



In the name of GOD

Title:

Automate irrigation and online reading data

Supervisor:

DR Mosahebfard

Creator:

Farzin Sharifzade Javidi

Fall 2022

# Contents

Abstract .....	3
Hardware .....	3
Wiring the circuit .....	7
Arduino Code .....	7
Save the Data on local system .....	9
Django .....	10
Online with internet .....	13
Online website.....	17
Conclusion .....	18
References .....	18

# Abstract

In this project we want to automate the irrigation using Arduino and create a web site to show the moisture level of the soil. to do so we need to create control system with Feedback using Arduino micro controller. For the web site We can use Django and python or use any CMS we want. in this project we use both ways And also talk about the API for sending data to the desire webservice whether it is CMS or Django.

## Hardware

In the beginning let's talk about the Hardware of the project. In this project we use Arduino uno How ever the same project can be done with any other kind of micro such as Arduino Nano or Raspberry pi or Node MCU etc. Moisture sensor to sense the humidity of the Soil, 5 Volt relay to connect the Arduino to our pump, And an adaptor for our pump, It's better to use adaptor to power the pump the alternative way is to connect to pump directly to the Arduino but In this scenario You may burn Your Arduino so It's better to get a separated power source.

- Arduino uno
- Moisture sensor
- Single-Channel 5V Relay
- 12V DC pump
- 12V Adaptor
- Jumper Wire

In the following I will explain each component

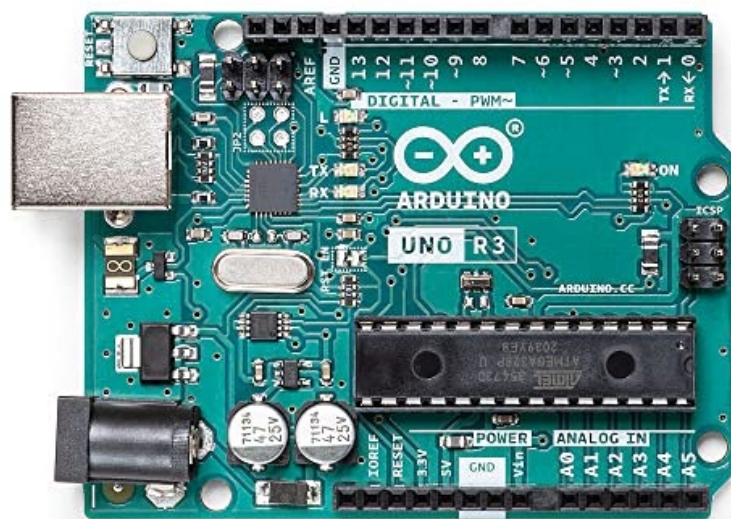
Arduino uno:

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc and initially released in 2010. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable

of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB to serial converter.



## Moisture sensor:

The soil moisture sensor consists of two probes that are used to measure the volumetric content of water. The two probes allow the current to pass through the



soil, which gives the resistance value to measure the moisture value.

## Single-Channel 5V Relay:

Relay is an electromechanical device that uses an electric current to open or close the contacts of a switch. The single-channel relay module is much more than just a plain relay, it comprises of components that make switching and connection easier and act as indicators to show if the module is powered and if the relay is active or not.



12V DC pump:

Any kind of pump can be used. In this project I've used Pride Car Windshield. The important Tip in Choosing Pump is that, to find pump that it would work on the 15V DC at max.



12V DC Adaptor:

As I said earlier It's optional to use adaptor to supply the power for pump, It's better to do so, cause it will prevent over current draw In our Arduino.

In the following I will explain how to bypass adaptor to pump so that the pump will gain it's Currents from adaptor and not Arduino.



