GOVERNMENT COLLEGE OF ENGINEERING
ERODE

B.E Electronics and Communication Engineering

SMART PLANT MONITORING BY WATER MANAGEMENT SYSTEM

Name of the Students:	University Register no:
Team Leader:	
Subhasree S	731121106047
Team Members:	
Yashica S	731121106055
Kaviya T S	731121106025
Shobana C	

Under the mentor of

Dr.M.Poongothai

Department of Information Technology (IT)

Department of Electronics and Communication Engineering

Government College of Engineering

Erode, PO, near Vasavi College, TamilNadu-63831

Affiliated to Anna University, Chennai.

SMART PLANT MONITORING BY WATER MANAGEMENT

OBJECTIVES:

- The purpose of a smart plant monitoring system is to improve plant health and productivity. By collecting and analysing data in real time, farmers and gardeners can ensure that their plants are getting the optimal conditions they need to thrive. This can lead to increased yields, improved quality, and reduced costs.

- It provides the efficient water usage, real-time monitoring, remote access and customized settings.

- Energy consumption for pumps and other equipment used in irrigation to reduce operational costs.

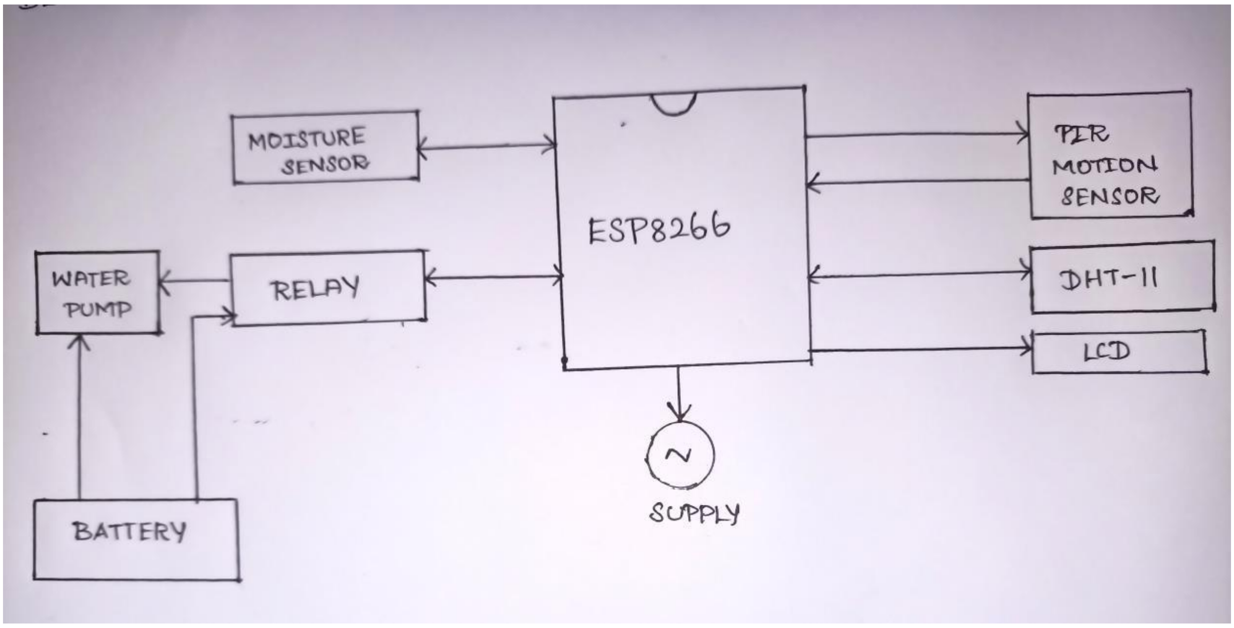
- Monitors the water quality in irrigation sources to prevent contamination and maintain crop health.

- Reduce water consumption and associated costs while conserving natural resources.

