

Firestore Home Automation System Using NodeMCU

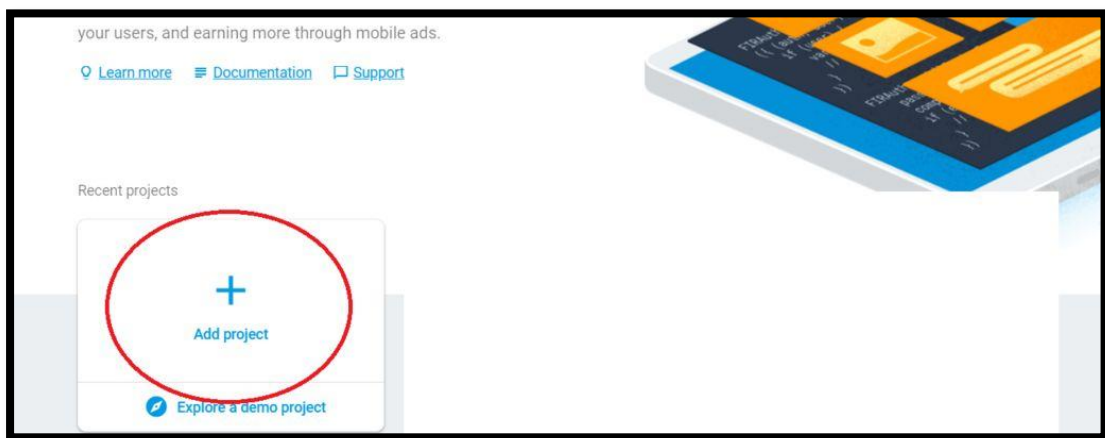
Firestore provides a quick way to persist sensory data collected at the device level, and it works great with the Android APIs, which is supported by Android Things. A lot of mobile and device programmers that I have come across struggle with server-side programming. Firestore can really help bridge that gap and make it easier. It will be interesting to see developers use its offline features. If you are new to IoT or in general any device that collects data and needs to transmit it over networks, the golden rule to be assumed is that network connectivity cannot be assumed. As a result, you will need to collect the data offline and when network is available, transmit this over to your server. Firestore with its offline feature can really make this simple for a lot of developers.

I will divide this project in three part

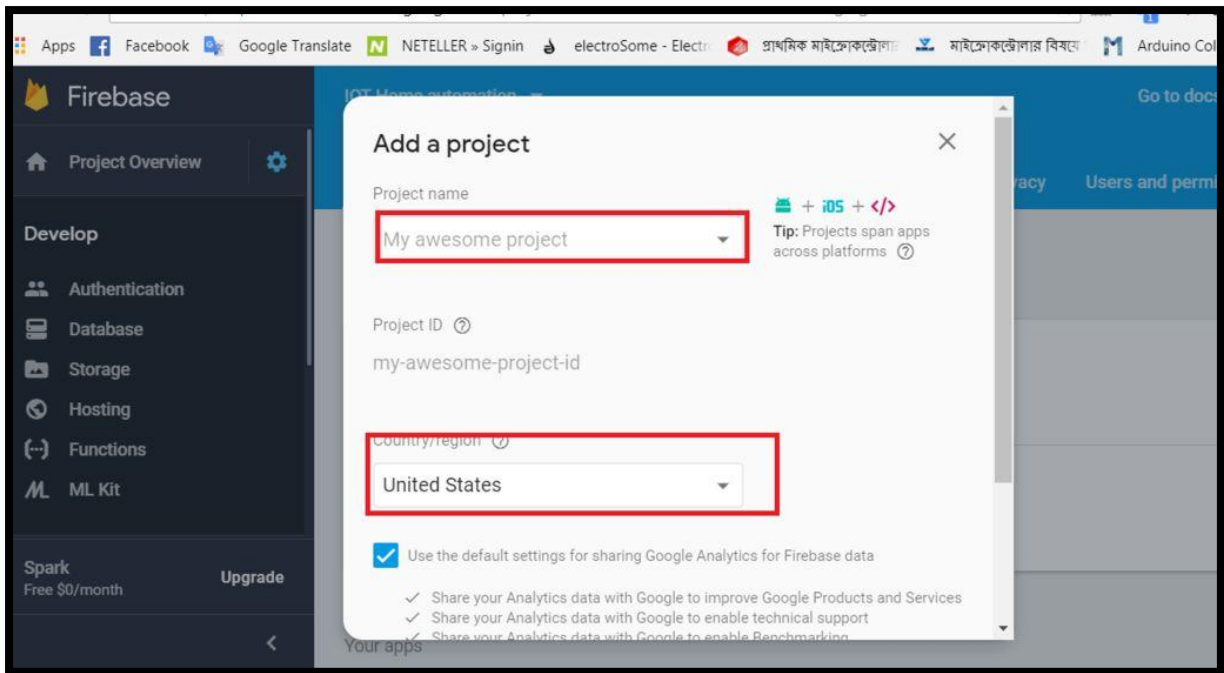
1. Creating firestore account
2. App making
3. 3.Arduino program part

