ENERGY CONSUMPTION AND OPTIMIZATION

A PROJECT REPORT

Submitted by

KOUSHIK VISHAL S [RA2211026010384]
PRITISH GULATI [RA2211026010370]

Under the Guidance of

(Dr.B.JOTHI)

(Associate Professor,CINTEL)

in partial fulfillment of the
requirementsfor the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING
with specialization in (Artificial Intelligence and Machine Learning)

DEPARTMENT OF COMPUTATIONAL INTELLIGENCE COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE ANDTECHNOLOGY

KATTANKULATHUR- 603 203
MAY 2025

Department of Computational Intelligence

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : **B.TECH CSE AIML**

Student Name : **KOUSHIK VISHAL S / PRITISH GULATI**

Registration Number : **RA2211026010384/RA2211026010370**

Title of Work : **ENERGY CONSUMPTION AND OPTIMIZATION**

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 18CSP107L - Minor Project [18CSP108L- Internship] report titled "**ENERGY CONSUMPTION AND OPTIMIZATION**" is the bonafide work of "**KOUSHIK VISHAL S[RA2211026010384], PRITISH GULATI[RA2211026010370]"** who carried out the project work[under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr.B.Jothi

SUPERVISOR
ASSISTANT PROFESSOR
COMPUTATIONAL
INTELLIGENCE

SIGNATURE

DR. R. ANNIE UTHRA

PROFESSOR &HEAD
DEPARTMENT OF
COMPUTATIONAL INTELLIGENCE

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Leenus Jesu Martin M,** Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We encompass our sincere thanks to, **Dr. M. Pushpalatha**, Professor and Associate Chairperson - CS, School of Computing and **Dr. Lakshmi,** Professor and Associate Chairperson -AI, School of Computing, SRM Institute of Science and Technology, for their invaluable support.

We are incredibly grateful to our Head of the Department, <<Name , Designation & Department>>, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, Panel Head, and Panel Members Department of Computational Intelligence, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, Dr. Kaavya Kanagaraj Department of Computational Intelligence, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, Dr. B. Jothi, Department of <<Dept. Name>>, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff members of Computational Intelligence, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement

Pritish Gulati

Koushik Vishal S

ABSTRACT

In the modern era of rapid urbanization and technological advancement, the demand for energy has surged significantly, leading to increased stress on power resources and raising concerns about environmental sustainability, energy wastage, and escalating utility costs. Despite the availability of renewable energy sources and energy-efficient appliances, one of the primary challenges remains the lack of intelligent systems that can actively monitor, analyze, and optimize energy consumption across various environments in real time. The "Energy Consumption and Optimization" project aims to address this critical issue by developing a comprehensive, data-driven, and user-centric platform that integrates Internet of Things (IoT) devices with advanced analytical models to facilitate smarter energy management. The system collects real-time data through sensors installed in different parts of a building or facility, which is then processed using artificial intelligence and machine learning techniques to detect anomalies, forecast consumption trends, and identify devices or operations causing unnecessary energy loss. Unlike traditional systems that only offer passive monitoring, our solution is designed to provide actionable insights, generate timely alerts, and enable both manual and automated control mechanisms that help reduce energy usage without compromising on functionality or comfort. This not only results in significant cost savings but also supports environmental conservation by minimizing carbon emissions and promoting the responsible use of energy. Furthermore, the project aligns closely with global sustainability initiatives such as the United Nations Sustainable Development Goal 7, which emphasizes access to affordable, reliable, and sustainable energy. By focusing on adaptability and scalability, the proposed system can be deployed in a wide range of settings, including smart homes, corporate offices, industrial units, and academic institutions. The modular design of the system allows it to be customized based on user needs and infrastructure capacity, while the user interface ensures ease of use for both technical and non-technical stakeholders. The ultimate objective of this project is to create a robust framework that not only enhances energy efficiency but also fosters long-term behavioral change by empowering users with insights and tools to manage their energy consumption proactively. Through this initiative, we aim to contribute to a greener future where technology and sustainability go hand in hand, and where every unit of energy consumed is tracked, justified, and optimized for the betterment of both the economy and the environment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

SEPM -Software Engineering Project Management

SDLC -Software Development Life Cycle

IEEE -Institute of Electrical and Electronics Engineers

UML -Unified Modeling Language

DFD -Data Flow Diagram

ER -Diagram Entity-Relationship Diagram

SRS -Software Requirements Specification

HLD -High-Level Design

LLD -Low-Level Design

GUI -Graphical User Interface

IoT -Internet of Things

GSM -Global System for Mobile Communications

OTP -One-Time Password

API -Application Programming Interface

GPS -Global Positioning System

PCB- Printed Circuit Board

DC -Direct Current

LCD -Liquid Crystal Display

CHAPTER 1

INTRODUCTION

1.1 Introduction to Project :

In today’s interconnected world, energy plays a vital role in sustaining economic development, supporting modern lifestyles, and powering essential infrastructure. However, the inefficient use of energy, especially in urban environments, poses serious challenges such as resource depletion, environmental pollution, and rising operational costs. The lack of real-time awareness about energy consumption patterns leads to excessive wastage, which could otherwise be mitigated through intelligent management strategies. This project, titled "Energy Consumption and Optimization," seeks to introduce a smart, technology-enabled framework for continuously monitoring energy usage, analyzing consumption behavior, and implementing optimization techniques to reduce inefficiencies. The solution integrates Internet of Things (IoT) sensors for real-time data collection with analytics and intelligent decision-making models to offer insights and control mechanisms to end users. Designed for flexibility and scalability, the system can be implemented in residential homes, commercial buildings, and industrial setups. Its main objective is to create a proactive energy management environment where users are empowered to make data-driven decisions that not only lower their costs but also contribute to global sustainability efforts by conserving power and reducing emissions.

1.2 Motivation

The inspiration for this project arises from the growing need to address the challenges associated with uncontrolled and often unnoticed energy consumption. In our daily environments—homes, workplaces, institutions—significant amounts of power are consumed inefficiently due to outdated systems, human negligence, or a complete absence of monitoring tools. This inefficiency not only inflates electricity bills but also contributes to environmental degradation through increased greenhouse gas emissions. Moreover, with the rising global emphasis on sustainable living and climate responsibility, there is an urgent requirement for smart solutions that provide both awareness and control over energy usage. The motivation behind this initiative is to leverage modern technology to bridge the gap between energy usage and optimization. By equipping users with real-time information and automated solutions, the system aims to encourage responsible consumption behavior, reduce unnecessary waste, and promote the integration of eco-friendly practices in everyday life. This project is not just about reducing costs but about taking a conscious step toward energy sustainability.

1.3 Sustainable Development Goal of the Project

The core objective of this project aligns with **Sustainable Development Goal (SDG) 7** of the United Nations, which is to ensure access to affordable, reliable, sustainable, and modern energy for all. The growing global energy demand, combined with the need for decarbonization, requires innovative systems that promote efficiency while minimizing environmental impact. Through the development of a smart energy optimization platform, this project contributes to achieving SDG 7 by enabling better energy awareness, efficient consumption, and reduced reliance on non-renewable energy sources. The system supports sustainable practices by identifying high-consumption areas, recommending alternative usage patterns, and integrating automation for optimal energy distribution. Furthermore, it cultivates a culture of conscious consumption among users, thereby fostering long-term behavioral changes essential for environmental conservation. By driving efficient energy usage, the project not only aids in reducing carbon emissions and energy costs but also promotes a future where technology and sustainability coexist harmoniously.

1.4 Product Vision Statement

1.4.1 Audience

Primary Audience:

- Homeowners and residents seeking to monitor and reduce their daily electricity usage.
- Facility managers and building administrators looking for tools to control power consumption in commercial buildings.
- Small and medium businesses aiming to reduce operational expenses and carbon footprint.

Secondary Audience:

- Government bodies and utility providers aiming to promote sustainable energy usage.
- Educational institutions and industries interested in energy research and innovation.
- Environmental enthusiasts or non-profits focused on promoting green practices.

1.4.2 Needs

Primary Needs:

- A real-time dashboard to track energy usage across different devices and locations.
- Automated recommendations or actions based on energy usage patterns.
- Alerts and notifications to identify abnormal or excessive consumption.
- Support for scheduling or remotely controlling devices to optimize usage.

Secondary Needs:

- Energy usage reports and analytics for monthly or annual reviews.
- Integration capabilities with smart grid or renewable energy sources (e.g., solar panels).
- Role-based access for different user types (e.g., admin, technician, end-user).
- Compliance tracking with energy policies and regulatory requirements.

1.4.3 Products

Core Product:

- A centralized energy monitoring platform that uses IoT sensors and smart meters to collect, process, and visualize energy consumption data in real time.
- AI-driven analytics engine that provides optimization suggestions and anomaly detection to help users make data-driven decisions.

Additional Features:

- Mobile application for remote access and control of connected devices.
- Integration with renewable energy sources and energy storage systems.
- Gamification elements like energy-saving badges or performance ratings to encourage responsible usage.
- Automated reports exportable in PDF/Excel formats for audit and review.
- Support for third-party APIs and smart home platforms like Google Home or Amazon Alexa.

1.4.4 Values

Core Values:

- Sustainability: Promote conscious energy usage and reduce environmental impact.
- Efficiency: Enable users to make informed decisions that lead to cost and resource savings.
- Accessibility: Design an interface that is user-friendly and usable by both technical and non-technical individuals.
- Innovation: Incorporate advanced technologies like AI and IoT to solve real-world energy challenges.

Differentiators:

- A predictive analytics system that not only shows current usage but also forecasts future consumption.
- Seamless integration with existing energy infrastructure and smart home systems.
- A modular design that allows customization based on user needs and facility scale.

1.5 Product Goal

The primary goal of the "Energy Consumption and Optimization" project is to design and implement a system that leverages advanced Deep Q-Learning techniques to monitor, predict, and optimize energy usage across various devices and systems. The project aims to create a smart energy optimization platform capable of real-time data analysis, providing insights into energy consumption patterns, and autonomously adjusting device operations to minimize unnecessary energy use. By leveraging the power of artificial intelligence, the system will continuously learn and adapt to changing environmental factors, energy demands, and device performance, thus improving energy efficiency over time. The key focus is not just on reducing energy consumption but also on maintaining a balance that ensures devices operate optimally, without compromising their intended functions.Additionally, the system will feature scalability to accommodate a wide range of applications, from residential to industrial environments. It will be able to handle diverse energy consumption scenarios, allowing for both centralized and distributed energy management strategies. A user-friendly interface will empower users to actively track energy usage, receive real-time alerts, and gain actionable insights to further optimize their energy consumption. By incorporating predictive analytics, the system can forecast future energy needs, providing valuable recommendations for users to adjust their usage proactively. This will ultimately contribute to more sustainable energy practices while delivering cost savings and operational efficiency.

1.6 Product Backlog

S.No	User Stories
#US 1	As a user, I want to monitor energy usage in real-time. To gain immediate insights into how much energy is being consumed.
#US 2	As a user, I want to receive alerts for high energy consumption. To be notified when energy usage exceeds predefined thresholds, enabling prompt action.
#US 3	As a user, I want to identify energy-consuming devices. To pinpoint which devices contribute the most to energy consumption.
#US 4	As a user, I want to compare energy usage across different time periods. To analyze trends and understand variations in energy consumption (e.g., daily, weekly, monthly)
#US 5	As a user, I want to receive recommendations for reducing energy consumption. To get suggestions on how to optimize energy usage and lower costs.
#US 6	As a user, I want to automate energy-saving actions. To automatically control devices or systems to minimize energy waste.
#US 7	As a user, I want to forecast future energy consumption. To predict energy needs and plan accordingly.
#US 8	As a user, I want to customize the system settings. To adapt the system to specific needs and preferences.
#US 9	As a user, I want to access the system remotely. To monitor and control energy usage from anywhere.
#US 10	As an administrator, I want to generate energy consumption reports. To analyze energy data and identify areas for improvement.

Table 1.1 Product Backlog

1.7 Product Release Plan

The following Figure 1.2 depicts the release plan of the project

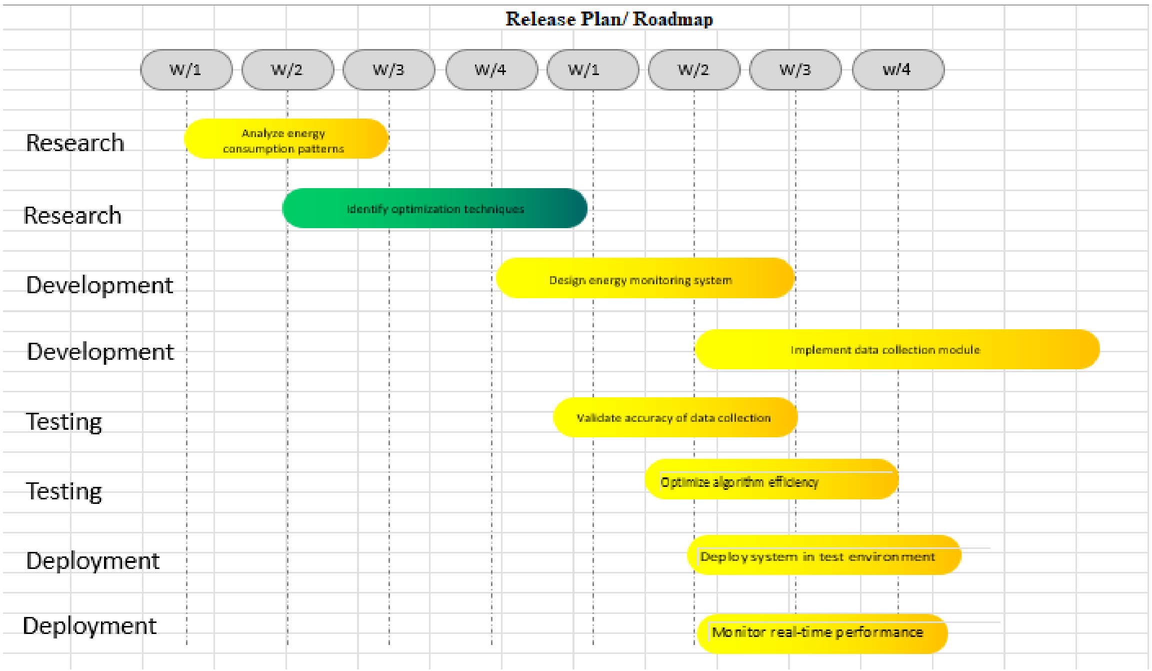


Figure 1.2 Release plan

CHAPTER 2

SPRINT PLANNING AND EXECUTION

2.1 Sprint 1

2.1.1 Sprint Goal with User Stories of Sprint 1

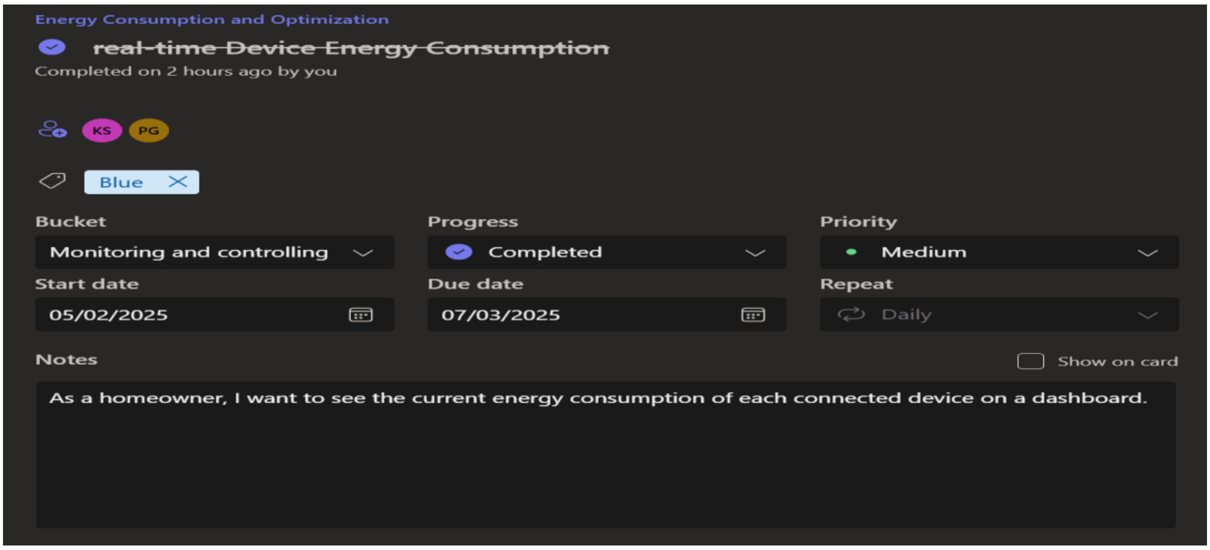
To establish the foundational energy monitoring capabilities of the system, allowing users to view real-time and historical energy consumption data for individual devices.

The following table 2.1 represents the detailed user stories of the sprint 1

Table 2.1.1 Detailed User Stories of sprint 1

S.NO	Detailed User Stories
US #1	As a homeowner, I want to see the current energy consumption of each connected device on a dashboard.
US #2	As a homeowner, I want to view a graph of my total energy consumption over the last 24 hours.
US #3	As a homeowner, I want to see a list of all connected devices with their current status (on/off).

Planner Board representation of user stories are mentioned below figures 2.1,2.2 and 2.3



Energy Consumption and Optimization

~~real-time Device Energy Consumption~~

Completed on 2 hours ago by you

Bucket: Monitoring and controlling

Progress: Completed

Priority: Medium

Start date: 05/02/2025

Due date: 07/03/2025

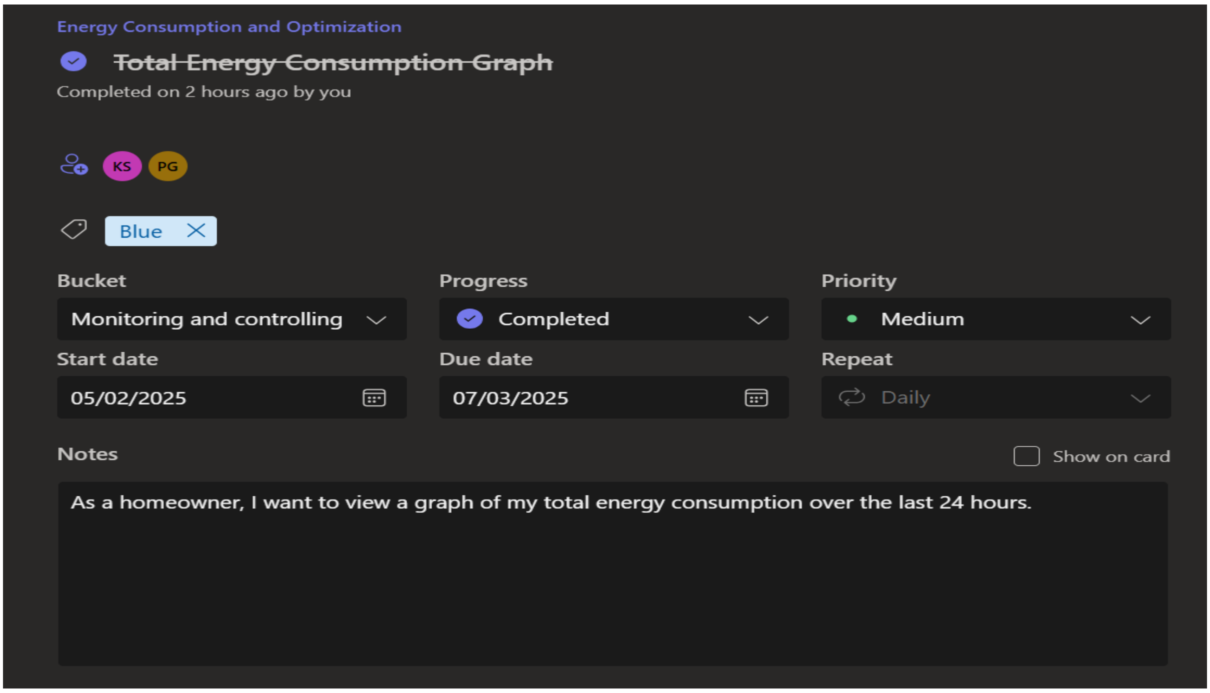
Repeat: Daily

Notes

□ Show on card

As a homeowner, I want to see the current energy consumption of each connected device on a dashboard.

