Temperature and humidity monitor

*Team:*

*Duc Hien Nguyen (Jerry) – n10327622*

*Khanh Duy Nguyen (Sam) – n9784616*

*Ashley De Xuan Poon – n9629238*

QUTIC

ITD 102 Mini Project

# Statement of Contribution

By completing this form, we agree to the following:

1. We declare that all of the work submitted for this assignment is our own original work except for material that is explicitly referenced and for which we have permission, or which is freely available (and also referenced).
2. We agree that QUT may archive this assignment for an indefinite period of time, and use it in the future for educational purposes including, but not limited to: as an example of previous work; as the basis for assignments, lectures or tutorials; for comparison when scanning for plagiarism, etc.
3. We agree to indemnify QUT and hold it blameless if copyright infringements are found in this work and the copyright owner takes action against QUT that is not covered by the normal terms of Educational Use.

**Student 1**

|  |  |
| --- | --- |
| Name | Duc Hien Nguyen |
| Statement of contribution to the project, what did you do? | Technical: web server, LCD  Report: Review and Discussion of Technology, Design and Implementation, video. |
| Signature |  |

**Student 2**

|  |  |
| --- | --- |
| Name | Khanh Duy Nguyen |
| Statement of contribution to the project, what did you do? | Technical: web database management  Report: Set up LCD , Design and Implementation, Appendix. |
| Signature |  |

**Student 3**

|  |  |
| --- | --- |
| Name | Ashley De Xuan Poon |
| Statement of contribution to the project, what did you do? | Technical: temperature and humidity sensor.  Report: Project objectives, Design and Implementation, References. |
| Signature |  |

# Project Objectives

The availability of accurate relative humidity and temperature readings would play a beneficial role in informing users on the current situation within an immediate area in comparison to the general weather forecast. There is a wide array of objects, devices and living entities that depends on these levels to maintain or preserve their safety as well as functionalities. To better get control of the temperature at some certain places, we pushed the project further by updating the temperature and humidity on our website. With data gathered from website the user could combine with their smart home app to control their home temperature to an appropriate degree for their pets’ safety or for some devices to function accurately. Furthermore, users can keep track of their temperature at home to prevent some unexpected accident from happening such as fire or explosion in the house. Additionally, this could act as a weather station for researchers to study the weather and climate.

# Review and Discussion of Technologies Used

## Web server

Web server is one of the most outstanding features in our project. In particularly, there are two specific ways to build a web server; namely static web server and dynamic web server. As going through some discussion, we finally came to a conclusion to build a dynamic web server using a php-based database with the help of WordPress. As for its flexibility and adaptability, we can easily make a post or thread or page without creating or deleting any html files thanks to the help of MySQL database and WordPress. Web server works by allowing connection from outside through a port then redirect that connection to the host machine which is the raspberry pi, and on that raspberry pi, there will be certain files for storing the database as well as the whole website. Additionally, to get access to the website the user can either enter the IP of the home network or through a domain which required the people who make the website assign it to a domain.

WordPress: An open-source content management system based on PHP & MySQL that uses database at the heart of the system rather than a combination of multiple html files, which eases the process of managing the website. WordPress has an online store consisting of many plugins and add-ons that implement extra features to website without the need for complex coding. Additionally, WordPress is open-source software that has a huge community that supports their evolution and development

The additional technology used for building a web server is to control the UDP/TCP port, as for the router will automatically block any external connection into the private network due to its built-in firewall, the idea is to port forward the website-hosting port for the IP which is used by the raspberry pi in order to permit connection from outside the home network. Only after that, users can get access to the website by either entering the IP or domain address.

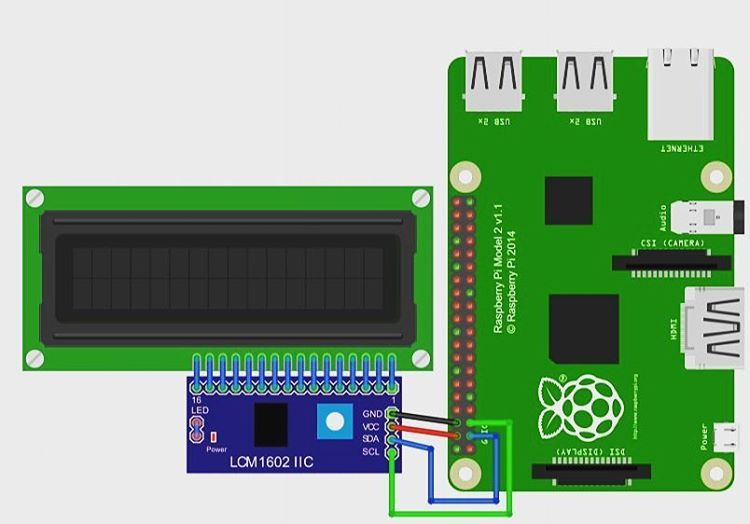
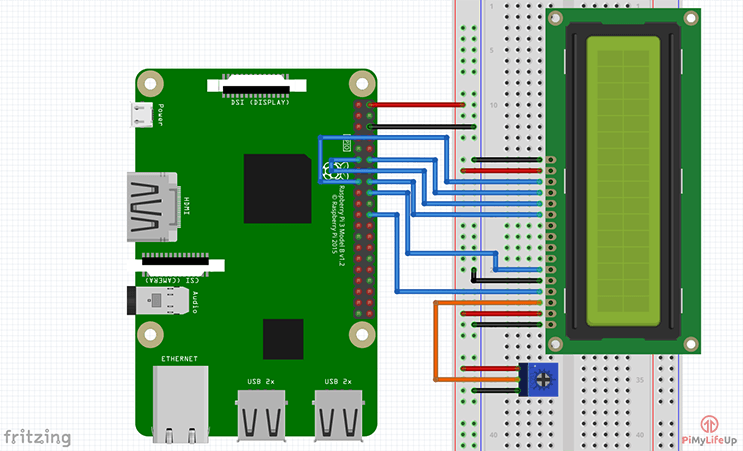
The final technology we used for controlling the webserver is the SQL database, we used MariaDB as well as phpMyAdmin to get control of the whole WordPress database. In case we need to move the website to another host or IP, we could use SQL command to change all the old URL to the new one or we could save all the database and migrate to the new host which show a promising potential of portability.

