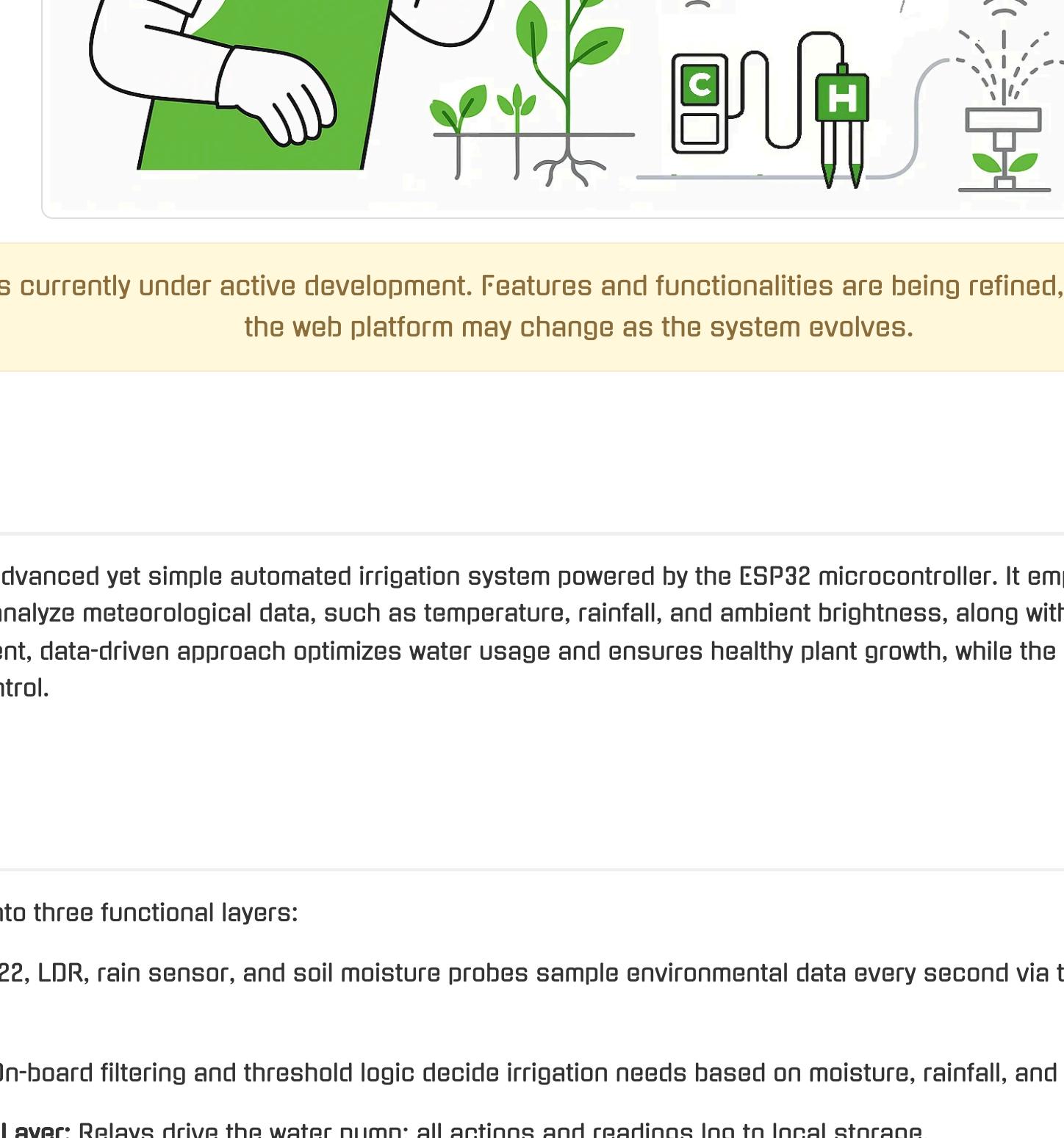


Smart Irrigation System

Overview Concept Key Functions Materials Required Wiring Diagram Connection Details

Upload Requirements



Notice: This project is currently under active development. Features and functionalities are being refined, and the concept for hosting the web platform may change as the system evolves.

Overview

This project presents an advanced yet simple automated irrigation system powered by the ESP32 microcontroller. It employs real-time environmental monitoring to collect and analyze meteorological data, such as temperature, rainfall, and ambient brightness, along with plant-specific metrics such as soil moisture. This intelligent, data-driven approach optimizes water usage and ensures healthy plant growth, while the system's web platform facilitates remote monitoring and control.

Concept

The system is organized into three functional layers:

- Sensing Layer:** DHT22, LDR, rain sensor, and soil moisture probes sample environmental data every second via the ESP32's inputs.
- Processing Layer:** On-board filtering and threshold logic decide irrigation needs based on moisture, rainfall, and ambient data.
- Actuation & Control Layer:** Relays drive the water pump; all actions and readings log to local storage.

