

Hack@CEWIT 2023

Designing the Internet of Things Workshop
Presented by Siri S N

What is IoT?

The term "Internet of Things" (IoT) is used to describe the interconnection and processing of data from embedded sensors and actuators in machines and other physical objects through the use of intelligently connected devices and systems. The Internet of Things (IoT) is predicted to grow rapidly in the coming years, and with it comes a converging set of technologies that will open up a whole new realm of services that boost both individual well-being and business efficiency, heralded as the "Connected Life" opportunity.

About The Workshop

Through this workshop you will build a Weather Monitoring Project, in line with the theme of the hackathon which is sustainability utilizing the concepts of IoT. You will learn how to interface a microcontroller(ESP8266) with a sensor (DHT11) and integrate it with a cloud platform (Ubidots) to store and analyze data and at the same time read the data on your phone by building your own App!

Let's Begin

Project Design Development-User Requirements

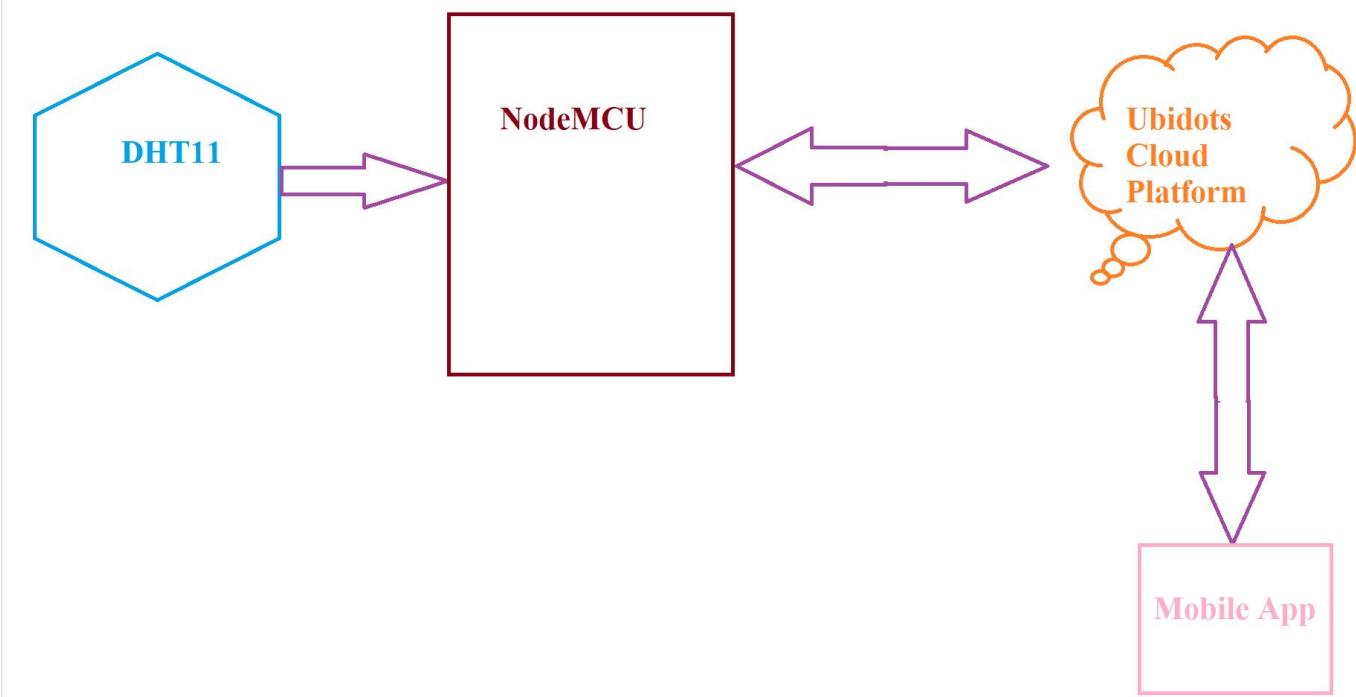
Hardware Requirements:

1. NodeMCU-12E
2. DHT11 sensor
3. Jumper wires
4. Micro-USB Cable

Software Requirements:

1. Arduino IDE
2. Drivers for NodeMCU
3. Libraries-DHT11, Ubidots, Adafruit Unified Sensor
4. Ubidots Cloud Account
5. MIT App Inventor

Flow Diagram



Links to download the IDE, Drivers and Libraries:

Arduino IDE: <https://www.arduino.cc/en/software>(Select and download the appropriate installer based on your system requirements.)

Drivers: https://www.silabs.com/documents/public/software/CP210x_Windows_Drivers.zip

(If you have another OS then navigate to the website of Silicon Labs, and choose your OS, and download the drivers

Link: <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>)

Note: Restart your PC once the drivers are installed.

Libraries:

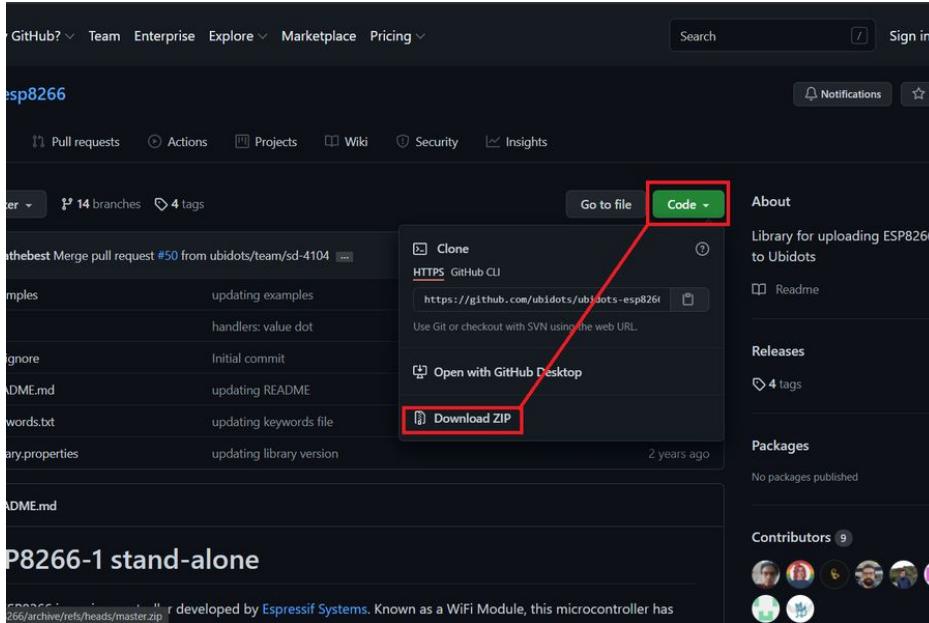
DHT11: <https://github.com/adafruit/DHT-sensor-library>