➤ Creating Firestore Account

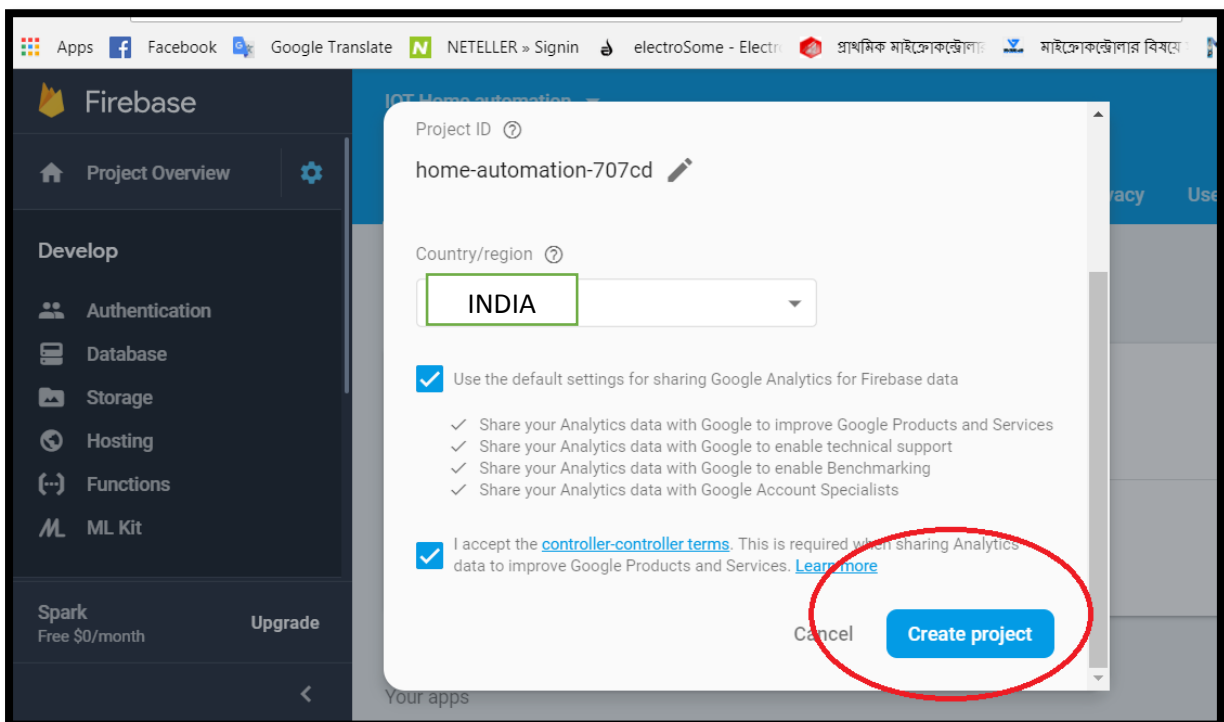
- 1) First go to <https://console.firebase.google.com/> and login
- 2) Click on Add project



