**COMSATS University**

**Islamabad**



**Lab Report # 13-14**

**Real Time Embedded Systems**

**(EEE-446)**

|  |
| --- |
| Control Home Appliances using MIT App and Google Firebase Real-time Database |

**Submitted By:**

**Arwa Aamir (FA16-EEE-002)**

**Submitted To:**

**Dr. Ahsen Malik**

# Lab # 13 - 14

# Control Home Appliances using MIT App and Google Firebase Real-time Database

### Objectives

* Connect DH11 Humidity/Temperature Sensor to Local Server
* Use Smart Config to configure anonymous ESP32 with local Wifi
* Real time database creation using Google Firebase
* Connectivity of database with Android App and ESP32 Module
* Control of home appliances using ESP32

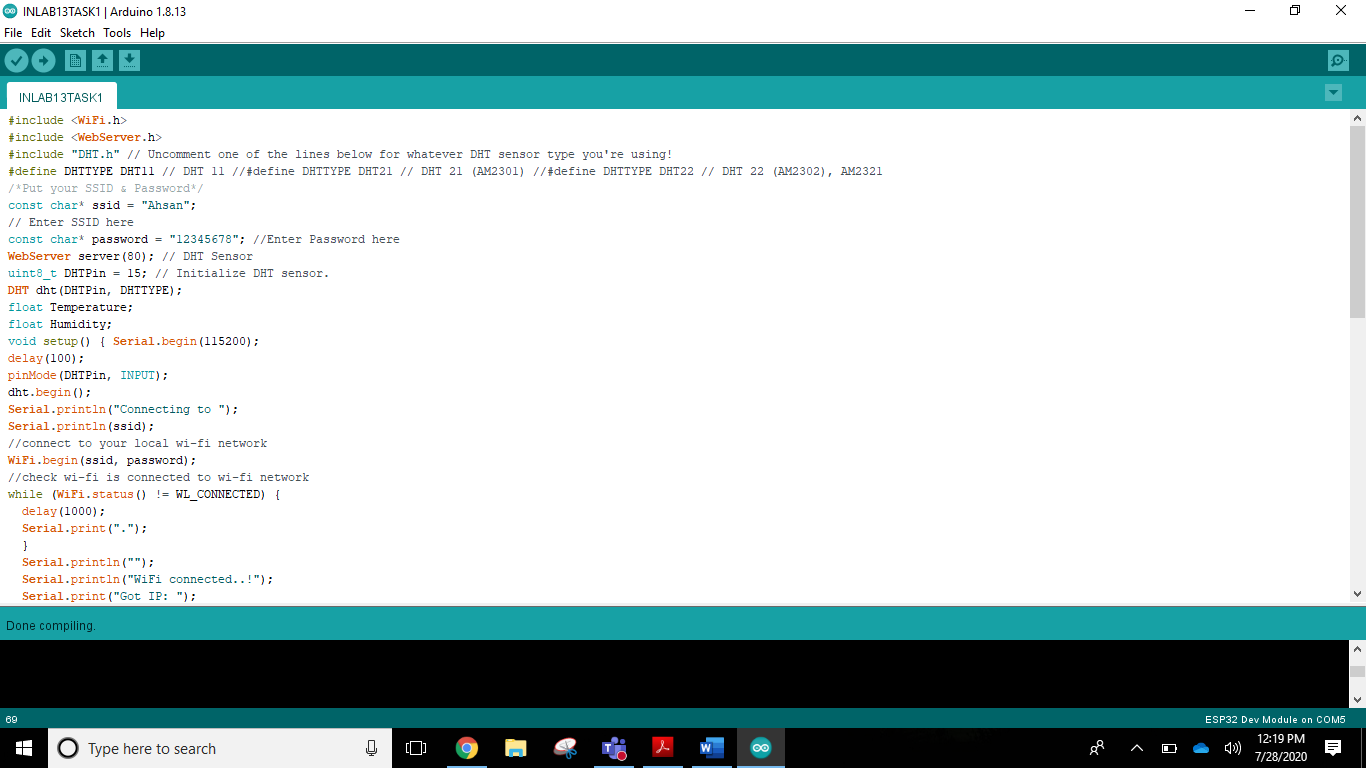
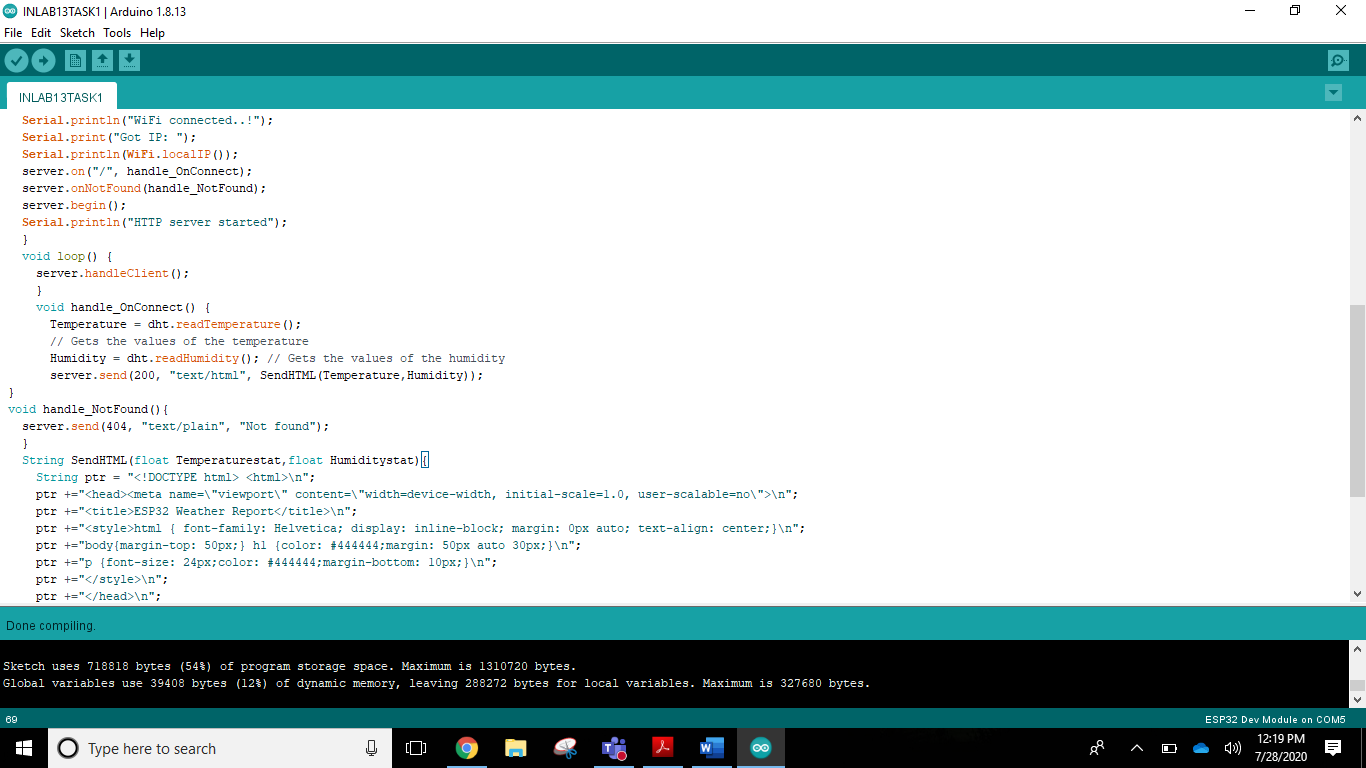
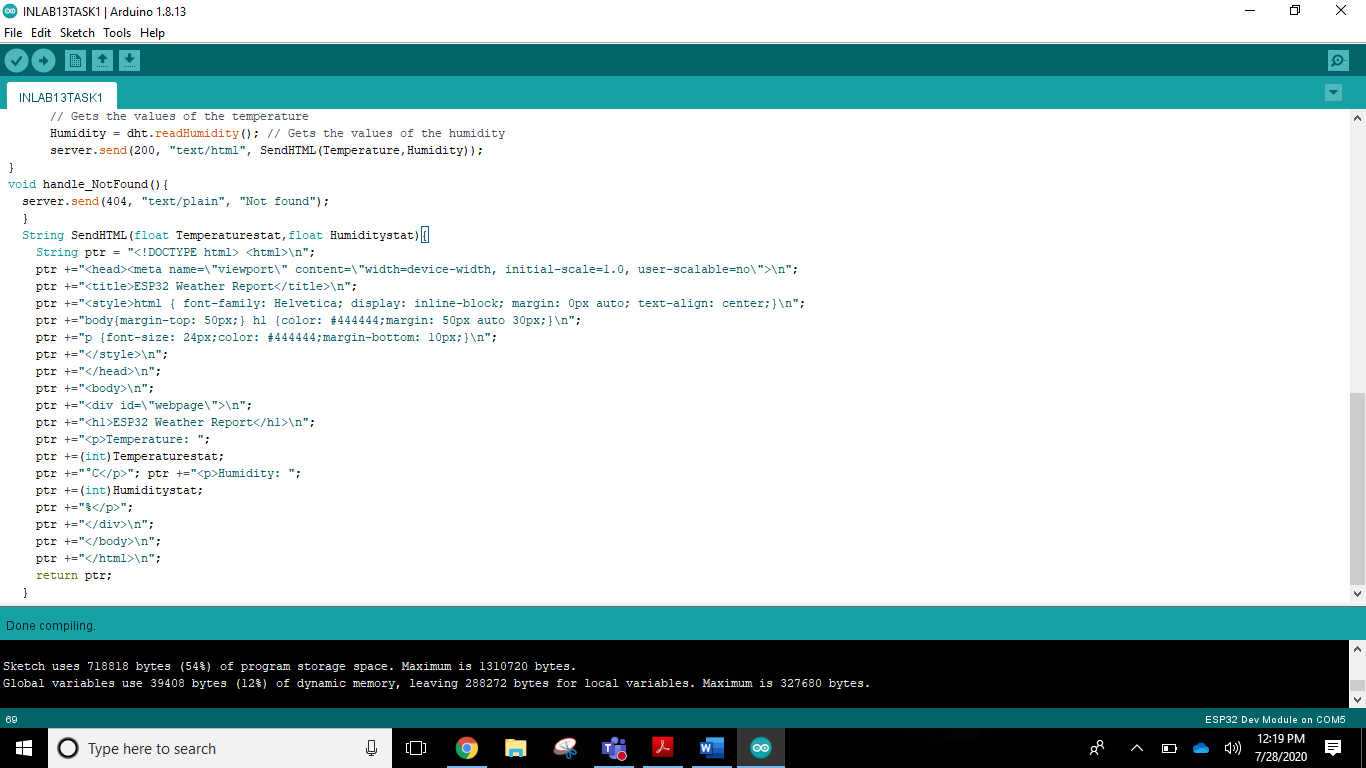
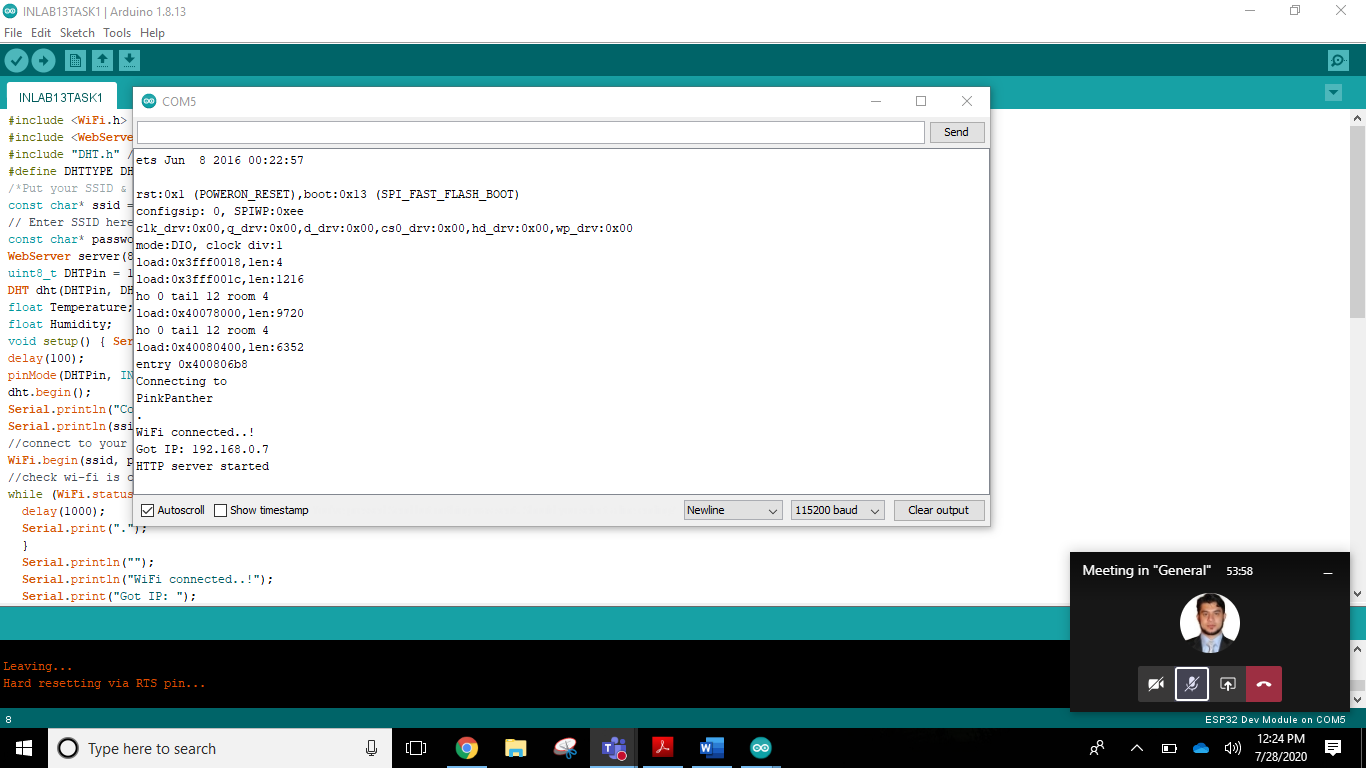
### Tools

