**IOT CASE STUDY – 1**

**SMART ENERGY MONITOR**

1. **Model Overview:**

**Introduction:**

The Smart Electricity Monitor is a project that aims to provide an efficient solution for monitoring and analysing the electricity consumption of a home or a building. The project uses IOT-Enabled micro-controllers to collect data on the electrical parameters such as voltage, current, and power consumption.

The data is then processed and analysed using various data analytics techniques to provide insights to the user, helping them optimize their energy usage and reduce their electricity bills.

The system also provides a mobile application interface that allows the user to view the live readings of the meter, view historic data and analysis, remotely control the power supply to their devices and receive alerts and notifications in case of voltage fluctuations or abnormal electricity usage.

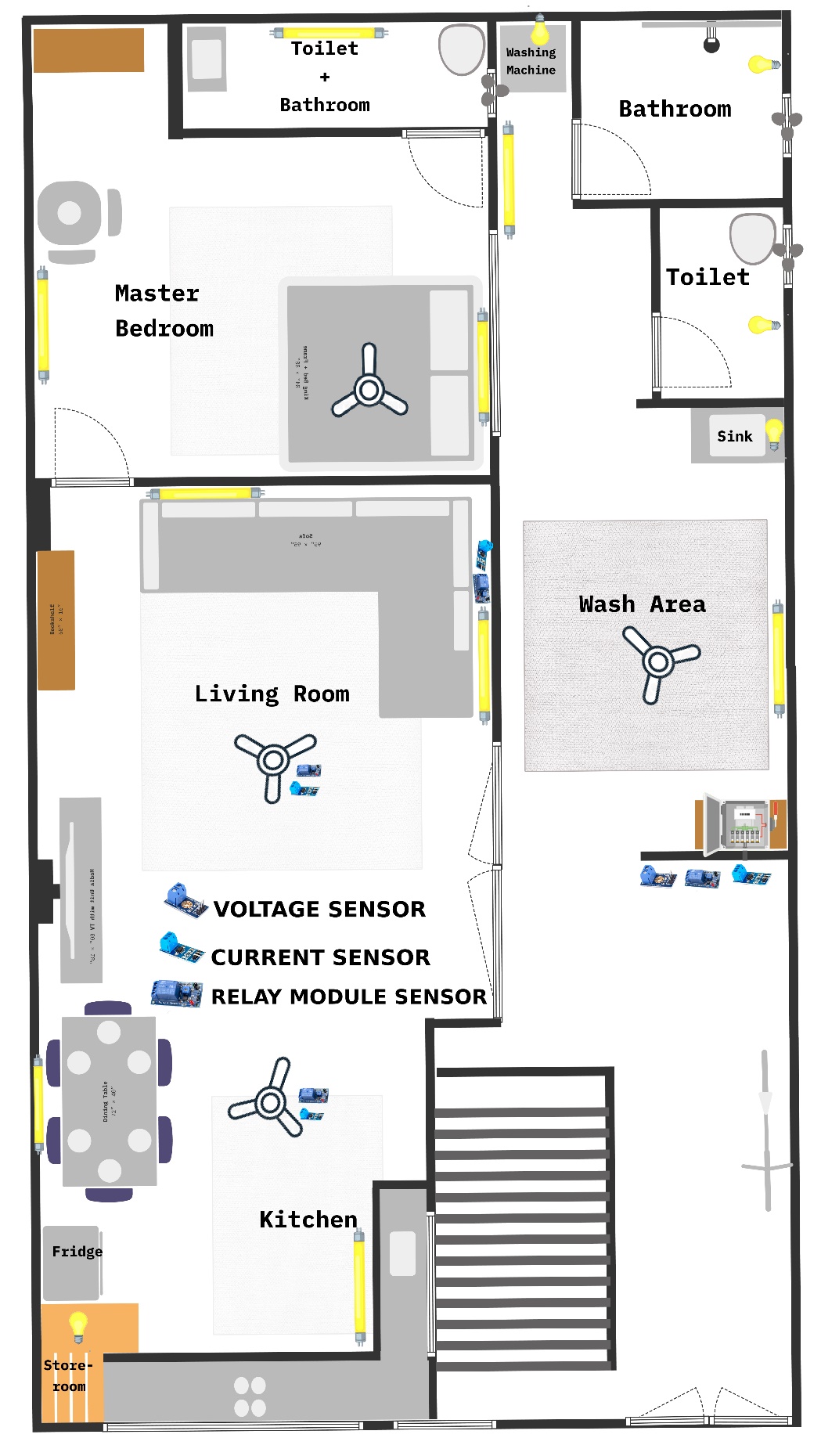
The project utilizes hardware components such as sensors, microcontrollers, and relay switches to collect, process, and control the electricity consumption.