Jumper wire:

Since we're working with Arduino We need some jumper wire to connect our modules to Arduino

## Wiring the circuit

In the following I will explain the wiring and schematic of circuit

- 5V from Power Arduino to VCC of sensor
- GND from Power Arduino to GND of sensor
- A3 from Analog pin to AO pin of sensor (AO stands for Analog Output)
- Vin from Power Arduino to VCC of Relay
- GND from Power Arduino to GND of Relay
- Digital pin 7 from Arduino to IN pin of Relay
- NO port of Relay to Positive pole of Pump (NO stands for Normally open)
- Com port of Relay to Positive Wire of Adaptor
- Negative Wire of the pump to Negative of adaptor

Now that wiring is finished Let's talk about Arduino Code

## Arduino Code

---

```
// if the soil is dryer than this number, then start watering
```

```
const int dry = 450;
```

```
const int pumpPin = 7;
```

```
const int soilSensor = A4;
```

```
void setup() {
```

```
  pinMode(pumpPin, OUTPUT);
```

```

pinMode(soilSensor, INPUT);
Serial.begin(9600);
digitalWrite(pumpPin, HIGH);
delay(5000);
}
void loop() {
    // read current moisture
    int moisture = analogRead(soilSensor);
    Serial.println(moisture);
    delay(5000);
    if (moisture >= dry) {
        // the soil is too dry, water!
        Serial.println("Watering starts now..moisture is " + String(moisture));
        digitalWrite(pumpPin, LOW);
        // keep watering for 3 sec
        delay(3000);
        // turn off water
        digitalWrite(pumpPin, HIGH);
        Serial.println("Done watering.");
    } else {
        digitalWrite(pumpPin, HIGH);
        Serial.println("Moisture is adequate. No watering needed " + String(moisture));
    }
}

```



```
}
```

---

Now that We write the code let's explain it a little bit:

At first we Declare our Variable such as the desire Moisture Level and Arduino Pins we want to work with For Example in this case Digital pin 7 and Analog pin 4

Then in Void setup() we must determine output or input of the chosen pin

And finally in the Void loop() section we must write the down code

Since in this project we want to observe the moisture level online it's better to print the moisture level each time we irrigate the soil.

Logic of code is that when the sensor sense the Moisture lower than the Const int Declare in the first, the Arduino will send command to pump and pump will start working for 3 Sec, after that sensor will sense the Moisture again if the Moisture is higher than the declared desired level the irrigation will stop else: it will keep watering till moisture level is equal or higher than the declared variable.

Now if You connect Arduino to the laptop and upload the code into it and put the sensor in the soil and connect the adaptor to the Power outlet you can see that You micro is working properly And also you can see the serial monitor of the Moisture Level.

## Save the Data on local system

As I said Earlier one of the goal in this project is to upload the Data online into the website we create. The easy way of doing this is to use Module Name ESP8226 And connect the module to Arduino and it will upload the data automatically to Arduino cloud and All you need to do is to sign up in the Arduino Cloud : )

In our scenario we want to do it the hard way.

First we will create our own website, As I said earlier we can use the CMS or Django python I've done both so I explain both here:

First Let's talk about Django Webservice

# Django

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

To create our first web page with Django first you must Download Python and install Django. to do that after you install your python you must go in Your command prompt And type in this command:

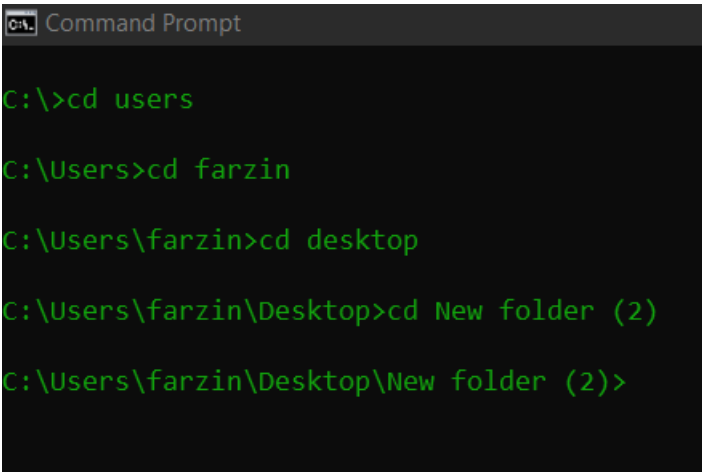
- "Pip install Django"