* Arduino
* ESP32 Board
* 2 LEDs
* DHT11 sensor
* Relay board
* 220V AC Light

### Task 1

**Temperature/Humidity Sensor Read using DHT Sensor:**

|  |
| --- |
| #include <WiFi.h> #include <WebServer.h> #include "DHT.h"  // Uncomment one of the lines below for whatever DHT sensor type you're using!  #define DHTTYPE DHT11 // DHT 11  //#define DHTTYPE DHT21 // DHT 21 (AM2301)  //#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321  /\*Put your SSID & Password\*/  const char\* ssid = "Ahsan"; // Enter SSID here  const char\* password = "12345678"; //Enter Password here  WebServer server(80);  // DHT Sensor uint8\_t DHTPin = 15;  // Initialize DHT sensor. DHT dht(DHTPin, DHTTYPE);  float Temperature; float Humidity;  void setup() { Serial.begin(115200); delay(100);  pinMode(DHTPin, INPUT);  dht.begin();  Serial.println("Connecting to "); Serial.println(ssid);  //connect to your local wi-fi network WiFi.begin(ssid, password);  //check wi-fi is connected to wi-fi network while (WiFi.status() != WL\_CONNECTED) { delay(1000);  Serial.print(".");  }  Serial.println(""); Serial.println("WiFi connected..!");  Serial.print("Got IP: "); Serial.println(WiFi.localIP());  server.on("/", handle\_OnConnect); server.onNotFound(handle\_NotFound);  server.begin();  Serial.println("HTTP server started");  }  void loop() { server.handleClient();  }  void handle\_OnConnect() {  Temperature = dht.readTemperature(); // Gets the values of the temperature  Humidity = dht.readHumidity(); // Gets the values of the humidity  server.send(200, "text/html", SendHTML(Temperature,Humidity));  }  void handle\_NotFound(){  server.send(404, "text/plain", "Not found");  }  String SendHTML(float Temperaturestat,float Humiditystat){ String ptr = "<!DOCTYPE html> <html>\n";  ptr +="<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";  ptr +="<title>ESP32 Weather Report</title>\n";  ptr +="<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";  ptr +="body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;}\n";  ptr +="p {font-size: 24px;color: #444444;margin-bottom: 10px;}\n"; ptr +="</style>\n";  ptr +="</head>\n"; ptr +="<body>\n";  ptr +="<div id=\"webpage\">\n";  ptr +="<h1>ESP32 Weather Report</h1>\n";  ptr +="<p>Temperature: "; ptr +=(int)Temperaturestat; ptr +="°C</p>";  ptr +="<p>Humidity: "; ptr +=(int)Humiditystat; ptr +="%</p>";  ptr +="</div>\n"; ptr +="</body>\n"; ptr +="</html>\n"; return ptr;  } |



### Task 2

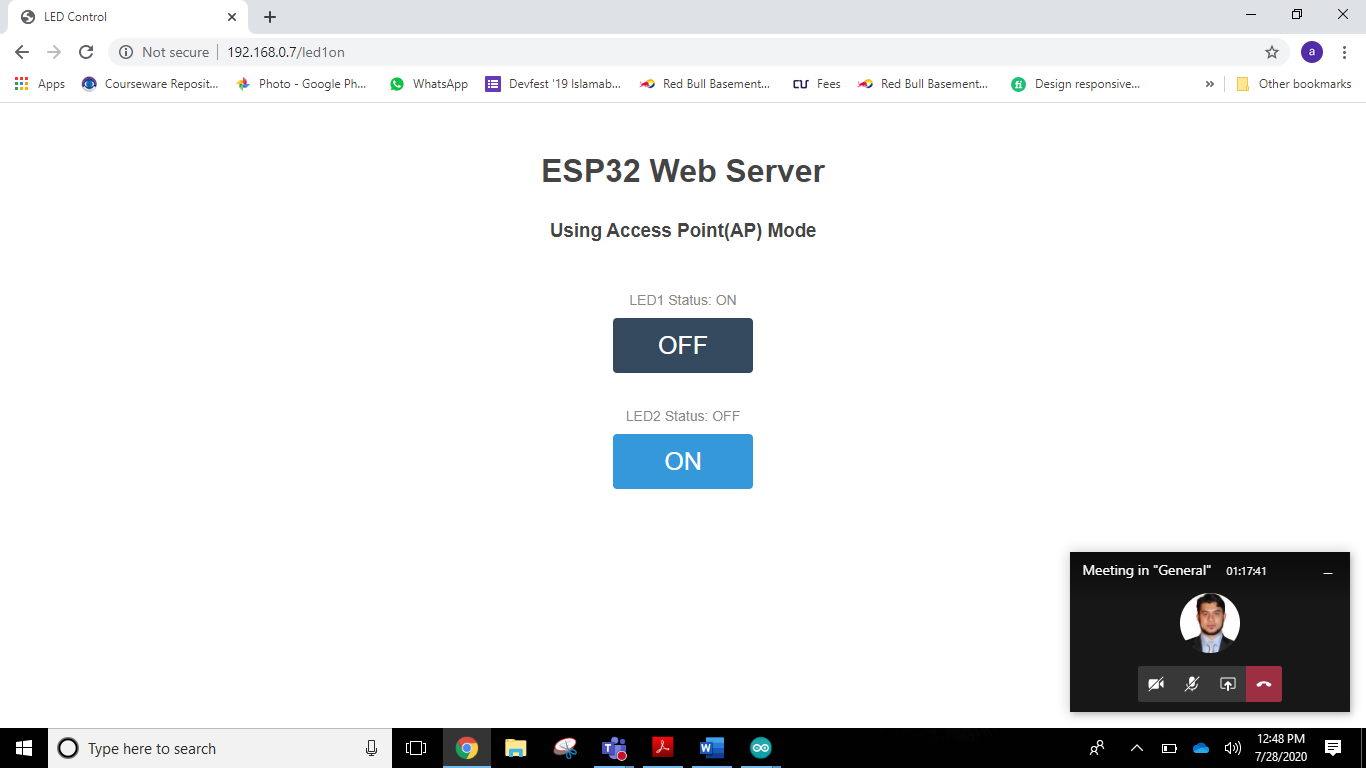
# Use ESP8266 Smart Config App from Google Play Store to allocate IP to ESP32 by your mobile at Run-time

