School of Electronics

Internet of Things

By: Murtaza Ameen
School of Electronics, Devi Ahilya Vishwavidyalaya, Indore.
Contact no. 9669635246

Overview

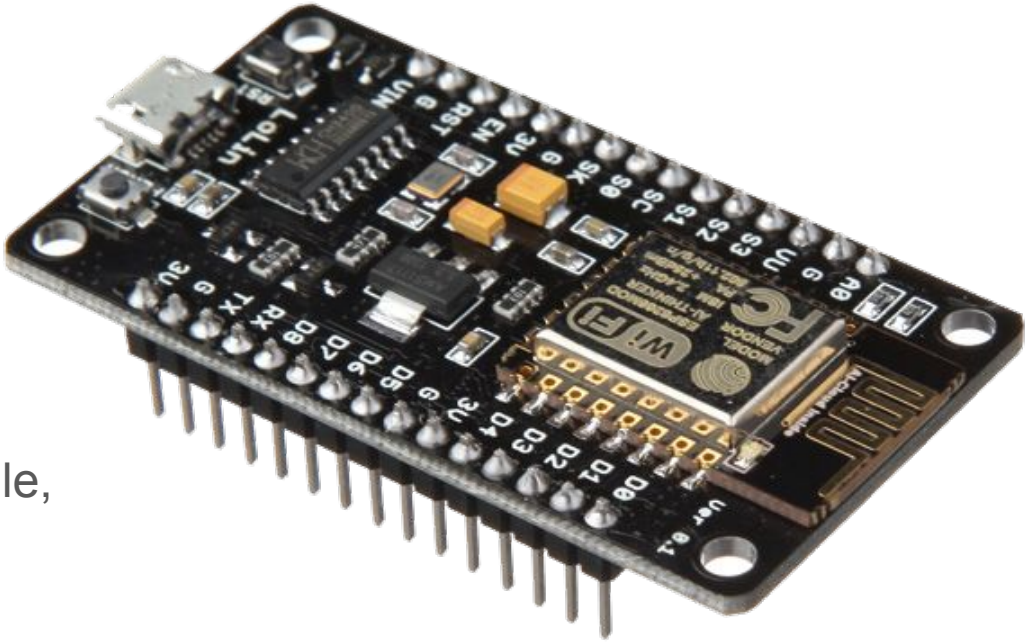
- Introduction to IoT.
- NodeMCU Devkit 1.0.
- NodeMCU Pin Configuration.
- Introduction to Firebase.
- Firebase Realtime Database.
- Firebase REST API.
- NodeMCU with Arduino IDE.
- Example 1: Temperature Monitoring using NodeMCU.
- Example 2: Uploading Temperature data over Google Firebase Server.
- Android App Development for monitoring uploaded values.
- Conclusion
- References

Brief Introduction to IoT

- Ecosystem of connected physical objects.
- Objects to be sensed or controlled remotely across existing network.
- Things with IP.
- Interaction with Environment.
- Examples: smart microwaves, self-driving cars, wearable fitness device etc.

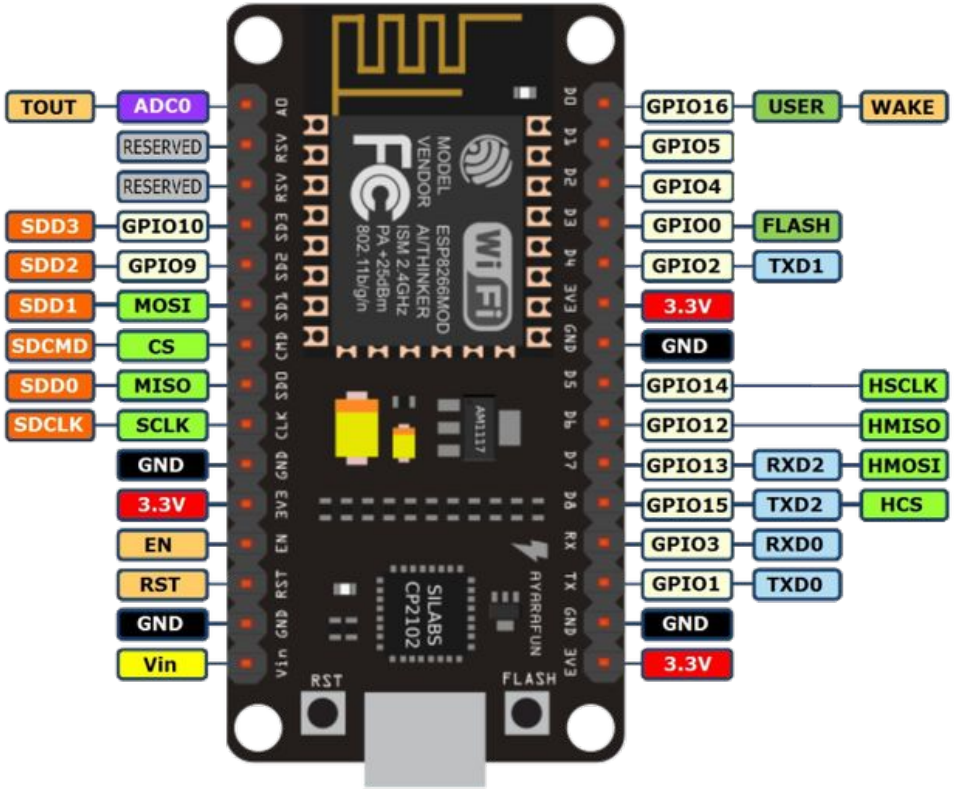
NodeMCU Devkit 1.0

- Memory - 128kBytes
- Storage - 4MBytes
- Power - USB
- Open Source
- Lua script
- OS - XTOS
- Microcontroller - ESP-12E module, with Espressif ESP8266 32bits
- Operating Voltage - 3.3v
- Clock Speed - 80MHz
- Connectivity - IEEE 802.11 b/g/n Wi-Fi



