VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Embedded Systems And IOT Lab DA5

Topic: FARM TECHBOT

Slot: BITE403L

Faculty: Dr.Kumaresan P

TEAM MEMBERS:

Sabesh Krishna – 21BIT0362

Allen Gilchrist – 21BIT0266

Danush N – 21BIT0555

Shaik Alisha Kauser – 21BIT0009

Sneha Vejju- 21BIT0647

Mitta Ushasri- 21BIT0593

AIM:

The aim of the Farm TechBot project is to develop an intelligent agricultural robot that utilizes a robot kit, NODEMCU ESP32 microcontroller, and various sensors to assist farmers in monitoring and maintaining optimal environmental conditions for crop growth.

Hardware Components:

- Robot kit
- NODEMCU ESP32
- DHT sensor
- Moisture sensor
- Ultrasonic sensor
- 9Volt Battery

Abstract:

The Farm TechBot is an innovative agricultural robot designed to assist farmers in monitoring and maintaining optimal environmental conditions for crop growth. This intelligent robot incorporates a range of sensors, including a DHT sensor, moisture sensor, and ultrasonic sensor, along with a Robot kit and NODEMCU ESP32 microcontroller. The objective of this project is to provide a cost-effective and efficient

solution for automating agricultural tasks and empowering farmers with real-time data for decision-making.

The Farm TechBot's architecture enables it to gather crucial information such as temperature, humidity, soil moisture, and obstacle detection. The robot's mobility, provided by the Robot kit, allows it to navigate through fields and gather data from various locations. The NODEMCU ESP32 microcontroller acts as the control center, processing sensor readings and facilitating wireless communication.

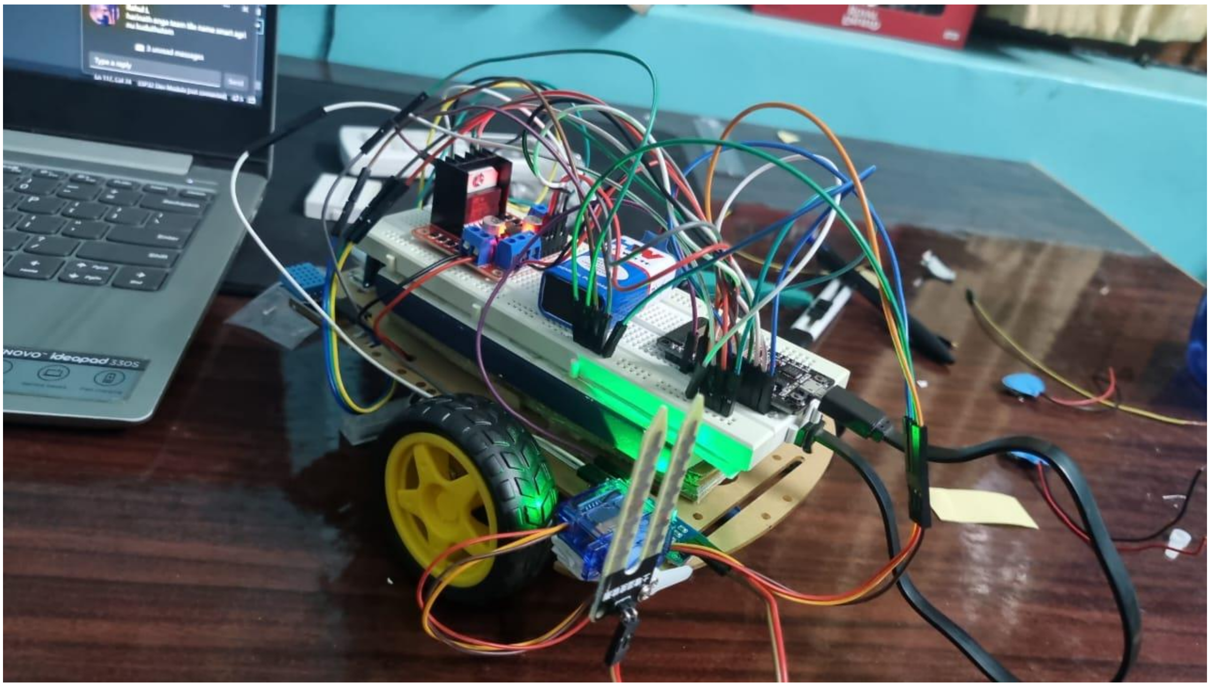
With the collected data, farmers can make informed decisions regarding irrigation, pest control, and crop management. By accurately measuring environmental parameters, the Farm TechBot empowers farmers to optimize their resource usage, reduce water waste, and implement precision agriculture techniques.