|  |
| --- |
| #include <WiFi.h>  #include <WebServer.h>  // Use ESP8266 Smart Config App from playstore to allocate ip to ESP32 by your mobile,  // Use Wifi SSID and Pwd, which will assign IP to ESP 32  // No need to define Password and SSID within your code.  WebServer server(80);  uint8\_t LED1pin = 4;  bool LED1status = LOW;  uint8\_t LED2pin = 5;  bool LED2status = LOW;  void setup() {  Serial.begin(115200);  /////////////////////////////////////////////  //Serial.begin(115200);  //Init WiFi as Station, start SmartConfig  WiFi.mode(WIFI\_AP\_STA);  WiFi.beginSmartConfig()  //Wait for SmartConfig packet from mobile  Serial.println("Waiting for SmartConfig.");  while (!WiFi.smartConfigDone()) {  delay(500);  Serial.print(".");  }  Serial.println("");  Serial.println("SmartConfig received.");  //Wait for WiFi to connect to AP  Serial.println("Waiting for WiFi");  while (WiFi.status() != WL\_CONNECTED)  { delay(500);  Serial.print(".");  }  Serial.println("WiFi Connected.");  Serial.print("IP Address: ");  Serial.println(WiFi.localIP());  ///////////////////////////////////////////////  pinMode(LED1pin, OUTPUT); pinMode(LED2pin, OUTPUT);  //WiFi.softAP(ssid, password);  //WiFi.softAPConfig(local\_ip, gateway, subnet);  delay(100);  server.on("/", handle\_OnConnect);  server.on("/led1on", handle\_led1on);  server.on("/led1off", handle\_led1off);  server.on("/led2on", handle\_led2on);  server.on("/led2off", handle\_led2off);  server.onNotFound(handle\_NotFound);  server.begin();  Serial.println("HTTP server started");  }  void loop() { server.handleClient(); if(LED1status)  {digitalWrite(LED1pin, HIGH);}  else  {digitalWrite(LED1pin, LOW);}  if(LED2status)  {digitalWrite(LED2pin, HIGH);}  else  {digitalWrite(LED2pin, LOW);}  }  void handle\_OnConnect() {  LED1status = LOW;  LED2status = LOW;  Serial.println("GPIO4 Status: OFF | GPIO5 Status: OFF");  server.send(200, "text/html", SendHTML(LED1status,LED2status));  }  void handle\_led1on() {  LED1status = HIGH;  Serial.println("GPIO4 Status: ON");  server.send(200, "text/html", SendHTML(true,LED2status));  }  void handle\_led1off() {  LED1status = LOW;  Serial.println("GPIO4 Status: OFF");  server.send(200, "text/html", SendHTML(false,LED2status));  }  void handle\_led2on() { LED2status = HIGH;  Serial.println("GPIO5 Status: ON");  server.send(200, "text/html", SendHTML(LED1status,true));  }  void handle\_led2off() {  LED2status = LOW;  Serial.println("GPIO5 Status: OFF");  server.send(200, "text/html", SendHTML(LED1status,false));  }  void handle\_NotFound(){  server.send(404, "text/plain", "Not found");  }  String SendHTML(uint8\_t led1stat,uint8\_t led2stat){ String ptr = "<!DOCTYPE html> <html>\n";  ptr +="<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";  ptr +="<title>LED Control</title>\n";  ptr +="<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";  ptr +="body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;} h3 {color: #444444;margin-bottom: 50px;}\n";  ptr +=".button {display: block;width: 80px;background-color: #3498db;border: none;color: white;padding: 13px 30px;text-decoration: none;font-size: 25px;margin: 0px auto 35px;cursor: pointer;border-radius: 4px;}\n";  ptr +=".button-on {background-color: #3498db;}\n";  ptr +=".button-on:active {background-color: #2980b9;}\n"; ptr +=".button-off {background-color: #34495e;}\n";  ptr +=".button-off:active {background-color: #2c3e50;}\n";  ptr +="p {font-size: 14px;color: #888;margin-bottom: 10px;}\n"; ptr +="</style>\n";  ptr +="</head>\n"; ptr +="<body>\n";  ptr +="<h1>ESP32 Web Server</h1>\n";  ptr +="<h3>Using Access Point(AP) Mode</h3>\n";  if(led1stat)  {ptr +="<p>LED1 Status: ON</p><a class=\"button button-off\" href=\"/led1off\">OFF</a>\n";} else  {ptr +="<p>LED1 Status: OFF</p><a class=\"button button-on\" href=\"/led1on\">ON</a>\n";}  if(led2stat)  {ptr +="<p>LED2 Status: ON</p><a class=\"button button-off\" href=\"/led2off\">OFF</a>\n";} else  {ptr +="<p>LED2 Status: OFF</p><a class=\"button button-on\" href=\"/led2on\">ON</a>\n";}  ptr +="</body>\n"; ptr +="</html>\n"; return ptr;  } |

