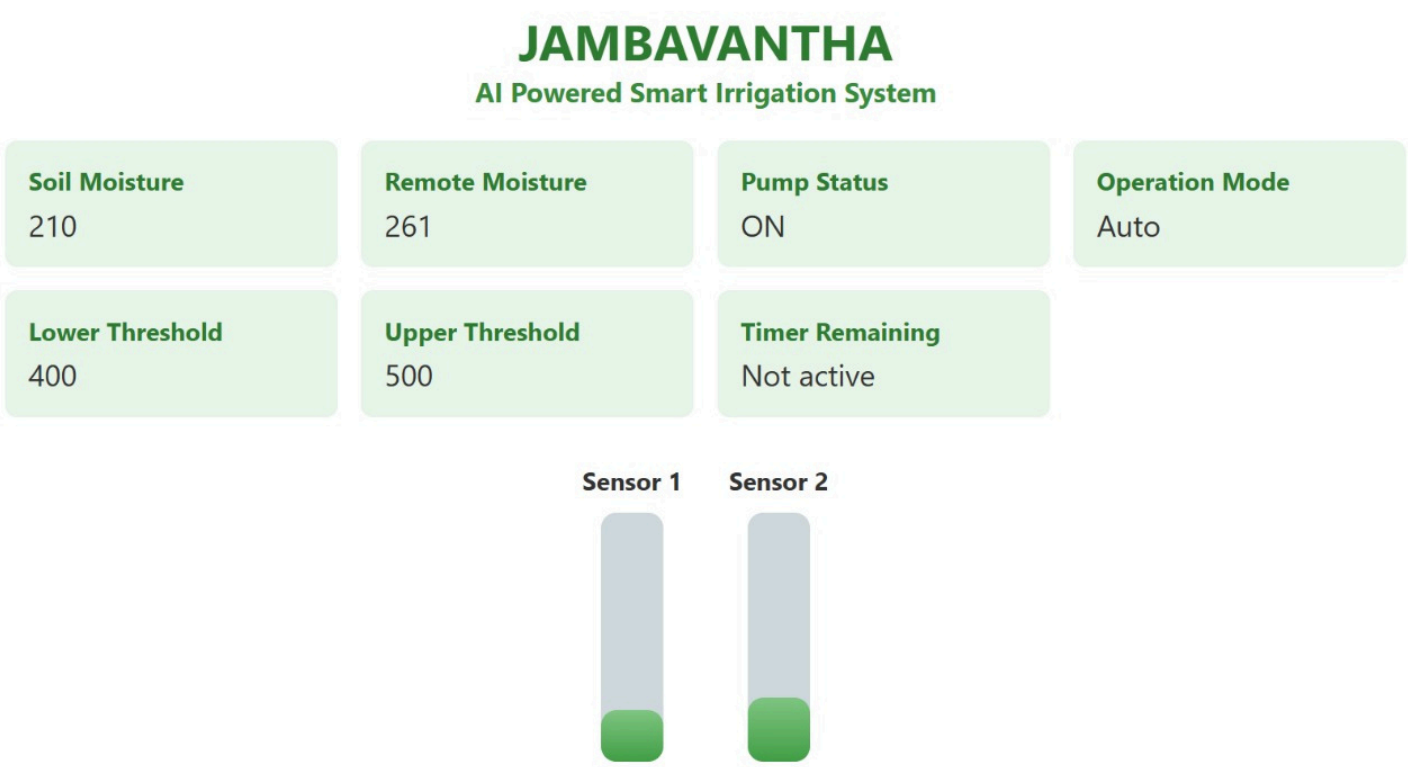
🌱 Smart Irrigation System – AI & IOT Documentation

🟦 Overview

The **Smart Irrigation System** is an IoT-based automation solution designed to optimize water usage for agriculture and gardening. It intelligently controls irrigation based on **soil moisture levels**, **weather conditions**, and **AI-powered recommendations**.

🚀 Key Features

- Soil moisture detection (local and remote)
- AI-based irrigation recommendations using Gemini API
- Weather-aware decision making (OpenWeather API)
- Manual and Auto irrigation modes
- Web-based control and monitoring interface



.

🧰 Hardware Components

- - ESP8266 (NodeMCU): Main microcontroller with WiFi
- - NRF24L01: Wireless data from remote soil sensors
- - Soil Moisture Sensor: Analog sensor for moisture detection
- - Relay Module: Switch to control water pump
- - Power Supply: 5V-12V to power all components

