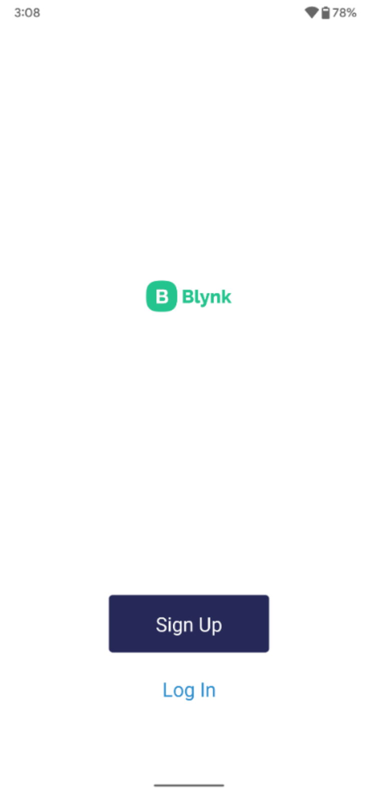
AES Practical 10

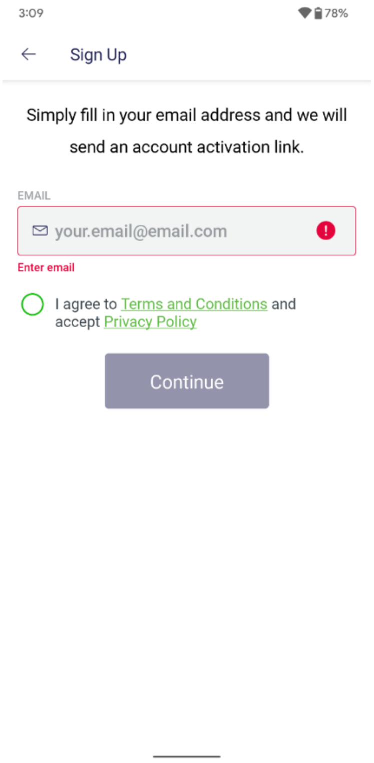
Blynk Iot App Configuration:

To control LED connected to NodeMCU remotely we need to install Blynk IoT app in our mobile as follows –

1. Download Blynk IoT app from play store or app store.



2. After downloading, create an account by clicking on Sign Up.



Simply fill in your email address and we will send an account activation link.

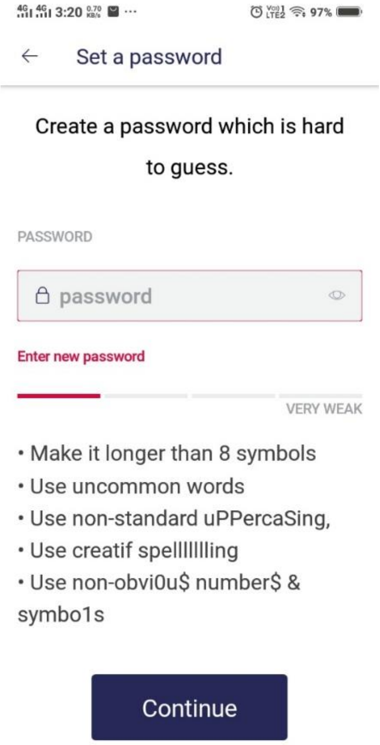
EMAIL

your.email@email.com

Enter email

○ I agree to Terms and Conditions and accept Privacy Policy

3. Enter your Email ID and then you will receive a link in your inbox where you need to enter a password for your account.



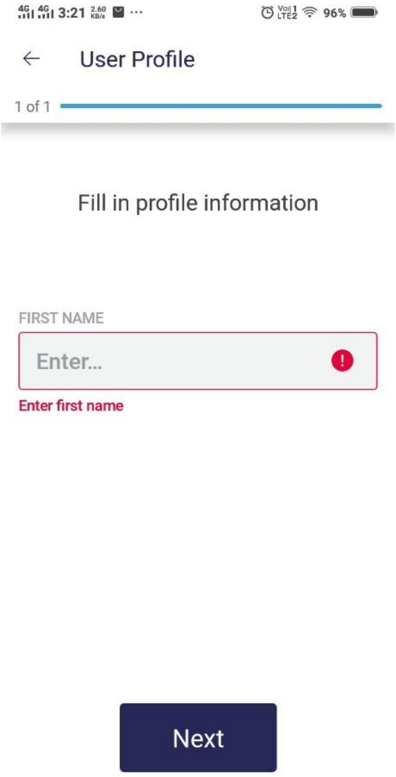
Create a password which is hard to guess.