### 

### 

### 

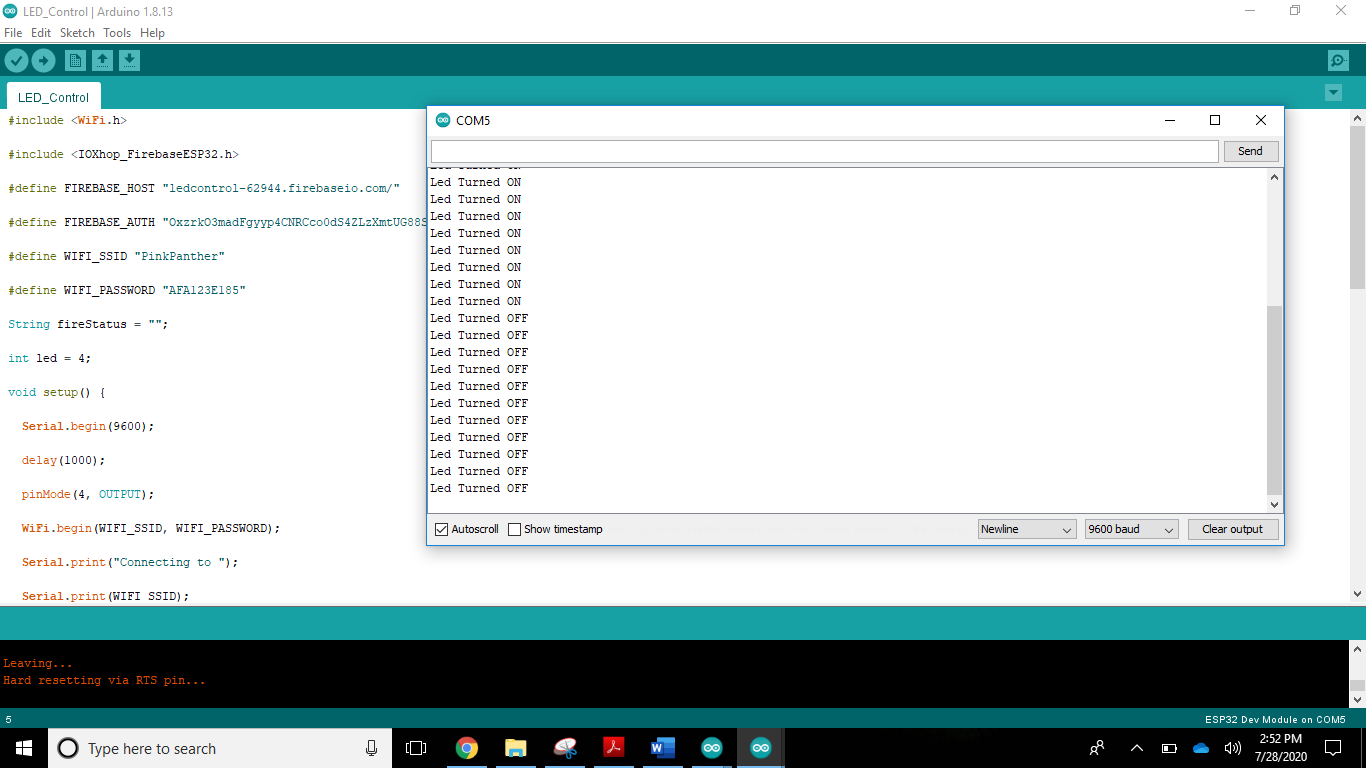


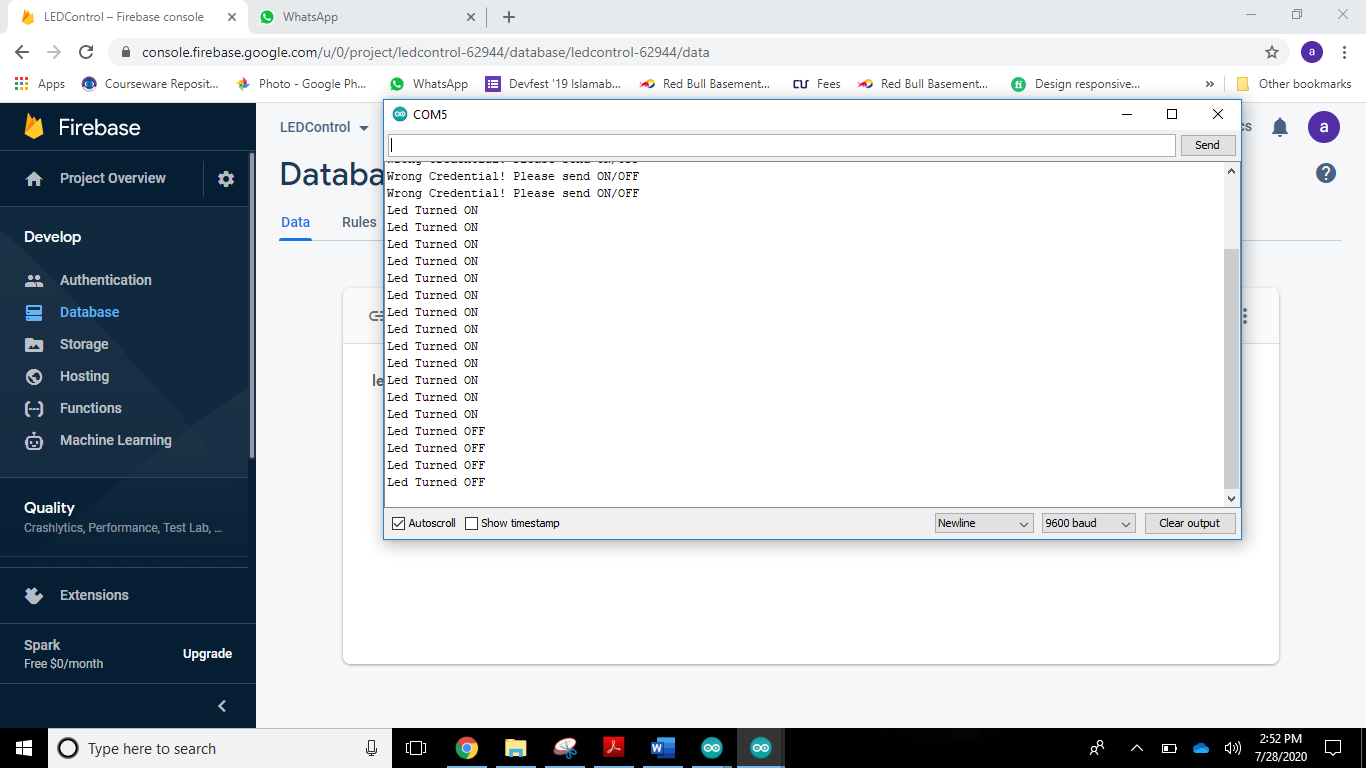