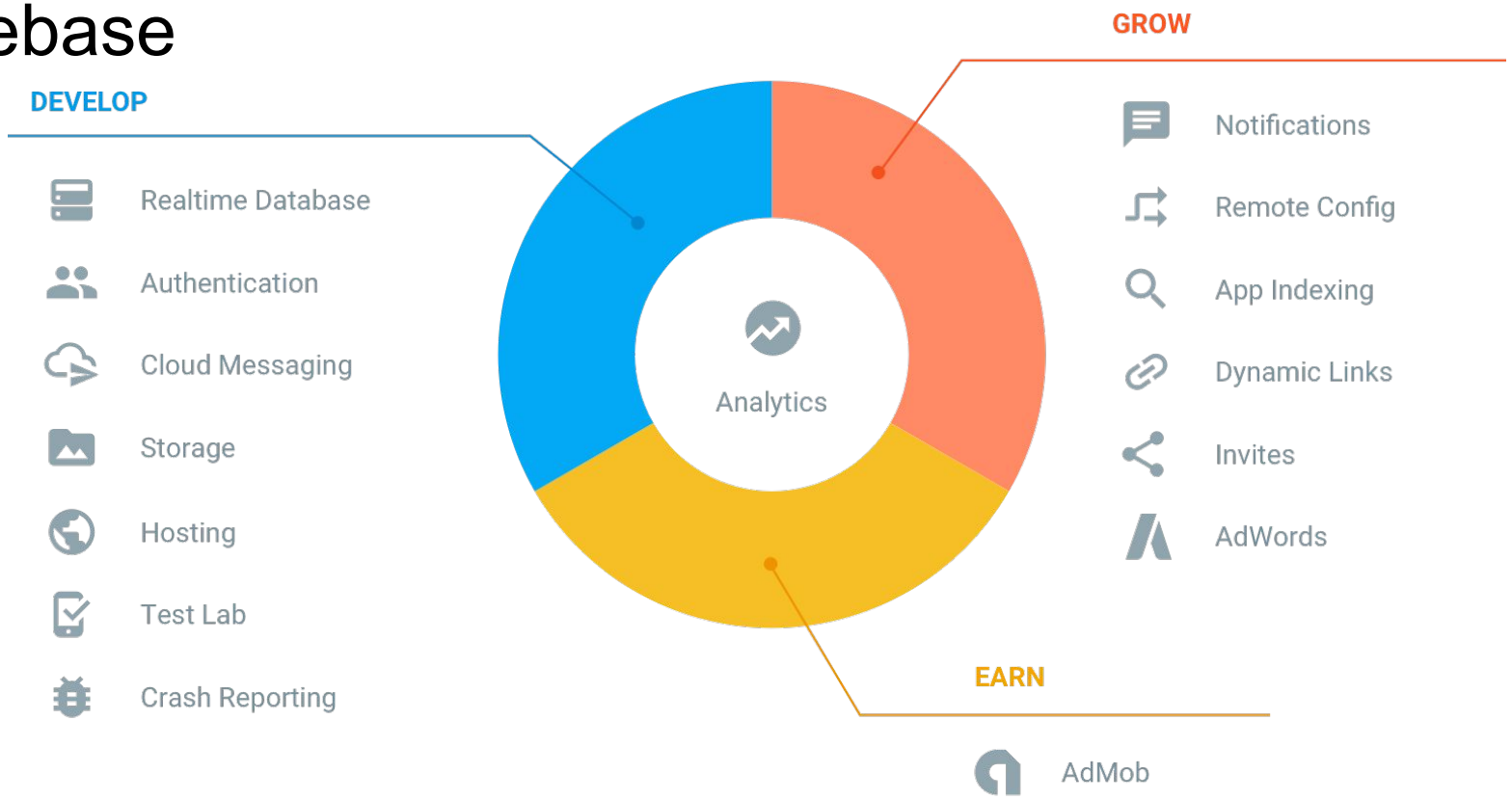
Source github ref-2

NodeMCU Pin Configuration



Source github ref - 2

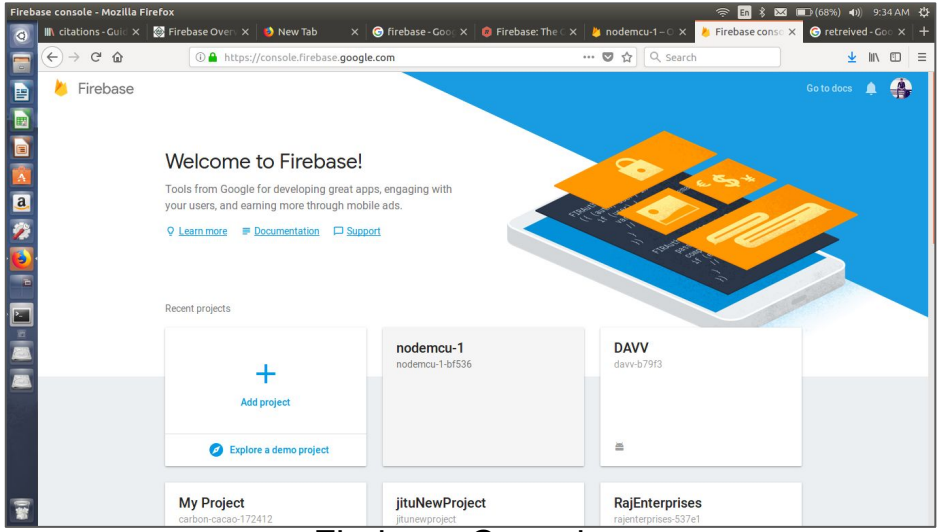
Firebase



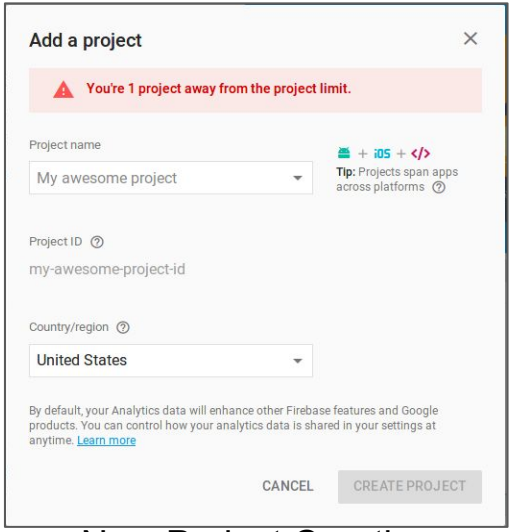
Source: ‘The Good, Bad, and the Ugly’, retrieved from https://www.raizlabs.com/dev/2016/12/firebase-case-study/

Project Creation on Firebase

1. Goto website: https://console.firebase.google.com/
2. Create project by clicking on ‘Add Project’ button, provide the project name and country name in popup window, and click on create project.