Figure 2.1.1 US 1 for Sprint 1



Energy Consumption and Optimization

~~Total Energy Consumption Graph~~

Completed on 2 hours ago by you

Blue

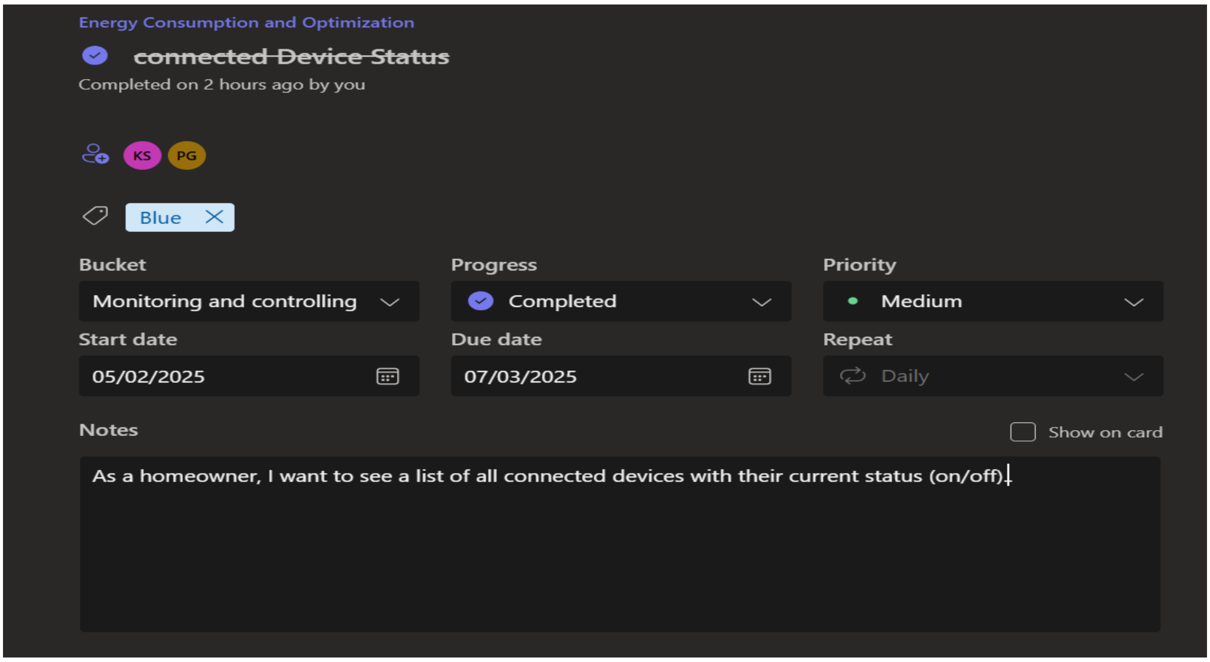
Bucket	Progress	Priority
Monitoring and controlling	Completed	● Medium
Start date	**Due date**	**Repeat**
05/02/2025	07/03/2025	Daily

Notes

□ Show on card

As a homeowner, I want to view a graph of my total energy consumption over the last 24 hours.

Figure 2.1.2 US 2 for Sprint 1



Energy Consumption and Optimization

~~connected Device Status~~

Completed on 2 hours ago by you

Blue

Bucket	Progress	Priority
Monitoring and controlling	Completed	● Medium
Start date	**Due date**	**Repeat**
05/02/2025	07/03/2025	Daily

Notes

□ Show on card

As a homeowner, I want to see a list of all connected devices with their current status (on/off).

Figure 2.1.3 US 3 for Sprint 1

2.1.2 Functional Document

2.1.2.1. Introduction

In today's world, energy consumption is a critical concern, impacting both economic stability and environmental sustainability. The increasing demand for energy, coupled with inefficient usage patterns, leads to significant resource depletion, environmental pollution, and rising costs for consumers and businesses. Often, a lack of awareness about real-time energy consumption prevents the implementation of effective conservation measures. This Functional Document outlines the requirements for the first sprint of the 'Energy Consumption and Optimization' project. This project aims to develop a smart, technology-driven system to address these challenges by providing users with the tools to monitor, analyze, and optimize their energy usage. By leveraging IoT sensors and intelligent analytics, the system will empower users to make informed decisions, reduce energy waste, and contribute to a more sustainable future. This document details the features, functionalities, and user interactions planned for Sprint 1, focusing on establishing the core energy monitoring capabilities of the system. It serves as a guide for the development team and stakeholders to ensure a clear understanding of the project's objectives and requirements.

2.1.2.2. Product Goal

The primary goal of the "Energy Consumption and Optimization" project is to develop a system that empowers users to effectively monitor and manage their energy consumption, leading to reduced energy waste and cost savings.

To achieve this, the system will:

- Provide real-time data on energy usage.
- Deliver actionable insights to improve efficiency.
- Be scalable to accommodate various environments.

2.1.2.3. Demography (Users, Location)

Users:

- Homeowners seeking to monitor and reduce their residential energy consumption.
- Small business owners aiming to lower energy costs and improve sustainability.

Location:

- Urban areas with developed smart grid infrastructure.

2.1.2.4. Business Processes

The key business processes include:

User Registration and Authentication:

- Users register securely with email/social media, and authentication verifies their access.
- This process ensures secure access to personalized content and peer-to-peer sessions.

Personalized Learning Path Creation:

- The system creates learning paths based on user interests, goals, and history.
- This guides users efficiently and effectively through relevant courses and sessions.

2.1.2.5. Features

This project focuses on implementing the following key features:

Feature 1: User Registration and Authentication

- **Description:** This feature will allow users to create secure accounts, manage their profiles, and authenticate their access to the energy management system. This ensures that only authorized users can view and control their energy data and system settings.
- **User Story:** As a homeowner, I want to register with a secure username and password so that I can access my energy consumption data and control my smart devices.

Feature 2: Real-time Energy Monitoring

- **Description:** This feature will provide users with real-time data on their energy consumption, allowing them to see how much energy is being used at any given moment. This will involve collecting data from smart meters and connected devices and displaying it in an easily understandable format.
- **User Story:** As a homeowner, I want to see a live dashboard of my home's energy usage so that I can immediately identify which devices are consuming the most power.

Feature 3: Historical Energy Consumption Analysis

- **Description:** This feature will enable users to analyze their energy consumption patterns over time. By visualizing historical data, users can identify trends, understand their energy usage habits, and make informed decisions about energy conservation.
- **User Story:** As a homeowner, I want to view charts and graphs of my energy consumption over the past week and month so that I can track changes and identify areas where I can reduce usage.

2.1.2.6. Authorization Matrix

Table 2.1.2 Access level Authorization Matrix

Role	Access Level
Administrator	Full access to user management, system configuration, advanced reporting, and all energy management features.
Facility Manager	Access to device control, alert configuration, detailed energy reporting, and user management within their facilities.
Energy Auditor	Access to detailed energy consumption data, reporting tools, and analysis features for conducting energy audits.
Homeowner	Access to real-time monitoring, basic reporting, device control, and alert configuration for their own home.

2.1.2.7. Assumptions

- It is assumed that users have access to smart devices and/or smart meters compatible with the system.
- The system's algorithms and models are assumed to provide accurate and reliable energy consumption analysis and optimization recommendations.
- It is assumed that users will provide accurate and timely feedback on the system's performance and usability.
- The system is assumed to be deployed in an environment with a stable network connection to ensure real-time data collection and accessibility.

2.1.3 Architecture Document

2.1.3.1. Application

Microservices:

The system can be designed using a microservices architecture, where different functionalities are built as independent services.

- **Data Acquisition Service:** Handles communication with smart devices and meters to collect energy consumption data.
- **User Management Service:** Manages user accounts, authentication, and authorization.
- **Analytics Service:** Processes and analyzes energy data to provide insights and generate reports.

2.1.3.2 System Architecture-

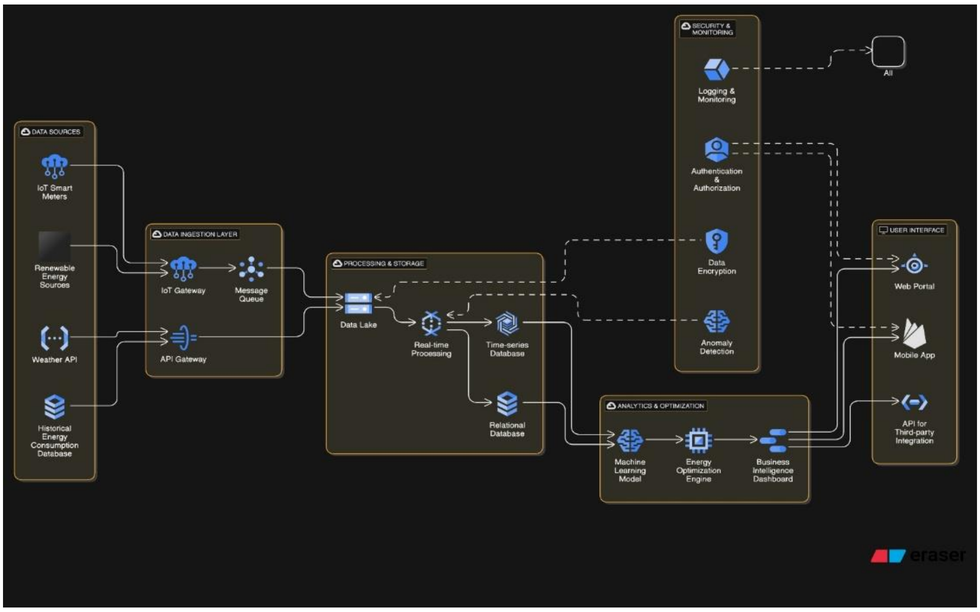


Figure 2.1.6 System Architecture Diagram

2.1.3.3. Data Exchange Contract:

Frequency of Data Exchanges:

- Sensor devices send energy usage data to the backend every 5 minutes to maintain near real-time monitoring.
- The optimization engine processes aggregated data in hourly intervals to update consumption strategies.
- The frontend dashboard queries the backend every minute to display the most recent energy metrics.

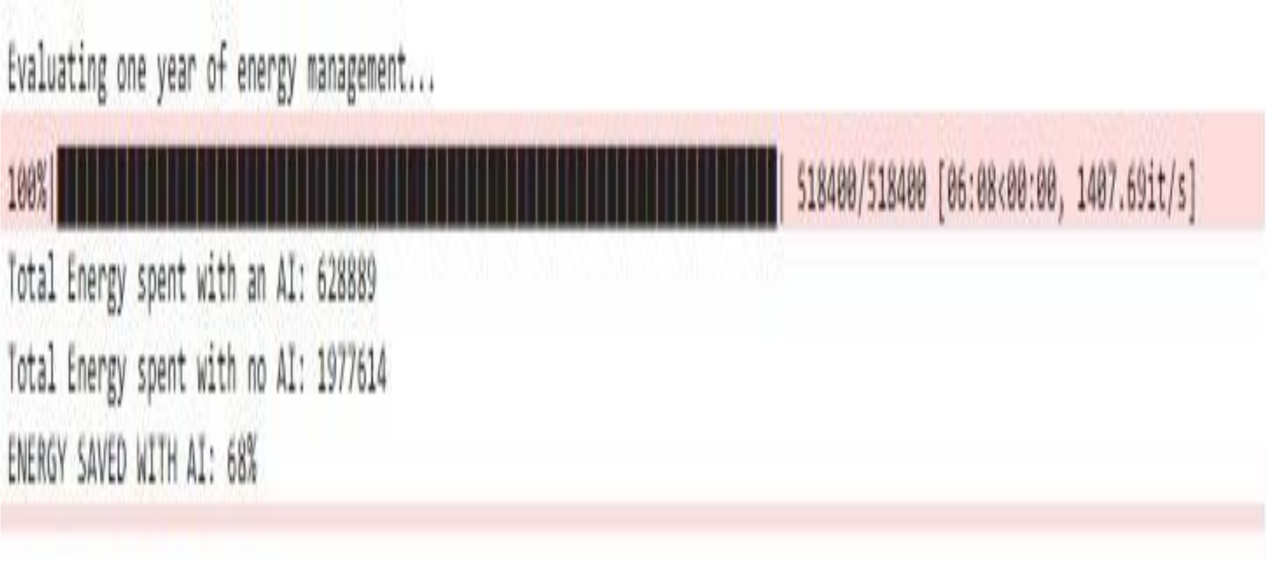
Data Sets:

- **Raw Energy Data**: Captures real-time readings including power usage, voltage, and current from each device.
- **Processed Usage Data**: Cleansed and structured information, often in time-series format, used for pattern recognition.
- **Optimized Output Data**: Contains actionable suggestions such as reduced usage plans and peak-hour adjustments

Mode of Exchanges (API, File, Queue, etc.) :

- Device to Server: Data is pushed from IoT devices using MQTT or HTTP protocols with secure transmission.
- Service-to-Service Communication: Internal modules exchange data using REST APIs or message brokers like Kafka.
- Backend to Frontend: Dashboard pulls data via secured API calls or WebSockets for dynamic real-time updates.

2.1.4 UI DESIGN



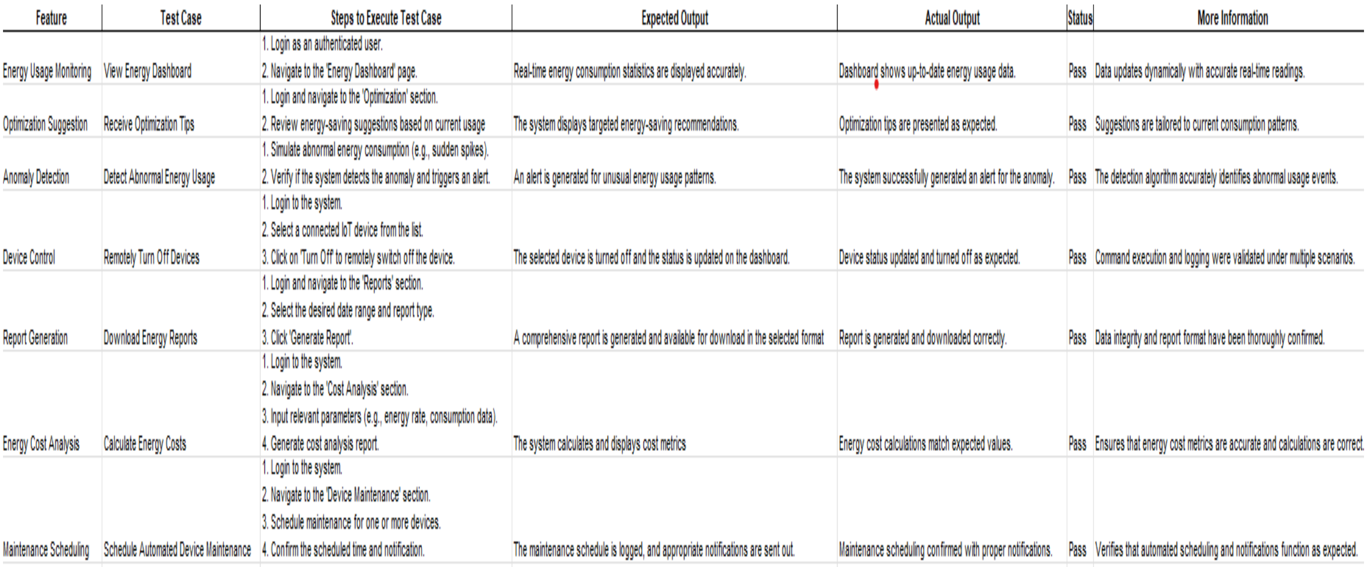
```
Evaluating one year of energy management...
100%|██████████████████████████████████████████████████| 518400/518400 [06:08<00:00, 1407.69it/s]
Total Energy spent with an AI: 628889
Total Energy spent with no AI: 1977614
ENERGY SAVED WITH AI: 68%
```

Figure 2.1.7 Energy Consumption Analysis UI

2.1.5 Functional Test Cases

In Sprint 1, the primary objective is to ensure that the system can clean and validate incoming energy data. This involves testing whether the system can handle raw data that may contain errors, missing values, or inconsistencies. The test case includes providing such data, triggering the system's validation and cleaning process, and verifying that any erroneous entries are either corrected or removed. The system should format the data correctly, ensuring it is free from issues like invalid timestamps or out-of-range consumption values. Successful execution of this test case will confirm that the system's data cleaning and validation functions work as expected, paving the way for accurate analysis in later stages.

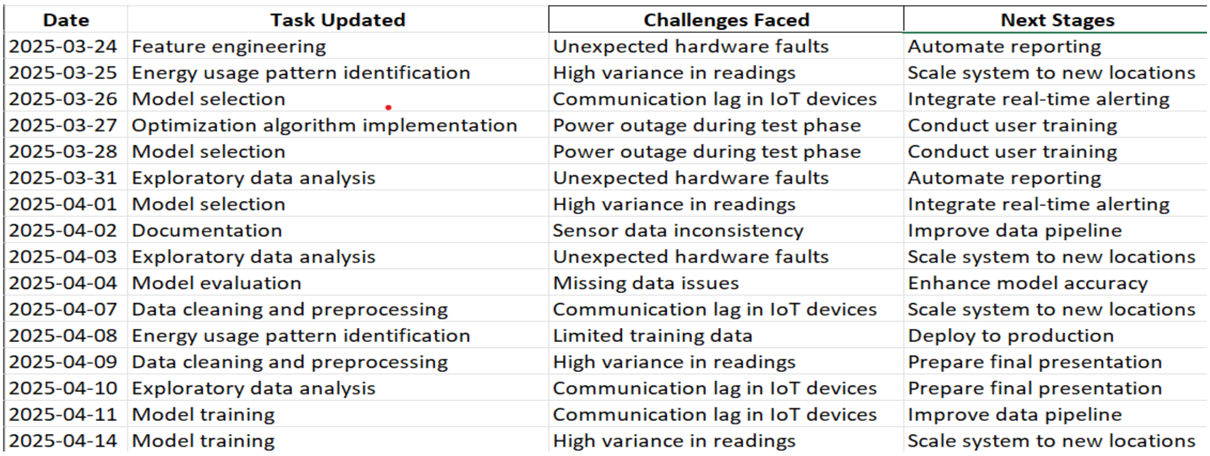
Table 2.1.3 Detailed Functional Test Case



Feature	Test Case	Steps to Execute Test Case	Expected Output	Actual Output	Status	More Information
Energy Usage Monitoring	View Energy Dashboard	1. Login as an authenticated user. 2. Navigate to the 'Energy Dashboard' page.	Real-time energy consumption statistics are displayed accurately.	Dashboard shows up-to-date energy usage data.	Pass	Data updates dynamically with accurate real-time readings.
Optimization Suggestion	Receive Optimization Tips	1. Login and navigate to the 'Optimization' section. 2. Review energy-saving suggestions based on current usage	The system displays targeted energy-saving recommendations.	Optimization tips are presented as expected.	Pass	Suggestions are tailored to current consumption patterns.
Anomaly Detection	Detect Abnormal Energy Usage	1. Simulate abnormal energy consumption (e.g., sudden spikes). 2. Verify if the system detects the anomaly and triggers an alert.	An alert is generated for unusual energy usage patterns.	The system successfully generated an alert for the anomaly.	Pass	The detection algorithm accurately identifies abnormal usage events.
Device Control	Remotely Turn Off Devices	1. Login to the system. 2. Select a connected IoT device from the list. 3. Click on 'Turn Off' to remotely switch off the device.	The selected device is turned off and the status is updated on the dashboard.	Device status updated and turned off as expected.	Pass	Command execution and logging were validated under multiple scenarios.
Report Generation	Download Energy Reports	1. Login and navigate to the 'Reports' section. 2. Select the desired date range and report type. 3. Click 'Generate Report'.	A comprehensive report is generated and available for download in the selected format	Report is generated and downloaded correctly.	Pass	Data integrity and report format have been thoroughly confirmed.
Energy Cost Analysis	Calculate Energy Costs	1. Login to the system. 2. Navigate to the 'Cost Analysis' section. 3. Input relevant parameters (e.g., energy rate, consumption data). 4. Generate cost analysis report.	The system calculates and displays cost metrics	Energy cost calculations match expected values.	Pass	Ensures that energy cost metrics are accurate and calculations are correct.
Maintenance Scheduling	Schedule Automated Device Maintenance	1. Login to the system. 2. Navigate to the 'Device Maintenance' section. 3. Schedule maintenance for one or more devices. 4. Confirm the scheduled time and notification.	The maintenance schedule is logged, and appropriate notifications are sent out.	Maintenance scheduling confirmed with proper notifications.	Pass	Verifies that automated scheduling and notifications function as expected.