The Smart Electricity Monitor project is an innovative and practical solution for energy management, contributing towards a sustainable future.

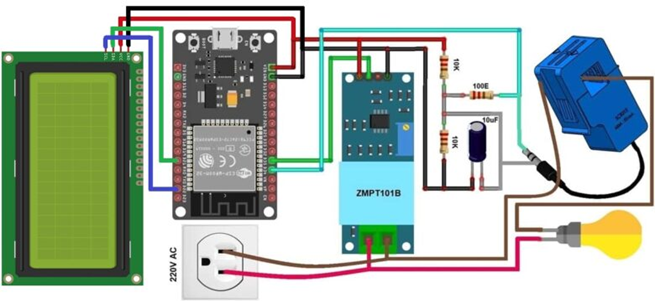
**Implementation:**

**Here’s a Step-By-Step Explanation of how we intent to implement the project:**

1. **Hardware Setup:** The hardware setup involves connecting sensors such as current transformers and voltage transformers to the electrical supply lines. These sensors are connected to the microcontroller such as the ESP32, which is used to process the data received from the sensors. Relay switches are also connected to the microcontroller, which can control the power supply to the devices.
2. **Data Collection:** The microcontroller collects data on the electrical parameters such as voltage, current, and power consumption from the sensors. The collected data is then stored in the microcontroller.
3. **Data Processing:** The microcontroller processes the collected data and calculates the power consumption of the devices. The processed data is then sent to the backend server using a communication protocol such as Wi-Fi, Bluetooth, or GSM.
4. **Data Analysis:** The backend server receives the data and analyses it using various data analytics techniques such as statistical analysis, data mining, and machine learning. The results of the analysis are then sent to the mobile application, which displays the insights to the user.
5. **User Interface:** The user interface consists of a mobile application that displays real-time data and analytics results such as daily, weekly and monthly power consumption to the user. The application provides features such as alerts, notifications, billing, and historical data analysis. The user can also remotely control the power supply to their devices using the mobile application.
6. **Power Control:** The microcontroller can control the power supply to the devices using the relay switches. The user can remotely turn on or off devices from the mobile application. The system also has the capability to turn off devices automatically when they are not in use, using sensors such as motion sensors or light sensors.
7. **Sensor placement model in farm for plant health identification and irrigation system.**



1. **Control unit design using Fritzig Software:**



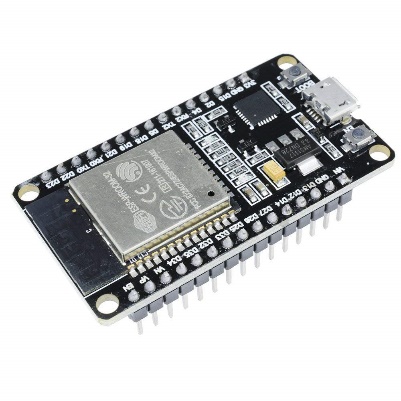
**4. Communication Technologies used in the project:**

* **ESP32:** This is a low-cost, low-power Wi-Fi and Bluetooth-enabled microcontroller that can be used as the primary communication device between the smart electricity meter and the internet.
* **Wi-Fi Router:** We need a Wi-Fi router to connect the Smart Meter to the backend for data sharing. We can use the Wi-Fi routers present in households for this purpose.
* **Smartphone:** This can be used as a user interface for the smart electricity monitoring application. The user can install our application to view live data and analysis of historic data, get the estimated monthly bill and also remotely control devices.
* **Cloud-based server**: This is the backend infrastructure that collects, stores, and processes the data from the smart electricity meter. The application server can communicate with the smart electricity meter and then process the data and send it to the user’s mobile application.

