

# Rain Water Storage and Monitoring

Using MicroPython

# Problem Statement

Design an application that aims to provide real-time data on rainfall patterns and water flow for efficient monitoring and management of rainwater harvesting practices.

# Project Requirements

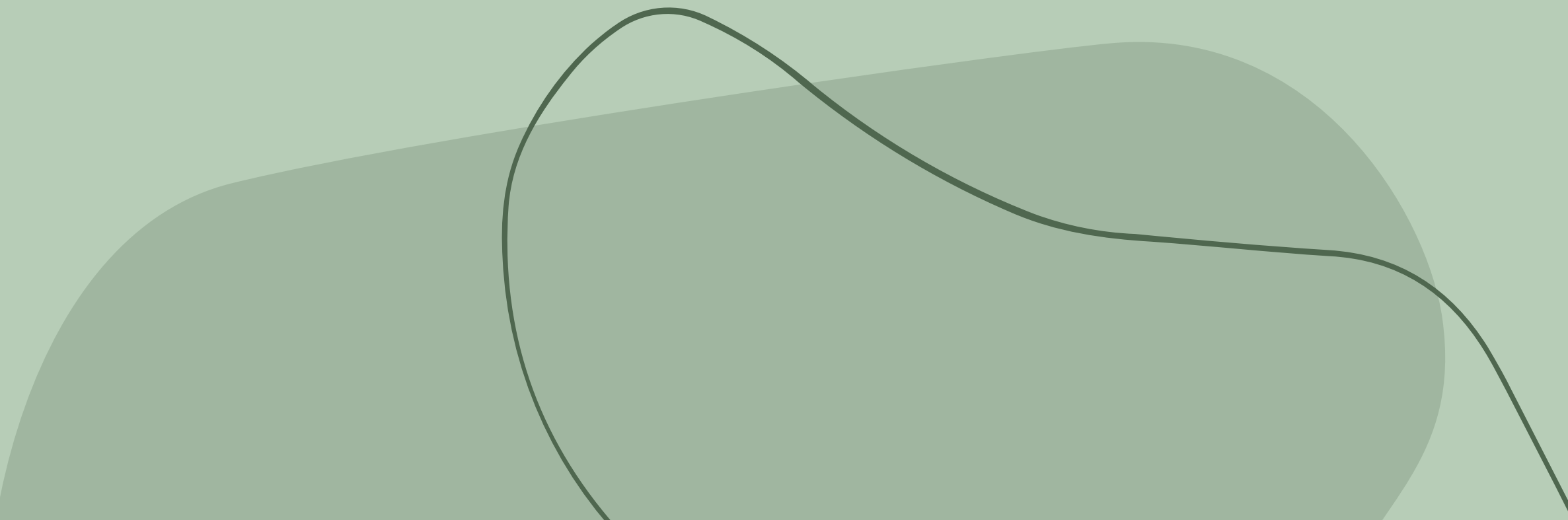
1. Calculating Amount of Rainwater Accumulated and wasted

2. Duration of Rainfall.

3. Amount of Water Stored in Ground Pit

4. Web Application

To collect, monitor and display the data from the sensors using micro-python



# Formula Used

# Formula Used

Amount of Water = Rainfall Intensity X Area of Terrace X Duration of Rainfall

## Amount of Water

The total volume of water fallen on the terrace (in liters or any other appropriate unit)

