Project Report On

Smart Vertical Farming and Predictive Analysis

Submitted

In partial fulfillment

For the award of the Degree of

PG-Diploma in Internet Of Things (PG-DIOT)

C-DAC,ACTS(Pune)

Guided By: Submitted By:

Mr.Suresh V

Chalake Nikhil Kumar PRN -230340126003

Choudhary Gayatri Rajesh PRN-230340126004

Gadakh Sonali Arun PRN- 230340126005

Gujarathi Kshitij Suryakant PRN-230340126006

Nandle Omkar Ravindra PRN-230340126016

Centre for Development of Advanced Computing(C-DAC),ACTS

(Pune-411008)

Acknowledgement

This is to acknowledge our indebtedness to our Project Guide, Mr.Suresh V C-DAC ACTS, Pune for his constant guidance and helpful suggestion for preparing this project Smart Vertical Farming and Predictive Analysis. We express our deep gratitude towards him for inspiration, personal involvement, constructive criticism that he provided us along with technical guidance during the course of this project.

We take this opportunity to thank Head of the department Mrs. Namrata Ailawar for providing us such a great infrastructure and environment for our overall development.

We express sincere thanks to Mrs. Namrata Ailawar, Process Owner, for their kind cooperation and extendible support towards the completion of our project.

It is our great pleasure in expressing sincere and deep gratitude towards Mrs. Risha P R (Program Head) and Mrs. Pratiksha (Course Coordinator, PG-DIOT) for their valuable guidance and constant support throughout this work and help to pursue additional studies.

Also, our warm thanks to C-DAC ACTS Pune, which provided us this opportunity to carry out this prestigious Project and enhance our learning in various technical fields.

Chalake Nikhil kumar PRN -230340126003

Choudhary Gayatri Rajesh PRN-230340126004

Gadakh Sonali Arun PRN- 230340126005

Gujarathi Kshitij Suryakant PRN-230340126006

Omkar Ravindra Nandle PRN-230340126016

ABSTRACT

The Internet of things (IoT) is refashioning agriculture enabling the farmers with a decent range of techniques like precision and sustainable agriculture to face challenges within the sector. IoT technology collects information about environmental conditions such as soil moisture, atmospheric temperature, and atmospheric humidity that are favorable for various microorganisms to develop, and cause diseases in crops. IoT supports farmer's to urge to connect his farm from anywhere and anytime in the world. Sensors connected to wireless networks are used for monitoring the farm conditions and microcontrollers are accustomed to control and automate the farm processes to remotely look at the conditions. A Smartphone allows farmers to remain updated with the continued conditions of his agricultural land using IoT at any time and any part of the world. IoT technology can reduce the challenges and enhance the productivity of traditional farming. Farmers can access the info in real-time and it enables him to search out the actual disease that arises because of favorable conditions of the environment. Hence farmers can take suitable steps timely to guard his crops from further damage because of disease. Real-time monitoring and accessing those data collected by sensors are also useful to the agricultural ministry in formulating better policies for farmers.

Table of Contents

Chapter 1

Introduction

Detection of text regions either from handwritten or printed document images containing various non-textual information is a difficult task, and it can be more challenging to locate the position of the text regions when we deal with a doctor's prescription.

1.1 Introduction

One of the most essential aspects of human survival is agriculture which is the main source of food. Unfortunately most of the farmers in our country use the normal way of farming which may be a hectic process to investigate data manually associated with soil and crops. This problem could be solved by using modern farming methods. The agriculture sector contributes a lot to the country's economic process, it's necessary to introduce the latest technologies such as IoT, automation, etc. in agriculture which relatively improves crop production and helps in developing the economy. Implementation of automation in agriculture results in effective crop health monitoring without human involvement within the field. The Internet of things is the network of physical objects embedded with sensors, software, and electronic components like microcontrollers, as sensors and microcontrollers cannot be connected to the internet directly. Crop productivity is dependent on a decent irrigation system, atmospheric conditions like temperature, humidity. IoT technology used in collecting information about conditions like weather, rainfall, humidity, temperature, and soil moisture. Wireless sensor networks are used for monitoring the farm conditions and microcontrollers are accustomed to control and automate the farm processes to look at remotely the conditions within the kind of image and video, wireless cameras are used. A smartphone allows farmers to stay updated with the continued conditions of his agricultural land using IoT at any time and any part of the world. IoT technology can minimize cost and enhance the productivity of traditional farming. The use of cloud services and creating a graphical user interface will bring healthy monitoring very easy. Farmers need not to understand the concept of using the data, GUI will make it easier to make correct decisions.

1.2 Objective and Specifications:

Smart vertical farming and predictive analysis aim to enhance crop production and optimize resource utilization by utilizing advanced technologies such as artificial intelligence, machine learning, and data analytics. The following are the objectives and specifications for implementing a smart vertical farming system with predictive analysis:

1. **Optimizing Crop Yield**: The primary objective of smart vertical farming is to optimize crop yield by providing the ideal growing conditions for crops. This can be achieved by monitoring environmental conditions such as temperature, humidity, light, and nutrient levels and adjusting them as needed.

2. **Resource Optimization:** Smart vertical farming aims to optimize resource utilization by minimizing water and nutrient waste and reducing energy consumption. This can be achieved by implementing efficient irrigation and nutrient delivery systems and utilizing renewable energy sources.

3. **Crop Growth Monitoring:** Crop growth monitoring is an essential component of smart vertical farming. It involves tracking the growth and development of crops and identifying any issues that may affect crop yield. This can be achieved by utilizing computer vision techniques and machine learning algorithms to analyze images or videos of the crops and extract relevant information such as plant height, leaf area, and growth rate.

4. **Predictive Analysis:** Predictive analysis involves using machine learning models to predict various outcomes such as crop yield, disease detection, and resource optimization. These models can help farmers make informed decisions and optimize their farming practices.