### 

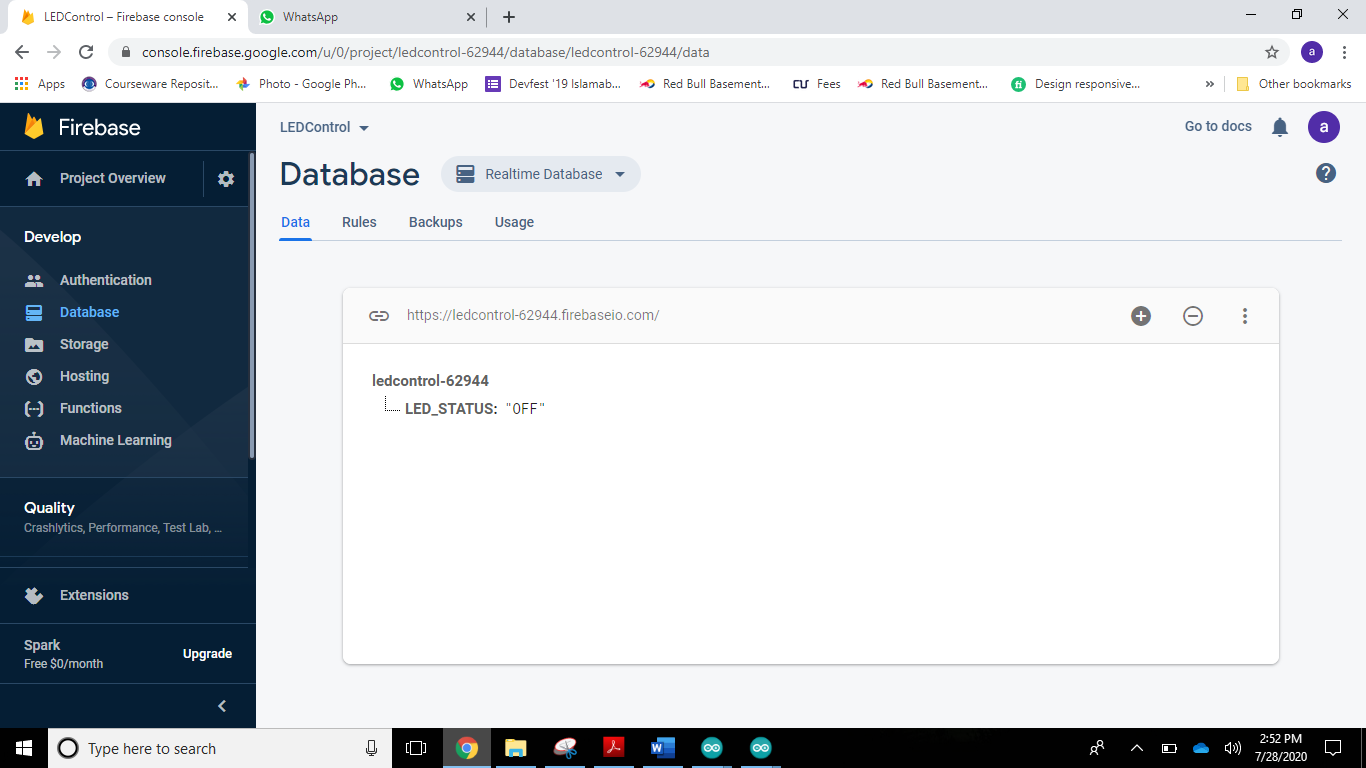
### Task 3