PASSWORD

Enter new password

• Make it longer than 8 symbols
• Use uncommon words
• Use non-standard uPPercaSing,
• Use creatif spellllllling
• Use non-obvi0u$ number$ & symbo1s

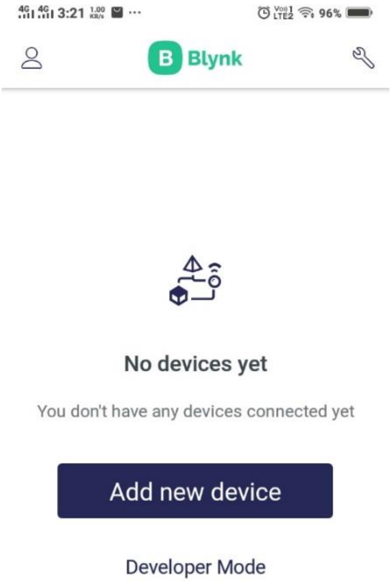
4. After this, you need to enter your first name to finish the setup of your account.



Fill in profile information

FIRST NAME

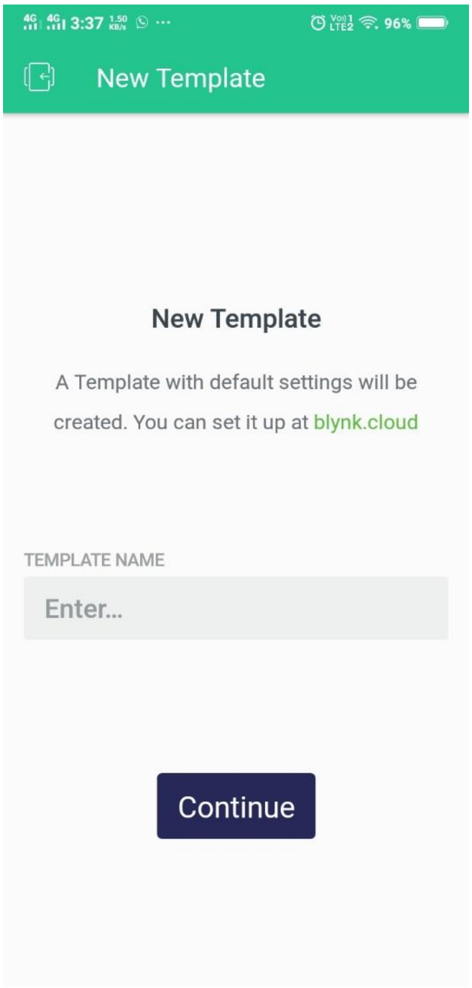
Enter first name



No devices yet

You don't have any devices connected yet

5. Click on Developer mode. Then you will be asked to enter a name for the template. Give a suitable template name and then click on continue.

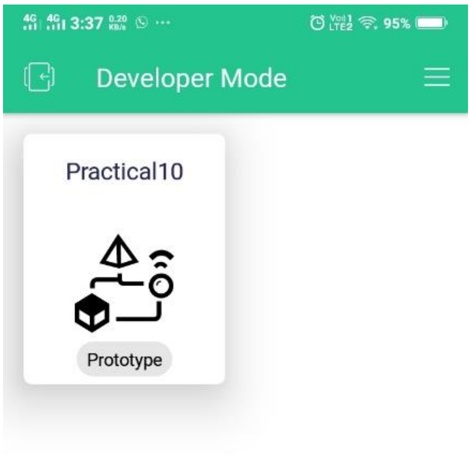


New Template

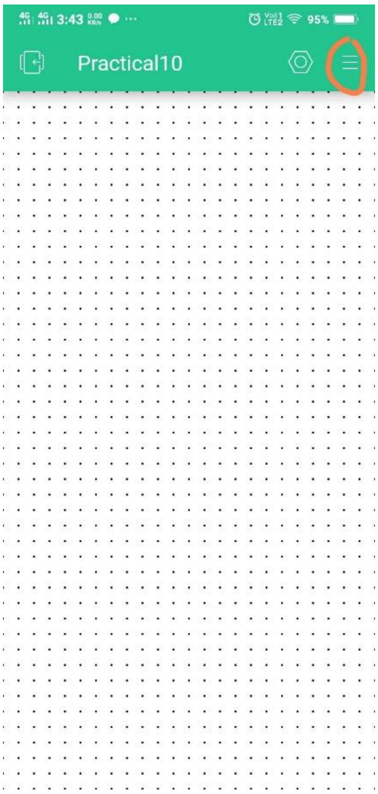
A Template with default settings will be created. You can set it up at blynk.cloud

