**INTRODUCUTION**

Rainwater harvesting has emerged as a sustainable solution to combat water scarcity, particularly in urban areas where water wastage is a growing concern. However, traditional methods of monitoring rainfall and managing water collection are often inefficient and lack real-time insights.

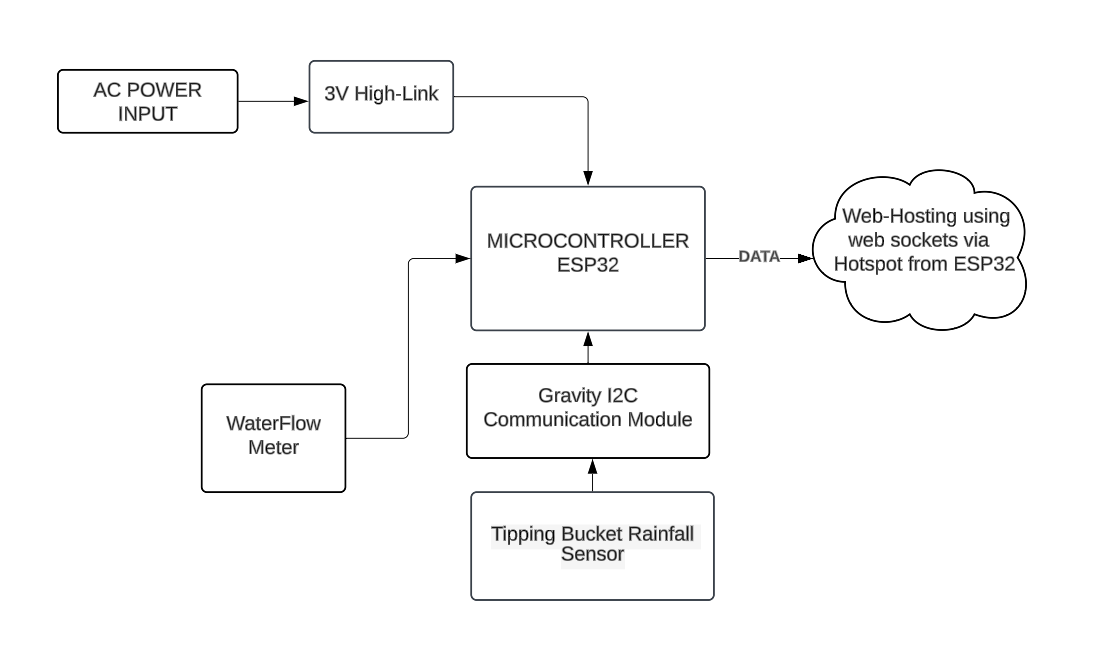
This project aims to design and develop a web-based platform that provides real-time data on rainfall patterns and water flow. By integrating sensors and microcontrollers with advanced web technologies, this platform enables users to monitor and manage rainwater harvesting systems effectively.

**PROBLEM STATEMENT:**

Design a webpage that aims to provide real-time data on rainfall datapatterns and water flow for efficient monitoring and management of rainwater harvesting practices.

**BLOCK DIAGRAM:**

The block diagram represents the architecture and workflow of the Rainwater Monitoring and Management System. It outlines the components involved and their interactions, providing a high-level overview of the project’s operation.

1. **Rainfall Intensity Sensor:**
   * Measures the intensity and duration of rainfall in real time.
   * Sends data to the ESP32 microcontroller for processing.
2. **ESP32 Microcontroller:**
   * Serves as the central processing unit for the system.
   * Collects data from sensors, processes it, and transmits it to web page hosted by the esp32.
3. **Water Flow Meter:**
   * Tracks the volume of water collected through the pipes from the terrace.
   * Sends pulses to esp32 for further calculations.

**SCHEMATIC DIAGRAM:**

The schematic diagram illustrates the connections between the components of this project.