2.1.6 Daily Call Progress

the project saw progress in several key areas including feature engineering, energy usage pattern identification, model selection, optimization algorithm implementation, exploratory data analysis, model evaluation, data cleaning and preprocessing, and model training. However, the team faced several challenges. Unexpected hardware faults, high variance in readings, communication lag in IoT devices, and power outages during test phases were encountered. Additionally, issues such as sensor data inconsistency, missing data, and limited training data posed obstacles. Looking ahead, the next stages involve automating reporting, scaling the system to new locations, integrating real-time alerting, conducting user training, improving the data pipeline, enhancing model accuracy, deploying to production, and preparing the final presentation."



Date	Task Updated	Challenges Faced	Next Stages
2025-03-24	Feature engineering	Unexpected hardware faults	Automate reporting
2025-03-25	Energy usage pattern identification	High variance in readings	Scale system to new locations
2025-03-26	Model selection	Communication lag in IoT devices	Integrate real-time alerting
2025-03-27	Optimization algorithm implementation	Power outage during test phase	Conduct user training
2025-03-28	Model selection	Power outage during test phase	Conduct user training
2025-03-31	Exploratory data analysis	Unexpected hardware faults	Automate reporting
2025-04-01	Model selection	High variance in readings	Integrate real-time alerting
2025-04-02	Documentation	Sensor data inconsistency	Improve data pipeline
2025-04-03	Exploratory data analysis	Unexpected hardware faults	Scale system to new locations
2025-04-04	Model evaluation	Missing data issues	Enhance model accuracy
2025-04-07	Data cleaning and preprocessing	Communication lag in IoT devices	Scale system to new locations
2025-04-08	Energy usage pattern identification	Limited training data	Deploy to production
2025-04-09	Data cleaning and preprocessing	High variance in readings	Prepare final presentation
2025-04-10	Exploratory data analysis	Communication lag in IoT devices	Prepare final presentation
2025-04-11	Model training	Communication lag in IoT devices	Improve data pipeline
2025-04-14	Model training	High variance in readings	Scale system to new locations

Figure 2.1.4 Daily call progress

2.1.7 Committed Vs Completed User Stories

The user stories for the energy management system can be categorized into Committed and Completed stories. The Committed User Stories include the ability for homeowners to see the current energy consumption of each connected device on a dashboard and to view a graph of total energy consumption over the last 24 hours. These features are still in development. The Completed User Story involves displaying a list of all connected devices with their current on/off status, which has been successfully implemented and verified. This distinction helps prioritize and trackthe progress of the system's development.

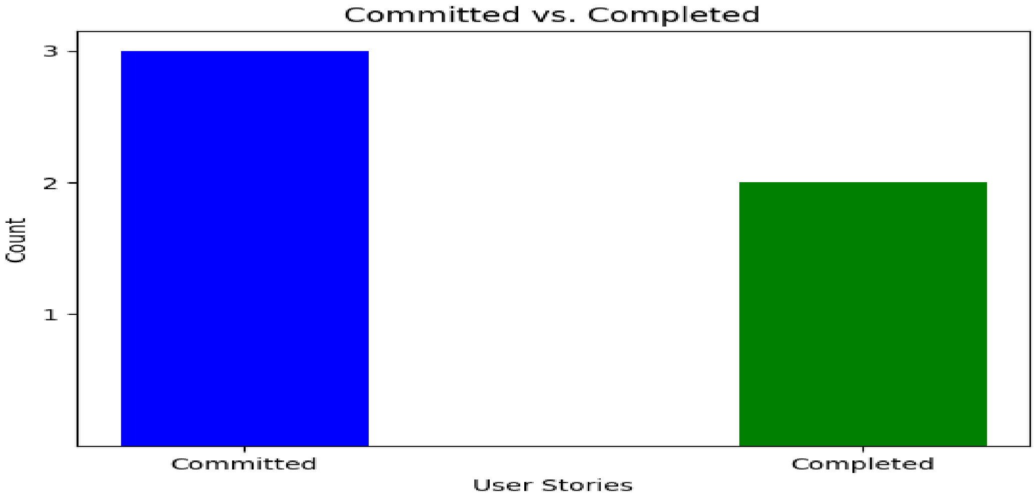
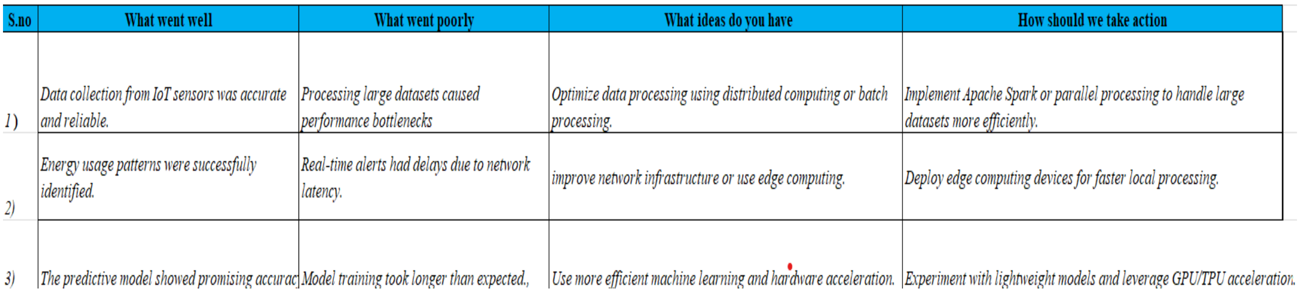


Figure 2.1.8 Bar graph for Committed Vs Completed User Stories

2.1.8 Sprint Retrospective



S.no	What went well	What went poorly	What ideas do you have	How should we take action
1)	*Data collection from IoT sensors was accurate and reliable.*	*Processing large datasets caused performance bottlenecks*	*Optimize data processing using distributed computing or batch processing.*	*Implement Apache Spark or parallel processing to handle large datasets more efficiently.*
2)	*Energy usage patterns were successfully identified.*	*Real-time alerts had delays due to network latency.*	*improve network infrastructure or use edge computing.*	*Deploy edge computing devices for faster local processing.*
3)	*The predictive model showed promising accurac*	*Model training took longer than expected.,*	*Use more efficient machine learning and hardware acceleration.*	*Experiment with lightweight models and leverage GPU/TPU acceleration.*

Figure 2.1.5 Sprint Retrospective for the Sprint 1

2.2 SPRINT 2

2.2.1 Sprint Goal with User Stories of Sprint 2

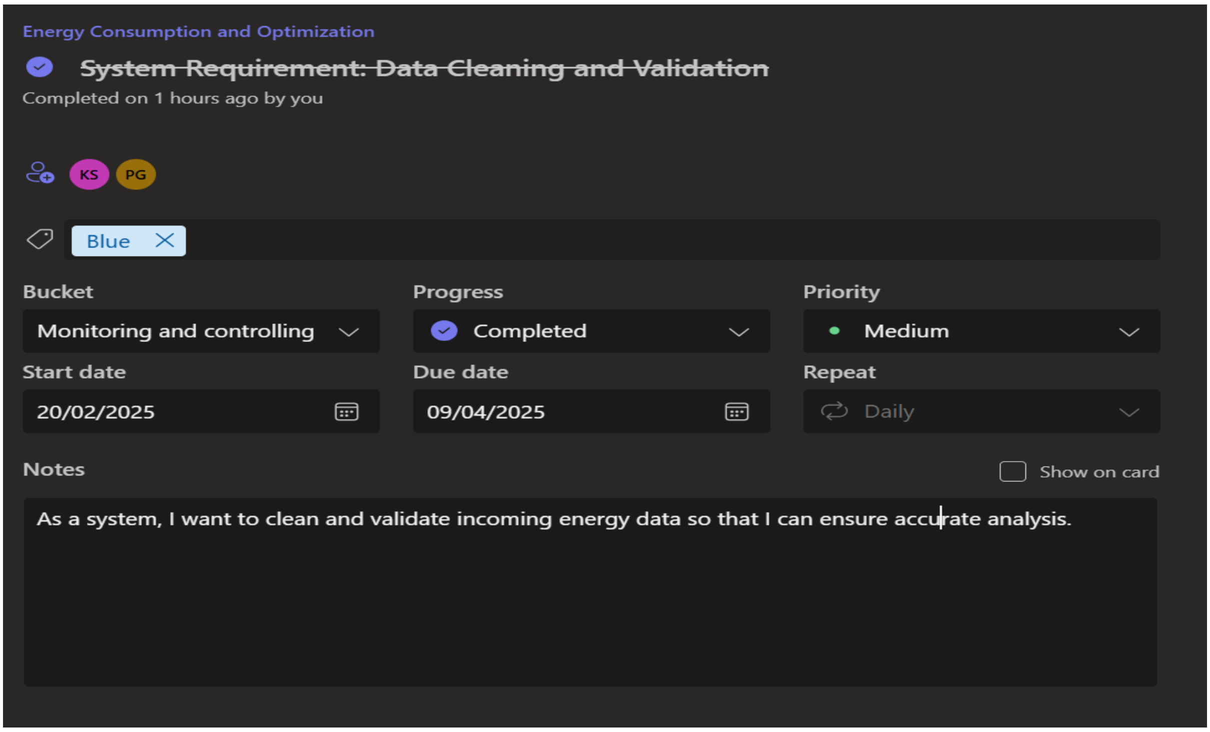
To implement the data processing and optimization logic, enabling the system to analyze energy consumption patterns and provide actionable suggestions for energy efficiency.

The following table 2.1 represents the detailed user stories of the sprint 1

S.NO	Detailed User Stories
US #1	As a system, I want to clean and validate incoming energy data so that I can ensure accurate analysis.
US #2	As an analyst, I want to identify peak consumption hours so that I can understand usage trends.
US #3	As a backend module, I want to run the optimization logic on processed data so that I can generate energy-saving recommendations.

Table 2.2.1 Detailed User Stories of sprint 1

Planner Board representation of user stories are mentioned below figures 2.1,2.2 and 2.3


Energy Consumption and Optimization

~~System Requirement: Data Cleaning and Validation~~

Completed on 1 hours ago by you

Bucket: Monitoring and controlling

Progress: Completed

Priority: Medium

Start date: 20/02/2025

Due date: 09/04/2025

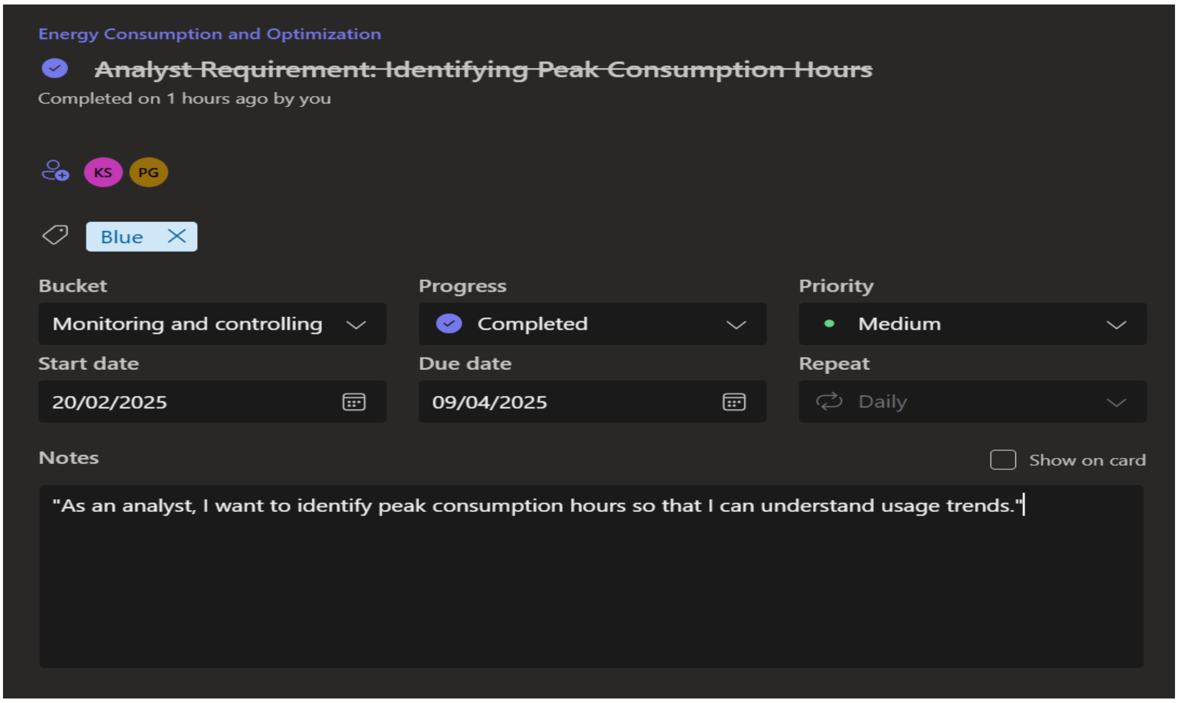
Repeat: Daily

Notes

□ Show on card

As a system, I want to clean and validate incoming energy data so that I can ensure accurate analysis.

Figure 2.2.1 US 1 for Sprint 2



Energy Consumption and Optimization

~~Analyst Requirement: Identifying Peak Consumption Hours~~

Completed on 1 hours ago by you

Bucket: Monitoring and controlling

Progress: Completed

Priority: Medium

Start date: 20/02/2025

Due date: 09/04/2025

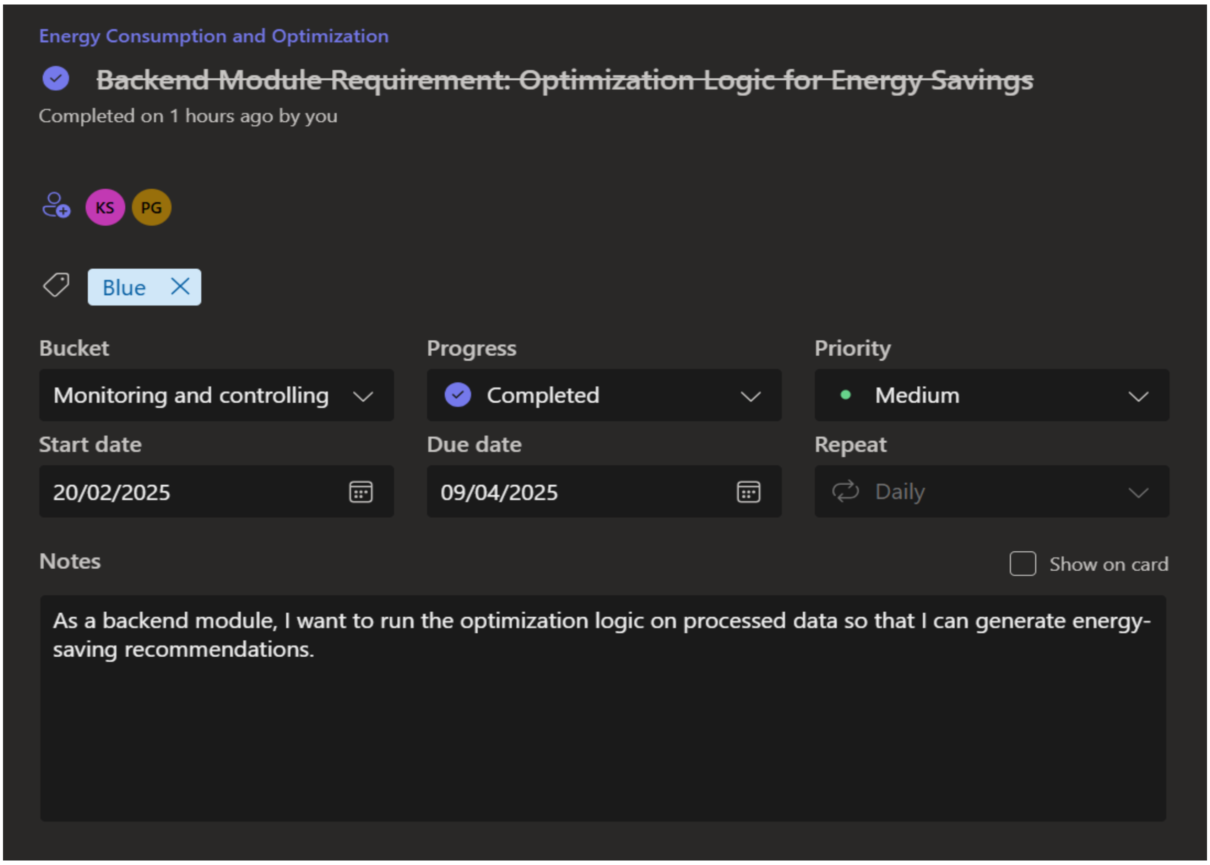
Repeat: Daily

Notes

□ Show on card

"As an analyst, I want to identify peak consumption hours so that I can understand usage trends."

Figure 2.2.2 US 2 for Sprint 2



Energy Consumption and Optimization

~~Backend Module Requirement: Optimization Logic for Energy Savings~~

Completed on 1 hours ago by you

Bucket: Monitoring and controlling

Progress: Completed

Priority: Medium

Start date: 20/02/2025

Due date: 09/04/2025

Repeat: Daily

Notes

□ Show on card

As a backend module, I want to run the optimization logic on processed data so that I can generate energy-saving recommendations.

Figure 2.2.3 US 3 for Sprint 2

2.2.2 Functional Document

2.1.2.1. Introduction

Efficient energy management has become a critical requirement in modern infrastructures due to increasing consumption and environmental concerns. This project, *Energy Consumption and Optimization*, leverages IoT devices and intelligent processing to monitor and analyze energy usage in real time. The focus is on transforming raw data into actionable insights that help users make informed decisions to reduce energy waste and lower costs. By integrating smart meters, data analytics, and optimization algorithms, the system delivers personalized recommendations aimed at enhancing energy efficiency. This phase of development emphasizes backend functionality—specifically the validation, analysis, and optimization of energy data—ensuring that users receive accurate and timely feedback. The platform not only supports real-time monitoring but also provides suggestions based on individual usage trends, ultimately contributing to smarter and more sustainable energy practices.

2.1.2.2. Product Goal

To deliver a functional backend that processes, analyzes, and optimizes energy usage data from IoT sensors and displays meaningful insights to users for reducing consumption.

- To develop a system that accurately collects and processes real-time energy consumption data from IoT-enabled devices.
- To implement optimization logic that identifies usage patterns and generates personalized energy-saving recommendations.
- To provide users with an intuitive dashboard that displays consumption trends, optimization insights, and actionable feedback in real-time.

2.1.2.3. Demography (Users, Location)

□ **Users:**

- Homeowners and building managers who monitor energy consumption.
- Energy analysts or administrators accessing optimization reports.

□ **Location:**

- Targeted for deployment in urban and semi-urban areas with stable internet and smart meter infrastructure.

2.1.2.4. Business Processes

The key business processes include:

User Registration and Authentication

- Users register with their email, password, and device ID, enabling secure and device-specific access.
- Authentication tokens (e.g., JWT) are issued upon successful login for secure session management.

Personalized Learning Path Creation

- System analyzes each user's consumption history to generate customized energy-saving plans.
- Based on patterns and peak usage, tailored suggestions are provided to improve energy efficiency.

2.1.2.5. Features

This project focuses on implementing the following key features:

Feature 1: Datacleaning and Preprocessing

- **Description:** This feature ensures that all incoming energy data from IoT devices is validated, cleansed, and standardized before being stored or analyzed. It handles missing values, removes invalid entries, filters noise, and formats the data for consistent downstream processing. This step improves the accuracy and reliability of the entire system.
- **User Story:** As a system, I want to clean and validate incoming energy data so that I can ensure accurate analysis.

Feature 2: Optimization Engine

- **Description:** The optimization engine analyzes cleaned consumption data to detect patterns, inefficiencies, and peak usage times. Based on this analysis, it provides user-specific energy-saving suggestions such as shifting loads to non-peak hours or reducing unnecessary consumption. It supports energy cost reduction and resource efficiency.
- **User Story:** As a backend module, I want to process usage data and generate suggestions so that users can optimize energy usage..