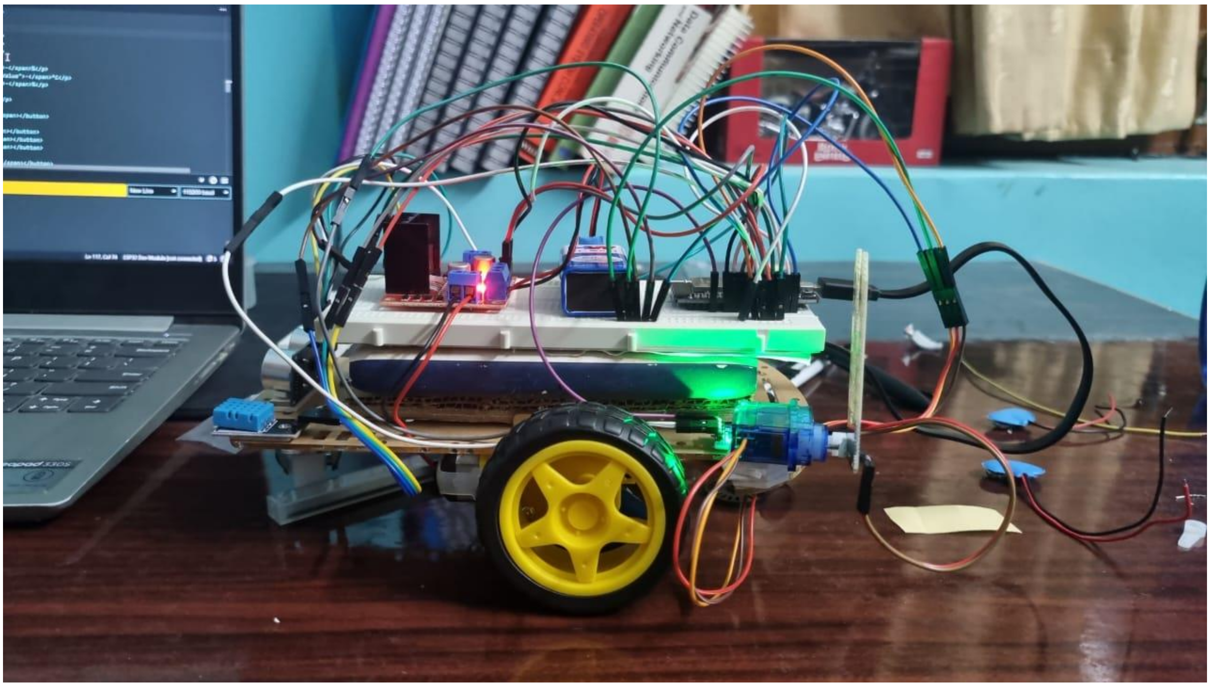
Furthermore, the Farm TechBot offers a user-friendly interface that allows farmers to access real-time data and remotely control the robot's movements. This feature enhances convenience and efficiency, enabling farmers to monitor their fields from any location.