## Temperature, humidity sensor and LCD screen

Temperature and humidity sensor: the temperature and humidity sensor we used is a DHT11 sensor which consists of a capacitive humidity sensor, a thermistor together with a 10KΩ resistor. The reason we chose DHT11 over the others such as DHT22, AM2302, AM2320, SHT71 is firstly because of its ultra-low cost over the other sensors. Secondly, it provides us with fair accuracy which can read humidity from 20 to 80% with 5% accuracy and 0 - 50°C temperature with 2°C accuracy. And with these given specs, it is powerful enough to fulfil our desirable task. On the market, there are two types of DHT11: the 3-pin type and the 4-pin one; which are basically the same apart from the fact that the 3-pin one has already got a 10K resistor on its circuit which can ease the process of wiring and buying an extra resistor which lead to our decision to choose the 3-pin one over the 4-pin. The DHT11 works by getting the data from the sensors then transfer those data through the GPIO port of the raspberry pi, the data consist of 5 bytes:

* Byte 1: the integral part of the humidity (RH%)
* Byte 2: the fractional part of the humidity (RH%)
* Byte 3: the integral part of the temperature (TC)
* Byte 4: the fractional part of the humidity (TC)
* Byte 5: evaluate the sum

LCD screen: LCD set up could be considered as one of the hardest and most complicated set up in our project as given many choices, alternatives on the market as well as complicated circuit. There are many types of LCD display on the market: 16x2, 32x48 TFT, touchscreen… But for efficiency and economical we picked the 16x2 LCD screen over the others because we only need two lines words: one for showing the temperature and the other for displaying the humidity. For the 16x2 LCD display, there are multiple choices: raw 16x2 LCD without pre-soldered headers, 16x2 LCD display with i2c LCM1602… for our project we came to a conclusion to pick the LCD with the LCM1602 rather than the raw one as for its efficiency and easy to use; since for the raw LCD we need to solder the headers and will take at least 12 pins of the raspberry pi which consist of 7 GPIO pins and could prevent us from adding in more features in the futures as well as sophisticating the circuit. As we could see in the image below for comparing the circuit of a raw 16x2 LCD to the one with the i2c, the LCD with the LCM1602 seems to take less space and more efficient if we need to add in more features:



## Python

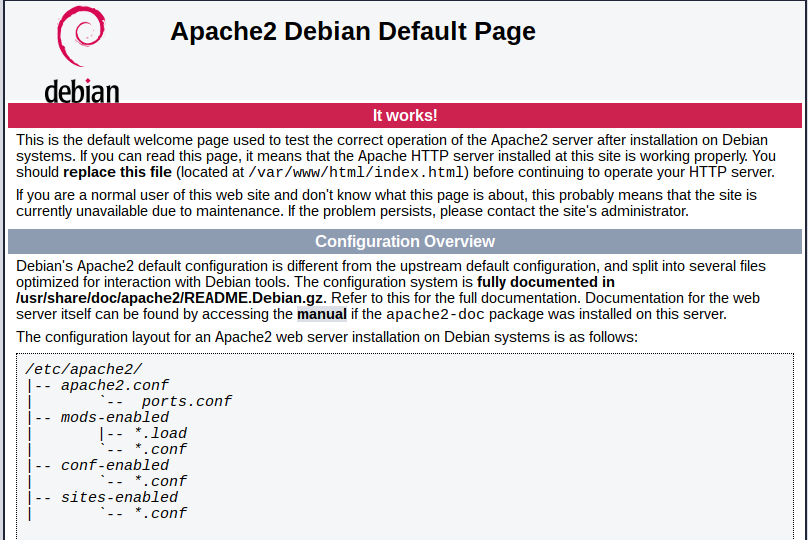
Python is a high-level programming language, which allowed developers to give make programs by writing a sequence of instructions or code. Python works by translating the human readable code into machine code with the help of a compiler then gives these instructions to machine hardware. The reason that we chose Python over the other languages is its ease to use, not too complicated, well-supported by Raspbian and community thanks to its large libraries and APIs. And in our project, we used Python as a tool for everything from checking the temperature and humidity through the sensor to uploading those data to the website and printing out the result to the LCD screen. With given APIs and libraries, we can easily get those hardware connected to by a few lines of code and get everything set up. In particular, we implemented the xmlrpc library to get help with uploading the data onto our WordPress website, Adafruit\_DHT to get the readings from our temperature and humidity sensor,GPIO to get control of the GPIO ports, time and datetime to give a life update on our posts, smbus for controlling the LCD.

