*Smart Water Management*

*Phase 2 IoT Report*

**Customer Needs**

My customer name is mr.x, he is a farmer, his farm is located in Pulyampatti village in Erode district, where temperature is around 85F to 105F. His land is located 15 km from bhavanisagar dam and he lives 5 km from the farm. As he is more than 55 years old, he is not able to go to the garden on time, so the trees and plants in his land are withering without water.

1. **Idea**

The pump set in his land is 7HP and his agricultural land is 4 acres, he is planting coconut trees in his garden, so he only needs to water the land once a week. This pump set can be monitored and controlled from his home, this can be done from his mobile.

1. **Block Diagram**

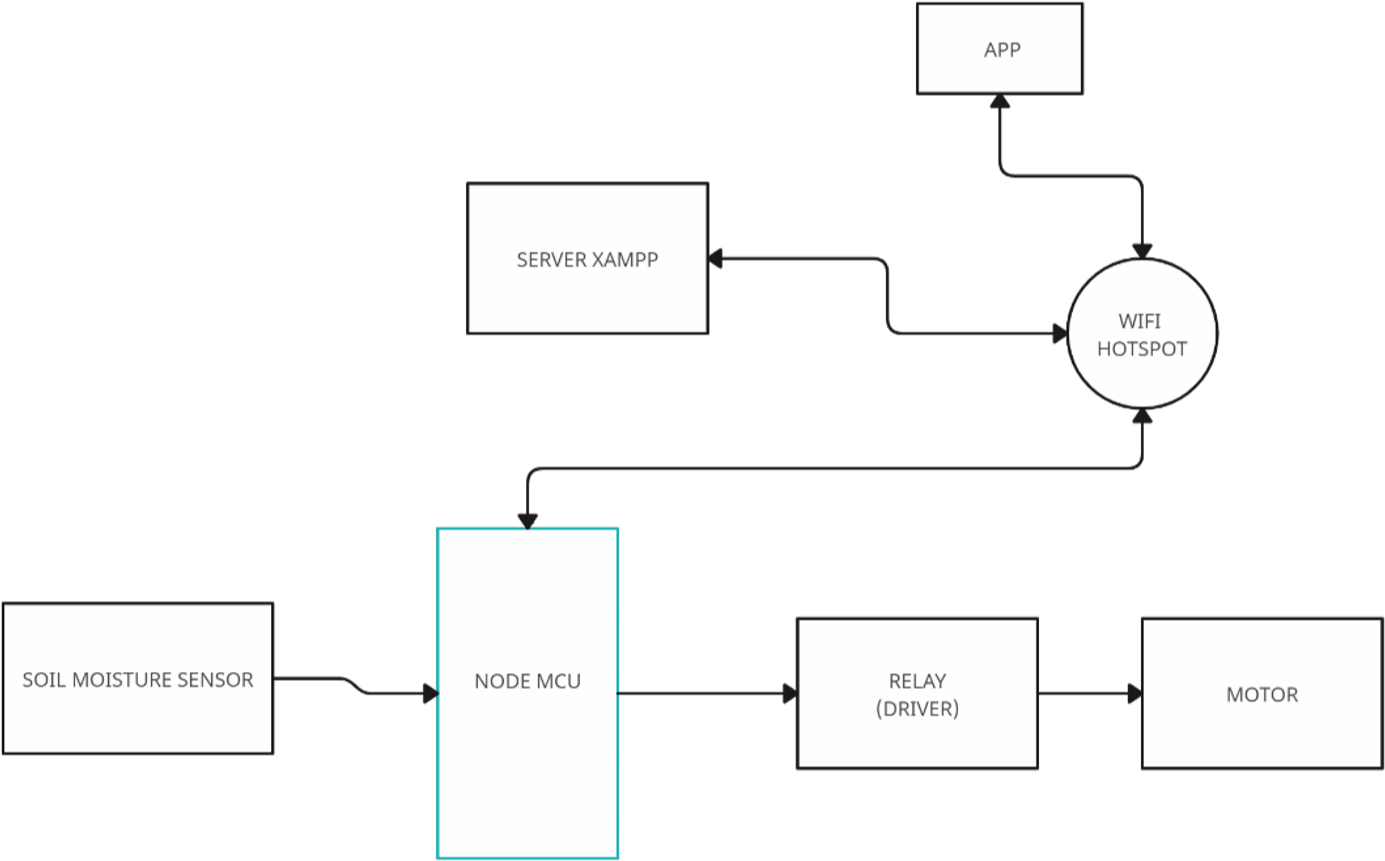
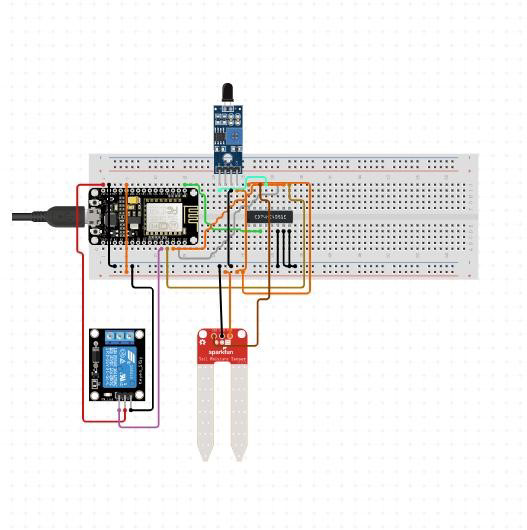