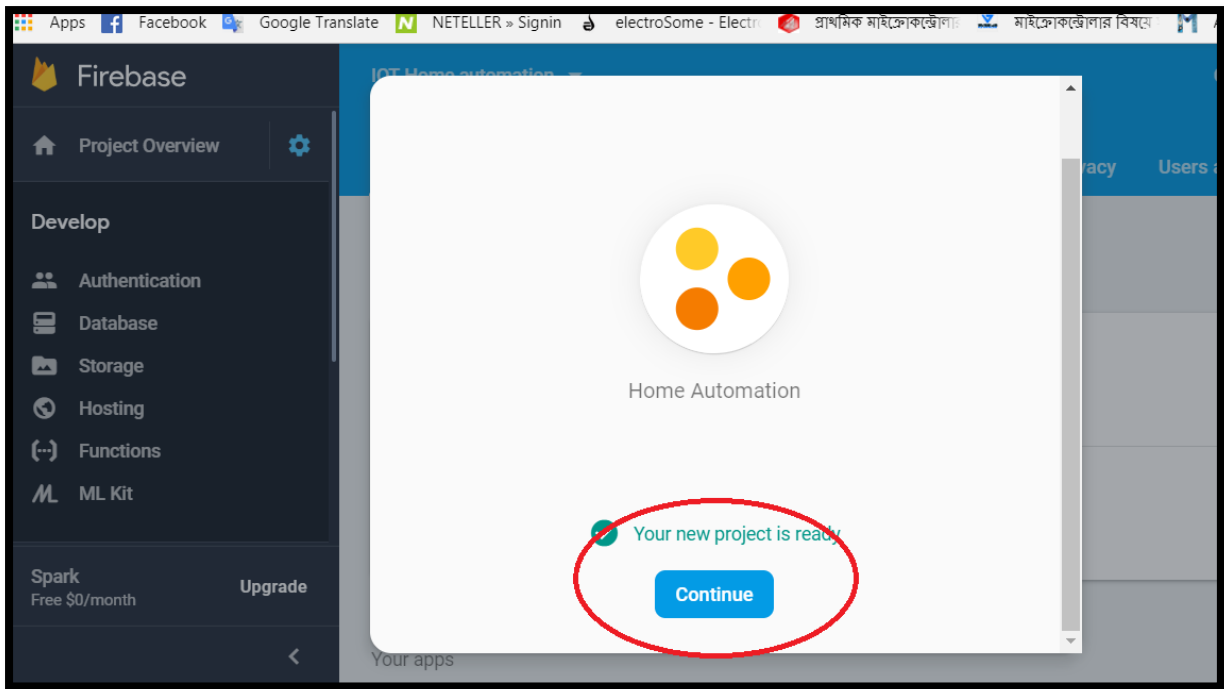
- 3) Then Show Like This Interface and Give Project Name and Select Country Name



