

How to send Data to Google Firebase Using ESP8266

Check out the Video Link for a step-by-step guide: <https://www.youtube.com/watch?v=7VXbVHssiQU>

IoT being an integral part of cloud computing, several cloud vendors make IoT part of the services they render. Also, with the emergence of IoT, there are different platform that makes IoT implementation available even to individuals. In this tutorial, I will be taking you through steps in sending data from NodeMCU, an IoT development platform to Google Firebase.

To ensure all necessary things are in place, you will have to set up your computer to communicate with NodeMCU or ESP8266 before you can follow along this tutorial.

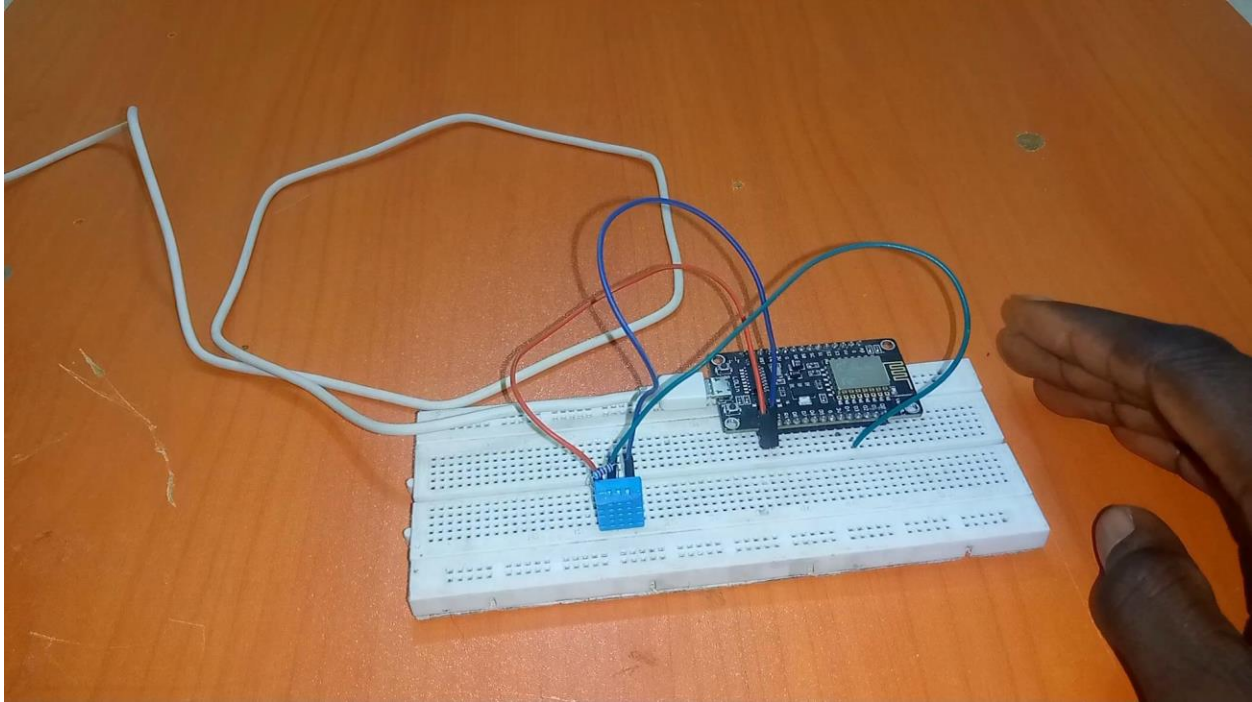
In this tutorial, we will be using DHT 11 sensor to send Temperature and Humidity data to Google Firebase.

Materials Needed

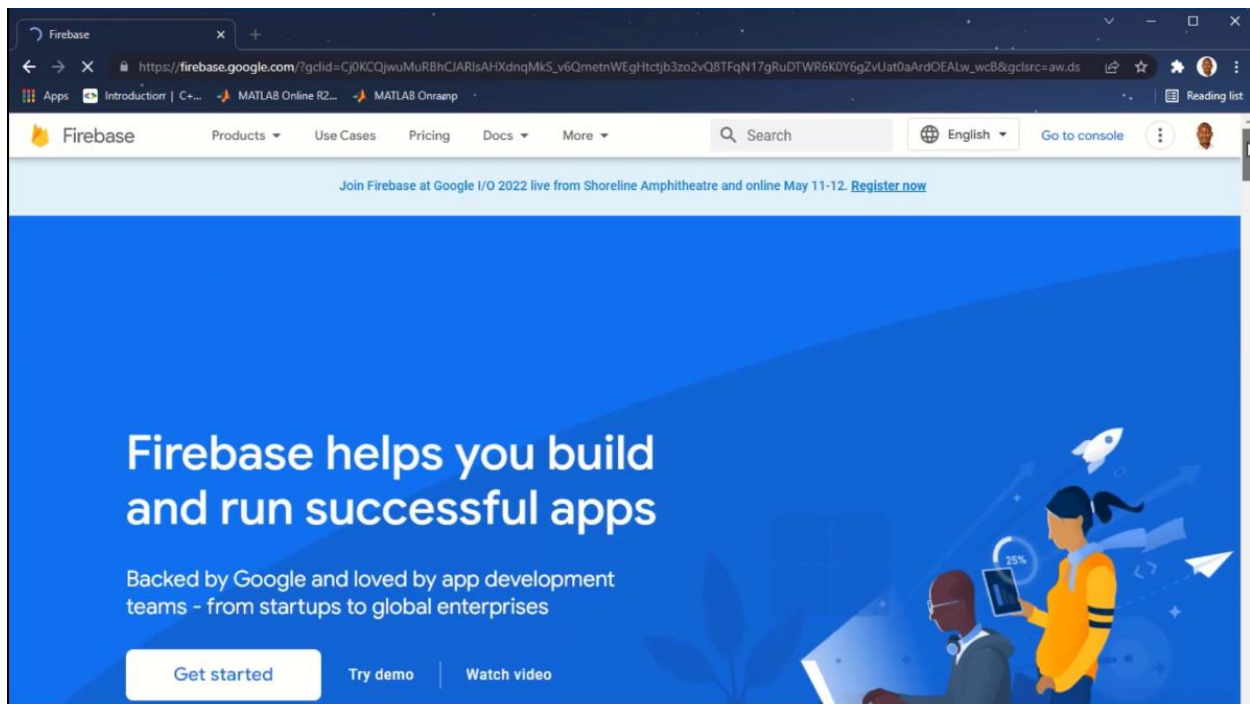
1. NodeMCU or ESP8266
2. DHT 11 sensor
3. 10K resistor
4. Personal Computer
5. Arduino IDE software
6. Jumper Wires

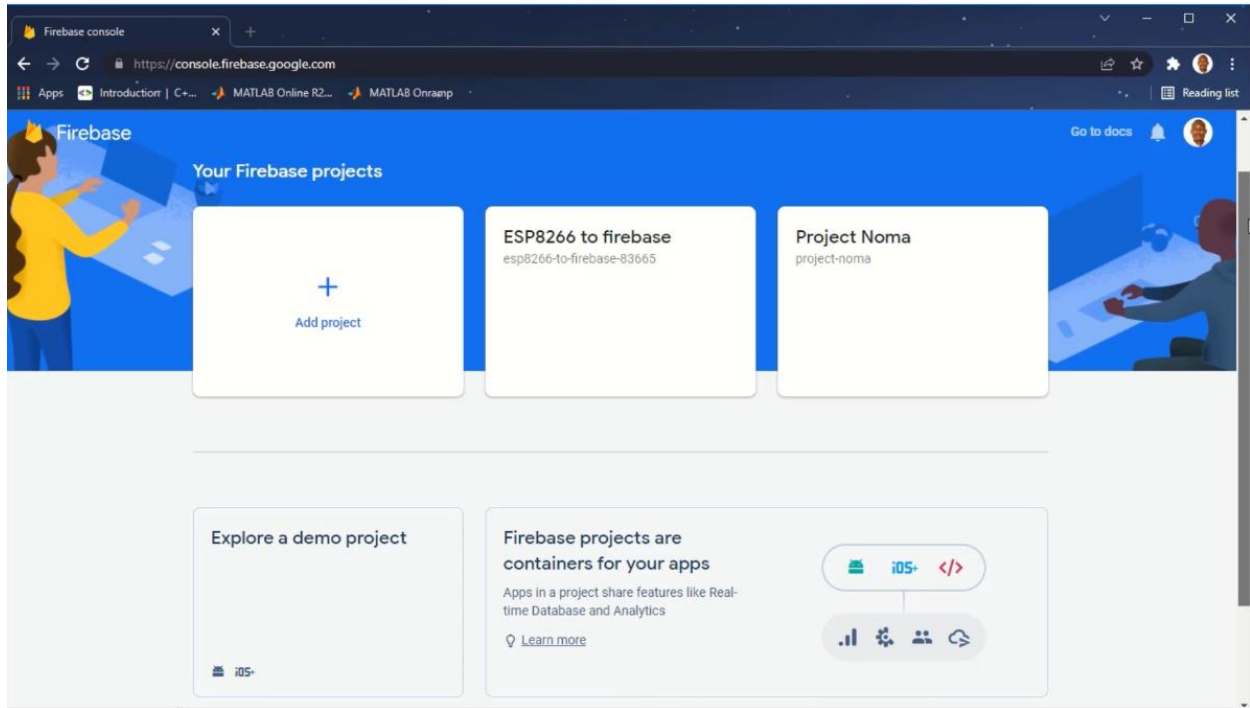
Step 1: The very first step is to ensure that your PC has been configured to communicate with ESP 8266. If you haven't did this already, now is the time. Once you have configured your PC, come right back to continue following this tutorial.