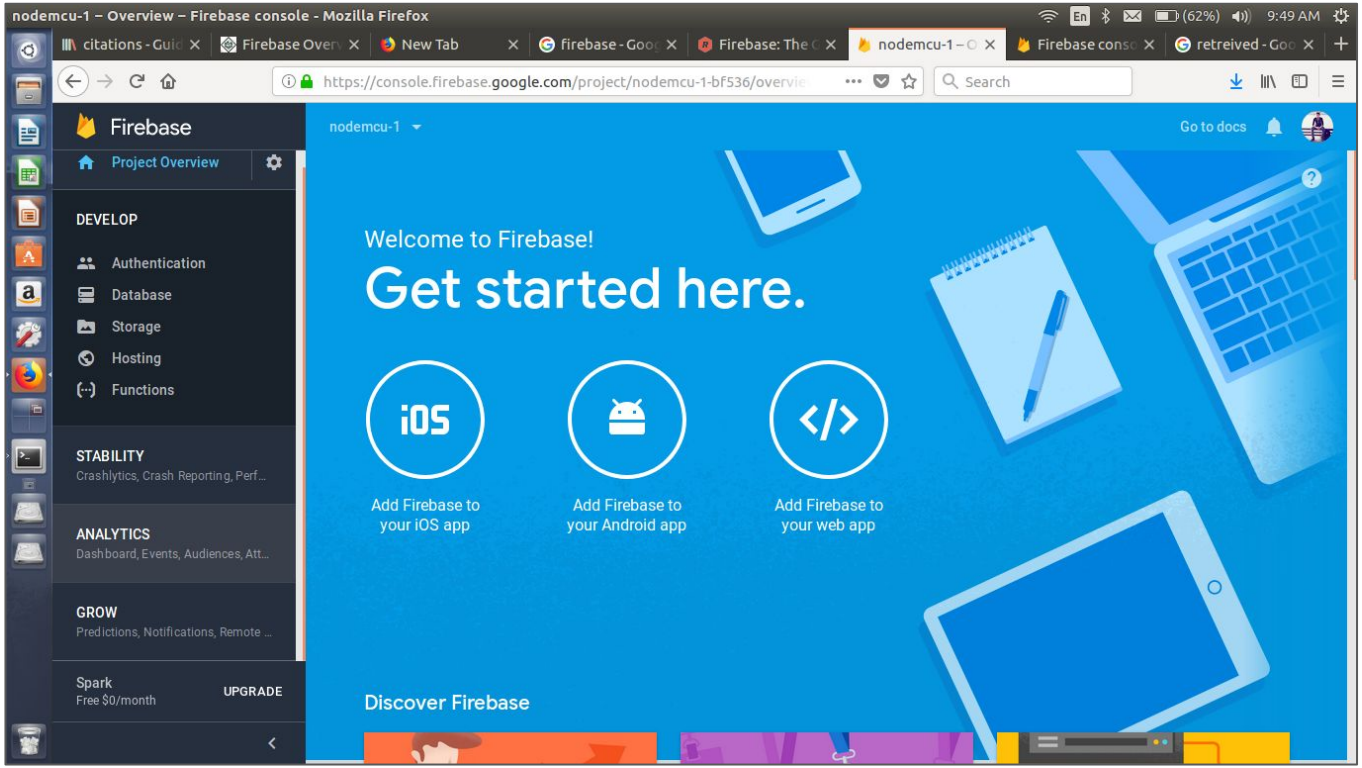


Firebase Console



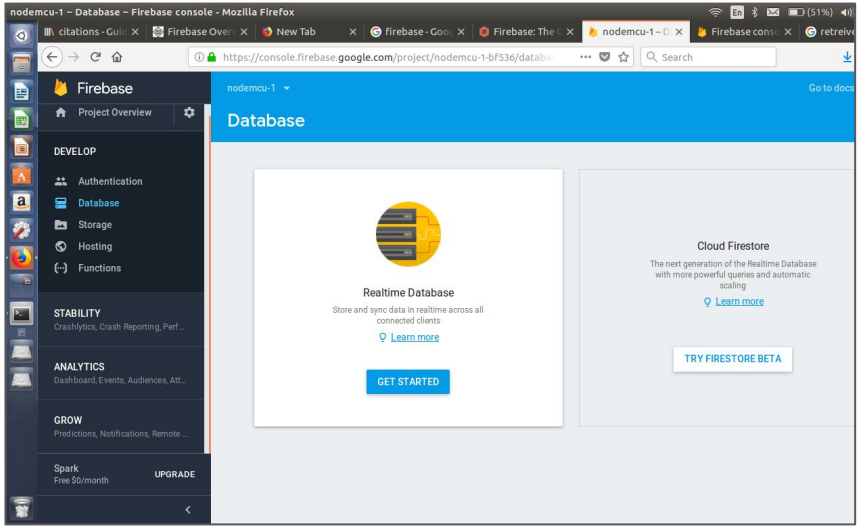
New Project Creation

Project Console Description

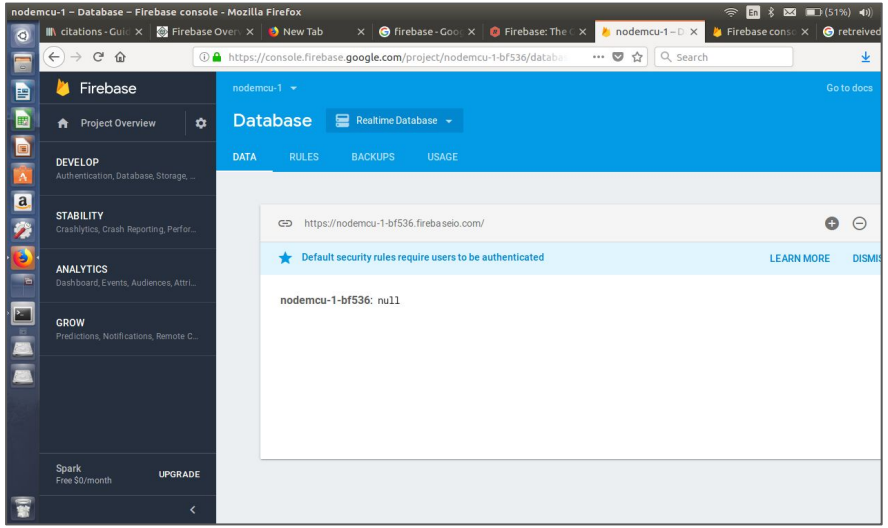


Project Console

Realtime Database Panel

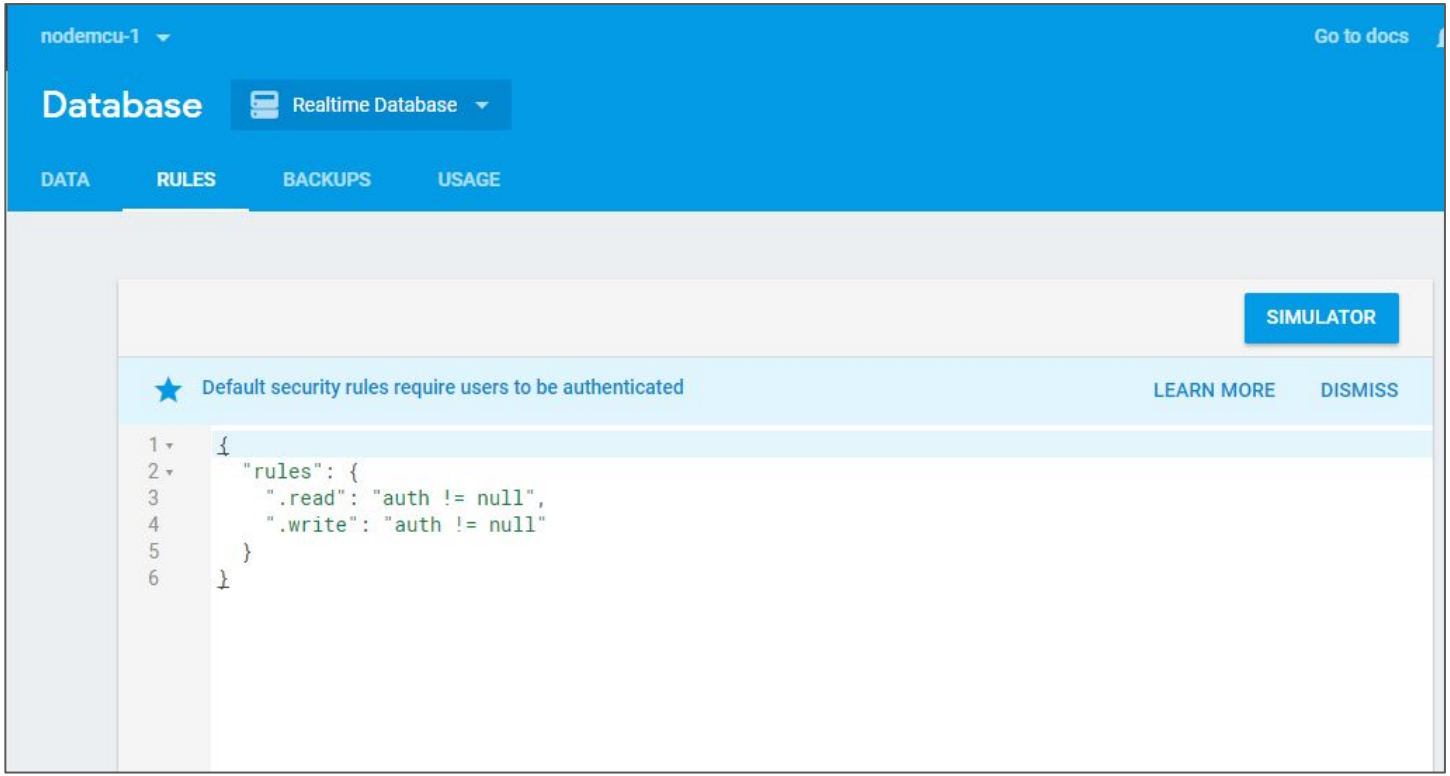


Realtime Database and Firestore