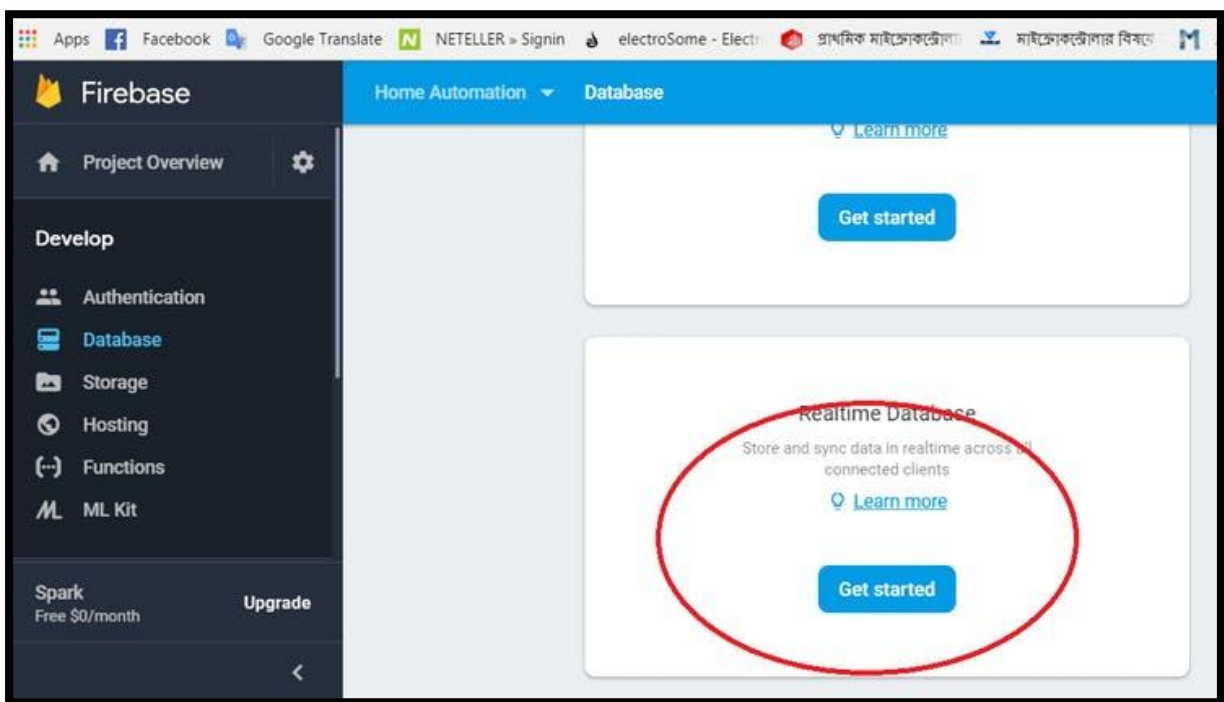
4) Here I Select Project Name Home Automation and Select My Country Then Press Create



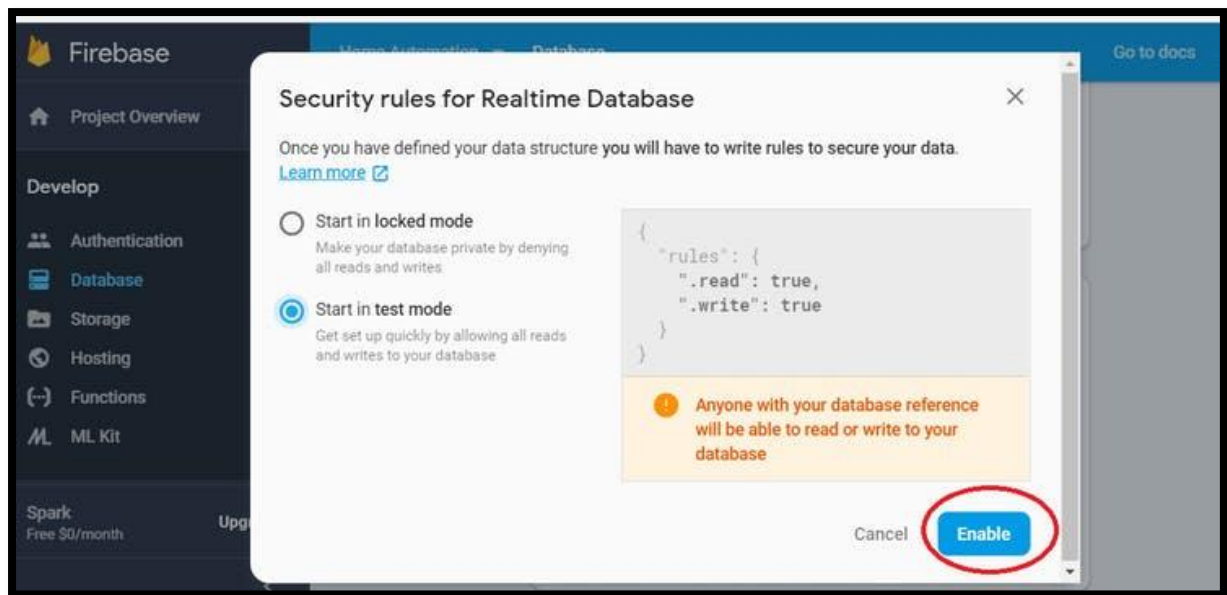
5) After a moment you should see the screen shown below; click on Continue.