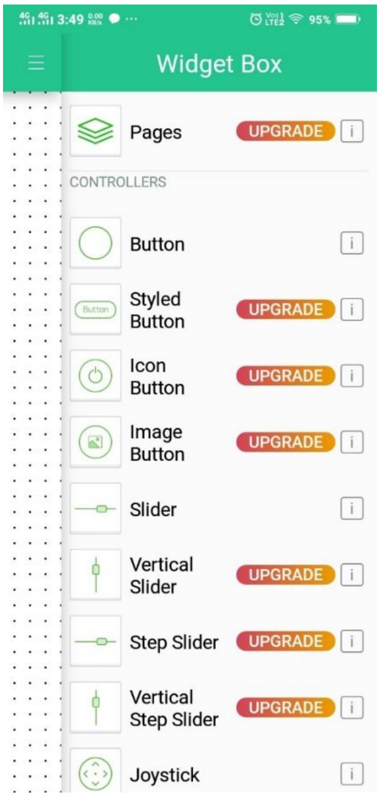
TEMPLATE NAME

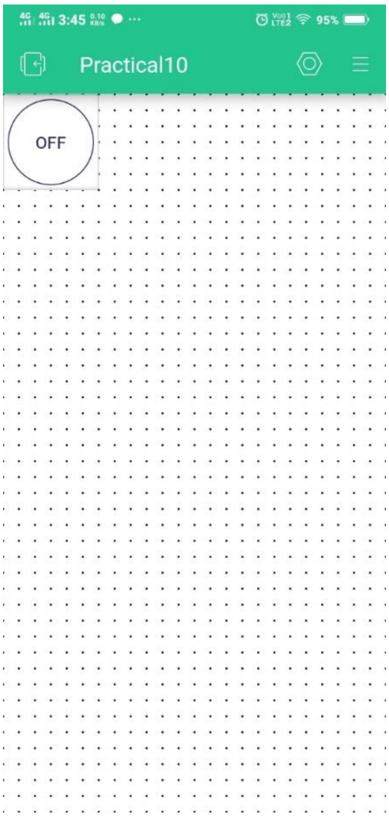
Enter...



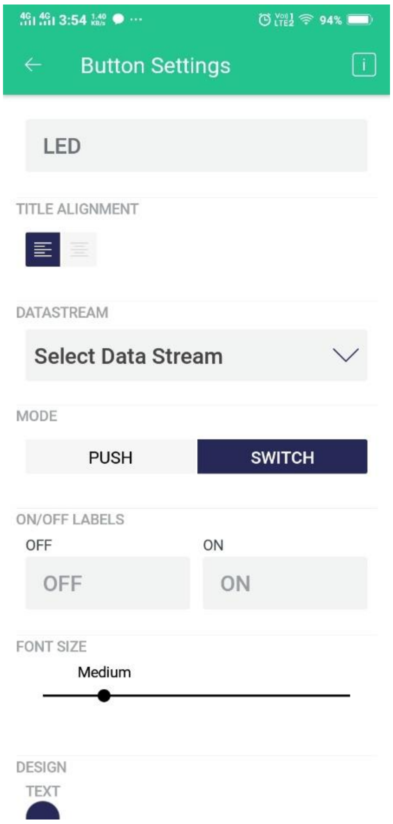
6. Now click on Prototype of Practical 10 a canvas will get opened. Then click on the circled part shown on the image and after that under controllers click on Button. A button with OFF written o it will be created on the canvas.







7. Now click on the button created it will open Button Settings. Enter title for the button and select the mode of the button to **SWITCH.**



Button Settings

LED

TITLE ALIGNMENT

DATASTREAM

Select Data Stream

MODE

PUSH [✓ SWITCH]