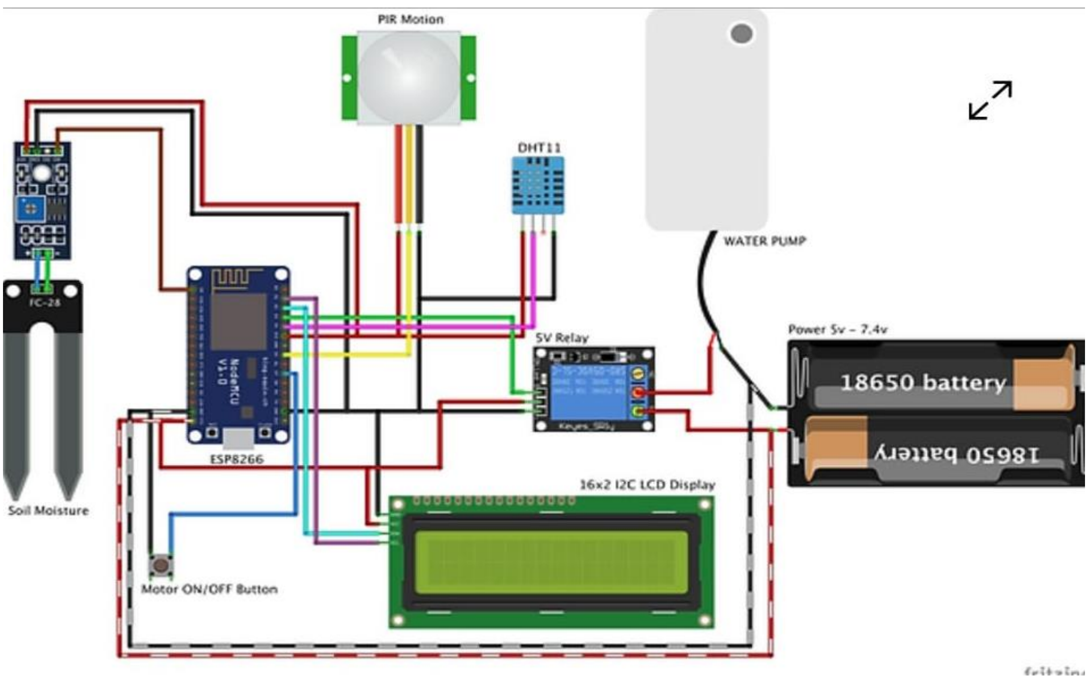
INTRODUCTION:

Smart plant monitoring by water management systems involves the use of advanced technologies to efficiently and effectively manage water resources for plant growth in various agricultural and horticultural settings. These systems integrate sensors and automation to optimize water usage, conserve resources, and enhance crop yield and quality. Here are some existing ideas and components commonly found in smart plant monitoring systems for water management.

BLOCK DIAGRAM:



CIRCUIT DIAGRAM:



COMPONENTS DESCRIPTION:

- Soil Moisture Sensor is helpful to measure the moisture content at different depths. They provide real-time data on soil moisture levels, enabling precise irrigation control.

- Passive Infrared (PIR) can help to make the sensor more immune to temperature, humidity and noise. PIR sensors are commonly utilized in security alarms. They pick up on the heat signature of the warm bodies, such as humans or plants that are warmer than their surroundings.

