# Team 1: IOT Sensor – Detailed Instructions & Explanation

If you are having difficulty getting your project up and running, these instructions might help.

* Once you have connected the Pi up to a power supply and a network cable you should be able to use the GUI to get the IP address
* Its probably easier to SSH into the box so you can copy and paste from your laptop your work.
  + Mac: Terminal: ssh pi@<ip address> You’ll then be asked for a password (raspberry)
  + Windows: You’ll need to download a client like PuTTY
* Next you’ll want to connect up the BMP sensor to the pi:
  + Power on Pin 1 (top left with the Ethernet bottom left)
  + SDA on pin 3 (second down on the left)
  + SDC on pin 5 (third down on the left)
  + GND on pin 9 (fifth down on the left)
* Check that its connect to the pi, by running
  + sudo i2cdetect –y 1
  + You should see one of the items in the table is a number. If not, check your wiring
* Install the Adafruit library
  + Sudo apt-get update
  + Sudo apt-get install git build-essential python-dev python-smbus
  + Git clone <https://github.com/adafruit/Adafruit_Python_BMP.git>
  + cd Adafruit\_Python\_BMP
  + sudo python setup.py install
* Try and run the simpletest.py in the examples folder (cd examples, sudo python simpletest.py)
  + If you get a good output, everything is good – you can start to build your own program!
* You now need to start creating your own code
  + You’ll either want to work on the Pi itself, with a command such as nano or vi (a text editor) e.g. nano myfilename.py
  + Or you can work on your laptop and upload to the Pi – you’ll want to grab either WinSCP for Windows or google how to use the scp command
* Take a look at the simpletest.py script and see if you can work out whats going on
* In the “code snippets”, under “pi” directory of the Git repository you have cloned there are a number of scripts.
* Think about what you want your script to do:
  + Import all of the required modules you are going to need for the task
  + Determine whether or not you want to read temperature or pressure
  + Take the reading
  + Create a HTTP connection to a URL with the appropriate readings in

# Team 2: Web Server – Detailed Instructions & Explanation

If you are having difficulty getting your project up and running, these instructions might help.

* Make sure you have Docker for mac / windows installed on your laptop

A quick 101 on how web applications tend to work:

All web applications are delivered by software called a web server. This refers to the software, not the physical hardware.

Common examples of web server software are IIS, Apache and nginx.

In this scenario we suggest we you use nginx for simplicity.

This software sits listening on ports waiting for requests to come in from users.

The webserver then runs scripts based on the requested URL – for example http://site.com/script.php will trigger the script.php script to run.

An important point to note about scripts is that they are generally triggered every time the URL is called – they do not store state. When you are building your code remember that the entire script will be run from the top every time it is called – you’ll need to devise a way to determine which part of the script to run.

Generally we construct web scripts to “pass” these parameters around in either the QUERY STRING or the POST section of the headers – for example you may have seen a URL that looks something like this:  
script.php?action=addReading&value=10

In our script, we could look to see what the action is and the value and act upon it accordingly.

These are some exact steps you’ll want to go through to get this working

* Create a folder somewhere and a file called index.php
  + Inside the line put the following:
  + <?php echo “Test” ?>
* With docker installed, open up your terminal
  + docker run –v <PATH TO YOUR FOLDER>:/app -p 80:80 -d webdevops/php-nginx
* In your browser, type <http://localhost:80/index.php> - you should see “Test”
* You now have a working web server and your first php script!

You now need to extend your index.php script to start getting data from the database.

In the git repository you have cloned there are a number of codesnippets for the webserver project.

Think about what the steps you need to go through are:

* Connect to the database
* Decide whether to:
  + Present the user with either a home page allowing the user to select the time frame and type of data they want
  + Or display the data they have requested
* Home Page:
  + Output a HTML input box which passes via a <FORM> tag the contents back to the script, but this time with an idenfier which specifies you want to display the results
* Results Page:
  + Retrieve the input values
  + Query the database
  + Print out the results from the database

# Team 3: Mobile App – Detailed Instructions & Explanation

If you are having difficulty getting your project up and running, these instructions might help.

Mobile Apps for iOS are creating used the xCode application, freely available in the Mac App Store.

* Once in XCODE, create a new project
* Make sure you have reviewed the basic IOS tutorials
* Drag on a label and a button into the Main.storyboard
* Click the two circles icon (top left) to bring up two windows
* Holding the CTRL key, drag your label and then your button into the ViewController as per the instructions on the apple IOS tutorial
* Within the git repository you cloned, under codesnippets and mobile you will find the XML parser code. Take just the parser functions and paste them in. You will also need to place the appropriate declerations at the top
* Next drag on an action from the button onto the ViewController
* In the codesnippets there is code that will call the parser – you’ll need to modify to get it to work
  + Readings are put into an array called temperatureReadings – you will need to work out how to read this array
  + You’ll then want to update your label with the values
* Next you’ll need to work on how to choose whether to read temperature or pressure