ON/OFF LABELS

OFF: OFF

ON: ON

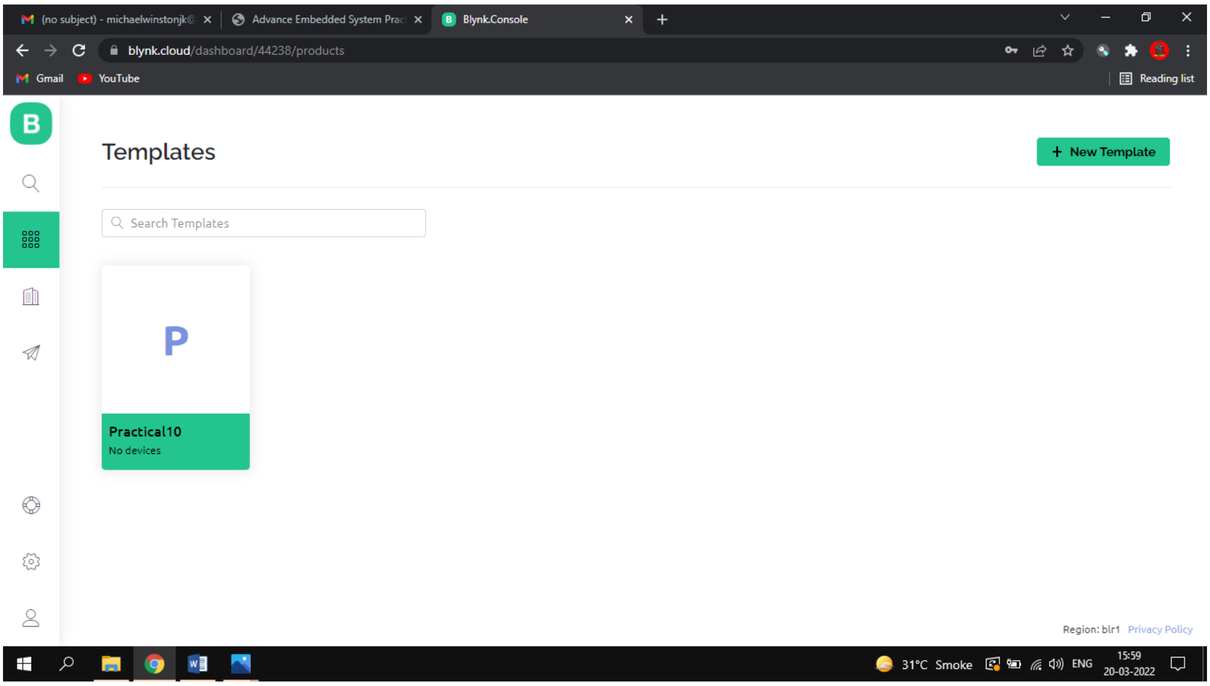
FONT SIZE

Medium

DESIGN

TEXT

8. Now, Login to the desktop site of blynk i.e blynk cloud and then go to Templates section there you can see the template which we created on our mobile is visible click on it.



blynk.cloud/dashboard/44238/products

Templates

+ New Template

Search Templates

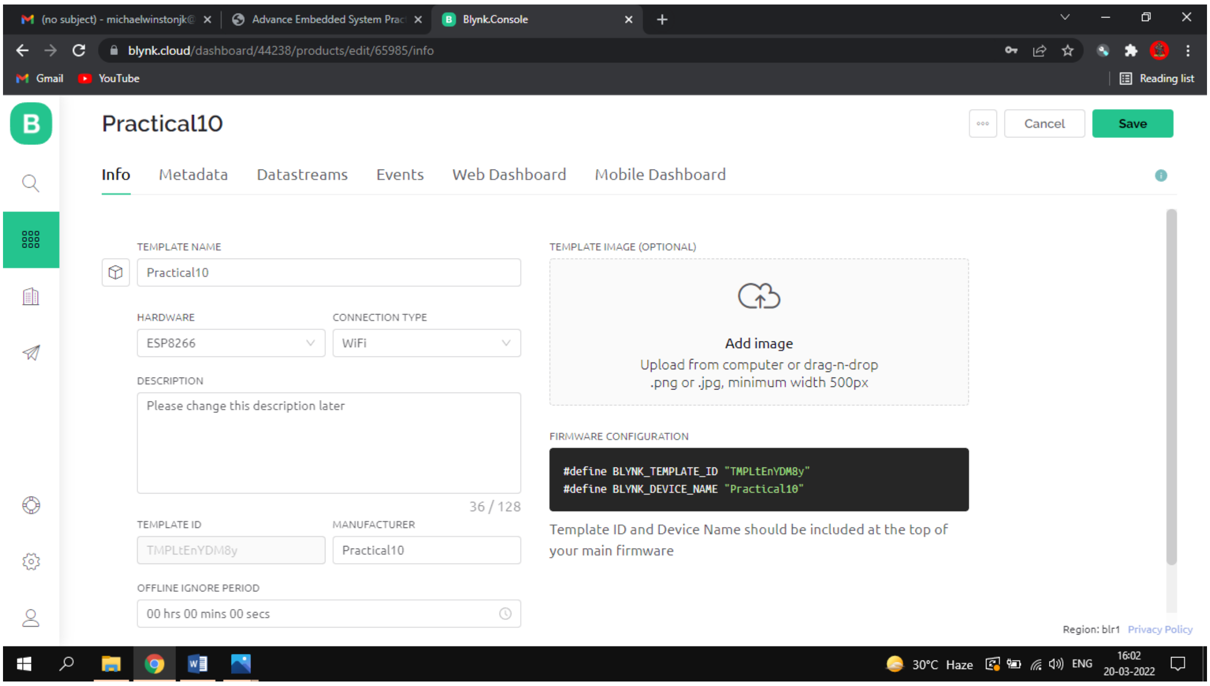
P

Practical10

No devices

Region: blr1 Privacy Policy

9.Now click on edit and change the **HARDWARE** to **esp8266** by selecting esp8266 from the list and then click on save.


(no subject) - michaelwinstonjk
Advance Embedded System Prac
Blynk.Console
blynk.cloud/dashboard/44238/products/edit/65985/info
Gmail
YouTube
Reading list
Practical10
Cancel
Save
Info
Metadata
Datastreams
Events
Web Dashboard
Mobile Dashboard
Add image
Upload from computer or drag-n-drop
.png or .jpg, minimum width 500px
Region: blr1
Privacy Policy
30°C Haze
ENG
16:02
20-03-2022