5. **Real-time Monitoring and Control:** Smart vertical farming systems should have real-time monitoring and control capabilities to enable farmers to remotely access and manage the system, receive alerts, and make adjustments as needed. This can be achieved by utilizing connectivity options such as Wi-Fi or cellular networks.

6. **Sensors:** Sensors play a crucial role in monitoring and collecting data in a vertical farming system. The specific sensors used may vary depending on the requirements of the farm, but some commonly used sensors include temperature and humidity sensors, light sensors, pH sensors.

7. **Control System**: A control system is necessary to automate and regulate various parameters in the vertical farming environment. This can include controlling lighting, temperature, humidity, nutrient delivery, and irrigation systems based on the data collected from sensors and the predictions made by the predictive analysis models.

Chapter 2
LITERATURE REVIEW

Researchers developed a sensor network which is wireless, to observe the conditions of farming and increasing crop production and quality. Sensors are used to monitor environmental parameters such as rainfall percentage, atmospheric humidity, temperature, etc. The microcontroller ATMEGA328P and sensor nodes with wireless transceiver module supported Zig bee protocol is used in designing the system. Web application and database enabled in retrieving and storing the data. In this experiment the sensor node failure and energy efficiency are monitored. An experiment conducted on a smart agriculture greenhouse monitoring system based on ZigBee technology. The system performs data acquisition, processing, transmission, and reception functions.

The objective of the experiment is to understand the greenhouse environment system, where the system is efficient in managing the environmental area and reduces the cost of farming and also saves energy.The gateway has a Linux operating system and cortex A8 processor which act as a core. Overall the planning implements remote smart monitoring and control of the greenhouse and also replaces the old wired technology to wireless, also reduces manpower cost. Operations and fulfillment are suitable places to prove efficiency gains. Researchers studied the work of a rural farming community that replaces some of the traditional techniques. The sensor nodes have different external sensors namely soil moisture sensor, soil pH, atmospheric humidity, and temperature sensors connected to it. Based on the soil moisture, the sensor activates a motor for water discharging during the period of water scarcity and switches off after the required amount of water is discharged.

This leads to conservation of water and soil pH is shipped to the bottom layer and successively the base layer intimates the farmer about soil pH via SMS using GSM model. This information helps the farmers to reduce the amount of fertilizers used. A development of rice crop monitoring using IoT is proposed to provide a helping hand in real-time monitoring and increasing rice production. The automated control of water discharge for irrigation and the ultimate supply of information is implemented using a wireless sensor network.

Chapter 3

Methodology and Techniques

1. IoT Sensors: Implementing a network of sensors in vertical farms to monitor environmental factors such as temperature, humidity, light intensity, and soil moisture. These sensors collect real-time data, which is then used for predictive modeling.

2. Data Collection and Storage: Storing sensor data in databases or cloud platforms for easy access and analysis. Technologies like Big Data and NoSQL databases are often used to handle large volumes of data efficiently.

3. Machine Learning Algorithms: Applying machine learning techniques to analyze historical and real-time data. Some common algorithms include regression analysis, decision trees, random forests, and neural networks. These algorithms can predict crop growth, yield, and health.

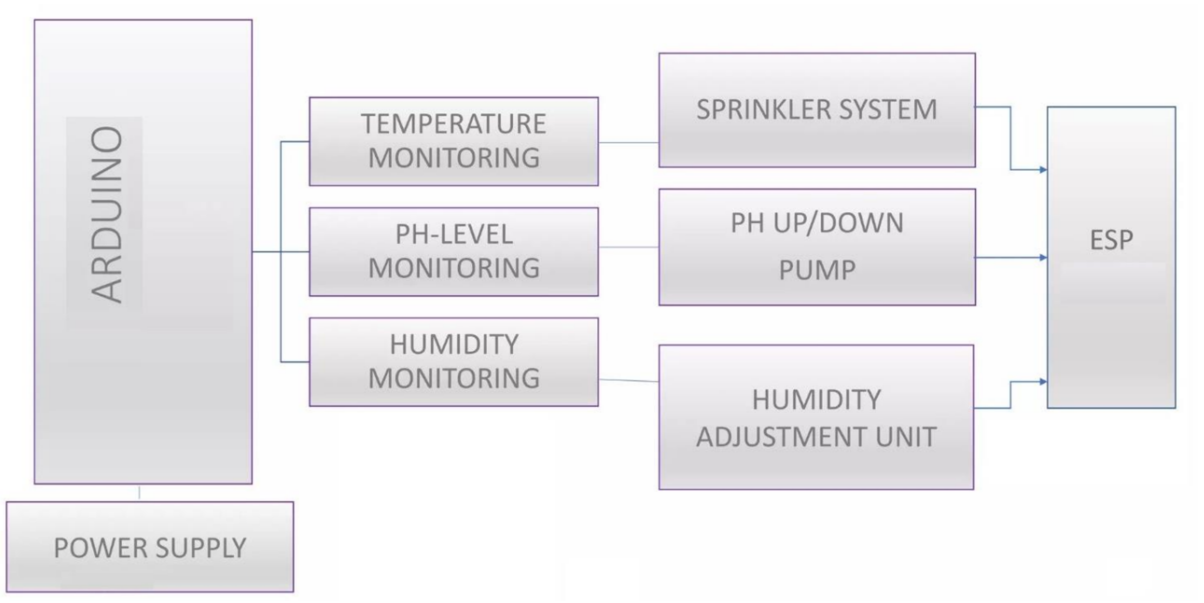
4. Crop Modeling: Creating mathematical models that simulate plant growth under different environmental conditions. These models can predict crop development, yield, and quality based on current and forecasted environmental data.

5. Remote Monitoring and Control: Enabling farmers to remotely monitor and control their vertical farms through mobile apps or web interfaces. Predictive analysis can inform remote decision-making.

