#### A Report on

**CROPMITRA** **🌱 (Smart Irrigation System Using IOT)**

Submitted in partial fulfillment of the Academic Requirement for the Award of Degree of

**BACHELOR OF TECHNOLOGY**

in

### COMPUTER SCIENCE AND ENGINEERING

#### Submitted By

|  |  |
| --- | --- |
| **K. Sri Harsha** | **(24R05A0524)** |
| **M. Navya** | **(24R05A0527)** |
| **M. Lokesh** | **(24R05A0530)** |
| **P. Navyasri** | **(24R05A0535)** |

Under the Esteemed guidance of

Mrs.K.Mouneshwari

(Assistant Professor, Dept. of CSE)

### 

### CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMUS)

Approved by AICTE, Permanent Affiliation to JNTUH, Accredited by NBA and NAAC

*Kandlakoya(V), Medchal Dist – 501401*

[**www.cmrithyderabad.edu.in**](http://www.cmrithyderabad.edu.in/)

**2024-25**

**CMR INSTITUTE OF TECHNOLOGY (UGC AUTONOMOUS)**

*(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad) Kandlakoya, Medchal Road, Hyderabad.*

**CERTIFICATE**

This is to certify that a Real Time Societal Research Project entitled with:

“CropMitra” is being

Submitted By

|  |  |
| --- | --- |
| **K. Sri Harsha** | **(24R05A0524)** |
| **M. Navya** | **(24R05A0527)** |
| **M. Lokesh** | **(24R05A0530)** |
| **P. Navyasri** | **(24R05A0535)** |

In partial fulfillment of the requirement for award of the Real Time Societal Research Project Laboratory of II B. Tech II Semester in CSE to the CMRIT, Hyderabad is a record of a bonafide work carried out under our guidance and supervision.

**Signature of Guide Signature of Project Coordinator Signature of HOD Mrs.K.Mouneshwari Mrs.Shahzadi asra (HOD of CSE)**

**(Asst Professor,Dept of CSE) (Asst Professor,Dept of CSE)**

**Signature of External Examiner**

# ACKNOWLEDGEMENT

We are extremely grateful to Director, Principal and Head of Department, of Computer Science and Engineering, CMR Institute of Technology for their inspiration and valuable guidance during entire duration.

We are extremely thankful to our guide **Mrs.Shahzadi asra ,**Asst Professor,**Mrs.Shilapa Sargari,**Asst Professor Department of Computer Science and Engineering, CMR Institute of Technology for his constant guidance, encouragement and moral support throughout the project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for successful completion of the project.

|  |  |
| --- | --- |
| **K. Sri Harsha** | **(24R05A0524)** |
| **M. Navya** | **(24R05A0527)** |
| **M. Lokesh** | **(24R05A0530)** |
| **P. Navyasri** | **(24R05A0535)** |

# 

# ABSTRACT

CropMitra is an innovative, IoT-enabled smart irrigation solution that empowers farmers to optimize water usage and improve crop yields. By seamlessly integrating advanced sensors, a user- friendly website interface, and automation techniques, CropMitra addresses critical challenges in traditional irrigation practices, such as high operational costs, unreliable connectivity, and limited adaptability to diverse crop requirements.

This cutting-edge system monitors real-time soil moisture levels and weather conditions, providing timely irrigation alerts and customized recommendations to farmers. These insights enable more precise water management, reducing wastage and promoting efficient resource utilization. The system's affordability and adaptability make it suitable for both small-scale and large-scale farming operations, bridging the gap in accessibility to advanced agricultural technology.

CropMitra is designed with a focus on sustainability, energy efficiency, and practicality. By leveraging IoT technology, it ensures seamless data collection and analysis, enabling farmers to make informed decisions. The solution’s intuitive interface ensures ease of use, making it accessible even to those with minimal technical expertise. Additionally, its flexibility allows for customization based on specific crop and soil requirements, ensuring optimal irrigation for a variety of agricultural contexts.

Through its holistic approach, CropMitra not only enhances farming productivity but also contributes to sustainable agricultural practices. By reducing water waste and promoting responsible farming techniques, it aligns with global efforts to address environmental challenges and support food security. CropMitra is a step forward in the journey toward smarter, more efficient, and eco-friendly farming solutions.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter**  **No.** | **INDEX** | **Page No.** |
|  | ACKNOWLEDGEMENT | III |
|  | ABSTRACT | IV |
|  | TABLE OF CONTENTS | V |
| 1. | INTRODUCTION | 1 |
| 2. | LITERATURE SURVEY | 2 |
| 3. | SYSTEM ANALYSIS | 3 |
|  | 3.1 EXISTING SYSTEM | 3-5 |
|  | 3.2 DISADVANTAGES OF EXISTING SOLUTIONS | 5 |
|  | 3.3PROPOSED SYSTEM | 5 |
|  | 3.4ADVANTAGES OF PROPOSED SYSTEM | 6 |
| 4. | HARDWARE AND SOFTWARE REQUIREMENTS | 7 |
|  | 4.1 HARDWARE REQUIREMENTS | 7-9 |
|  | 4.2 SOFTWARE REQUIREMENTS | 10 |
| 5. | METHODOLOGY | 11 |
|  | 5.1 CONCEPTUAL DESIGN | 12 |
|  | 5.2 BLOCK DIAGRAM | 12 |
| 6. | ARCHITECTURE | 13 |
|  | 6.1 WORKING PRINCIPLE | 14 |
|  | 6.2 USECASE DIAGRAM | 15 |
| 7. | WEB APPLICATION OVERVIEW | 16-20 |
| 8. | IMPLEMENTATION | 21-24 |
| 9. | OUTPUT SCREENSHORTS | 25-26 |
| 10. | CONCLUSION | 27 |
| 11. | REFERENCES | 28 |

**1.INTRODUCTION**

A smart irrigation system uses advanced technology to control the watering process. It includes sensors that measure soil moisture levels and weather data, such as rainfall and temperature. By collecting this real-time information, the system can decide exactly when and how much water plants need. For example, if the soil is already moist or if rain is expected, the system can delay watering or reduce the amount of water used. This helps conserve water and ensures plants get the right amount of moisture to stay healthy.

The main goal of smart irrigation is to reduce water waste while improving plant health. Since water is a limited and valuable resource, using it wisely is very important, especially in areas facing drought or water shortages. Smart irrigation systems help farmers, gardeners, and landscapers save water and reduce their water bills. Moreover, by watering plants only when necessary, these systems promote stronger, healthier plants and reduce water.

Water is one of the most vital resources for growing plants in agriculture, gardening, and landscaping. However, traditional irrigation systems often water plants on fixed schedules without considering the actual needs of the plants or the environment. This can cause water to be wasted through overwatering or harm plants through underwatering. As water scarcity becomes an increasing global concern, it is essential to adopt smarter methods to manage water efficiently. Smart irrigation systems offer a promising solution by combining technology, data, and automation to optimize water use and improve plant health.

A smart irrigation system integrates sensors, weather monitoring, and automated controls to deliver water precisely when and where it is needed. Soil moisture sensors measure how much water is available in the soil, and weather stations provide information on rainfall, temperature, humidity, and sunlight. This real-time data allows the system to make informed decisions about irrigation timing and amounts. For example, if rain is forecasted, the system can delay watering, preventing unnecessary water use. If the soil is dry, it can increase watering to ensure plants receive adequate moisture.

One of the key benefits of smart irrigation is water conservation. Agriculture is one of the largest consumers of freshwater worldwide, and inefficient irrigation contributes to significant water waste. By applying water only when necessary, smart irrigation reduces water consumption, which is especially important in regions facing droughts or limited water supply. This efficient use of water helps protect natural ecosystems, supports sustainable farming practices, and reduces the strain on local water resources.

In addition to conserving water, smart irrigation promotes healthier plants. Overwatering can lead to root rot, fungal diseases, and nutrient loss, while underwatering can cause stress and poor growth. Because the system adjusts watering based on real-time plant needs, it maintains optimal soil moisture levels that support strong roots and healthy development. This results in higher crop yields for farmers and more vibrant landscapes for gardeners and landscapers.

Smart irrigation systems also offer economic advantages. Although the initial investment in sensors and automated controllers may be higher than traditional systems, the reduction in water use and improved plant health often lead to lower water bills and increased productivity over time. Many systems can be managed remotely via smartphones or computers, providing convenience and control that saves labor and time.