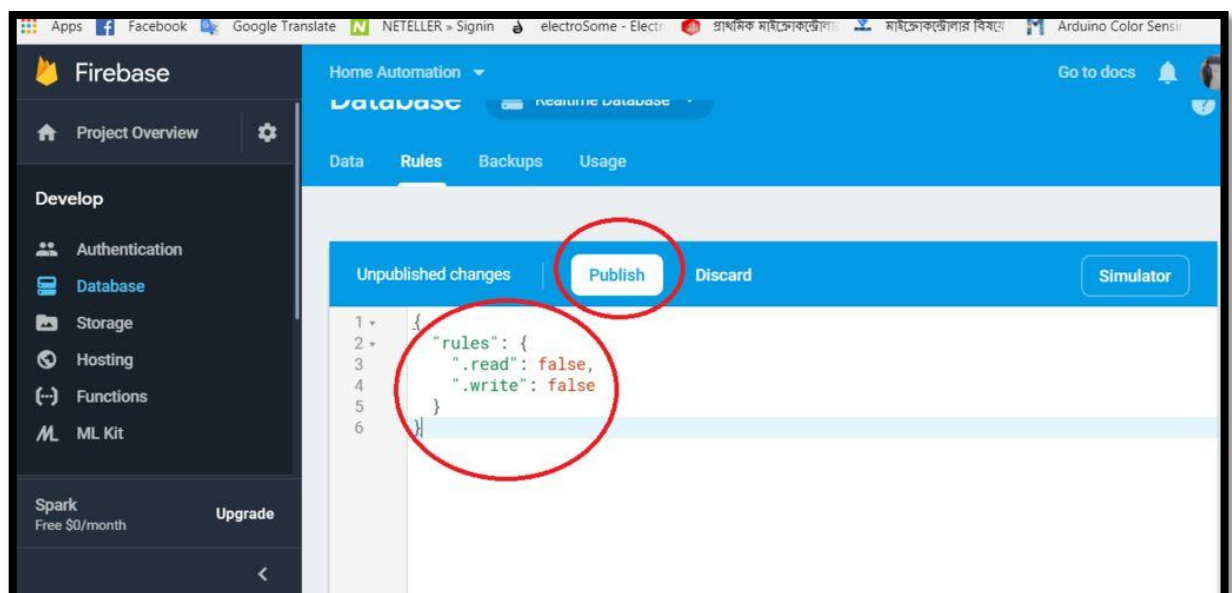
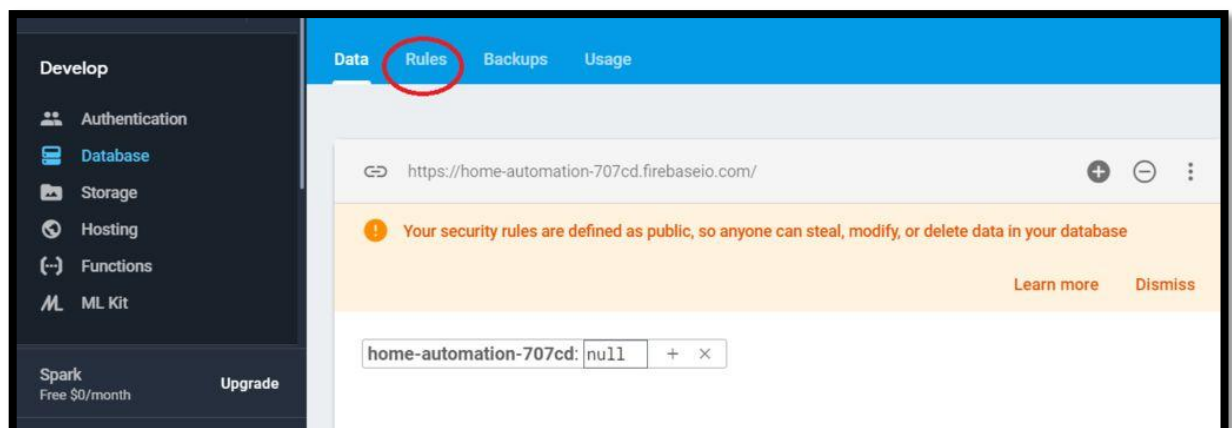
6) Then go to Get started.