Adafruit Unified Sensor: https://github.com/adafruit/Adafruit_Sensor

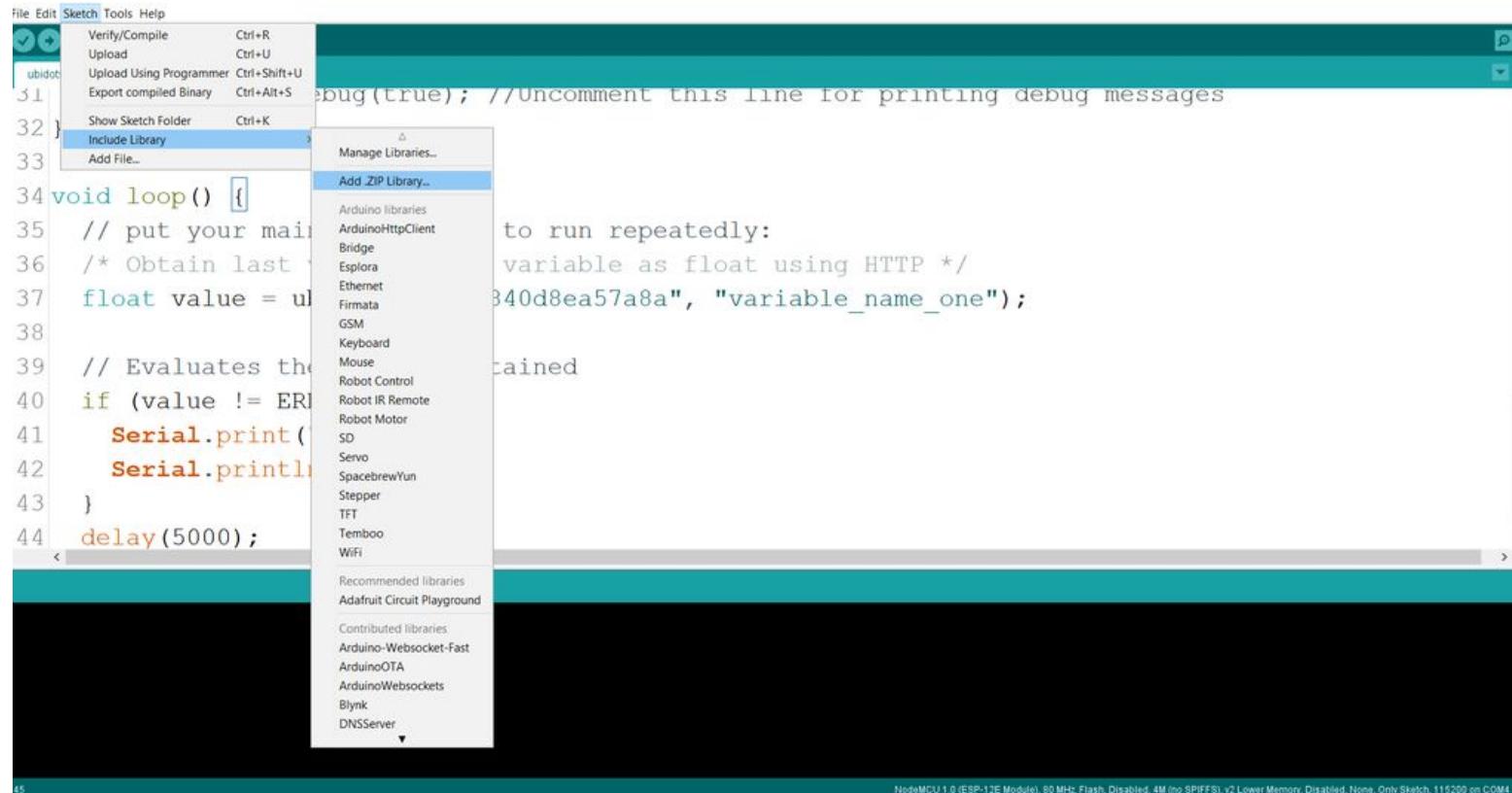
ESP8266-Ubidots Library: <https://github.com/ubidots/ubidots-esp8266>

Adding Libraries to the Arduino IDE

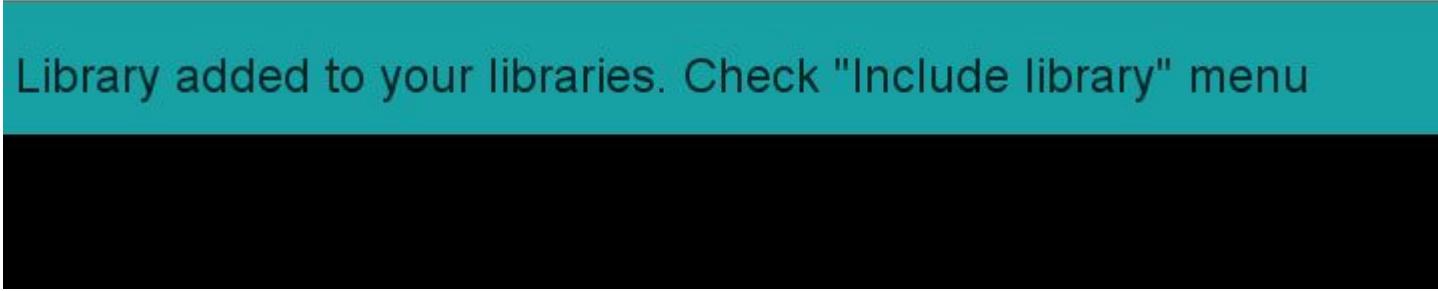
1. Open the links given in the previous slide one by one.
2. Click on code i.e. the green button and Download ZIP.



3. Open the Arduino IDE click **Sketch > Include Library > Add .ZIP Library**.



4. After successful installation, you will receive the following message in the bottom left corner of your Arduino IDE



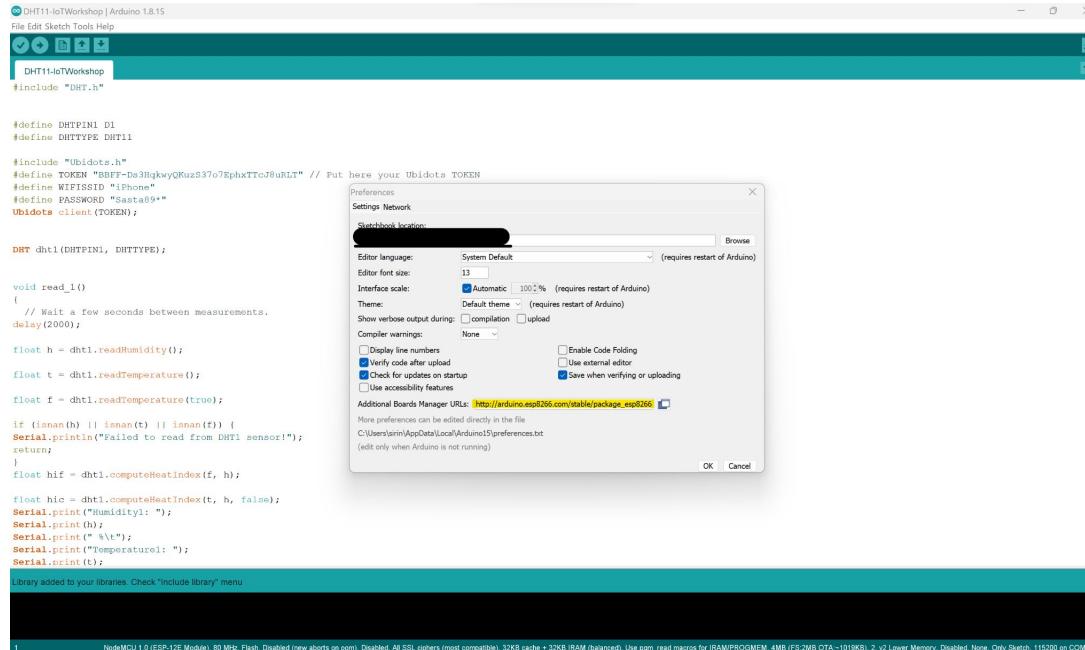
Library added to your libraries. Check "Include library" menu