After you done that You are almost ready to create your first web page. To do so

You must create folder and target that folder in your command prompt using command : "cd"

For example in my case I create a folder in my desktop Name it New folder (2)

And to target that directory I must go to it using cd command As did in below picture:



```
Command Prompt

C:\>cd users

C:\Users>cd farzin

C:\Users\farzin>cd desktop

C:\Users\farzin\Desktop>cd New folder (2)

C:\Users\farzin\Desktop\New folder (2)>
```

When you done that (going to the directory), You must type this code in your command prompt:

- “django-admin startproject mysite”

Congratulation you’ve created your first python web service

Now if you type this command You will run you server on the local machine

- “python manage .py runserver”

After typing this command in Your cmd You must see such feedback giving you the address of your localhost:

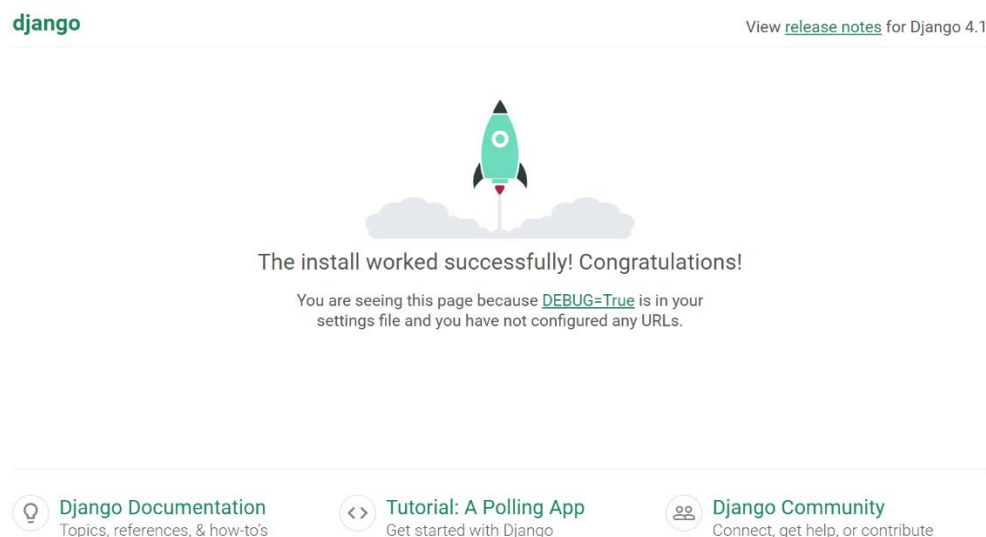
```
C:\Windows\System32\cmd.exe - python manage.py runserver

C:\Users\farzin\Desktop\Farzin\thisone>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
October 14, 2022 - 03:09:50
Django version 4.1.1, using settings 'thisone.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[14/Oct/2022 03:09:58] "GET / HTTP/1.1" 200 10681
[14/Oct/2022 03:09:58] "GET /static/admin/css/fonts.css HTTP/1.1" 304 0
[14/Oct/2022 03:09:58] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 304 0
[14/Oct/2022 03:09:58] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 304 0
[14/Oct/2022 03:09:58] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 304 0
```

And If you go the 127.0.0.1 on your browser You must see your first webpage like this:



Now that we have create our first webpage There is few more step to finish project.

After running server now we need to send our data to webpage, to do that we need to save the data coming from Arduino. There are some ways to do this

One is to use the putty Application or Cool Termwin in this project I use cool-Termwin

By using this app we can save data from our Arduino.

After we did this and save data from our Arduino to our pc we must send data to website.

for example if we want to show the data in the specific page of our website

in this project I wanted to show data on polls page of my website so I Created a Polls page and upload data into it.to that you need to do these step:

first you must create poll app page with the following command in Your command prompt:

- “python manage.py startapp polls”

That will create a directory polls to your website

Now that we have created our desire directory we must upload the data that we saved from our Arduino.to that we must go the folder of our website and then the polls folder. In polls folder find view.py and click on it.

