Sustainable Monitoring: Reducing Carbon Footprint and Energy Use with Sensor Fusion

TEAM MEMBERS

Prakhar Shrivastava
Puja Rani Bhuyan
Rakshit Verma
Harsh Yadav

1.Detailed Project Plan:

1. Project objective :

The project aims to develop a **cloud-based web application**, with future **mobile app integration**, designed to help users monitor and optimize energy consumption while reducing their carbon footprint. Leveraging **IoT-enabled smart home sensors**, the application will utilize **AI-driven algorithms** to analyze real-time data, such as electricity usage, appliance efficiency, and environmental conditions. The system will provide users with actionable insights, offering **automated recommendations** to optimize energy usage, such as modifying appliance schedules or switching to energy-efficient modes. The AI component will leverage **machine learning** to detect consumption patterns and suggest optimizations, contributing to reduced energy costs and carbon emissions. The application will be hosted on the **Vultr cloud platform**, providing **scalable infrastructure** for real-time data processing and **reliable performance**. By offering users detailed analytics and visualization tools to track energy usage and environmental impact, the solution promotes sustainability and supports **global energy efficiency standards**. The platform will expand to mobile devices, enhancing accessibility and user convenience.

2. Detailed timeline with milestones

Month 1: Initial Setup and Core Backend Development

Weeks 1-2: Project Initiation and Environment Setup

1. **Milestones:**

- Finalize project requirements and architecture specifications.
- Set up the development environment and team collaboration tools.
- Create initial prototypes of core components for testing.

1. Tasks:

1. Define functional and non-functional requirements.

2.Finalize technology stack and libraries.

3.Set up Vultr cloud account and configure initial instances.

4.Deploy databases (e.g., time-series database, user profile database).

Weeks 3-4: IoT Device Integration and Edge Gateway Development

2.Milestones:

a. Integrate IoT-enabled sensors with basic data collection capabilities.
b. Develop and test the Edge Gateway for data aggregation and secure transmission to the cloud.

2.Tasks:

1. Install IoT devices in a test environment and configure connectivity.
2. Implement the Edge Gateway's Data Aggregation and Secure Data Transmission modules.
3. Test real-time data transmission from IoT devices through the Edge Gateway to the cloud.
4. Ensure data integrity and accuracy from IoT devices to cloud storage.

Month 2: Backend Processing, AI Module, and Initial Frontend Development

Weeks 5-6: Real-Time Data Processing and Storage Setup

3.Milestones:

a. Complete the setup of backend processing modules on Vultr, including the Data Ingestion API and Real-Time Data Processing Module.
b. Implement initial data storage configurations (time-series, user profiles, cache).

3.Tasks:

c. p caching mechanisms.
d. Test end-to-end data flow from IoT sensors to cloud storage.Develop the Data Ingestion API to receive data from the Edge Gateway.
e. Configure real-time data storage and set u

Weeks 7-8: AI and Machine Learning Model Development

4.Milestones:

a. Develop and train initial machine learning models for pattern detection and energy usage predictions.
b. Build the Recommendation Engine to generate actionable insights.

4.Tasks:

c. Preprocess data and train models to identify energy consumption patterns and forecast future usage.
d. Develop recommendation algorithms based on usage patterns and optimization goals.
e. Test and validate AI outputs to ensure accuracy and relevance.
f. Begin integrating AI-generated insights with backend processing.

Month 3: Frontend Development, Testing, and Deployment Preparation

Weeks 9-10: Frontend Development and Mobile App Prototype

5.Milestones:

a. Develop core features of the web application’s frontend, including the dashboard, data visualizations, and user settings.
b. Build a prototype of the mobile app for testing (if feasible).

4.Tasks:

c. Design a user-friendly web UI for monitoring energy consumption, accessing insights, and managing settings.
d. Implement visualizations (charts, graphs) to display real-time and historical energy data.
e. Develop API calls between the frontend and backend to retrieve and display data.
f. Create a basic mobile app prototype to test accessibility and UI components.

Weeks 11-12: System Testing, Optimization, and Final Deployment

6.Milestones:

a.Conduct system-wide testing, covering unit tests, integration tests, and performance tests.

b.Finalize configurations for live deployment on Vultr.

6.Tasks:

A . Perform functional and performance testing of the web and mobile interfaces, backend, and IoT integration.

B. Optimize data processing pipelines, caching, and frontend responsiveness.

C .Finalize the setup for user data encryption, access control, and secure data transmission.

D .Deploy the web application to production on Vultr, and conduct post-deployment tests

Description of deliverables

1.Web Application MVP:

- A functional, user-friendly web interface allowing users to monitor energy consumption in real-time.
- Initial analytics dashboards showcasing key metrics like electricity usage, appliance efficiency, and overall environmental impact.