7) Now enable it.



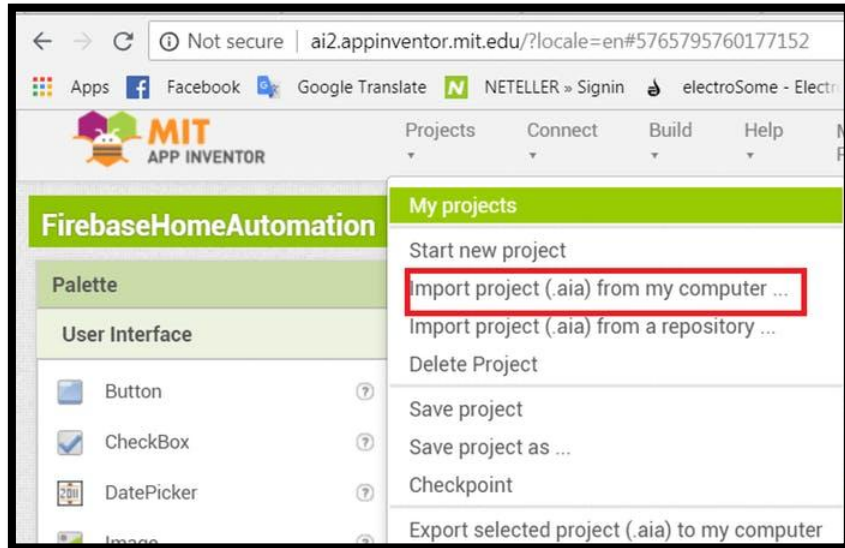
8) Go to Rules and edit the code as seen in the images. Copy this link and paste in MIT app firebase component.



Part one is complete...