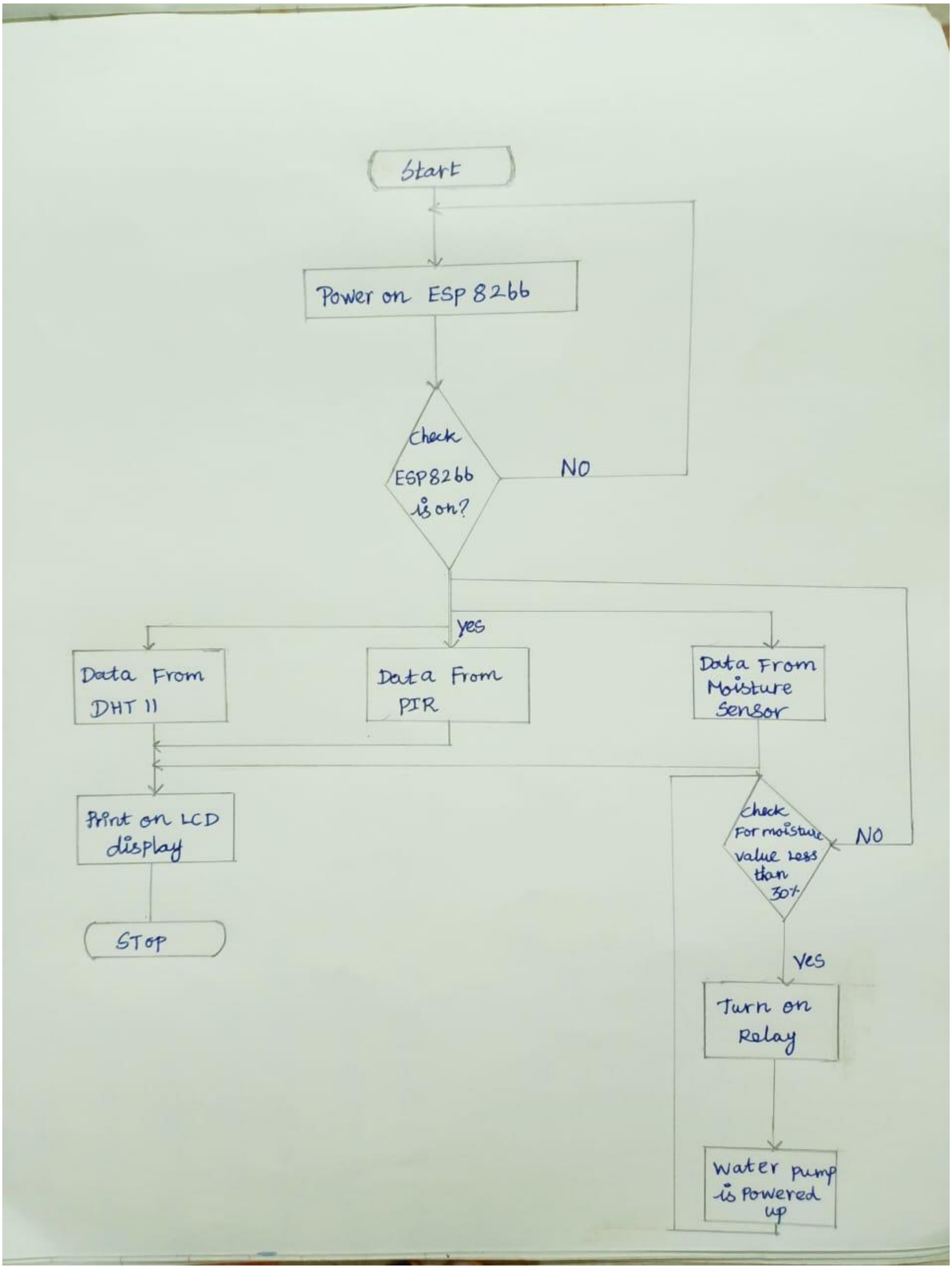
- The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor. LCD display sensor is user-friendly, enabling users to monitor and control the system easily while providing valuable insights into plant conditions and system performance.

- Incorporating an LCD (Liquid Crystal Display) into the smart plant monitoring and water management system can provide real-time data visualization and enhance user interaction.

- Data Presentation is used to connect the LCD display to the microcontroller (e.g., Arduino, ESP8266) or central control unit. Program the microcontroller to display essential information such as soil moisture levels, temperature, humidity, and watering status.

- System Status Display the overall status of water management system, indicating whether it's currently watering, in standby mode, or if there are any issues.

FLOW CHART:


Start
Power on ESP 8266
Check ESP8266 is on?
NO
yes
Data From DHT 11
Data From PIR
Data From Moisture Sensor
Print on LCD display
STOP
Check For moisture value less than 30%
NO
yes
Turn on Relay
Water pump is Powered up

IMPLEMENTATION:

- The Hardware implementation for plant monitoring by water management is done using Arduino board with the help of sensors which show the moisture content and someother information that the user may require.

- Using app development the hardware implementation is executed and it is notified by user via Wifi linked devices.

- The software tool determines the sensors working and alert whenever there is a case of emergency.