**2.LITERATURE SURVEY**

* Smart irrigation systems have emerged as vital tools for optimizing water usage in agriculture and landscaping by leveraging technology such as sensors, weather data, and automation. Several commercially available solutions demonstrate varying approaches and technologies to address water conservation and plant health management.
* **Netafim Smart Irrigation Solutions** are widely recognized for their precision drip irrigation technology, which delivers water directly to plant roots to minimize evaporation and runoff. The system integrates soil moisture sensors and weather-based automation, allowing real-time adjustments to irrigation schedules, thereby enhancing water efficiency. Additionally, remote control capabilities via mobile apps or web portals provide users with convenient monitoring. However, Netafim's solution has drawbacks, including complex installation requirements and ongoing maintenance to prevent clogging of drip lines. Moreover, the system’s scale and cost make it less practical for small residential gardens, limiting accessibility for small-scale users .
* **Weather-Based Smart Irrigation Systems**, represented by products like Rachio, Rain Bird, and Orbit B-hyve, rely heavily on real-time weather data to dynamically adjust watering schedules. These systems use predictive weather forecasts to avoid overwatering or underwatering and offer integration with home automation platforms such as Alexa and Google Home. Their user-friendly mobile applications facilitate remote control and monitoring these systems depend on the accuracy of weather forecasts, which can be imprecise at the microclimate level, potentially affecting irrigation effectiveness. Additionally, the upfront costs of hardware and installation are often high, which can be a barrier for some users.
* **Hydrawise by Hunter Industries** combines weather-based irrigation adjustments with remote management and detailed analytics of water usage, enabling users to optimize irrigation schedules effectively. While offering valuable data insights, Hydrawise shares common challenges with other weather-dependent systems, including the need for stable internet connectivity and higher costs, which may limit adoption among small-scale farmers and in resource-constrained areas.
* Despite the technological advancements, several gaps exist in current smart irrigation solutions. Accessibility remains a significant concern; many systems are expensive and complex, making them unsuitable for small-scale farmers or gardeners with limited budgets. Connectivity is another challenge, especially in rural or remote regions where stable internet access is unreliable or unavailable. Customization of irrigation schedules to accommodate diverse crops and soil conditions is often limited, reducing the adaptability of these systems to specific agricultural needs. Furthermore, many smart irrigation systems require continuous power supply, which can be problematic in off-grid or energy-limited environments. Lastly, the complexity of data presentation in these systems can overwhelm users without technical backgrounds, hindering timely and effective decision-making.

**3. SYSTEM ANALYSIS**

### EXISTING SYSTEM

**3.1.1NETAFIM (Smart Irrigation Solutions)**

**FEATURES:**

* **Precision Drip Irrigation**: Delivers water directly to plant roots, ensuring minimal evaporation and runoff.
* **Soil Moisture Sensors**: Measures soil moisture and adjusts irrigation based on real- time data.
* **Weather-based Automation**: Integrates weather data to prevent overwatering, optimizing water use.
* **Remote Control**: Can be monitored and controlled remotely using mobile apps or web portals.
* **Water Efficiency**: Designed to conserve water by minimizing evaporation, runoff, and deep percolation.



FIG.3.1.1: Netafim(Smart irrigation System)

**DISADVANTAGES:**

* **Complex Installation**: Requires professional setup, particularly for larger systems or those with specialized components like sensors.
* **Maintenance**: Drip lines and emitters can become clogged over time, requiring regular maintenance and cleaning.
* **Not Ideal for Small Gardens**: Overkill for residential yards unless you have a garden with high water needs (e.g., orchards or large agricultural plots).

## 3.1.2 WEATHER -BASED SMART IRRIGATION SYSTEM

* A weather-based smart irrigation system uses environmental sensors and weather forecasting data to determine the precise watering needs of crops or plants. Unlike traditional timers, this system adapts in real time to changes.
* **Examples:** Rachio, Rain Bird, Orbit B-hyve.

## FEATURES:

* + Use real-time weather data to adjust watering schedules.
  + Predict rain or drought conditions to prevent overwatering or underwatering.
  + Mobile app integration for monitoring and control.
  + Integration with home automation systems like Alexa or Google Home**.**



FIG.3.1.2: WEATHER -BASED SMART IRRIGATION SYSTEM

**DISADVANTAGES:**

* Relies heavily on accurate weather data; inaccuracies can affect performance.
* May not account for specific microclimates within a region.
* Expensive upfront costs for hardware and installation**.**

**3.1.3 HYDRAWISE BY HUNTER INDUSTRIES**

* Hydrawise by Hunter Industries is a sophisticated, cloud-based irrigation management platform designed to optimize water usage through smart technology. It caters to both homeowners and professional contractors, offering remote control, predictive scheduling, and real-time system monitoring

**FEATURES:**

* **Weather-Based Adjustments:** Automatic scheduling changes based on local weather conditions.
* **Remote Control:** Can be monitored and controlled via a mobile app.
* **Comprehensive Analytics:** Provides detailed water usage reports**.**

**3.2 DISADVANTAGES IN EXISTING SOLUTIONS:**

All the existing solutions have many disadvantages. The following are the gaps we found in those existing solutions:

* **Accessibility for Small-Scale Farms:** Many systems are costly and complex, making them unsuitable for small-scale farmers.
* **Connectivity Issues:** Solutions like GroGuru depend on stable internet, which is often lacking in rural areas.
* **Customization for Diverse Crops**: Existing systems offer limited customization for specific crop and soil needs.
* **Energy Dependence:** Many solutions require consistent power, which can be unsustainable in areas with limited resources.
* **User-Friendly Interfaces:** Complex data presentation makes it difficult for farmers to interpret and act on information quickly

### 3.3 PROPOSED SYSTEM

Develop a smart irrigation system that uses sensors and IoT technology to monitor soil moisture levels and weather conditions. This system will provide farmers with real-time data, helping them make informed decisions about irrigation scheduling. The goal is to optimize water usage and improve crop yield. The solution will include a smart irrigation module integrated with moisture sensors and a mobile app, delivering timely alerts and irrigation recommendations based on real-time data.

The proposed solution of IoT technology to monitor soil moisture levels and weather conditions continuously. This system provides farmers with real-time data through a mobile application, enabling them to make informed decisions about irrigation scheduling.

The core components include a smart irrigation module integrated with soil moisture sensors and weather data inputs. The system delivers timely alerts and personalized irrigation recommendations based on the real-time environmental data, ensuring water is used efficiently.

**The primary goals of the system are to:**

* Optimize water usage by irrigating only when necessary, preventing overwatering or underwatering.
* Improve crop yield and quality by maintaining optimal soil moisture conditions.
* Provide convenience through automated monitoring and remote control via the mobile app.

This IoT-enabled approach supports sustainable agriculture by conserving water resources and enhancing farm productivity.

### 3.4 ADVANTAGES OF PROPOSED SYSTEM

**1. Optimized Water Usage**  
 Delivers precise irrigation based on real-time soil moisture readings, preventing overwatering and underwatering.  
 Reduces water waste by irrigating only when necessary—vital in water-scarce regions.

**2. Increased Crop Yield and Quality**  
 Ensures crops receive the right amount of water at the optimal time.  
 Results in healthier plants, leading to better yield and improved produce quality.

**3. Cost Savings**  
 Lowers water and energy bills by eliminating unnecessary irrigation.  
 Reduces manual labor costs through automation and remote management.

**4. Real-time Monitoring and Control**  
 Allows farmers to track soil moisture and weather data anytime via a mobile app.  
 Provides instant alerts and smart recommendations for quick, informed decision-making.

**5. Weather-Adaptive Scheduling**  
 Integrates with weather forecasts to postpone irrigation when rain is anticipated.  
 Prevents overwatering and protects crops from moisture-related damage.

**6. Automation and Customization**  
 Supports automated irrigation based on sensor thresholds or scheduled timings.  
 Easily customizable for different crops, soil types, and climatic conditions.

**7. Data-Driven Farming**  
 Collects historical data for long-term analysis and strategic planning.  
 Enables precision agriculture, enhancing farm productivity over time.

**8. Environmentally Friendly**  
 Promotes sustainable farming by conserving water and reducing runoff.  
 Minimizes fertilizer leaching and soil erosion, supporting ecosystem health.