**Communication Protocols used in the project:**

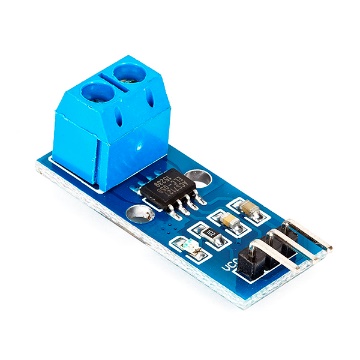
* **Wi-Fi:** The Wi-Fi technology can be used to establish a wireless connection between the microcontroller and the backend server. The ESP32 microcontroller has built-in Wi-Fi capabilities, making it an ideal choice for the project.
* **HTTP:** The HTTP (Hypertext Transfer Protocol) technology can be used to transfer data between the mobile application and the backend server. The mobile application can use HTTP requests to send data to the backend server and receive responses in JSON format.

**5. Selection of hardware components in the solution.**

**List of hardware sensors used for this project:**

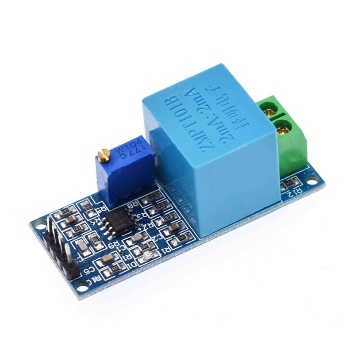
* **ESP32 Microcontroller:** The ESP32 is a low-cost, low-power consumption, and highly integrated microcontroller with built-in Wi-Fi and Bluetooth connectivity.

The ESP32 will be the main microcontroller used in the project, responsible for collecting data from sensors, processing the data, and transmitting it to the cloud server and the user's mobile app.



* **ACS712 Current Sensor:** The ACS712 is a Hall Effect-based current sensor that can measure AC and DC currents accurately.

It will be used to measure the current consumption of household appliances by connecting it in series with the power supply of each device, and the output can be fed into the IoT device to calculate the total energy consumption of each device.



* **ZMPT101B Voltage Sensor:** The ZMPT101B voltage sensor is a transformer-based voltage sensor that can measure AC voltages accurately.

Similar to the current sensor, this sensor also can be connected to the microcontroller to calculate the total voltage consumption of each device.



* **JHD162A LCD Display:** The JHD162A is a 16x2 character LCD display that can display alphanumeric characters.

It will be used to display the real-time data of the energy consumption, current and voltage readings, and alerts to the user.

* **Relay Module:** A relay module is an electrical switch that can be controlled by a digital signal.

In our project, it is used to let the user control his home appliances remotely from their mobile app.



* **Breadboard:** A breadboard is a reusable solderless board used to build and test electronic circuits.

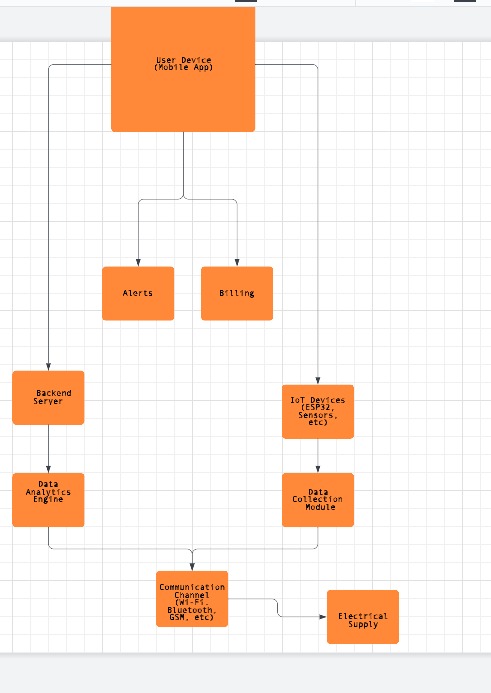
It can be used to connect the microcontroller to the other sensors in the meter.



* **Resistors:** Resistors are electronic components that restrict the flow of electric current in a circuit.

Resistors will be used in the project to limit the current and voltage levels, and to provide a biasing voltage to the sensors and other electronic components used in the circuit.