Chapter 4

Implementation

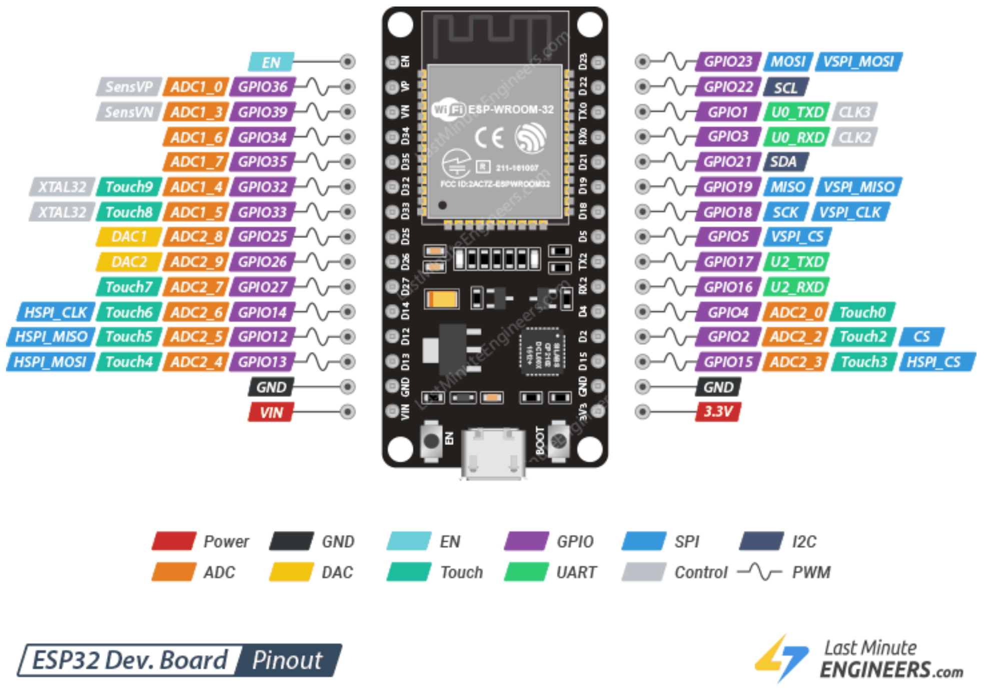
1. ESP 32
2. DHT Sensor
3. Soil Moisture Sensor
4. Light Dependent Resistor (LDR)



ESP 32 AND IT’S PIN DIAGRAM

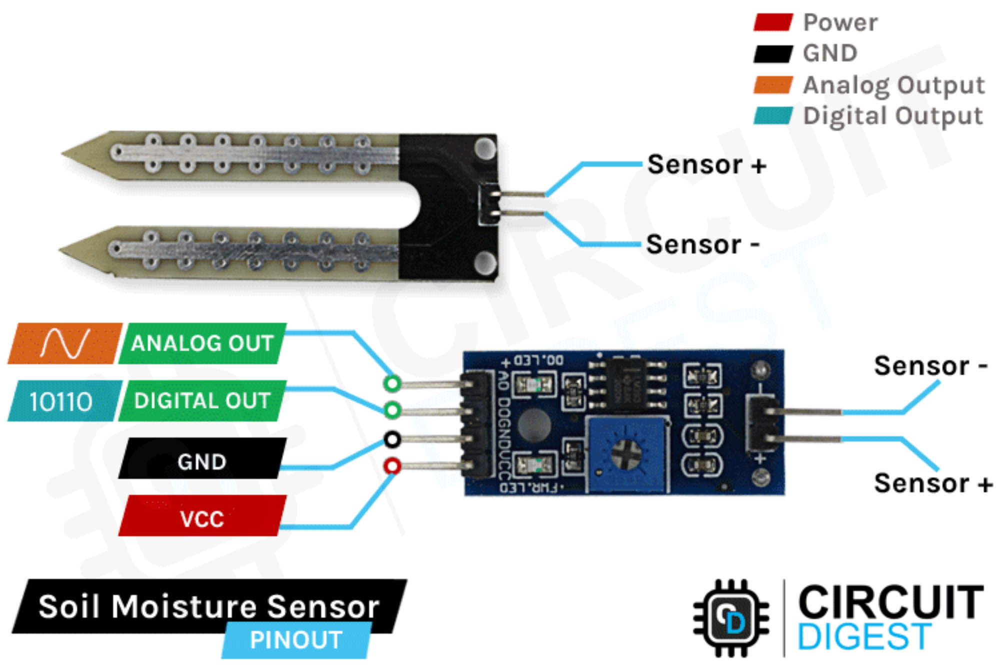
1. The ESP32, developed by Espressif Systems, is a series of low-cost microcontrollers that consume very low power and the successor to the vastly successful ESP8266 Wi-Fi SoC.

2. It is a 32-bit processor with an operating frequency of 160 kHz. It has a RAM of 52 kB. The wireless connectivity extends to Wi-Fi 802.11 b/g/n, and Bluetooth v.42 BR/EDR and BLE.



SOIL MOISTURE SENSOR

1. The soil moisture sensor is a simple sensor which senses the amount of moisture in the soil.
2. This gives us an estimate of how much water content is present in the soil based on which decision can be made as to whether the plant needs to be watered or not based on its requirement.
3. The sensor outputs an analog voltage corresponding to the moisture content in the soil.
4. There is a potentiometer onboard to set a threshold value so that any excess is indicated via an LED.