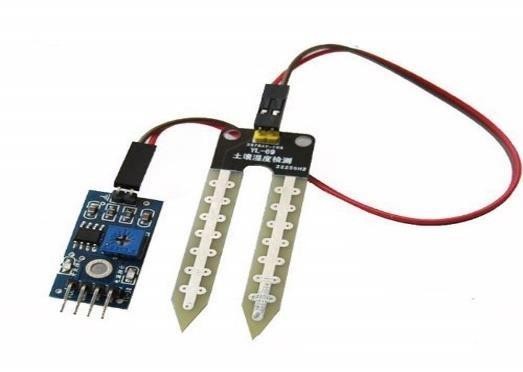
**4.HARDWARE AND SOFTWARE REQUIREMENTS**

**4.1 HARDWARE REQUIREMENTS**

**SOIL MOISTURE SENSOR:**

Soil moisture sensors measure the volumetric [water content](https://en.wikipedia.org/wiki/Water_content) in [soil](https://en.wikipedia.org/wiki/Soil)[.[1]](https://en.wikipedia.org/wiki/Soil_moisture_sensor#cite_note-1) Since the direct [gravimetric](https://en.wikipedia.org/wiki/Gravimetric_analysis) [measurement](https://en.wikipedia.org/wiki/Gravimetric_analysis) of free soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with [neutrons](https://en.wikipedia.org/wiki/Neutron), as a proxy for the moisture content

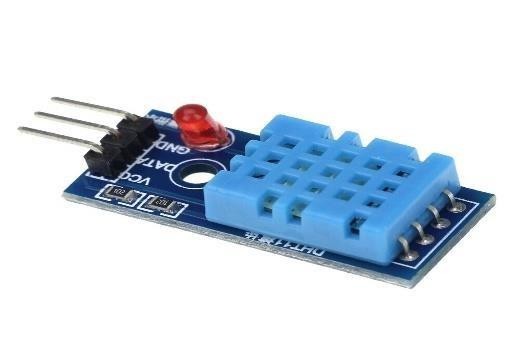
### 



### DHT 11:

Humidity is the measure of water vapour present in the air. The level of humidity in air affects various physical, chemical and biological processes. In industrial applications, humidity can affect the business cost of the products, health and safety of the employees

### 



**NODEMCU ESP8266**:

**NodeMCU** is a low-cost open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things) platform.[[4](https://en.wikipedia.org/wiki/NodeMCU#cite_note-nodemcu_sdhn_Kanchan_firmware-4)[][5]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-5) It initially included [firmware](https://en.wikipedia.org/wiki/Firmware) which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) [SoC](https://en.wikipedia.org/wiki/System_on_a_chip) from [Espressif](https://en.wikipedia.org/wiki/Espressif) Systems, and hardware which was based on the ESP- 12 module[.[6](https://en.wikipedia.org/wiki/NodeMCU#cite_note-Espressif_Systems-6)[][7]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-7) Later, support for the [ESP32](https://en.wikipedia.org/wiki/ESP32) 32-bit MCU was added.



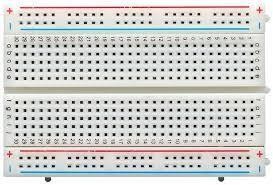
**RELAY MODULE :**

Relay modules are simply circuit boards that house one or more relays. They come in a variety of shapes and sizes but are most commonly rectangular with 2, 4, or 8 relays mounted on them, sometimes even up to 16 relays.



**BREADBOARD:**

A breadboard, solderless breadboard, or protoboard is a construction base used to build semi-permanent prototypes of electronic circuits. Unlike a perfboard or stripboard, breadboards do not require soldering or destruction of tracks and are hence reusable. A construction base used for prototyping and testing circuits without soldering. It allows easy connections of components using jumper wires



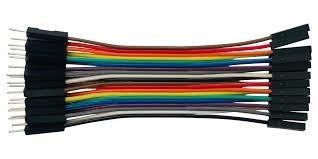
**PUMP:**

A **pump** is a device that moves fluids ([liquids](https://en.wikipedia.org/wiki/Liquid) or [gases](https://en.wikipedia.org/wiki/Gas)), or sometimes [slurries](https://en.wikipedia.org/wiki/Slurry)[,[1]](https://en.wikipedia.org/wiki/Pump#cite_note-SlurryPump-1) by mechanical action, typically converted from electrical energy into hydraulic energy.



**JUMPER WIRES:**

Wires used to make connections between different components on the breadboard and the Arduino Uno.Jumper wires are short, flexible, conductive wires used to make temporary connections in electrical circuits without soldering. They are commonly used with breadboards and other prototyping tools, allowing for easy modifications and testing of circuits. Jumper wires come in various types, including male-to-male, male-to-female, and female-to-female, depending on the connector pins at each end.



**BATTERY:**



**4.2 SOFTWARE REQUIREMENTS:**

The software ecosystem for this system includes tools for programming the microcontroller and simulating or extending the functionality:

**Arduino IDE:**

The Arduino IDE (Integrated Development Environment) is a free, open-source software designed for writing, compiling, and uploading code to Arduino boards. It's based on the Processing programming language and provides a user-friendly interface for interacting with Arduino hardware.

**Using the IDE:**

**Installation:** Download and install the [Arduino IDE](https://www.arduino.cc/en/software) on your computer.

**Open a Sketch:** Open a new sketch or open an existing one from your sketchbook.

**Write Code:** Write your Arduino code in the code editor.

**Verify:** Use the "Verify" button to check your code for syntax errors.

**Upload:** Use the "Upload" button to compile and upload your code to the Arduino board.

**Serial Monitor:** Use the serial monitor to interact with your Arduino board and view output.

**Benefits of Using the Arduino IDE:**

**Ease of Use:** The IDE is designed to be user-friendly, making it accessible to beginners.

**Cross-Platform Compatibility:** The IDE is available for Windows, macOS, and Linux.

**Open-Source and Free:** The IDE is free to download and use.

**Large Community Support:** The Arduino community offers a wealth of resources and support for users.

**5.METHODOLOGY**

The methodology for CropMitra involves a systematic approach to integrating hardware and software for smart farming. First, the project identifies farmer needs, such as soil moisture monitoring and irrigation automation, and selects cost-effective hardware like soil sensors, ESP8266 modules, relays, and pumps. The hardware is assembled and programmed to measure soil moisture and control the pump based on predefined thresholds. Simultaneously, a web application is developed using HTML, CSS, and JavaScript, featuring a login page, real-time dashboard, and summary page. The ESP8266 communicates sensor data to the web application, which updates dynamically without requiring a database. After integration, the system undergoes rigorous testing in simulated and real farming environments to ensure reliability and ease of use. This lightweight, scalable approach ensures CropMitra is accessible, user-friendly, and effective for modern agricultural needs.

**5.1CONCEPTUAL DESIGN:**

Our design includes very simple mechanism. The design of the prototype is as shown below in the figure

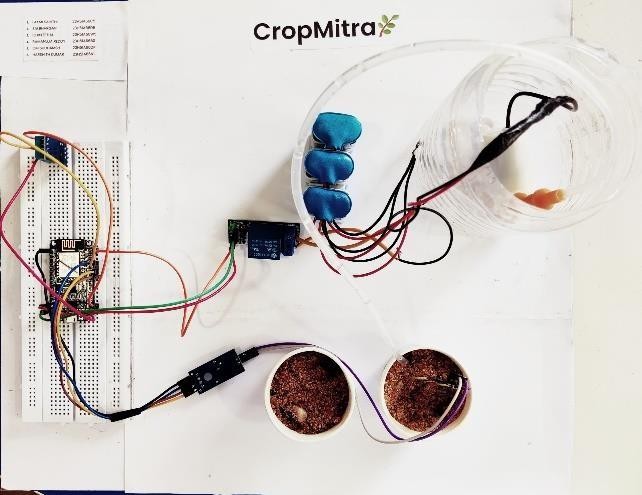


FIG.5.1: PROTOTYPE

The conceptual design of CropMitra revolves around creating an IoT-based smart farming system that integrates hardware and software for efficient irrigation and real-time monitoring. The hardware setup includes a soil moisture sensor to measure soil conditions, an ESP8266 Wi-Fi module to enable data transmission, a relay module to control a water pump, and a power source for the components. This hardware operates on a predefined moisture threshold, automating the pump's operation to maintain optimal soil conditions. The software complements the hardware with a web application developed using HTML, CSS, and JavaScript. The application provides a login interface, a dashboard for real-time data visualization, and a summary page for user details. It dynamically fetches sensor data from the ESP8266, eliminating the need for complex databases. Designed to be user-friendly, responsive, and lightweight, CropMitra simplifies irrigation management, making it accessible to farmers with minimal technological resources.CropMitra is a smart agricultural solution that combines hardware and software to help farmers monitor and manage crop health efficiently. The system integrates IoT technology to collect real- time environmental and soil data, automating irrigation processes and providing actionable insights through a responsive web application. Its primary goal is to optimize water usage, reduce manual intervention, and ensure healthy crop growth.