1. ESP32 Microcontroller:

\* The following pins were used:

i. GPIO – 4 (Data pin for YGS-201)

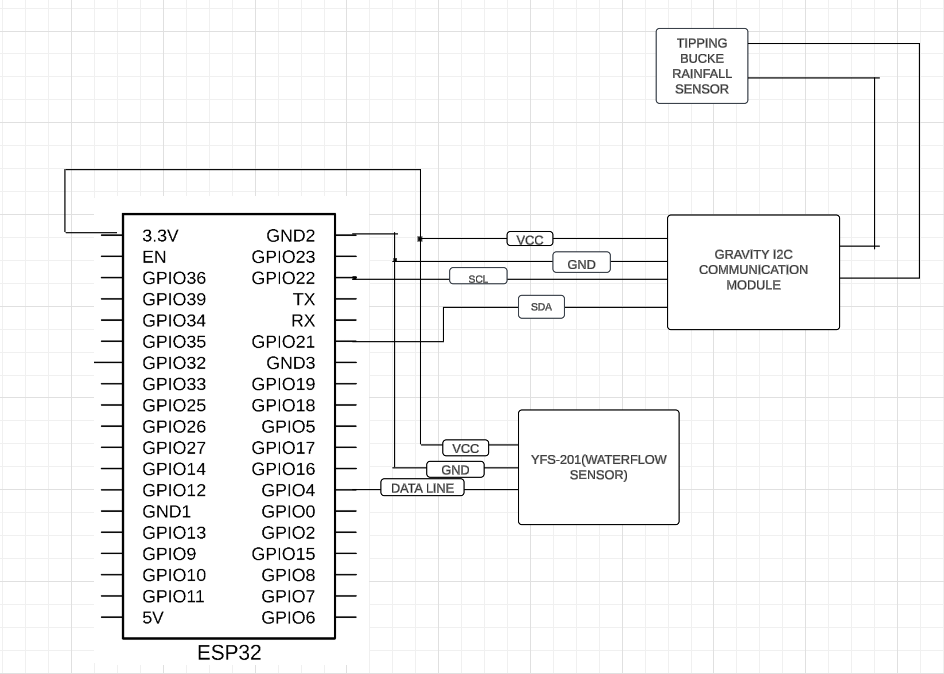
ii. GPIO – 21, 22 – For SDA, SCL Pins of the i2c Communication module.

iii. VCC and GND pins for YSF-201 and I2C Communication module.

1. Tipping Bucket Rainfall Sensor:
   * Measures the amount of rainfall using a tipping bucket mechanism.
   * Connected to the Gravity I2C Communication Module for translating sensor data into I2C signals.
2. Gravity I2C Communication Module:
   * Serves as an intermediary between the tipping bucket rainfall sensor and the ESP32.
   * Handles I2C communication using SDA (data) and SCL (clock) lines.
   * Ensures reliable data transmission from the rainfall sensor to the ESP32.
3. YFS-201 Water Flow Sensor:
   * Tracks the water flow rate through a pipe and sends signals to the ESP32.

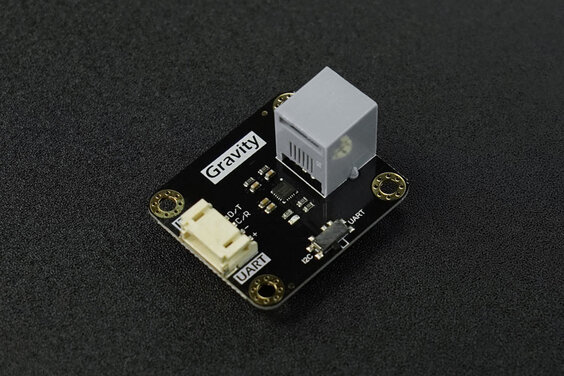
\* Data Line: Transmits flow rate data to the ESP32 for processing.

1. Power Connections:
   * The ESP32 supplies power to the sensors and modules through its 3.3V and 5V pins.



**Hardware Details**

1. **Tipping Bucket Rainfall Sensor (SKU: SEN0575)**
   * **Introduction:**  
     This sensor measures rainfall based on the principle of a tipping bucket mechanism. Rainwater accumulates in a bucket, and when it reaches a certain level, the bucket tips, generating a signal that corresponds to the rainfall measurement. This design ensures stability and sensitivity without using electronic components inside the sensor.
   * **Key Features:**
     + **Rainwater Drainage:** The hollow bottom design allows automatic drainage, ensuring smooth operation and accurate readings.
     + **High Compatibility:** Supports I2C and UART interfaces, making it compatible with multiple platforms like Arduino, ESP32, micro:bit, and Raspberry Pi.
     + **Applications:** Ideal for weather stations, environmental monitoring systems, and smart farming solutions.
     + **Robust Design:** Operates reliably across a wide temperature range of -40°C to 85°C.
   * **Specifications:**
     + **Working Voltage:** 3.3V–5.5V DC
     + **Current Consumption:** <3mA
     + **Resolution:** 0.28mm rainfall per tip
     + **Output Signals:** I2C or UART