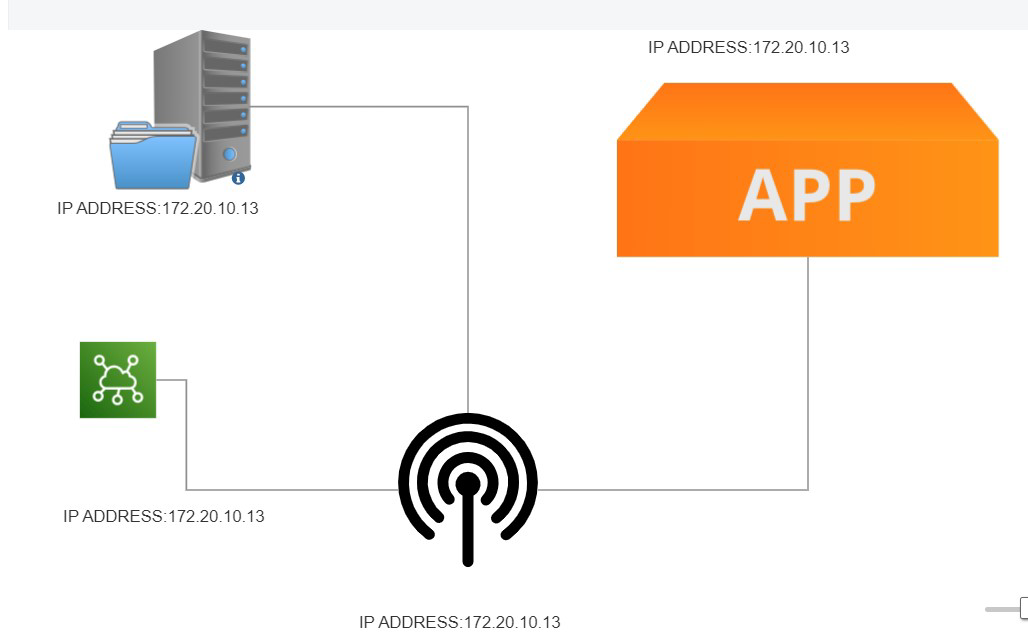
Fig: Block Diagram

**4. Circuit Diagram**



# Fig: Circuit Diagram

**5. Network Structure Diagram**



# Fig: Network Diagram

1. **Hardware Components Selection**

The components required for this problem are listed.

| **Si No** | **Description** | **Type** |  | **Qty** |
| --- | --- | --- | --- | --- |
| 1 | NodeMCU | V1.0E | 1 |  |
| 2 | Bread Board | Big Size | 1 |  |
| 3 | Moisture Sensor | Sensor | 1 |  |
| 4 | Relay module | connecting | 1 |  |
| 5 | Wires | - | 1 Set |  |
| 6 | Water pump | - | 1 |  |
| 7 | Laptop and internet | Intel(R) Core(TM) i3-10110U CPU @ 2.10GHz 2.59 GHz (8.00 GB RAM) and wifi or mobile data. | 1 |  |

1. **Hardware Specification**

| **Si No** | **Description** | **Specification** |
| --- | --- | --- |
| 1 | Laptop | Intel(R) Core(TM) i3-10110U CPU @ 2.10GHz 2.59  GHz |
| 2 | Wireless Router | WPA3-Personal.  ISM Band: 2.4GHz  Coverage speed : 72/72 (Mbps) |

**Table:** Manufactured with this hardware specification

1. **Software Selections**

| **Si No** | **Description** | **Minimum Requirement** |
| --- | --- | --- |
| 1 | Arduino IDE | Version : 1.8.0 |
| 2 | Windows OS | Windows 11 or Latest (64bit) |
| 3 | XAMMP | 8.0.28 / PHP 8.0.28 |
| 4 | Browser (Chrome, Firefox) | Latest |
| 5 | High Speed Internet Connection | 32 Mbps |