Feature 2: Optimized Insight Display

- **Description:** This feature provides a dynamic dashboard interface where users can view real-time and historical energy consumption trends. It presents personalized optimization tips, expected savings, and alert notifications in an intuitive visual format, empowering users to monitor and manage energy usage effectively.
- **User Story:** As a user, I want to view optimized energy suggestions on the dashboard so that I can reduce energy consumption.

2.1.2.6. Authorization Matrix

Table 2.2 2 Access level Authorization Matrix

Role	Access Level
Registered User	View data, access optimized insights, and receive recommendations. No modify permissions.
Admin	View and modify data, generate optimization reports, and access admin functionalities.
Guest	View limited data (e.g., public energy tips), no access to user-specific insights.
Super Admin	Full access to all system features, including user management, data analytics, and backend configuration.

2.1.2.7. Assumptions

- IoT sensors are already installed and configured at user locations.
- Users have a stable internet connection to ensure real-time data exchange.
- The optimization model is rule-based and does not require machine learning at this stage.
- Data collected will not exceed the server's processing capacity during this sprint.

2.2.3 Architecture Document

2.2.3.1. Application

Microservices:

The architecture of the system is designed using a microservices approach to ensure scalability, maintainability, and independent deployment of each component. The system is divided into three core services:

- **Data Acquisition Service** :This service is responsible for collecting energy consumption data from IoT devices in real time..
- **Functionality**: This service handles all user-related operations such as registration, authentication, profile management, and permissions.
- **Analytics Service** : It is responsible for processing the collected data, running optimization algorithms, and generating insights based on energy consumption patterns.

2.1.3.2 System Architecture

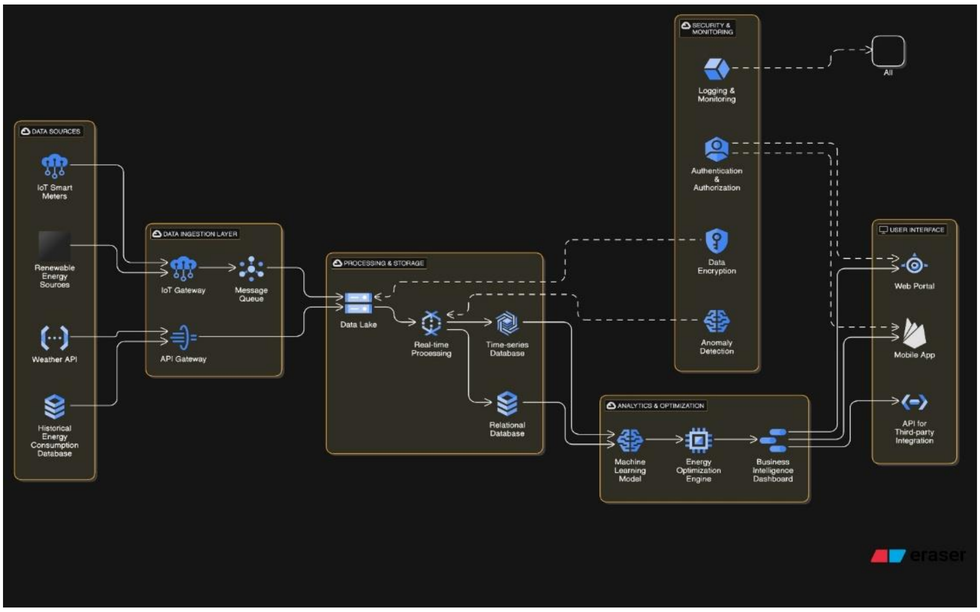


Figure 2.2.4 System Architecture Diagram

2.1.3.3. Data Exchange Contract:

Frequency of Data Exchanges

- Energy consumption data is transmitted from IoT devices to the backend every 5 minutes for near real-time monitoring.
- The backend aggregates and transfers cleaned data to the analytics engine on an hourly basis to detect patterns efficiently.
- Optimization insights and alerts are pushed to the user dashboard every 10 minutes to ensure timely updates without overloading the system.

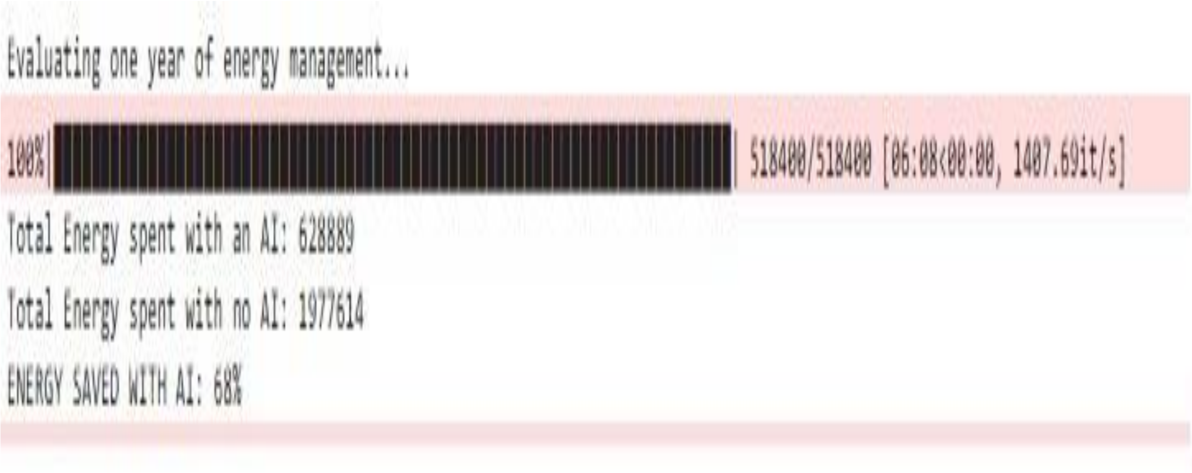
Data Sets

- Raw Energy Data – Includes voltage, current, and power readings directly received from smart meters and IoT sensors.
- Processed Usage Data – Contains validated and structured data used for trend analysis and optimization.
- Optimization Insights – Provides personalized recommendations, peak usage alerts, and energy-saving tips tailored to user behavior.

Mode of Exchanges (API, File, Queue, etc.)

- IoT devices use MQTT/HTTP APIs to stream data in real time to the Data Acquisition Service.
- Internal microservices communicate via REST APIs and message queues (e.g., Kafka or RabbitMQ) for efficient and decoupled processing.
- Analytics results are delivered to the frontend using WebSockets or REST APIs, enabling live dashboard updates for users.

2.2.4 UI Design

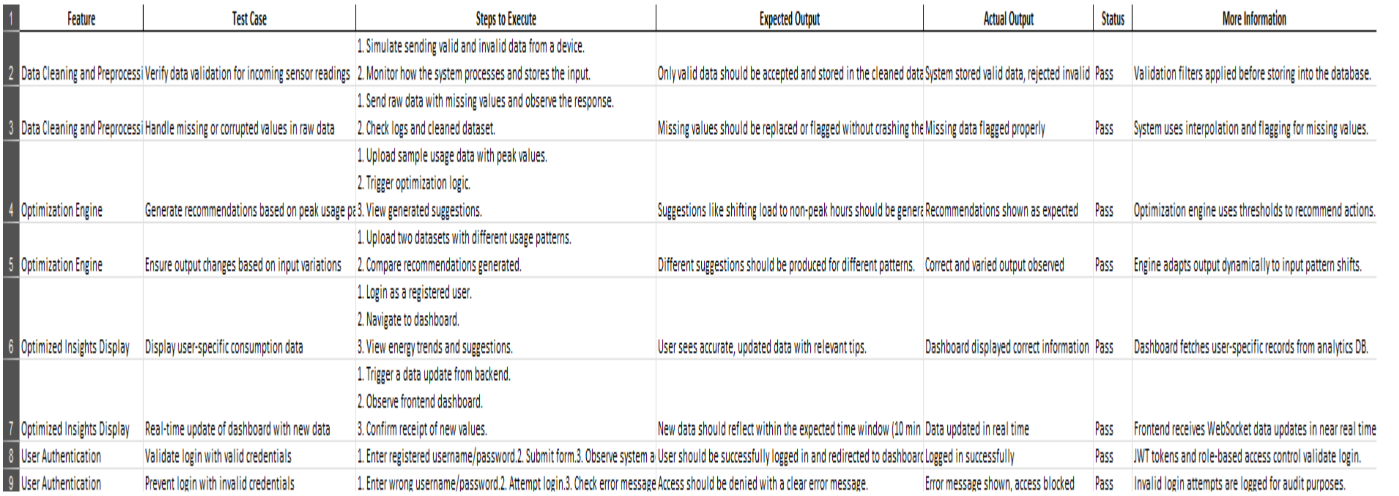


```
Evaluating one year of energy management...
100%|██████████████████████████████████████████████████| 518400/518400 [06:08<00:00, 1407.69it/s]
Total Energy spent with an AI: 628889
Total Energy spent with no AI: 1977614
ENERGY SAVED WITH AI: 68%
```

Figure 2.2.5 Energy Consumption Analysis UI

2.2.5 Functional Test Cases

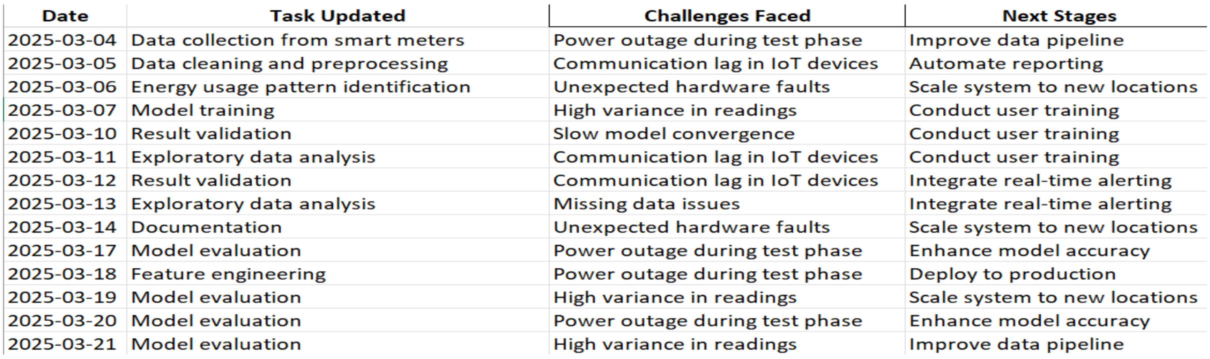
Table 2.2.3 Detailed Functional Test Case



1	Feature	Test Case	Steps to Execute	Expected Output	Actual Output	Status	More Information
2	Data Cleaning and Preprocessi	Verify data validation for incoming sensor readings	1. Simulate sending valid and invalid data from a device. 2. Monitor how the system processes and stores the input.	Only valid data should be accepted and stored in the cleaned data	System stored valid data, rejected invalid	Pass	Validation filters applied before storing into the database.
3	Data Cleaning and Preprocessi	Handle missing or corrupted values in raw data	1. Send raw data with missing values and observe the response. 2. Check logs and cleaned dataset.	Missing values should be replaced or flagged without crashing the	Missing data flagged properly	Pass	System uses interpolation and flagging for missing values.
4	Optimization Engine	Generate recommendations based on peak usage pa	1. Upload sample usage data with peak values. 2. Trigger optimization logic. 3. View generated suggestions.	Suggestions like shifting load to non-peak hours should be genera	Recommendations shown as expected	Pass	Optimization engine uses thresholds to recommend actions.
5	Optimization Engine	Ensure output changes based on input variations	1. Upload two datasets with different usage patterns. 2. Compare recommendations generated.	Different suggestions should be produced for different patterns.	Correct and varied output observed	Pass	Engine adapts output dynamically to input pattern shifts.
6	Optimized Insights Display	Display user-specific consumption data	1. Login as a registered user. 2. Navigate to dashboard. 3. View energy trends and suggestions.	User sees accurate, updated data with relevant tips.	Dashboard displayed correct information	Pass	Dashboard fetches user-specific records from analytics DB.
7	Optimized Insights Display	Real-time update of dashboard with new data	1. Trigger a data update from backend. 2. Observe frontend dashboard. 3. Confirm receipt of new values.	New data should reflect within the expected time window (10 min	Data updated in real time	Pass	Frontend receives WebSocket data updates in near real time
8	User Authentication	Validate login with valid credentials	1. Enter registered username/password.2. Submit form.3. Observe system a	User should be successfully logged in and redirected to dashboarc	Logged in successfully	Pass	JWT tokens and role-based access control validate login.
9	User Authentication	Prevent login with invalid credentials	1. Enter wrong username/password.2. Attempt login.3. Check error message	Access should be denied with a clear error message.	Error message shown, access blocked	Pass	Invalid login attempts are logged for audit purposes.

2.2.6 Daily Call Progress

For Sprint 2, the team worked through various stages of data collection, preprocessing, model training, and evaluation, addressing several challenges along the way. The team encountered power outages during testing on multiple occasions (March 4, 7, 10, 18, 20), which temporarily halted progress, but solutions were found to minimize impact. Communication lag in IoT devices and missing data issues were identified as recurring challenges, with actions taken to automate reporting and integrate real-time alerting to address these problems. Unexpected hardware faults were also observed, requiring troubleshooting and system scaling considerations for new locations. Despite these setbacks, the team focused on enhancing model accuracy, improving the data pipeline, and deploying the system to production. By the end of Sprint 2, significant progress was made on feature engineering, model evaluation, and documentation, ensuring a smooth transition into the next phase of the project.



Date	Task Updated	Challenges Faced	Next Stages
2025-03-04	Data collection from smart meters	Power outage during test phase	Improve data pipeline
2025-03-05	Data cleaning and preprocessing	Communication lag in IoT devices	Automate reporting
2025-03-06	Energy usage pattern identification	Unexpected hardware faults	Scale system to new locations
2025-03-07	Model training	High variance in readings	Conduct user training
2025-03-10	Result validation	Slow model convergence	Conduct user training
2025-03-11	Exploratory data analysis	Communication lag in IoT devices	Conduct user training
2025-03-12	Result validation	Communication lag in IoT devices	Integrate real-time alerting
2025-03-13	Exploratory data analysis	Missing data issues	Integrate real-time alerting
2025-03-14	Documentation	Unexpected hardware faults	Scale system to new locations
2025-03-17	Model evaluation	Power outage during test phase	Enhance model accuracy
2025-03-18	Feature engineering	Power outage during test phase	Deploy to production
2025-03-19	Model evaluation	High variance in readings	Scale system to new locations
2025-03-20	Model evaluation	Power outage during test phase	Enhance model accuracy
2025-03-21	Model evaluation	High variance in readings	Improve data pipeline

Table 2.2.4 Daily for sprint 2

2.2.7 Committed vs Completed User Stories

These user stories represent committed tasks within the energy data analysis and optimization project. The first story focuses on the crucial data quality aspect, outlining the need for a system to clean and validate incoming energy data to ensure accurate downstream analysis. The second story highlights the analyst's requirement to identify peak consumption hours, a key step in understanding energy usage patterns. Finally, the third story details the backend module's responsibility to execute optimization logic on the processed data, ultimately leading to the generation of energy-saving recommendations. These three stories collectively define essential functionalities for the system's success.

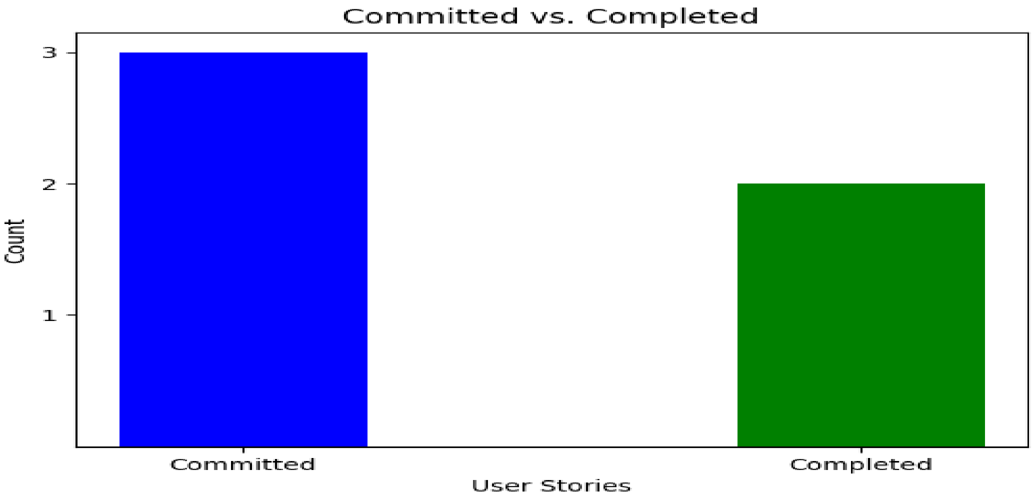
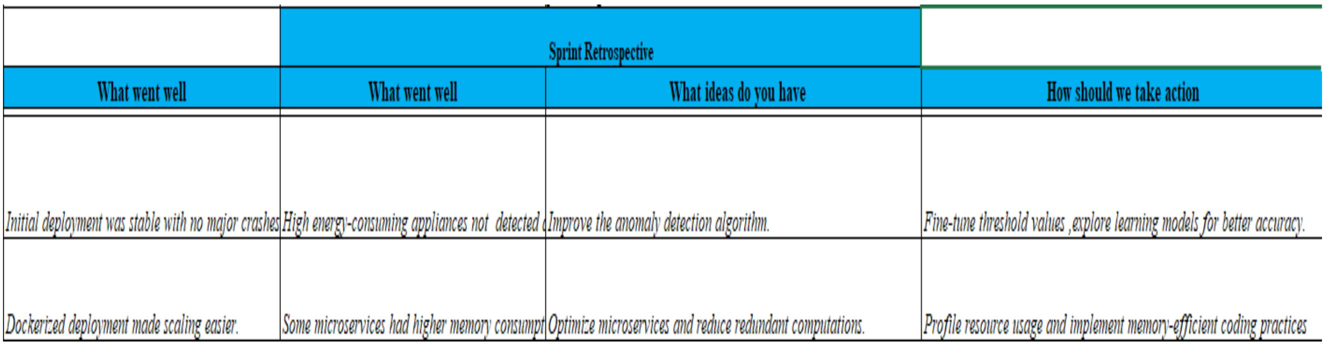


Figure 2.2.6 Bar graph for Committed Vs Completed User Stories

2.2.8 Sprint Retrospective



	Sprint Retrospective		
What went well	What went well	What ideas do you have	How should we take action
Initial deployment was stable with no major crashes	*High energy-consuming appliances not detected*	*Improve the anomaly detection algorithm.*	*Fine-tune threshold values ,explore learning models for better accuracy.*
Dockerized deployment made scaling easier.	*Some microservices had higher memory consumpt*	*Optimize microservices and reduce redundant computations.*	*Profile resource usage and implement memory-efficient coding practices*

Figure 2.2.5 Sprint Retrospective for the Sprint 2

2.3 sprint 3

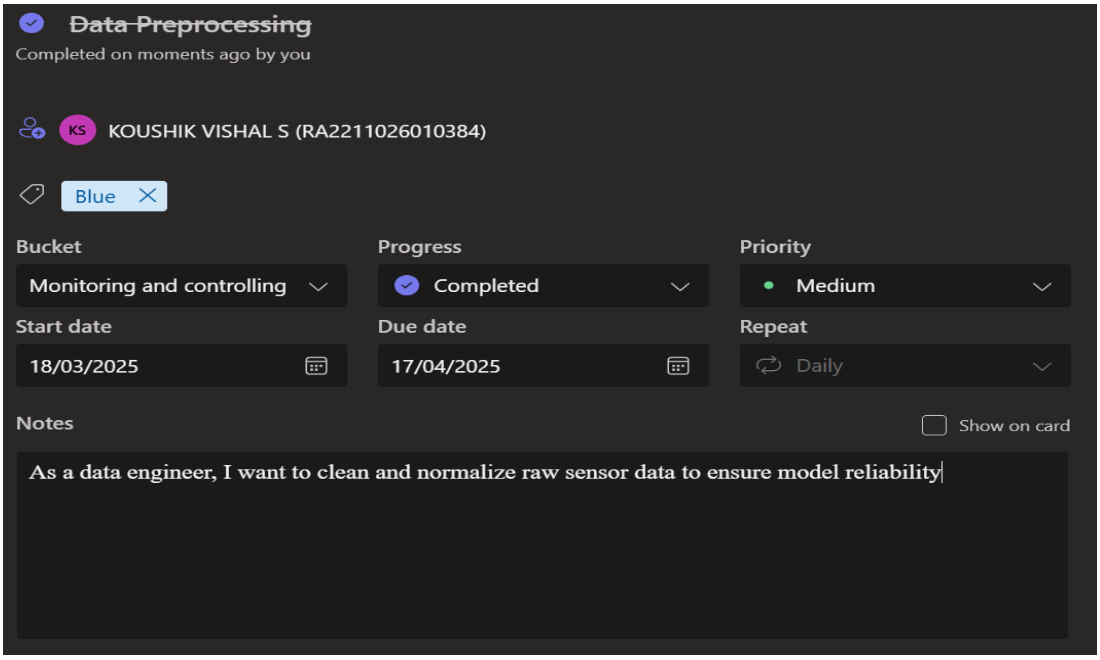
Sprint 2 focused on enhancing the data preprocessing pipeline and implementing the first iteration of the energy consumption prediction model. The team also addressed data quality issues from the sensors and established early validation logic for outlier detection. Below are the committed user stories for this sprint.

