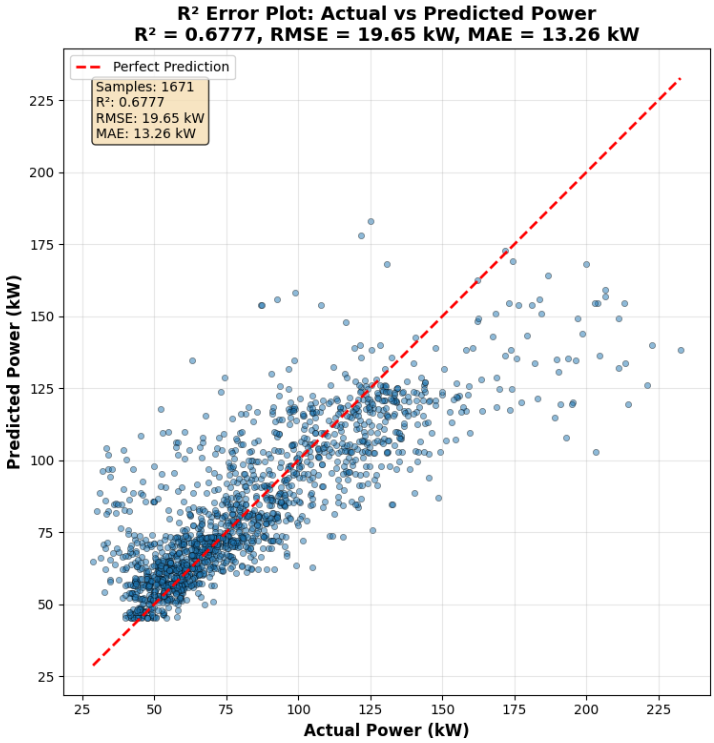

Figure 6.8: Actual vs Predicted Power Plot for Random Forest Model

A.3 Project Planning and Implementation Schedule

This section presents the detailed project implementation timeline in the form of a Gantt chart. The chart illustrates the distribution of tasks across the project duration and highlights the overlap between development phases.

The planning phase included requirement analysis and initial system design. The research and learning phase focused on studying smart meter technologies, APIs, and machine learning forecasting techniques. The development phase covered frontend, backend, database integration, and model implementation. Testing and debugging were conducted alongside development to ensure system reliability. Documentation was completed during the final stage of the project.

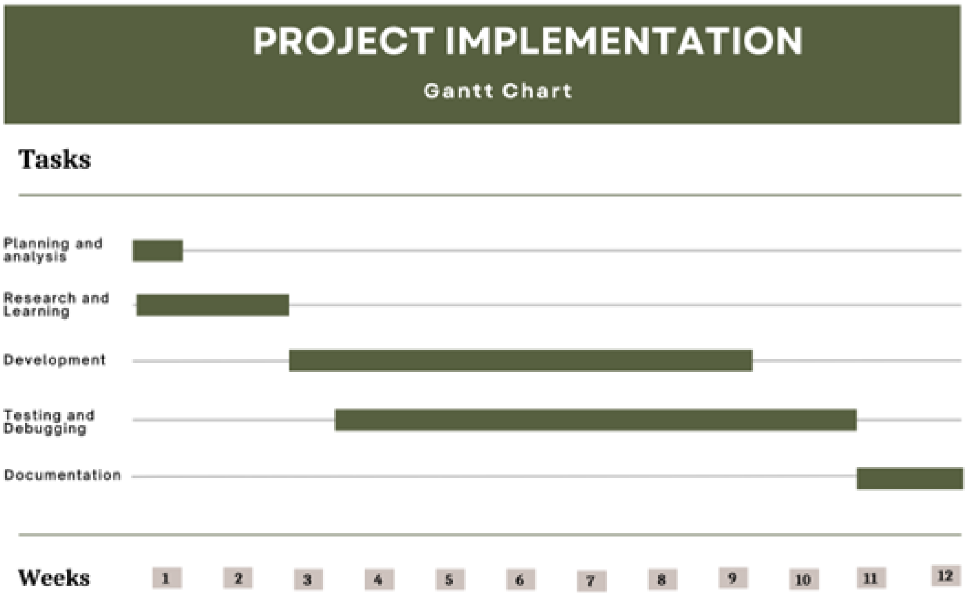

Figure 6.9: Project Implementation Gantt Chart