**Table:** Software Selection & Its Minimum Requirement

1. **Software Versions**

| **Si No** | **Description** | **Version** |
| --- | --- | --- |
| 1 | Arduino IDE | **Version :** 1.8.5 Windows 64bit |
| 2 | Windows OS | Windows 11 , 64bit, |
| 3 | XAMMP | 8.0.28 / PHP 8.0.28  XAMPP Control Panel v3.1.0.3.1.0  **Apache Status -** True  **MySQL -** True  **FileZilla –** Not Tested  **Mercury –** Not Tested  **Tomcat –** Not Tested |
| 4 | Browser (Chrome) | **Package Name:** Google Chrome  Version: Version 105.0.5195.102 (Official Build)  (x86\_64)  **Update:** Auto  **Test Mode:** Incognito |
| 5 | MIT App Inventor  (Cloud Platform) | **Built: April 24** 2023 Version: nb192  **Use Companion:** 2.66 or 2.66u  **Target Android SDK:** 31 (Android 12.0) |

1. Use Case Diagram

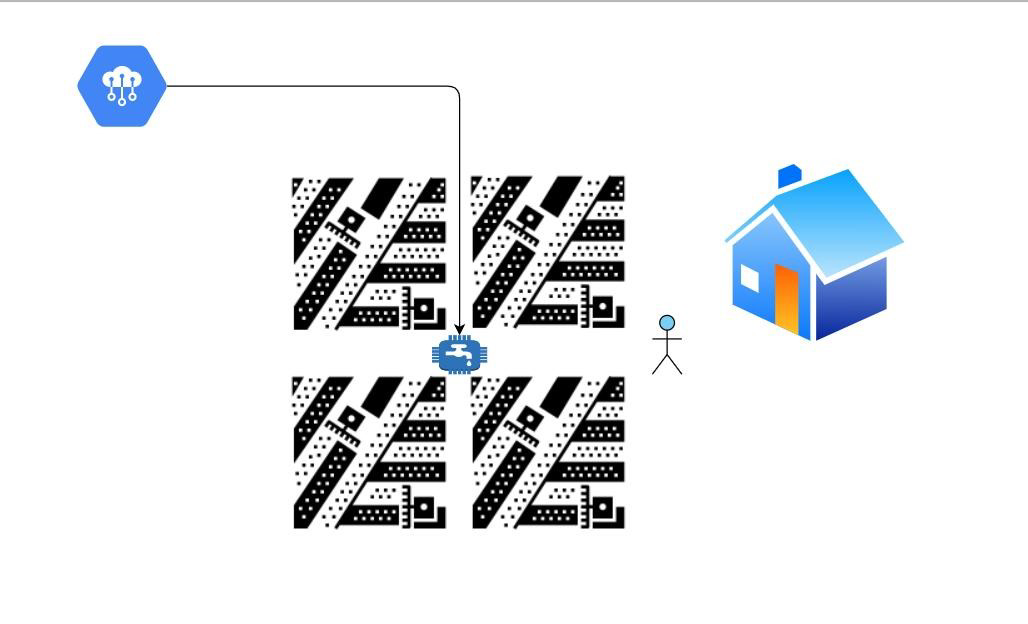
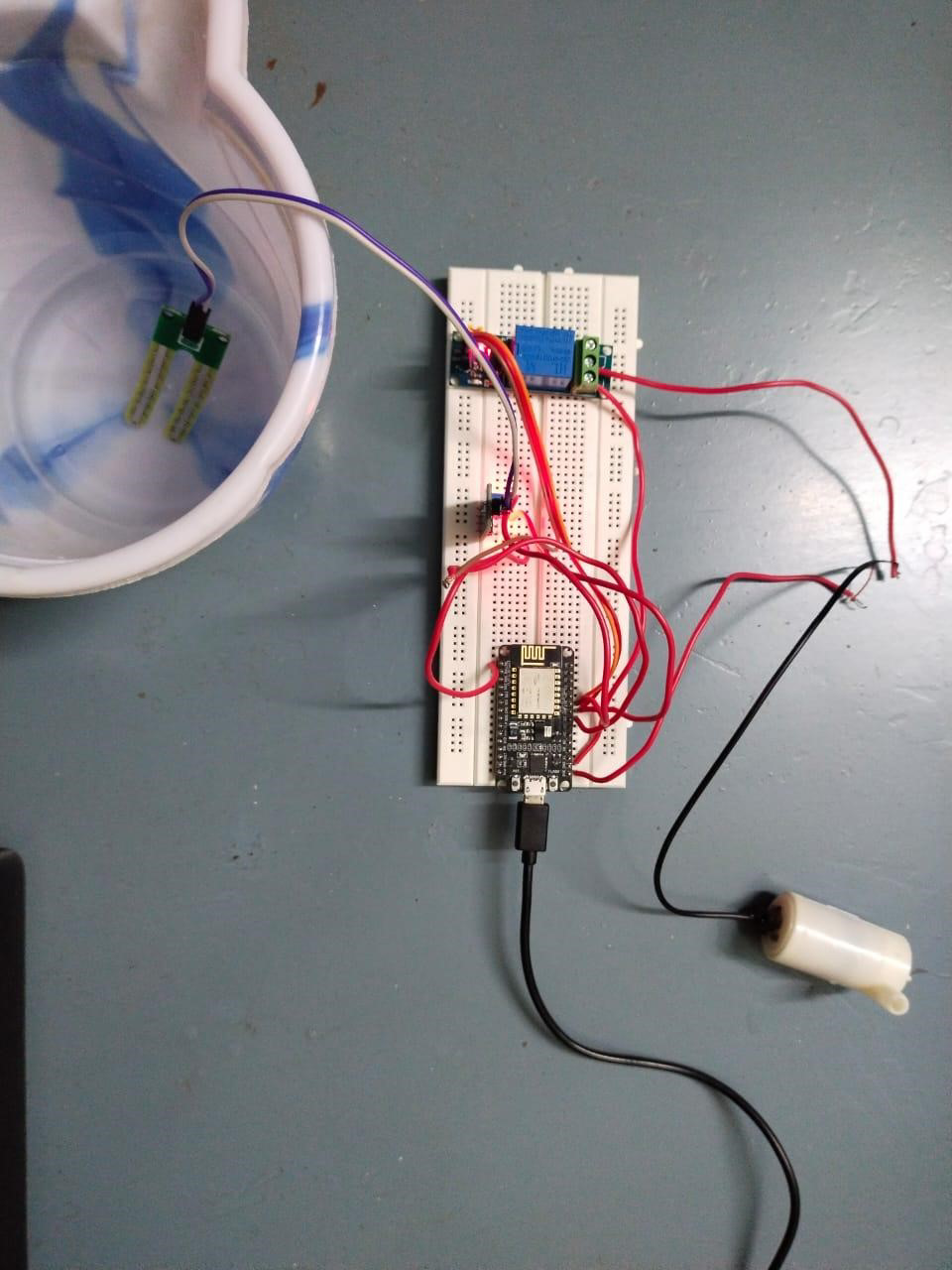