## Area of Terrace

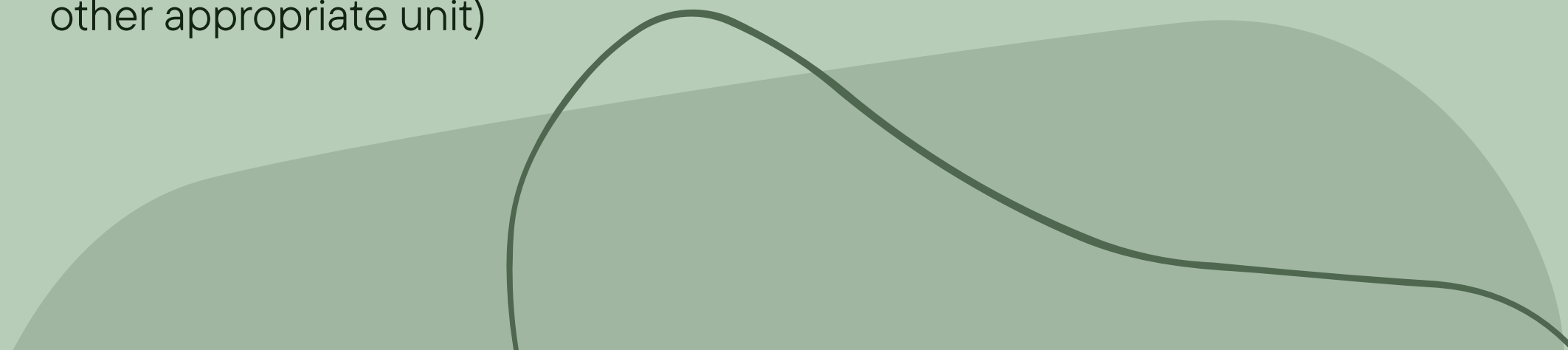
the total area of the terrace (in square meters or any other appropriate unit)

## Rainfall Intensity

the rate of rainfall (in millimeters per hour or any other appropriate unit)