TEMPLATE NAME: Practical10

HARDWARE: ESP8266

CONNECTION TYPE: WiFi

DESCRIPTION: Please change this description later

36 / 128

TEMPLATE ID: TMPLtEnYDM8y

MANUFACTURER: Practical10

OFFLINE IGNORE PERIOD: 00 hrs 00 mins 00 secs

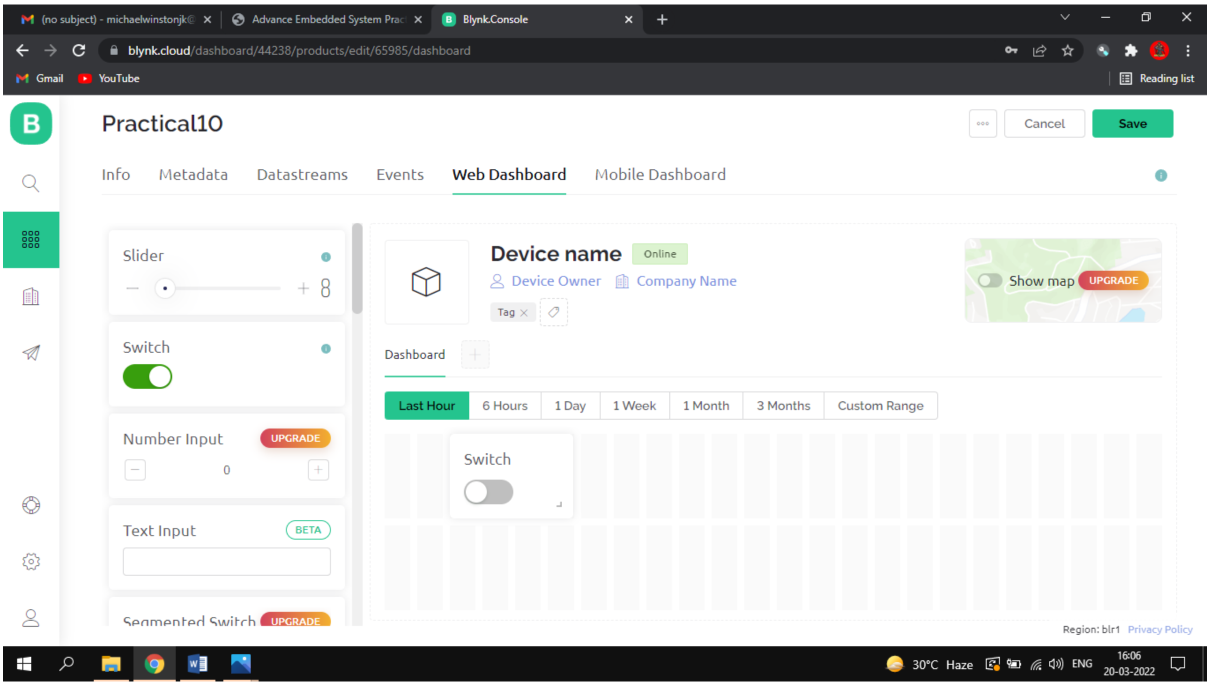
TEMPLATE IMAGE (OPTIONAL)

FIRMWARE CONFIGURATION

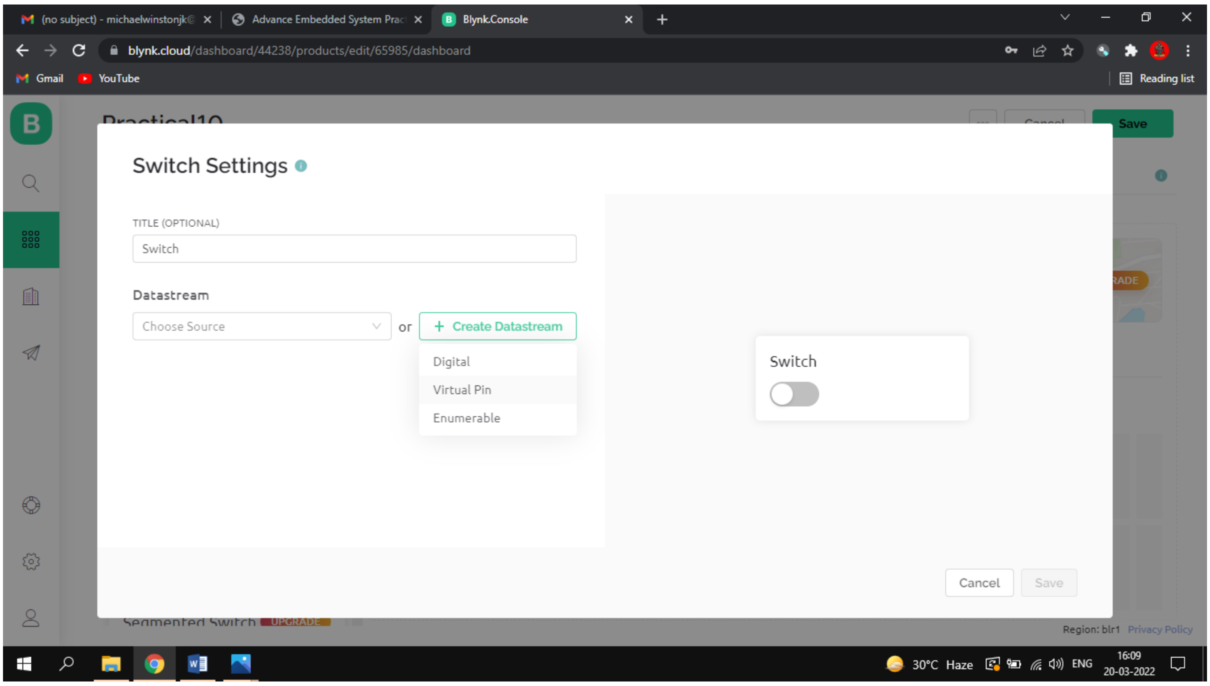
```
#define BLYNK_TEMPLATE_ID "TMPLtEnYDM8y"
#define BLYNK_DEVICE_NAME "Practical10"
```