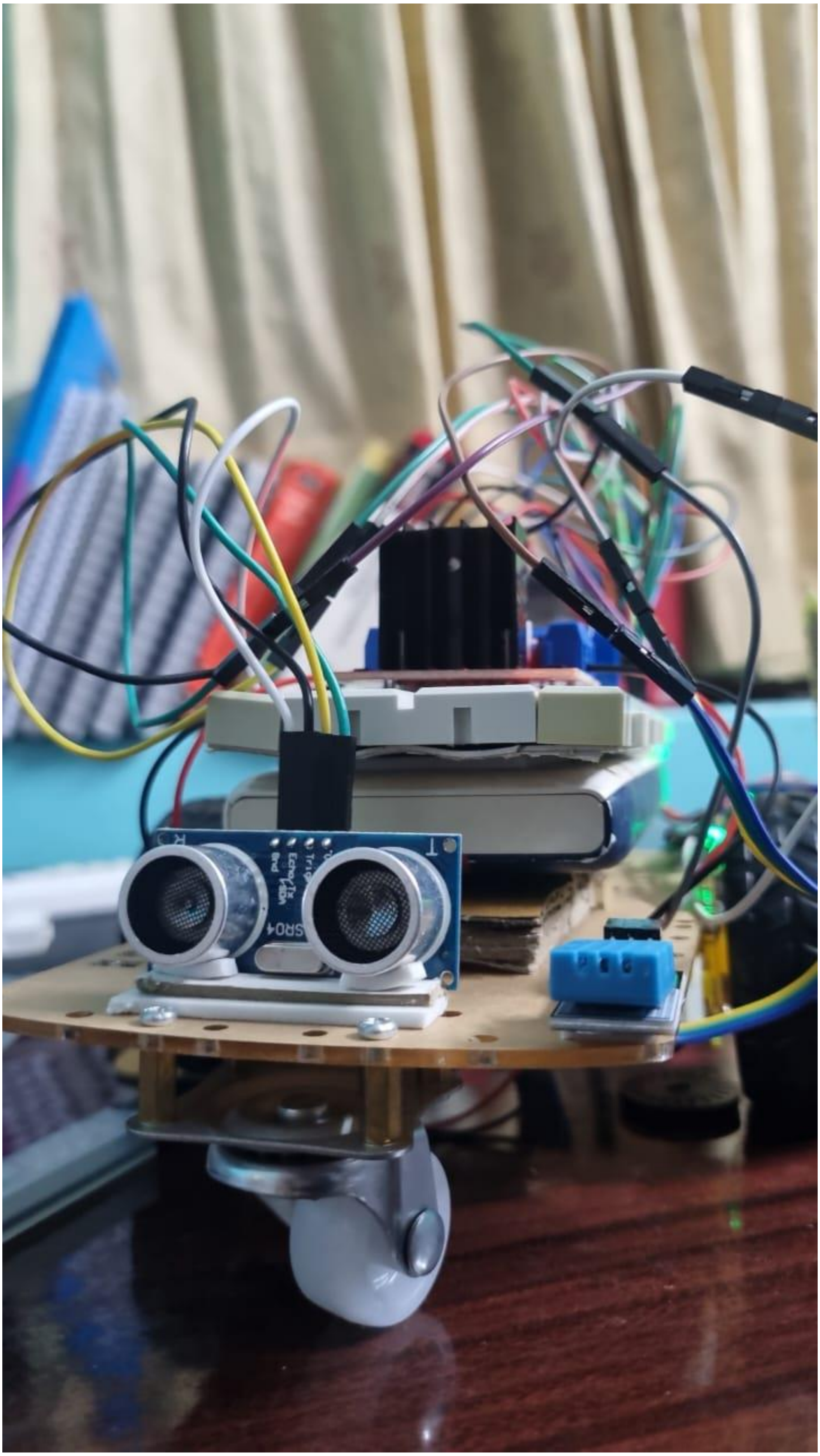
The Farm TechBot project demonstrates the potential of combining robotics, IoT, and sensor technologies to revolutionize the agriculture industry. By automating tasks and providing valuable insights, this intelligent robot contributes to increased productivity, reduced labor costs, and sustainable farming practices.

In conclusion, the Farm TechBot project showcases the capabilities of a robot equipped with advanced sensors and control systems. By leveraging these technologies, farmers can optimize their farming practices, improve crop yields, and ensure the efficient use of resources. The Farm TechBot represents a significant step towards a more technologically advanced and sustainable future for agriculture.