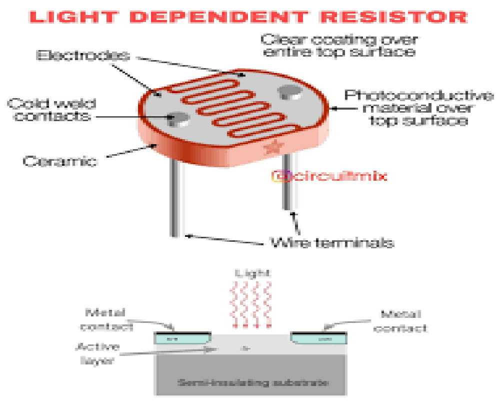
LIGHT DEPENDENT RESISTOR (LDR)

1. An LDR is nothing but a resistor whose resistance value is controlled by the amount of light incident on its surface.

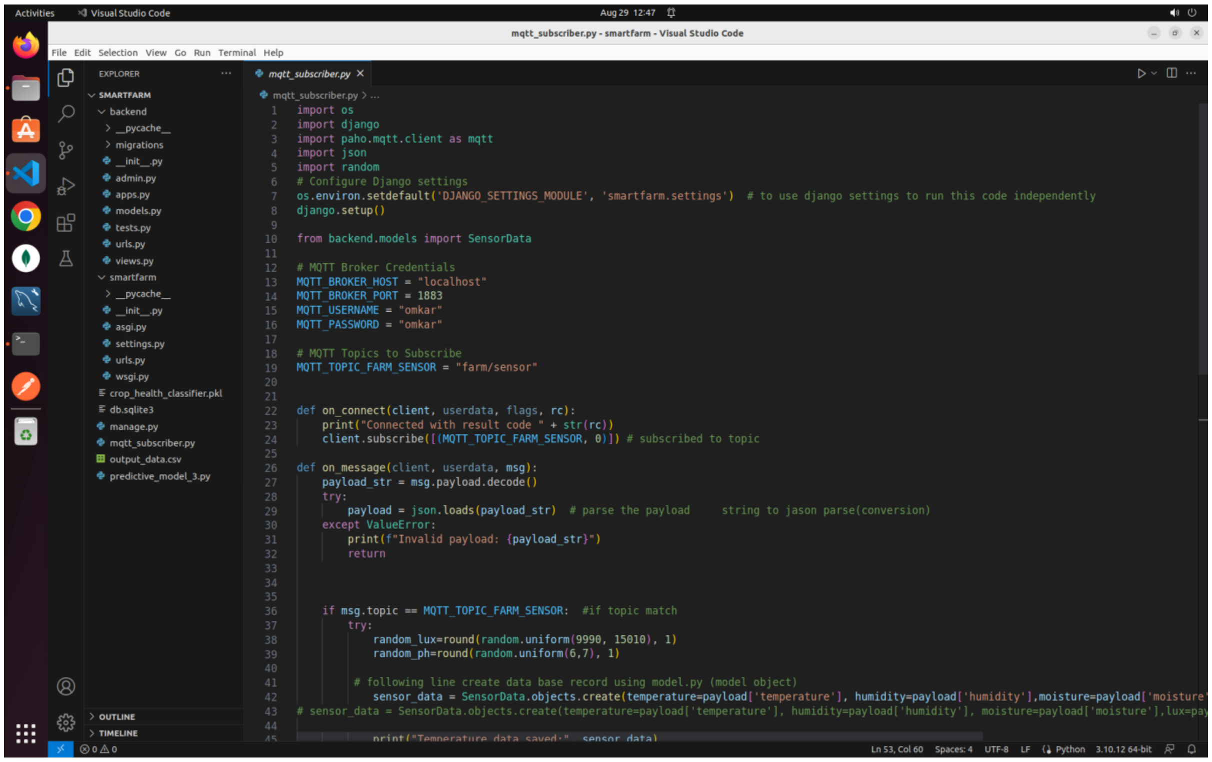
2. It works on the phenomenon of photoconductivity.

3. Whenever light falls on the LDR its resistance decreases due to the energy provided by the photons to the electrons to conduct.

4. This way, the light sensor when connected to a microcontroller outputs an analog value which can be calibrated to measure the amount of light.

