

COLLEGE CODE : 3114

COLLEGE NAME : MEENAKSHI COLLEGE OF ENGINEERING

DEPARTMENT : CSE (AI&ML)

STUDENT NM-ID: : 75F0631C310586CABDB4850F0A3942F7

ROLL NO :311423148039

PROJECT NAME WEATHER IMPACT ANALYSIS IN CROP MONITORING AND PEST MANAGEMENT

SUBMITTED BY,

NAME R. PRIYA DHARSHINI

TEAM MEMBERS NAME :

1. PRABHA. L

2. HARINI. E

3. SAI CHANDINI. P

4. SANDHIYA.M

WEATHER IMPACT ANALYSIS IN CROP MONITORING AND PEST MANAGEMENT

STIMULATION CODE:

#include <Wire.h>

#include <Adafruit\_SSD1306.h>

#include <DHT.h>

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

#define OLED\_RESET -1

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

// DHT22 settings

#define DHTPIN 3

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

// Pins

#define SOIL\_PIN 34 // Analog pin for potentiometer (soil moisture)

#define RAIN\_PIN 35 // Digital pin for button (rain sensor)

#define BUZZER\_PIN 25

#define LED\_PIN 2

void setup() {

Serial.begin(115200); // Initialize serial communication

dht.begin();

pinMode(RAIN\_PIN, INPUT\_PULLUP); // Using internal pull-up resistor

pinMode(BUZZER\_PIN, OUTPUT);

pinMode(LED\_PIN, OUTPUT);

if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

Serial.println("OLED failed");

while (true);

}

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306\_WHITE);

// Greeting sequence

display.setCursor(0, 0);

display.println("Hello! Welcome to");

display.println("Smart Farm System");

display.display();

delay(2000);

// Team Members

display.clearDisplay();

display.setCursor(0, 0);

display.println("Team Members:");

display.println("1.Prabh.L");

display.println("2.Priya dharshini.R");

display.println("3.Harini.E");

display.println("4.Sandhiya.M");

display.println("5.Sai chandini.P");

display.display();

delay(3000);

// Project Details

display.clearDisplay();

display.setCursor(0, 0);

display.println("Project:");

display.println("Weather impact");

display.println("Analysis and");

display.println("Crop Monitoring");

display.display();

delay(3000);

// Thank You

display.clearDisplay();

display.setCursor(0, 0);

display.println("Thank you!");

display.display();

delay(2000);

}

void loop() {

delay(2000); // Give time between sensor reads

// Reading temperature, humidity from DHT22

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();

// Check if reading failed

if (isnan(temperature) || isnan(humidity)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

// Reading soil moisture value from potentiometer (0-4095)

int soilValue = analogRead(SOIL\_PIN);

// Reading rain sensor (button press = rain)

bool isRaining = digitalRead(RAIN\_PIN) == LOW;

// Display values on OLED screen

display.clearDisplay();

display.setCursor(0, 0);

display.print("Temp: ");

display.print(temperature);

display.println(" C");

display.print("Humidity: ");

display.print(humidity);

display.println(" %");

display.print("Soil: ");

display.print(soilValue); // Raw analog value

display.println(" (analog)");

display.print("Rain: ");

display.println(isRaining ? "Yes" : "No");

display.display();

// Output to Serial Monitor

Serial.print("Temp: ");

Serial.print(temperature);

Serial.println(" C");

Serial.print("Humidity: ");

Serial.print(humidity);

Serial.println(" %");

Serial.print("Soil Moisture (analog): ");

Serial.println(soilValue);

Serial.print("Rain Detected: ");

Serial.println(isRaining ? "Yes" : "No");

// Alert if soil is dry or it's raining

if (soilValue > 3000 || isRaining) {

digitalWrite(BUZZER\_PIN, HIGH);

digitalWrite(LED\_PIN, HIGH);

} else {

digitalWrite(BUZZER\_PIN, LOW);

digitalWrite(LED\_PIN, LOW);

}

}

OUTPUT LINK FOR THE STIMULATION CODE :

<https://wokwi.com/projects/430904918314288129>

CODE FOR INDEX.HTML:

<!DOCTYPE html>

<html>

<head>

<title>Weather and Crops</title>

<link rel="stylesheet" href="style.css">

<!-- SweetAlert library for nice alerts -->

<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>

</head>

<body>

<h1>🌦 Weather and Crop Helper</h1>

<h2>Enter Your City:</h2>

<input type="text" id="cityInput" placeholder="Type city name here">

<button onclick="getWeather()">Get Weather</button>

<h2>📡 Weather Info:</h2>

<div id="weatherResult">Waiting for your city...</div>

<h2>🌱 Crop Suggestion:</h2>

<div id="cropSuggestion">Get weather first!</div>

<h2>🐛 Pest Tip:</h2>

<div id="pestSuggestion">Get weather first!</div>

<script src="script.js"></script>

</body>

</html>

CODE FOR STYLE.CSS:

body {

font-family: Arial, sans-serif;

background-color: #e0f7fa;

padding: 20px;

}

h1 {

color: #00796b;

}

input, button {

padding: 10px;

margin-top: 5px;

}

div {

background: white;

padding: 10px;

margin-top: 10px;

border: 1px solid #ccc;

}

img {

width: 60px;

height: 60px;

}

CODE FOR SCRIPT.JS:

function getWeather() {

const city = document.getElementById("cityInput").value;

const apiKey = "3963a6d3f318e5bf4bd974ba5619b1eb";

const url = https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=metric;

fetch(url)

.then(response => response.json())

.then(data => {

const temp = data.main.temp;

const humidity = data.main.humidity;

const iconCode = data.weather[0].icon;

const iconUrl = https://openweathermap.org/img/wn/${iconCode}@2x.png;

const now = new Date();

const dateTime = now.toLocaleString();

const weatherText = `

<img src="${iconUrl}" alt="Weather Icon">

<br>🌡 Temperature: ${temp}°C

<br>💧 Humidity: ${humidity}%

<br>📅 Checked on: ${dateTime}

`;

document.getElementById("weatherResult").innerHTML = weatherText;

// 🌱 Crop Suggestion

let cropMsg = "";

if (temp > 32 && humidity > 60) {

cropMsg = "Suggested Crops: Rice, Sugarcane, Banana 🌾🍌";

} else if (temp >= 28 && temp <= 32 && humidity >= 40 && humidity <= 60) {

cropMsg = "Suggested Crops: Maize, Cotton, Soybean 🌽🧺";

} else if (temp >= 22 && temp < 28 && humidity >= 30 && humidity <= 60) {

cropMsg = "Suggested Crops: Wheat, Tomato, Groundnut 🍅🥜";

} else if (temp >= 15 && temp < 22 && humidity >= 40 && humidity <= 80) {

cropMsg = "Suggested Crops: Mustard, Cabbage, Potato 🥬🥔";

} else if (temp < 15) {

cropMsg = "Suggested Crops: Peas, Carrot, Garlic 🥕🧄";

} else {

cropMsg = "Conditions are unusual. Please consult local agriculture expert.";

}

document.getElementById("cropSuggestion").innerText = cropMsg;

// 🐛 Pest Suggestion

let pestMsg = "";

if (temp > 32 && humidity > 70) {

pestMsg = "⚠ High Risk: Fungal diseases, Stem borers, Brown plant hoppers.";

} else if (temp >= 25 && temp <= 32 && humidity >= 50 && humidity <= 70) {

pestMsg = "⚠ Risk: Aphids, Fall armyworms.";

} else if (temp >= 20 && temp < 28 && humidity < 60) {

pestMsg = "⚠ Risk: Bollworms, Whiteflies.";

} else if (temp < 20) {

pestMsg = "⚠ Risk: Cabbage worms, Aphids.";

} else if (temp > 30 && humidity < 50) {

pestMsg = "⚠ Risk: Thrips, Mites.";

} else if (humidity > 80) {

pestMsg = "⚠ Risk: Leaf spot, Rhizome rot.";

} else {

pestMsg = "✅ Pest risk is low now.";

}

document.getElementById("pestSuggestion").innerText = pestMsg;

// 🛎 Show notification popup for pest alert

if (pestMsg.includes("⚠")) {

swal("Pest Alert!", pestMsg, "warning");

} else {

swal("Good News!", "Pest risk is low. Crops are safe 🌿", "success");

}

})

.catch(error => {

document.getElementById("weatherResult").innerText = "❌ City not found!";

document.getElementById("cropSuggestion").innerText = "-";

document.getElementById("pestSuggestion").innerText = "-";

swal("Error", "City not found or API error.", "error");

});

}