1. **ESP32 Microcontroller: Detailed Overview**

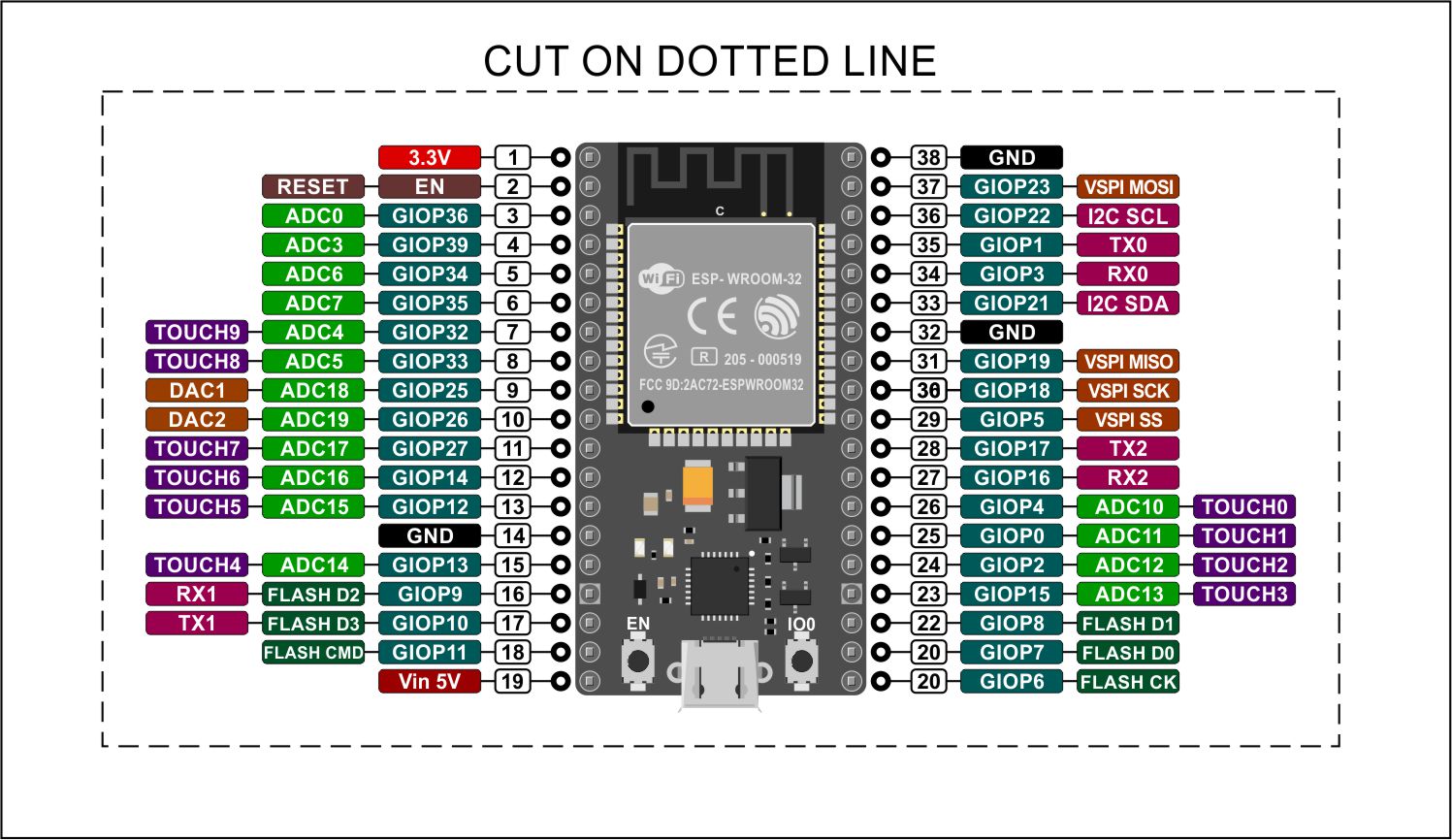
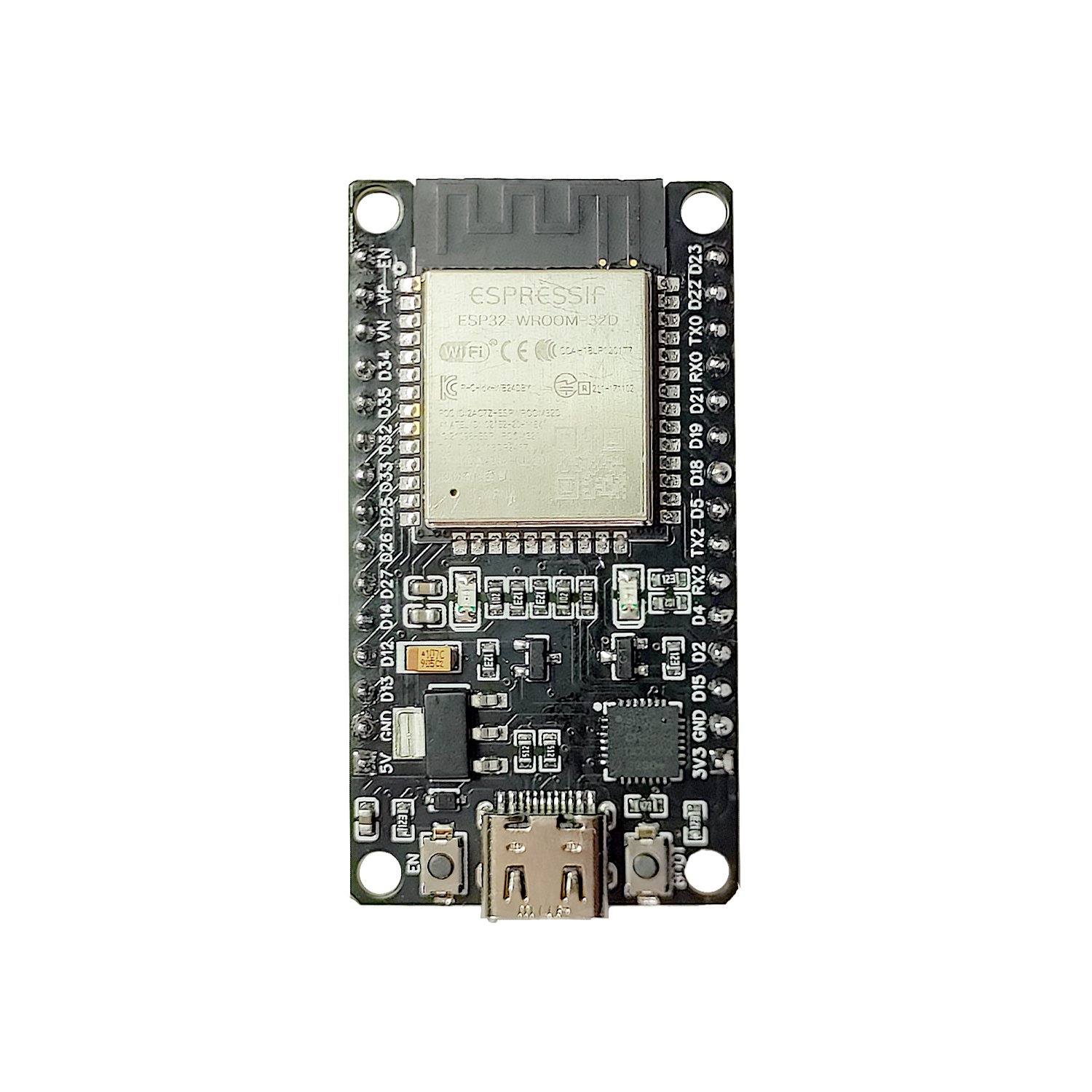
The **ESP32** is a powerful microcontroller developed by **Espressif Systems**. Known for its versatility and robust features, it is widely used in IoT (Internet of Things) applications. In this rainfall monitoring project, the ESP32 serves as the primary controller, managing sensor inputs, processing data, and facilitating communication with external systems.