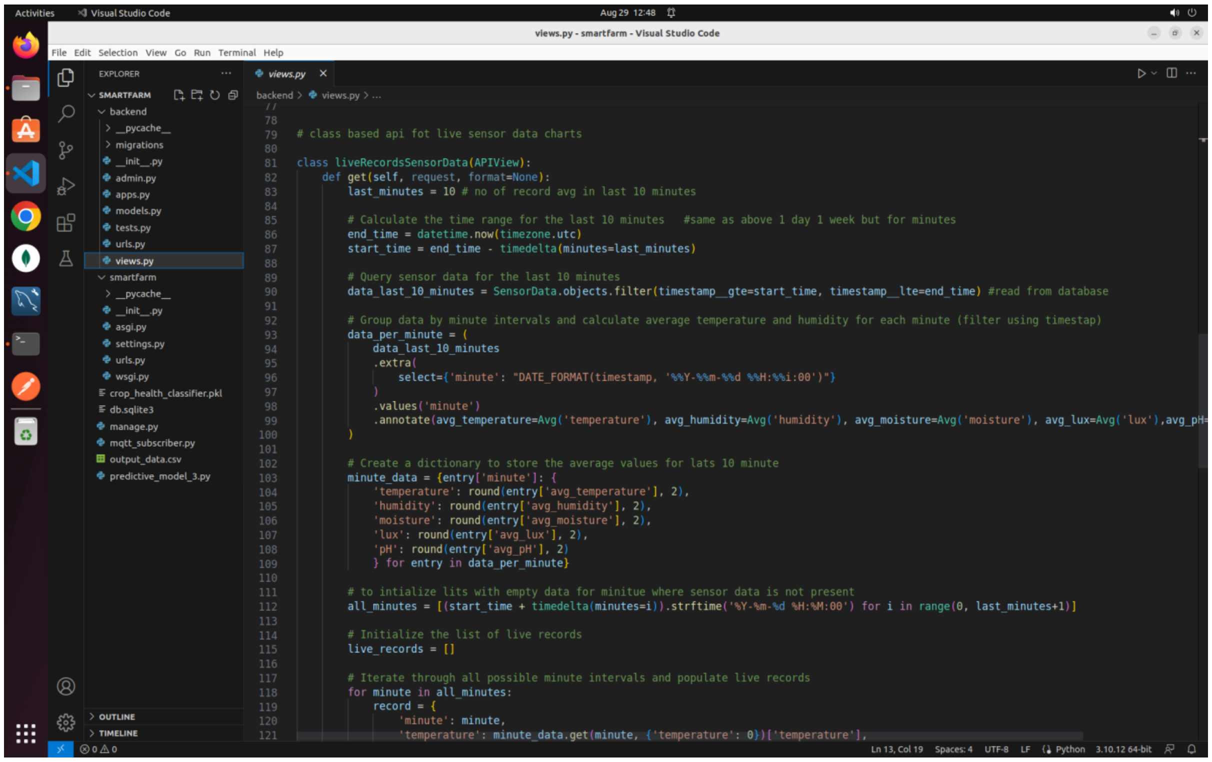


Subscriber-Backend code:



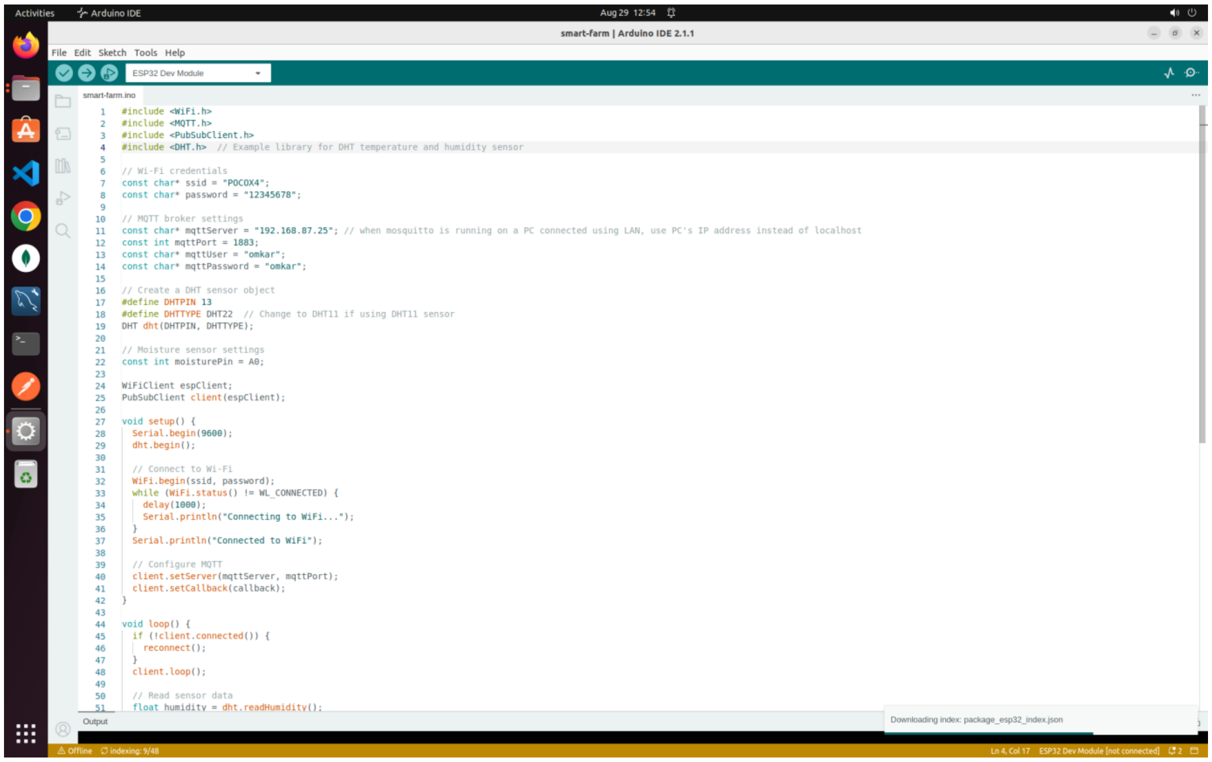
```python
import os
import django
import paho.mqtt.client as mqtt
import json
import random
# Configure Django settings
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'smartfarm.settings')  # to use django settings to run this code independently
django.setup()

from backend.models import SensorData

# MQTT Broker Credentials
MQTT_BROKER_HOST = "localhost"
MQTT_BROKER_PORT = 1883
MQTT_USERNAME = "omkar"
MQTT_PASSWORD = "omkar"

# MQTT Topics to Subscribe
MQTT_TOPIC_FARM_SENSOR = "farm/sensor"


def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))
    client.subscribe([(MQTT_TOPIC_FARM_SENSOR, 0)]) # subscribed to topic

def on_message(client, userdata, msg):
    payload_str = msg.payload.decode()
    try:
        payload = json.loads(payload_str)  # parse the payload     string to jason parse(conversion)
    except ValueError:
        print(f"Invalid payload: {payload_str}")
        return



    if msg.topic == MQTT_TOPIC_FARM_SENSOR:  #if topic match
        try:
            random_lux=round(random.uniform(9990, 15010), 1)
            random_ph=round(random.uniform(6,7), 1)

          # following line create data base record using model.py (model object)
            sensor_data = SensorData.objects.create(temperature=payload['temperature'], humidity=payload['humidity'],moisture=payload['moisture'
# sensor_data = SensorData.objects.create(temperature=payload['temperature'], humidity=payload['humidity'], moisture=payload['moisture'],lux=pay

            print("Temperature data saved:", sensor_data)
```