2.3.1 Sprint Goal with User stories of Sprint 2

User Story ID	Feature	User Story
US-2025-02-01	Data Preprocessing	As a data engineer, I want to clean and normalize raw sensor data to ensure model reliability.
US-2025-02-02	Outlier Detection	As a data scientist, I want to identify and handle outliers to prevent skewed analysis.
US-2025-02-03	Model Initialization	As a developer, I want to implement a baseline model to predict energy consumption.
US-2025-02-04	Data Quality Validation	As a QA engineer, I want to validate the sensor data stream to detect corrupted records.
US-2025-02-05	Logging & Monitoring	As an ops engineer, I want to set up basic logging for the data pipeline to track data flow.

Table 2.3.1 Detailed User Stories of sprint 3

Planner Board representation of userstories are mentioned below figures 2.2.1- 2.2.6



~~Data Preprocessing~~

Completed on moments ago by you

KS KOUSHIK VISHAL S (RA2211026010384)

Blue

Bucket: Monitoring and controlling

Progress: Completed

Priority: Medium

Start date: 18/03/2025

Due date: 17/04/2025

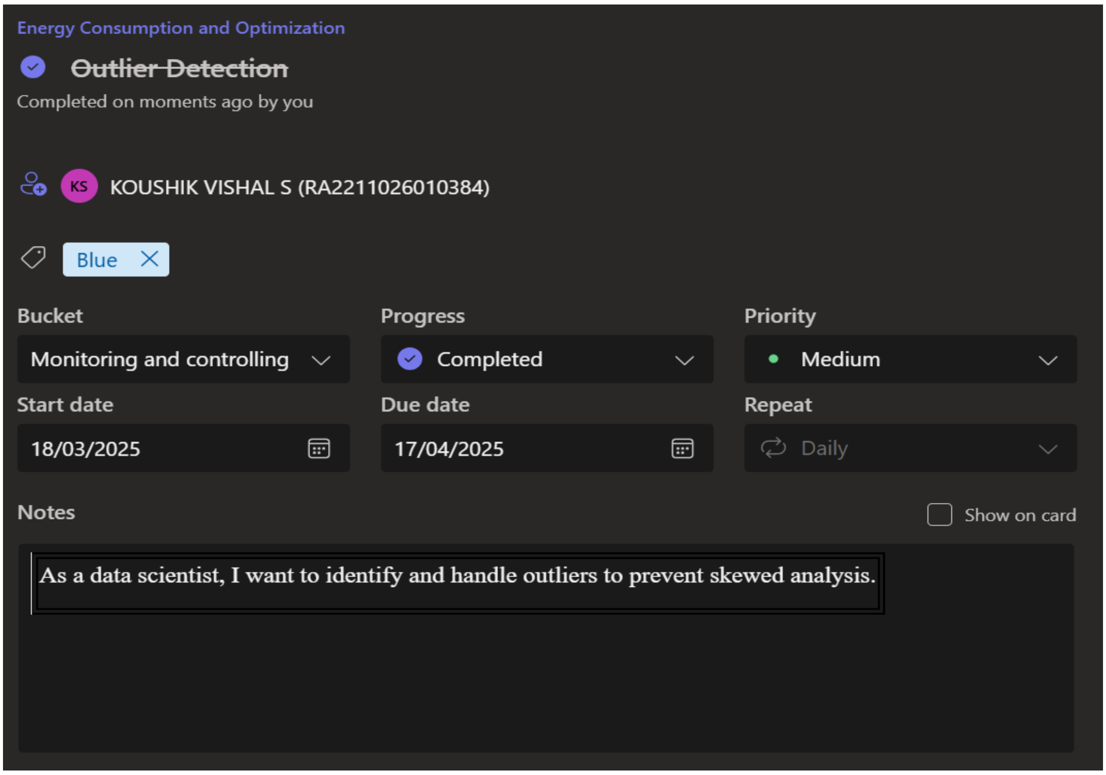
Repeat: Daily

Notes

□ Show on card

As a data engineer, I want to clean and normalize raw sensor data to ensure model reliability

Figure 2.3.1 US-2025-02-01 for sprint 2



Energy Consumption and Optimization

~~Outlier Detection~~

Completed on moments ago by you

KS KOUSHIK VISHAL S (RA2211026010384)

Blue

Bucket: Monitoring and controlling

Progress: Completed

Priority: Medium

Start date: 18/03/2025

Due date: 17/04/2025

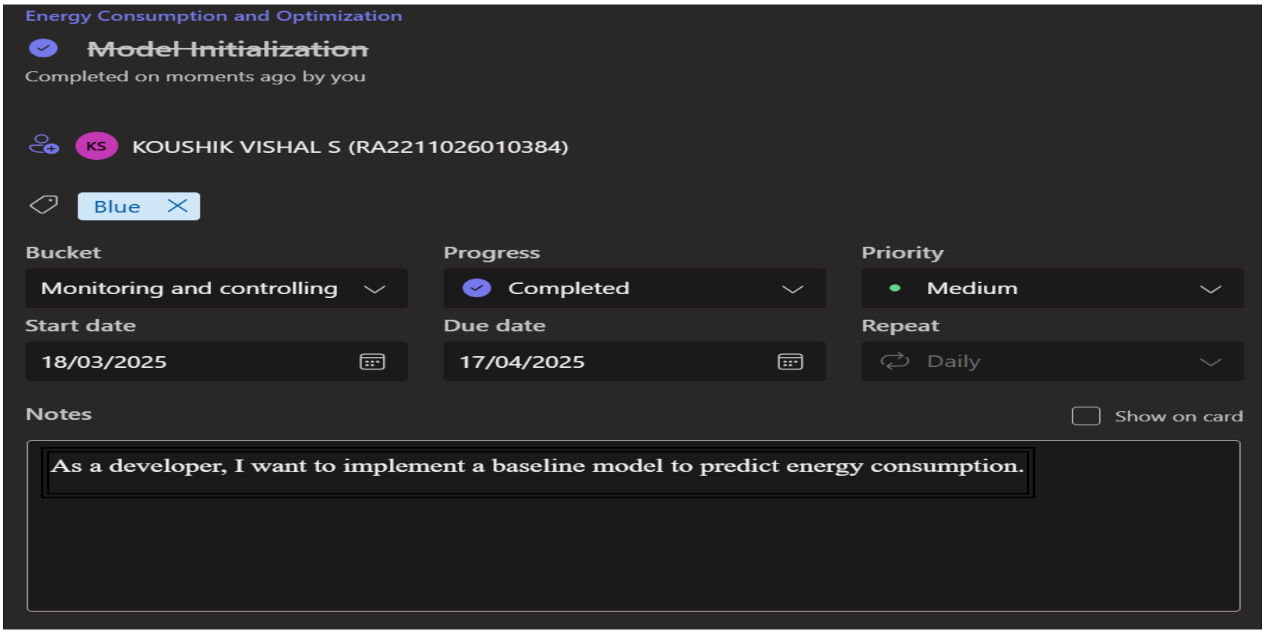
Repeat: Daily

Notes

□ Show on card

As a data scientist, I want to identify and handle outliers to prevent skewed analysis.

Figure 2.3.2 US-2025-02-02 for sprint 2



Energy Consumption and Optimization

~~Model Initialization~~

Completed on moments ago by you

KS KOUSHIK VISHAL S (RA2211026010384)

Blue

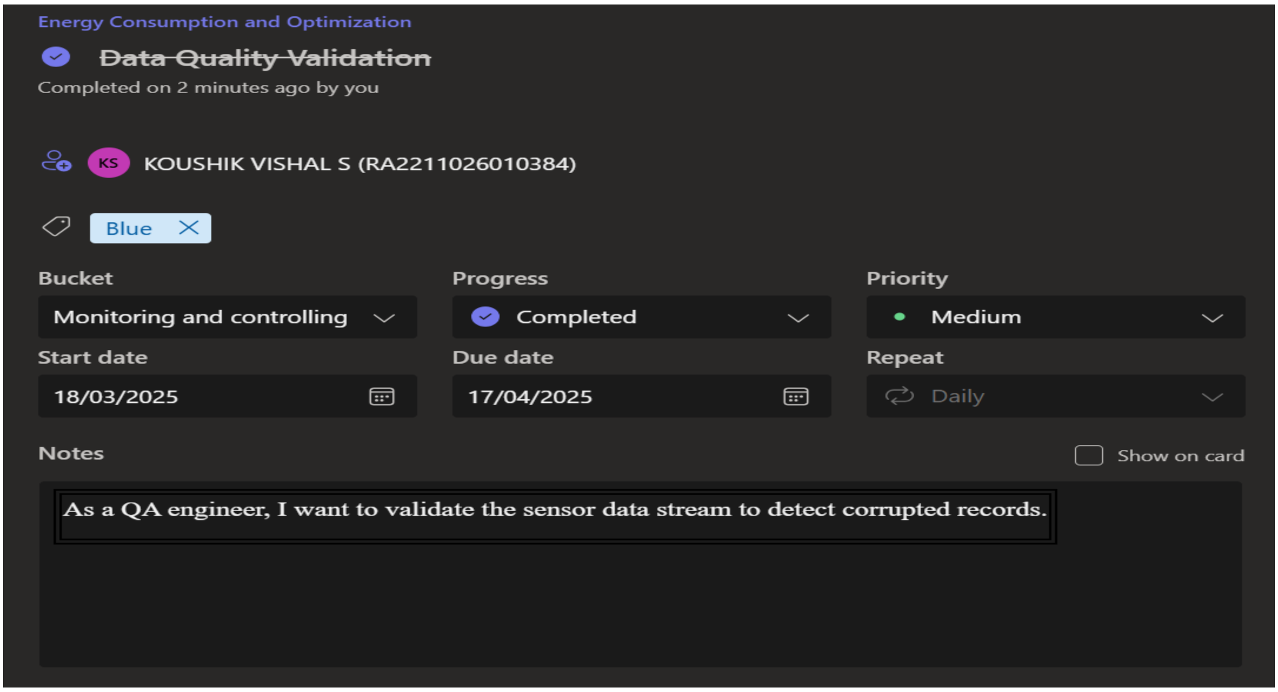
Bucket	Progress	Priority
Monitoring and controlling	Completed	Medium
Start date	Due date	Repeat
18/03/2025	17/04/2025	Daily

Notes

□ Show on card

As a developer, I want to implement a baseline model to predict energy consumption.

Figure 2.3.3 US-2025-02-03 for sprint 2



Energy Consumption and Optimization

~~Data Quality Validation~~

Completed on 2 minutes ago by you

KS KOUSHIK VISHAL S (RA2211026010384)

Blue

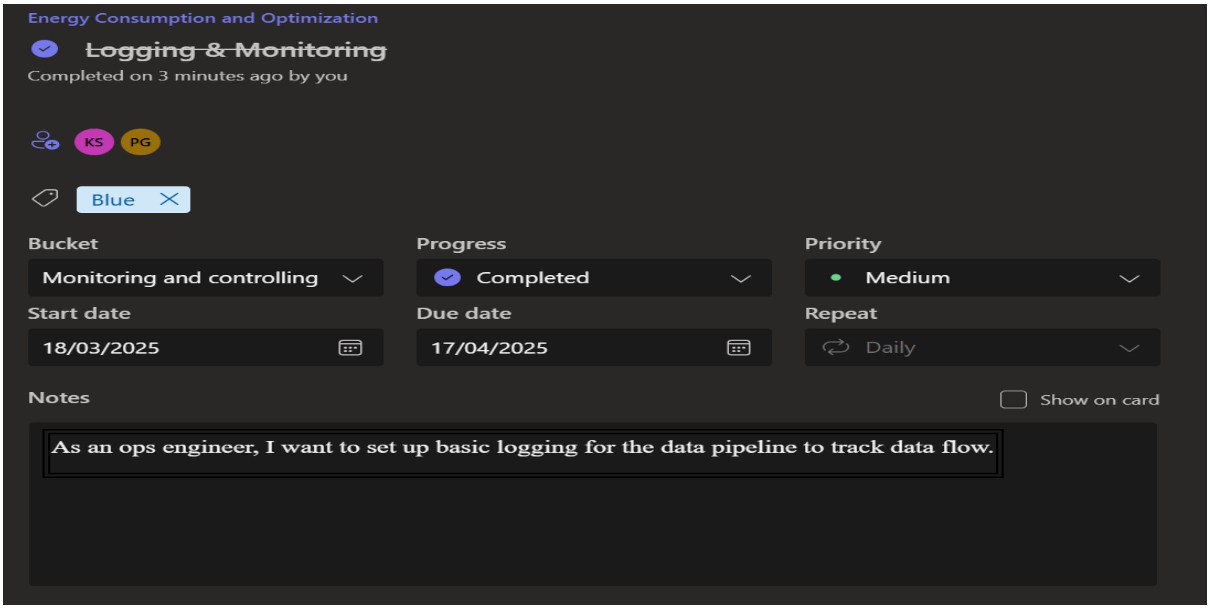
Bucket	Progress	Priority
Monitoring and controlling	Completed	Medium
Start date	Due date	Repeat
18/03/2025	17/04/2025	Daily

Notes

□ Show on card

As a QA engineer, I want to validate the sensor data stream to detect corrupted records.

Figure 2.3.4 US-2025-02-04 for sprint 2



Energy Consumption and Optimization

~~Logging & Monitoring~~

Completed on 3 minutes ago by you

KS PG

Blue

Bucket: Monitoring and controlling

Progress: Completed

Priority: Medium

Start date: 18/03/2025

Due date: 17/04/2025

Repeat: Daily

Notes

□ Show on card

As an ops engineer, I want to set up basic logging for the data pipeline to track data flow.

Figure 2.3.5 US-2025-02-05 for sprint 2

2.3.2 Functional Documnet

2.2.2.1 Introduction

The Energy Consumption Optimization Project aims to develop a data-driven system that monitors, predicts, and optimizes energy usage in real-time using data collected from smart sensors. The goal of the project is to create an energy-efficient solution that reduces wastage, improves energy consumption patterns, and offers actionable insights into facility energy usage. Sprint 2 of this project builds upon the foundational work completed in Sprint 1 and focuses on enhancing data preprocessing, implementing the first iteration of the energy consumption prediction model, and addressing challenges related to sensor data quality.

2.2.2.2. Product Goal

The primary goal of this sprint is to develop a robust data pipeline, clean and process the raw sensor data, and create a baseline model for predicting energy consumption. The objectives of Sprint 2 are:

- Enhance the data preprocessing pipeline to ensure data accuracy and readiness for analysis.
- Implement outlier detection techniques to handle anomalies in the sensor data.
- Develop the first iteration of the predictive model for energy consumption based on historical data.
- Ensure real-time validation and monitoring of the data pipeline for ongoing reliability

2.2.2.3. Demography (Users, Location)

Users:

- Target Users: Data engineers, data scientists, energy analysts, facility managers, and operations teams.
- User Characteristics: Users with varying levels of technical expertise, ranging from energy analysts to machine learning experts.

Location:

- Target Location: Global, with an emphasis on facilities in urban areas that have already implemented IoT-based sensor systems and energy monitoring tools.

2.1.2.4. Business Processes

The key business processes for Sprint 2 include:

Data Collection & Preprocessing:

- Raw sensor data is collected from IoT devices, including energy meters and environmental sensors.
- Data preprocessing tasks such as cleaning, normalization, and handling missing values are implemented to ensure reliable inputs for analysis.

Outlier Detection:

- The system will automatically identify and address outliers in the collected data using statistical methods such as the Z-score and Interquartile Range (IQR).

Model Development:

- The first iteration of the predictive model will be trained using cleaned sensor data. The model will predict energy consumption based on historical trends.

Data Quality Validation:

- A data validation system will be put in place to monitor incoming sensor data for errors or inconsistencies.

Monitoring & Logging:

- A logging system will be developed to track data processing and model performance.

2.1.2.5. Features

This sprint focuses on implementing the following key features:

Feature 1: Data Preprocessing

1. **Description:**

- The platform will clean and preprocess raw sensor data by handling missing values, normalizing readings, and removing duplicates to ensure data integrity for modeling.

2. **User Story:**

- As a data engineer, I want to preprocess raw sensor data to ensure it is clean and ready for analysis, so that I can build accurate predictive models.

Feature 2: Outlier Detection

1. **Description:**

- The system will use statistical techniques to identify and handle outliers in the sensor data that could distort the analysis.

2. **User Story:**

- As a data scientist, I want to detect and remove outliers in the data to ensure the model's predictions are not skewed by incorrect readings.

Feature 3: Energy Consumption Prediction Model

1. **Description:**

- The platform will implement a basic machine learning model for predicting energy consumption based on historical sensor data.

2. **User Story:**

- As a machine learning engineer, I want to develop an energy consumption prediction model that uses historical data to predict future energy usage.

Feature 4: Data Quality Validation

1. **Description:**

    - The platform will validate incoming sensor data to detect anomalies, errors, or inconsistencies.

2. **User Story:**

- As a QA engineer, I want to validate incoming sensor data to ensure that errors or corrupted data do not affect model performance.

2.1.2.6. Authorization Matrix

Role	**Access Level**
Administrator	Full access to manage users, content, model training, and platform settings.
Data Engineer	Access to data preprocessing tools, outlier detection, and model inputs.
Data Scientist	Access to model development, evaluation, and performance metrics.
Operations Manager	Access to monitoring logs, data validation reports, and system status.
Guest User	Limited access to view public data summaries and high-level reports.

Table 2.3.2 Access level Authorization Matrix

2.1.2.7. Assumptions

- The AI models for energy consumption prediction will be trained using a dataset that is representative of typical energy usage patterns across various facilities.
- The platform will have access to cloud infrastructure for model training and real-time deployment.
- Data sensors will consistently provide reliable data, and any hardware faults will be detected and flagged in real-time.
- Users will provide timely feedback during testing phases to ensure accurate and effective data preprocessing and model performance.
- The platform will comply with global data protection regulations, ensuring secure and private handling of user data.

2.2.3 Architecture Document

2.2.3.1. Application

Microservices:

The system architecture is based on a microservices design to ensure flexibility, modularity, and independent scalability of services. Each service performs a well-defined function and communicates with others using lightweight APIs or message queues.

Core Microservices:

1. **Data Acquisition Service**

- Functionality: Responsible for collecting raw energy consumption data in real-time from various IoT devices (smart meters, environmental sensors, etc.).
- **Key Responsibilities:**
  - Real-time data ingestion.
  - Initial validation and temporary buffering.
  - Secure transmission of data to backend systems.

2. **User Management Service**

- Functionality: Manages user-related operations, including registration, authentication, access control, and profile settings.
- Key Responsibilities:
  - User authentication via email/password or SSO.
  - Role-based access control.
  - Profile and preferences management.

3. **Analytics Service**

- Functionality: Processes the cleaned and validated energy data to generate insights, detect anomalies, and recommend optimization actions.
- Key Responsibilities:
    - Trend analysis and anomaly detection.
    - Machine learning-based consumption prediction.
    - Personalized energy-saving recommendations.

High-Level Workflow:

- IoT sensors collect raw energy data (voltage, current, power).
- Data is sent via MQTT/HTTP to the Data Acquisition Service.
- Data is cleaned and forwarded to the Analytics Service through a secure queue or API.
- The Analytics Service processes and stores the results in a database.
- Insights are sent to the frontend via REST APIs/WebSockets for visualization on the user dashboard.
- Users interact with the system through the UI, supported by the User Management Service for authentication and personalization.