The hardware design consists of essential components like a soil moisture sensor, relay module, ESP8266 Wi-Fi module, and a water pump. The soil moisture sensor continuously monitors the moisture levels in the soil, while the ESP8266 module facilitates communication between the hardware and the cloud server. When the moisture level falls below a set threshold, the relay module activates the water pump to irrigate the crops automatically. Once optimal moisture is achieved, the pump shuts off, conserving water and energy. All components are powered by a stable power source, with provisions for a solar-powered alternative for greater sustainability.

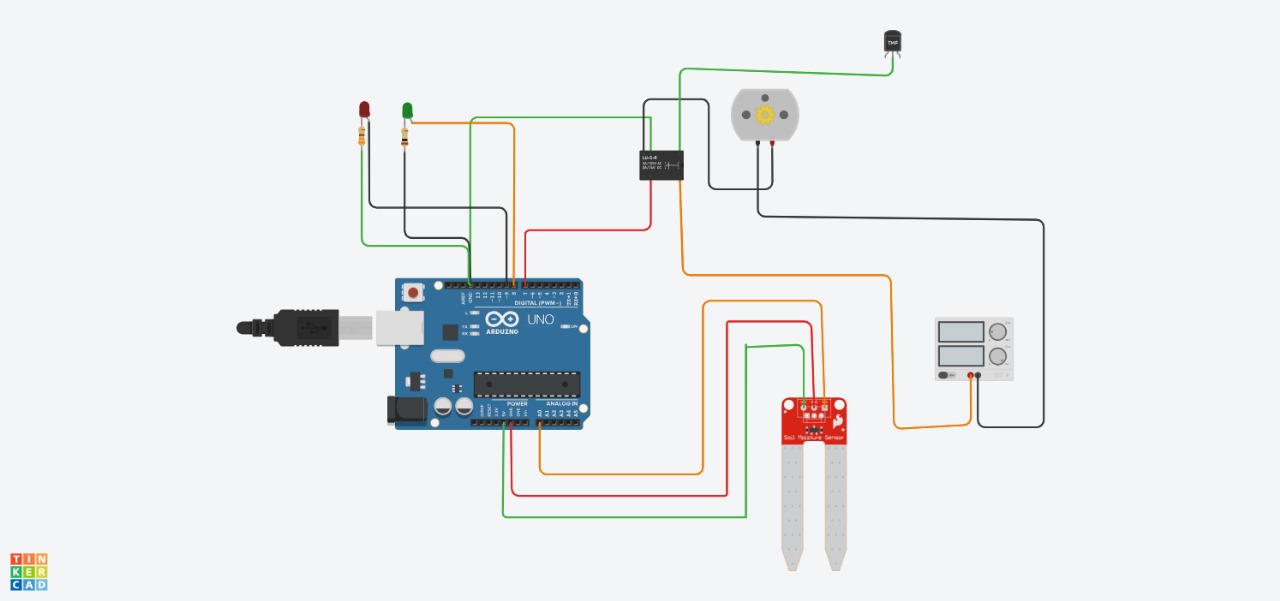
**5.2 BLOCK DIAGRAM:**

****

FIG5.2.: BLOCK DIAGRAM

A Smart irrigation system that automates watering based on soil moisture levels. It consists of three main components: a soil moisture sensor, a NodeMCU ESP8266 microcontroller, and a pump or motor. The soil moisture sensor, placed in the soil, continuously monitors the moisture content and sends the data to the NodeMCU ESP8266. Acting as the controller, the NodeMCU processes this input and determines whether the soil needs watering by comparing the moisture level to a predefined threshold. If the soil is found to be dry, the controller activates the pump or motor, which then irrigates the soil. Once the moisture level reaches the desired value, the controller stops the pump. This setup helps in conserving water and reducing manual effort by automating the irrigation process based on real-time soil conditions.

**6.ARCHITECTURE DIAGRAMS**

****

* **Arduino UNO** (central microcontroller):Controls the entire system by reading sensor data and triggering outputs like LEDs or the water pump.
* **Soil Moisture Sensor**:

Detects the moisture level in the soil.

Connected to the analog pin of the Arduino to provide continuous readings.

When the soil is dry, the sensor triggers watering.

* **DC Water Pump (or motor)**:

Waters the plants when soil moisture is low.

Controlled via a **transistor or relay module** that switches the pump ON/OFF based on signals from Arduino.

* **Relay Module**:

Acts as a switch between the pump and power source.

Triggered by the Arduino based on soil sensor input.

* **TMP (Temperature Sensor)**:

Measures temperature; can be used to enhance irrigation logic based on environmental conditions.

Connected to an analog pin of the Arduino.

* **LEDs (Red and Green)**:

Indicate the status of the soil moisture.

* **Green LED**:

Soil has enough moisture.

* **Red LED**:

Soil is dry; irrigation is needed or pump is on.

* **Power Supply Module**:

Provides stable power to the Arduino and the motor circuit (through relay).

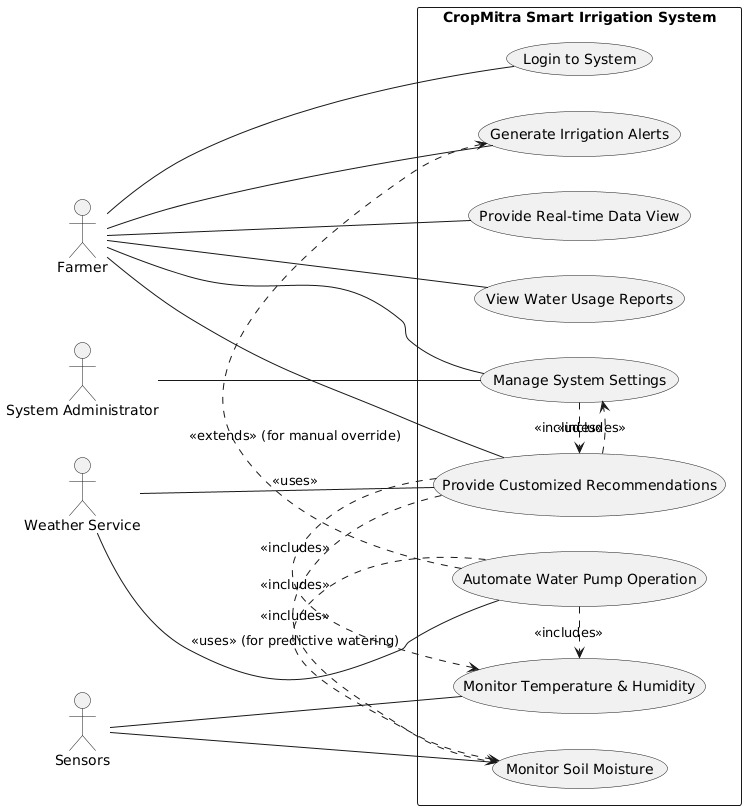
* **LCD Display** (Optional here based on image details):

Not clearly active in this image, but may be used to show real-time sensor readings like moisture or temperature

**6.1 Working Principle:**

1. The soil moisture sensor checks the soil condition.
2. If the soil is dry, the Arduino:
   1. Turns ON the water pump via the relay.
   2. Lights up the red LED as a warning.
3. If the soil is wet, the system:
   1. Keeps the pump OFF.
   2. Lights up the green LED to indicate good condition.
4. Optionally, **temperature readings** can be displayed or used to optimize irrigation.

**6.2 USECASE DIAGRAM**



**7 .WEB APPLICATION OVERVIEW**

The software for CropMitra is built using lightweight and accessible technologies, including HTML, CSS, and JavaScript, to create an intuitive, responsive, and user-friendly web application. The application is designed to display real-time data from the hardware components and provide farmers with actionable insights for efficient crop management.

**The frontend is structured around three key pages:**

**1.Login Page:**

The system allows users to securely access the platform, meaning it includes an authentication mechanism (such as a login form) where users enter their credentials (like a username and password) to verify their identity. This helps protect the platform and user data from unauthorized access. The interface itself is designed to be simple and clean, built using HTML to structure the content and CSS to style and enhance its visual appearance, ensuring it's user-friendly and easy to navigate. Additionally, JavaScript is incorporated to handle form validation (e.g., checking if required fields are filled, email format is correct, password meets criteria, etc.) and session management (e.g., keeping the user logged in, handling logouts, and preventing access to restricted pages without login). Altogether, this ensures a secure, responsive, and user-friendly login experience.

* Allows users to securely access the platform.
* A simple, clean interface built with HTML and styled using CSS.
* Incorporates JavaScript for validation and session management.