BOT IMAGES:






R
Gnd
Echo/Tx
Trig
SR04
T

SOURCE CODE:

```
#include <DHT.h>
#include <WiFi.h>
#include <ESPAsyncWebSrv.h>
#include <ArduinoJson.h>
#include <ESP32Servo.h>   // Include the Servo library

#define DHTPIN 14        // Pin D14 connected to the DHT sensor
#define DHTTYPE DHT11     // DHT sensor type

DHT dht(DHTPIN, DHTTYPE);  // Initialize DHT sensor
Servo servo;             // Create a servo object

// WiFi credentials
const char* ssid = "your_SSID";
const char* password = "your_password";

// Pin definitions for L298N motor driver
const int motor1In1Pin = 26;
const int motor1In2Pin = 25;
const int motor2In1Pin = 33;
const int motor2In2Pin = 32;

// Ultrasonic sensor pin definitions
const int trigPin = 13;
const int echoPin = 12;

// Soil moisture sensor pin definition
const int sensorPin = 34;  // Soil moisture sensor O/P pin
```

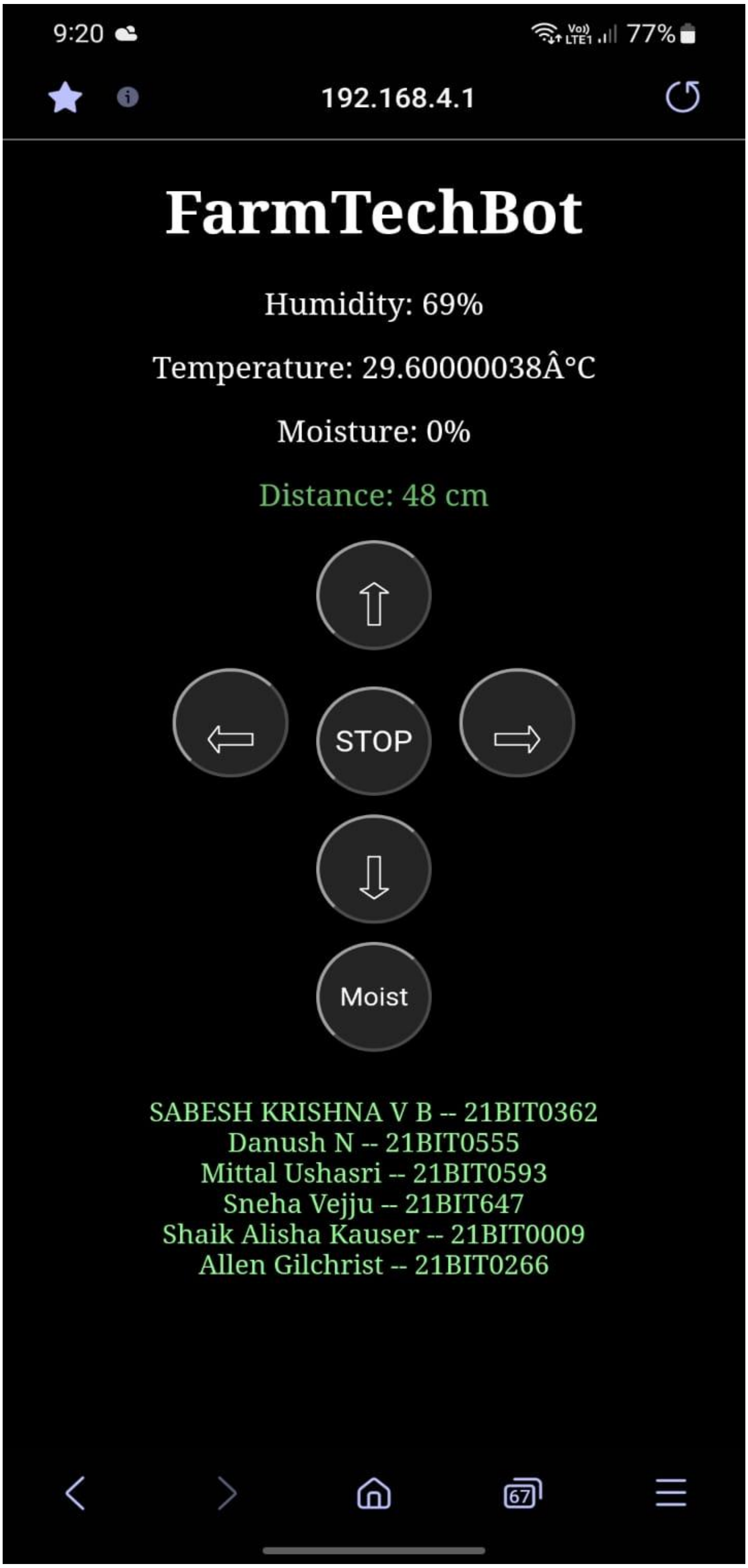
```
// HTML page content
const char* htmlPage = R"HTML(
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    .arrows {
      font-size: 70px;
      color: black;
      position: center;
    }
    .stopButton {
      font-size: 25px;
      position: center;
      color: black;
    }
    .button {
      width: 100px;
      height: 100px;
      font-size: 20px;
      margin: 10px;
      border-radius: 25%;
    }
    .distance {
      color: lightgreen;
      font-size: 30px;
      text-align: center;
      margin-bottom: 10px;
    }
    .obstacle {
      color: red;
      font-size: 34px;
      text-align: center;
      margin-top: 20px;
    }
  </style>
  <script>
    function updateValues(humidity, temperature, moisture) {
      document.getElementById("humidityValue").textContent = humidity;
      document.getElementById("temperatureValue").textContent = temperature;
      document.getElementById("moistureValue").textContent = moisture;
    }

    function move(direction) {
      var xhttp = new XMLHttpRequest();
      xhttp.open("GET", "/move?dir=" + direction, true);
      xhttp.send();
    }

    function updateDistance(distance) {
      var distanceElement = document.getElementById("distance");
      var obstacleTextElement = document.getElementById("obstacleText");

      distanceElement.textContent = distance + " cm";

      if (distance < 10) {
        obstacleTextElement.style.display = "block";
      } else {
```

```
    obstacleTextElement.style.display = "none";
   }
  }

  function rotateServo() {
   var xhttp = new XMLHttpRequest();
   xhttp.open("GET", "/rotate", true);
   xhttp.send();
  }

  setInterval(function() {
   var xhttp = new XMLHttpRequest();
   xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