Template ID and Device Name should be included at the top of your main firmware

10. Now head towards the **Web Dashboard** of our template and then click on edit. After that drag and drop **Switch** widget from the widgets present on left of the page to the canvas.



11. Now click on settings button of the switch widget it will open Switch Settings in that click on **Create Datastream** select **Virutal Pin.**



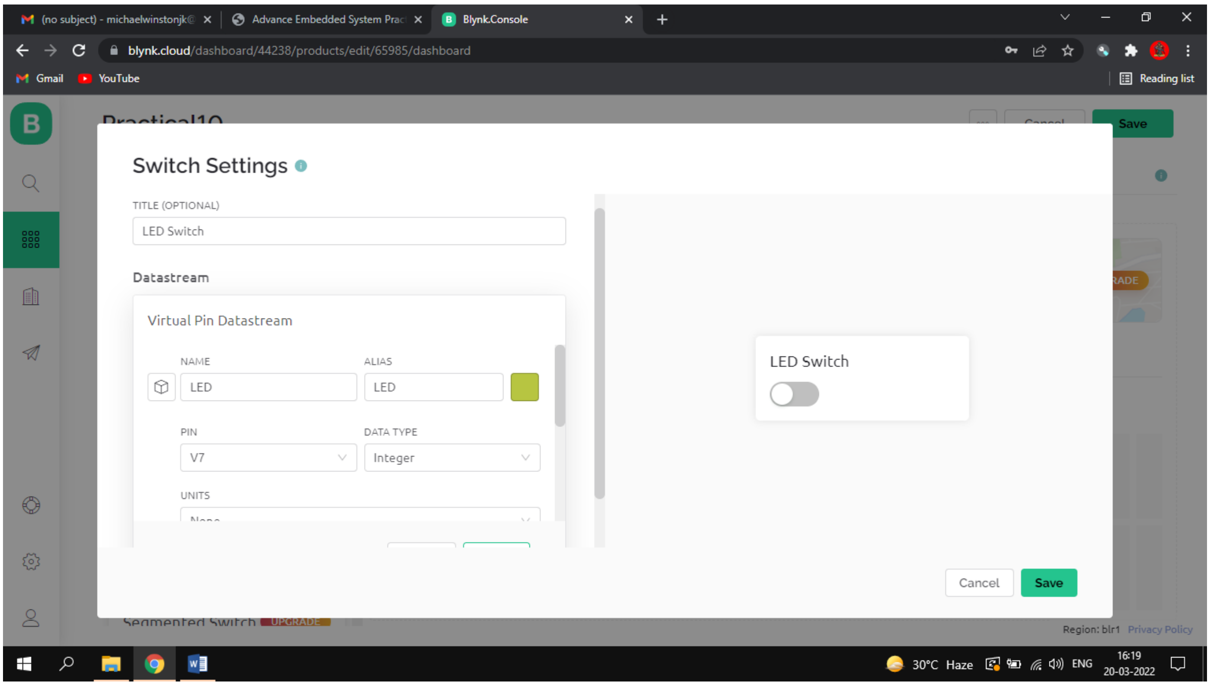
blynk.cloud/dashboard/44238/products/edit/65985/dashboard