🛠️ Hardware Wiring Guide

Soil Moisture Sensor

- **VCC → 3.3V**
- **GND → GND**
- **AOUT → A0 (ESP8266)**

Relay Module

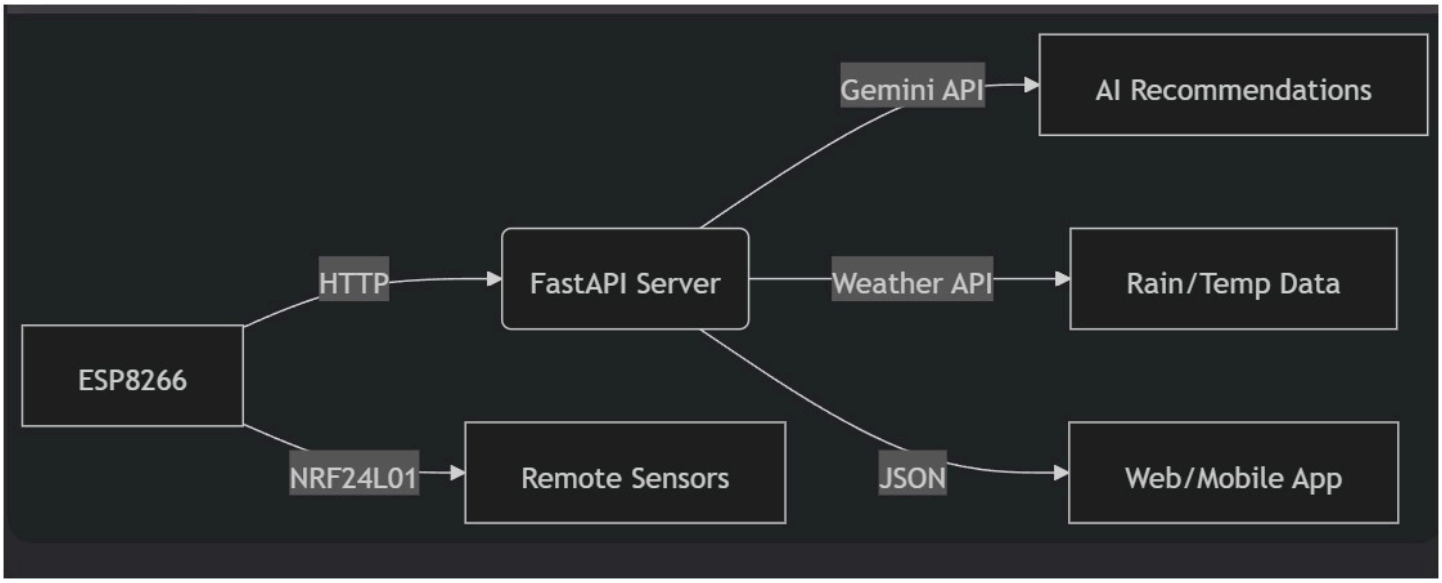
- **IN → D0**
- **VCC → 3.3V or 5V**
- **GND → GND**

NRF24L01 Module

- **CE → D4**

- **CSN → D5**
- **MOSI, MISO, SCK → Connect via SPI pins of ESP8266 (D7, D6, D5 respectively)**
- **Power using capacitor (10µF–100µF) for stability**

🧠 System Architecture



👷 Software Architecture

Core Functional Blocks

1. **Main Loop**
   - Reads moisture sensors
   - Handles WiFi + HTTP requests
   - Makes AI API calls
   - Controls relay based on logic
2. **Web Interface**
   - Accessible via browser
   - View soil data & pump status
   - Manual or Auto mode
   - Timer setting feature
3. **AI Integration**
   - Prompts sent to Gemini
   - Parses and acts on AI recommendation
4. **Weather Integration**
   - Gets current temp + rainfall via OpenWeather API
   - Adds context to AI prompts

🌐 Web Interface Functionality

- Real-Time View: Live soil moisture, pump status, mode

- Mode Control: Toggle Manual or Auto

- Pump Toggle: Manually turn pump ON/OFF

- Timer Feature: Set pump ON for specific minutes

🧠 AI Prompt and Example

Prompt Structure

cpp Copy Edit

```cpp
String prompt = "Give irrigation advice (1 sentence) for: ";
prompt += "Temperature: 28°C, ";
prompt += "Rainfall: 0 mm. ";
prompt += "Soil moisture thresholds: " + String(lowerThreshold) + "-" + String(upperThreshold) + '
prompt += "Format exactly: 'PumpStatus: [on/off] Recommendation: [action]. Reason: [reason]. '";
```

Example Response

json Copy Edit

```json
PumpStatus: on Recommendation: Start irrigation. Reason: Soil is too dry and weather is hot with
```

API Documentation

- - GET - Returns HTML page /status
- - GET - Returns JSON status /control
- - POST - Toggle mode or pump /set_timer
- - POST - Set timer for irrigation).

System Threshold Configuration

int lowerThreshold = 400; // Below this = too dry

int upperThreshold = 500; // Above this = adequately moist

Weather Integration

Uses OpenWeatherMap API to fetch temperature and rainfall. Data is sent to Gemini AI for better irrigation decisions.

const char* OPENWEATHER_API_KEY = "YOUR_KEY"			
String weatherURL = "https://api.openweathermap.org/data/2.5/weather?q=YOUR_CITY&appid=YOUR_KEY"			

Installation Steps

1. Hardware Setup

- Mount ESP8266 on a breadboard or PCB
- Connect sensors and modules as per wiring
- Power via USB or stable 5V supply

2. Software Configuration

- const char* ssid = "YOUR_WIFI_SSID";
- const char* password = "YOUR_WIFI_PASSWORD";
- const char* GEMINI_API_KEY = "YOUR_GEMINI_API";
- const char* OPENWEATHER_API_KEY = "YOUR_OPENWEATHER_API";

3. Deploy and Flash

Use Arduino IDE / PlatformIO

Install libraries:

- ESP8266WiFi
- ESP8266WebServer
- ArduinoJson
- RF24
- WiFiClientSecure

🧪 Usage Instructions

1. Connect to ESP IP address in browser
2. View real-time:
   - Moisture level
   - Pump status
   - Current mode
3. Switch between Auto/Manual
4. Set timers for irrigation

Using Serial Monitor

- Use for debugging
- See API request/response
- Read sensor data
- Trace AI recommendations

💡 Potential Future Upgrades

✅ Add **water flow meter** for usage tracking

✅ Enable **SMS/Telegram alerts**

✅ Integrate **solar-powered battery backup**

✅ Store logs using SPIFFS or SD card

✅ Voice control using Google Assistant / Alexa