     var data = JSON.parse(this.responseText);
     updateValues(data.humidity, data.temperature, data.moisture);
     updateDistance(data.distance);
    }
   };
   xhttp.open("GET", "/data", true);
   xhttp.send();
  }, 1000);
 </script>
</head>
<body class="noselect" align="center" style="background-color:black">
 <h1 style="color: white; text-align: center;">Control Your Car!!</h1>
 <p style="color: black; text-align: center;">Humidity: <span id="humidityValue">-
</span>%</p>
 <p style="color: black; text-align: center;">Temperature: <span id="temperatureValue">-
</span>°C</p>
```

```
  <p style="color: black; text-align: center;">Moisture: <span id="moistureValue">-</span>%</p>
  <p class="distance">Distance: <span id="distance">-</span></p>
  <p class="obstacle" id="obstacleText" style="display: none;">Obstacle Detected</p>

  <button class="button" onclick="move('forward')"><span class="arrows">&#8679;</span></button>
  <br>
  <button class="button" onclick="move('left')"><span class="arrows">&#8678;</span></button>
  <button class="button" onclick="move('stop')"><span class="stopButton">STOP</span></button>
  <button class="button" onclick="move('right')"><span class="arrows">&#8680;</span></button>
  <br>
  <button class="button" onclick="move('backward')"><span class="arrows">&#8681;</span></button>
  <br>
  <button class="button" onclick="rotateServo()">Moist</button>  <!-- New button for servo control -->
</body>
</html>
)HTML";

AsyncWebServer server(80);
String currentDirection = "stop";

void setup() {
  Serial.begin(115200);

  WiFi.softAP(ssid, password);
  IPAddress IP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(IP);

  pinMode(motor1In1Pin, OUTPUT);
  pinMode(motor1In2Pin, OUTPUT);
  pinMode(motor2In1Pin, OUTPUT);
  pinMode(motor2In2Pin, OUTPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  dht.begin();

  servo.attach(27);  // Attach the servo to pin 4

  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send(200, "text/html", htmlPage);
  });

  server.on("/move", HTTP_GET, [](AsyncWebServerRequest *request){
    String direction = request->getParam("dir")->value();
    // Update currentDirection only if the direction is valid
    if (direction == "forward" || direction == "backward" || direction == "left" || direction == "right" || direction == "stop") {
      currentDirection = direction;
    }

    request->send(200, "text/plain", "OK");
  });
  server.on("/rotate", HTTP_GET, [](AsyncWebServerRequest *request){
    rotateServo();  // Call the rotateServo function
    request->send(200, "text/plain", "OK");
  });