Fig: Use Case Diagram

**Model Working :**





**Implementation, Handle risk factors & Troubleshooting Implementation:**

**In the proposed Irrigation system IoT is implemented, in this system all the information that are received from the sensors and the various parameters are given to the arduinouno microcontroller as an analog input. A preset value of soil moisture sensor is fixed in microcontroller and also for fencing. When it goes beyond the particular threshold value water is automatically irrigated to the crops and once the required amount of water is fulfilled it stops. The Microcontroller transmits that information on the internet through a network of IoT in the form of wifi module ESP8266 that is attached to it.**

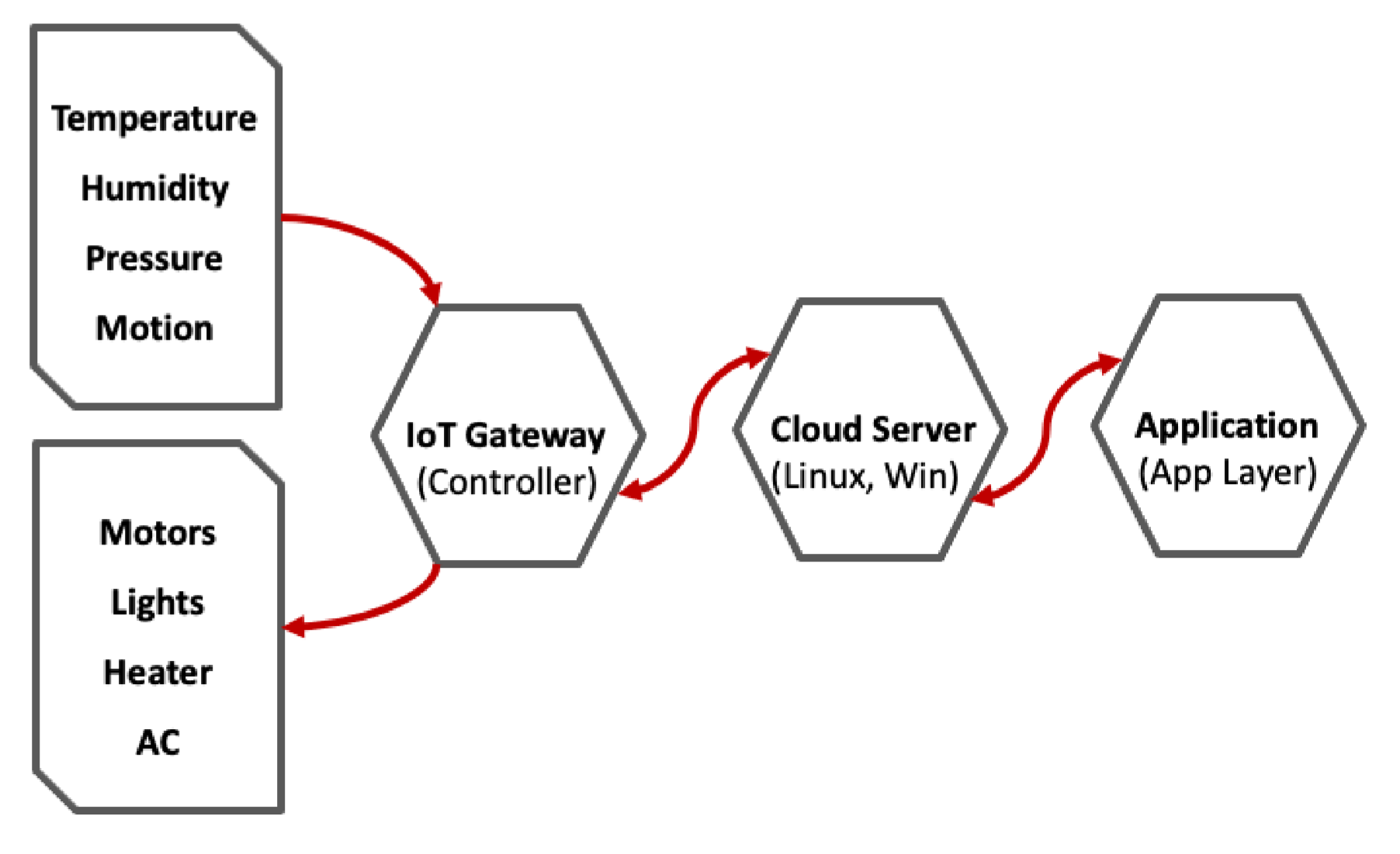