5. Restart the Arduino IDE once all libraries are installed.

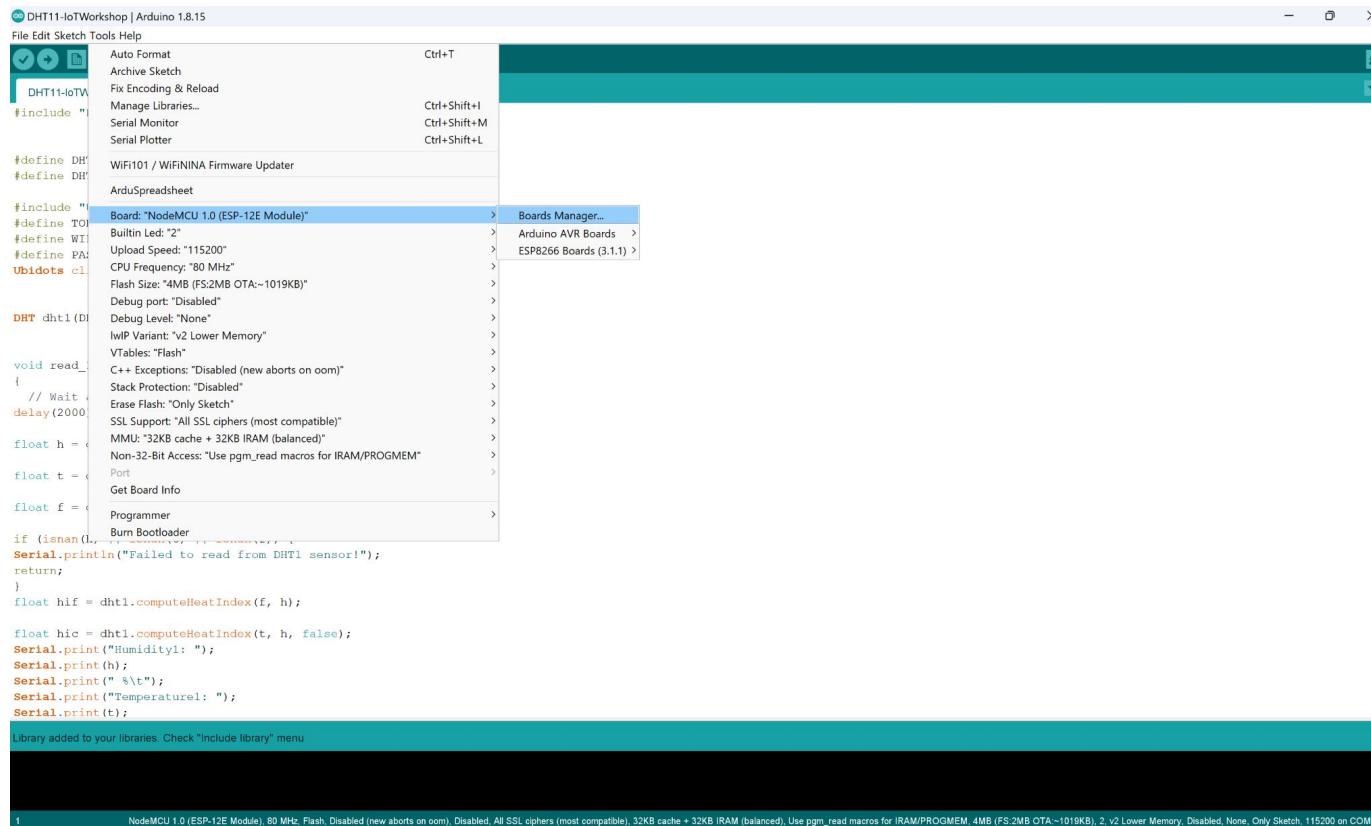
Installation of NodeMCU Board Library

1. Open the Arduino IDE, select **Files > Preferences** and enter the **URL** :

http://arduino.esp8266.com/stable/package_esp8266com_index.json into **Additional Board Manager URLs** field. **IMPORTANT NOTE**: You can add multiple URLs by separating them with commas.



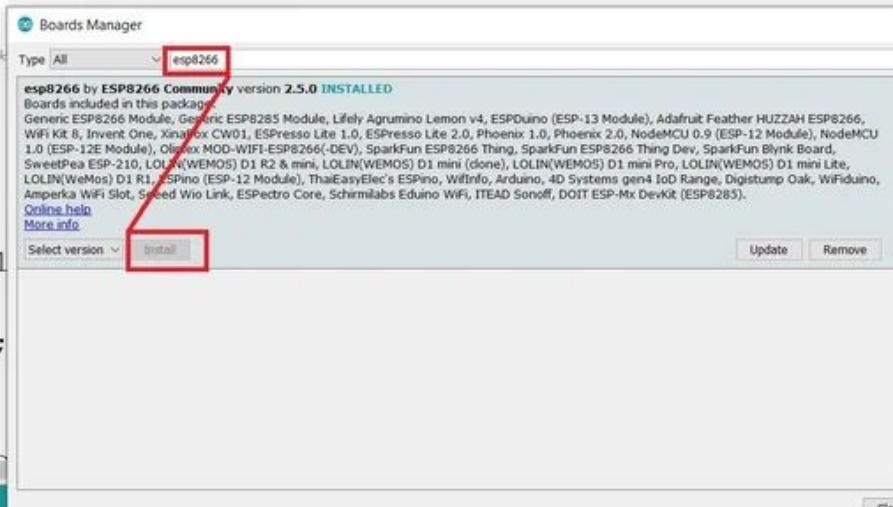
2. Now go to **Tools > Board > Boards Manager** and search for “ESP8266” board. Once you see the ESP listing, select the latest version click on install button.





Posting_data_to_Ubidots_from_NodeMCU

```
1 /*****
2 * Posting Data to Ubidots from NodeMCU
3 * Tested By Aerobotix
4 *****/
5 #include "Ubidots.h"
6 #include <DHT.h>
7
8 #define PIN 4
9 #define DHT_TYPE DHT11
10
11 DHT dht(PIN, DHT_TYPE);
12 float t,h;
13
```



Done uploading.