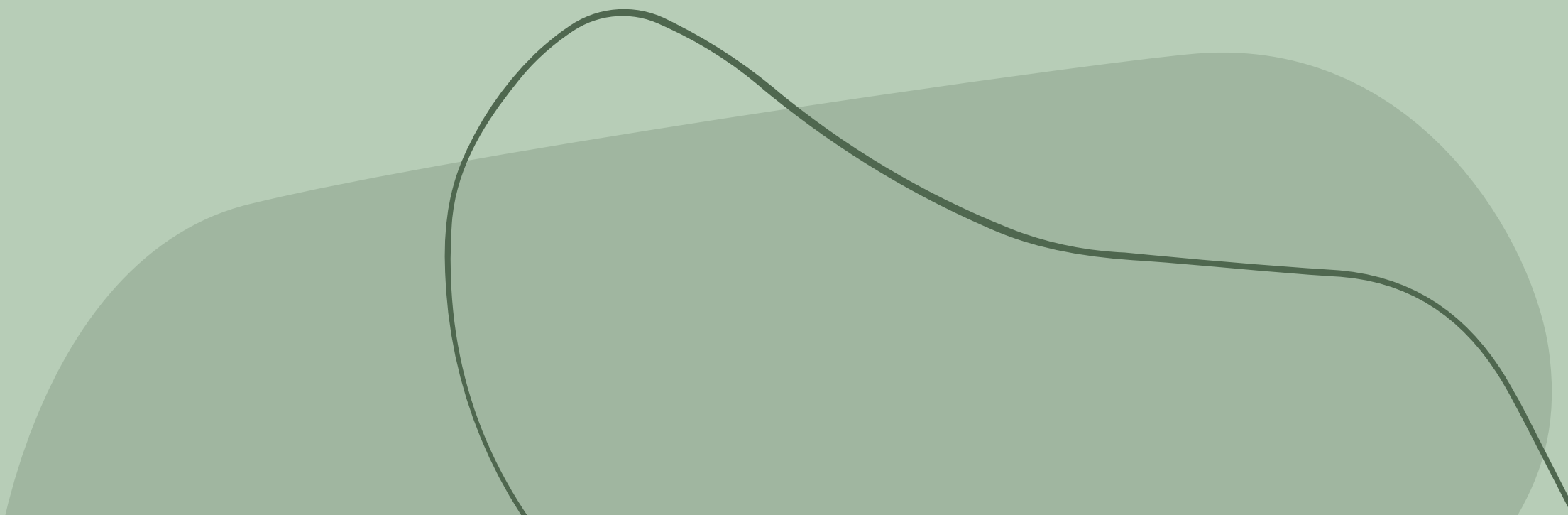
## Duration of Rain Fall

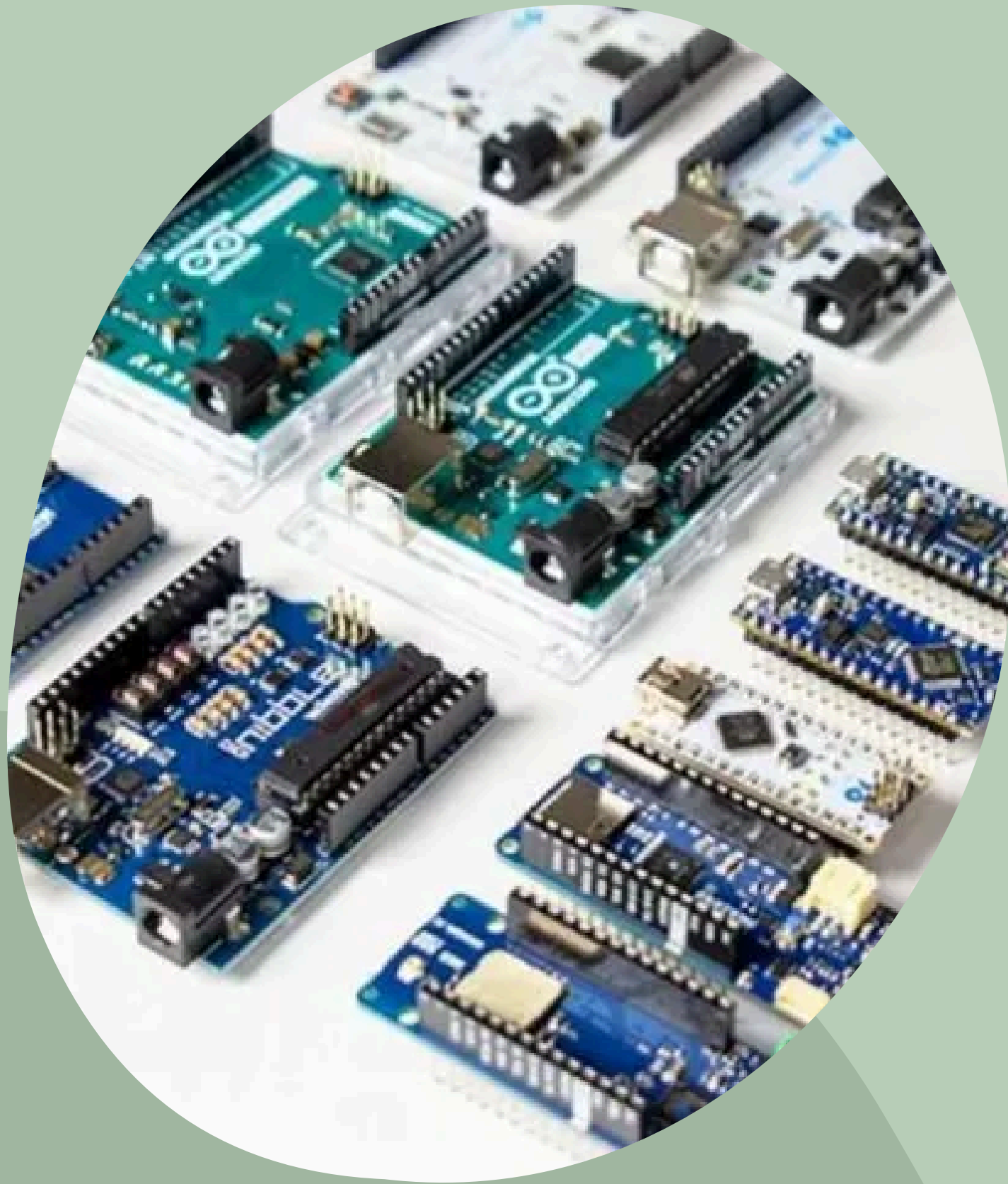
In Minutes



Amount of Water = Rainfall Intensity X Area of  
Terrace X Duration of Rainfall

This formula assumes that the rainfall intensity remains constant over the duration of the rainfall. If the rainfall intensity varies, you may need to use more advanced techniques, such as integration or numerical methods, to get a more accurate result