LiveData-Send:



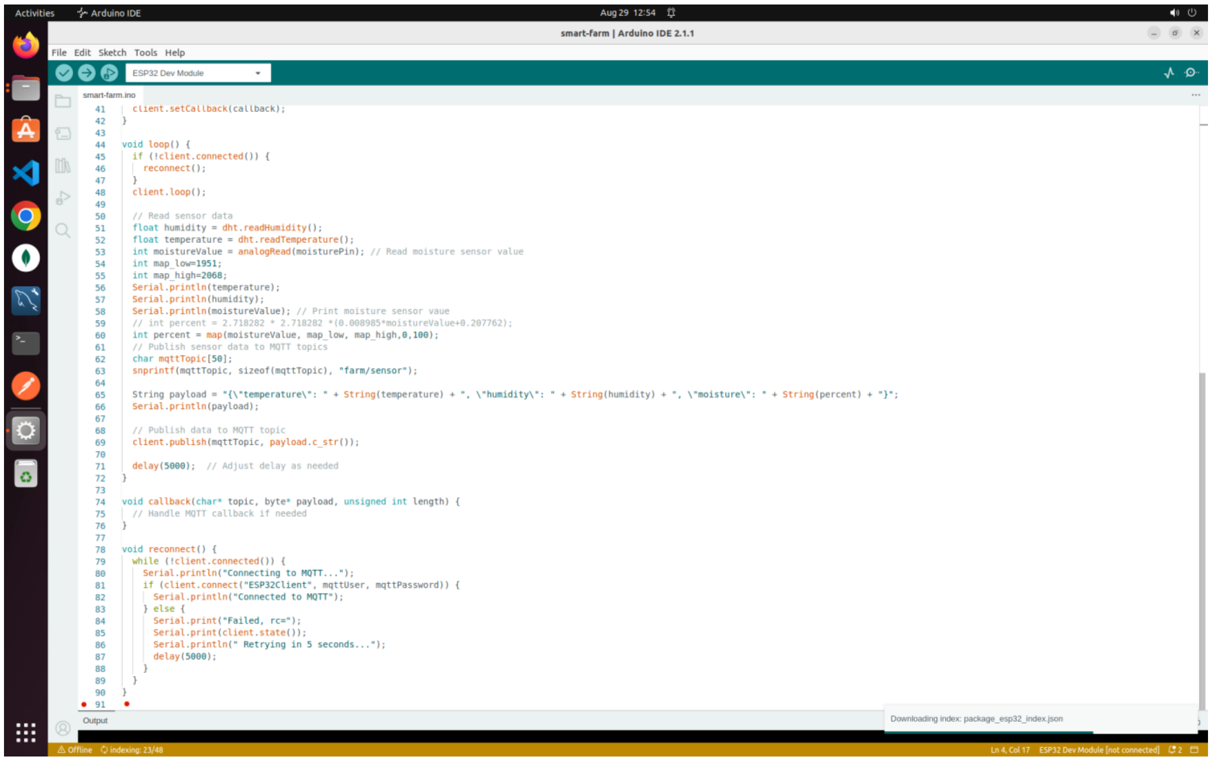
```python

# class based api fot live sensor data charts

class liveRecordsSensorData(APIView):
    def get(self, request, format=None):
        last_minutes = 10 # no of record avg in last 10 minutes

        # Calculate the time range for the last 10 minutes   #same as above 1 day 1 week but for minutes
        end_time = datetime.now(timezone.utc)
        start_time = end_time - timedelta(minutes=last_minutes)

        # Query sensor data for the last 10 minutes
        data_last_10_minutes = SensorData.objects.filter(timestamp__gte=start_time, timestamp__lte=end_time) #read from database

        # Group data by minute intervals and calculate average temperature and humidity for each minute (filter using timestap)
        data_per_minute = (
            data_last_10_minutes
            .extra(
                select={'minute': "DATE_FORMAT(timestamp, '%%Y-%%m-%%d %%H:%%i:00')"}
            )
            .values('minute')
            .annotate(avg_temperature=Avg('temperature'), avg_humidity=Avg('humidity'), avg_moisture=Avg('moisture'), avg_lux=Avg('lux'),avg_pH=
        )

        # Create a dictionary to store the average values for lats 10 minute
        minute_data = {entry['minute']: {
            'temperature': round(entry['avg_temperature'], 2),
            'humidity': round(entry['avg_humidity'], 2),
            'moisture': round(entry['avg_moisture'], 2),
            'lux': round(entry['avg_lux'], 2),
            'pH': round(entry['avg_pH'], 2)
            } for entry in data_per_minute}

        # to intialize lits with empty data for minitue where sensor data is not present
        all_minutes = [(start_time + timedelta(minutes=i)).strftime('%Y-%m-%d %H:%M:00') for i in range(0, last_minutes+1)]

        # Initialize the list of live records
        live_records = []

        # Iterate through all possible minute intervals and populate live records
        for minute in all_minutes:
            record = {
                'minute': minute,
                'temperature': minute_data.get(minute, {'temperature': 0})['temperature'],
```

Publisher-code-1:



```
#include <WiFi.h>
#include <MQTT.h>
#include <PubSubClient.h>
#include <DHT.h>  // Example library for DHT temperature and humidity sensor

// Wi-Fi credentials
const char* ssid = "POCOX4";
const char* password = "12345678";

// MQTT broker settings
const char* mqttServer = "192.168.87.25"; // when mosquitto is running on a PC connected using LAN, use PC's IP address instead of localhost
const int mqttPort = 1883;
const char* mqttUser = "omkar";
const char* mqttPassword = "omkar";

// Create a DHT sensor object
#define DHTPIN 13
#define DHTTYPE DHT22  // Change to DHT11 if using DHT11 sensor
DHT dht(DHTPIN, DHTTYPE);

// Moisture sensor settings
const int moisturePin = A0;

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  Serial.begin(9600);
  dht.begin();

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  // Configure MQTT
  client.setServer(mqttServer, mqttPort);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  // Read sensor data
  float humidity = dht.readHumidity();
```