CODE IMPLEMENTATION:

```
 //Include the library files
#define BLYNK_TEMPLATE_ID "TMPL3RpinReNQ"
#define BLYNK_TEMPLATE_NAME "smart plant monitoring"
#define BLYNK_AUTH_TOKEN "n_U0uPZSqA2_tr9C-ZfWBK7HDn6c0zch"
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
//Initialize the LCD display
LiquidCrystal_I2C lcd(0x3F, 16, 2);
#define FIREBASE_HOST "https://water-iot-3-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH
"T0gvmm1NVBnVDR4kh6RQhTlcXPkE1q1IDlCijV05"
char auth[] = "n_U0uPZSqA2_tr9C-ZfWBK7HDn6c0zch"; //Enter your Blynk
Auth token
char ssid[] = "Redmi 9 Prime"; //Enter your WIFI SSID
char pass[] = "subi2423"; //Enter your WIFI Password
DHT dht(D4, DHT11);//(DHT sensor pin,sensor type) D4 DHT11 Temperature
Sensor
BlynkTimer timer;
//Define component pins
#define soil A0 //A0 Soil Moisture Sensor
#define PIR D5 //D5 PIR Motion Sensor
```

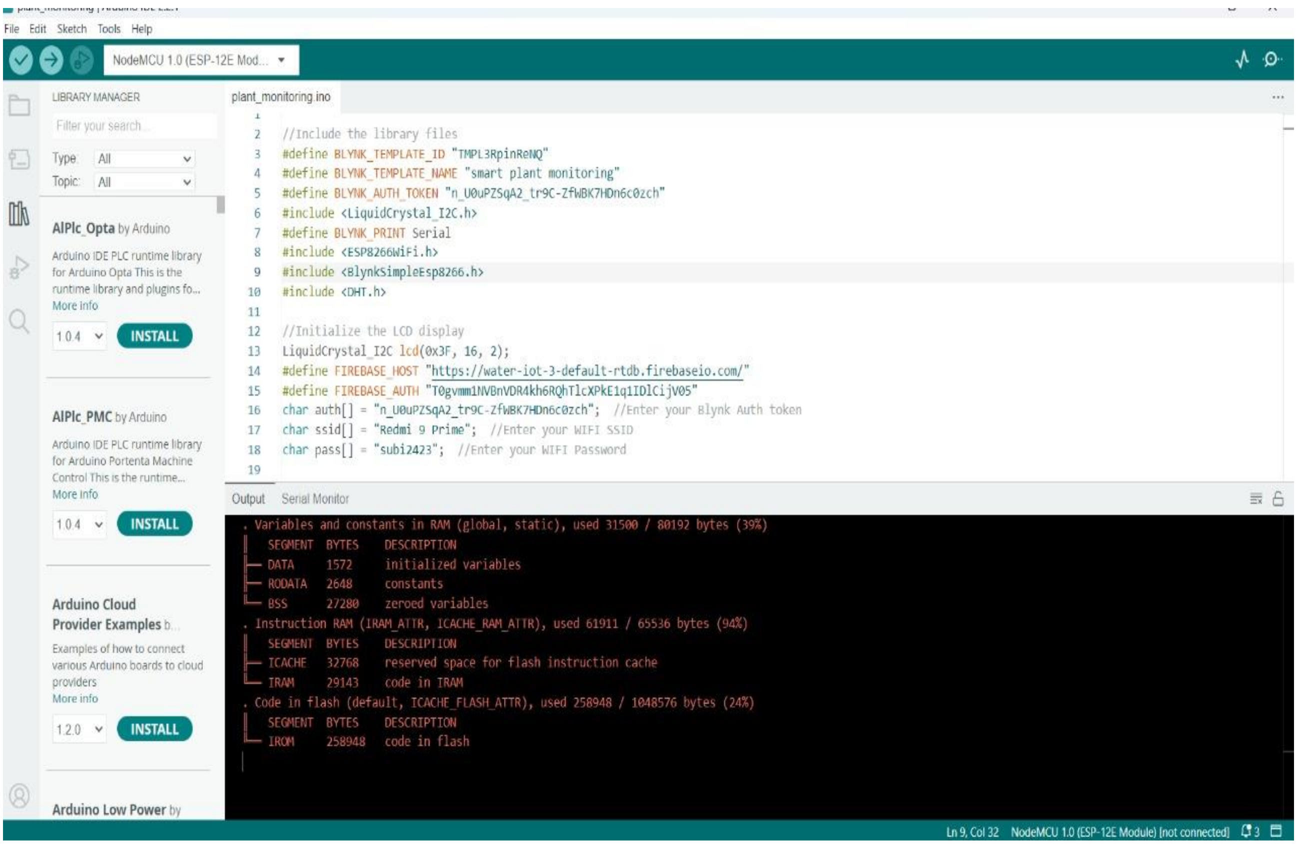
```
int PIR_ToggleValue;
void checkPhysicalButton();int relay1State = LOW;
int pushButton1State = HIGH;
#define RELAY_PIN_1 D3 //D3 Relay
#define PUSH_BUTTON_1 D7 //D7 Button
#define VPIN_BUTTON_1 V12
//Create three variables for pressure
double T, P;
char status;
void setup() {
Serial.begin(9600);
lcd.begin();
lcd.backlight();
pinMode(PIR, INPUT);
pinMode(RELAY_PIN_1, OUTPUT);
pinMode(PUSH_BUTTON_1, INPUT_PULLUP);
digitalWrite(RELAY_PIN_1, relay1State);
Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
dht.begin();
lcd.setCursor(0, 0);
lcd.print(" Initializing ");
for (int a = 5; a <= 10; a++) {
lcd.setCursor(a, 1);
lcd.print(".");
delay(500);
}
lcd.clear();
lcd.setCursor(11, 1);
lcd.print("W:OFF");
//Call the function
timer.setInterval(100L, soilMoistureSensor);
timer.setInterval(100L, DHT11sensor);
timer.setInterval(500L, checkPhysicalButton);
}//Get the DHT11 sensor values
void DHT11sensor() {
float h = dht.readHumidity();
float t = dht.readTemperature();
if (isnan(h) || isnan(t)) {
Serial.println("Failed to read from DHT sensor!");
return;
}
```

```
Blynk.virtualWrite(V0, t);
Blynk.virtualWrite(V1, h);
lcd.setCursor(0, 0);
lcd.print("T:");
lcd.print(t);
lcd.setCursor(8, 0);
lcd.print("H:");
lcd.print(h);