Step 2: Hardware Setup: Connect the NodeMCU and the DHT 11 to the breadboard, depending on your model of DHT 11, it can be a 3-pin ready made model or a 4-pin model, in this tutorial, I am using a 3-pin model. If you are using a 3-pin model, connect the VCC to 3V of the NodeMCU, GND of the DHT 11 to GND on the NodeMCU and connect the data pin to Pin 4 of the NodeMCU. If it is 4-pin model you are using, connect Pin 1 of the DHT11 to 3V of the NodeMCU, connect the 10K resistor between pin 1 and pin 2 of the DHT 11, leave pin 3 of the DHT 11 sensor unconnected and connect pin 4 of the DHT 11 to GND on the NodeMCU. Connect the NodeMCU via a USB cable to your computer and you are good to go.

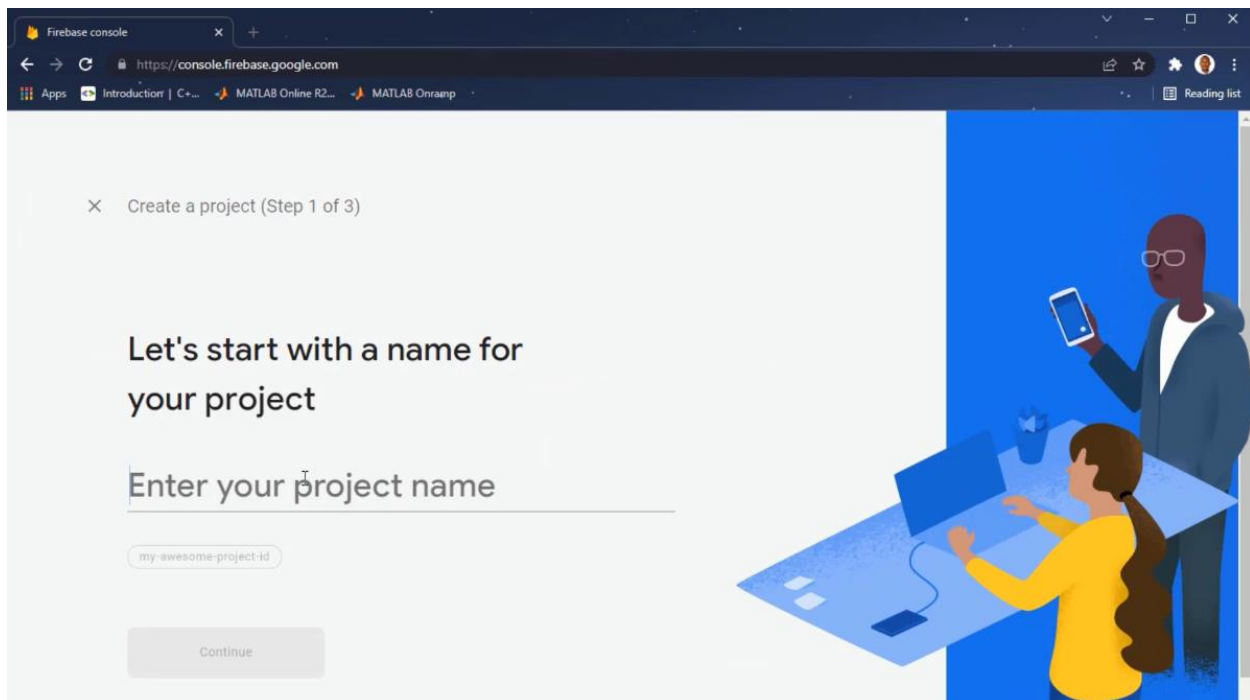


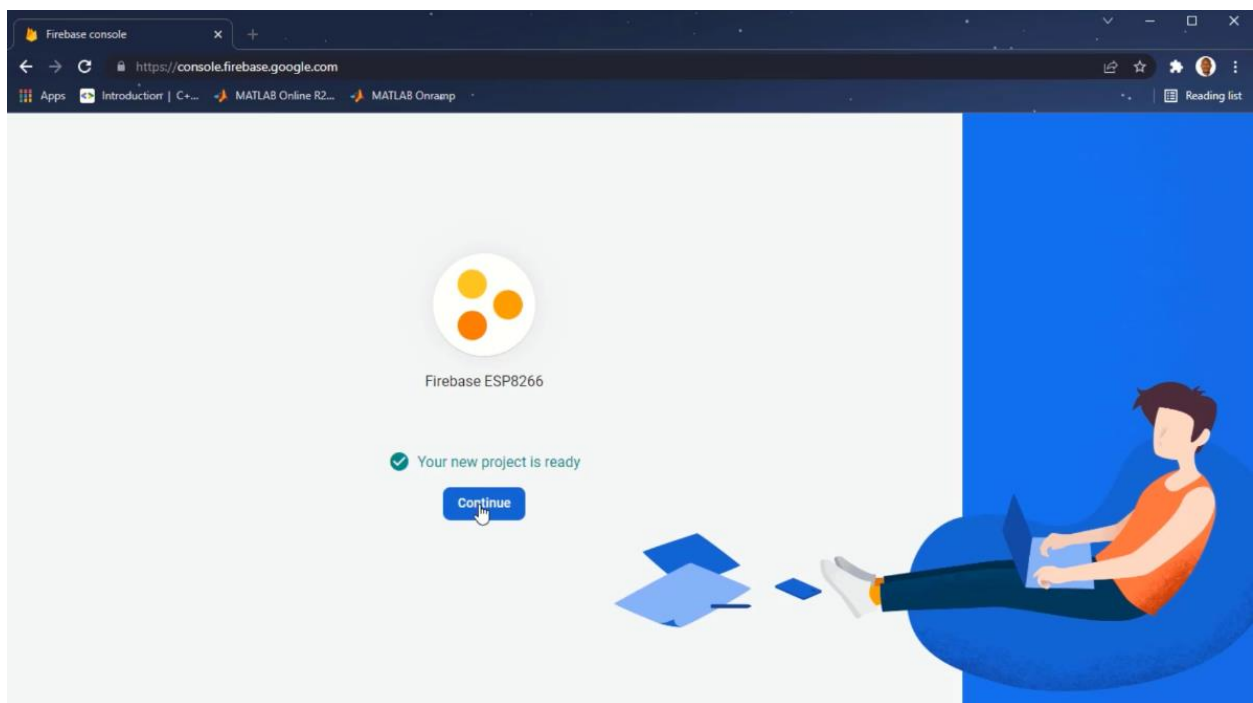
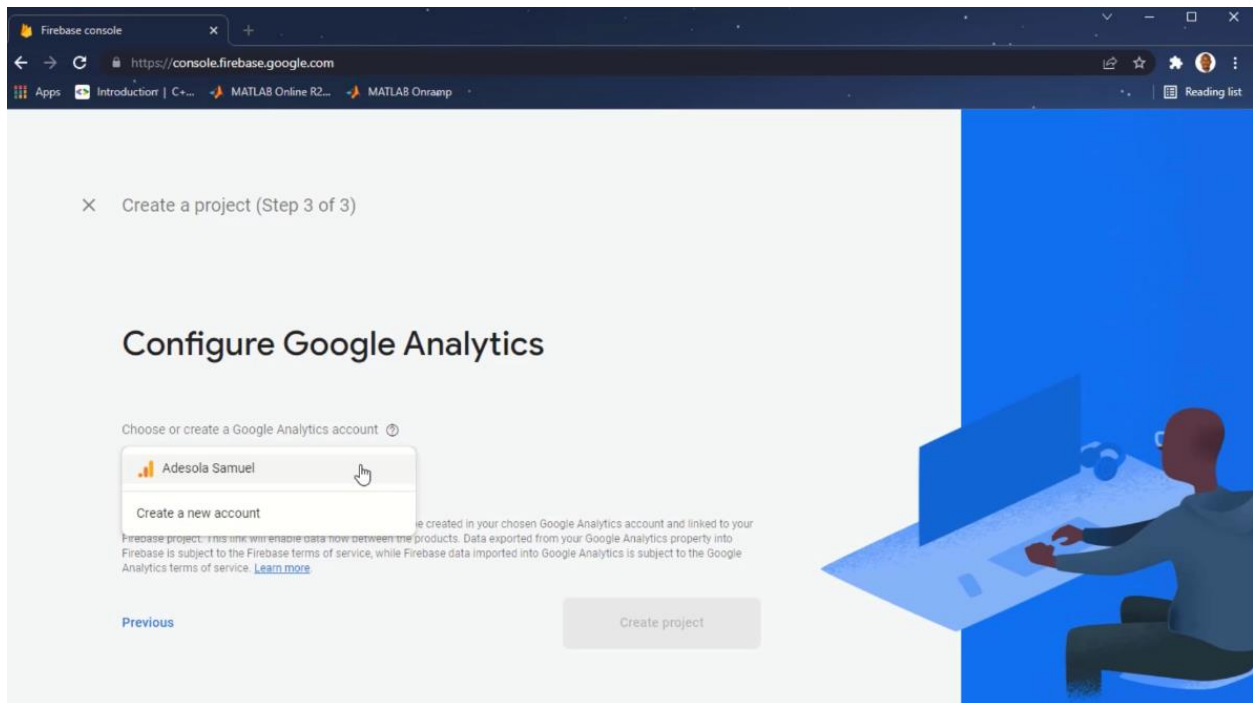
Step 3: After all necessary hardware connections, what you want to do is to configure Google Firebase to receive data from your EXSP8266, to access Google Firebase, open your browser and head over to firebase.google.com, and click on **Get Started**. You will be directed to the console where you can add new project, click on **Add Project** to create a new project.