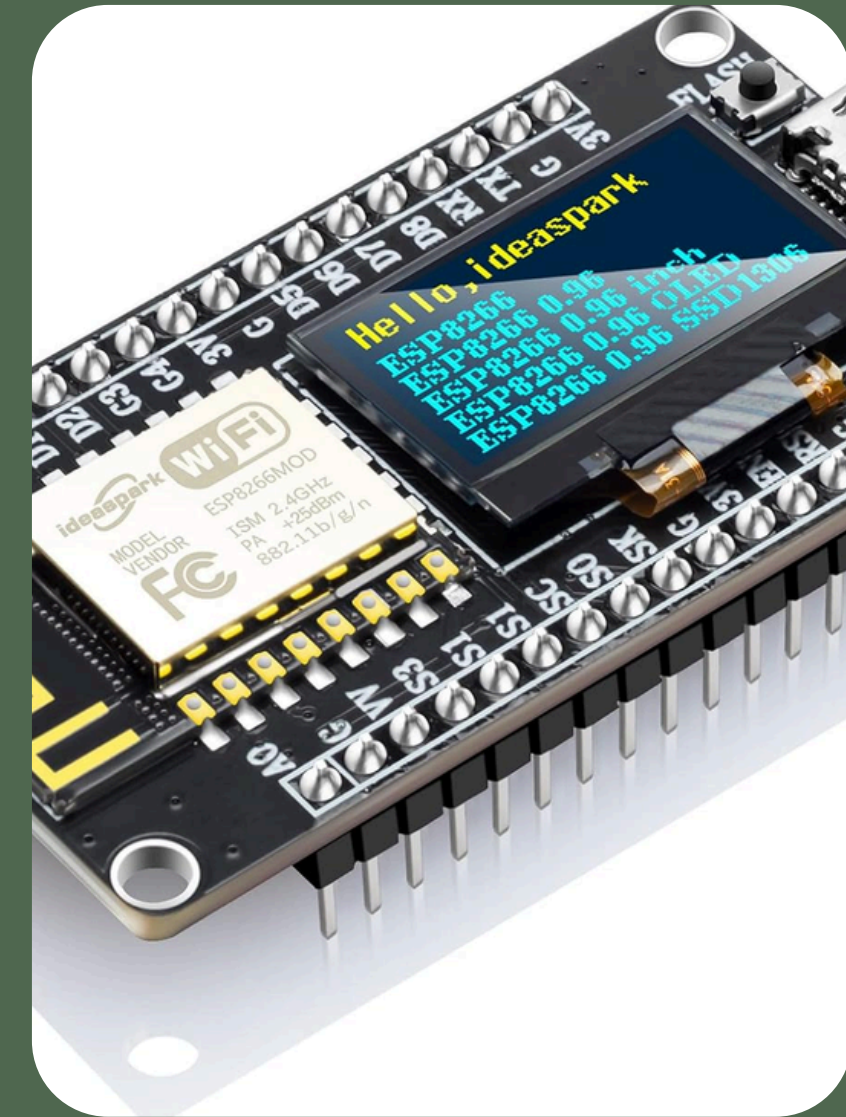
# Hardware Requirements

# Sensors and Micro Controllers



Tipping Bucket Rain  
Gauge

an instrument used by meteorologists and hydrologists to gather and measure the amount of liquid precipitation over a predefined area, over a period of time



Arduino or  
ESP8266

The ESP8266 is a low-cost Wi-Fi microcontroller, with built-in TCP/IP networking software, and microcontroller capability



# Sensors and Micro Controllers



## Water Flow Meter

An integrated magnetic hall effect sensor that outputs an electrical pulse with every revolution. The hall effect sensor is sealed from the water pipe and allows the sensor to stay safe and dry