Publisher-code-2:

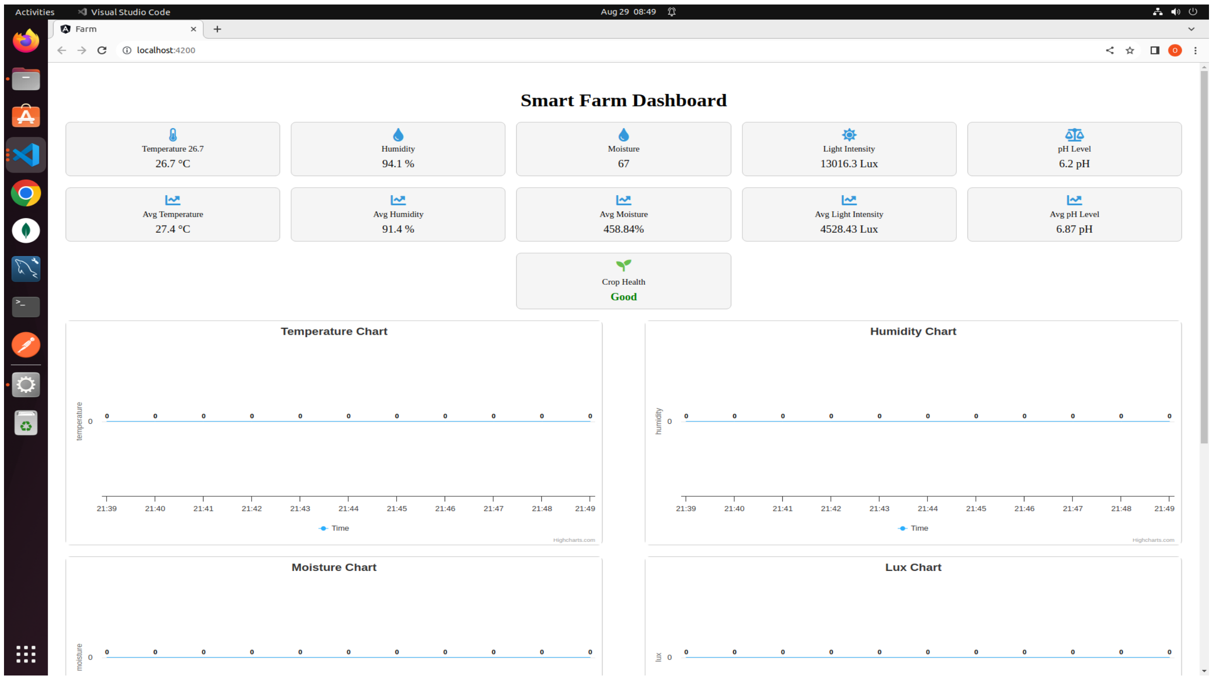


```
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  // Read sensor data
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
  int moistureValue = analogRead(moisturePin); // Read moisture sensor value
  int map_low=1951;
  int map_high=2068;
  Serial.println(temperature);
  Serial.println(humidity);
  Serial.println(moistureValue); // Print moisture sensor vaue
  // int percent = 2.718282 * 2.718282 *(0.008985*moistureValue+0.207762);
  int percent = map(moistureValue, map_low, map_high,0,100);
  // Publish sensor data to MQTT topics
  char mqttTopic[50];
  snprintf(mqttTopic, sizeof(mqttTopic), "farm/sensor");

  String payload = "{\"temperature\": " + String(temperature) + ", \"humidity\": " + String(humidity) + ", \"moisture\": " + String(percent) + "}";
  Serial.println(payload);

  // Publish data to MQTT topic
  client.publish(mqttTopic, payload.c_str());

  delay(5000);  // Adjust delay as needed
}

void callback(char* topic, byte* payload, unsigned int length) {
  // Handle MQTT callback if needed
}

void reconnect() {
  while (!client.connected()) {
    Serial.println("Connecting to MQTT...");
    if (client.connect("ESP32Client", mqttUser, mqttPassword)) {
      Serial.println("Connected to MQTT");
    } else {
      Serial.print("Failed, rc=");
      Serial.print(client.state());
      Serial.println(" Retrying in 5 seconds...");
      delay(5000);
    }
  }
}
```