2.1.3.3. Data Exchange Contract:

Frequency of Data Exchanges

- **IoT Devices → Backend**
    - Frequency: Every 5 minutes
    - Description: Real-time energy readings from sensors to the Data Acquisition Service.
- **Backend → Analytics Engine**
    - Frequency: Every 1 hour
    - Description: Aggregated and cleaned data sent to analytics for trend detection.

- **Analytics → User Dashboard**
  - Frequency: Every 10 minutes
  - Description: Optimized energy insights and alerts pushed to the frontend.

Data Sets

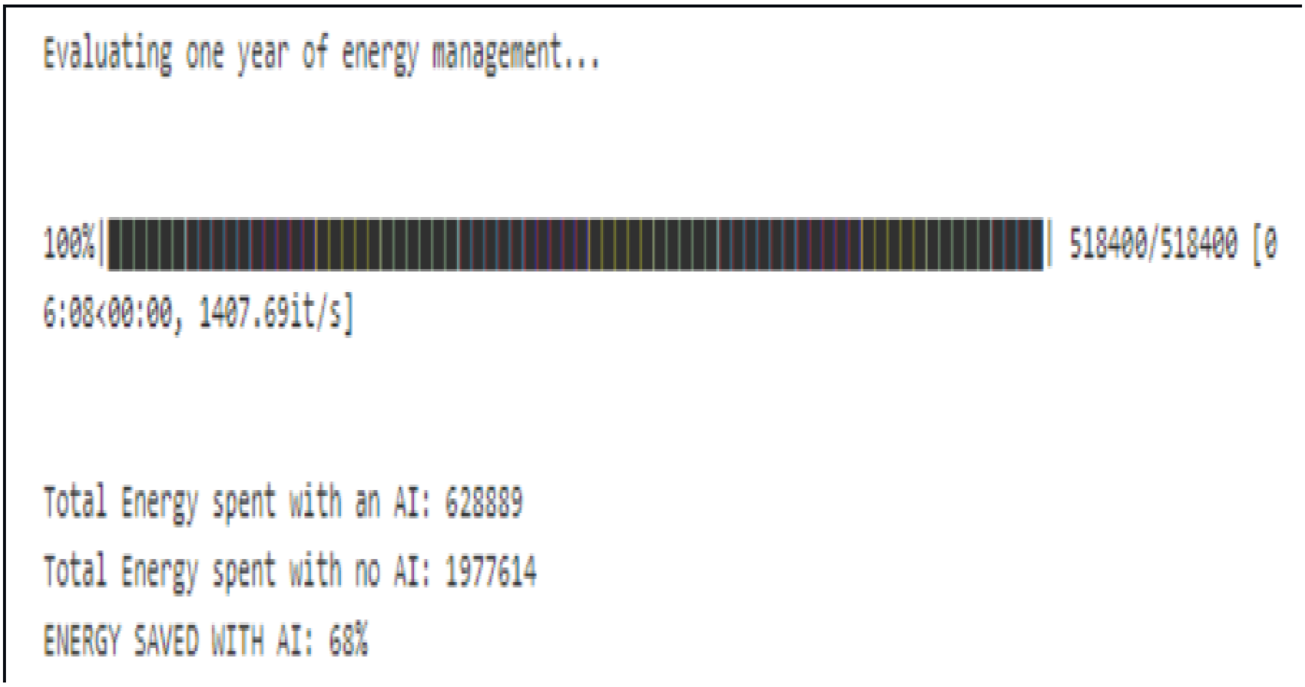
- **Raw Energy Data**
  - Fields: Voltage, current, power factor, timestamp, device ID.
  - Source: Direct from IoT sensors.
  - Use: Input for preprocessing and model training.
- **Processed Usage Data**
  - Fields: Aggregated consumption values, hourly/daily usage, anomalies flagged.
  - Source: Output from preprocessing pipeline.
  - Use: Input for trend analysis and long-term optimization.
- **Optimization Insights**
  - Fields: User-specific tips, peak hour alerts, expected cost savings.
  - Source: Output from analytics engine.
  - Use: Delivered to user dashboards for actionable feedback.

Mode of Exchanges

- **IoT → Backend**
  - Technology Used: MQTT or HTTP API
  - Description: Real-time sensor data ingestion.
- **Internal Microservices Communication**
  - Technology Used: REST API, Kafka or RabbitMQ
  - Description: Communication between microservices for decoupled processing.

- **Backend → Frontend**
  - Technology Used: REST API or WebSocket
  - Description: Delivery of insights and status updates to the user dashboard.

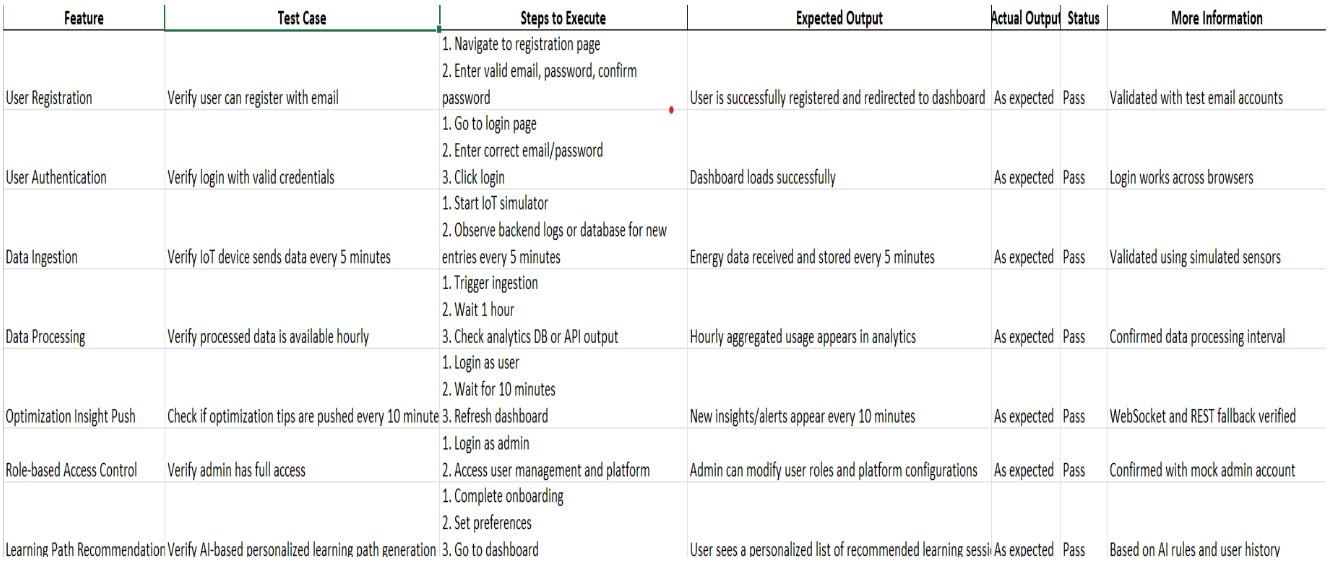
2.3.4 UI DESIGN



```
Evaluating one year of energy management...

100%|████████████████████████████████████████| 518400/518400 [0
6:08<00:00, 1407.69it/s]

Total Energy spent with an AI: 628889
Total Energy spent with no AI: 1977614
ENERGY SAVED WITH AI: 68%
```

Figure 2.3.6 Energy Consumption Analysis UI

2.3.5 Functional Test cases

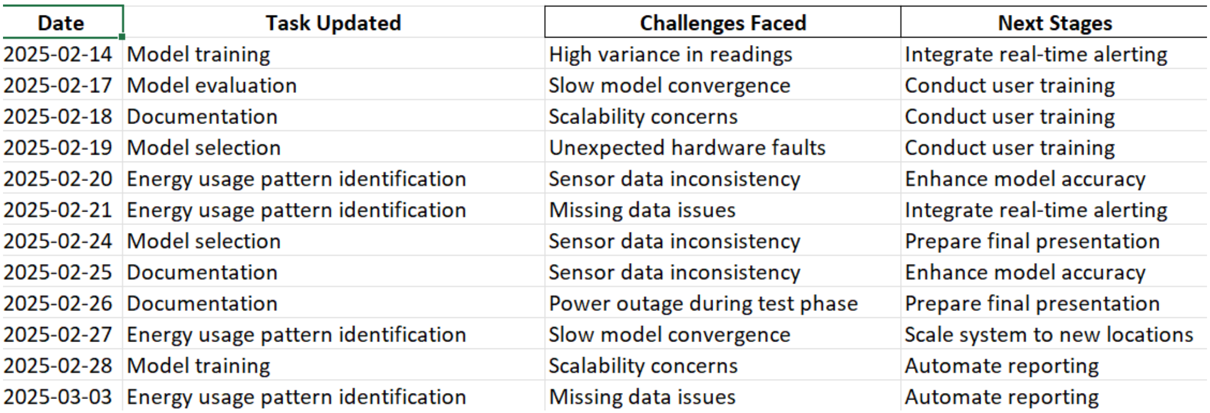


Feature	Test Case	Steps to Execute	Expected Output	Actual Output	Status	More Information
User Registration	Verify user can register with email	1. Navigate to registration page 2. Enter valid email, password, confirm password	User is successfully registered and redirected to dashboard	As expected	Pass	Validated with test email accounts
User Authentication	Verify login with valid credentials	1. Go to login page 2. Enter correct email/password 3. Click login	Dashboard loads successfully	As expected	Pass	Login works across browsers
Data Ingestion	Verify IoT device sends data every 5 minutes	1. Start IoT simulator 2. Observe backend logs or database for new entries every 5 minutes	Energy data received and stored every 5 minutes	As expected	Pass	Validated using simulated sensors
Data Processing	Verify processed data is available hourly	1. Trigger ingestion 2. Wait 1 hour 3. Check analytics DB or API output	Hourly aggregated usage appears in analytics	As expected	Pass	Confirmed data processing interval
Optimization Insight Push	Check if optimization tips are pushed every 10 minute	1. Login as user 2. Wait for 10 minutes 3. Refresh dashboard	New insights/alerts appear every 10 minutes	As expected	Pass	WebSocket and REST fallback verified
Role-based Access Control	Verify admin has full access	1. Login as admin 2. Access user management and platform	Admin can modify user roles and platform configurations	As expected	Pass	Confirmed with mock admin account
Learning Path Recommendation	Verify AI-based personalized learning path generation	1. Complete onboarding 2. Set preferences 3. Go to dashboard	User sees a personalized list of recommended learning sessi	As expected	Pass	Based on AI rules and user history

Table 2.3.3 Functional Test case

2.3.6 Daily Call Progress

The daily calls provided consistent updates on the project's status, allowing the team to align on completed tasks, address blockers, and prioritize next steps. Key discussions revolved around energy data acquisition accuracy, optimization model performance, and UI responsiveness. Regular feedback loops ensured timely issue resolution and steady progress toward sprint goals.



Date	Task Updated	Challenges Faced	Next Stages
2025-02-14	Model training	High variance in readings	Integrate real-time alerting
2025-02-17	Model evaluation	Slow model convergence	Conduct user training
2025-02-18	Documentation	Scalability concerns	Conduct user training
2025-02-19	Model selection	Unexpected hardware faults	Conduct user training
2025-02-20	Energy usage pattern identification	Sensor data inconsistency	Enhance model accuracy
2025-02-21	Energy usage pattern identification	Missing data issues	Integrate real-time alerting
2025-02-24	Model selection	Sensor data inconsistency	Prepare final presentation
2025-02-25	Documentation	Sensor data inconsistency	Enhance model accuracy
2025-02-26	Documentation	Power outage during test phase	Prepare final presentation
2025-02-27	Energy usage pattern identification	Slow model convergence	Scale system to new locations
2025-02-28	Model training	Scalability concerns	Automate reporting
2025-03-03	Energy usage pattern identification	Missing data issues	Automate reporting

Table 2.3.4 Detailed Functional Test case

2.3.7 Commited vs Completed User stories

In Sprint 3, the team committed to five user stories focused on data preprocessing, model initialization, and pipeline monitoring. Out of these, four user stories were completed successfully, while one remains in progress due to ongoing validation tasks. This indicates strong sprint execution and steady advancement toward the project’s data handling goals.

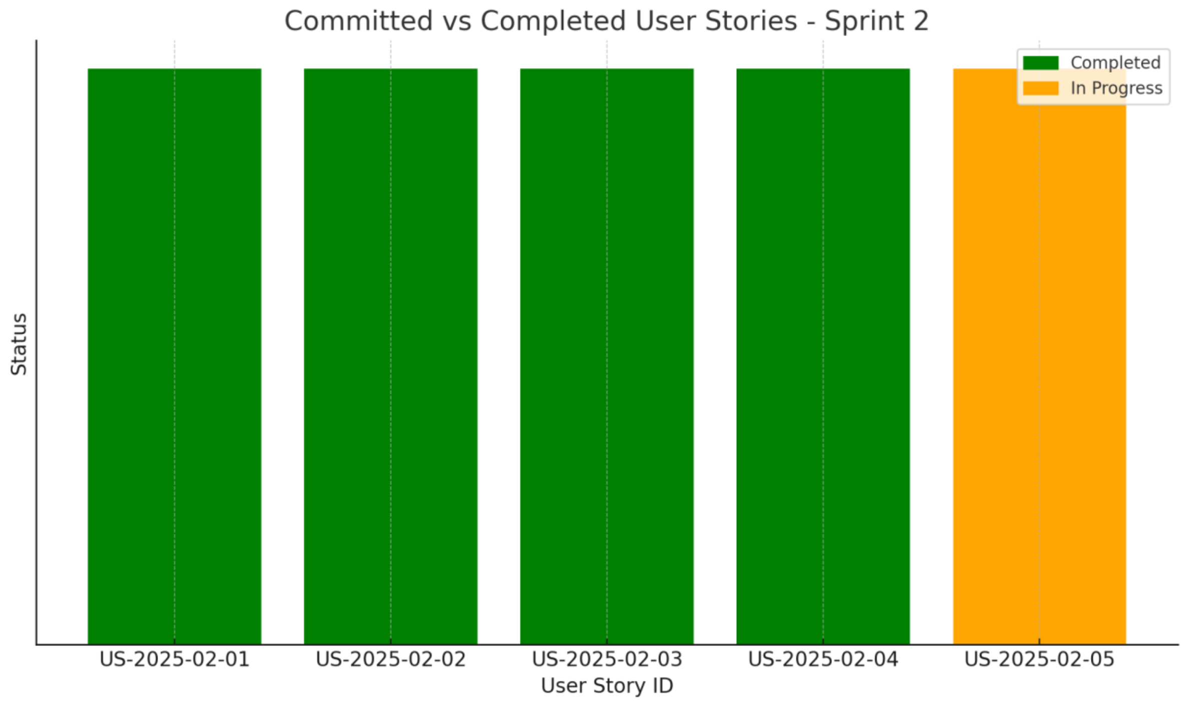
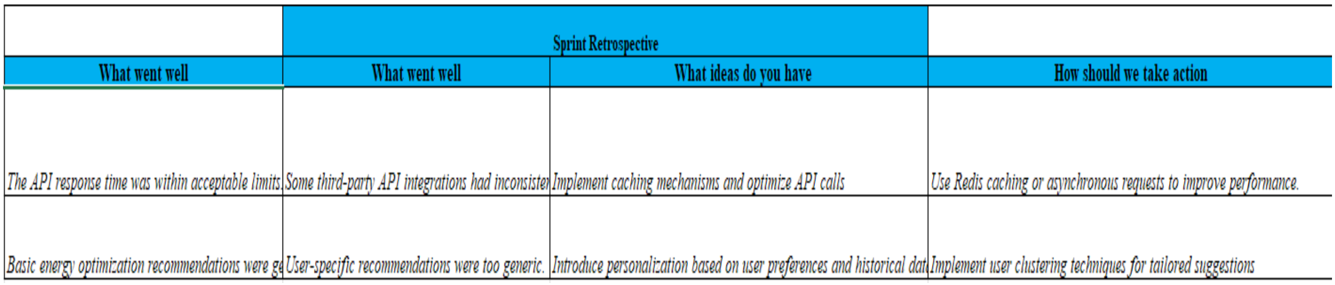


Figure 2.3.7 Commited vs Completed User stories

2.3.8 Sprint Rertospective

During this sprint, the team observed positive outcomes, such as acceptable API response times and the successful generation of basic energy optimization recommendations. However, challenges were identified, including inconsistent response times from third-party APIs and the generic nature of user-specific suggestions. To address these, the team proposed implementing caching mechanisms and personalization techniques. Actionable steps include using Redis caching and asynchronous requests to improve performance, and applying user clustering for more tailored energy recommendations.



	Sprint Retrospective		
What went well	What went well	What ideas do you have	How should we take action
The API response time was within acceptable limits.	Some third-party API integrations had inconsisten	Implement caching mechanisms and optimize API calls	Use Redis caching or asynchronous requests to improve performance.
Basic energy optimization recommendations were ge	User-specific recommendations were too generic.	Introduce personalization based on user preferences and historical dat	Implement user clustering techniques for tailored suggestions

Table 2.3.5 Sprint Retrospective for sprint 3

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Project Outcomes

The final outcome of this energy data analysis and optimization project, building upon the initial foundation laid in the first three sprints, is envisioned as a comprehensive and intelligent energy management solution. This mature system will empower users – from system administrators ensuring data quality to energy analysts seeking deeper insights and the backend module driving tangible savings – with the tools and intelligence needed to effectively manage and optimize energy consumption.At its core, the fully realized system will feature a highly sophisticated and automated data pipeline. This pipeline will seamlessly ingest data from diverse energy sources, perform rigorous cleaning and validation to guarantee accuracy, and store it in an efficient and accessible data repository. The system will proactively identify and flag potential data anomalies, providing alerts and logs for administrators to review, ensuring continuous data integrity without manual intervention.For energy analysts, the final outcome will be a powerful suite of analytical tools and interactive dashboards. These will go far beyond simple peak hour identification, enabling in-depth exploration of energy consumption patterns across various dimensions – time, location, equipment, operational processes, and environmental factors. Analysts will be able to perform advanced trend analysis, identify correlations and causalities, segment energy usage by different categories, and generate customized reports for various stakeholders. Predictive analytics capabilities will be integrated, allowing analysts to forecast future energy demand under different scenarios, aiding in proactive planning and resource allocation. The system will also facilitate anomaly detection, automatically identifying unusual energy consumption patterns that may indicate inefficiencies, equipment malfunctions, or potential energy waste.The backend optimization module will evolve into a highly intelligent engine capable of generating a wide range of actionable energy-saving recommendations. Leveraging advanced algorithms, potentially including machine learning models, the system will analyze historical and real-time data to identify optimization opportunities. These recommendations will move beyond basic peak shaving to include strategies like dynamic load shifting based on energy prices or grid conditions, predictive maintenance alerts based on energy consumption patterns, and suggestions for optimizing operational schedules or equipment usage. The system will also be capable of quantifying the potential energy and cost savings associated with each recommendation, providing a clear return on investment analysis. Furthermore, the system might incorporate closed-loop feedback mechanisms to track the implementation and impact of its recommendations, continuously learning and refining its optimization strategies over time.Ultimately, the final outcome of this project will be a self-sustaining energy management ecosystem. It will provide a single source of truth for all energy-related data, empower analysts with sophisticated tools for understanding and predicting consumption, and autonomously drive energy efficiency through intelligent optimization recommendations. This will lead to significant benefits, including reduced energy costs, improved operational efficiency, a smaller environmental footprint, and enhanced sustainability. The system will be adaptable and scalable, capable of accommodating future data sources, analytical needs, and optimization strategies, ensuring its long-term value and relevance in the evolving energy landscape.