}
//Get the soil moisture values
void soilMoistureSensor() {
int value = analogRead(soil);
value = map(value, 0, 1024, 0, 100);
value = (value - 100) * -1;
if(value < 40){
digitalWrite(RELAY_PIN_1,LOW);
}
else if(value >= 80){
digitalWrite(RELAY_PIN_1,HIGH);
}
Blynk.virtualWrite(V3, value);
lcd.setCursor(0, 1);
lcd.print("S:");
lcd.print(value);
lcd.print(" ");
}//Get the PIR sensor values
void PIRsensor() {
bool value2 = digitalRead(PIR);
if (value2) {
Blynk.logEvent("PIRMOTION","WARNING! Motion Detected!"); //Enter
your Event Name
WidgetLED LED(V5);
LED.on();
} else {
WidgetLED LED(V5);
LED.off();
}
}
BLYNK_WRITE(V6)
{
PIR_ToggleValue = param.asInt();
```

```
}
BLYNK_CONNECTED() {
// Request the latest state from the server
Blynk.syncVirtual(VPIN_BUTTON_1);
}
BLYNK_WRITE(VPIN_BUTTON_1) {
relay1State = param.asInt();
digitalWrite(RELAY_PIN_1, relay1State);
}
void checkPhysicalButton()
{
if (digitalRead(PUSH_BUTTON_1) == LOW) {
// pushButton1State is used to avoid sequential toggles
if (pushButton1State != LOW) {
// Toggle Relay state
relay1State = !relay1State;
digitalWrite(RELAY_PIN_1, relay1State); // Update Button Widget
Blynk.virtualWrite(VPIN_BUTTON_1, relay1State);
}
pushButton1State = LOW;
} else {
pushButton1State = HIGH;
}
}
void loop() {
if (PIR_ToggleValue == 1)
{
lcd.setCursor(5, 1);
lcd.print("M:ON ");
PIRsensor();
}
else
{
lcd.setCursor(5, 1);
lcd.print("M:OFF");
WidgetLED LED(V5);
LED.off();
}
if (relay1State == HIGH)
{
lcd.setCursor(11, 1);
```

```
lcd.print("W:ON ");
}
else if (relay1State == LOW)
{
lcd.setCursor(11, 1);
lcd.print("W:OFF");
}
Blynk.run();//Run the Blynk library
timer.run();//Run the Blynk timer
}
```

OUTPUT:



WEB DEVELOPMENT:

The monitoring system is developed for transmission and reception of the information received from various data sources with the use of sensors integrated with nodemcuESP8266. The wireless sensing real-time data are transmitted into desired form across the network through internet connection. It is able to monitor concentration of soil moisture, temperature and relative humidity and stores the concentration values in the database.