Chapter 5

Results:-

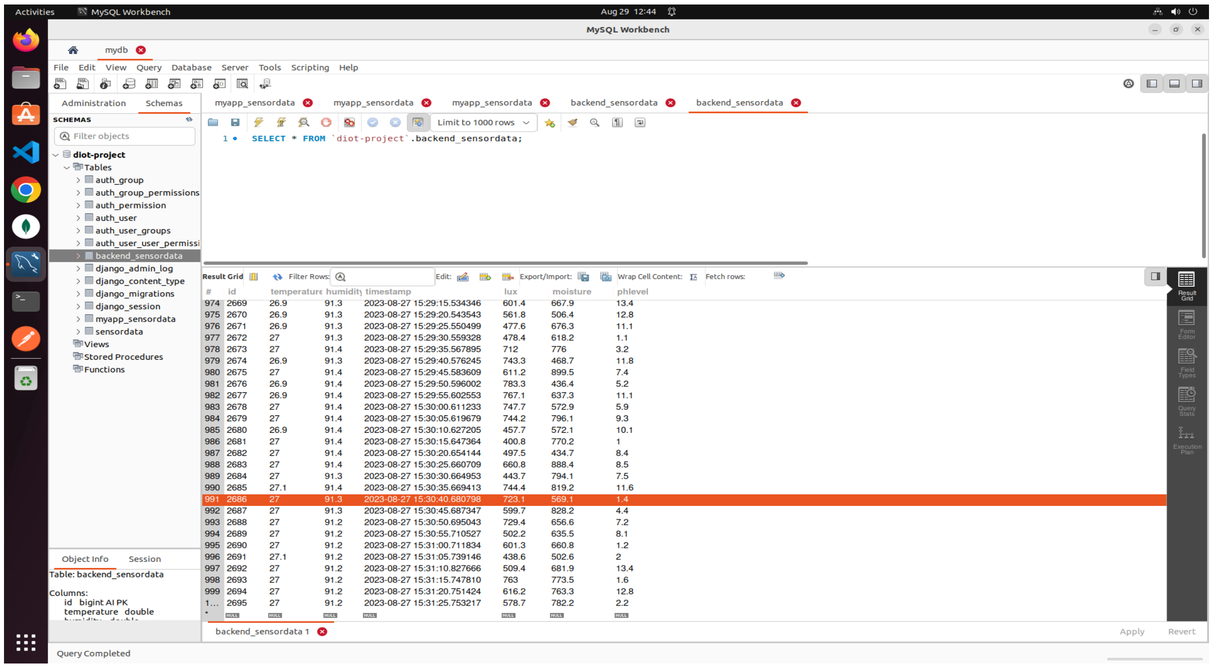
FrontEnd dashboard:



Smart Farm Dashboard

Temperature 26.7 26.7 °C	Humidity 94.1 %	Moisture 67	Light Intensity 13016.3 Lux	pH Level 6.2 pH
Avg Temperature 27.4 °C	Avg Humidity 91.4 %	Avg Moisture 458.84%	Avg Light Intensity 4528.43 Lux	Avg pH Level 6.87 pH
		Crop Health **Good**		

Database:



```
1 • SELECT * FROM `diot-project`.backend_sensordata;
```

#	id	temperature	humidity	timestamp	lux	moisture	phlevel
974	2669	26.9	91.3	2023-08-27 15:29:15.534346	601.4	667.9	13.4
975	2670	26.9	91.3	2023-08-27 15:29:20.543543	561.8	506.4	12.8
976	2671	26.9	91.3	2023-08-27 15:29:25.550499	477.6	676.3	11.1
977	2672	27	91.3	2023-08-27 15:29:30.559328	478.4	618.2	1.1
978	2673	27	91.4	2023-08-27 15:29:35.567895	712	776	3.2
979	2674	26.9	91.3	2023-08-27 15:29:40.576245	743.3	468.7	11.8
980	2675	27	91.4	2023-08-27 15:29:45.583609	611.2	899.5	7.4
981	2676	26.9	91.4	2023-08-27 15:29:50.596002	783.3	436.4	5.2
982	2677	26.9	91.4	2023-08-27 15:29:55.602553	767.1	637.3	11.1
983	2678	27	91.4	2023-08-27 15:30:00.611233	747.7	572.9	5.9
984	2679	27	91.4	2023-08-27 15:30:05.619679	744.2	796.1	9.3
985	2680	26.9	91.4	2023-08-27 15:30:10.627205	457.7	572.1	10.1
986	2681	27	91.4	2023-08-27 15:30:15.647364	400.8	770.2	1
987	2682	27	91.4	2023-08-27 15:30:20.654144	497.5	434.7	8.4
988	2683	27	91.4	2023-08-27 15:30:25.660709	660.8	888.4	8.5
989	2684	27	91.3	2023-08-27 15:30:30.664953	443.7	794.1	7.5
990	2685	27.1	91.4	2023-08-27 15:30:35.669413	744.4	819.2	11.6
991	2686	27	91.3	2023-08-27 15:30:40.680798	723.1	569.1	1.4
992	2687	27	91.3	2023-08-27 15:30:45.687347	599.7	828.2	4.4
993	2688	27	91.2	2023-08-27 15:30:50.695043	729.4	656.6	7.2
994	2689	27	91.2	2023-08-27 15:30:55.710527	502.2	635.5	8.1
995	2690	27	91.2	2023-08-27 15:31:00.711834	601.3	660.8	1.2
996	2691	27.1	91.2	2023-08-27 15:31:05.739146	438.6	502.6	2
997	2692	27	91.2	2023-08-27 15:31:10.827666	509.4	681.9	13.4
998	2693	27	91.2	2023-08-27 15:31:15.747810	763	773.5	1.6
999	2694	27	91.2	2023-08-27 15:31:20.751424	616.2	763.3	12.8
1…	2695	27	91.2	2023-08-27 15:31:25.753217	578.7	782.2	2.2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

backend_sensordata 1

Table: backend_sensordata

Columns:
id bigint AI PK
temperature double

Chapter 6

Conclusion :-

With this application, it is quite clear that there is definitely a possibility for effective farming in urban areas and cultivating our own vegetables and produce.

The low-cost devices function suitable to the need and their low power consumption translates to money saving on a large scale over a period of time.

This shows the importance of deploying IoT networks for various applications to achieve the desired results in a short span of time and conserve resources by a combination of efficient and low-cost devices.

Chapter 7

References :-

1. Rohini Shete; Sushma Agrawal, IoT based urban climate monitoring using Raspberry Pi, International Conference on Communication and Signal Processing (ICCSP), 6-8 April, 2016. Melmaruvathur, India.

2. Ravi Kishore Kodali, Archana Sahu, An IoT based soil moisture monitoring on Losant platform, 2nd International Conference on Contemporary Computing and Informatics (IC3I), 14-17 Dec. 2016. Noida, India.

3. Ravi Kishore Kodali; Kopulwar Shishir Mahesh, A low cost implementation of MQTT using ESP8266, 2nd International Conference on Contemporary Computing and Informatics (IC3I), 14-17 Dec. 2016, Noida, India.