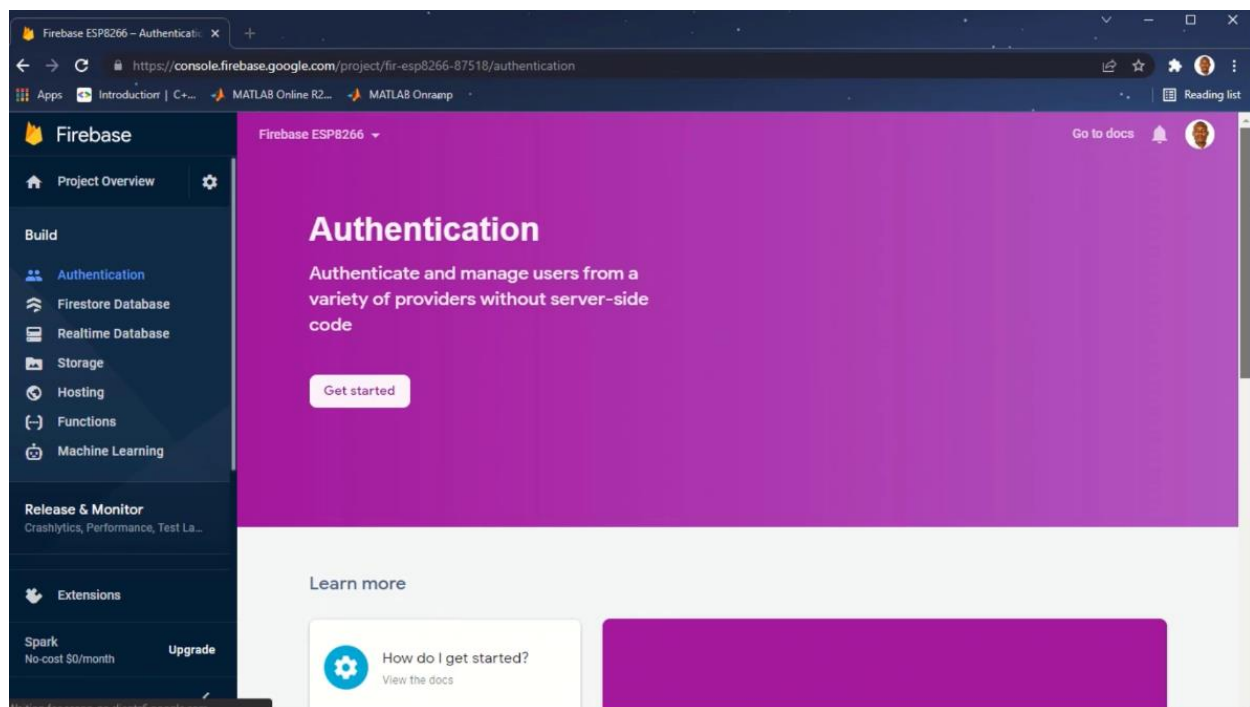
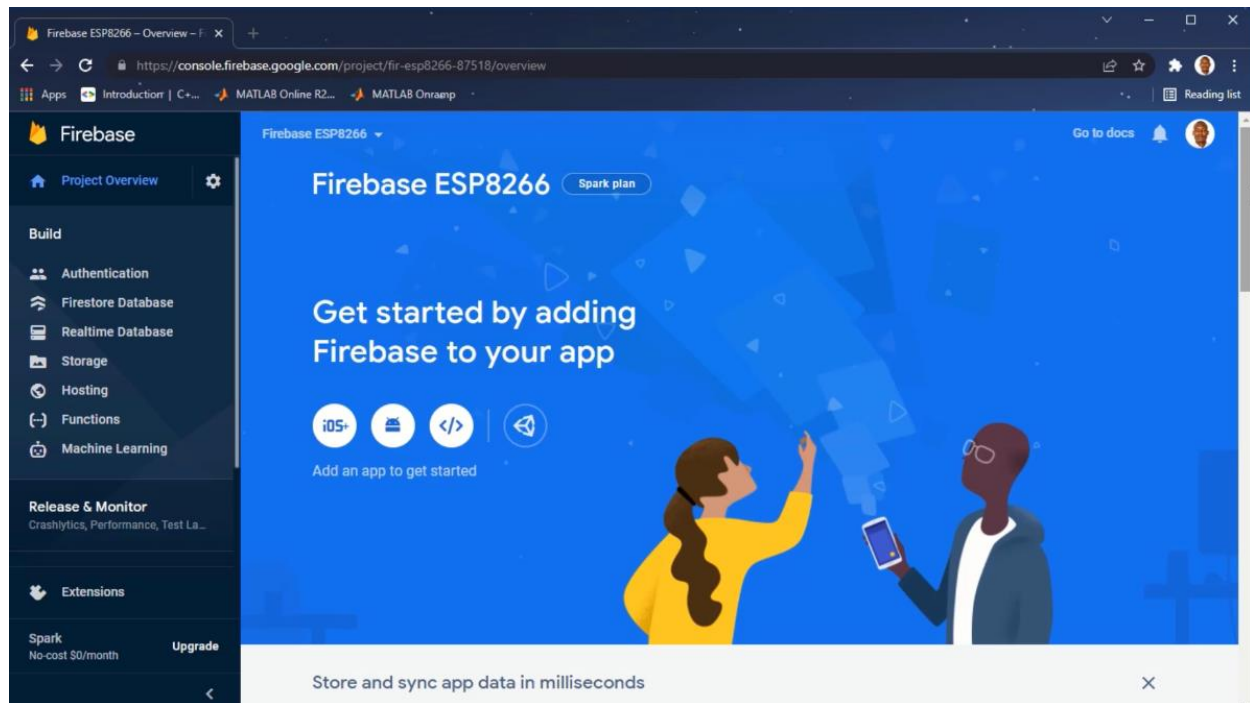


Step 4: After clicking on Add project, you will be asked to give your project a name, type in a suitable name for your project and click on **Continue**. This is step 1, on step 2 page, leave everything as default and lastly on step 3, select an account for your project, this will usually be the google account you are using for firebase. Click on **continue** and allow firebase to finish setting up your project.

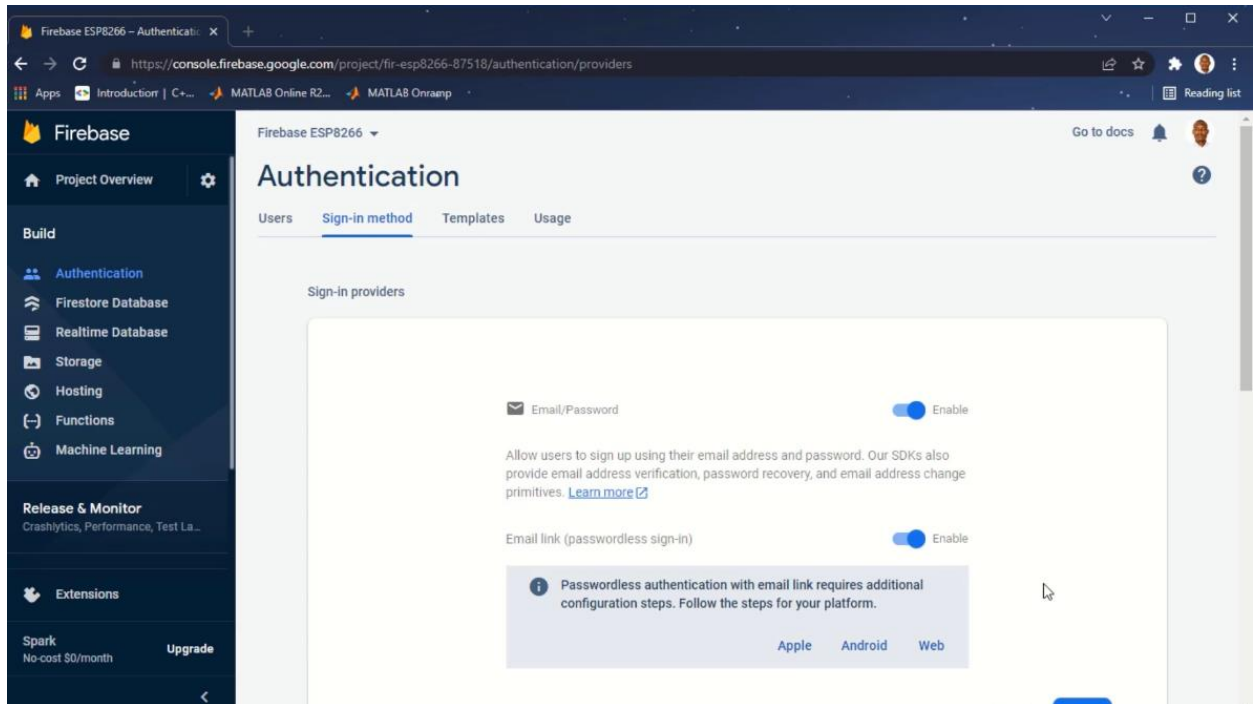
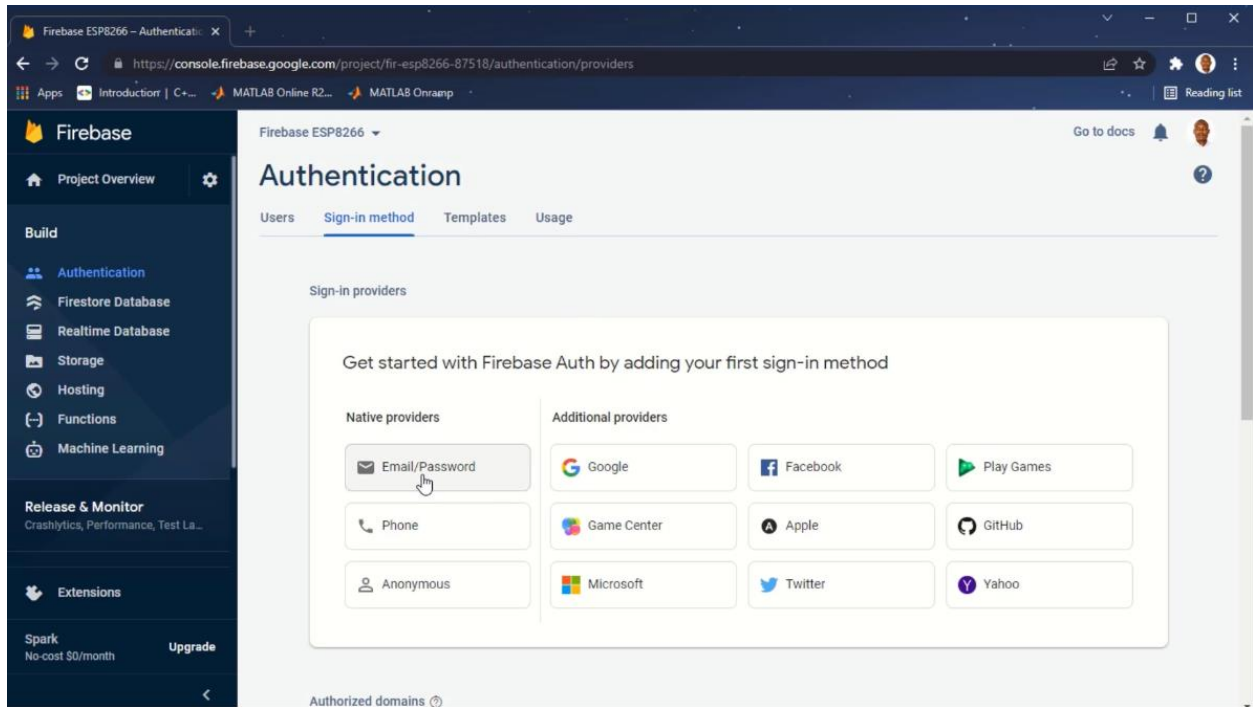