# Design and Implementation

## Building a web server

Type in the following command: sudo raspi-config. Then select **Interfacing options** from the menu => Select SSH and enable it to allow terminal internet connection to raspberry.

Type in the following command to install apache2: *sudo apt-get install apache2 –y.* You will now be able to open your website on your computer, keep in mind that your computer and raspberry pi has to be in the same network, type in: *ifconfig* to get the ip of the raspberry pi which is the **inet** under the wlan0 line. Type the ip in the browser, this browser window, which shows that your web server is working, should appear:



Once the webserver is up and running, we can install the PHP with the following command: *sudo apt-get install php –y*. Following that, install MySQL using: *sudo apt-get install mysql-server php-mysql –y* and restart the Apache webserver: *sudo service apache2 restart.* The reason for the selection of these packages is due to the architecture that WordPress is built upon.

WordPress installation:

* Change the current directory to */var/www/html/* and delete all the files in the folder with:

*cd /var/www/html/*

*sudo rm\**

* Download WordPress:

*sudo wget http://wordpress.org/latest.tar.gz*

* Extract the compressed WordPress tar files:

*sudo tar xzf latest.tar.gz*

* Move the extracted files to its current directory from the wordpress directory:

*sudo mv wordpress/\* .*

and remove the empty wordpress directory for organisation:

*sudo rm -rf wordpress latest.tar.gz*

* Lastly, change all the file ownership to the Apache user:

*sudo chown -R www-data: .*

Setting up the database for WordPress:

* Run the secure installation of MySQL: *sudo mysql\_secure\_installation*
* Hit *enter* when asked for the current password for root
* Set a root password by typing *Y* followed by *enter* and entering your desired password
* Remove anonymous users
* Disallow root login remotely
* Remove test database and access to it
* Reload privilege tables

Create the database for WordPress:

* Using *sudo mysql -uroot -p* to run **mysql** in the terminal window
* Enter the created root password
* At the **MariaDB [(none)]>** prompt, create the database for the WordPress installation by using this command:

create database wordpress;

If successful, the following line of code should be visible: Query OK, 1 row affected (0.00 sec)

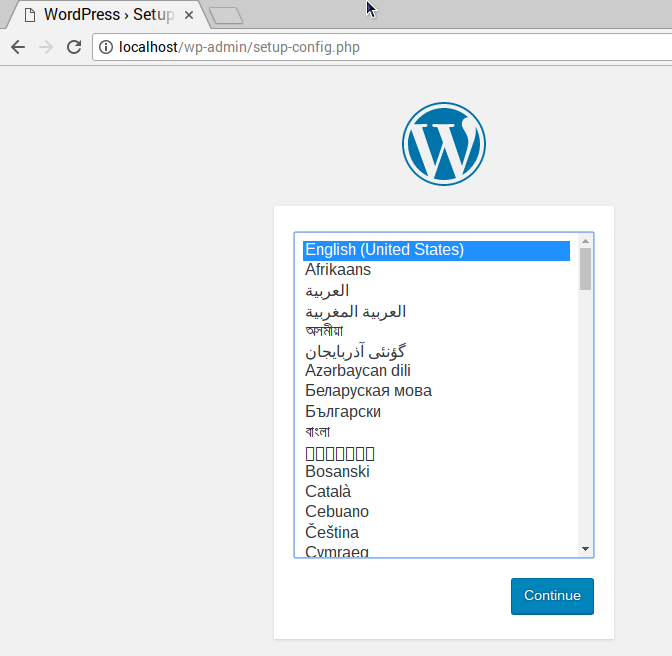
* Grant database privileges to the root user with this command (Type your password after “identified by”):

*GRANT ALL PRIVILEGES ON wordpress.\* TO 'root'@'localhost' IDENTIFIED BY 'YOURPASSWORD';*

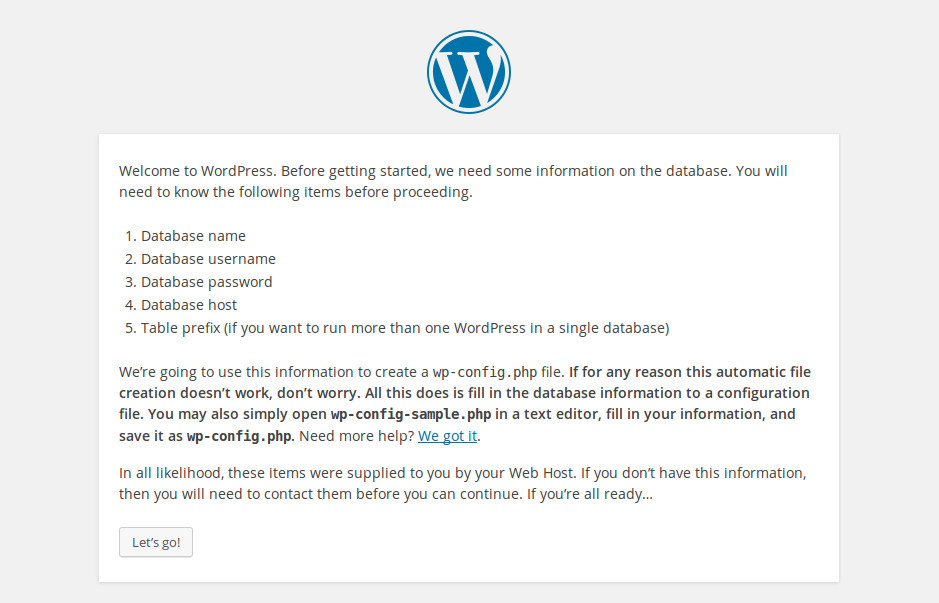
* Flush the database privileges to allow the changes to take effect by: *FLUSH PRIVILEGES;*