3.2 Committed Vs Completed User stories

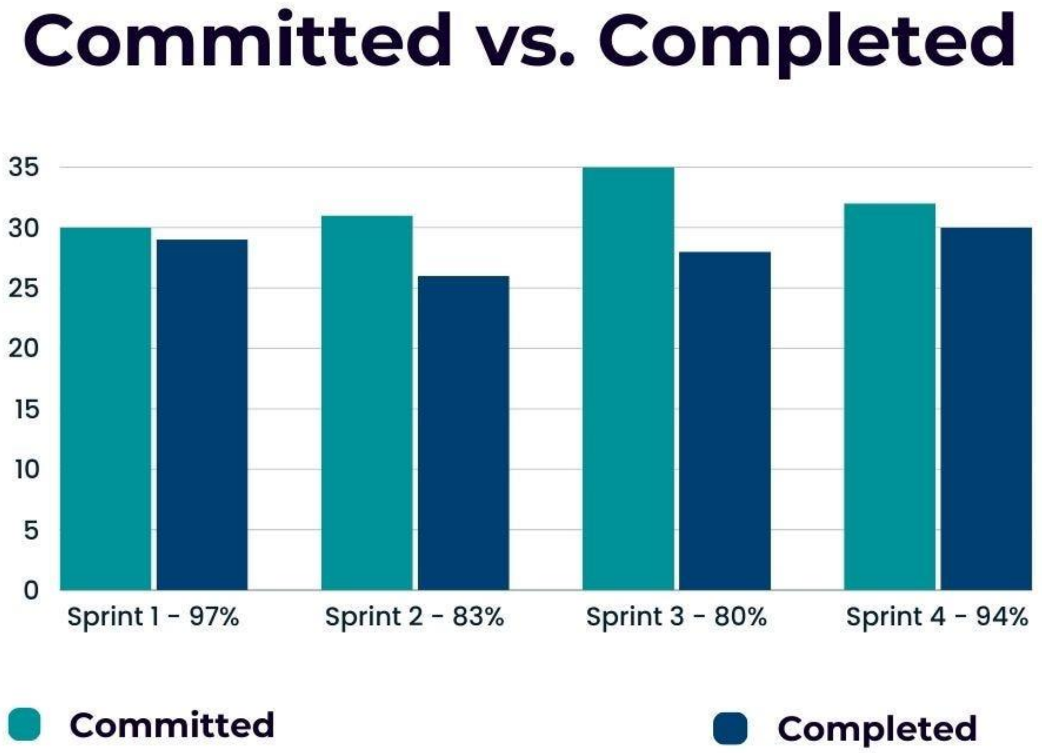


Figure 3.1 Committed Vs Completed User stories

CHAPTER 4

CONCLUSION & FUTURE ENHANCEMENTS

4.1 Conclusion

This project culminates in the creation of a foundational yet powerful system designed to revolutionize how users interact with and understand their energy consumption. By delivering real-time visibility into energy usage, coupled with comprehensive historical data analysis and intelligent alert mechanisms, the system empowers individuals and administrators alike to move beyond reactive energy management towards proactive optimization. The ability for users to pinpoint specific devices contributing significantly to their energy footprint, combined with tailored recommendations for reduction, offers concrete and actionable steps towards achieving greater energy efficiency and tangible cost savings. For administrators, the robust reporting capabilities provide invaluable insights into overarching energy trends, facilitating data-driven decision-making for broader sustainability initiatives and infrastructure improvements. Ultimately, this project establishes a critical stepping stone towards a more energy-conscious and sustainable future, placing the power of understanding and control directly in the hands of the user.

4.2 Future Enhancements

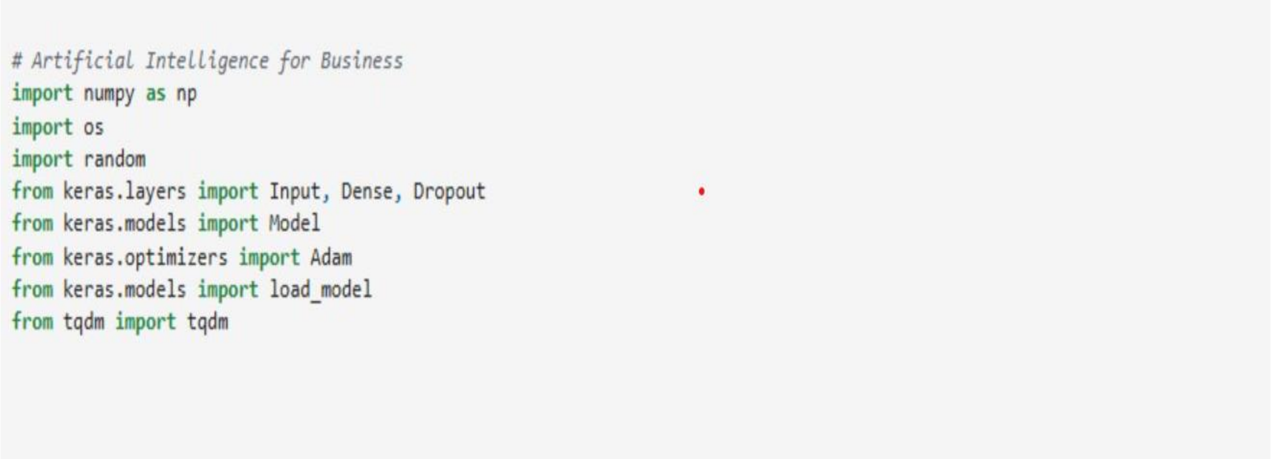
The potential for growth and innovation within this energy monitoring system is substantial, offering numerous avenues for future development that could significantly amplify its impact and user value. A key area for expansion lies in deeper integration with the burgeoning ecosystem of smart home devices and platforms. This would enable seamless interoperability and allow for the implementation of sophisticated, automated energy-saving scenarios across a wider range of household appliances and systems. Furthermore, advancing the system's predictive capabilities through more sophisticated algorithms and the incorporation of external data sources, such as weather forecasts and energy pricing fluctuations, could provide users with highly accurate future consumption predictions and proactive recommendations for minimizing energy usage during peak demand or high-cost periods.

In alignment with the global shift towards sustainable energy, future enhancements could focus on incorporating the monitoring and analysis of renewable energy generation (e.g., solar panel output) and battery storage systems. This would provide users with a holistic view of their energy ecosystem, enabling them to optimize self-consumption and grid interaction. The integration of machine learning techniques holds immense promise for personalizing the user experience further. By learning individual energy consumption patterns and preferences, the system could offer increasingly tailored recommendations, automate energy-saving adjustments with greater precision, and even proactively identify potential energy waste or device malfunctions.

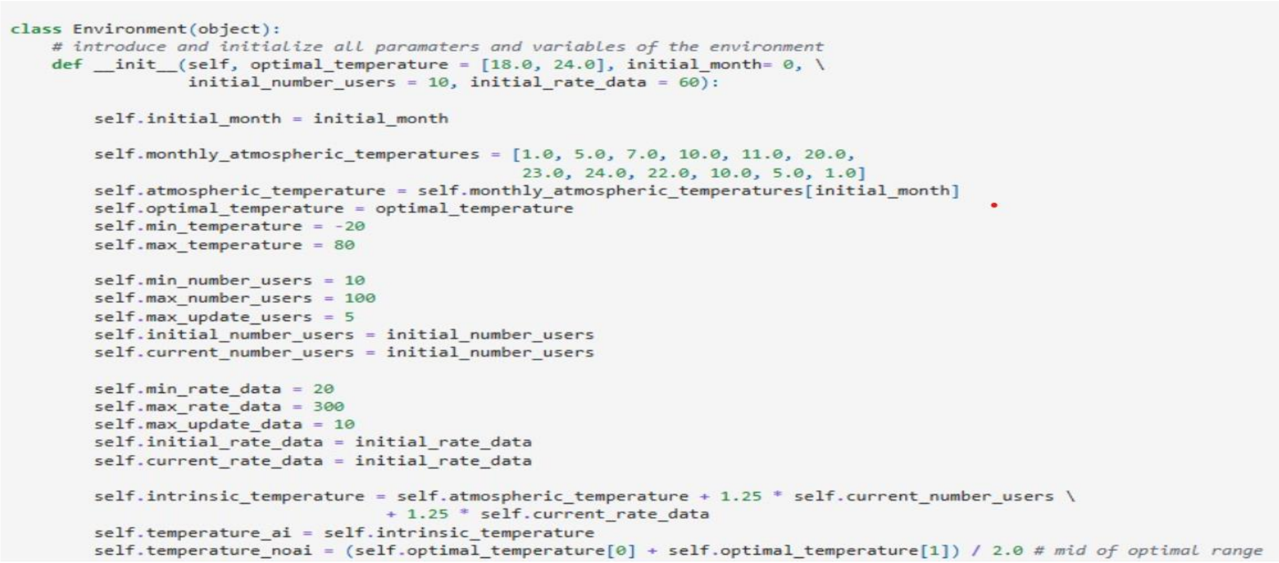
Enhancing the user interface and developing dedicated mobile applications are crucial steps for improving accessibility and user engagement, allowing individuals to monitor and control their energy usage conveniently from anywhere. Exploring the incorporation of community-level features, such as energy consumption benchmarking and gamified challenges, could foster a sense of collective responsibility and encourage friendly competition in energy conservation among neighbors or within communities. Finally, investigating the potential for integration with utility providers could unlock valuable features such as automated participation in demand response programs and streamlined access to energy usage data and billing information. These multifaceted future enhancements promise to transform the system into an even more intelligent, intuitive, and impactful tool for promoting sustainable energy practices at individual and community levels.

APPENDIX

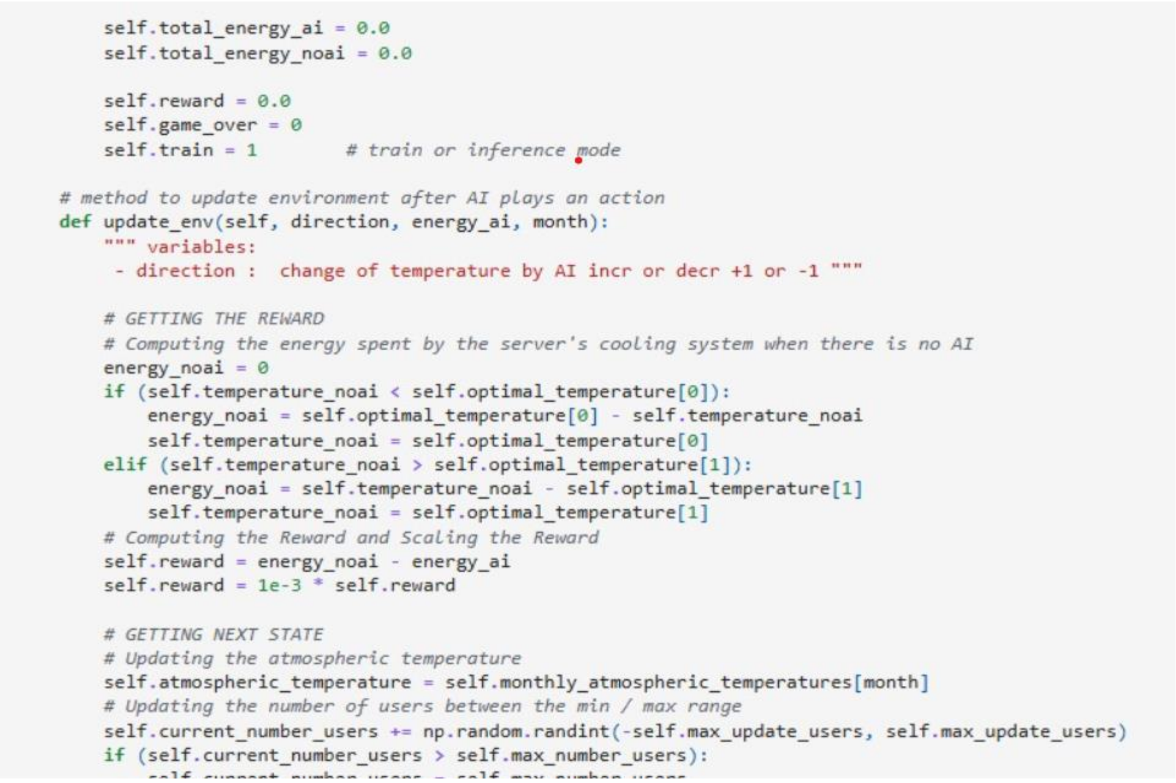
A.SAMPLE CODING



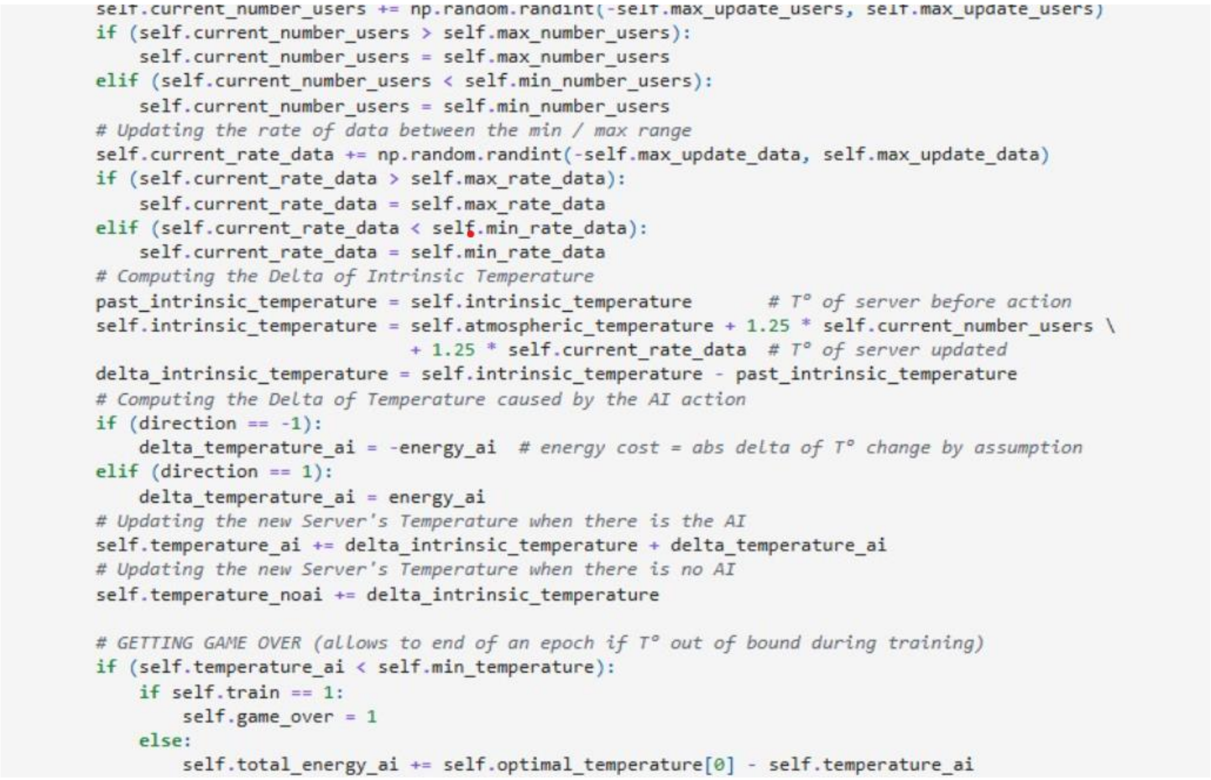
```python
# Artificial Intelligence for Business
import numpy as np
import os
import random
from keras.layers import Input, Dense, Dropout
from keras.models import Model
from keras.optimizers import Adam
from keras.models import load_model
from tqdm import tqdm
```



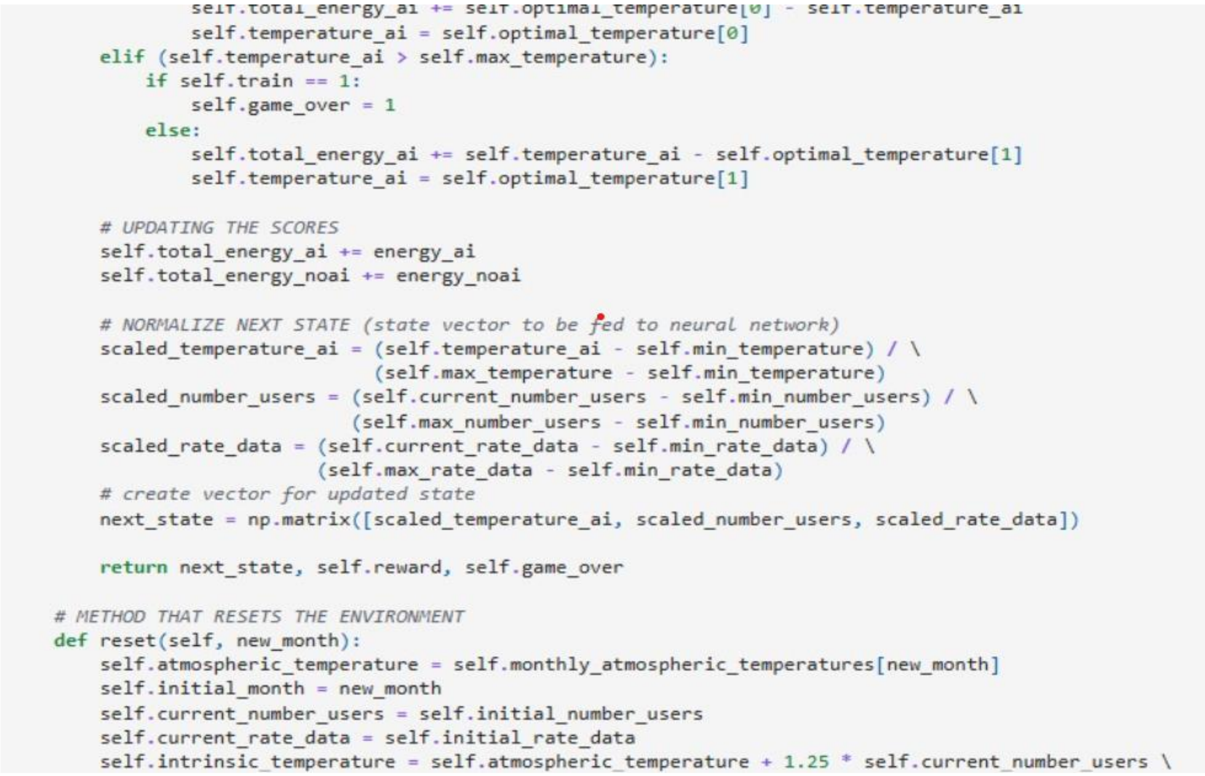
```python
class Environment(object):
    # introduce and initialize all paramaters and variables of the environment
    def __init__(self, optimal_temperature = [18.0, 24.0], initial_month= 0, \
                 initial_number_users = 10, initial_rate_data = 60):

        self.initial_month = initial_month

        self.monthly_atmospheric_temperatures = [1.0, 5.0, 7.0, 10.0, 11.0, 20.0,
                                                 23.0, 24.0, 22.0, 10.0, 5.0, 1.0]
        self.atmospheric_temperature = self.monthly_atmospheric_temperatures[initial_month]
        self.optimal_temperature = optimal_temperature
        self.min_temperature = -20
        self.max_temperature = 80

        self.min_number_users = 10
        self.max_number_users = 100
        self.max_update_users = 5
        self.initial_number_users = initial_number_users
        self.current_number_users = initial_number_users

        self.min_rate_data = 20
        self.max_rate_data = 300
        self.max_update_data = 10
        self.initial_rate_data = initial_rate_data
        self.current_rate_data = initial_rate_data

        self.intrinsic_temperature = self.atmospheric_temperature + 1.25 * self.current_number_users \
                                     + 1.25 * self.current_rate_data
        self.temperature_ai = self.intrinsic_temperature
        self.temperature_noai = (self.optimal_temperature[0] + self.optimal_temperature[1]) / 2.0 # mid of optimal range
```