FIG 7.1:LOGIN PAGE

1. **Dashboard:**

The **Dashboard** serves as the central interface for users to monitor the smart irrigation system. It provides a real-time overview of key data, such as the **soil moisture levels** detected by sensors and the **status of the water pump** (whether it is currently operating or idle). This live data helps users make informed decisions or track how the system is performing over time To make the information clear and easy to understand, the dashboard uses **interactive charts and visual indicators** developed with **JavaScript**. These visuals not only display current values but also help highlight trends and patterns, like how moisture levels change throughout the day or how often the pump activates. The dashboard is also equipped with **automatic updates**, meaning it fetches fresh data directly from the **ESP8266 microcontroller** without requiring manual refreshes. This ensures that users always see the most up-to-date information, making the dashboard a powerful tool for both monitoring and managing the irrigation process efficiently.

* Displays real-time data from sensors, including soil moisture levels and pump status.
* Features dynamic charts and visual indicators, created with JavaScript, to help users analyze trends.
* Automatically updates in real-time as sensor data is fetched from the ESP8266 module

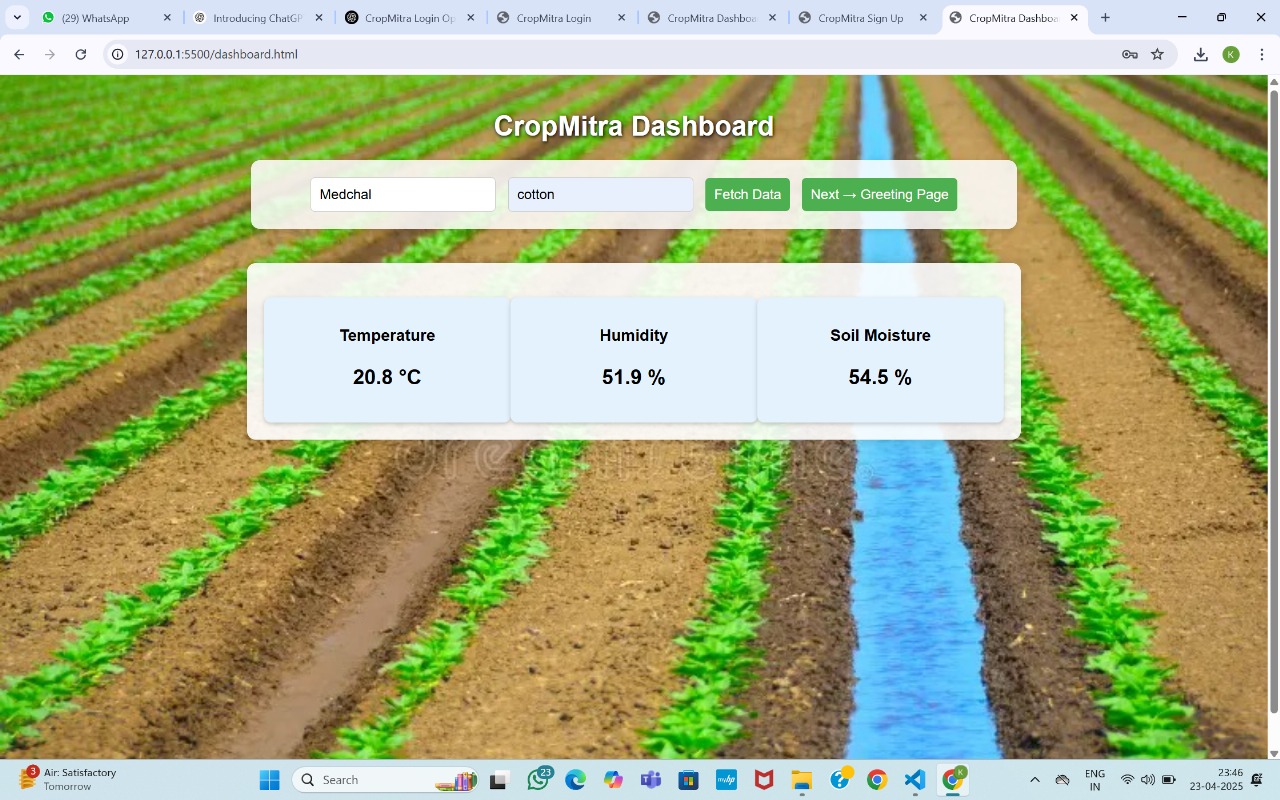
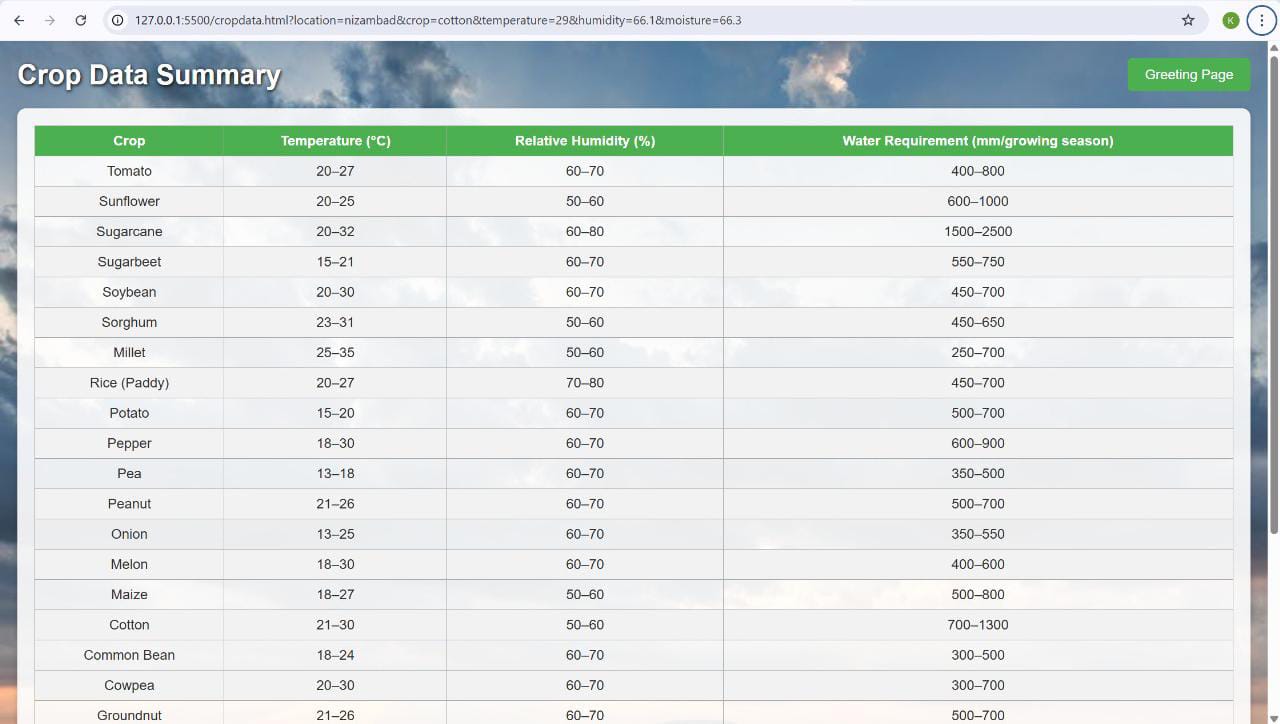


FIG.7..2:DASHBOAED



The image shows a displaying a summary table of different crops and their preferred growing conditions. Each crop is listed with its ideal temperature range, relative humidity range, and water requirements for the growing season. The table helps users quickly compare different crops and see which conditions they need to thrive.

Temperature range (°C) suitable for growth,

Relative Humidity (%) range they prefer,

Water Requirement (mm/growing season).

**3.Greetings Page**:

The **Greetings Page** in the smart irrigation system acts as a welcoming interface that appears when a user logs in or opens the application. It typically **summarizes user details**, such as their name, location, or system preferences, along with a brief overview of **previously displayed information**, like the last recorded soil moisture level or recent irrigation activity. This helps users quickly get oriented with the current state of their system.

* Summarizes user details and previously displayed information.
* Designed for a personalized user experience with a visually appealing layout.