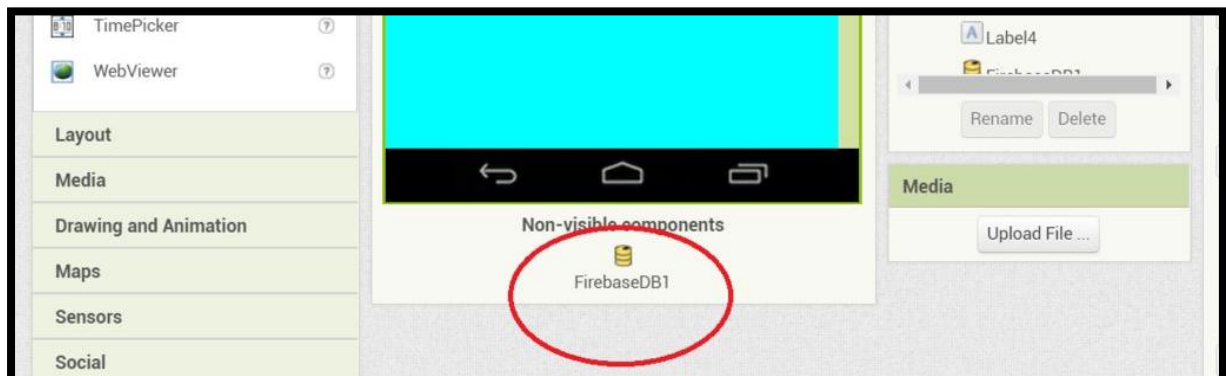
➤ **Making the App**

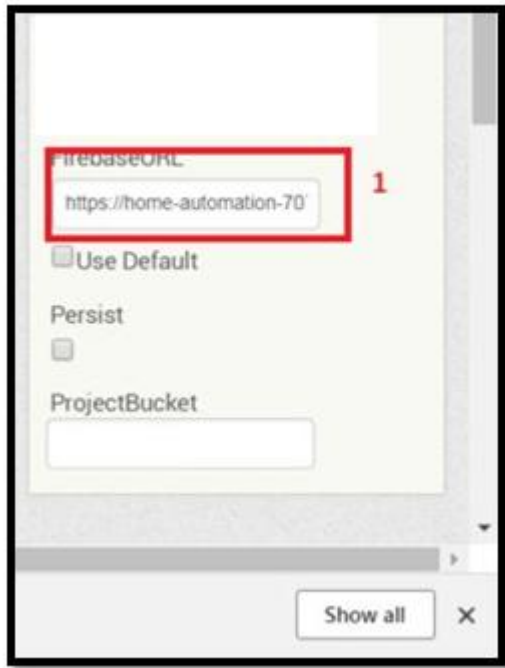
1) Now go to ai2.appinventor.mit.edu to open this file.



2) Now go to My Projects and import the Homeautomation.aia file on your computer.

3) Click on the option circled in red, and follow the below image's directions.





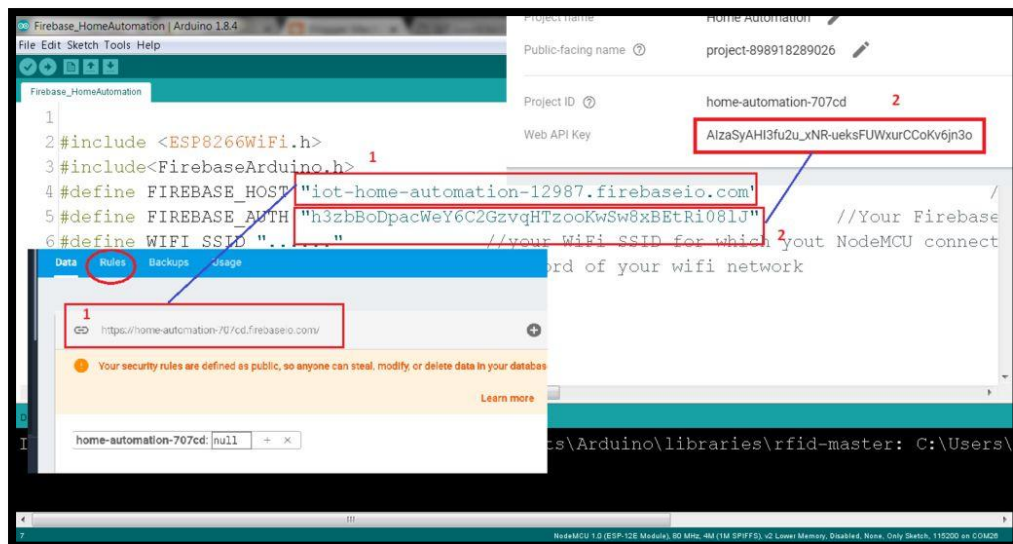
OK, now this part is over...