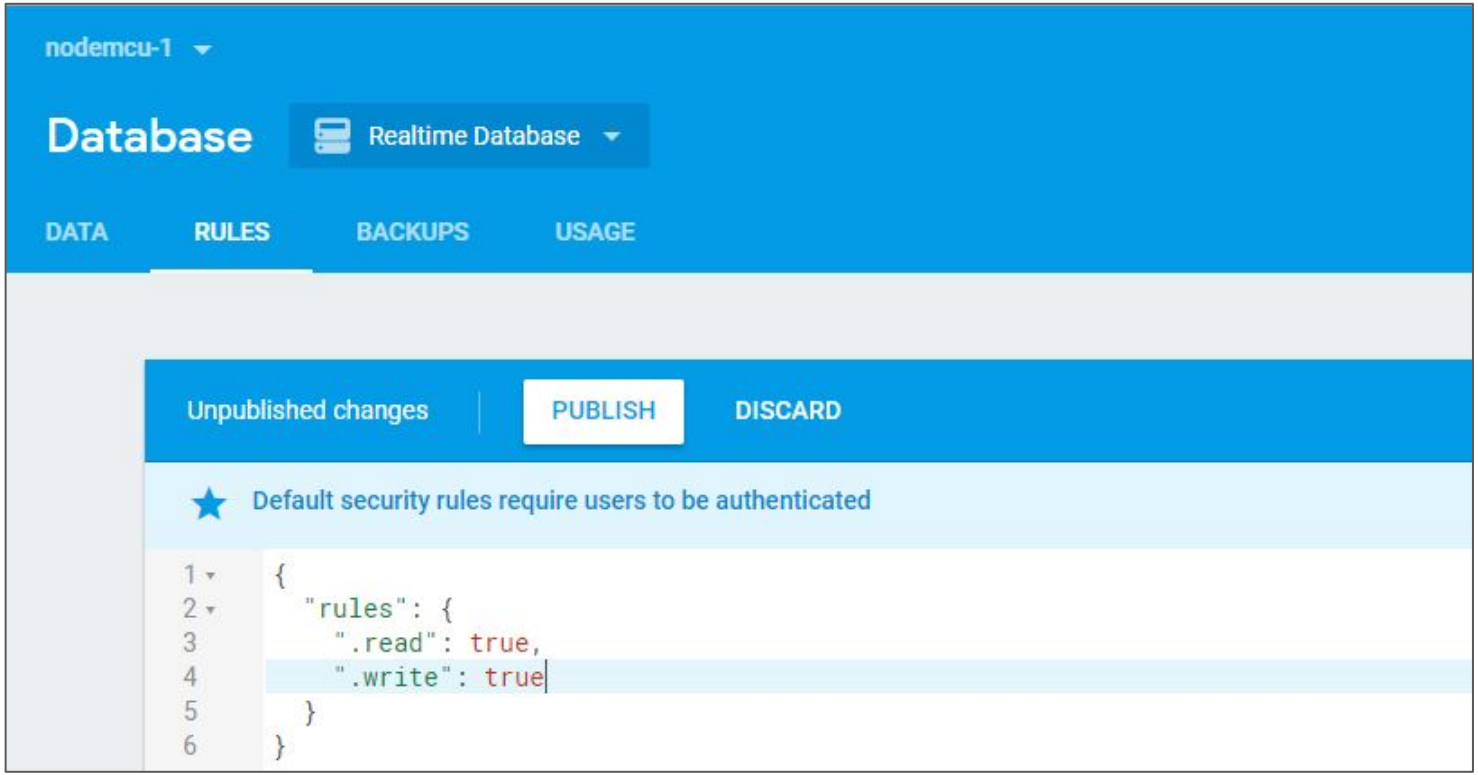
Realtime Database

Rules



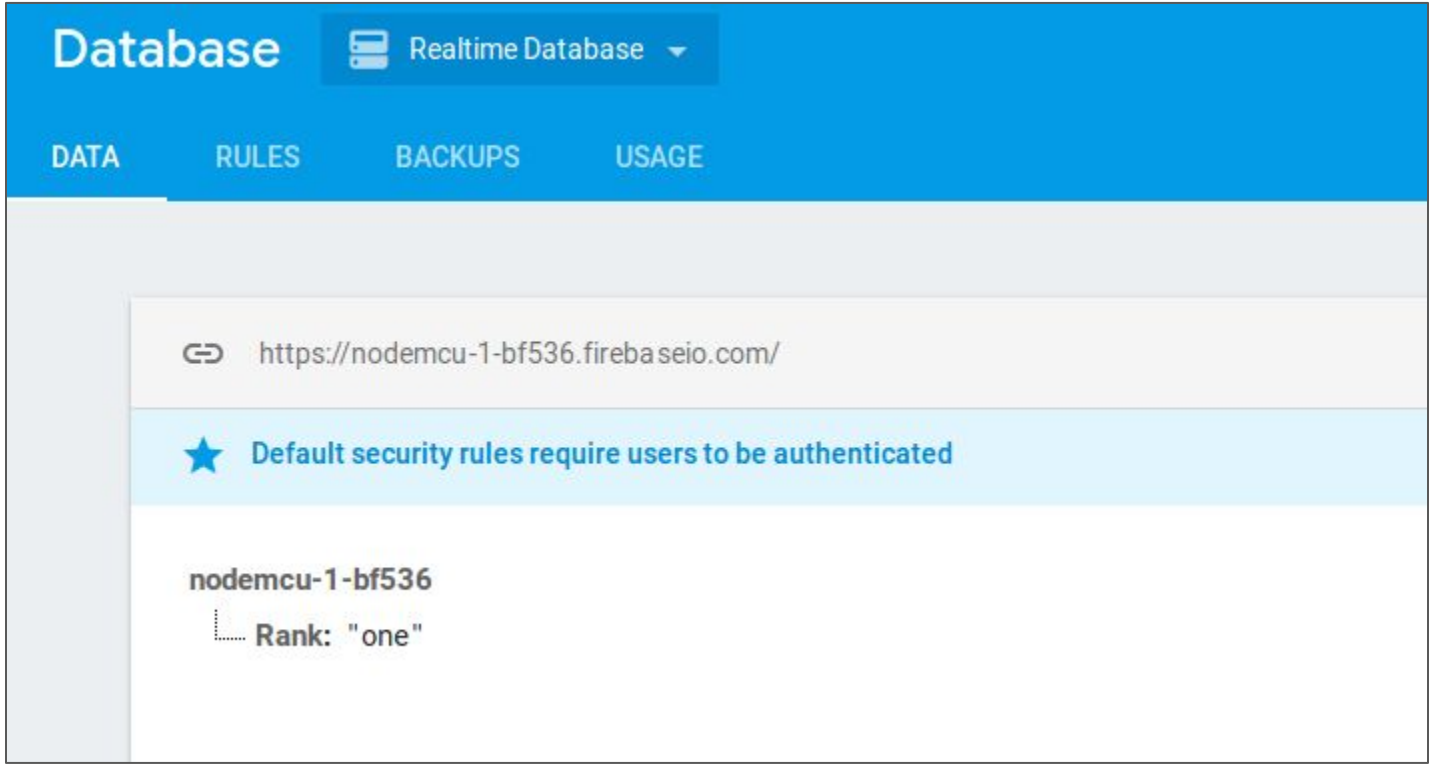
Realtime Database Rules

Rules



Realtime Database Rules

Key and Value



Key value in Realtime Database

Firebase Database REST API

- A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.