WordPress configuration

* Go to <http://localhost> on your browser. This window should show up:



* Select your language and click **Continue,** it then leads to the welcome screen:



* Click **Let’s go!**. Then fill in the site information:

*Database Name: wordpress*

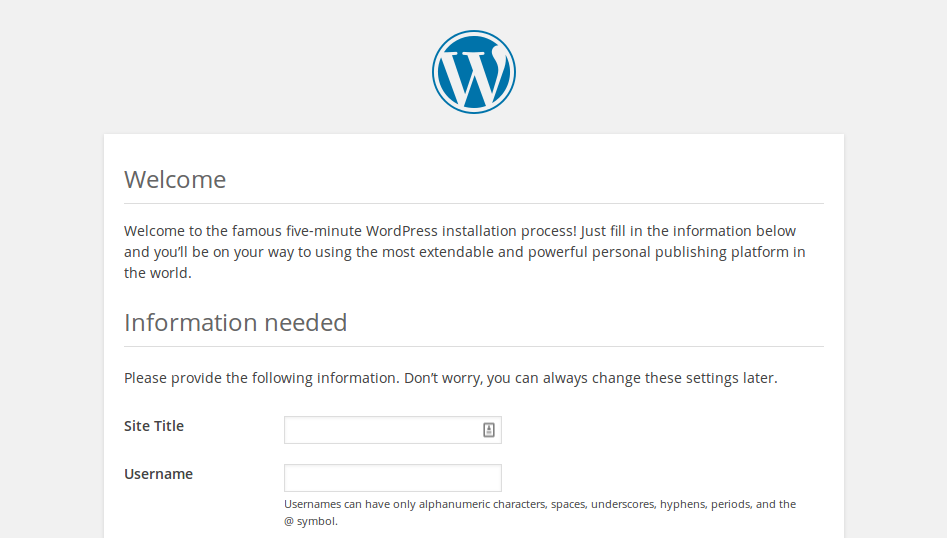
*User Name: root*

*Password: <YOUR PASSWORD>*

*Database Host: localhost*

*Table Prefix: wp\_*

* Click **Submit**  => Click **Run the install** button:



* Fill in the information => Press **Install WordPress** => log in using your account

Change the permalink settings to make the URLS friendlier:

* Log in to WordPress => go to dashboard => go to **Setting => Permalinks** => select  **Post name** option => click  **Save Changes.**
* Enable Apache’s rewrite mod using : *sudo a2enmod rewrite*
* Edit the Apache configuration file for your virtual host: *sudo nano /etc/apache2/sites-available/000-default.conf*
* Add these lines after line 1:

*<Directory "/var/www/html">*

*AllowOverride All*

*</Directory>*

* Save the file and restart Apache: sudo service apache2 restart

## Assign domain to home IP address and put the website on the Internet

One of the most interesting part of our project is to make the website widely accessible through the Internet. To be able to do that firstly we need to port forward the necessary UDP/TCP port on the router for accepting connection from outside the network. The first thing we need to do is a static IP for the raspberry pi then port forward the web hosting port for that IP as if the IP is dynamic, we need to port forward every time it changes so the best way is to set the static IP for it.

**Set static IP**: Open terminal on Putty, type in: *sudo nano /etc/dhcpcd.conf =>* go to the end of the file and add the following line, if you are using the Wi-Fi connecting:

interface wlan0

static ip\_address=<your current ip address>

static routers=<your router ip address>

static domain\_name\_servers=<your router ip address>

and this for ethernet connection (LAN cable):

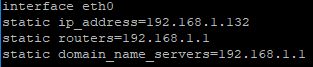
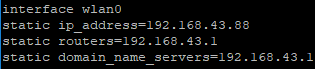
interface eth0