➤ **Programming Arduino**

You have to need to make some changes for your Arduino code:

- ***FIREBASE_HOST*** : copy and paste into the Firebase database
- ***FIREBASE_AUTH*** : copy and paste into Project Settings

Setup your WIFI name and password.



NODEMCU Code: -

```

#include <ESP8266WiFi.h>
#include<FirebaseArduino.h>
#define FIREBASE_HOST "my-smart-home-99033.firebaseio.com"
//Your Firebase Project URL goes here without "http:" , "\" and "/"
#define WIFI_SSID "my_empire" //your WiFi SSID for which your
NodeMCU connects
#define WIFI_PASSWORD "@myempire" //Password of your wifi
network

int Relay1= D5;
int val1;

#define Relay2 D6
int val2;

#define Relay3 D7
int val3;

#define Relay4 D8
int val4;

```

```

void setup()
{
  Serial.begin(115200);                // Select the same baud rate if you
  want to see the datas on Serial Monitor
  pinMode(Relay1,OUTPUT);
  pinMode(Relay2,OUTPUT);
  pinMode(Relay3,OUTPUT);
  pinMode(Relay4,OUTPUT);

  digitalWrite(Relay1,LOW);
  digitalWrite(Relay2,HIGH);
  digitalWrite(Relay3,HIGH);
  digitalWrite(Relay4,HIGH);

  WiFi.begin(WIFI_SSID,WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected:");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST); //Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.setInt("S1",1);        //Here the varialbe "S1", "S2", "S3" and "S4" needs to
  be the one which is used in our Firebase and MIT App Inventor
  Firebase.setInt("S2",0);
  Firebase.setInt("S3",0);
  Firebase.setInt("S4",0);

```



```

}
//void firebaseconnect()
//{
// Serial.println("Trying to reconnect");
//  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
// }

void loop()
{
// if (Firebase.failed())
//  {
//    Serial.print("setting number failed:");
//    Serial.println(Firebase.error());
//    firebaseconnect();
//    return;
//  }

  val1=Firebase.getString("S1").toInt();           //Reading the value of the
  variable Status from the firebase

  if(val1==1)                                       // If, the Status is 1, turn on the Relay1
  {
    digitalWrite(Relay1,LOW);
    Serial.println("light 1 ON");
  }
  else if(val1==0)                                 // If, the Status is 0, turn Off the Relay1
  {
    digitalWrite(Relay1,HIGH);
    Serial.println("light 1 OFF");
  }
}

```

```
val2=Firebase.getString("S2").toInt();           //Reading the value of the  
variable Status from the firebase
```

```
if(val2==1)                                     // If, the Status is 1, turn on the Relay2  
{  
    digitalWrite(Relay2,LOW);  
    Serial.println("light 2 ON");  
}  
else if(val2==0)                               // If, the Status is 0, turn Off the Relay2  
{  
    digitalWrite(Relay2,HIGH);  
    Serial.println("light 2 OFF");  
}
```

```
val3=Firebase.getString("S3").toInt();           //Reading the value of the  
variable Status from the firebase
```

```
if(val3==1)                                     // If, the Status is 1, turn on the Relay3  
{  
    digitalWrite(Relay3,LOW);  
    Serial.println("light 3 ON");  
}  
else if(val3==0)                               // If, the Status is 0, turn Off the Relay3  
{  
    digitalWrite(Relay3,HIGH);  
    Serial.println("light 3 OFF");  
}
```

```
val4=Firebase.getString("S4").toInt();           //Reading the value of the  
variable Status from the firebase
```

```
if(val4==1)                                // If, the Status is 1, turn on the Relay4
{
    digitalWrite(Relay4,LOW);
    Serial.println("light 4 ON");
}
else if(val4==0)                           // If, the Status is 0, turn Off the Relay4
{
    digitalWrite(Relay4,HIGH);
    Serial.println("light 4 OFF");
}
}
```