Switch Settings

TITLE (OPTIONAL): Switch

Datastream: Choose Source or + Create Datastream

12. Now enter Title and name in Virtual Pin Destination and then click on create and after that click on save.



blynk.cloud/dashboard/44238/products/edit/65985/dashboard

Switch Settings

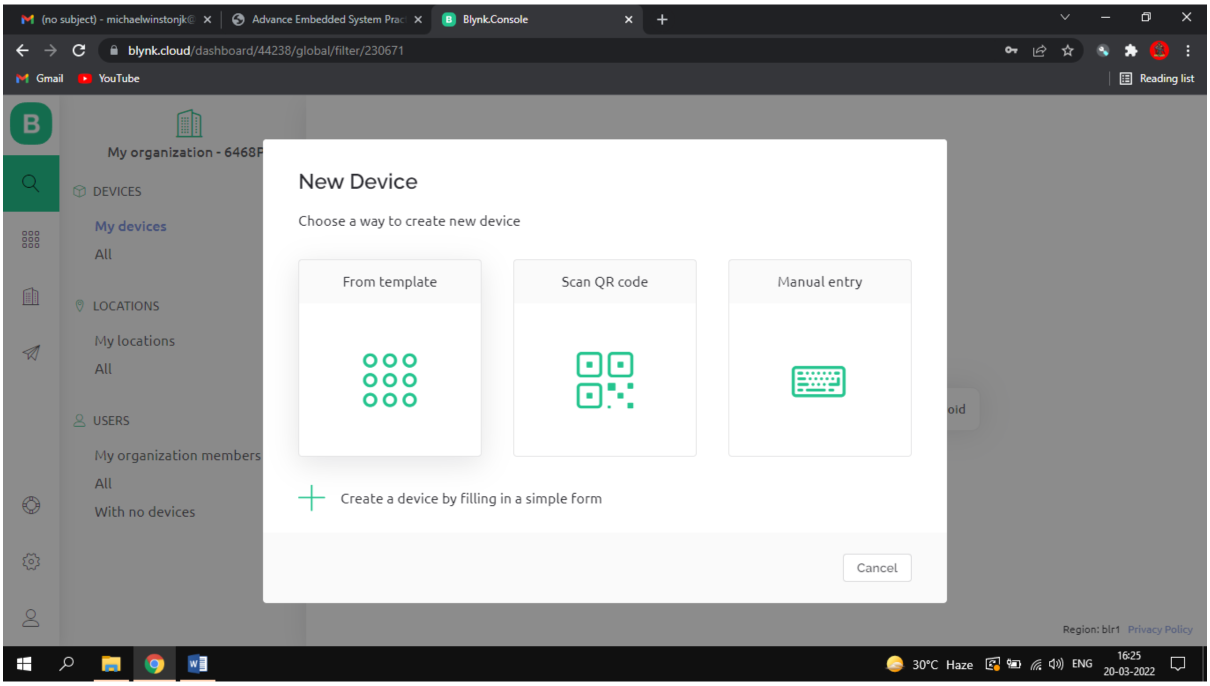
TITLE (OPTIONAL): LED Switch

Datastream

Virtual Pin Datastream

NAME: LED
ALIAS: LED
PIN: V7
DATA TYPE: Integer
UNITS: None

13. Now click on search on top left side and then click on **My devices.** Then click on **New Device** and click on **From Template** after that select template to Practical 10 or any suitable name given by you and then click on **Create.**