static ip\_address=<your current ip address>

static routers=<your router ip address>

static domain\_name\_servers=<your router ip address>

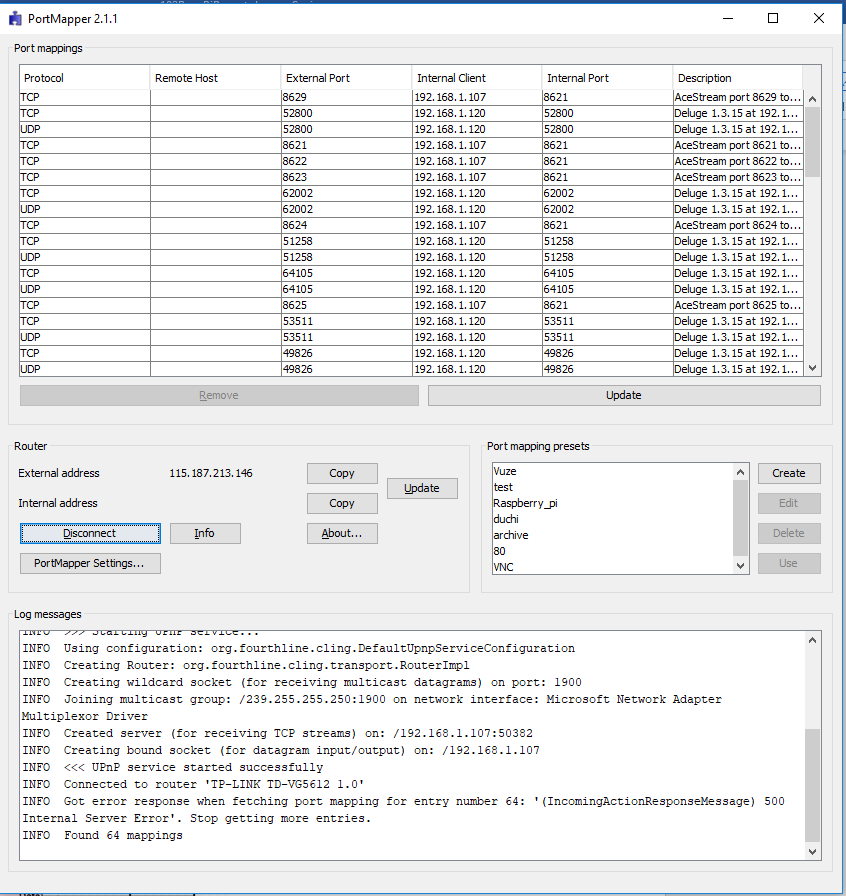
after finishing you should be able to see something like this:



**Port forwarding**: The second step is to port forward the port. Each router has different ways of port forwarding but little know that there is a software that can add port forward without entering the router, that is UPnP PortMapper:

<https://sourceforge.net/projects/upnp-portmapper/>

Our next step is to port forward the necessary ports for the website and raspberry pi. Open up the port mapper => click **Connect =>** click **Update**:



Now click Create => uncheck the “**use local host**” => type in the IP address of the raspberry pi in the “**Internal Client**” => click **Add** twice, now 2 rows showing the port will appear in the **Port** tab:



Change the upper **Protocol** to UDP => change both **External Port** and **Internal Port** of both lines to 80 (which is the port of the website being hosted). Finally write a description for the port in this case we chose the name “80” for simplicity => click **Save.** Now click into the **80** in the **Port mapping presets** section => click **Use** => click **Update** in the **Port mappings** tab, u will now be able to see your port appear on the screen:

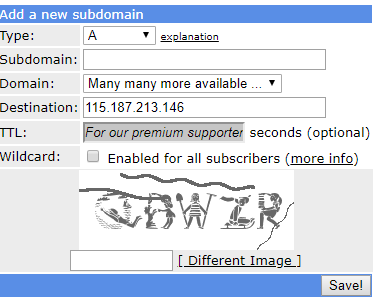


We could also add the port 22 for connecting to raspberry pi terminal on the internet through ssh connection. Now we can type in the IP address of your home network anywhere to get access to our website (if you want to get the IP address of your house just google “my IP address” and the result will appear). But it’s a little bit hard to remember such complicated number, right? That’s why we need to assign an easy to remember domain to our IP.

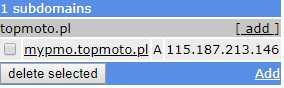
**Domain:** there are many web services that supply domains some are free some are not. This time we are going to use the free service. Firstly, go to this website and sign up an account:

<https://freedns.afraid.org/pricing/>

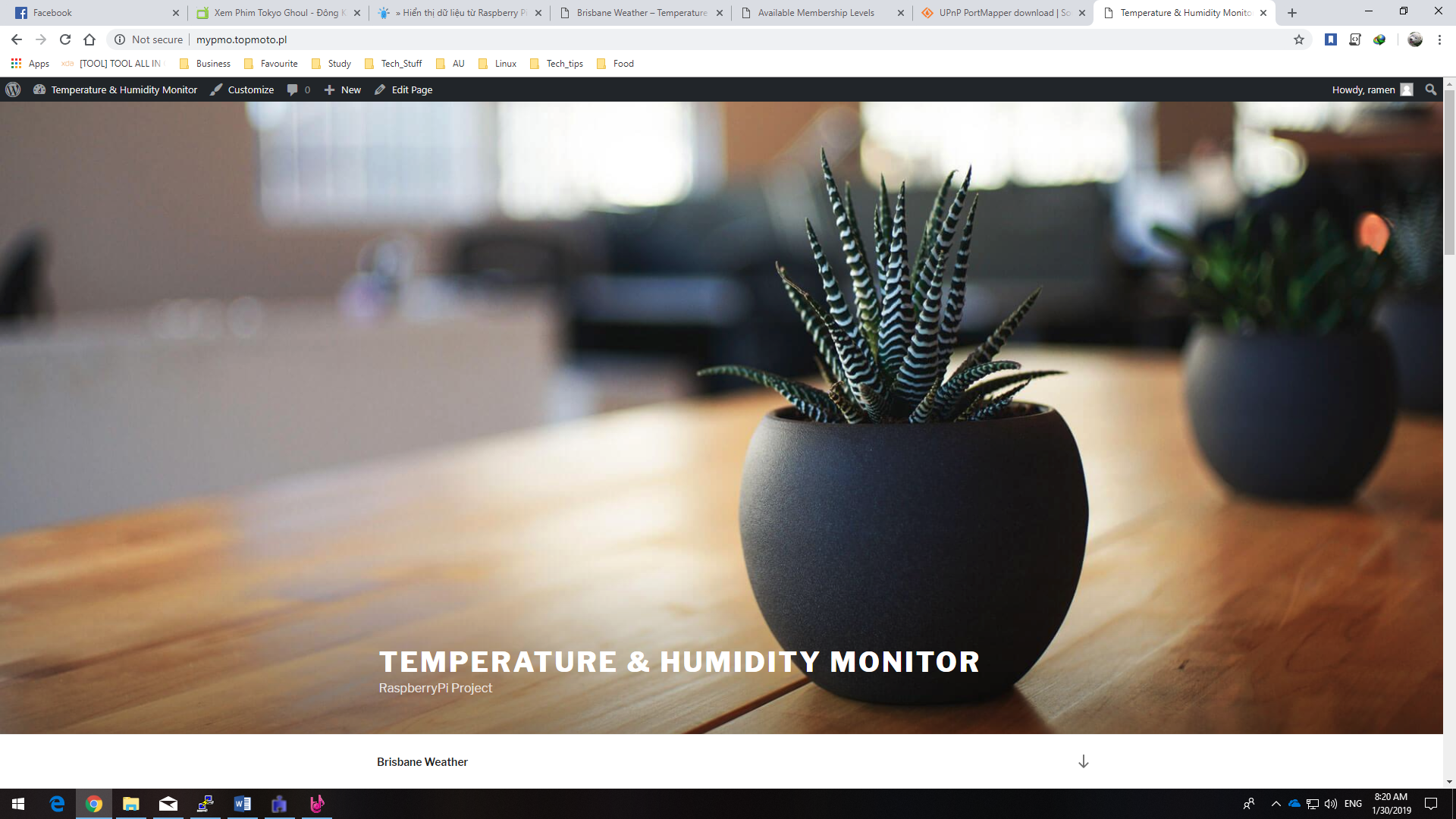
after that click into the **Subdomains** section and sign in => Click into **Subdomains** => Click **Add**. Now fill each section appropriately:



We only need to type in the Subdomain and the image to confirm as everything else as well as our IP has already been set up in the **Destination** section. For **Subdomain** we should use something unique from other websites as if we use the same Subdomain it will coincide with the others and report errors. After that click **Save!** We should be able to see something like this:

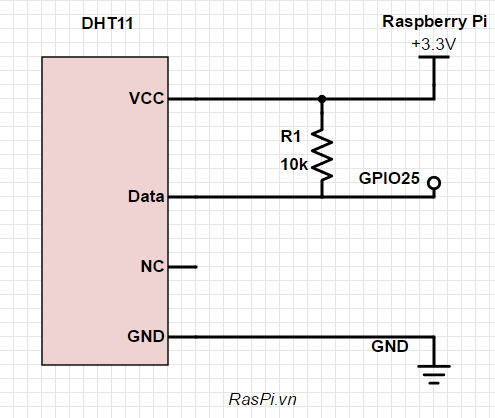


Now type in the domain on the browser we should be able to get access to our website.



## Set up DHT11

Firstly, the DHT11 sensor need to be connected to the raspberry pi according to the following diagram:



We can use any GPIO port, in this case we chose the GPIỌ25. As we chose the DHT11 with a pre-soldered 10KΩ resistor there’s no need for wiring the resister we just need to connect the VCC to the 3.3V port, data to the GPIO25 and GND to the GND port on the raspberry pi.

After setting up the hardware, we need to install libraries for the DHT11 from Adafruit to be able to acquire the temperature and humidity from python, enter these following code line by line:

*cd ~*

*git clone https://github.com/adafruit/Adafruit\_Python\_DHT.git*

*cd Adafruit\_Python\_DHT*

*sudo apt-get update*

*sudo apt-get install build-essential python-dev*

*sudo python setup.py install*

After installing the necessary libraries, we now can be able to get the humidity and temperature data from the sensor through python with this code, this will later on be used in the main python file together with imported LCD module file for our main project:



## Set up LCD 16x2 with i2c LCM1602

Type in *raspi-config* to the Terminal. Then the Setup Options should show up.

Choose **Interfacing Options => I2C => Yes => Ok => Finish => Reboot.**

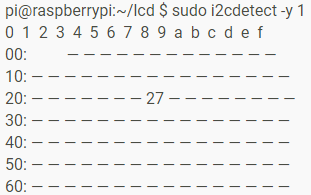
Install smbus and i2c-tools with the following command:

*sudo apt-get update* && *sudo apt-get install –y python-smbus i2c-tools*

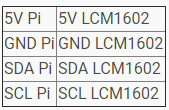
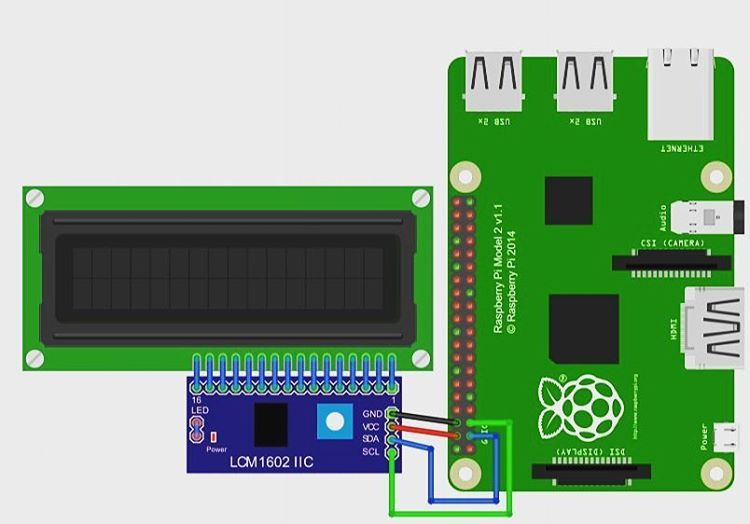
Use the following command to check if the I2C is activated: *lsmod | grep i2c\_*