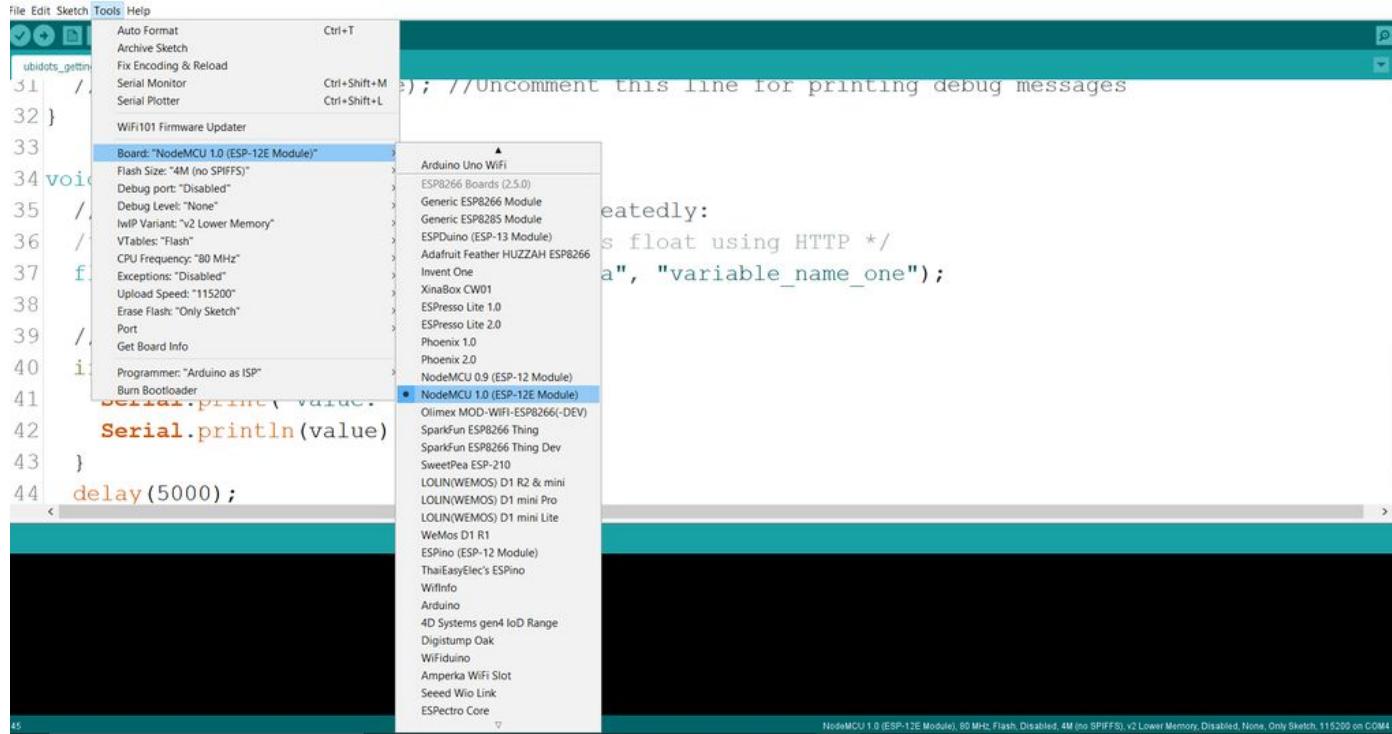
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash: Disabled, 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 921600 on COM4



Type here to search

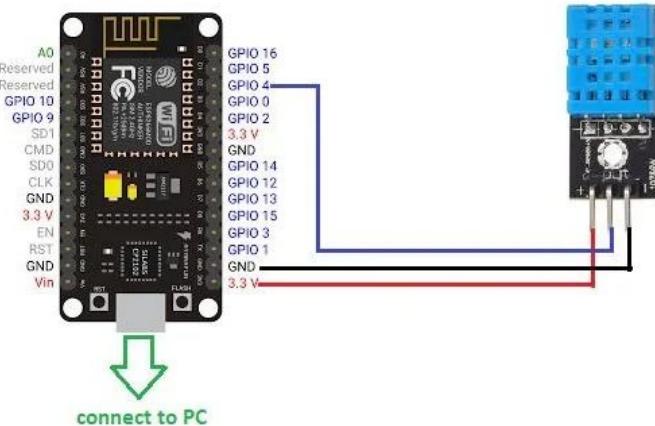
28°C ENG 09:06 PM
13-09-2021

3. Once installation is completed, Now select the board from **Tools > Board > NodeMCU 1.0** (This board may appear in separate ESP category in the boards menu) to continue working.



Hardware Connections

Note: The data “out” pin from the sensor can be connected to any of the 9 Digital pins in NodeMCU but the same should be updated accordingly in the code.



1. Connect as shown in the previous slide
2. Once done, select the the right “COM” port in the Arduino IDE to upload the code
3. If there are many devices connected to your Computer you can check the “Device Manager” to check on which port the ESP8266 is connected.

DHT11-IoTWorkshop | Arduino 1.8.15

File Edit Sketch Tools Help

DHT11-IoTWorkshop

```
#include "DHT.h"

#define DHTPIN 2
#define DHTTYPE DHT11

#include "Ubidots.h"
#include "TOF.h"
#define WI 1
#define PA 0
#define UBIDOTS_CLOUD_ID "YOUR_CLOUD_ID"

DHT dht1(DHTPIN, DHTTYPE);

void read_() {
    // Wait
    delay(2000);

    float h = dht1.readHumidity();
    float t = dht1.readTemperature();
    float f = dht1.computeHeatIndex(h, t);

    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Failed to read from DHT11 sensor!");
        return;
    }

    float hic = dht1.computeHeatIndex(t, h, false);
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print(" %\t");
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.print(" \t");
    Serial.print(hic);
}

Library added to your libraries. Check "Include library" menu
```

Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
ArduSpreadsheet
Board: "NodeMCU 1.0 (ESP-12E Module)">
Builtin Led: "2">
Upload Speed: "115200">
CPU Frequency: "80 MHz">
Flash Size: "4MB (FS:2MB OTA:~1019KB)">
Debug port: "Disabled">
Debug Level: "None">
IwIP Variant: "v2 Lower Memory">
VTables: "Flash">
C++ Exceptions: "Disabled (new aborts on com)">
Stack Protection: "Disabled">
Erase Flash: "Only Sketch">
SSL Support: "All SSL ciphers (most compatible)">
MMU: "32KB cache + 32KB iRAM (balanced)">
Non-32-Bit Access: "Use pgm_read macros for iRAM/PROGMEM">
Port: "COM3">
Get Board Info
Programmer
Burn Bootloader

Serial ports

COM3

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash: Disabled (new aborts on com), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB iRAM (balanced), Use pgm_read macros for iRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

Device Manager

File Action View Help



SidduSheelu

- > Audio inputs and outputs
- > Batteries
- > Bluetooth
- > Cameras
- > Computer
- > DellInstrumentation
- > Disk drives
- > Display adapters
- > Firmware
- > Human Interface Devices
- > Keyboards
- > Mice and other pointing devices
- > Monitors
- > Network adapters
- > Ports (COM & LPT)
 - Silicon Labs CP210x USB to UART Bridge (COM3)**
- > Print queues
- > Processors
- > Security devices
- > Sensors
- > Software components
- > Software devices
- > Sound, video and game controllers
- > Storage controllers

Code Installation

1. Install the code from github(

<https://github.com/siri-CodeforGood/Hack-CEWIT2023-IoTWorkshop>)

and after extraction of the .zip file, open the code file “DHT11-IoTWorkshop.ino”.