In View.py of polls page we can write down the desire html or anything we want to see on the polls page, to do so let’s create function like this:

Code:

---

```
from django.shortcuts import render
from django.http import HttpResponse
import os
os.chdir(r"C:\Users\farzin\Desktop")
f= open("test.txt","r")
```

```
def index(request):  
    return HttpResponse(f)
```

---

We import OS library here to tell the python to go to the folder that our coolTermwin will save the data as Text file, read that file and append it to our page in our website As you can see in my code first I determine the path that data will be save on and then open the data and read it to memory and finally return the Value as HttpResponse.

Now if we write this address in our browser we must see the data in our webpage:

<http://127.0.0.1:8000/polls/>

we must see such result as answer:



Later on if we want to online this process using internet we must buy the domain and hosting service and when we done that we will have this kind of web address

<http://example.com/polls/> instead of localhost.

## Online with internet

But what if we want a more user friendly and also online with internet website.

In that case we can use the CMS And while this is demo test we can use some free domain and hosting service in my case I use this hosting service:

<https://profreehost.com/>

It got good c-panel no limit size for the site and also provide you a free domain name.

<http://thisistestsokeepitdownonme.liveblog365.com/>

this is the website that I created for this project and because its free domain it got expire date of my free membership so it's expected that the website will be shut down after 2023.

Now that we have a friendly user interface website and not just some simple html code, the main challenge is to post the data from our local machine to the backend of our online webservice

Hopefully In world of open source programming and free internet, there is API for that.

Code:

---

```
import pandas as pd
import base64
import requests
import json
import os

os.chdir(r"C:\Users\farzin\Desktop")

user = 'farzin' #Type the username of your cms
```

```
password ='*****' #tType in your own password here
url ='http://localhost/wordpress/wp-json/wp/v2/pages/'
wp_connection = user + ':' + password
token = base64.b64encode(wp_connection.encode())
headers = {'Authorization': 'Basic ' + token.decode('utf-8')}
token.decode('utf-8')
post_title = "Null:)"
post_body = open("Arduino data.txt","r")
page = {'title': post_title,
        'status': 'publish',
        'content': post_body,
        'format': 'standard'
        }
print("done")
wp_request = requests.post(url + '/post', headers=headers, json=page)
print(wp_request)
```

---

### explanation of API code:

As the explanation to our code: first we need pandas library because we will work with the json data.

We also need request library cause we need to post data to webserver

And base64 library to encode and decode data

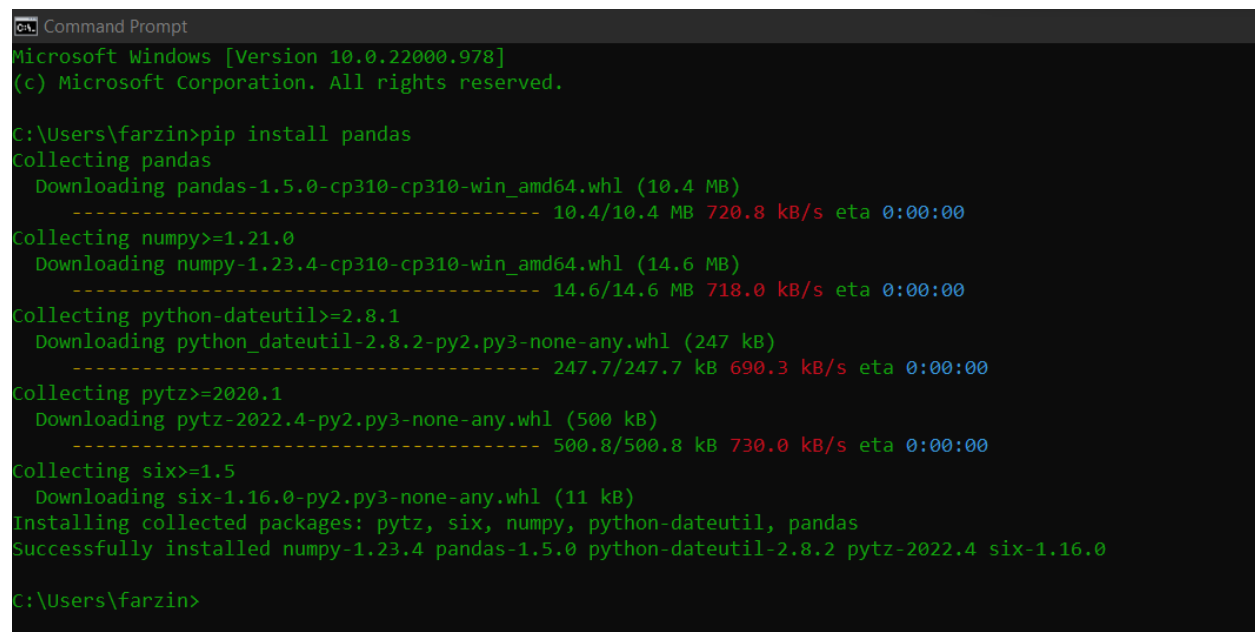
And json library to recognize the json data