2.Data Integration with IoT Sensors:

- Integration with IoT-enabled smart home sensors to gather real-time data on energy usage.
- API connections for data ingestion, including support for various smart home devices.

3.AI-Driven Analytics and Recommendations Engine:

- Development of machine learning algorithms to analyze energy consumption patterns.
- A recommendations engine providing automated suggestions, such as optimized appliance schedules, energy-efficient modes, and insights into reducing costs and emissions.

4.Cloud Infrastructure on Vultr:

- Deployment of the web application and data processing backend on Vultr for scalability and performance.
- Implementation of data storage solutions for historical energy usage and environmental impact data.

5.Detailed Analytics & Visualization Tools:

- Creation of interactive dashboards and visualizations, allowing users to track energy metrics over time.
- Comparative analysis tools to benchmark against efficiency standards or personal energy goals.

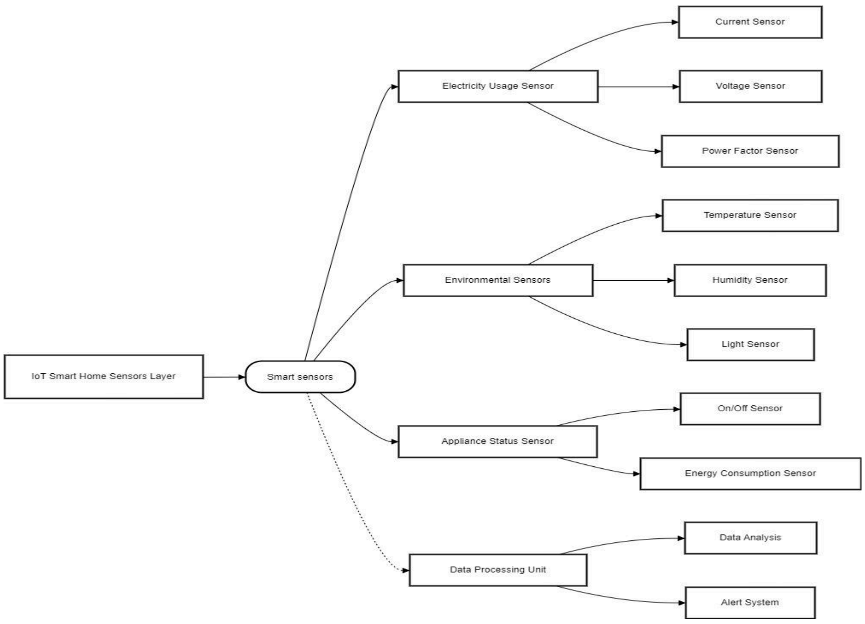
6.Future Mobile App Expansion:

- Planning and documentation for mobile app integration to improve user accessibility.
- Prototyping a responsive design to ensure a seamless user experience across web and mobile platforms.

7.Documentation and Training:

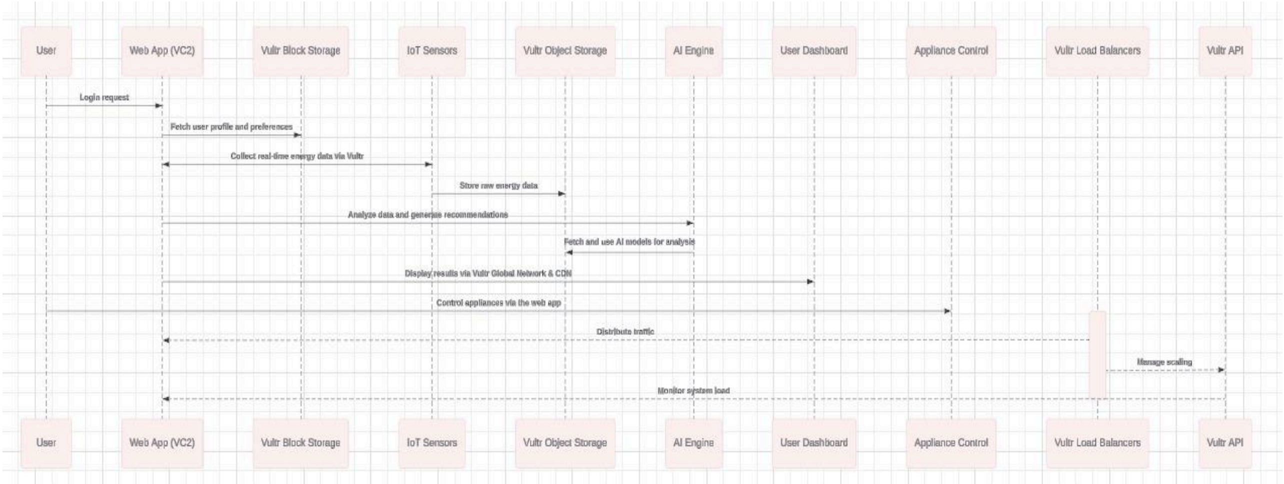
- Comprehensive technical documentation for the system architecture, data flows, and API integrations.
- User guides, FAQs, and training materials to help users fully leverage the app’s functionalities.