1. **GET - Reading Data:** Data from your Realtime Database can be read by issuing an HTTP GET request to an endpoint.

   Request:: curl https://nodemcu-1-bf536.firebaseio.com/.json

   Response: {"Rank":"one"}

Firebase Database REST API

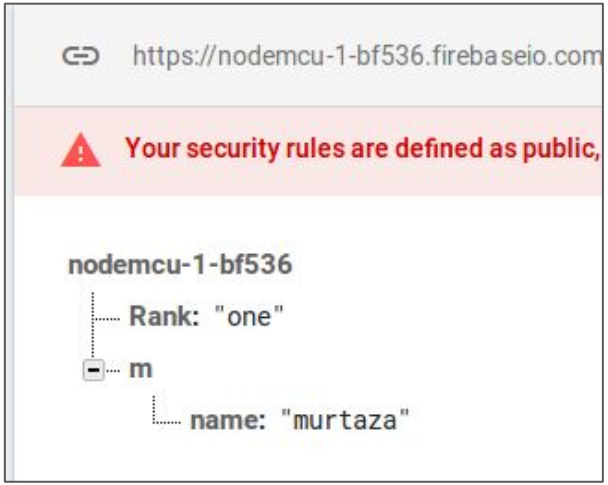
2. **PUT - Writing Data:** You can write data with a PUT request.