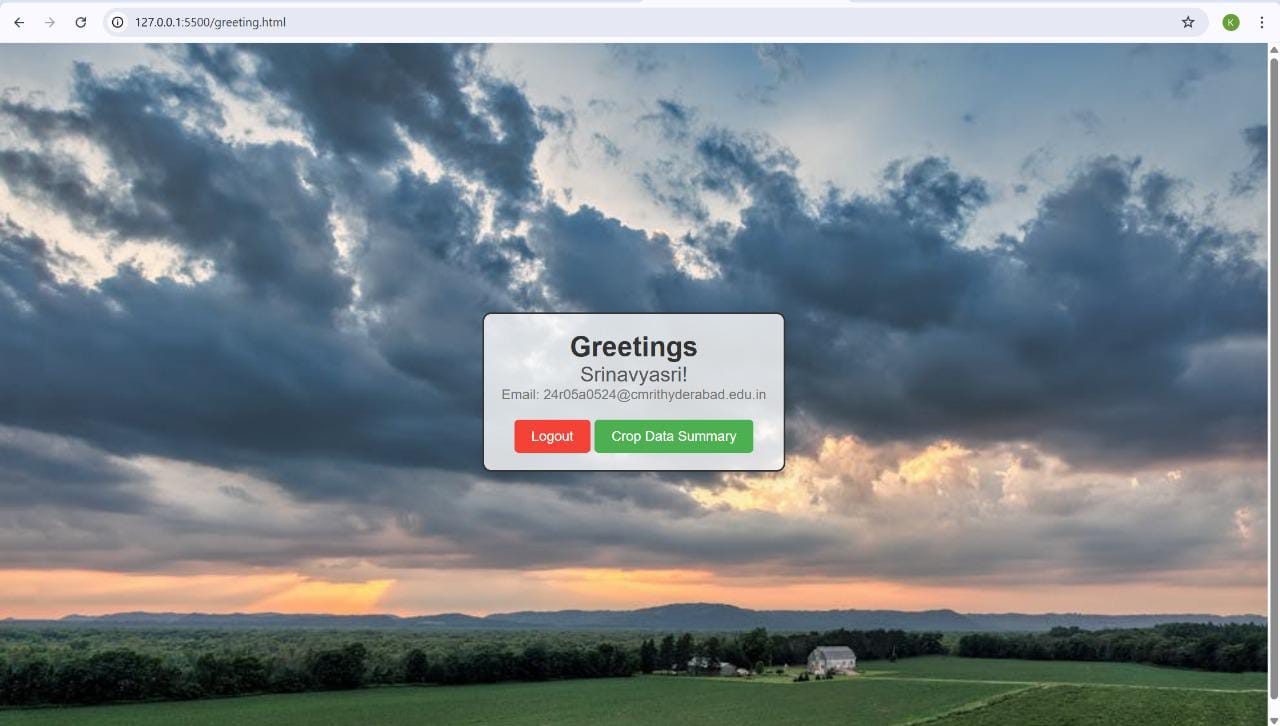


FIG:7.3 GREETINGS

**4.SIGN UP:**

A sign-up page in CropMitra, which appears to be part of a smart irrigation system. This page is designed to register new users by collecting essential information such as Username, Email, Location, Password, and Confirm Password. The page gathers essential information like Username, Email, Location, and Password, which helps identify and authenticate users uniquely. The Location field is especially important in a smart irrigation system because it can be used to fetch local weather data and tailor irrigation schedules according to regional conditions. Password protection ensures that only authorized users can monitor or control irrigation devices like pumps or sensors.

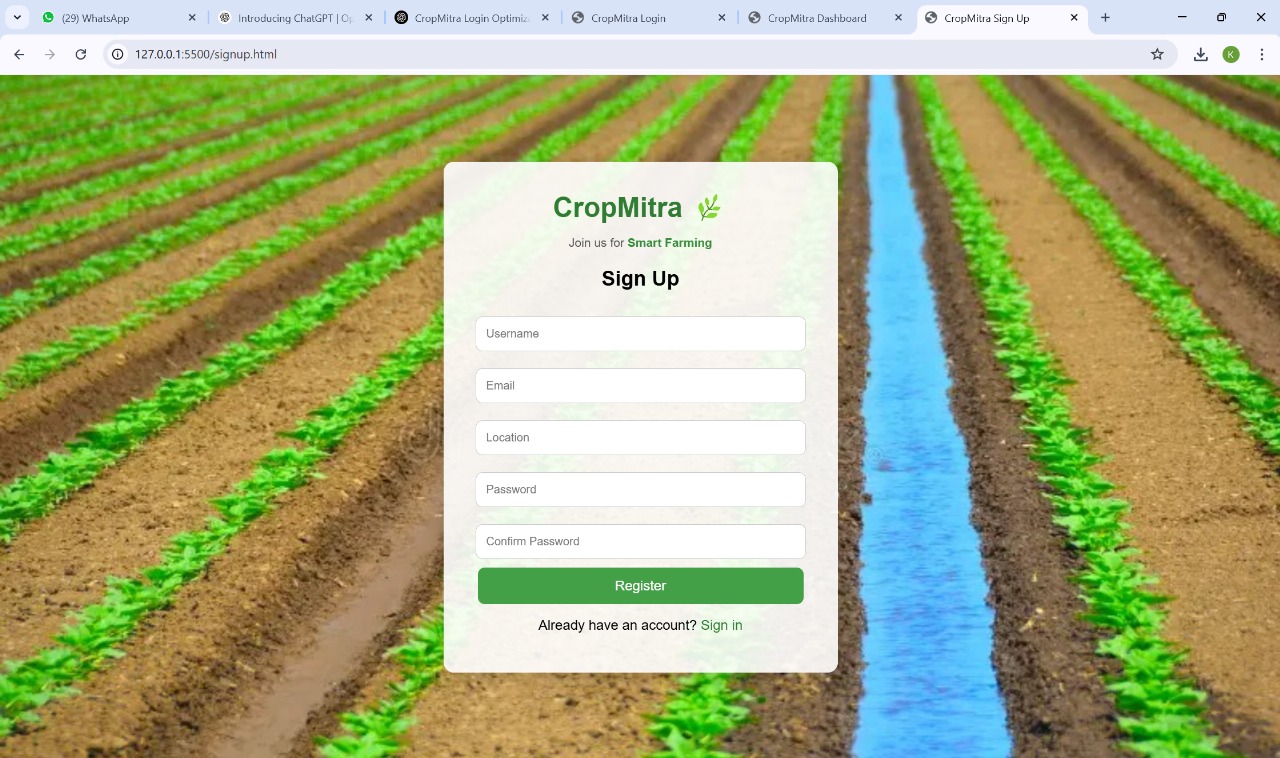


FIG:7.4: SIGN UP

During the design phase, tools like Figma were used to prototype and refine the user interface, ensuring the application is both aesthetically pleasing and easy to navigate. The layouts are responsive, enabling seamless usage across mobile devices, tablets, and desktops.

JavaScript is used to handle real-time communication with the ESP8266 module. It dynamically fetches sensor data and updates the dashboard without requiring page reloads, providing a smooth user experience. Local storage is employed for temporarily saving user preferences and session details, eliminating the need for a complex database system and keeping the application lightweight.

By leveraging basic but effective technologies, the software for CropMitra ensures a balance between simplicity and functionality. This approach allows for easy deployment, minimal maintenance, and accessibility, making it ideal for use in farming environments with limited technological resources.

**8.IMPLEMENTATION**

#include <DHT.h>

#include <ESP8266WiFi.h>

// Pin Definitions

#define DHTPIN 2 // GPIO2 (D4)

#define DHTTYPE DHT11

#define MOISTURE\_SENSOR\_PIN A0

#define RELAY\_PIN 5 // GPIO5

#define DHT\_POWER\_PIN 16 // GPIO16 (D0)

// WiFi Credentials

const char\* ssid = "ESP8266\_Default";

const char\* password = "123456789";

// Moisture thresholds

const int MOISTURE\_THRESHOLD\_VERY\_WET = 200;

const int MOISTURE\_THRESHOLD\_WET = 400;

const int MOISTURE\_THRESHOLD\_DRY = 600;

const int MOISTURE\_THRESHOLD\_VERY\_DRY = 800;

// Temperature thresholds

const float TEMPERATURE\_THRESHOLD\_HIGH = 35.0;

const float TEMPERATURE\_THRESHOLD\_LOW = 15.0;

// Timing

const unsigned long SENSOR\_READ\_INTERVAL = 5000; // 5 seconds

const unsigned long WIFI\_TIMEOUT = 30000; // 30 seconds

const unsigned long MAX\_VALVE\_OPEN\_TIME = 300000; // 5 minutes

const unsigned long DHT\_RETRY\_DELAY = 2000; // 2 seconds

const int SERIAL\_BAUD\_RATE = 115200;

// DHT setup

DHT dht(DHTPIN, DHTTYPE);

unsigned long lastReadTime = 0;

unsigned long lastDHTRetry = 0;

unsigned long valveOpenTime = 0;

bool valveIsOpen = false;

int dhtErrorCount = 0;

const int DHT\_MAX\_ERRORS = 5;

void setup() {

Serial.begin(SERIAL\_BAUD\_RATE);

delay(2000);