2. In `#define WIFISSID "XXXXXX" //Insert Network Name.`

`#define PASSWORD "XXXXXX" //Insert Network Password here.`

Ubidots Account Creation and API Token Retrieval

1. Navigate to the url and create an account

Link: <https://stem.ubidots.com/accounts/signin/>

2. Once done, Under “Profile” click on “API Credentials” and copy the “Default Token”

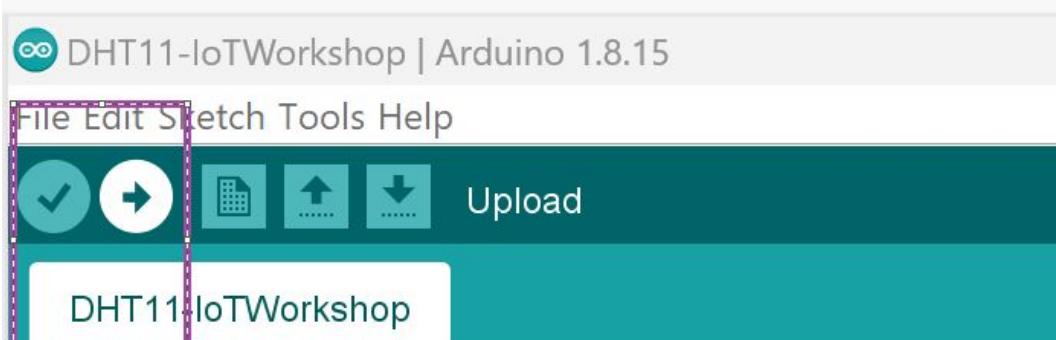
The screenshot shows the Ubidots platform interface. At the top, there is a dark header with the Ubidots logo, navigation links for Devices, Data, Users, and Apps, and user status indicators. Below the header, there are two main sections: "API Key" and "Tokens". The "Tokens" section is highlighted with a purple box and contains a "Default token" field with a "Click to show" button and a copy icon. To the right of the tokens section, there is a user profile area with a "Username: sirish2127", "My Profile" link, and an "API Credentials" link, which is also highlighted with a purple box and has an arrow pointing to it from the profile icon. At the bottom, there is a footer with a "☰ IoT Workshop" button, a date and time selector showing "Mar 02 2023 07:25 - Now", and links for "Docs" and "Log out".

3. Paste the token copied into the code as shown and save it:

```
#define TOKEN "BBFF-Ds3HqkwyQKuzS37o7EphxTTcJ8uRLT" // Put your Ubidots API TOKEN here
```

Uploading Code to NodeMCU

1. First “Verify” the code to ensure no errors are present.
2. Once its verified/compiled with no errors, upload the code to NodeMCU



The screenshot shows the Arduino IDE interface with the title bar "DHT11-IoTWorkshop | Arduino 1.8.15". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for Verify (green checkmark), Run (play button), Open, Save, Upload, and a refresh symbol. The main workspace displays the sketch name "DHT11-IoTWorkshop" and the following code:

```
#include "DHT.h"

#define DHTPIN1 D1
#define DHTTYPE DHT11
```

Done compiling.

```
|- DATA      1600      initialized variables
|- RODATA    2596      constants
[- BSS       25928     zeroed variables
. Instruction RAM (IRAM_ATTR, ICACHE_RAM_ATTR), used 60667 / 65536 bytes (92%)
|| SEGMENT   BYTES   DESCRIPTION
|- ICACHE    32768     reserved space for flash instruction cache
[- IRAM      27899     code in IRAM
. Code in flash (default, ICACHE_FLASH_ATTR), used 377452 / 1048576 bytes (35%)
|| SEGMENT   BYTES   DESCRIPTION
[- IROM      377452    code in flash
```

Done uploading.

```
Writing at 0x00034000... (73 %)
Writing at 0x00038000... (78 %)
Writing at 0x0003c000... (84 %)
Writing at 0x00040000... (89 %)
Writing at 0x00044000... (94 %)
Writing at 0x00048000... (100 %)
Wrote 413696 bytes (303787 compressed) at 0x00000000 in 27.1 seconds (effective 122.3 kbit/s)...
Hash of data verified.
```

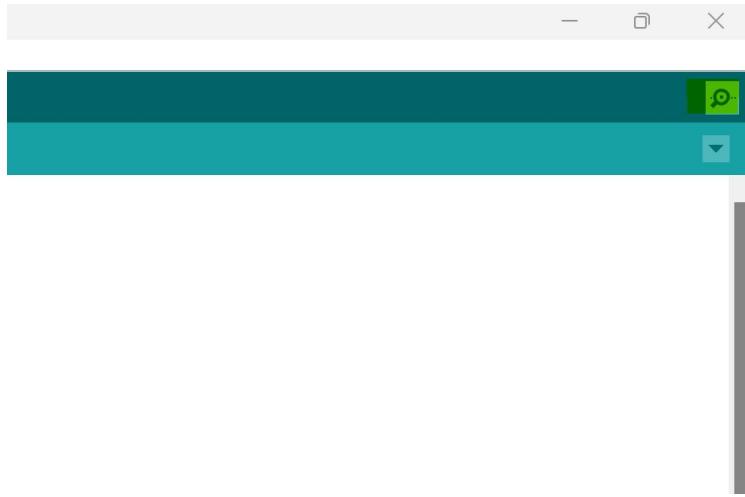
Leaving...

Hard resetting via RTS pin...

Results

1. Open the Serial Monitor
2. Ensure NodeMCU displays the successful connection to the WiFi network message
3. View the Humidity and Temperature readings, and

Notice here, we are utilizing the HTTP method POST to send data to the cloud platform(Ubidots).



```
.....Your network SSID cannot be reached
Reconnecting
Attempt 1
.....Your network SSID cannot be reached
Reconnecting
Attempt 2
.....WiFi connected
IP address:
172.20.10.4
Humidity1: 31.00 %      Temperature1: 24.80 *C 76.64 *F Heat index1: 24.14 *C 75.46 *F
-----
payload:
UbidotsESP8266/3.1.0|POST|BBFF-Ds3HqkwyQKuzS37o7EphxTTCJ8uRLT|2CF4321021F3:2CF4321021F3=>Humidity 1 (%):31,Temperature 1 (*C):24.799999237060547|end
-----

Sending data...
Humidity1: 30.00 %      Temperature1: 24.80 *C 76.64 *F Heat index1: 24.12 *C 75.41 *F
-----
payload:
UbidotsESP8266/3.1.0|POST|BBFF-Ds3HqkwyQKuzS37o7EphxTTCJ8uRLT|2CF4321021F3:2CF4321021F3=>Humidity 1 (%):30,Temperature 1 (*C):24.799999237060547|end
-----

Sending data...
Humidity1: 30.00 %      Temperature1: 24.80 *C 76.64 *F Heat index1: 24.12 *C 75.41 *F
-----
payload:
UbidotsESP8266/3.1.0|POST|BBFF-Ds3HqkwyQKuzS37o7EphxTTCJ8uRLT|2CF4321021F3:2CF4321021F3=>Humidity 1 (%):30,Temperature 1 (*C):24.799999237060547|end
-----

Sending data...
Humidity1: 30.00 %      Temperature1: 24.80 *C 76.64 *F Heat index1: 24.12 *C 75.41 *F
-----
payload:
UbidotsESP8266/3.1.0|POST|BBFF-Ds3HqkwyQKuzS37o7EphxTTCJ8uRLT|2CF4321021F3:2CF4321021F3=>Humidity 1 (%):30,Temperature 1 (*C):24.799999237060547|end
-----

Sending data...
Humidity1: 31.00 %      Temperature1: 24.80 *C 76.64 *F Heat index1: 24.14 *C 75.46 *F
-----
payload:
UbidotsESP8266/3.1.0|POST|BBFF-Ds3HqkwyQKuzS37o7EphxTTCJ8uRLT|2CF4321021F3:2CF4321021F3=>Humidity 1 (%):31,Temperature 1 (*C):24.799999237060547|end
-----
```

Configuring Ubidots Dashboard to Display Sensor Data

1. Open “Devices” in Ubidots and observe that the NodeMCU gets added as a device
2. Click on the device and you will observe 2 variables, temperature and humidity

The screenshot shows the Ubidots Devices dashboard. At the top, there is a navigation bar with the Ubidots logo, a search bar, and tabs for Devices, Data, Users, and Apps. Below the navigation bar is a section titled "Devices" with a search bar and a "Devices" button. The main content area displays a table of devices:

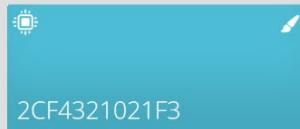
NAME	LAST ACTIVITY	CREATED AT	ACTIONS
2CF4321021F3	5 hours ago	2023-03-03 03:32:07 -05:00	
A020A6126799	3 days ago	2023-02-28 07:49:34 -05:00	
Demo	3 days ago	2023-02-28 07:29:21 -05:00	

At the bottom of the table, there is a "ROWS PER PAGE" dropdown set to 30, and navigation arrows for the first and last pages.