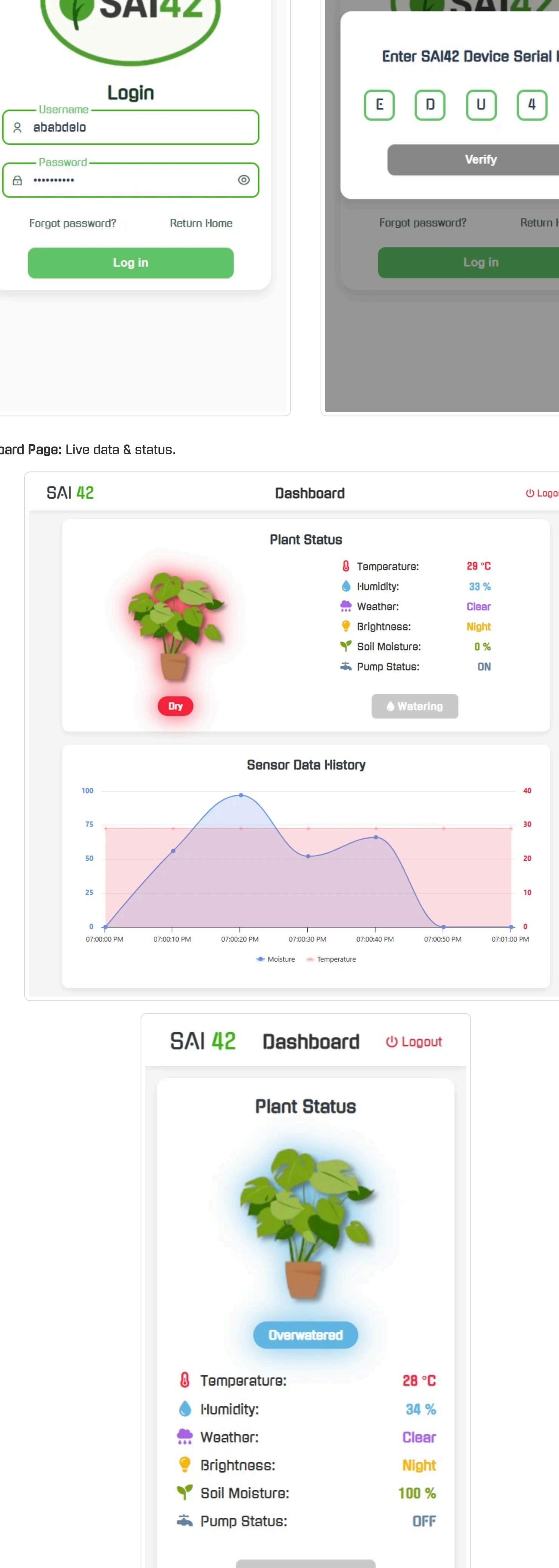
The ESP32 also hosts a secure web dashboard with live charts, historical trends, manual overrides, and alert notifications.

Key Functions

- Real-Time Data Acquisition:**
 - Temperature & Humidity: via DHT22.
 - Brightness: via LDR.
 - Rainfall: via rain sensor.
 - Soil Moisture: via capacitive sensor.
- Intelligent Decision Making:** Threshold algorithms determine when and for how long to irrigate.
- Advanced IoT Integration:** ESP32 serves the web UI, Wi-Fi connectivity, and data logging.
- User Interaction & Security:**

The Intuitive web platform is divided into three sections:

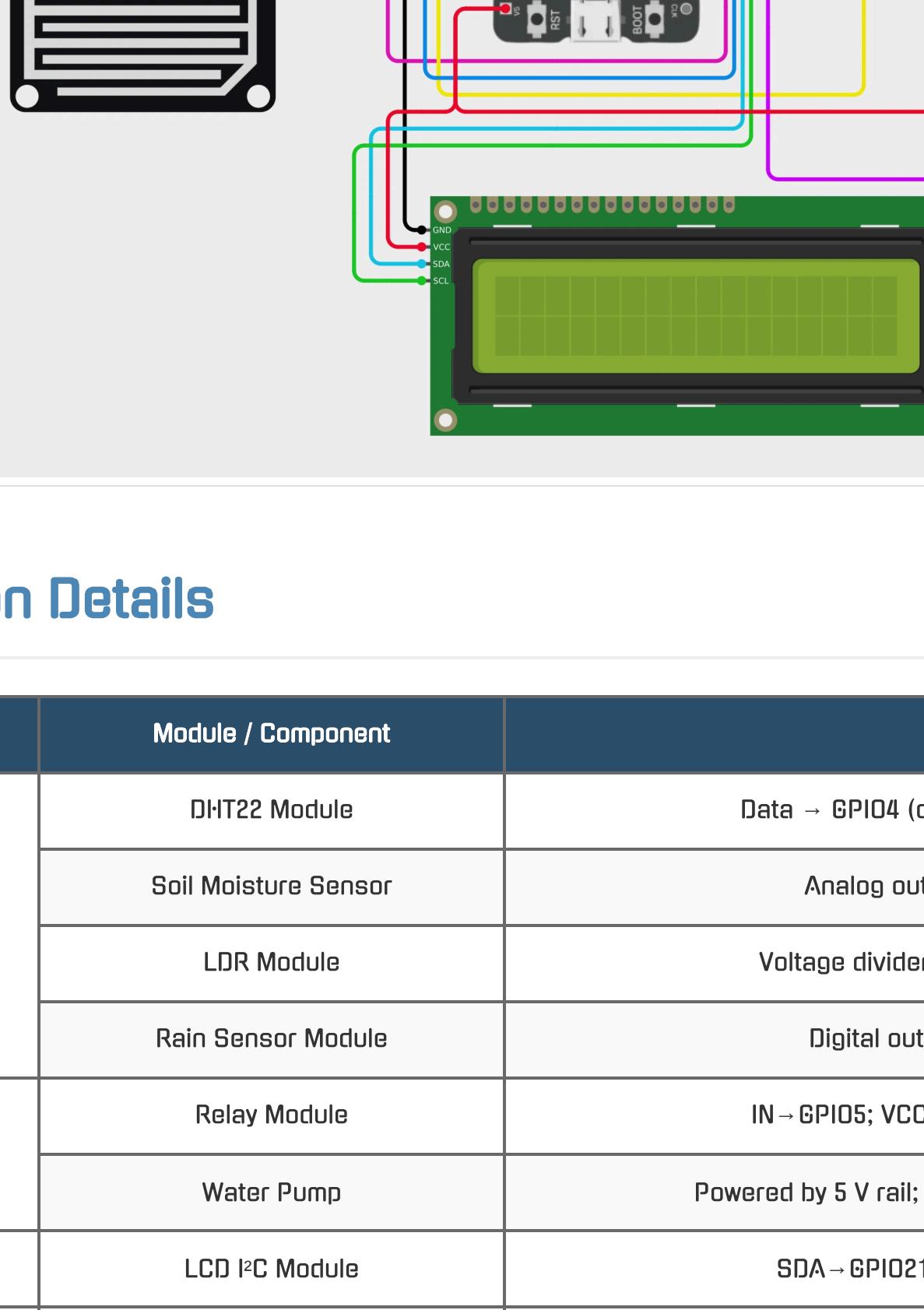
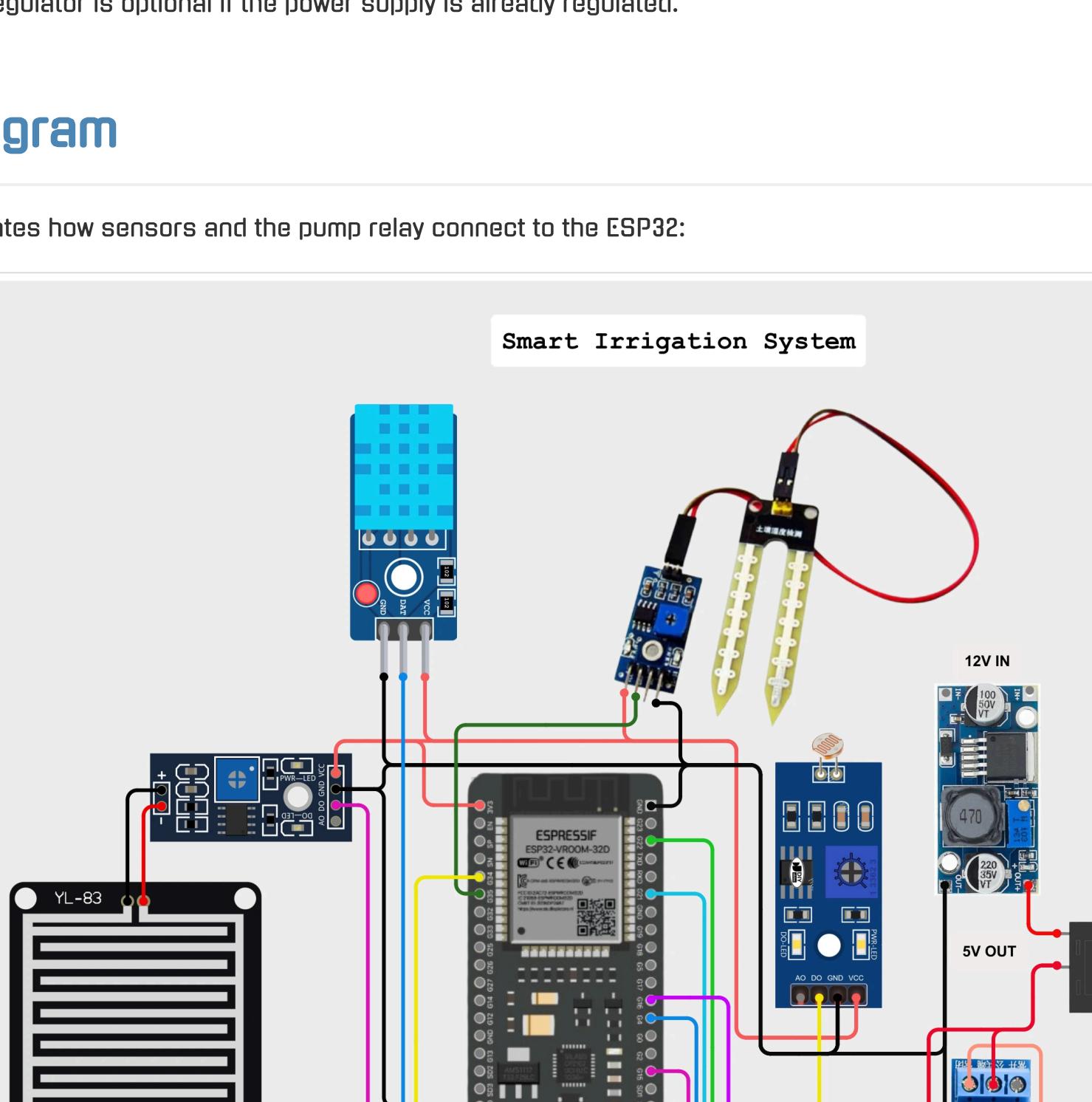
- Home Page:** Introduces core features.