1. **Deployment level diagram of problem.**

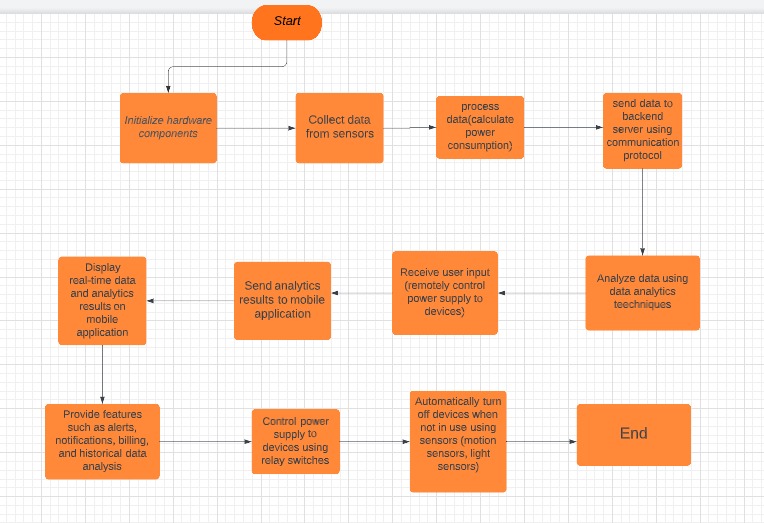


In the above diagram, the IoT devices (ESP32, sensors, and relay switches) are deployed on the electrical supply lines to collect data on electrical parameters such as voltage, current, and power consumption. The data collected is then processed and analysed by the data analytics engine deployed on the backend server.

The backend server communicates with the IoT devices using communication channels such as Wi-Fi, Bluetooth, or GSM. The server also provides a mobile application interface to the user, where they can view real-time data and analytics results, control power supply to their devices, and receive alerts and notifications.

The electrical supply is the physical infrastructure where the IoT devices are deployed, and where the power supply to the devices is controlled by the relay switches.

1. **Flowchart of the software process.**



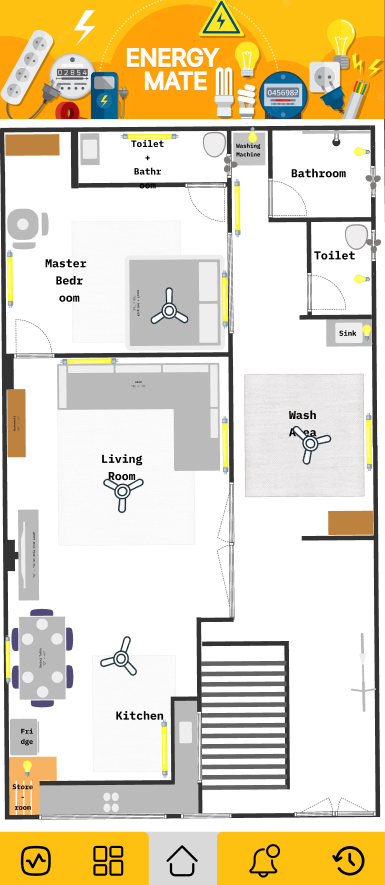
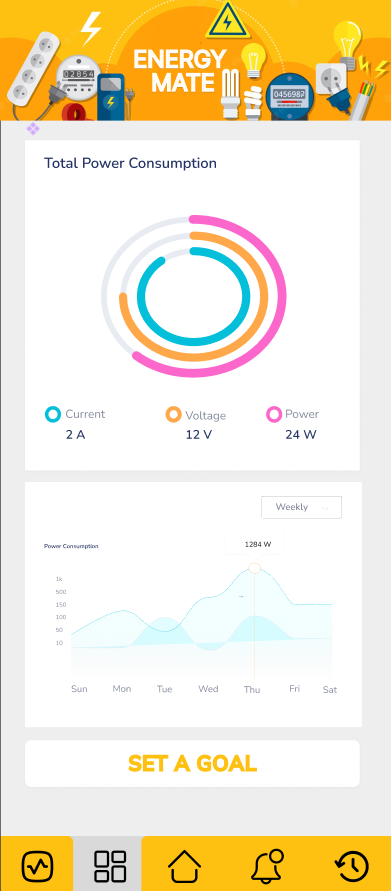
1. **Data analytics in edge or cloud**