Data Collection:

Establishing data collection systems to gather information from the sensors. This can involve IOT devices and stationary monitoring stations.

Data Processing:

Developing algorithms and software for data processing, water usage assurance, and real-time analysis to generate accurate need of water and moisture level assurance.

Communication:

Implementing methods to transmit data to a central server or database. This may include cellular networks, Wi-Fi or other wireless communication technologies.

User Interface:

Creating user-friendly interfaces, such as mobile apps or web platforms, to display water level information to the domestic users or relevant authorities.

Alerts and Warnings:

Integrating alert systems that notify users when water level becomes insufficient and also notifies whenever it exceeds, enabling them to take appropriate precautions.

Mobile Monitoring Units:

Developing mobile monitoring units that can be deployed to specific locations or events to assess monitoring levels in real time.

Mobile Apps and Wearables:

The development of mobile apps and wearable devices equipped with plant monitoring sensors provide individuals with real-time data on the water level needs for the plant monitoring.

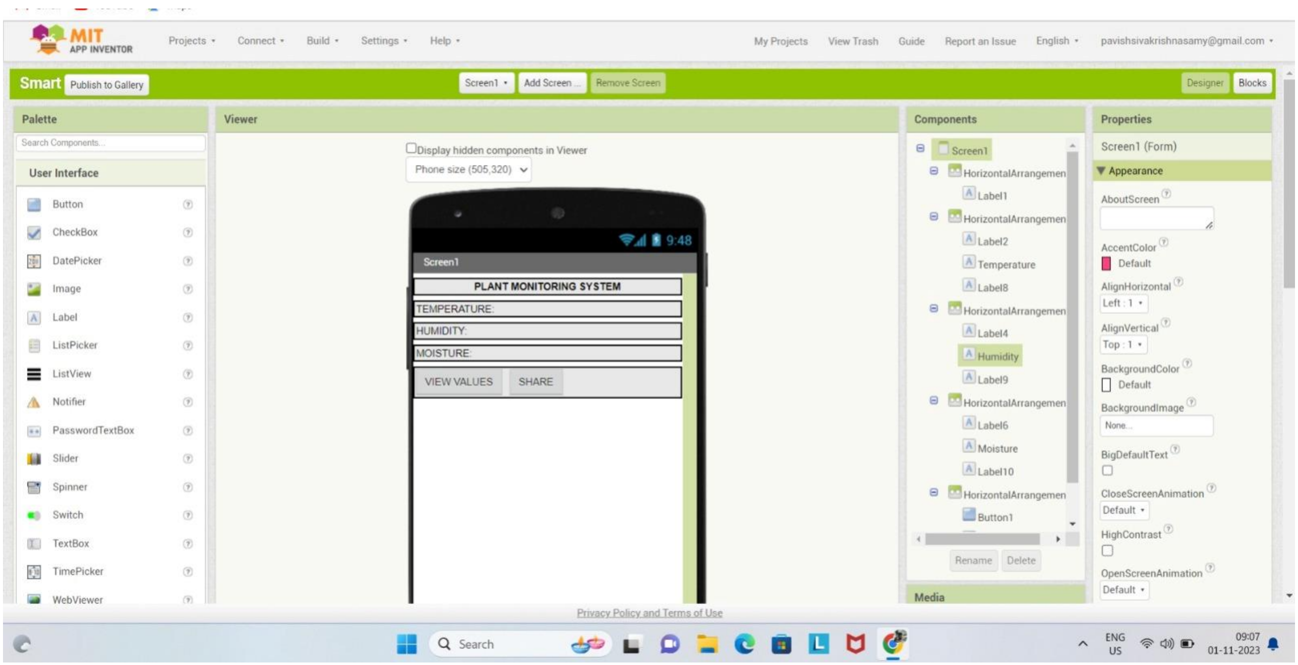
Historical Data Analysis:

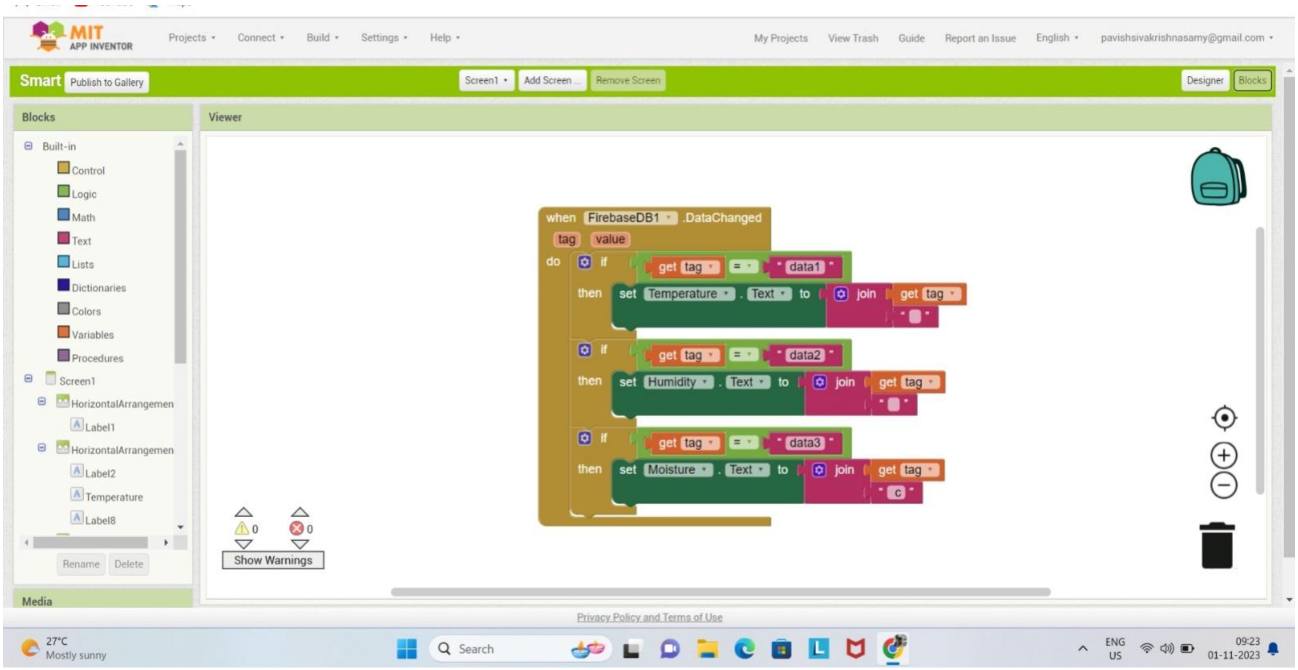
Historical water quality data can provide valuable insights into long-term trends and the effectiveness of past interventions. Analysing this data can inform future policy decisions and environmental planning.

CREATING SOFTWARE TOOL:

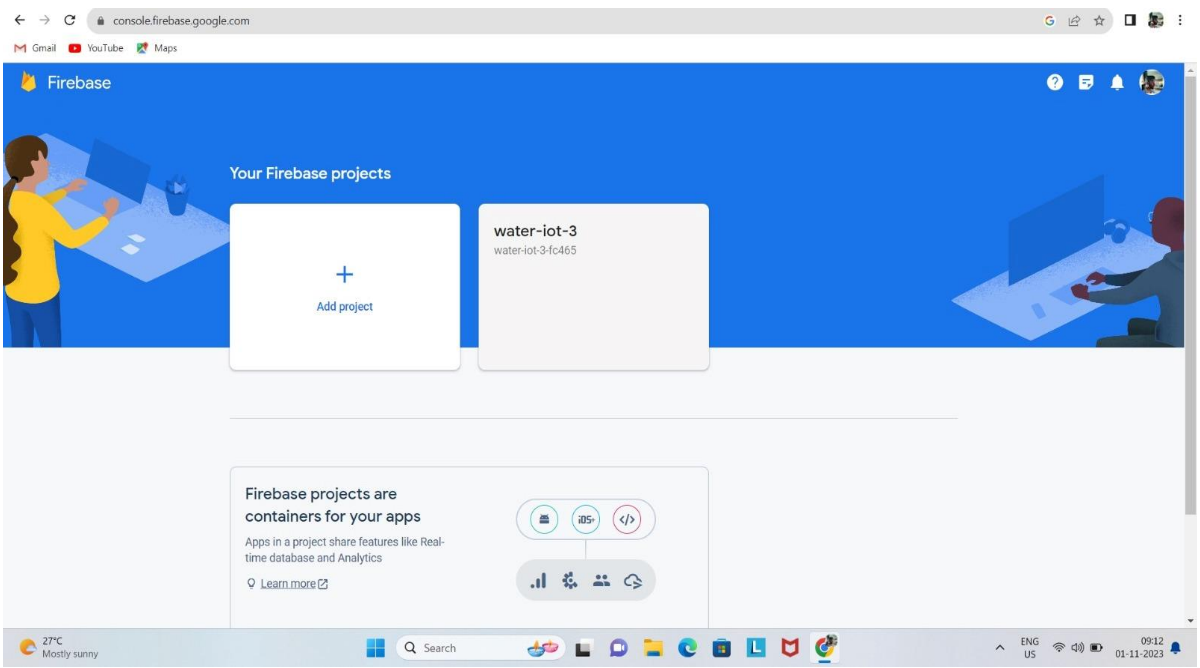
- Software application for plant monitoring by water management is created by MIT App Inventor.