**Handle risk factors :**

**According to the experts, malicious actors can create a botnet of smart irrigation systems by infecting various types of Internet-connected devices with malware (e.g. routers, laptops, smartphones). The malware searches the local network for irrigation systems and takes control of them using various security flaws.**

**Troubleshooting :**

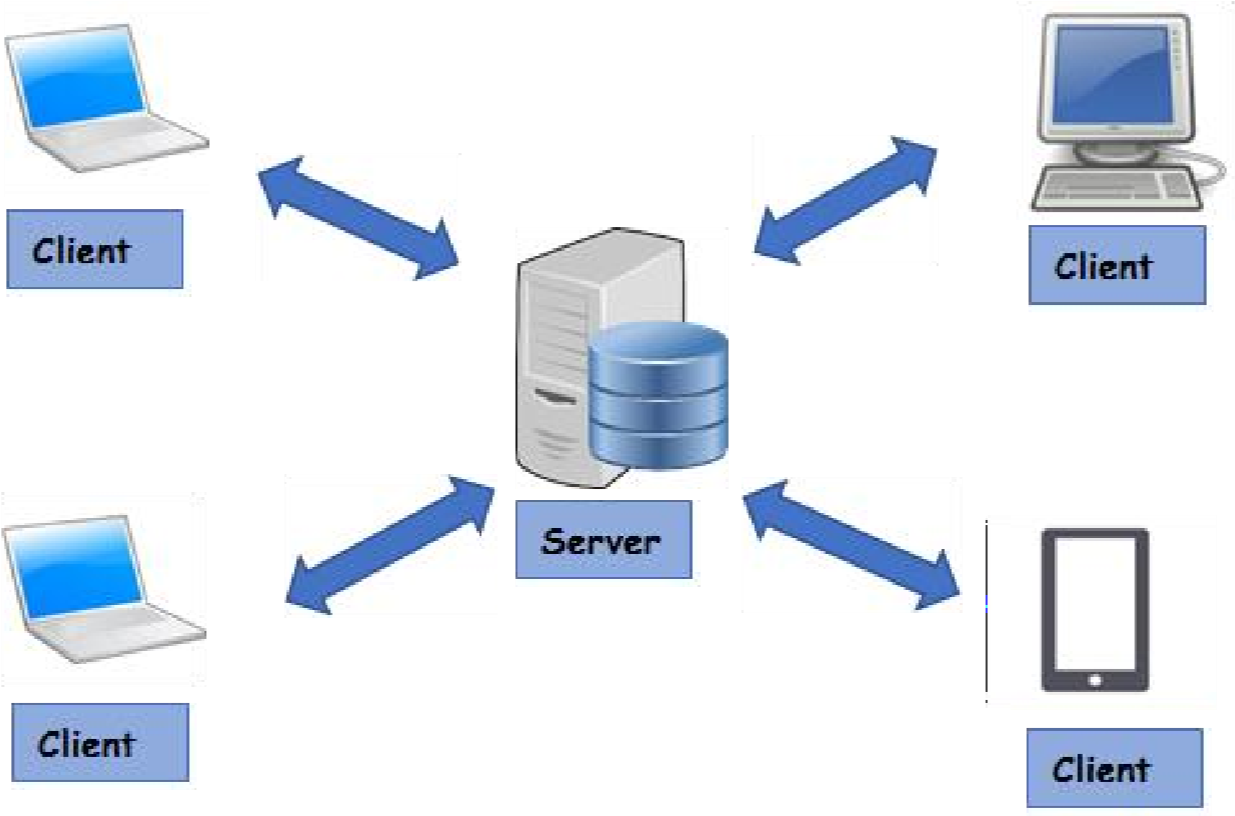
**Water can be managed using a few or a multitude of components designed to provide, measure, pressurize, restrict, divert, limit, and finally release it. To troubleshoot a water system you must know how each component is used within the water system, how they work, and why they were installed where they are—the main objective is to move water from point A to point B. One component installed incorrectly can affect the whole system, and, one small piece of debris in the wrong place can cause either unlimited flow of water resulting in damaged landscape, or no water flow at all ending in plant material death**