2.Architecture Diagrams:



3.Technical Documentation:

- System Architecture And Design:



- **IoT Device Layer**
- **Edge Gateway Layer**
- **Cloud Infrastructure Layer (Vultr Cloud)**
- **AI and Analytics Layer**
- **Application Layer**
- **Mobile App Integration Layer**

1.Smart Sensors

- **Function**: Collect real-time data on energy consumption, appliance activity, and environmental factors (e.g., temperature, humidity).
- **Purpose**: Provide data for analyzing energy usage and determining efficiency levels, which are essential for detecting usage patterns and inefficiencies.

2.**Data Aggregation Module**

- **Function**: Collects data from multiple sensors and consolidates it into a single data stream.
- **Purpose**: Ensures efficient data transmission to the cloud, reducing bandwidth usage and minimizing delays.

Local Processing Module

- **Function**: Conducts basic data filtering, processing, and light computations (e.g., detecting abnormal usage patterns).
- **Purpose**: Reduces the amount of raw data sent to the cloud, making the system faster and more efficient while ensuring that only useful data is processed further.

3. Data Ingestion API

- **Function**: Receives incoming data from the Edge Gateway, organizes it, and routes it for processing and storage.
- **Purpose**: Acts as the system's entry point, managing the incoming data flow to ensure that all data is appropriately handled.

4. Data Preprocessing Module

- **Function**: Cleans and formats raw data for compatibility with AI models.
- **Purpose**: Prepares data for machine learning analysis, enhancing the accuracy and reliability of the insights generated by the AI.

Machine Learning Models

- **Pattern Recognition Model**: Identifies common consumption patterns by analyzing historical data.
- **Purpose**: Provides insights into typical energy use behaviors, helping to recognize potential inefficiencies.

5.API Gateway

- **Function**: Manages all communication between the application frontend and backend services, ensuring smooth and secure data flow.
- **Purpose**: Centralizes access to data and services, optimizing and securing interactions between the user interface and backend components.

6. Mobile Frontend (iOS/Android)

- **Function**: Provides a mobile-optimized interface with core functionalities of the web application.
- **Purpose**: Increases accessibility, allowing users to check their energy usage and receive alerts while on the go.

Push Notification Service

- Function: Delivers notifications and real-time alerts directly to the user's mobile device.
- Purpose: Keeps users actively engaged with timely recommendations and alerts, supporting energy-saving actions anytime, anywhere.

Usage Instructions

1. User Registration and Login

- Open the web or mobile application and register a new user account.

- Log in using your credentials to access the energy monitoring dashboard.

2. Linking Devices and Configuring Settings

- **Add Devices**: In the application's device management section, register each IoT device by providing its unique identifier and connecting it to your account.
- **Device Settings**: Configure specific settings for each device, such as scheduling or energy-saving modes.
- **Profile Preferences**: Set your preferred notification frequency, energy-saving goals, and other personalization settings.

3. Monitoring and Analytics Dashboard

- **Energy Consumption View**: Navigate to the dashboard to see real-time data on energy consumption, appliance efficiency, and environmental impact.
- **Historical Data Analysis**: View historical data to analyze usage trends and patterns over different time periods.
- **Environmental Impact Metrics**: Access detailed metrics on your carbon footprint and see suggestions to improve sustainability.

4. Recommendations and Alerts

- **Automated Insights**: Review AI-driven recommendations for energy usage optimizations, like adjusting schedules or switching devices to eco-mode.
- **Real-Time Alerts**: Enable notifications to receive real-time alerts about unusual energy consumption or immediate suggestions to reduce energy usage.

5. Modifying Settings and Controlling Appliances

- **Remote Appliance Control**: Use the application to modify appliance settings, turn devices on/off, or switch them to energy-saving modes.
- **Automation Rules**: Create custom automation rules (e.g., turn off lights at a certain time) to enhance energy savings without manual intervention.

6. Accessing Data Visualizations and Reports

- **Charts and Graphs**: View dynamic graphs displaying your energy consumption, cost projections, and sustainability metrics.
- **Export Reports**: Download reports summarizing energy usage over selected periods, useful for tracking progress or sharing insights.

Maintenance Tips

- **Update Models**: Periodically update AI models to improve accuracy as new data becomes available.
- **Monitor Device Health**: Check IoT device connections and status regularly to ensure accurate data flow.
- **Software Updates**: Apply updates to the application and backend as they become available to ensure compatibility and security.

4.Prototype or Initial Codebase:

GitHub repository link: https://github.com/Lucifer2987/EnergyEye