- Login Page:** Secure authentication and credential recovery.



- Dashboard Page:** Live data & status.



Materials Required

Component	Quantity	Description
ESP32	1	Controller & web server.
DHT22	1	Temp & humidity sensor.
Soil Moisture Sensor	1	Capacitive probe.
LDR Module	1	Light detection.
Rain Detector	1	Rain sensing module.
1-Channel Relay	1	Pump control.
Water Pump	1	Irrigation pump.
LCD I2C Module (Opt.)	1	On-site display.
3-Pin ON/OFF Switch	1	Power Control (ON/OFF)
LM2598 Regulator	1	Power regulation.

Note: The LM2598 regulator is optional if the power supply is already regulated.

Wiring Diagram

This diagram illustrates how sensors and the pump relay connect to the ESP32:

Connection Details

Type	Module / Component	Wiring Notes
Sensor	DHT22 Module	Data → GPIO4 (on-board pull-up; VCC - 3.3 V, GND - GND)
	Soil Moisture Sensor	Analog out → GPIO35; VCC - 3.3 V, GND - GND
	LDR Module	Voltage divider out → GPIO34; VCC - 3.3 V, GND - GND
	Rain Sensor Module	Digital out → GPIO15; VCC - 3.3 V, GND - GND
Actuator	Relay Module	IN - GPIO5; VCC - 5 V, GND - GND (common with ESP32)
	Water Pump	Powered by 5 V rail; NO/COM contacts on relay switch its ground
Display	LCD I2C Module	SDA - GPIO21, SCL - GPIO22; VCC - 5 V, GND - GND
Power Rail	ESP Power	ESP32 VIN - 5 V supply (LM2598 or USB), GND - GND
	3.3 V & GND	All sensor modules share the ESP32's 3.3 V & GND rails (do not feed > 3.3 V into any ESP32 pin)
	5 V Rail	Relay, pump and display use a separate 5 V supply (LM2598 or USB); common GND with ESP32

Upload Requirements

- Install the [Arduino IDE](#) (latest stable).

Install [ESP32 board support](#); for guidance follow this tutorial: [Getting Started with ESP32](#)

Install these libraries:

- [ArduinoJson](#)

- [AsyncTCP](#)

- [ESPAsyncWebServer](#)

- [LiquidCrystal_I2C](#)

- [DHT sensor library](#)

- [LittleFS](#)

Install and configure the [LittleFS](#) plugin; for guidance follow this tutorial: [Arduino IDE 2.0: Install ESP32 LittleFS](#)

After successfully uploading your sketch to the ESP32, press [Ctrl + Shift + P](#), type [Upload LittleFS](#), and run to flash the web platform files.

Developed by edunwant42

Licensed under the terms described in the [LICENSE](#) file.