- The required information can be implemented using blocks and logic tools in MIT App Inventor.



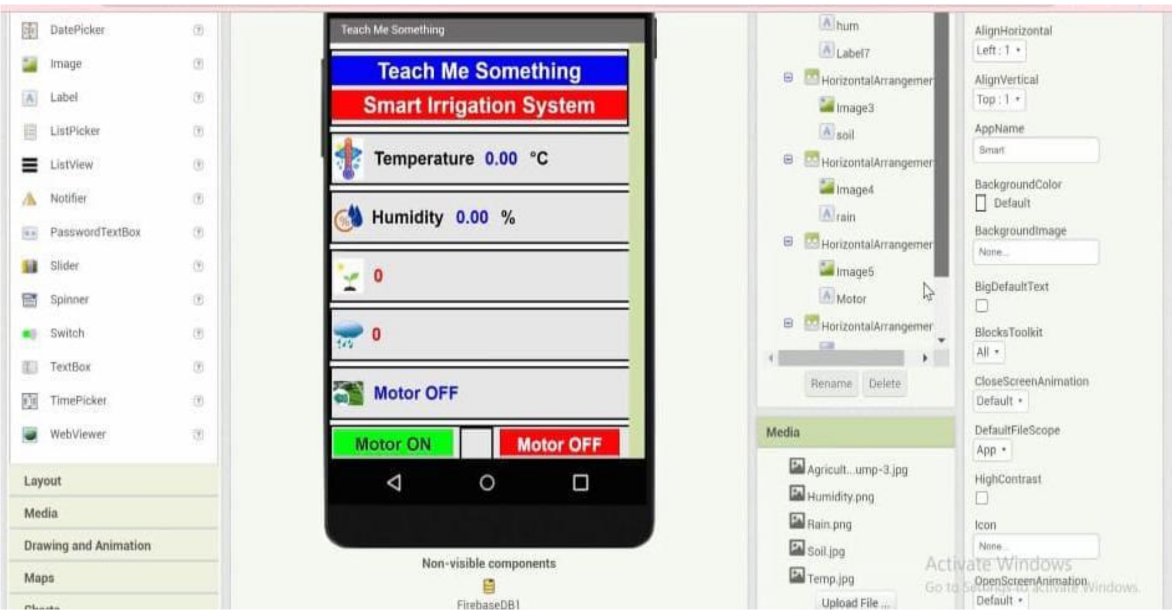


➢ The Firebase installation is created by Firebase console website.



➢ This firebase authentication and token ID is applied in code and implemented and viewed in the created application.

➢ The stored databases are shown by the software application.



REALTIME MONITORING:

IoT water management systems leverage numerous sensors that collect real-time insights on how resources are used. These devices transmit the gathered data to the user's application online. This information empowers analysis of consumption patterns and encourages more rational water consumption.

- Reduced equipment maintenance expenses.
- Transparency and better communication among stake holders.
- Remote monitoring.
- Automation and Optimized human resources use.
- Reduced risks.

CONCLUSION:

Wifi, Apps and online sites are included to monitor the quality of water which are measured by the Smart Water Management system installed in the appropriate geographical area to monitor the quality of water. It will be helpful for peoples to know about the Quality of water they are using for irrigation. It should be mainly installed in areas surrounded by farming lands even in home gardening to often check the water quality to avoid severe consequences.