Serial.println("\nInitializing system...");

pinMode(DHT\_POWER\_PIN, OUTPUT);

pinMode(RELAY\_PIN, OUTPUT);

pinMode(DHTPIN, INPUT\_PULLUP);

digitalWrite(DHT\_POWER\_PIN, HIGH); // Power on DHT

digitalWrite(RELAY\_PIN, LOW); // Ensure valve is closed

Serial.println("Initializing DHT sensor on D4 (GPIO2)...");

dht.begin();

delay(2000); // Give DHT sensor time to stabilize

float testTemp = dht.readTemperature();

float testHum = dht.readHumidity();

if (isnan(testTemp) || isnan(testHum)) {

Serial.println("DHT Sensor Test Failed! Please check:");

Serial.println("1. DHT power connection (VCC)");

Serial.println("2. DHT data connection (D4/GPIO2)");

Serial.println("3. 4.7k-10k pull-up resistor between data and VCC");

} else {

Serial.println("DHT Sensor Test Successful!");

Serial.print("Initial Temperature: "); Serial.print(testTemp); Serial.println("°C");

Serial.print("Initial Humidity: "); Serial.print(testHum); Serial.println("%");

}

if (setupWiFi()) {

Serial.println("System initialization complete!");

} else {

Serial.println("WiFi connection failed! Continuing without WiFi...");

}

}

void loop() {

unsigned long currentTime = millis();

if (currentTime - lastReadTime >= SENSOR\_READ\_INTERVAL) {

readAndProcessSensors();

lastReadTime = currentTime;

}

if (valveIsOpen && (currentTime - valveOpenTime >= MAX\_VALVE\_OPEN\_TIME)) {

closeValve("Maximum valve open time reached");

}

}

void readAndProcessSensors() {

int moistureValue = readMoistureSensor();

float temperature = NAN;

float humidity = NAN;

if (millis() - lastDHTRetry >= DHT\_RETRY\_DELAY) {

temperature = dht.readTemperature();

humidity = dht.readHumidity();

lastDHTRetry = millis();

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

dhtErrorCount++;

if (dhtErrorCount >= DHT\_MAX\_ERRORS) {

Serial.println("Critical DHT Error! Please check wiring:");

Serial.println("- Verify DHT power is connected to 3.3V");

Serial.println("- Check data pin connection (D4/GPIO2)");

Serial.println("- Ensure pull-up resistor is present");

Serial.print("- Current error count: "); Serial.println(dhtErrorCount);

}

} else {

dhtErrorCount = 0;

}

}

printSensorValues(moistureValue, temperature, humidity);

controlRelay(moistureValue, temperature);

}

int readMoistureSensor() {

int sum = 0;

for (int i = 0; i < 5; i++) {

sum += analogRead(MOISTURE\_SENSOR\_PIN);

delay(100);

}

return sum / 5;

}

void printSensorValues(int moisture, float temperature, float humidity) {

Serial.println("\n--- Sensor Readings ---");

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

if (moisture < MOISTURE\_THRESHOLD\_VERY\_WET) {

Serial.println("DANGEROUSLY WET)");

} else if (moisture < MOISTURE\_THRESHOLD\_WET) {

Serial.println("WET)");

} else if (moisture > MOISTURE\_THRESHOLD\_VERY\_DRY) {

Serial.println("DANGEROUSLY DRY)");

} else if (moisture > MOISTURE\_THRESHOLD\_DRY) {

Serial.println("DRY)");

} else {

Serial.println("OPTIMAL)");

}

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

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

} else {

Serial.print("Temperature: "); Serial.print(temperature); Serial.println("°C");

Serial.print("Humidity: "); Serial.print(humidity); Serial.println("%");

}

Serial.println();

}

void openValve(const char\* reason) {

if (!valveIsOpen) {

digitalWrite(RELAY\_PIN, HIGH);

valveIsOpen = true;

valveOpenTime = millis();

Serial.print("Valve Status: OPEN ("); Serial.print(reason); Serial.println(")");

}

}

void closeValve(const char\* reason) {

if (valveIsOpen) {

digitalWrite(RELAY\_PIN, LOW);

valveIsOpen = false;

Serial.print("Valve Status: CLOSED ("); Serial.print(reason); Serial.println(")");

}

}

void controlRelay(int moisture, float temperature) {

if (isnan(temperature)) {

closeValve("Temperature sensor error");

return;

}

if (temperature > TEMPERATURE\_THRESHOLD\_HIGH) {

openValve("High temperature alert");

return;

}

if (temperature < TEMPERATURE\_THRESHOLD\_LOW) {

closeValve("Low temperature alert");

return;

}

if (moisture > MOISTURE\_THRESHOLD\_VERY\_DRY) {

openValve("Emergency: Soil critically dry");

} else if (moisture > MOISTURE\_THRESHOLD\_DRY) {

openValve("Soil needs water");

} else if (moisture < MOISTURE\_THRESHOLD\_VERY\_WET) {

closeValve("Emergency: Soil critically wet");

} else if (moisture < MOISTURE\_THRESHOLD\_WET) {

closeValve("Soil sufficiently watered");

}

}

bool setupWiFi() {

unsigned long startAttemptTime = millis();

Serial.print("Connecting to WiFi: "); Serial.println(ssid);

WiFi.begin(ssid, password);

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

millis() - startAttemptTime < WIFI\_TIMEOUT) {

delay(500);

Serial.print(".");

}

if (WiFi.status() == WL\_CONNECTED) {

Serial.println("\nWiFi connected successfully!");

Serial.print("IP Address: "); Serial.println(WiFi.localIP());

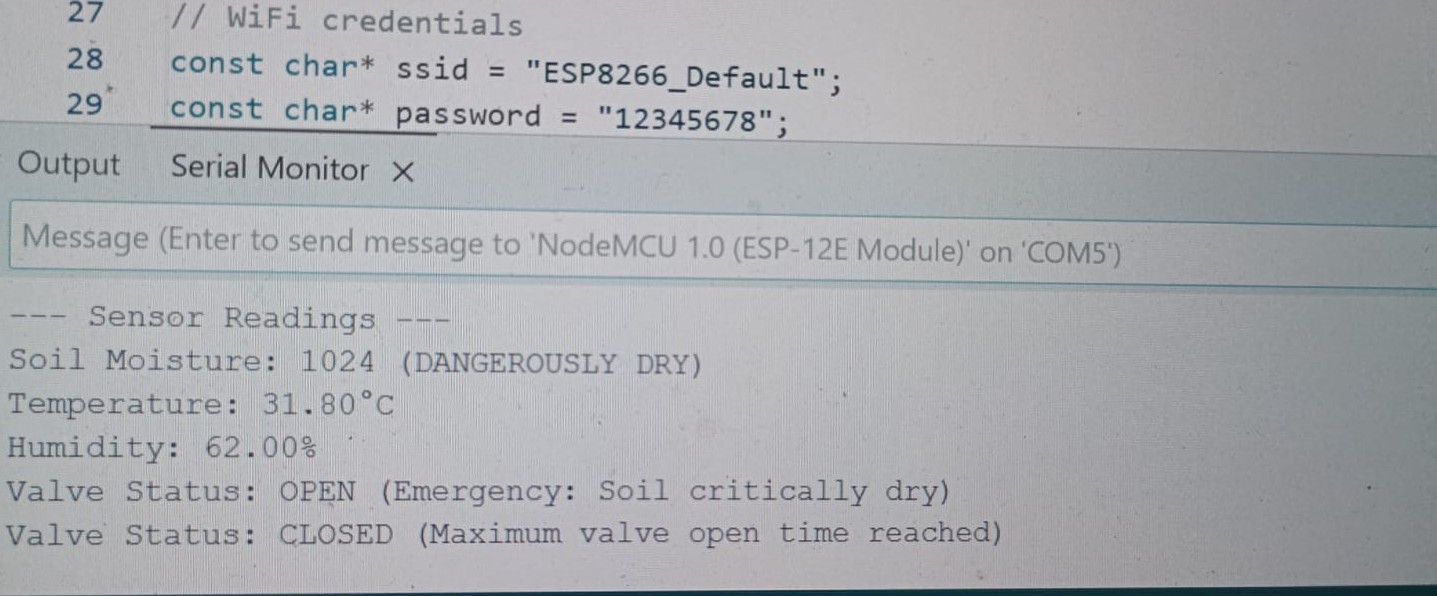
return true;