Request:

curl -X PUT -d '{"name":"murtaza"}' 'https://nodemcu-1-bf536.firebaseio.com/m.json'

Response:

{"name":"murtaza"}



Screenshot: 1

Firebase Database REST API

3. **POST - Pushing Data:** To accomplish the equivalent of the push() method in android, you can issue a POST request.

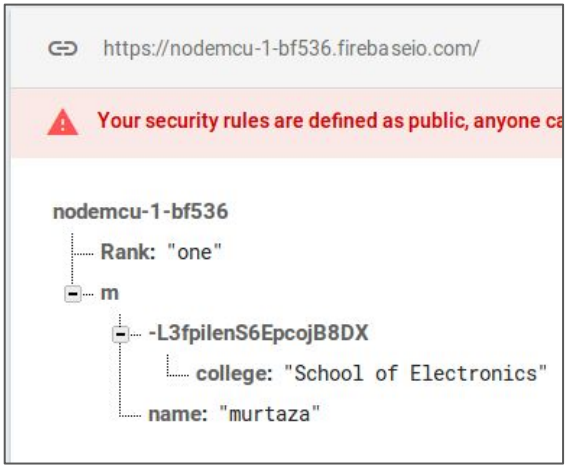
   Request:

   curl -X POST -d '{"college":"School of Electronics"}' \

   'https://nodemcu-1-bf536.firebaseio.com/m.json'

   Response:

   {"name":"-L3fpiIenS6EpcojB8DX"}



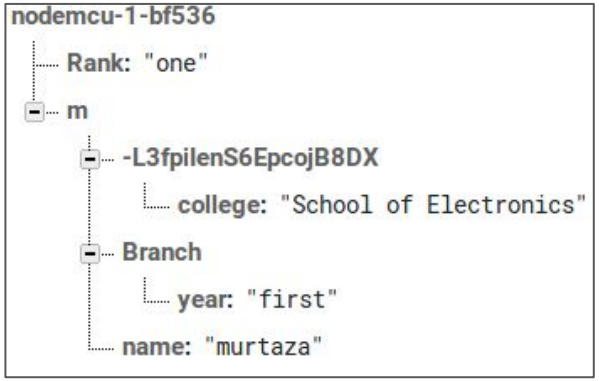
Screenshot: 2

Firebase Database REST API

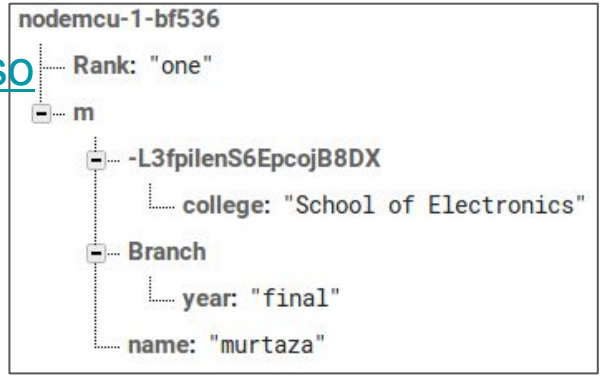
3. **PATCH - Updating Data:** You can update specific children at a location without overwriting existing data using a PATCH request.

Request:
curl -X PATCH -d '{"year":"final"}' 'https://nodemcu-1-bf536.firebaseio.com/m/Branch/.json'

Response:
{"year":"final"}



Screenshot: 3



Screenshot: 4

Firebase Database REST API

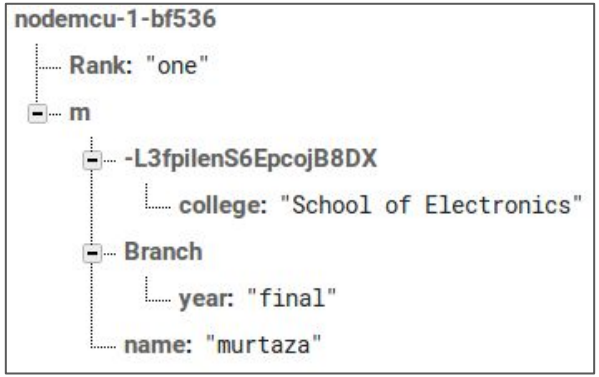
5. **DELETE - Removing Data :**You can delete data with a DELETE request.

Request:

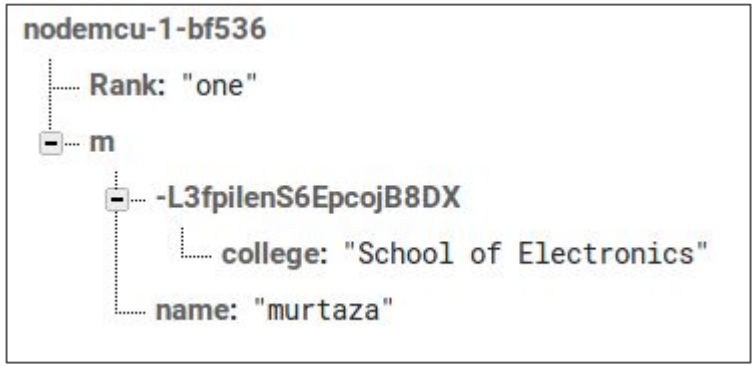
curl -X DELETE 'https://nodemcu-1-bf536.firebaseio.com/m/Branch/year.json'

Response:

null



Screenshot: 5



Screenshot: 6

Arduino IDE Installation

Step-1: Goto https://www.arduino.cc/en/Main/Software

Step-2: Download Arduino IDE from the web page, and Extract the file.

Step-3: Copy the file in the /opt directory:
sudo mv /home/murtaza/Downloads/arduino-1.8.5 /opt/

Step-4: Change directory: cd /opt/arduino-1.8.5/ and run installation file by running the command sudo sh install.sh

Step-5: Open IDE and check the serial port to provide write permission over the port by running command sudo chmod a+rw /dev/ttyACM0

Add ESP8266 core Library

Step-1: Open the Arduino IDE and press ctrl + comma.

Step-2: Enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into *Additional Board Manager URLs* field.