**IOT ARCHITECTURE**



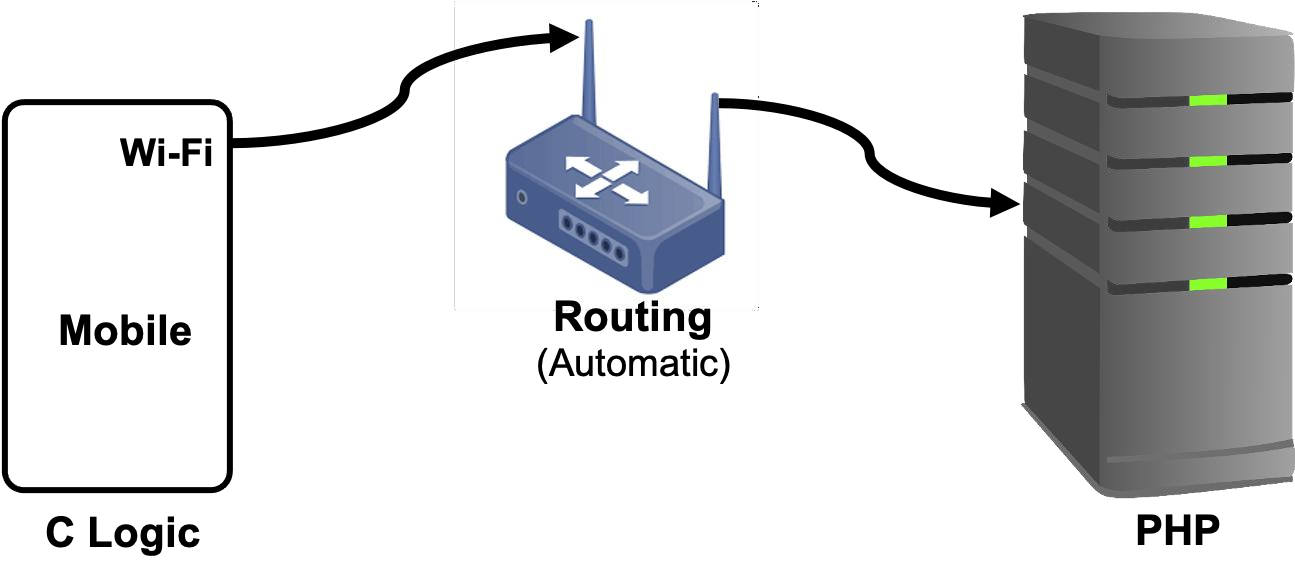
# Fig : IoT Architecture

**Server vs Client**



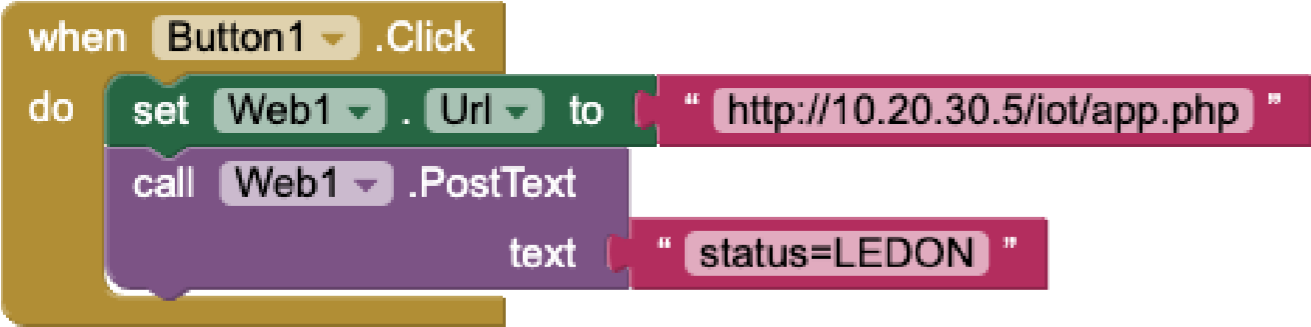
# Fig: request and response

**APPLICATION LAYER – APP Data Send To Server**



**Fig:** APP Data send to server Network Diagram

**Sending request to the server**



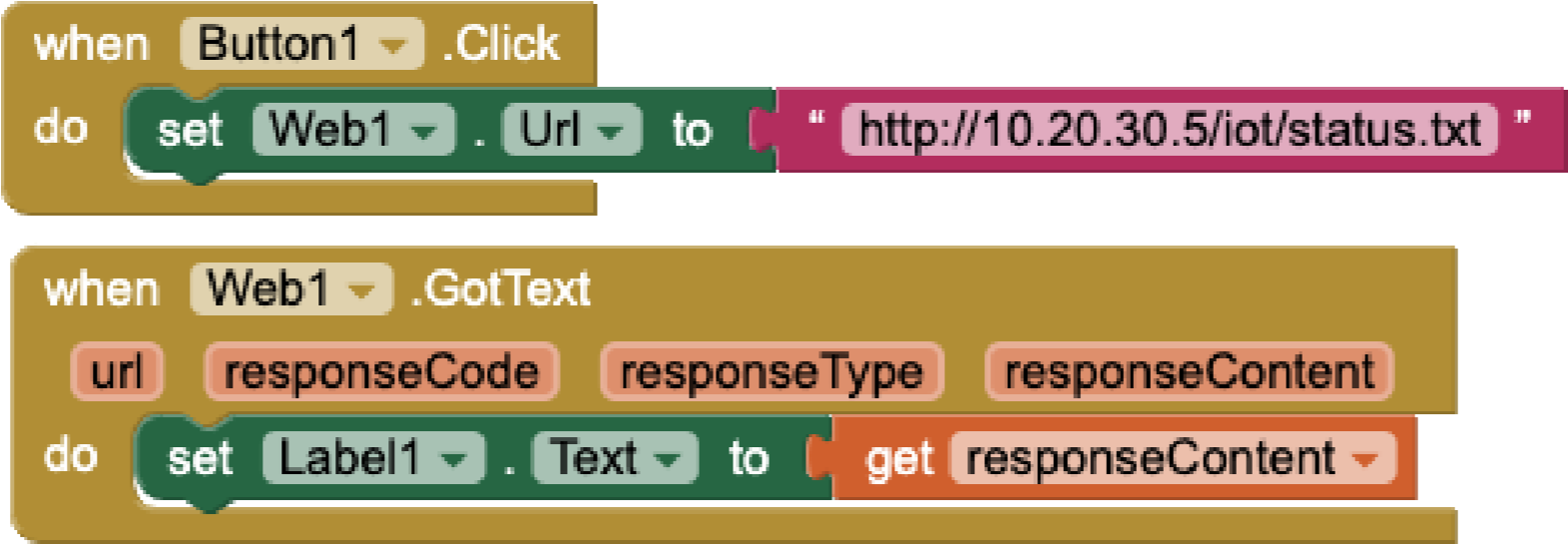
**Fig:** APP Data Send to Server (APP Blocks)