}

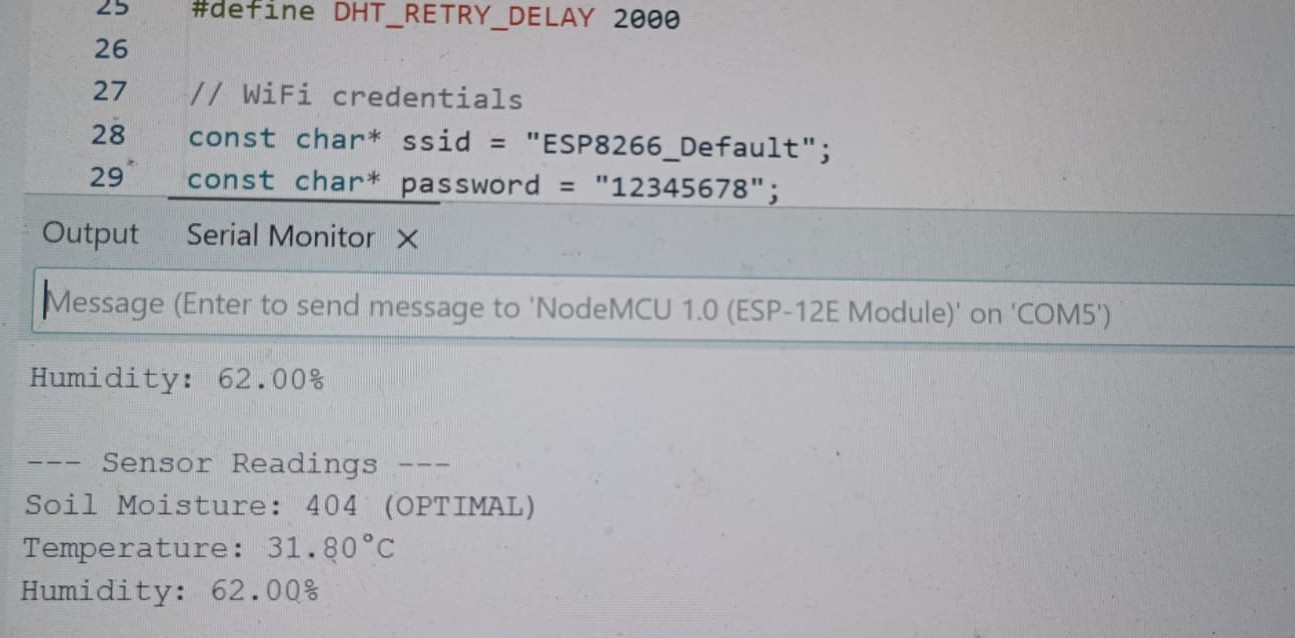
return false;

}

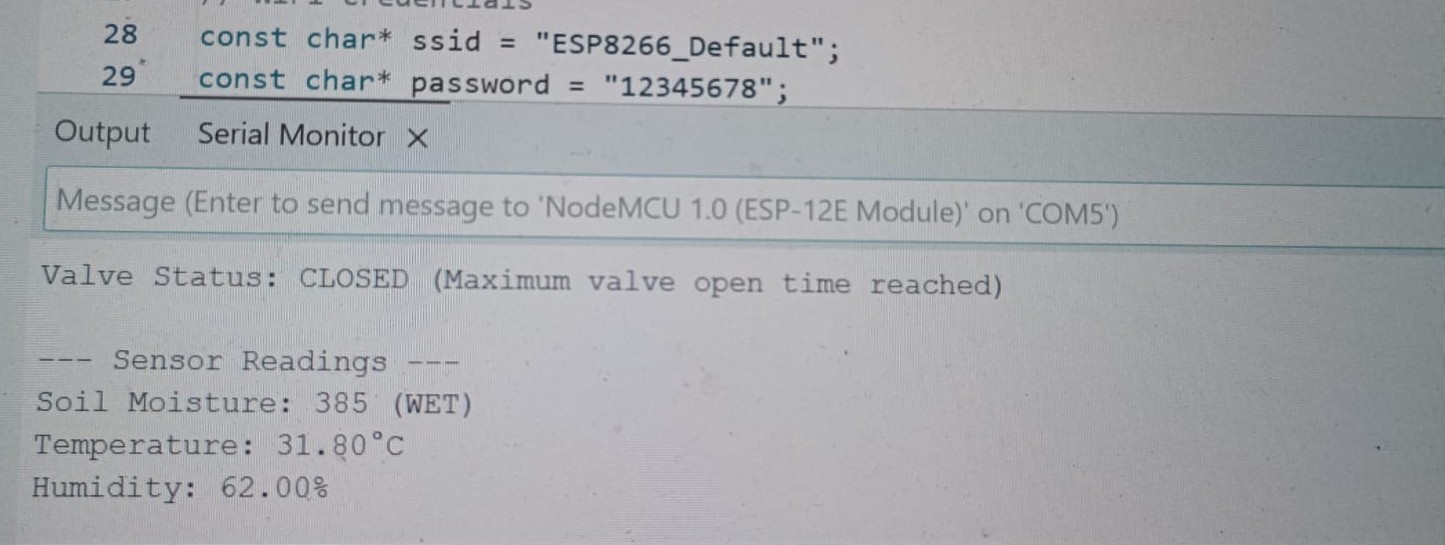
**8.OUTPUT SCREENSHOTS**

****

**FIG:8.1. SOIL DRY**

****

**FIG:8.2. SOIL OPTIMAL**

****

**FIG:8.3 SOIL WET**

**9.CONCLUSION**

A Smart Irrigation System designed for precision farming presents a powerful tool to transform traditional agricultural practices into more efficient, sustainable, and profitable operations. By optimizing water usage through precise monitoring and control, these systems help farmers reduce unnecessary water wastage, conserve precious resources, and contribute positively to environmental sustainability. The ability to gather and analyze real-time data on soil moisture, weather conditions, and crop needs enables farmers to make informed decisions, leading to improved crop health and increased yields. Automation further enhances operational efficiency by ensuring irrigation is applied only when and where it is needed, reducing labor costs and minimizing human error.

Despite these significant advantages, the adoption of smart irrigation technology is not without its challenges. The initial investment required for purchasing, installing, and integrating sensors, communication networks, and control systems can be substantial, especially for small-scale farmers or those in developing regions. Additionally, ongoing maintenance and technical support are necessary to ensure the system’s reliability and accuracy over time. Dependence on stable internet connectivity can also pose problems in remote or underserved areas, potentially limiting the system’s effectiveness. Furthermore, as these systems collect and transmit sensitive agricultural data, concerns about data privacy and cybersecurity must be carefully managed to protect farmers’ information.

Nonetheless, with thoughtful planning, adequate training, and support infrastructure, these challenges can be mitigated. Farmers who invest in smart irrigation systems stand to gain not only through cost savings and yield improvements but also by positioning themselves at the forefront of sustainable agriculture. As the global demand for food production intensifies amidst growing environmental concerns and water scarcity, precision irrigation technologies will play an increasingly vital role in ensuring food security while preserving natural resources.

Ultimately, while smart irrigation systems require careful allocation of resources and technical expertise, their long-term benefits far outweigh the challenges. They offer a path toward more resilient, sustainable, and profitable farming, making them invaluable assets for farmers committed to advancing agricultural innovation and environmental.

**10.REFERENCES**

**APPENDIX**

* + - [**https://ieeexplore.ieee.org/document/9239397/**](https://ieeexplore.ieee.org/document/9239397/)
    - [**https://www.researchgate.net/publication/367398621\_IoT\_Smart\_Irrigation\_System\_for\_Preci**](https://www.researchgate.net/publication/367398621_IoT_Smart_Irrigation_System_for_Precision_Agriculture)[**sion\_Agriculture**](https://www.researchgate.net/publication/367398621_IoT_Smart_Irrigation_System_for_Precision_Agriculture)

**REFERENCES:**

* + - [**https://ieeexplore.ieee.org/document/9239397/**](https://ieeexplore.ieee.org/document/9239397/)
    - [**https://www.researchgate.net/publication/367398621\_IoT\_Smart\_Irrigation\_System\_for\_Preci**](https://www.researchgate.net/publication/367398621_IoT_Smart_Irrigation_System_for_Precision_Agriculture)[**sion\_Agriculture**](https://www.researchgate.net/publication/367398621_IoT_Smart_Irrigation_System_for_Precision_Agriculture)
    - [**https://www.mdpi.com/1424-8220/23/4/2091**](https://www.mdpi.com/1424-8220/23/4/2091)
    - [**https://iopscience.iop.org/article/10.1088/1755-1315/1083/1/012066**](https://iopscience.iop.org/article/10.1088/1755-1315/1083/1/012066)