  server.on("/data", HTTP_GET, [](AsyncWebServerRequest *request){
    float humidity = dht.readHumidity();     // Read humidity value
    float temperature = dht.readTemperature(); // Read temperature value in Celsius
    int moisture = getMoisture();

    if (isnan(humidity) || isnan(temperature)) {
      Serial.println("Failed to read from DHT sensor!");
      return;
    }

    StaticJsonDocument<200> jsonDocument;
    jsonDocument["humidity"] = humidity;
    jsonDocument["temperature"] = temperature;
    jsonDocument["moisture"] = moisture;
    jsonDocument["distance"] = getDistance();

    String jsonData;
    serializeJson(jsonDocument, jsonData);

    request->send(200, "application/json", jsonData);
  });

  server.begin();
}
```

```
void loop() {
  if (currentDirection == "forward") {
    digitalWrite(motor1In1Pin, HIGH);
    digitalWrite(motor1In2Pin, LOW);
    digitalWrite(motor2In1Pin, HIGH);
    digitalWrite(motor2In2Pin, LOW);
  } else if (currentDirection == "backward") {
    digitalWrite(motor1In1Pin, LOW);
    digitalWrite(motor1In2Pin, HIGH);
    digitalWrite(motor2In1Pin, LOW);
    digitalWrite(motor2In2Pin, HIGH);
  } else if (currentDirection == "left") {
    digitalWrite(motor1In1Pin, LOW);
    digitalWrite(motor1In2Pin, HIGH);
    digitalWrite(motor2In1Pin, HIGH);
    digitalWrite(motor2In2Pin, LOW);
  } else if (currentDirection == "right") {
    digitalWrite(motor1In1Pin, HIGH);
    digitalWrite(motor1In2Pin, LOW);
    digitalWrite(motor2In1Pin, LOW);
    digitalWrite(motor2In2Pin, HIGH);
  } else if (currentDirection == "stop") {
    digitalWrite(motor1In1Pin, LOW);
    digitalWrite(motor1In2Pin, LOW);
    digitalWrite(motor2In1Pin, LOW);
    digitalWrite(motor2In2Pin, LOW);
  }

  // Rest of the code...
  float humidity = dht.readHumidity();     // Read humidity value
  float temperature = dht.readTemperature(); // Read temperature value in Celsius
  int moisture = getMoisture();

  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print("%, Temperature: ");
  Serial.print(temperature);
  Serial.print("°C, Moisture: ");
  Serial.print(moisture);
  Serial.println("%");

  // Delay before next reading
  delay(1000);
}

int getMoisture() {
  int sensor_analog = analogRead(sensorPin);
  int moisture = 100 - ((sensor_analog / 4095.0) * 100);
  return moisture;
}

long getDistance() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  if (distance < 10) {
    currentDirection = "stop";
  }

  return distance;
}

void rotateServo() {
  servo.write(120);     // Rotate servo to 120 degrees
  delay(1000);          // Wait for 5 seconds
  servo.write(0);       // Rotate servo back to 0 degrees
}
```