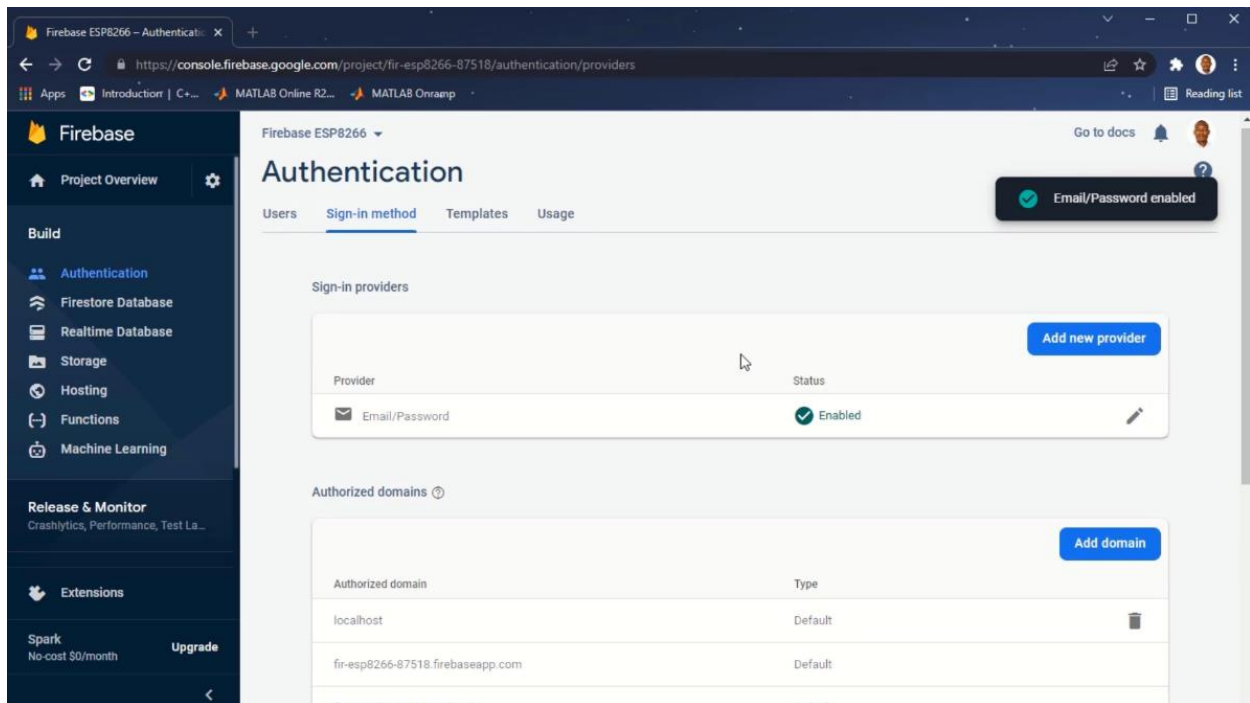


Step 5: After creating your project, you will be presented with the Firebase working console, here we will be configuring authentications and also setting up the Realtime database. Before creating our database, we need to define set rules on how to authenticate users sending data to our database, on the side bar, click on **Authentication** and click on **Get Started**.

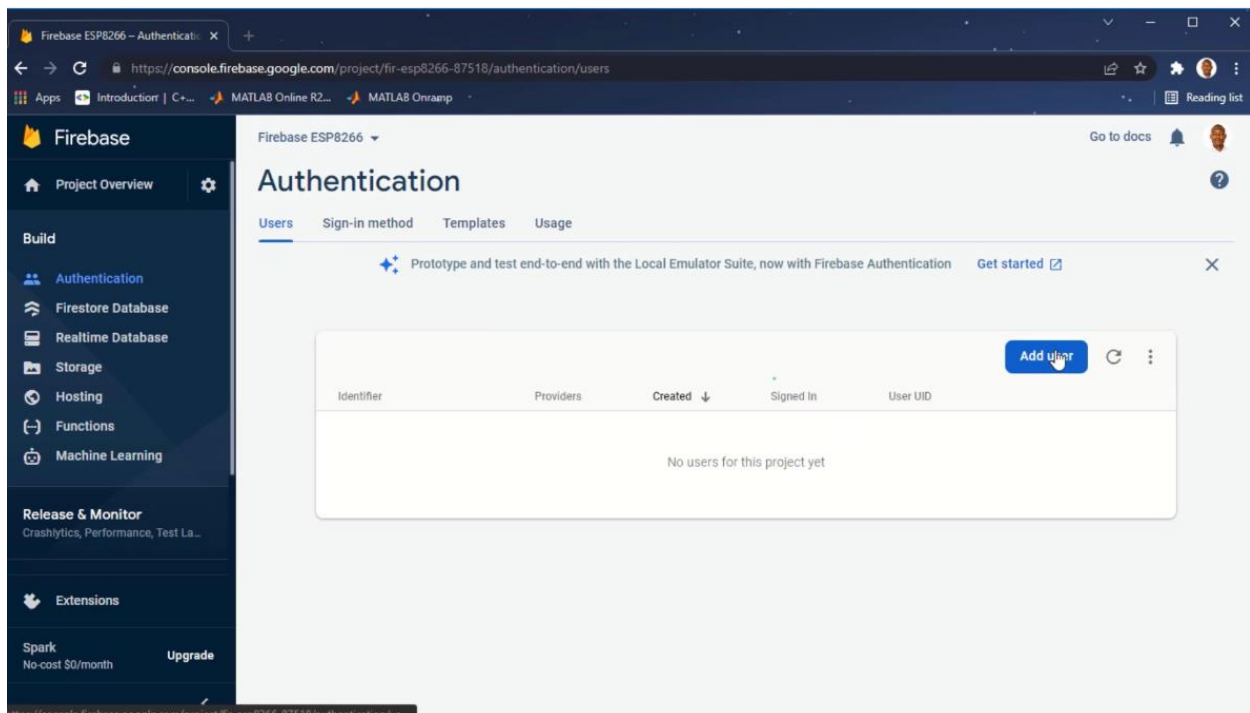


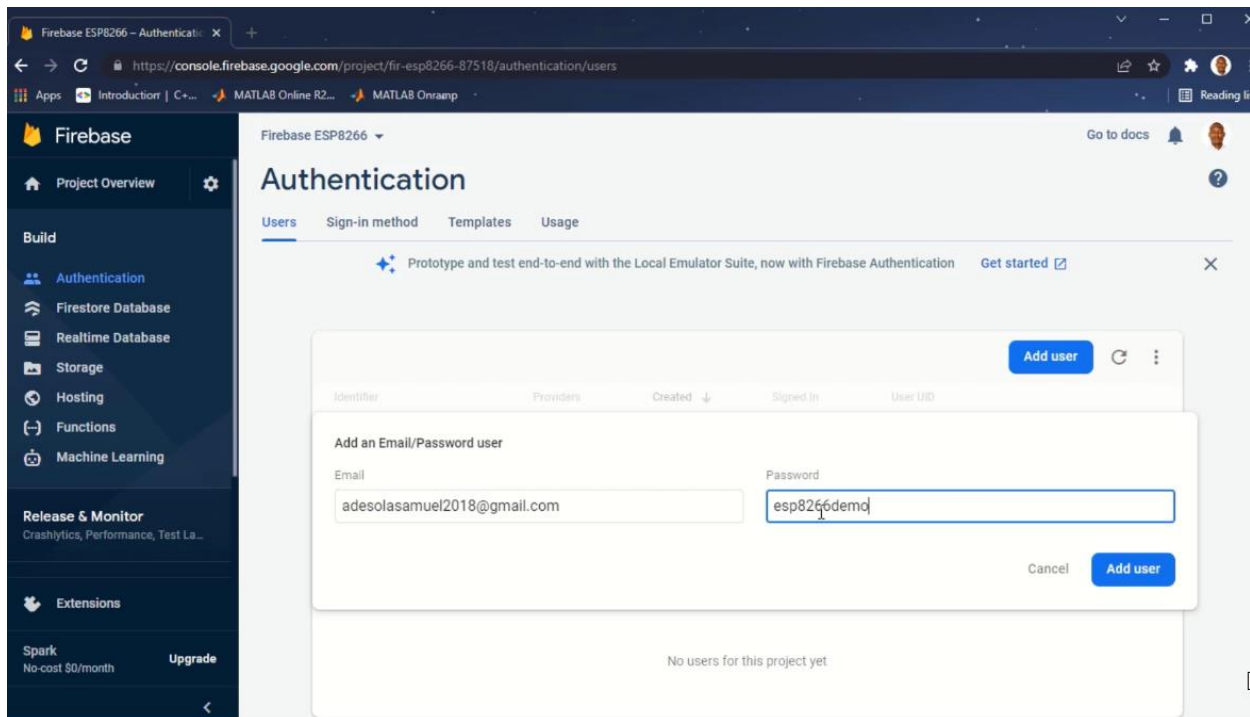
Click on **Email/Password** as sign in option and enable both **Email/Password** and **Email Link**, click **save** when you are done.