Step-3: Goto Tools → Boards → Boards Manager…

Step-4: Type ‘esp’ in search field and select ‘esp8266 by ESP8266 community’ and install it.

Step-5: Now check in Tools → Boards, there is a list of esp based boards, select NodeMCU 1.0 (ESP-12E module)

Add Firebase/arduino Library

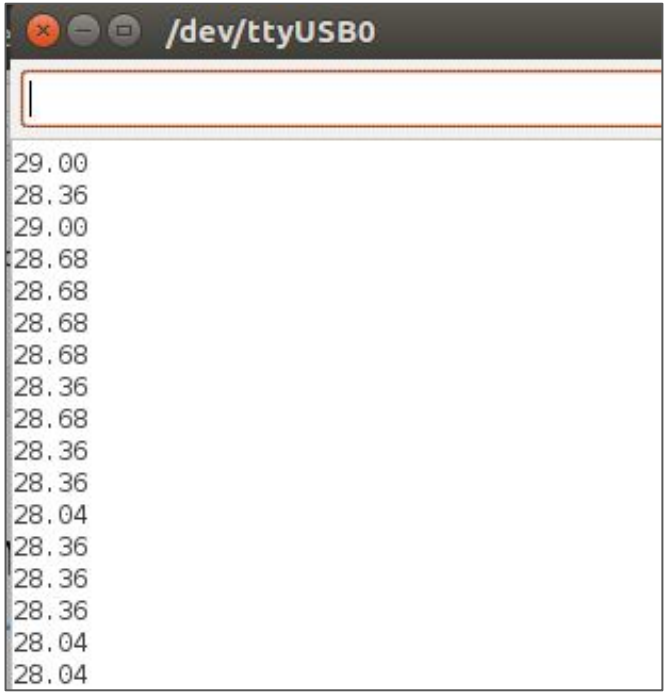
Step-1: Goto https://github.com/firebase/firebase-arduino , and download the ZIP file by clicking on ‘Clone or download’.

Step-2: Open Arduino IDE, goto Sketch → Add Library → .ZIP library, then provide the path of .zip file of library and click on okay.

Ex: Temperature measurement

```
int sensorPin = A0;

void setup() {
  Serial.begin(9600);
}

void loop() {

  sensorValue = analogRead(sensorPin);

  float millivolts = (sensorValue/1024.0) * 3300;
  //3300 is the voltage provided by NodeMCU
  float celsius = millivolts/10;

  Serial.println(celsius);
  delay(1000);

}
```

Code for temperature monitoring



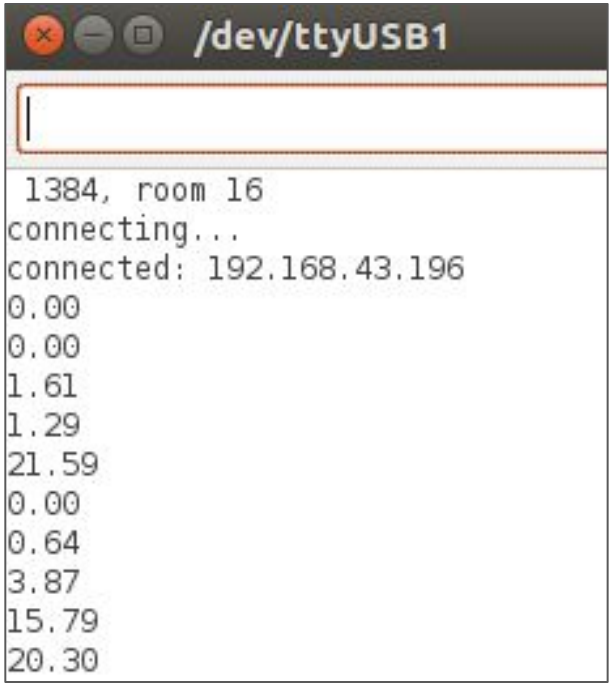
Output

Ex: Uploading Temperature readings over Server

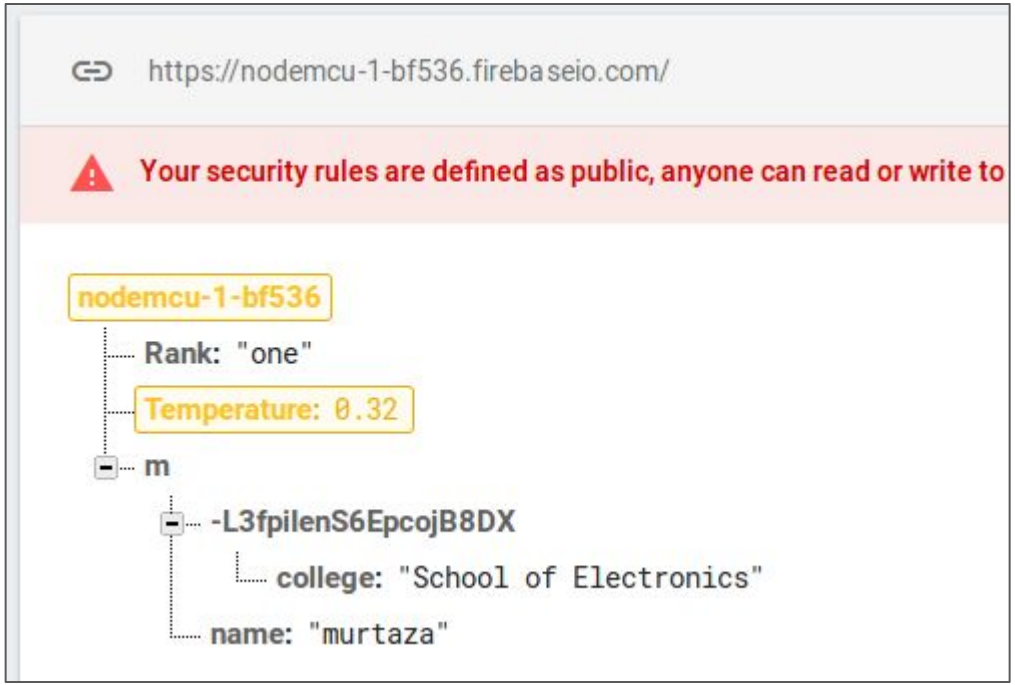
```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

#define FIREBASE_HOST "nodemcu-1-bf536.firebaseio.com"
#define WIFI_SSID "RobuByte"
#define WIFI_PASSWORD "esp@8265"

int sensorPin = A0;

void setup() {

  Serial.begin(9600);

  // connect to wifi.
  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);
}
```

```
void loop() {
float sensorValue = analogRead(sensorPin);

  float millivolts = (sensorValue/1024.0) * 3300;
  //3300 is the voltage provided by NodeMCU
  float celsius = millivolts/10;
  Firebase.setFloat("Temperature", celsius);
  Serial.println(celsius);
  delay(1000);
}
```

Output:



Serial Terminal



Firebase Database

Android App Development

Step-1: Open android IDE, and click on ‘Start a new Android Studio project’.