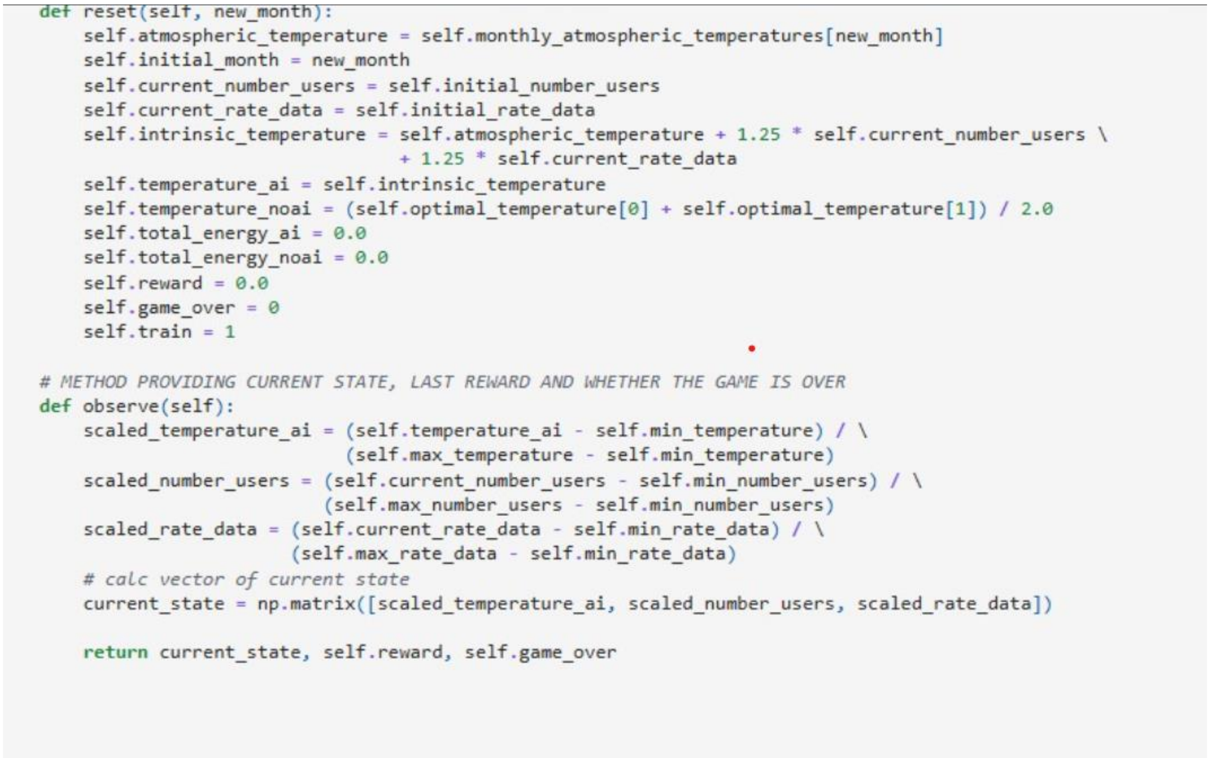
```
        self.total_energy_ai = 0.0
        self.total_energy_noai = 0.0

        self.reward = 0.0
        self.game_over = 0
        self.train = 1          # train or inference mode

    # method to update environment after AI plays an action
    def update_env(self, direction, energy_ai, month):
        """ variables:
         - direction :  change of temperature by AI incr or decr +1 or -1 """

        # GETTING THE REWARD
        # Computing the energy spent by the server's cooling system when there is no AI
        energy_noai = 0
        if (self.temperature_noai < self.optimal_temperature[0]):
            energy_noai = self.optimal_temperature[0] - self.temperature_noai
            self.temperature_noai = self.optimal_temperature[0]
        elif (self.temperature_noai > self.optimal_temperature[1]):
            energy_noai = self.temperature_noai - self.optimal_temperature[1]
            self.temperature_noai = self.optimal_temperature[1]
        # Computing the Reward and Scaling the Reward
        self.reward = energy_noai - energy_ai
        self.reward = 1e-3 * self.reward

        # GETTING NEXT STATE
        # Updating the atmospheric temperature
        self.atmospheric_temperature = self.monthly_atmospheric_temperatures[month]
        # Updating the number of users between the min / max range
        self.current_number_users += np.random.randint(-self.max_update_users, self.max_update_users)
        if (self.current_number_users > self.max_number_users):
            self.current_number_users = self.max_number_users
```



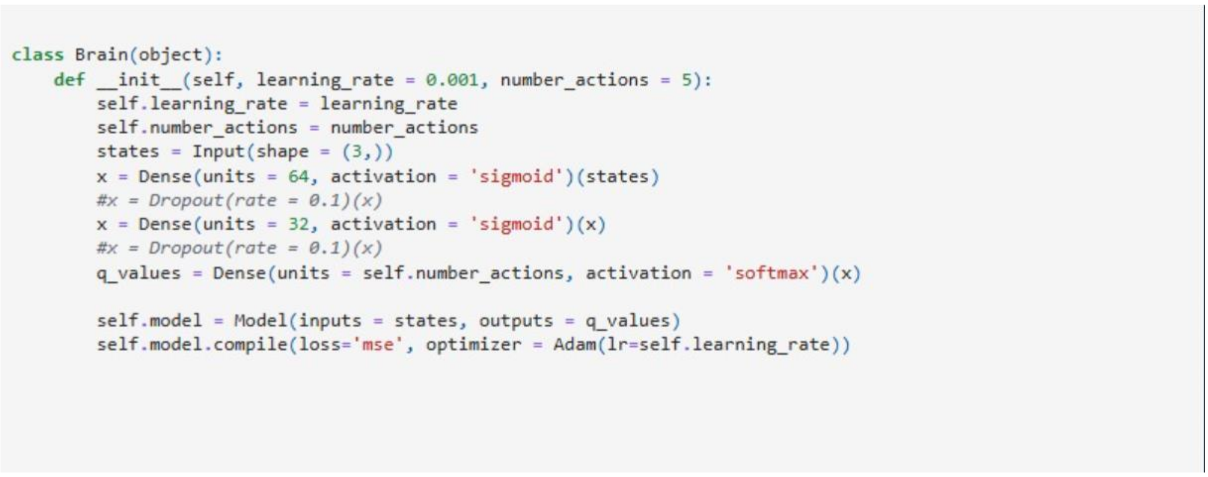
```
        self.current_number_users += np.random.randint(-self.max_update_users, self.max_update_users)
        if (self.current_number_users > self.max_number_users):
            self.current_number_users = self.max_number_users
        elif (self.current_number_users < self.min_number_users):
            self.current_number_users = self.min_number_users
        # Updating the rate of data between the min / max range
        self.current_rate_data += np.random.randint(-self.max_update_data, self.max_update_data)
        if (self.current_rate_data > self.max_rate_data):
            self.current_rate_data = self.max_rate_data
        elif (self.current_rate_data < self.min_rate_data):
            self.current_rate_data = self.min_rate_data
        # Computing the Delta of Intrinsic Temperature
        past_intrinsic_temperature = self.intrinsic_temperature       # T° of server before action
        self.intrinsic_temperature = self.atmospheric_temperature + 1.25 * self.current_number_users \
                                     + 1.25 * self.current_rate_data  # T° of server updated
        delta_intrinsic_temperature = self.intrinsic_temperature - past_intrinsic_temperature
        # Computing the Delta of Temperature caused by the AI action
        if (direction == -1):
            delta_temperature_ai = -energy_ai  # energy cost = abs delta of T° change by assumption
        elif (direction == 1):
            delta_temperature_ai = energy_ai
        # Updating the new Server's Temperature when there is the AI
        self.temperature_ai += delta_intrinsic_temperature + delta_temperature_ai
        # Updating the new Server's Temperature when there is no AI
        self.temperature_noai += delta_intrinsic_temperature

        # GETTING GAME OVER (allows to end of an epoch if T° out of bound during training)
        if (self.temperature_ai < self.min_temperature):
            if self.train == 1:
                self.game_over = 1
            else:
                self.total_energy_ai += self.optimal_temperature[0] - self.temperature_ai
```



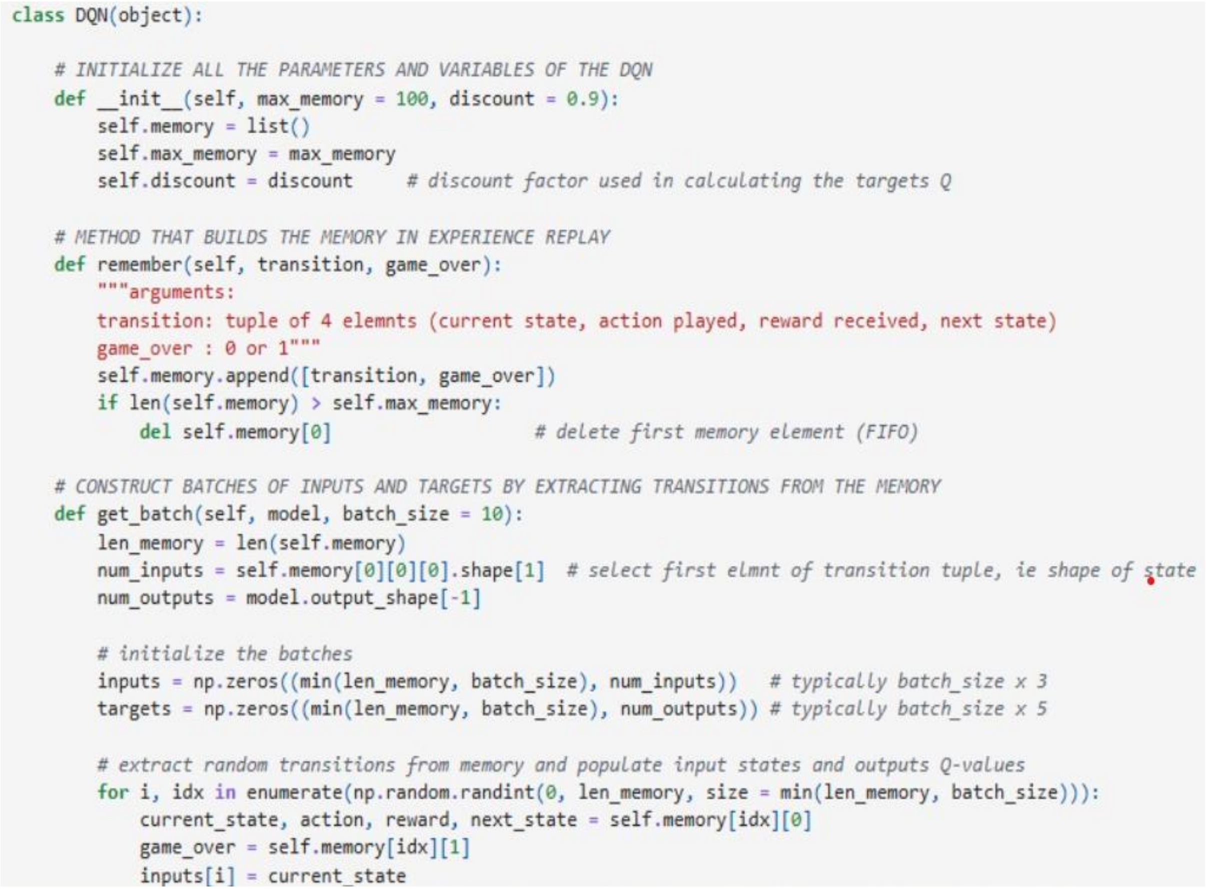
```
            self.total_energy_ai += self.optimal_temperature[0] - self.temperature_ai
            self.temperature_ai = self.optimal_temperature[0]
    elif (self.temperature_ai > self.max_temperature):
        if self.train == 1:
            self.game_over = 1
        else:
            self.total_energy_ai += self.temperature_ai - self.optimal_temperature[1]
            self.temperature_ai = self.optimal_temperature[1]

    # UPDATING THE SCORES
    self.total_energy_ai += energy_ai
    self.total_energy_noai += energy_noai

    # NORMALIZE NEXT STATE (state vector to be fed to neural network)
    scaled_temperature_ai = (self.temperature_ai - self.min_temperature) / \
                            (self.max_temperature - self.min_temperature)
    scaled_number_users = (self.current_number_users - self.min_number_users) / \
                          (self.max_number_users - self.min_number_users)
    scaled_rate_data = (self.current_rate_data - self.min_rate_data) / \
                       (self.max_rate_data - self.min_rate_data)
    # create vector for updated state
    next_state = np.matrix([scaled_temperature_ai, scaled_number_users, scaled_rate_data])

    return next_state, self.reward, self.game_over

# METHOD THAT RESETS THE ENVIRONMENT
def reset(self, new_month):
    self.atmospheric_temperature = self.monthly_atmospheric_temperatures[new_month]
    self.initial_month = new_month
    self.current_number_users = self.initial_number_users
    self.current_rate_data = self.initial_rate_data
    self.intrinsic_temperature = self.atmospheric_temperature + 1.25 * self.current_number_users \
```



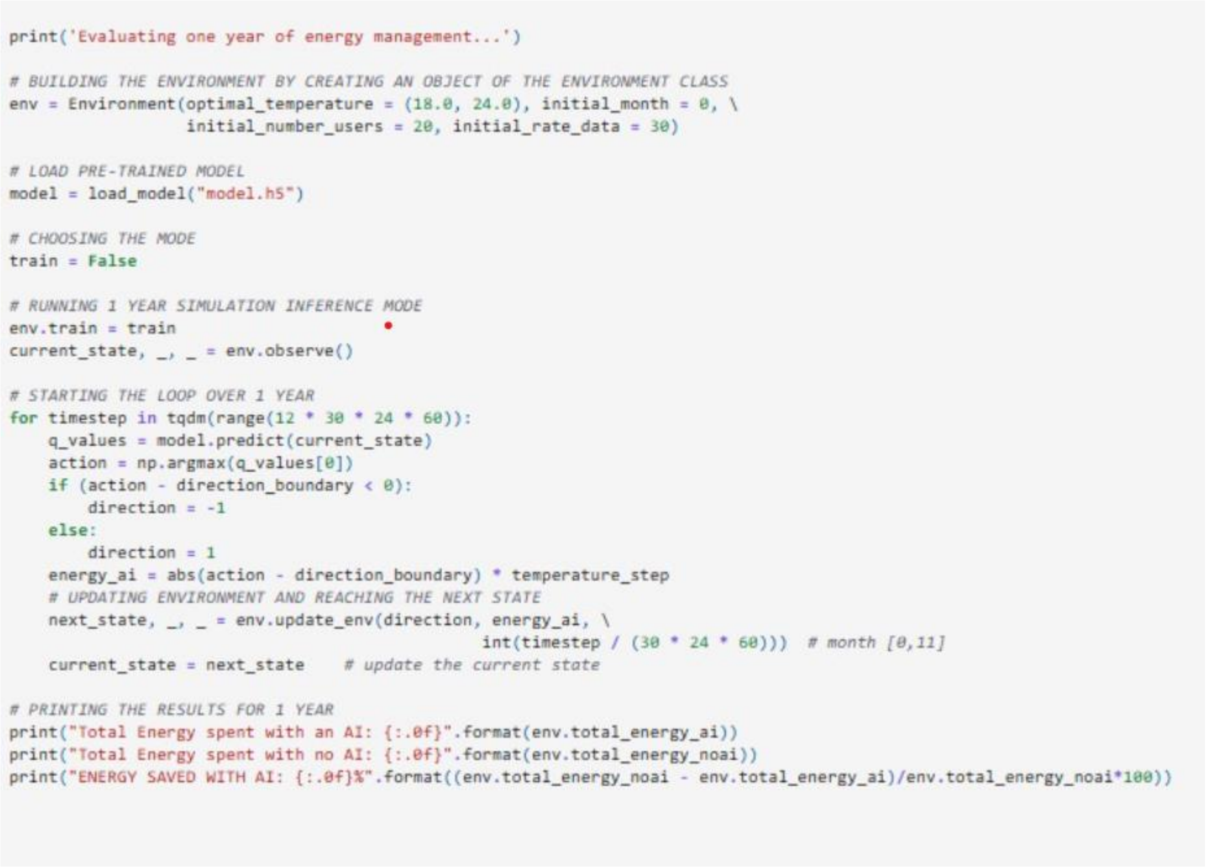
```
def reset(self, new_month):
    self.atmospheric_temperature = self.monthly_atmospheric_temperatures[new_month]
    self.initial_month = new_month
    self.current_number_users = self.initial_number_users
    self.current_rate_data = self.initial_rate_data
    self.intrinsic_temperature = self.atmospheric_temperature + 1.25 * self.current_number_users \
                                 + 1.25 * self.current_rate_data
    self.temperature_ai = self.intrinsic_temperature
    self.temperature_noai = (self.optimal_temperature[0] + self.optimal_temperature[1]) / 2.0
    self.total_energy_ai = 0.0
    self.total_energy_noai = 0.0
    self.reward = 0.0
    self.game_over = 0
    self.train = 1

# METHOD PROVIDING CURRENT STATE, LAST REWARD AND WHETHER THE GAME IS OVER
def observe(self):
    scaled_temperature_ai = (self.temperature_ai - self.min_temperature) / \
                            (self.max_temperature - self.min_temperature)
    scaled_number_users = (self.current_number_users - self.min_number_users) / \
                          (self.max_number_users - self.min_number_users)
    scaled_rate_data = (self.current_rate_data - self.min_rate_data) / \
                       (self.max_rate_data - self.min_rate_data)
    # calc vector of current state
    current_state = np.matrix([scaled_temperature_ai, scaled_number_users, scaled_rate_data])

    return current_state, self.reward, self.game_over
```



```
class Brain(object):
    def __init__(self, learning_rate = 0.001, number_actions = 5):
        self.learning_rate = learning_rate
        self.number_actions = number_actions
        states = Input(shape = (3,))
        x = Dense(units = 64, activation = 'sigmoid')(states)
        #x = Dropout(rate = 0.1)(x)
        x = Dense(units = 32, activation = 'sigmoid')(x)
        #x = Dropout(rate = 0.1)(x)
        q_values = Dense(units = self.number_actions, activation = 'softmax')(x)

        self.model = Model(inputs = states, outputs = q_values)
        self.model.compile(loss='mse', optimizer = Adam(lr=self.learning_rate))
```



```
class DQN(object):

    # INITIALIZE ALL THE PARAMETERS AND VARIABLES OF THE DQN
    def __init__(self, max_memory = 100, discount = 0.9):
        self.memory = list()
        self.max_memory = max_memory
        self.discount = discount     # discount factor used in calculating the targets Q

    # METHOD THAT BUILDS THE MEMORY IN EXPERIENCE REPLAY
    def remember(self, transition, game_over):
        """arguments:
        transition: tuple of 4 elemnts (current state, action played, reward received, next state)
        game_over : 0 or 1"""
        self.memory.append([transition, game_over])
        if len(self.memory) > self.max_memory:
            del self.memory[0]                  # delete first memory element (FIFO)

    # CONSTRUCT BATCHES OF INPUTS AND TARGETS BY EXTRACTING TRANSITIONS FROM THE MEMORY
    def get_batch(self, model, batch_size = 10):
        len_memory = len(self.memory)
        num_inputs = self.memory[0][0][0].shape[1]  # select first elmnt of transition tuple, ie shape of state
        num_outputs = model.output_shape[-1]

        # initialize the batches
        inputs = np.zeros((min(len_memory, batch_size), num_inputs))   # typically batch_size x 3
        targets = np.zeros((min(len_memory, batch_size), num_outputs)) # typically batch_size x 5

        # extract random transitions from memory and populate input states and outputs Q-values
        for i, idx in enumerate(np.random.randint(0, len_memory, size = min(len_memory, batch_size))):
            current_state, action, reward, next_state = self.memory[idx][0]
            game_over = self.memory[idx][1]
            inputs[i] = current_state
```



```
print('Evaluating one year of energy management...')

# BUILDING THE ENVIRONMENT BY CREATING AN OBJECT OF THE ENVIRONMENT CLASS
env = Environment(optimal_temperature = (18.0, 24.0), initial_month = 0, \
                  initial_number_users = 20, initial_rate_data = 30)

# LOAD PRE-TRAINED MODEL
model = load_model("model.h5")

# CHOOSING THE MODE
train = False

# RUNNING 1 YEAR SIMULATION INFERENCE MODE
env.train = train
current_state, _, _ = env.observe()

# STARTING THE LOOP OVER 1 YEAR
for timestep in tqdm(range(12 * 30 * 24 * 60)):
    q_values = model.predict(current_state)
    action = np.argmax(q_values[0])
    if (action - direction_boundary < 0):
        direction = -1
    else:
        direction = 1
    energy_ai = abs(action - direction_boundary) * temperature_step
    # UPDATING ENVIRONMENT AND REACHING THE NEXT STATE
    next_state, _, _ = env.update_env(direction, energy_ai, \
                                                int(timestep / (30 * 24 * 60)))  # month [0,11]
    current_state = next_state    # update the current state

# PRINTING THE RESULTS FOR 1 YEAR
print("Total Energy spent with an AI: {:.0f}".format(env.total_energy_ai))
print("Total Energy spent with no AI: {:.0f}".format(env.total_energy_noai))
print("ENERGY SAVED WITH AI: {:.0f}%".format((env.total_energy_noai - env.total_energy_ai)/env.total_energy_noai*100))
```

B.PLAGIARISM REPORT