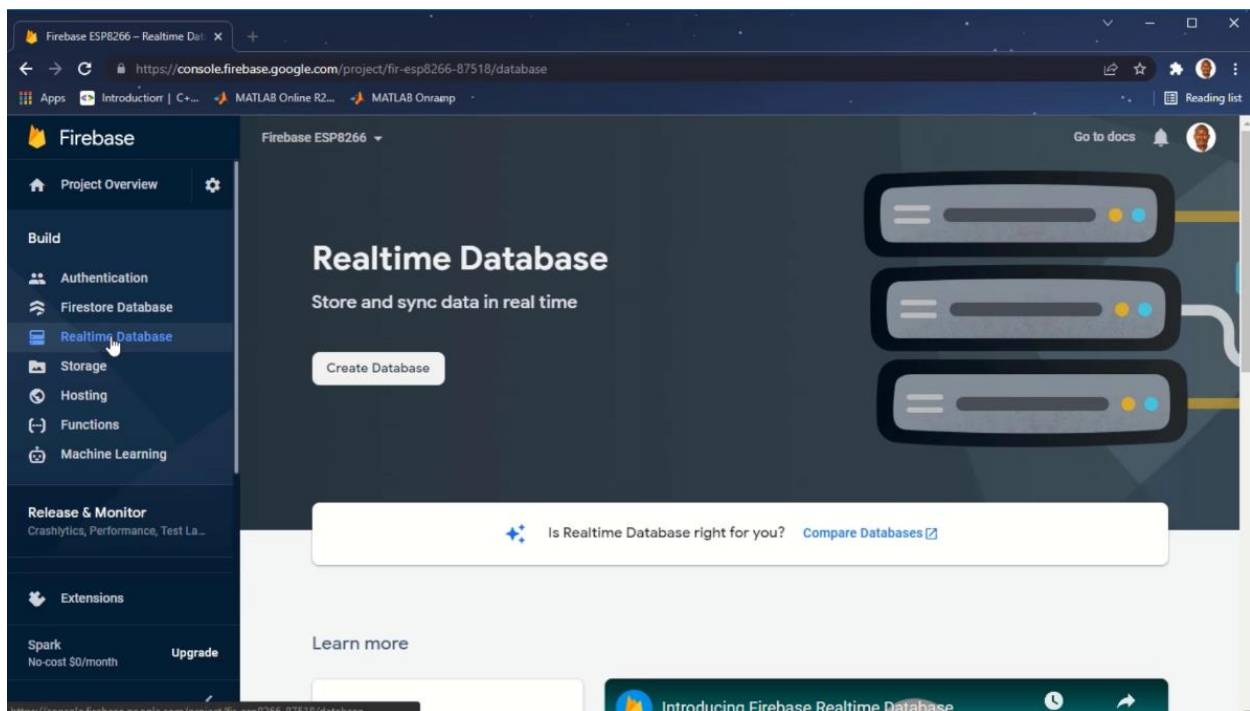


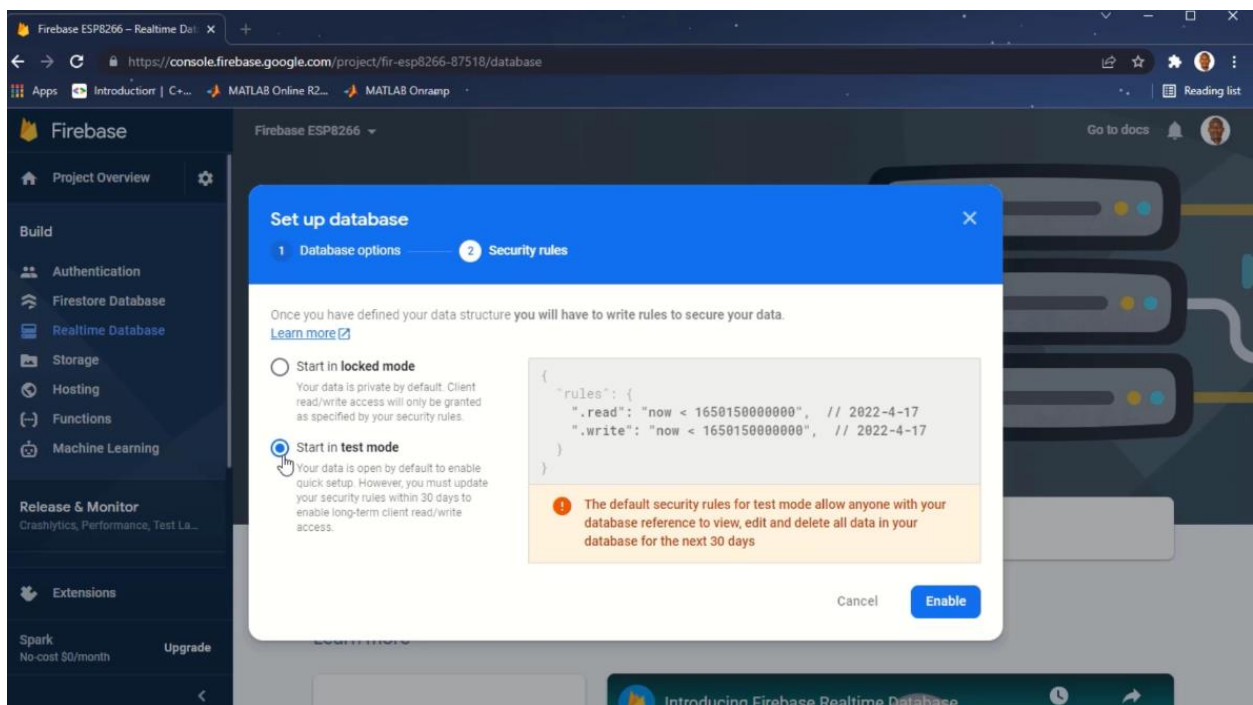
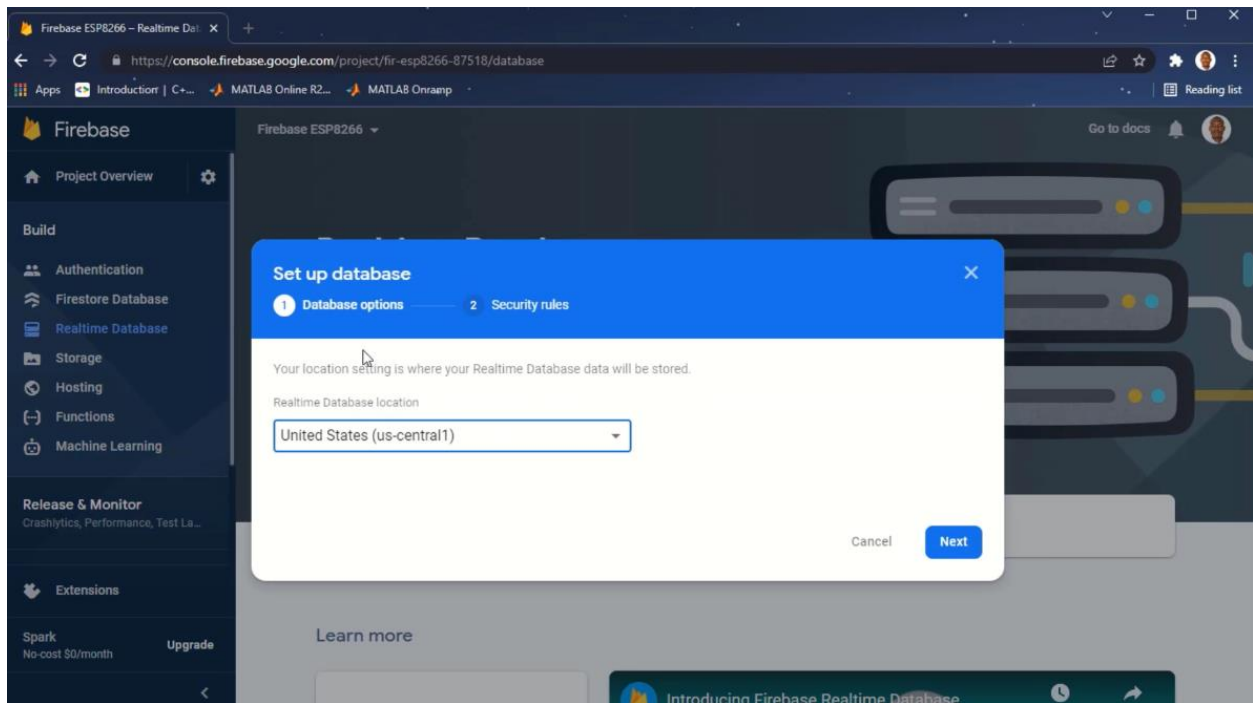
Step 6: Go back to Users and Click on **Add User**, enter both the user email and password you want the user to use in accessing the database, note these details somewhere as we will be using it in the code. With this, we are done with authentications.