Step-2: Provide a name to your application.

Step-3: Select the lowest API level for your app to run.

Step-4: Select empty Activity.

Step-5: Wait until your gradle is building.

Android App Development

TempMonitor - [~/AndroidStudioProjects/TempMonitor] - [TempMonitor] - ~/AndroidStudioProjects/TempMonitor/app/src/main/java/com/exa

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

TempMonitor > app > src > main > java > com > example > root > tempmonitor

app

Android

app

Gradle Scripts

activity_main.xml | MainActivity.java

```java
package com.example.root.tempmonitor;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Terminal | 0: Messages | 6: Logcat | TODO | Event Log | Gradle Console

Gradle build finished in 9s 181ms (moments ago)

1:1 LF: UTF-8 : Context: <no context>

Add Firebase to Your Application

Step-1: Goto tools and select Firebase.

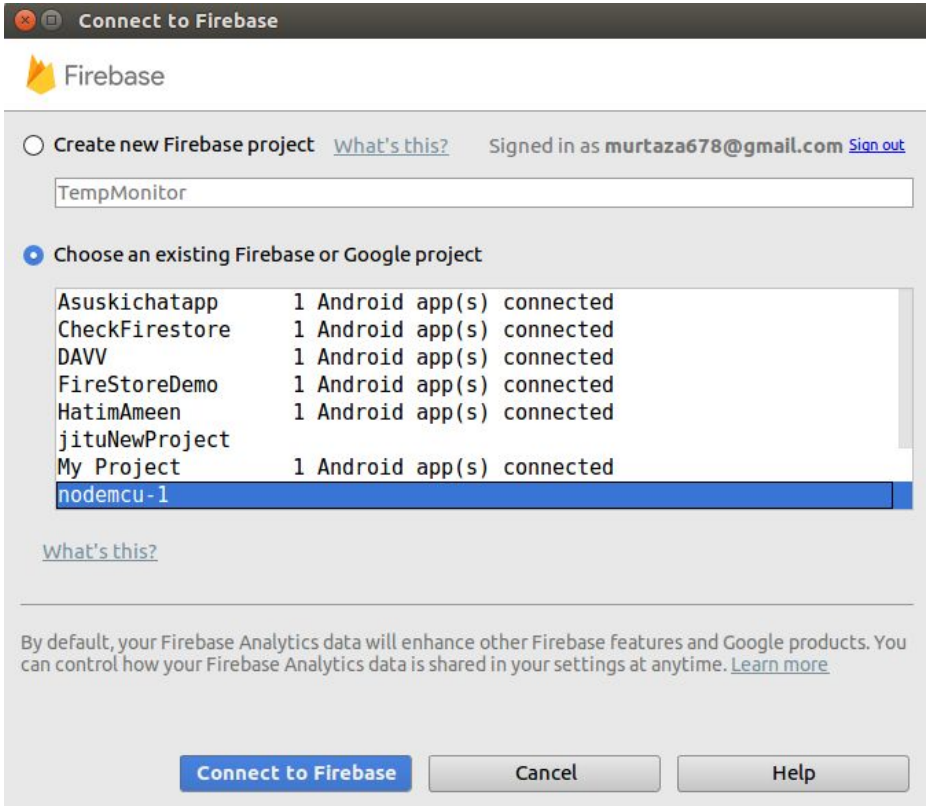
Step-2: Select ‘Realtime Database’ and then click on ‘save and retrieve data’.

Step-3: Click on ‘connect to Firebase’ and provide login details and permissions.

Step-4: Select the existing project from list or create new one as per need.

Step-5: Click on ‘Add the Realtime Database on your app’ and accept changes.

Add Firebase to Your Application



Add Firebase to Your Application

Add the Realtime Database to your app

Performing this action will make the following changes to your project.

```
build.gradle (project-level)

    Add Firebase Gradle buildscript dependency
        classpath 'com.google.gms:google-services:3.1.0'

app/build.gradle

    Add Firebase plugin for Gradle
        apply plugin: 'com.google.gms.google-services'

    build.gradle will include these new dependencies:
        compile 'com.google.firebase:firebase-database:11.0.4'
```

This will also enable the firebase-core library which includes Firebase Analytics. Learn more

Accept Changes | Cancel | Help

Add Firebase Event Listener

```
package com.example.root.tempmonitor;

import ...

public class MainActivity extends AppCompatActivity {
    TextView tv;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tv= findViewById(R.id.textView);

        FirebaseDatabase.getInstance().getReference( s: "Temp").addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
            float temp = dataSnapshot.getValue(Float.class);
            tv.setText(""+temp);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });
    }
}
```

Add more Features

Java File

```
public class MainActivity extends AppCompatActivity {
  TextView tv,data;
  Button send;
  EditText text;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    tv= findViewById(R.id.textView);
    send=findViewById(R.id.button2);
    data=findViewById(R.id.textView2);
    text=findViewById(R.id.editText);
```

```
send.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
          String data = text.getText().toString();

FirebaseDatabase.getInstance().getReference("data").setValue(data);
        }
      });


FirebaseDatabase.getInstance().getReference("data").addValueEvent
Listener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
        String temp = dataSnapshot.getValue(String.class);
        data.setText(temp);
        }
        @Override
        public void onCancelled(DatabaseError databaseError) {
        }
      });
  }
}
```

Conclusion

- The Internet of Things is closer to being implemented than the average person would think. Most of the necessary technological advances needed for it have already been made, and some manufacturers and agencies have already begun implementing a small-scale version of it.

References

- Getting to know NodeMCU and its DEVKIT board
- NodeMCU
- Firebase Realtime Database REST API
- ESP8266 core for Arduino IDE
- Firebase samples for Arduino
- Arduino Examples

Thank You