Os library to get the data from our local machine in the directory we've save the data

If this is the first that we run python in the machine we probably need to install some of these library first so we have:

- “pip install pandas”

And as result we expect such thing:



```
Command Prompt
Microsoft Windows [Version 10.0.22000.978]
(c) Microsoft Corporation. All rights reserved.

C:\Users\farzin>pip install pandas
Collecting pandas
  Downloading pandas-1.5.0-cp310-cp310-win_amd64.whl (10.4 MB)
----- 10.4/10.4 MB 720.8 kB/s eta 0:00:00
Collecting numpy>=1.21.0
  Downloading numpy-1.23.4-cp310-cp310-win_amd64.whl (14.6 MB)
----- 14.6/14.6 MB 718.0 kB/s eta 0:00:00
Collecting python-dateutil>=2.8.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
----- 247.7/247.7 kB 690.3 kB/s eta 0:00:00
Collecting pytz>=2020.1
  Downloading pytz-2022.4-py2.py3-none-any.whl (500 kB)
----- 500.8/500.8 kB 730.0 kB/s eta 0:00:00
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: pytz, six, numpy, python-dateutil, pandas
Successfully installed numpy-1.23.4 pandas-1.5.0 python-dateutil-2.8.2 pytz-2022.4 six-1.16.0

C:\Users\farzin>
```

In the end we will print the wp\_request Var cause we want to see the feedback from the http

If the http response was 404 it means we post the data to the wrong URL if it was 403 it means it's forbidden this error probably happen when your country ban your hosting server IP or the server IP provider put you under sanction