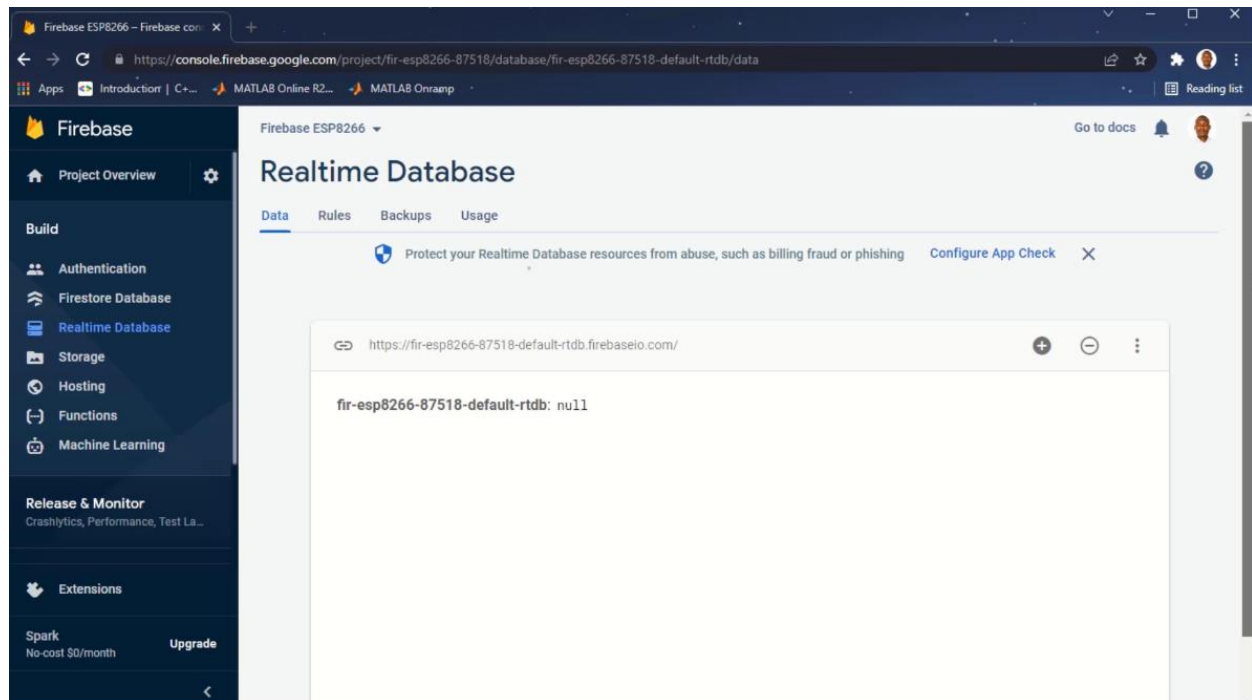




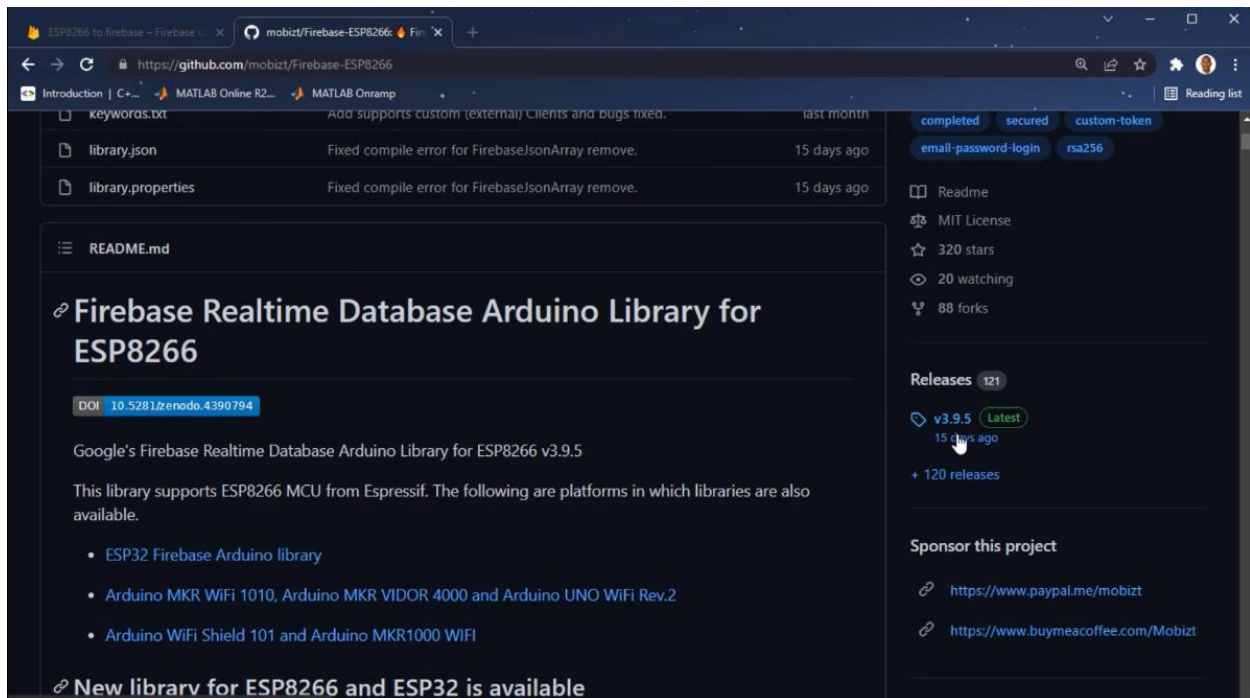
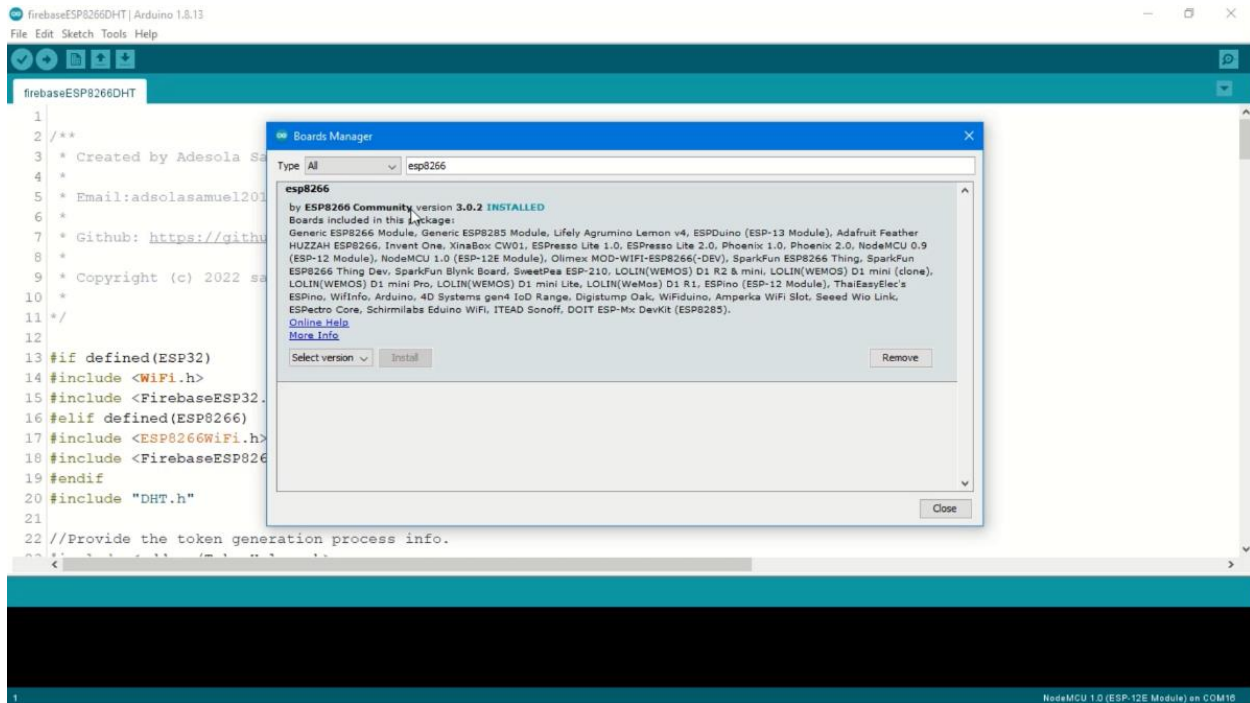
Step 7: From the side bar again, click on **Realtime Database** and click on **Create Database**. Select a region for your project, you will then be asked on which mode your project should start between **Locked Mode** and **Test Mode**. In test locked mode, both read and write access is set to false by default, you can change this later to true while in test mode, read and write access is set to true for some time, in this project, I will start with test mode. With that, our database has been set up.