**PHP Code: (to send response)**

<?php

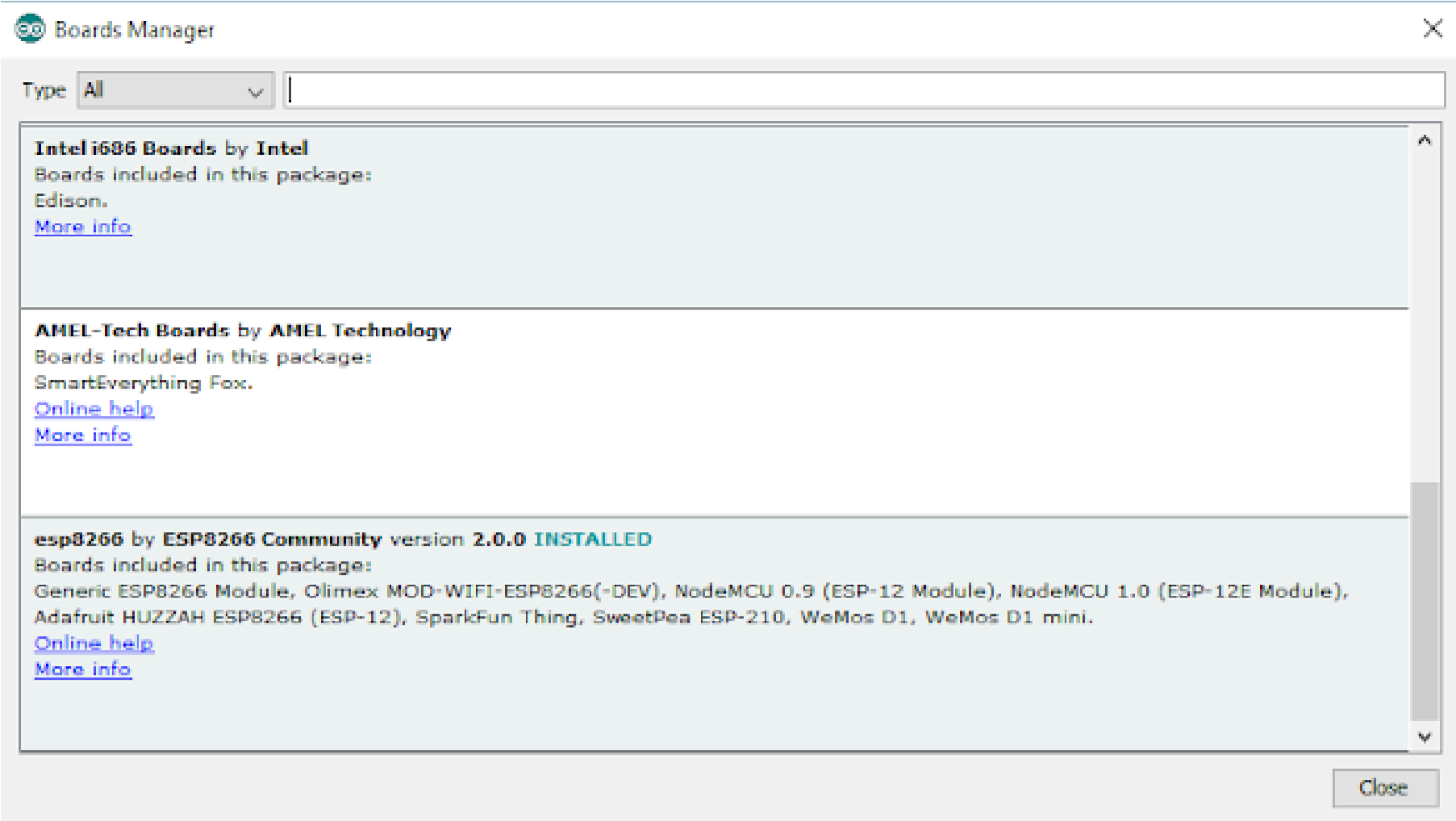
$status = $\_POST['status']; $name=fopen('status.txt','w'); fwrite($name, $status); fclose($name);

?>



**Fig:** APP Blocks(APP Data Fetch From Server)

**Working of arduino**



**Pin Connection**

| **NodeMCU PIN** | **LED** |
| --- | --- |
| D0 | Anode |
| GND | Cathode |

**Program for the project**

#include <ESP8266WiFi.h>

#include <FirebaseArduino.h>

#define WIFI\_SSID "your\_wifi\_ssid"

#define WIFI\_PASSWORD "your\_wifi\_password"

#define FIREBASE\_HOST "your\_firebase\_host"

#define FIREBASE\_AUTH "your\_firebase\_auth\_token"

#define SOIL\_MOISTURE\_PIN A0

#define WATER\_PUMP\_PIN D1

int soilMoistureThreshold = 500; // Adjust this value according to your soil moisture sensor

void setup() {

Serial.begin(9600);

pinMode(WATER\_PUMP\_PIN, OUTPUT);

digitalWrite(WATER\_PUMP\_PIN, LOW);

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Connecting to WiFi...");

}

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH);

}

void loop() {

int soilMoistureLevel = analogRead(SOIL\_MOISTURE\_PIN);

if (soilMoistureLevel < soilMoistureThreshold) {

digitalWrite(WATER\_PUMP\_PIN, HIGH); // Turn on the water pump

delay(5000); // Water for 5 seconds

digitalWrite(WATER\_PUMP\_PIN, LOW); // Turn off the water pump

Firebase.setInt("soilMoistureLevel", soilMoistureLevel);

Firebase.setInt("waterPumpStatus", 1);

} else {

digitalWrite(WATER\_PUMP\_PIN, LOW); // Turn off the water pump

Firebase.setInt("soilMoistureLevel", soilMoistureLevel);

Firebase.setInt("waterPumpStatus", 0);

}

delay(60000); // Wait for a minute before checking again

}