← Devices



SET LOCATION



Description

Change description

API Label

2cf4321021f3

ID

6401b0872e37d9000e7009f4

Token

.....



Tags

Add new tag

Last activity

5 hours ago

Device Type

Set Device Type



31.00

humidity-1

Last activity:
5 hours ago

24.80

temperature-1-c

Last activity:
5 hours ago

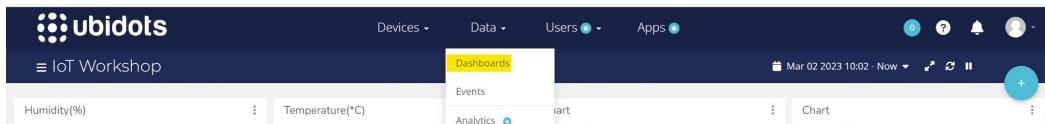
Add Variable

VARIABLES PER PAGE 30

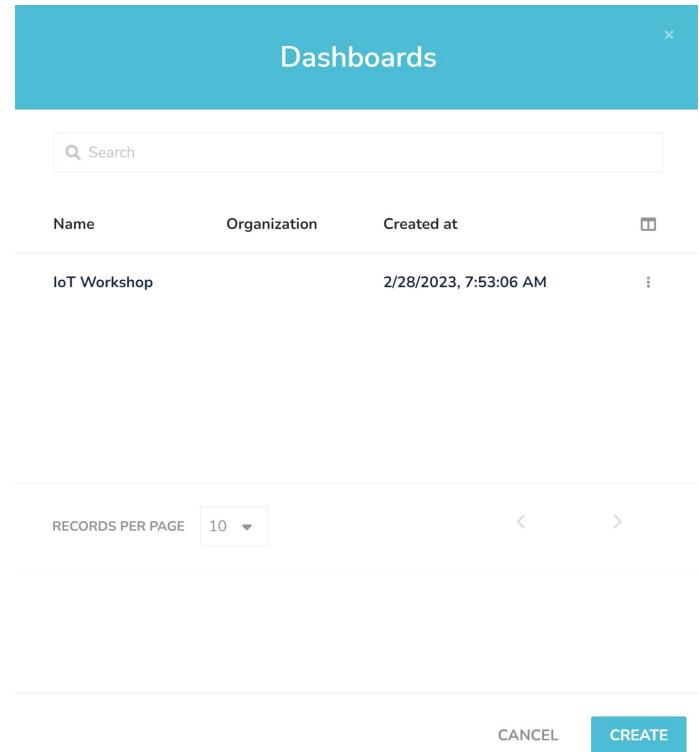


3. Create a New Dashboard

4. Configure Widgets of your choice using your device variables, based on how you want to represent your data on the dashboard



The Ubidots platform interface is shown. At the top, there are navigation tabs: Devices, Data, Users, and Apps. Below the tabs, a search bar contains the text "IoT Workshop". Underneath the search bar, there are two data series: "Humidity(%)" and "Temperature(*C)". In the top right corner of the main dashboard area, there is a prominent blue circular button with a white plus sign (+) on it. The overall background is dark blue.



A modal dialog box titled "Dashboards" is displayed. It contains a search bar at the top with the placeholder "Search". Below the search bar is a table with three columns: "Name", "Organization", and "Created at". The table shows one entry: "IoT Workshop" created on "2/28/2023, 7:53:06 AM". At the bottom of the dialog, there are buttons for "RECORDS PER PAGE" (set to 10), navigation arrows (< and >), and "CANCEL" and "CREATE" buttons.

Name	Organization	Created at
IoT Workshop		2/28/2023, 7:53:06 AM

RECORDS PER PAGE 10 < >

CANCEL CREATE

5. Here I configured “Metric” to display last value and “Line Chart” to display graph of sensor values v/s time

The screenshot shows the IoT Workshop interface with a sidebar on the left and a main content area on the right.

Left Sidebar:

- Top bar with icons for 0, ?, 🔍, and user profile.
- Date range selector: Mar 02 2023 10:02 - Now.
- Search bar.
- Chart tab is selected.
- Bottom bar with three dots and a plus sign icon.

Right Content Area:

IoT Workshop

Add new widget

Search bar

Metrics

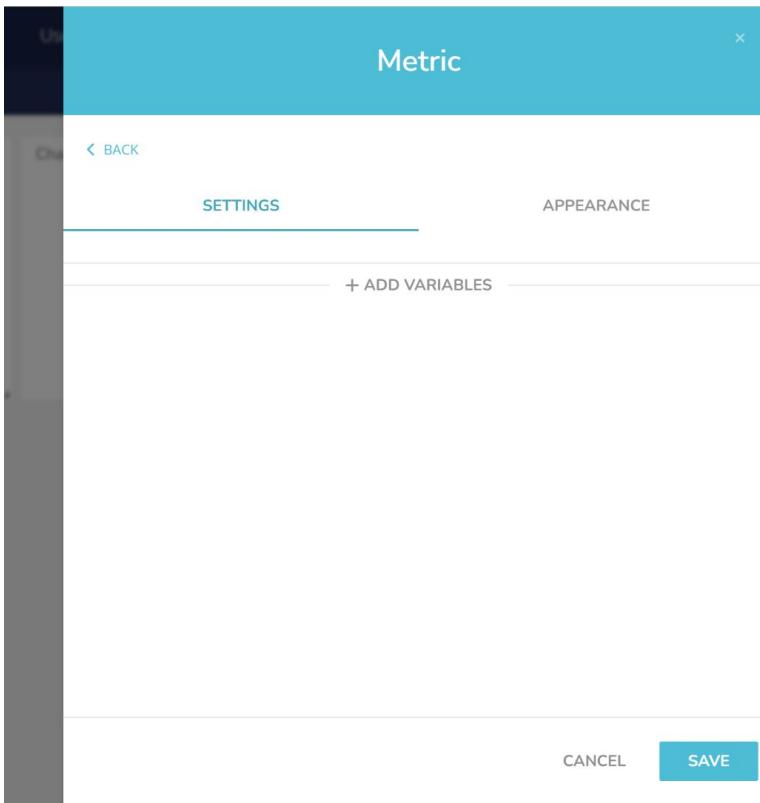
- Metric (selected)
- Thermometer
- Gauge
- Indicator
- Tank
- Battery

Charts

- Line chart (selected)
- Double Axis
- Bar chart
- Rose chart
- Histogram
- Pie chart

Tables

6. Select the device and the variable you want to represent, under each widget and configure accordingly.

A screenshot of a modal dialog box titled "Select Variables". The dialog has a teal header bar with the title "Select Variables" and a close button. Inside the dialog, there is a search bar with the placeholder "Search" and a table with three columns: "Name ↑", "Last activity", and "Created at". The table contains three rows of data:

Name ↑	Last activity	Created at
2CF4321021F3	5 hours ago	2023-03-03 03:32:07
A020A6126799	3 days ago	2023-02-28 07:49:34
Demo	3 days ago	2023-02-28 07:29:21

At the bottom of the dialog, there are navigation arrows, a message "0 variables selected", and two buttons: "CANCEL" and "SELECT".