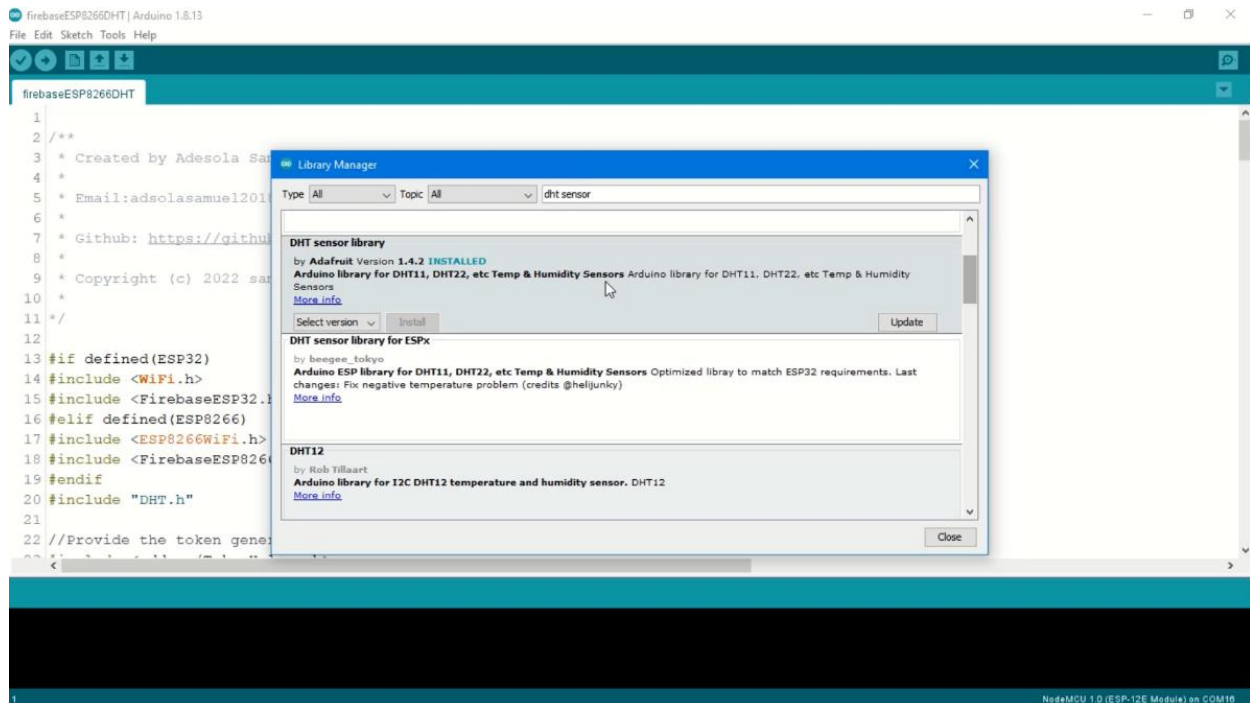




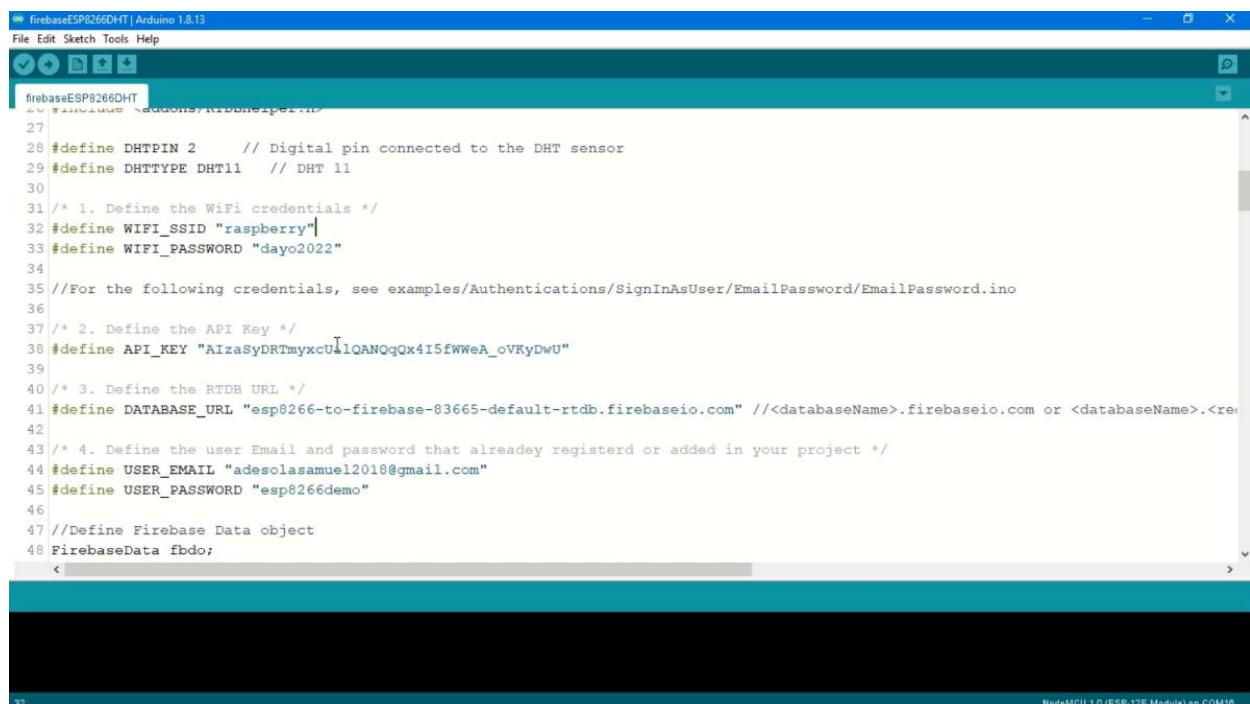
Step 8: Wow, that was quite a set up. Now is the coding time. The source code can be found at <https://github.com/adesolasamuel/ESP8266-NodeMCU-to-Google-Firebase>, due to lots of complications while trying to set up this tutorial, I will be stating the version of all tools I used in case you are finding it difficult, just stick to the version I am using and there should not be a problem. To start with, I was working on Arduino IDE 1.8.13. From board managers, download ESP8266 of which I am using version 3.0.2 for this tutorial, due to reasons I do not know, firebase library I downloaded from Library manager was not working so I had to download it from GitHub, <https://github.com/adesolasamuel/Firebase-ESP8266> which was version 3.9.5 at the time of writing this guide. Download the library ZIP file and add it to Arduino IDE through the library manager.

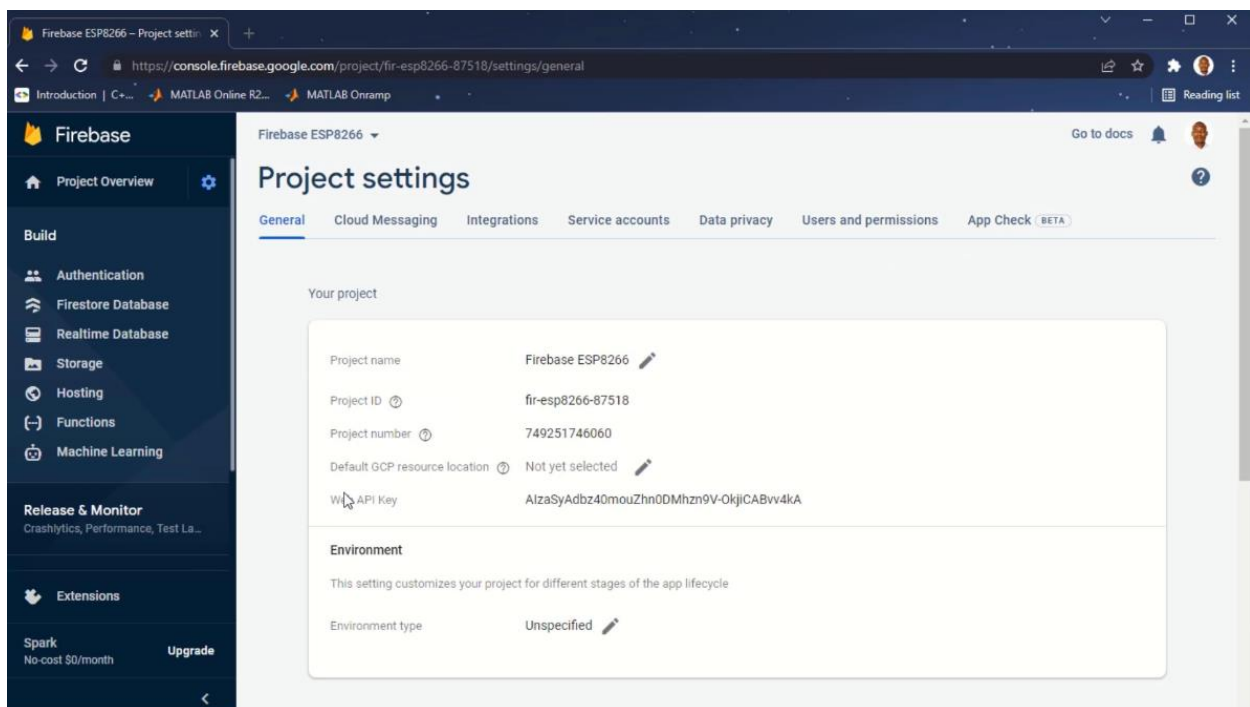
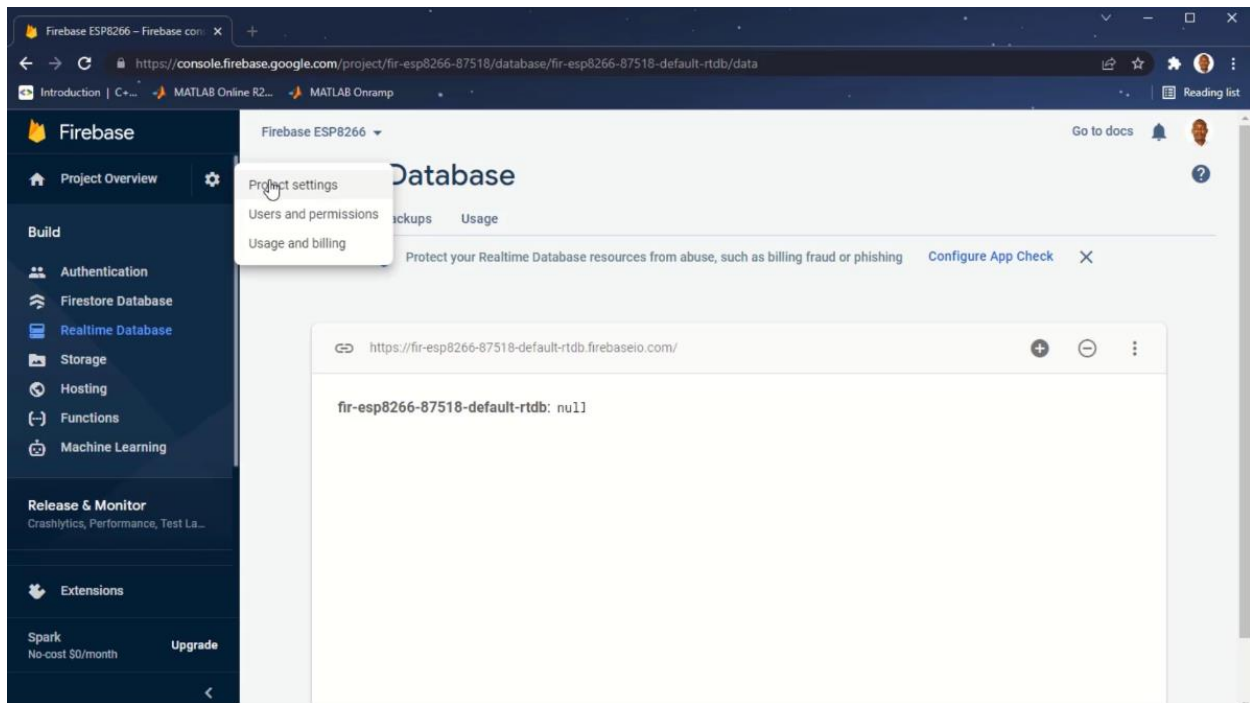


Also, since we are reading DHT11 data, download a Adafruit library for reading DHT Sensor data from Arduino IDE library manager. After installing all necessary dependencies, you can start editing the credentials to suite your board and network.