If the error was 401 it means your web page can't handle json data

The message we expect to show in our terminal as feedback is 200

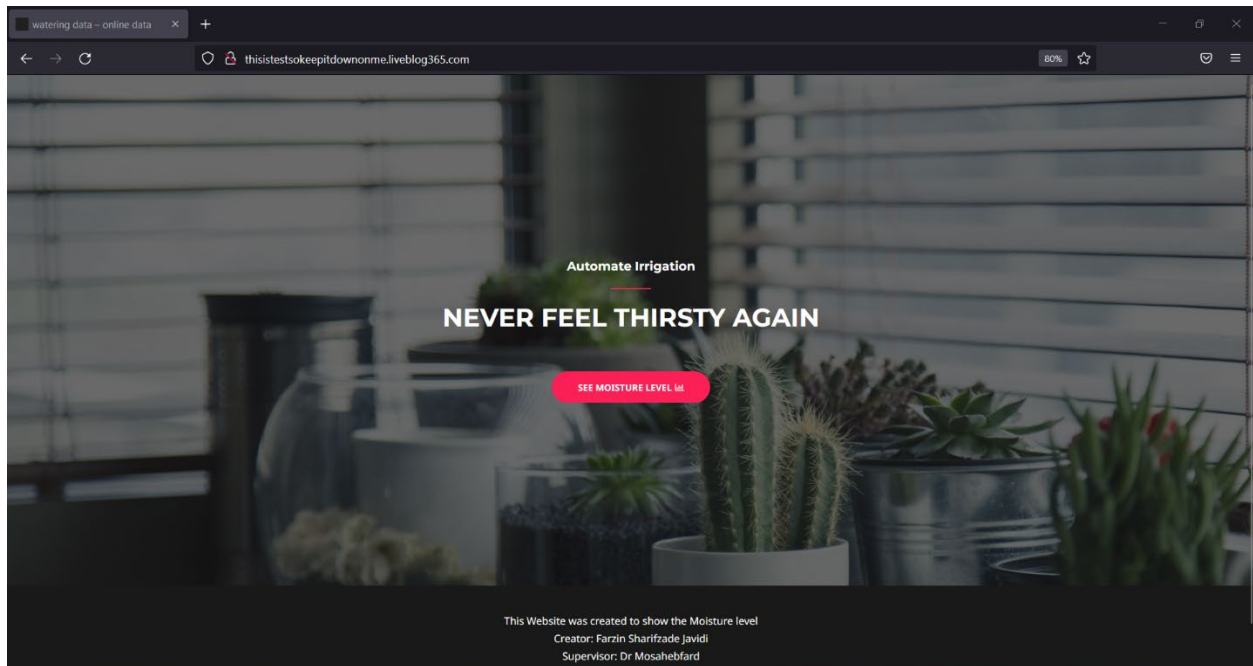
Http response 200 ok !

After we manage to the backend API and post the data on our web site we must see the data on the link we send data to it.

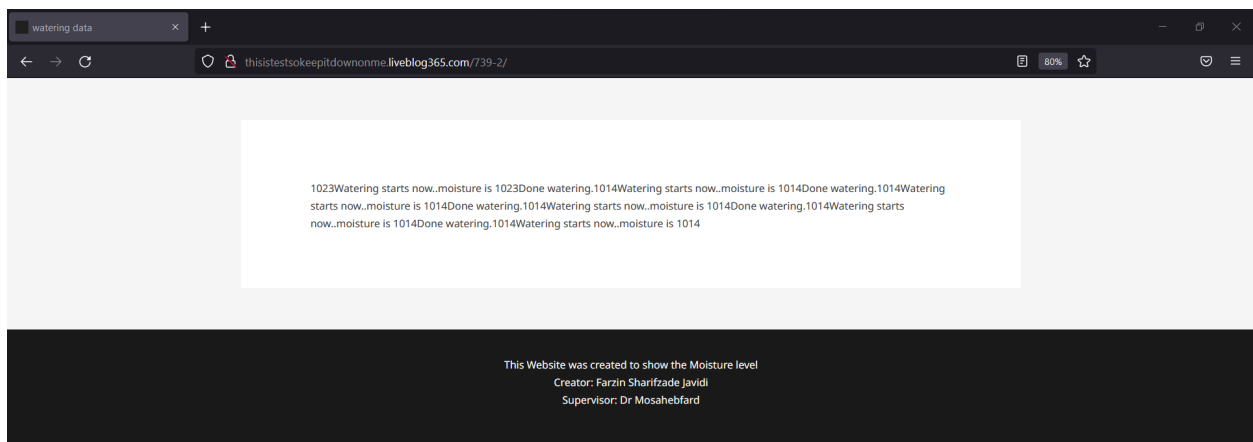


# Online website

This is the front of my web site and if You click on the Button you will headed to the page I created specially for representing data



This is how second page look like:



To see the html and css and javascript code simply click on inspect in your browser to see the code

# Conclusion

We create a close loop control system with feedback And create a website to observe the Data from our feedback. The main challenge in this project was to connect the local data on our machine to the backend of our website, The solution to this problem was The API.

If need be we there is always a way to draw a diagram, In python we can do it with matplotlib library. But since it was not the goal of this project we just pass it.

# References

- [Arduino Documention](#)
- [Django Documantion](#)
- [HTTP Respnose](#)