Use the following command to check for connected devices to the Raspberry Pi: *sudo i2cdetect -y 1*

Then, this result should appear, which shows that one device is connected at the address “27”:



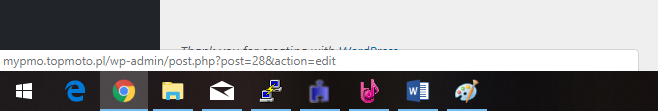
Hardware connection: the circuit should be connected like this:



Code implementation, this code will be used as a module to be imported in the main python to initialise the LCD and display the temperature with the humidity through the lcd\_string(message, line) method. Type in *nano lcd.py*to create the python file, and then copy the code in ([appendix 1](#_Appendix_1)).

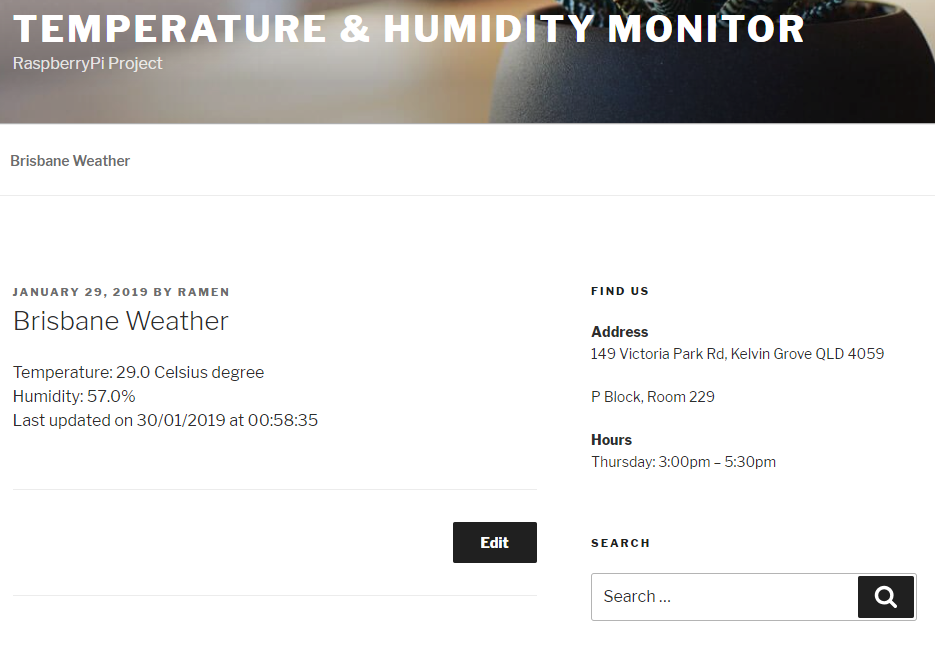
## Combine everything together and give it a go

The last step of our project is to make a python file to get the temperature and humidity to upload to the website as well as displaying on the LCD. Firstly, to be able to do that we need to install the xmlrpc library for python to be able to communicate with our WordPress website, type in the following command*: pip install python-wordpress-xmlrpc*. The next step is to make a post on WordPress and get the id of it, go to the browser => type in “http://Your\_IP\_Address/wp-admin” then sign in with the account we signed up for our WordPress, go to Posts section => click Add New => Name your title => Click publish. Now to be able to get the id of the post we just need to hover over the created Posts it will appear down in the bottom of your browser like this:



So our post id is 28. Now our final step is to make a python file, make sure that we are inside the same directory with the LCD python file that we created before, type this command to make the python file: *nano test.py*, then copy the code ([appendix 2](#_Appendix_2)) into your python file, remember to replace **username** and **password** with your WordPress username and password and replace **Your\_Post\_ID** with your post id.

Type in: *python test.py.* We should be able to see hour Humidity and Temperature update on our website



## Challenges

During our time processing and examining the project, we did come into some problems, some are minor and some are major. Particularly, these are the problems that we faced:

**Change the IP of the WordPress**: as we brought our raspberry pi to our friends’ house when we bring it to another place, the IP changed, and we cannot get access to the website because it was assigned to a different IP. For example, when we set up the WordPress website we chose the **WordPress Address** and **Site Address** to be “mypmo.topmoto.pl” when we changed to another network and we type in the IP address of the raspberry pi (for example: 115.187.213.146) on the browser it would redirect into the address “mypmo.topmoto.pl” and since that domain is not hosted on raspberry anymore we cannot get into the website. So, to be able to solve this problem we need to change all the “mypmo.topmoto.pl” to “115.187.213.146” in the WordPress database. There are two ways to do this; one is to install and use phpMyAdmin, the other is to use MariaDB and enter the codes to change the URL in the WordPress database, we did try both mentioned methods and both worked pretty well, here we will demonstrate how to do it in MariaDB. Firstly, type in: *mysql -u root -p* and type in the password, then type in these following commands, where **oldurl.com** is the old IP address or domain and **newurl.com** is the new address or domain we want to change to:

usewordpress;