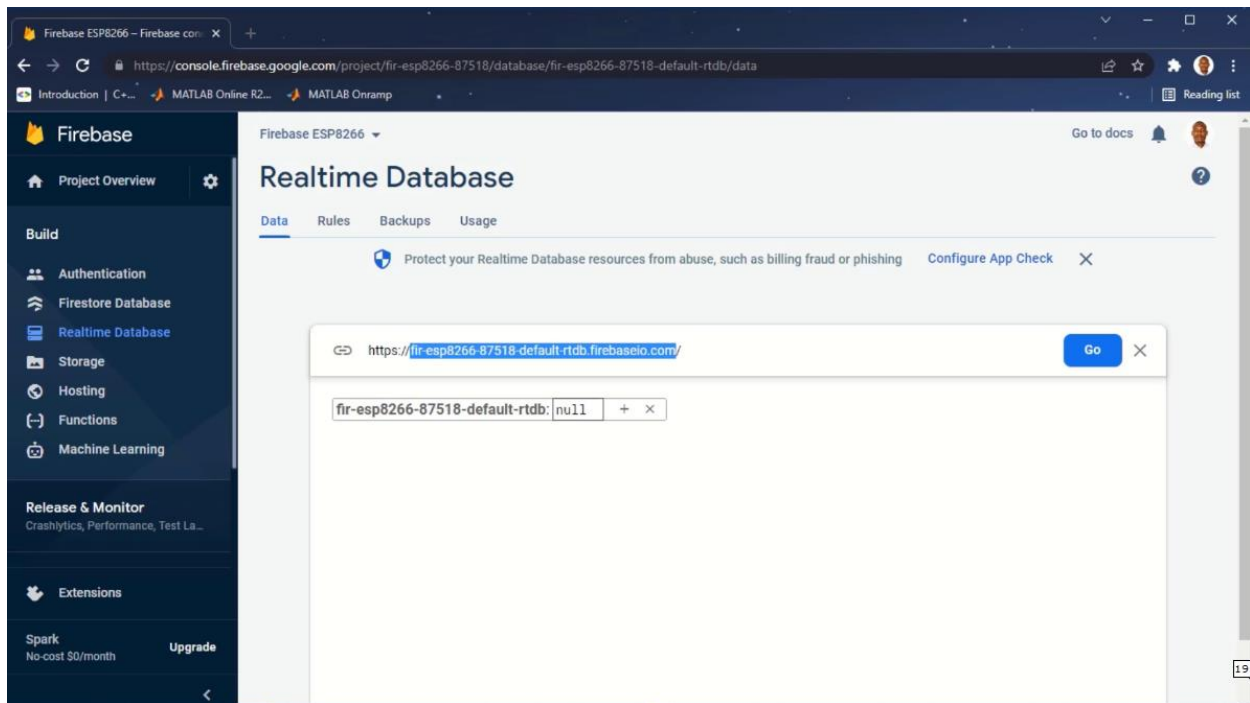


Step 9: Change the WIFI SSID and Password to your network's own. To get your API key, go back to firebase, click on the gear icon to go to project setting, on the general tab, you will see the web API displayed, copy it and paste it in the Arduino code.



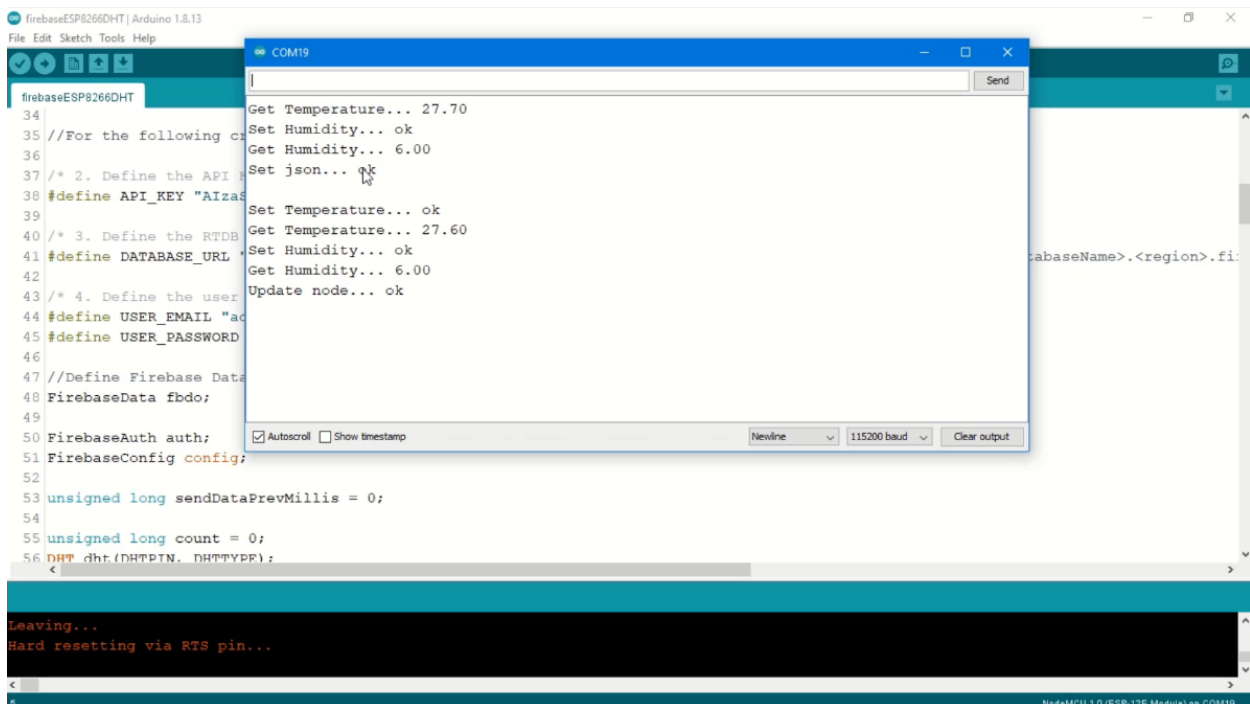


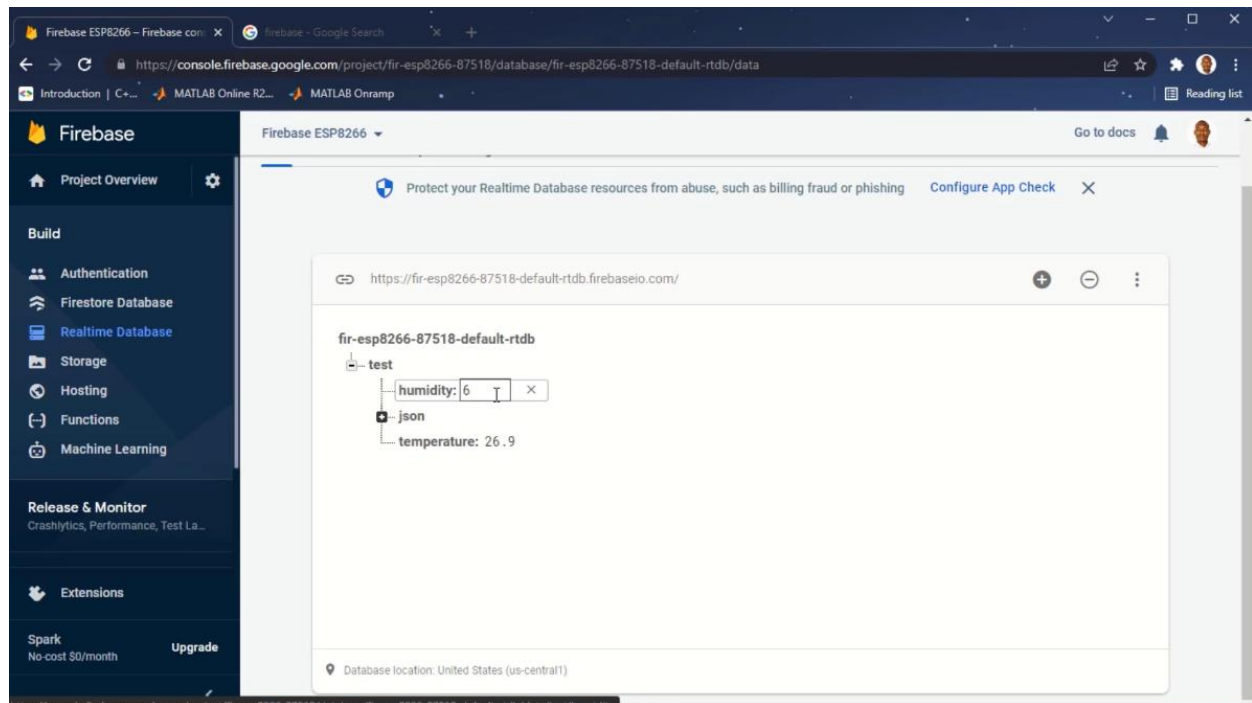
To get your database URL, go back to Realtime database and copy the URL displayed on top of the database without the https.



For user email and password, input the email and password that was defined during authentication.

Step 10: Study the code and note the line responsible for sending DHT data to cloud, verify the code and upload it, open the serial monitor and you will see data being sent, also, check your Firebase to read the data Realtime.





FOR VIDEO GUIDE, GO TO: <https://www.youtube.com/watch?v=7VxbVHssiQU>