Metric

X

Select Variables



Search

Name ↑

Last activity

Created at



2CF4321021F3

5 hours ago

2023-03-03 03:32:07



Search



humidity-1

Label: humidity-1



temperature-1-c

Label: temperature-1-c



A020A6126799

3 days ago

2023-02-28 07:49:34



Demo

3 days ago

2023-02-28 07:29:21



1 variables selected

CANCEL

SELECT

Similarly, it can be configured for “Line Chart” or any other widget of your choice

Building a Mobile Application using MIT App Inventor

1. Create an account here : <https://appinventor.mit.edu/>
2. Once created, Start a new project
3. Add any number of screens you want, I have added only 2 according to the requirement: Screen 1 and Dashboard

The screenshot shows the MIT App Inventor web application. At the top, there is a navigation bar with links for 'Projects', 'Connect', 'Build', 'Settings', and 'Help'. On the right side of the top bar, there are links for 'My Projects', 'View Trash', 'Guide', 'Report an Issue', and 'English' (with a dropdown arrow). Below the top bar, there is a green header bar with buttons for 'Start new project', 'Move To Trash', 'View Trash', 'Login to Gallery', and 'Publish to Gallery'. The main area is titled 'Projects' and contains a table with four rows of project information. The first row is highlighted with a pink border. The columns are labeled 'Name', 'Date Created', and 'Date Modified'. The projects listed are: Trial (Feb 28, 2023, 8:01:47 AM; Mar 3, 2023, 10:20:11 AM), AI_Assistant_1 (Jan 9, 2019, 10:44:02 PM; Feb 28, 2023, 8:09:03 AM), and AI_Assistant (Dec 3, 2018, 1:15:15 PM; Feb 28, 2023, 8:08:24 AM). At the bottom of the page, a modal dialog box is displayed with the title 'Create new App Inventor project'. It has a text input field labeled 'Project name:' and two buttons at the bottom: 'Cancel' and 'OK'. The entire 'Start new project' button in the header and the 'Create new App Inventor project' dialog are highlighted with a pink rectangular border.

Name	Date Created	Date Modified
Trial	Feb 28, 2023, 8:01:47 AM	Mar 3, 2023, 10:20:11 AM
AI_Assistant_1	Jan 9, 2019, 10:44:02 PM	Feb 28, 2023, 8:09:03 AM
AI_Assistant	Dec 3, 2018, 1:15:15 PM	Feb 28, 2023, 8:08:24 AM

3. Use the comprehensive logical blocks in different screens to design the application.

Screenshot of a mobile application builder interface titled "Trial".

The interface includes the following components:

- Top Bar:** Includes tabs for "Designer" and "Blocks", and buttons for "Screen1", "Add Screen ...", "Remove Screen", and "Publish to Gallery".
- Left Sidebar (Palette):** Groups components under "User Interface" and "Layout". "User Interface" includes: Button, CheckBox, DatePicker, Image, Label, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, and WebViewer. "Layout" includes: Media and Drawing and Animation.
- Center View (Viewer):** Shows a smartphone screen with the title "Screen1" and a single button labeled "Dashboard". It also displays a "Display hidden components in Viewer" checkbox and a "Phone size (505,320)" dropdown.
- Right Sidebar (Properties):** Shows properties for "Screen1" and its components "Button1".
 - Screen1 Properties:** Includes: AboutScreen (Screen1), AccentColor (Default), AlignHorizontal (Center : 3), AlignVertical (Top : 1), AppName (Trial), BackgroundColor (Default), BackgroundImage (None...), BigDefaultText (unchecked), BlocksToolkit (All), CloseScreenAnimation (Default), DefaultFileScope (App), HighContrast (unchecked), and Icon (None...).
 - Button1 Properties:** Includes: Media (Upload File ...).

Blocks

Built-in

- Control
- Logic
- Math
- Text
- Lists
- Dictionaries
- Colors
- Variables
- Procedures

Screen1

- Button1

Any component

Rename Delete

Media

Upload File ...

Viewer



when **Button1** .Click
do **open another screen** screenName **Dashboard**

**Show Warnings**

4. Add another screen named as “Dashboard” or anything to make it easier to distinguish from the first screen and use Web Viewer to display the Ubidots Dashboard API.
5. You can find the Ubidots Dashboard API as shown below:

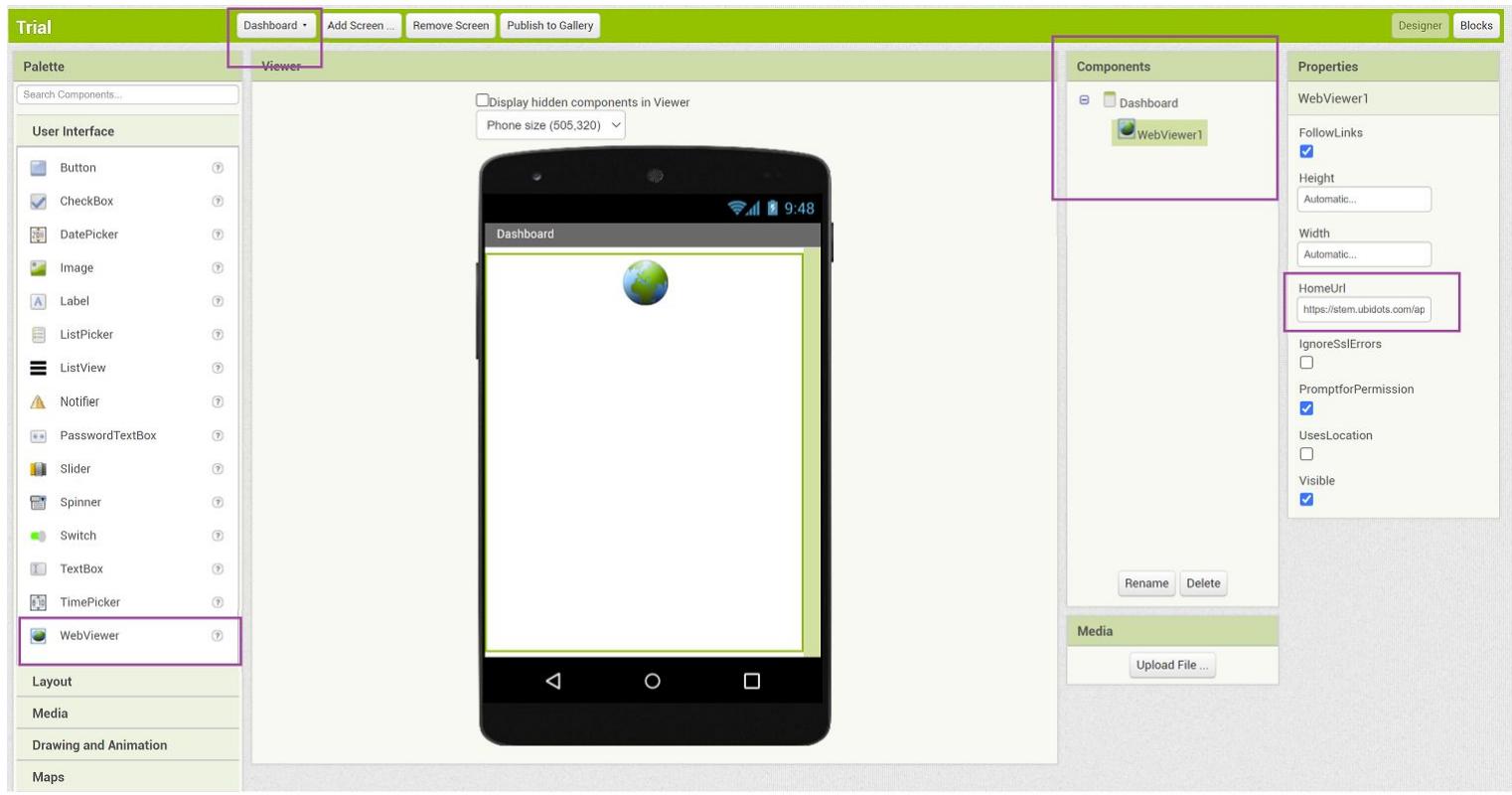
The screenshot shows the 'Dashboards' list page. A table displays a single row for a dashboard named 'IoT Workshop' created on '2/28/2023'. In the 'Actions' column for this row, a context menu is open, with the 'Share' option highlighted. Other options in the menu include 'Edit', 'Duplicate', 'Lock', 'Assign Organization', and 'Delete'. At the bottom of the page, there are buttons for 'CANCEL' and 'CREATE', and a dropdown for 'RECORDS PER PAGE' set to '10'.