WEB PAGE:



FarmTechBot

Humidity: 69%

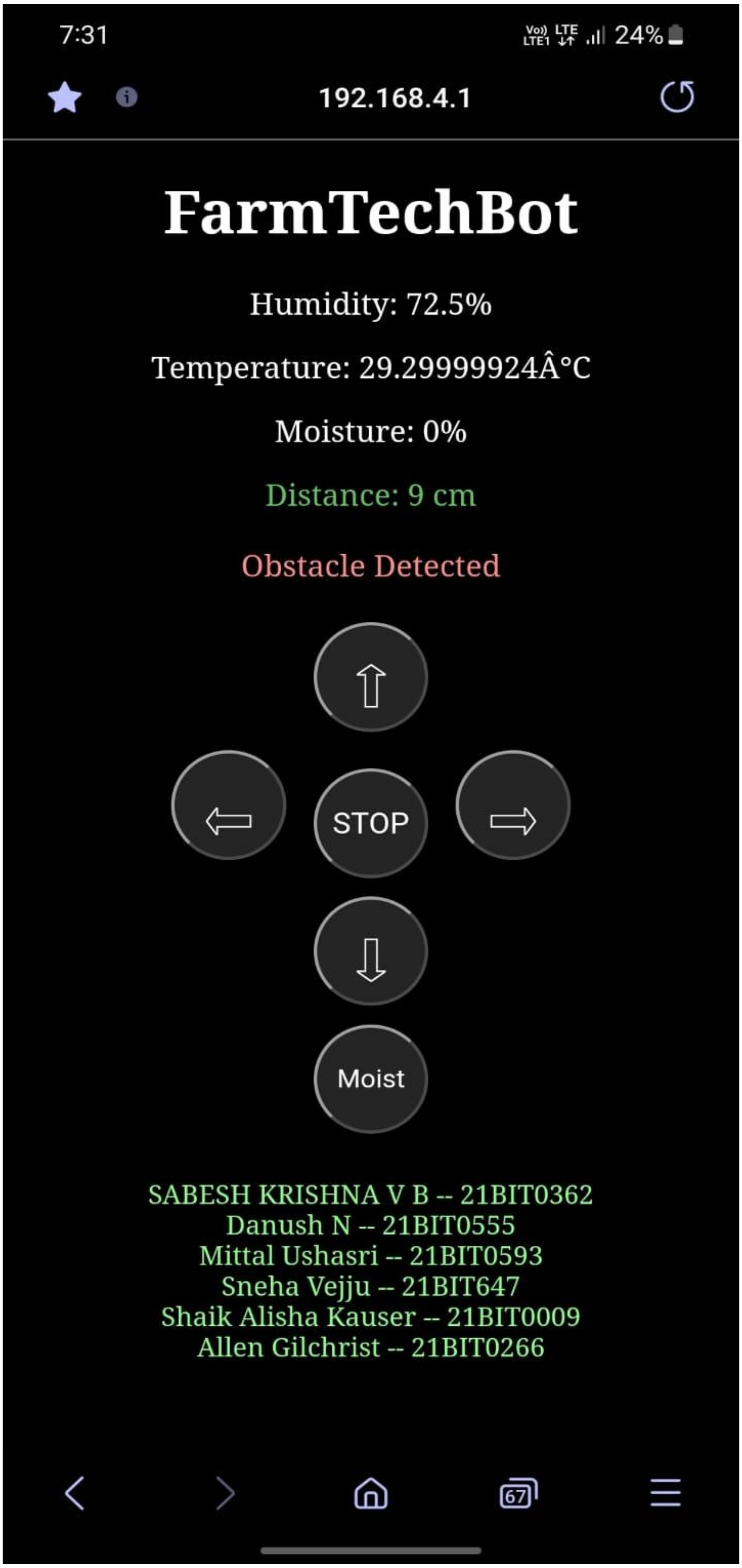
Temperature: 29.60000038Â°C

Moisture: 0%

Distance: 48 cm

SABESH KRISHNA V B -- 21BIT0362
Danush N -- 21BIT0555
Mittal Ushasri -- 21BIT0593
Sneha Vejju -- 21BIT647
Shaik Alisha Kauser -- 21BIT0009
Allen Gilchrist -- 21BIT0266

WHEN OBSTACLE DETECTS:



FarmTechBot

Humidity: 72.5%

Temperature: 29.29999924Â°C

Moisture: 0%

Distance: 9 cm

Obstacle Detected

SABESH KRISHNA V B -- 21BIT0362
Danush N -- 21BIT0555
Mittal Ushasri -- 21BIT0593
Sneha Vejju -- 21BIT647
Shaik Alisha Kauser -- 21BIT0009
Allen Gilchrist -- 21BIT0266

WORKING VIDEO DRIVE LINK:

https://drive.google.com/file/d/1IztOsi3Xqur0VEYXwCOKcvHmxUx8_IFg/view?usp=drive_link

CONCLUSION:

The Farm Tech Bot project successfully demonstrates the use of a robot kit, NODEMCU ESP32 microcontroller, and various sensors to create an intelligent agricultural robot. The robot's ability to monitor temperature, humidity, soil moisture, and obstacles helps farmers make informed decisions for optimal crop growth and resource management. The project offers a cost-effective and efficient solution for automating agricultural tasks, reducing manual labor, and increasing productivity. With further improvements and integration with advanced technologies, the Farm Tech Bot has the potential to revolutionize the agriculture industry and contribute to sustainable farming practices.