**Key Features of ESP32**

1. **High Processing Power:**
   * Powered by a dual-core Tensilica LX6 processor operating at up to 240 MHz, providing ample computational resources for real-time data processing and sensor management.
2. **Connectivity:**
   * **Wi-Fi:** Integrated 2.4 GHz Wi-Fi module for seamless internet connectivity, enabling real-time data transmission to servers or cloud platforms.
   * **Bluetooth (BLE):** Supports Bluetooth Low Energy for wireless peripheral communication, allowing future expandability, such as pairing with mobile apps.
3. **GPIO Pins:**
   * The ESP32 has 36 GPIO (General Purpose Input/Output) pins, which can be configured as digital inputs or outputs. Some pins also support analog input, PWM output, and communication protocols (like I2C, SPI, UART).
   * In this project:
     + **I2C Communication:** Used to interface with the Gravity I2C module and read data from the tipping bucket rainfall sensor.
     + **GPIO Pins:** Used to connect to the YFS-201 water flow sensor's output signal line.
4. **Memory:**
   * **RAM:** 520 KB of SRAM, ensuring smooth execution of tasks.
   * **Flash Storage:** Comes with built-in flash storage (typically 4 MB or higher) for storing firmware and configuration files.
5. **Analog-to-Digital Converter (ADC):**
   * Features a 12-bit ADC for reading analog sensor data. While not directly used in this project, it supports additional functionalities like monitoring battery voltage or analog sensors.
6. **Power Management:**
   * Operates at a supply voltage of **3.3V** or **5V**, making it compatible with most sensors and modules.
   * Offers deep sleep and low-power modes, which can be utilized in future iterations to conserve energy during inactive periods.
7. **Versatile Communication Protocols:**
   * **I2C:** Interfaces with multiple sensors (used for the rainfall sensor in this project).
   * **UART:** Communication with modules or external devices.

**Role of ESP32 in the Rainfall Monitoring System**

1. **Data Acquisition:**
   * The ESP32 collects rainfall data from the **tipping bucket rainfall sensor** via the **Gravity I2C Communication Module**.
   * Simultaneously, it reads water flow rate data from the **YFS-201 water flow sensor** through a GPIO pin.
2. **Data Processing:**
   * Converts raw sensor readings into meaningful metrics, such as rainfall in millimeters and water flow volume in liters.
   * Applies scaling factors and calibrations to ensure accuracy.
3. **Web Server Capability:**
   * The ESP32 can also host a lightweight web server, allowing users to directly access rainfall and water flow data from their devices within the same network.



1. **YFS-201 Water Flow Sensor**

* Measures the water flow rate and provides data on water collected and used for rainwater harvesting systems.
* Sends data in form of pulses.

1. **Hi-Link 3V AC to DC Converter:**

* The **Hi-Link 3V AC to DC Converter** is a compact and efficient module designed to convert AC mains voltage (typically 110V or 220V AC) to a stable 3.3V DC output. It is commonly used in low-power electronics projects where a reliable power source is required for microcontrollers, sensors, or other components that operate on low DC voltage.



Software requirements include HTML, CSS, JavaScript to run the webpage, micropython to control the ESP32 and the working of the project.

**Working Procedure:**

**Hardware Workflow:**

1. **Rainwater Measurement (Tipping Bucket Sensor):**
   * The tipping bucket sensor collects rainwater and triggers a pulse whenever a specific amount of water tips the bucket.
   * The Gravity I2C Communication Module interfaces with the sensor and transmits rainfall data (in millimeters) to the ESP32 through the I2C protocol.
   * The ESP32 reads the data via its SDA and SCL pins connected to the module.
2. **Water Flow Monitoring (YFS-201 Sensor):**
   * The YFS-201 sensor measures the water flow rate of harvested rainwater using a built-in Hall effect sensor that outputs pulses proportional to the flow rate.
   * The sensor's data line is connected to an ESP32 GPIO pin, where the ESP32 counts pulses and calculates the flow rate or total water volume based on pre-defined calibration constants.
3. **Data Processing (ESP32):**
   * The ESP32 processes rainfall and water flow data in real-time, using algorithms in the uploaded code to calculate:
     + Total rainfall.
     + Water flow rate and total water collected.
   * It dynamically stores and updates this data.
4. **Data Transmission and Display:**
   * The ESP32, with its built-in Wi-Fi module, transmits the processed data to a web socket
   * The data is displayed in real-time on a webpage or application for monitoring and analysis.
5. **Power Management:**
   * The Hi-Link AC-DC converter powers the entire circuit by stepping down the AC mains supply to 3V DC, ensuring safe operation of all components