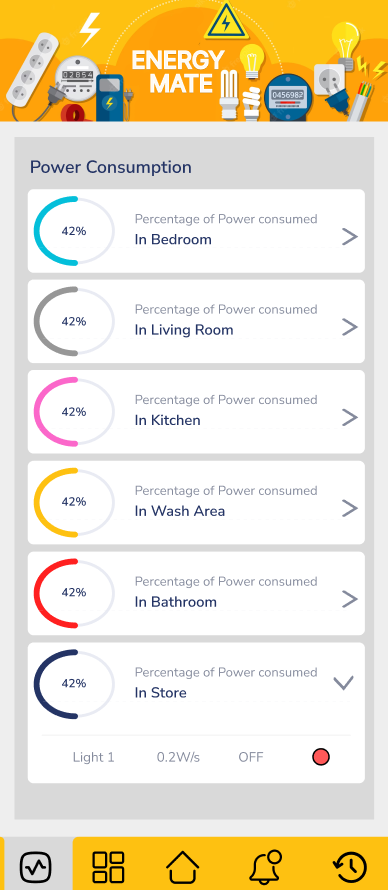
After collecting data from the meter, we then proceed to store the data in cloud storage. We then process the stored data and provide data visualizations to the user.

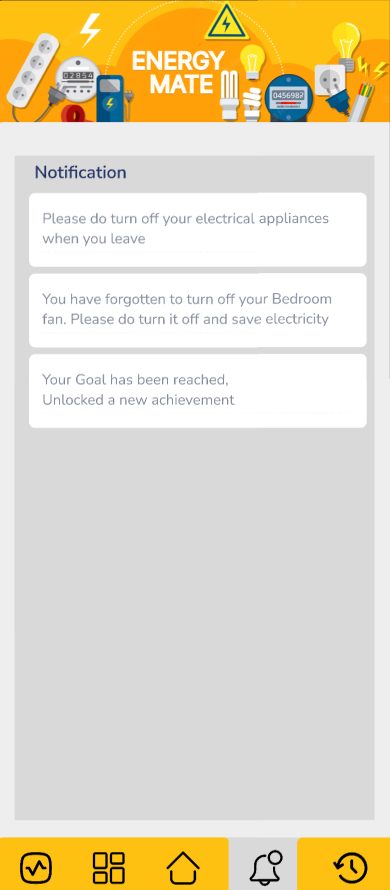
**Here is a list of steps that are involved in our data analysis phase:**

* **Data collection and storage:** In this step, the data like live current, voltage readings along with fluctuations alerts are received from the smart meter and stored in a database to perform historical data analysis.
* **Data Cleansing and pre-processing:** The received data might not be in proper format or might have some false data. Hence, we clean and pre-process the data before analysing it.
* **Data Analysis:** After pre-processing, we get the processed data and analyse the live data in order to provide any alerts/ notifications to the user in cases like power fluctuations. We can use the pre-processed data to provide an estimated Electricity Bill for that month to the user. Also, we retrieve historical data and display it to the user using data visualization techniques.
* **Data Visualization:** After fetching historical data, we can use data visualization graphs like bar charts and pie charts to the user in order to brief them about their power consumption on a daily, weekly, monthly or yearly basis.

1. **Application design with UI design (all web pages).**



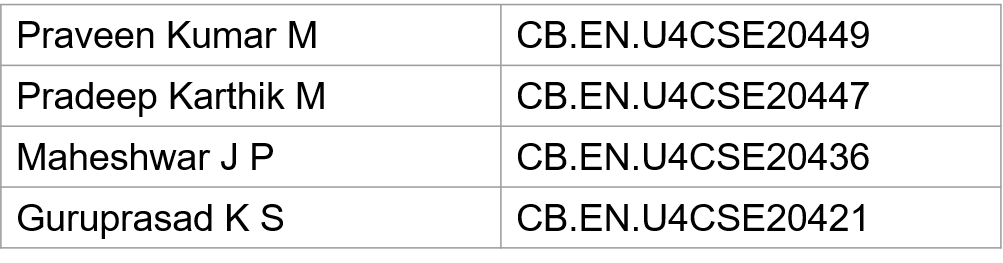






**10. Budget for Case study prototype building:**

* **Budget for Hardware components and sensors:**
* **ESP32 Microcontroller:** INR 500
* **ACS712 Current Sensor:** INR 200
* **ZMPT101B Voltage Sensor:** INR 120
* **JHD162A LCD DISPLAY:** INR 250
* **Relay Module:** INR 150
* **Breadboard:** INR 50
* **Resistors and Connecting Wires:** INR 50
* **Cloud Services:** We can use free hosting services like Render or Heroku. Or Else we can make use of the $100 free credits provided by AWS.



**TEAM MEMBERS**