# Software Requirements



# Software Requirements



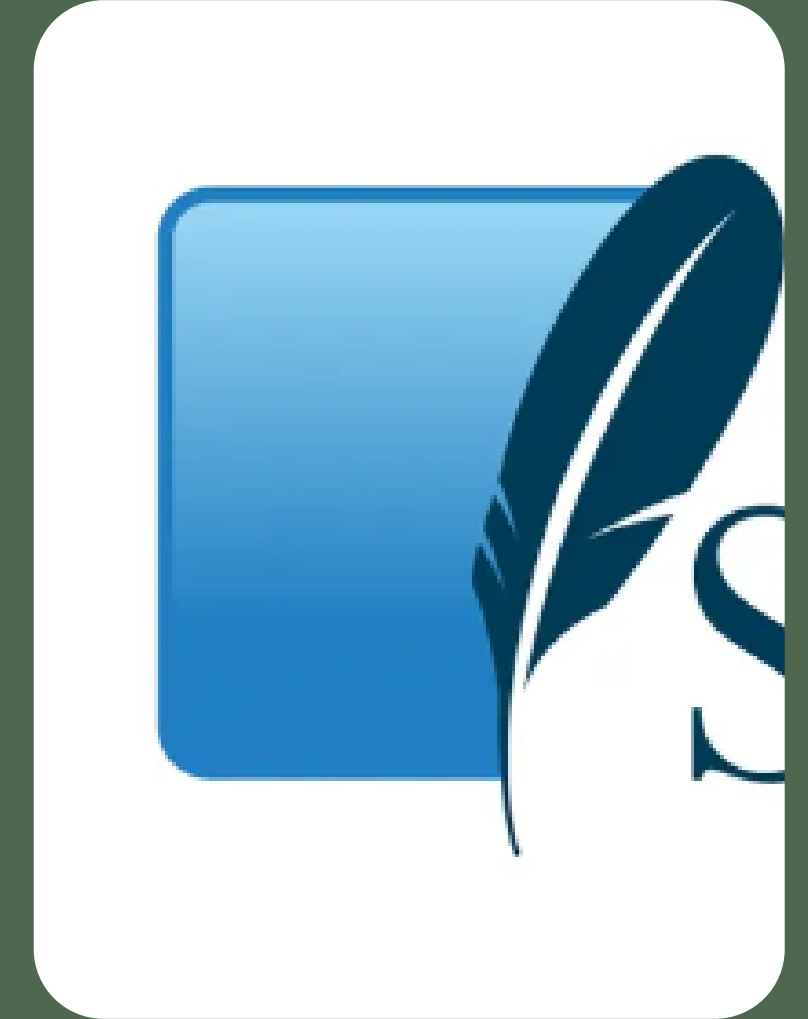
## Micro-Python

Micro-Python , frameworks working with the help of API's is used to code the arduino to perform required tasks



## HTML & CSS

HTML AND CSS are used to design the UI of the Web application



## SQLite3

Sqlite3 module is used to manage the databases using python

# Sending of data from Rain gauge to Web Application

# Process

## Read data from Rain Gauge Sensor

Use a microcontroller such as Arduino to interface with the tipping bucket rain gauge and read the pulses generated by the tipping mechanism. Each pulse corresponds to a certain amount of rainfall

## Process the Data

Convert the pulses received from the rain gauge into rainfall measurements, typically in units like millimeters or inches.

## Connecting to a Network

Use a networking module such as Wi-Fi, Ethernet, or GSM to connect the microcontroller to the internet.

## Sending data to Server

Set up a server to receive the rainfall data from the microcontroller. You can use various protocols such as HTTP, MQTT, or TCP/IP to send the data to the server.



# Process

## Storing Data on Server

Once the data is received by the server, you can store it in a database, process it further, or perform any necessary computations. This data can then be accessed or displayed as needed.

## Access data on Web Application

Develop a web application that can retrieve the rainfall data from the server. The app can communicate with the server using APIs or other methods to fetch the latest rainfall measurements.

## Displaying Data

Present the rainfall data in a user-friendly format within the web application, such as a graph showing rainfall over time or a simple text display of the current rainfall measurement.

## Receive Notifications

Implement push notifications or other notification mechanisms in the web application to alert users of significant rainfall events or other relevant information.



# Thank you!

We are ready for any  
suggestions!!