UPDATE wp\_options SET option\_value = replace(option\_value, 'oldurl.com', 'newurl.com') WHERE option\_name = 'home' OR option\_name = 'siteurl';

UPDATE wp\_posts SET guid = replace(guid, 'oldurl.com','newurl.com');

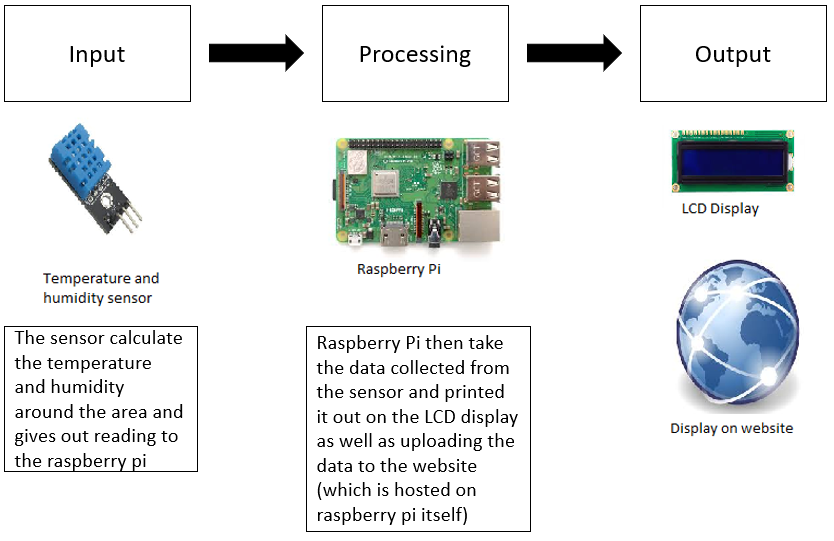
UPDATE wp\_posts SET post\_content = replace(post\_content, 'oldurl.com', 'newurl.com');

UPDATE wp\_postmeta SET meta\_value = replace(meta\_value,'oldurl.com','newurl.com');

**xmlrpc.client.Fault: <Fault -32700: 'parse error. not well formed'>:** this is the second problem that we came across and one of the most annoying one as it took us pretty long time to figure out while executing the python file to upload the temperature and humidity onto our website. The solution is to install the php-xml extension and restart the apache2 server with the following code:

*sudo apt-get install php-xml -y && sudo service apache2 restart*.

## System diagram



## Future directions and improvements

Overall, this project seems to have a promising role in future studies. We could push it further by getting all temperature and humidity data stored in a database then make a graph of average temp and humid changed by time or display trends for climate research purpose. Furthermore, we could implement the email feature to notify users every time the temperature or humidity go beyond a critical point.

# References

Castillo, I., Vidal, C., & Ortiz, N. (2017). FIX: Local IP address changed, can’t log in to WordPress in Ubuntu LAMP. Retrieved from https://isabelcastillo.com/fix-local-ip-address-changed-cant-log-into-wordpress-in-ubuntu-lamp?fbclid=IwAR0GXZRiHjsPQOQA1Yx5qCPubeI-FJre2NkccC-lkNpWcN64N9rneNMAQUk

Domantas, G. (2017). How to Change WordPress URLs in MySQL Database Using phpMyAdmin. Retrieved from https://www.hostinger.com/tutorials/wordpress/how-to-change-wordpress-urls-in-mysql-database-using-phpmyadmin?fbclid=IwAR2StfsDvsBC\_tIHExCUMAqqnRI3QiULXIjfOy6AZdtHGlqah173ml36sL8#gref

Nguyen, T. (2016). Measure temperature and humidity with Raspberry Pi using sensors DHT11. Retrieved from http://raspi.vn/2016/07/03/do-nhiet-do-va-do-am-voi-raspberry-pi-dung-cam-bien-dht11/

Nguyen, T. (2017). Display data from Raspberry Pi to LCD16x2 via LCM1602 relay circuit. Retrieved from http://raspi.vn/2017/01/11/hien-thi-du-lieu-tu-raspberry-pi-len-lcd16x2-qua-mach-chuyen-tiep-lcm1602/

(2017) xmlrpc.client.Fault: <Fault -32700: 'parse error. not well formed'> · Issue #110 · maxcutler/python-wordpress-xmlrpc. (2017). Retrieved from https://github.com/maxcutler/python-wordpress-xmlrpc/issues/110?fbclid=IwAR3GmporwDDupR4jMU7yze3poerDjXzjMgc8liKvvdbklp5r9vG8Nt5CPzU

(n.d.). Build a LAMP Web Server with WordPress. Retrieved from https://projects.raspberrypi.org/en/projects/lamp-web-server-with-wordpress

(n.d.) Using Python to post to WordPress – Yet Another Programming and Electronics Blog. Retrieved from <https://raspiblog.noblogs.org/post/2018/01/27/using-python-to-post-to-wordpress/?fbclid=IwAR2ekbjE5bTk-gDMdg4RXO65uG_dJ2ladpsU8g8-7trYii-K_8lz46a0Gd0>

# Appendix

## Appendix 1

**

## Appendix 2



# Video Link

<https://youtu.be/Pi7T39Kw3Ug>