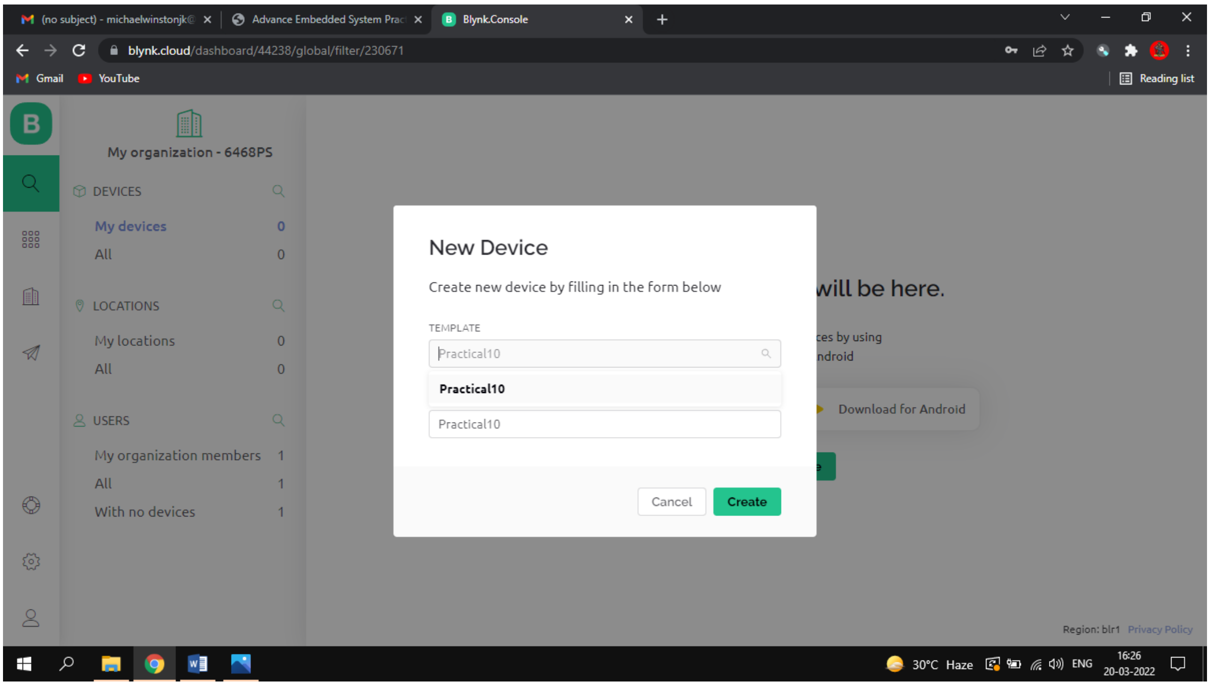
New Device

Choose a way to create new device

From template | Scan QR code | Manual entry

Create a device by filling in a simple form

Cancel



New Device

Create new device by filling in the form below

TEMPLATE

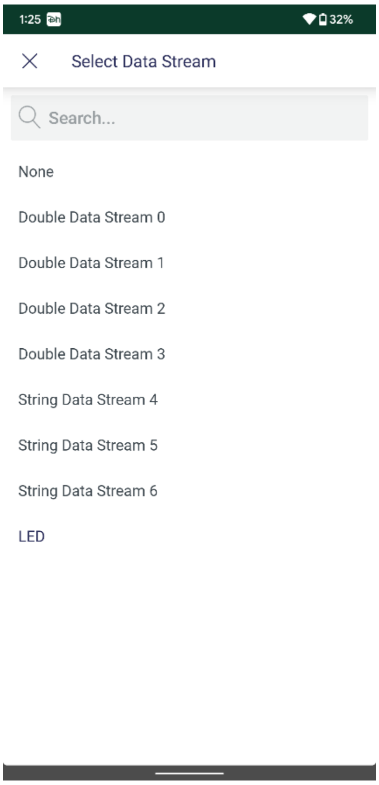
Practical10

Practical10

Practical10

Cancel Create

14. After that go to Blynk Iot App and click on the LED Button created on the template. Then click on Select Data Stream. A list of Datastream appears in that select 'LED'.



1:25 32%

Select Data Stream

Search...

None

Double Data Stream 0

Double Data Stream 1

Double Data Stream 2

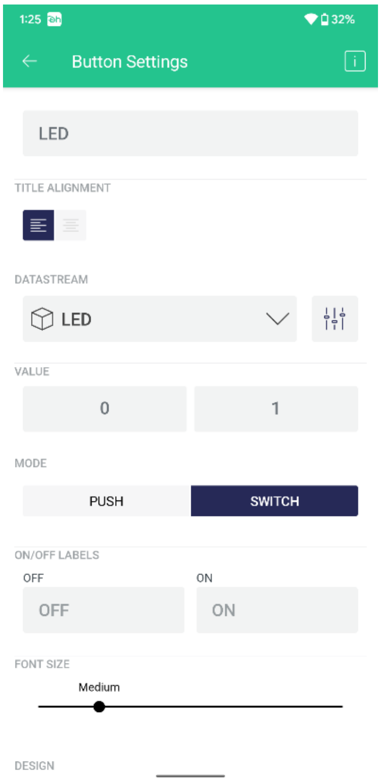
Double Data Stream 3

String Data Stream 4

String Data Stream 5

String Data Stream 6

LED



1:25 32%

Button Settings

LED

TITLE ALIGNMENT

DATASTREAM

LED

VALUE

0 1

MODE

PUSH SWITCH

ON/OFF LABELS

OFF: OFF

ON: ON

FONT SIZE

Medium

DESIGN

With this Blynk Iot app configuration is done.

Uploading Firmware:

1. Download following zip file:-

https://github.com/blynkkk/blynk-library/releases/download/v0.6.1/Blynk_Release_v0.6.1.zip

2. Extract this file → we can see 2 folder **tools** and **libraries**
3. The content of tools will be copied and pasted on C:\Program Files (x86)\ Arduino\tools
4. The content of libraries will be copied and pasted on C : \ Program Files (x86) \ Arduino \ libraries

Arduino IDE Setup:

Open arduino IDE and perform following configuration –

1. File → Preferences → In **Additional Boards Manager** text box enter:

   https://arduino.esp8266.com/stable/package_esp8266com_index.json →

   OK
2. Tools → Boards → Board Manager → Search for **esp8266 by ESP8266 community 2.6.3** → Install
3. Tool → Board → Select NodeMCU
4. Tools → Select COM port for communication

Code:

```
#define
BLYNK_P
RINT
Serial
#include
<ESP8266
WiFi.h>
#include <BlynkSimpleEsp8266_SSL.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon) → Auth
Tokens  → Copy allchar auth[] =
"YourAuthToken";
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";    // Wi-Fi Name char pass[] = "YourPassword";          //
Wi-Fi Passwordvoid setup()
{
  // Debug
  console
  Serial.begin(
  9600);
  Blynk.begin(
  auth, ssid,
  pass);
}
void loop()
{
  Blynk.run();
}
```

Note: - Before uploading, make sure to paste your authorization token into the auth [] variable. Also makesure to load your Wifi network settings into the Blynk.begin(auth, “ssid”, “pass”) function.

Now compile and Run the code.

Output:

Click the button from Blynk Iot app to switch ON and OFF the LED. We can test from remotely operating.