The screenshot shows a 'Share dashboard' modal window. The title of the dashboard is 'IoT Workshop'. The window contains several configuration options:

- 'Select language': Set to 'English (United States)' (checked).
- 'Date range': Set to '2/28'.
- 'Hide navbar': A toggle switch is off.
- 'Display dashboard title': A toggle switch is off.
- 'Embed to your website': A URL field contains the link <https://stem.ubidots.com/app/dashboards/public/dashboard>. Below the URL are three icons: a lock, a copy, and a trash can. To the right of the URL is a 'Create new link' button.

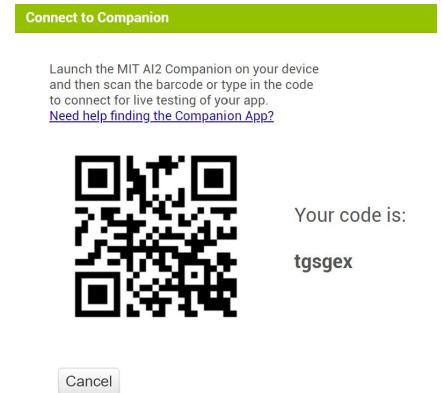
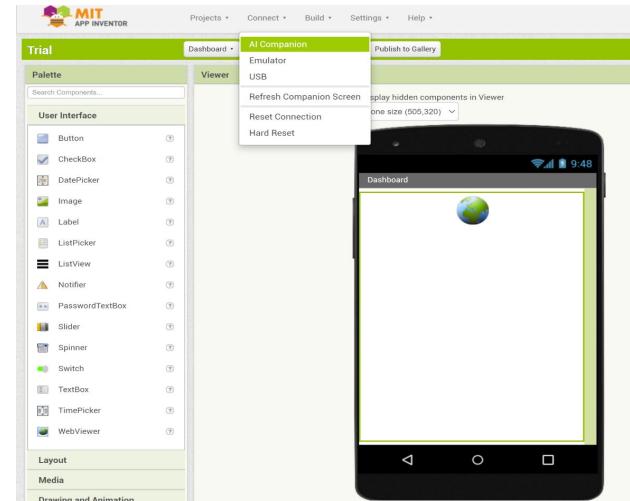
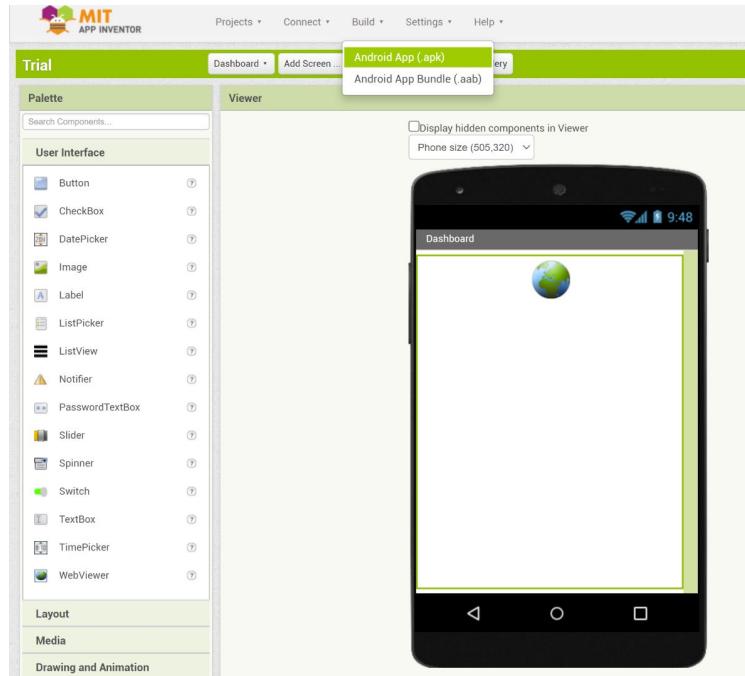
At the bottom of the modal is a 'Close' button.

6. Copy the shareable link and paste it into Web Viewer to embed the dashboard into the App.



7. Once the App is designed, we can test it as follows:

- Generating .apk for Android Users
- Generating the QR code for both iOS and Android users. For this we need to download the “MIT App Inventor” from Google Play Store or Apple App Store





Dashboard

Home

Dashboard



Humidity(%)

Last value |

31.00

Last Updated: 03/03/2023 05:12

Temperature(*C)

Last value |

24.80

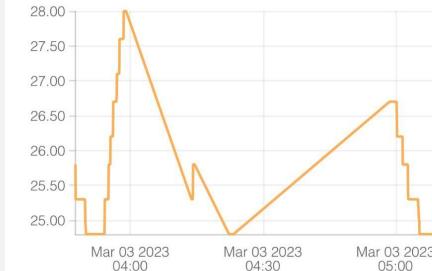
Last Updated: 03/03/2023 05:12

Chart

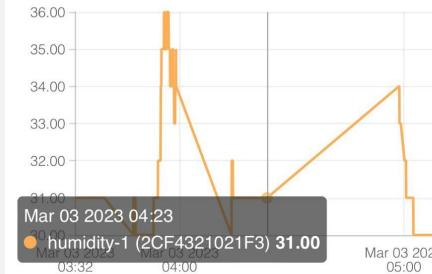
Home

Dashboard

Chart



Chart



Thank You Questions?

Build your next project:

Ref: <https://ieeexplore.ieee.org/document/9682788>