**Code Workflow:**

1. **Initialization:**
   * The ESP32 initializes GPIO pins for sensors and configures I2C communication.
   * It sets up the Wi-Fi module for data transmission.
2. **Rainfall Data Acquisition:**
   * The code periodically queries the tipping bucket sensor via I2C to retrieve rainfall values.
   * Each pulse recorded by the sensor corresponds to a fixed volume of rainwater, which is converted to millimeters of rainfall.
3. **Water Flow Data Acquisition:**
   * The ESP32 reads pulse signals from the YFS-201 sensor.
   * Using the predefined calibration factor, it converts pulses into the flow rate (liters per minute) or cumulative volume (liters).
4. **Data Processing and Storage:**
   * The rainfall and flow rate data are processed into human-readable formats.
   * Aggregated data, like total rainfall and total water collected, is computed and stored.
5. **Real-Time Data Transmission:**
   * Processed data is displayed directly on a webpage via Wi-Fi.
   * The ESP32 hosts a local server, allowing users to access data through a browser.
6. **Dynamic Webpage Update:**
   * The code generates an HTML page that displays data dynamically without reloading.
   * JavaScript on the page requests data updates from the ESP32 server, ensuring real-time monitoring.

**Results**

The **Rainwater Monitoring and Management System** successfully integrates hardware and software to provide accurate, real-time data on rainfall and water flow. The key results are:

1. **Real-Time Data Monitoring:**
   * The system continuously measures rainfall in millimeters and water flow in liters per minute.
   * The data is processed on the ESP32 microcontroller and displayed on a dynamic web interface, providing users with instant insights into rainfall and water collection.
2. **Data Accuracy:**
   * The tipping bucket rainfall sensor and YFS-201 water flow sensor provide accurate measurements, with calibration factors allowing precise conversion of pulse data into rainfall and flow rate values.
3. **Remote Access:**
   * Using the ESP32’s Wi-Fi capabilities, the system makes real-time data accessible from any device connected to the local network or even remotely if integrated with cloud services.
4. **Storage and Data Logging:**
   * The system logs rainfall and flow rate data for up to 10 readings, ensuring that users can track trends over time. It also includes features to reset readings.
5. **User-Friendly Interface:**
   * The web interface provides users with clear visualizations of key metrics, such as total rainfall, flow rate, and collected water volume. Charts and graphs offer an intuitive way to understand water collection patterns.

**Conclusion**

The **Rainwater Monitoring and Management System**, by combining accurate hardware with an efficient software platform, the system provides reliable data for better water management. The use of affordable sensors, coupled with the ESP32’s processing power and Wi-Fi capabilities, makes it a scalable and effective solution for smart water management.

Key conclusions include:

* The system is capable of accurately measuring rainfall and monitoring water flow in real-time.
* It offers a user-friendly web interface that simplifies the monitoring and analysis of rainwater data.
* The system’s scalability allows for future expansion, such as remote access through the cloud, predictive analytics, or integration with other smart home systems.
* It successfully integrates various hardware components and software technologies, creating a robust and efficient rainwater harvesting management solution.

**Future Scope**

The **Rainwater Monitoring and Management System** has significant potential for enhancement and can be expanded in several ways:

1. **Cloud Integration for Remote Access:** Implement cloud-based storage and data visualization (using platforms like AWS or Google Cloud). This will allow users to access the data remotely and analyze trends from anywhere, not just within the local network.
2. **Predictive Analytics:** Integrate machine learning algorithms to predict rainfall patterns and water collection trends based on historical data.
3. **Mobile Application:**  Develop a mobile application for real-time monitoring, notifications, and alerts to provide users with convenient access to the system on the go, allowing them to receive alerts for high rainfall or low water levels.
4. **Automated Water Flow Control:**  Integrate automated control systems that adjust water flow and storage based on sensor readings to reduce human intervention, making the system more autonomous.
5. **Power Efficiency Improvements:**
   * **Future Scope:** Implement low-power modes or solar power solutions to make it suitable for off-grid locations.