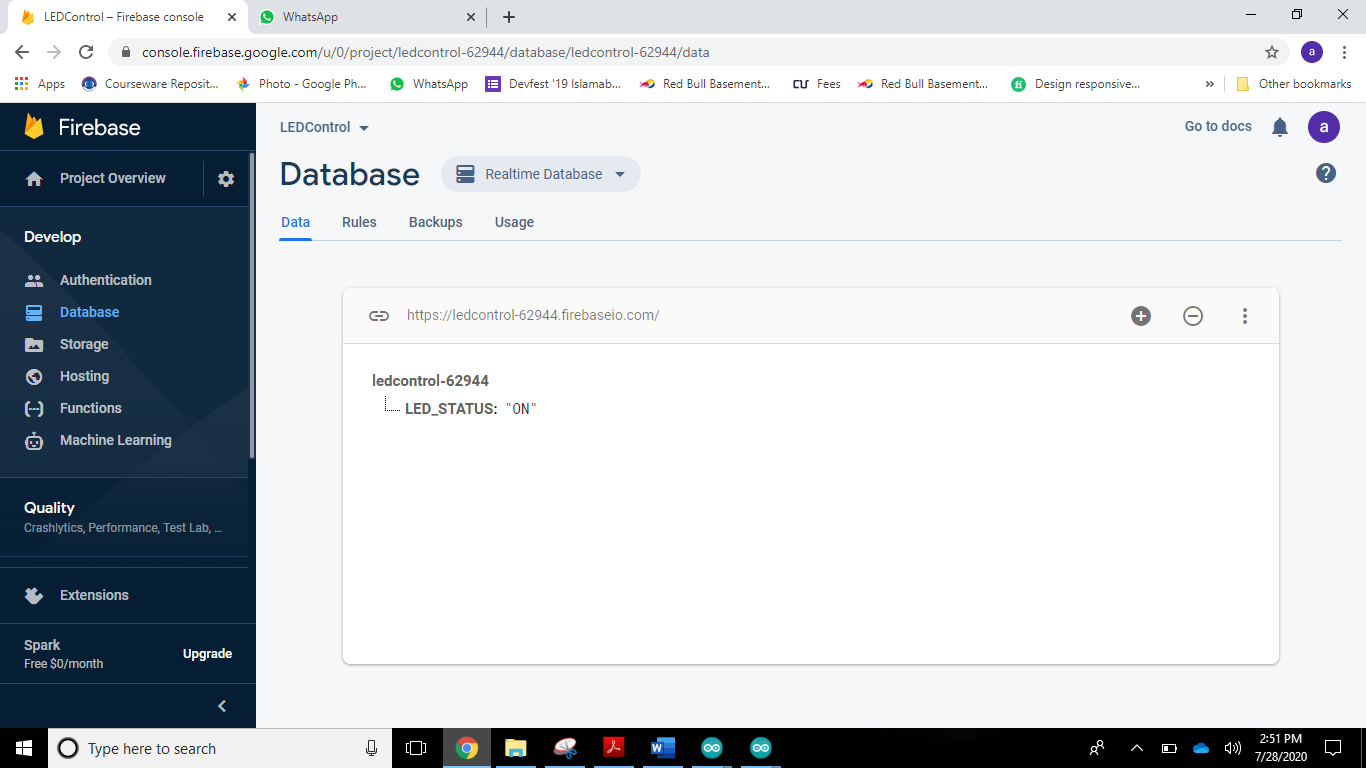
### LED control using Google Firebase and ESP32

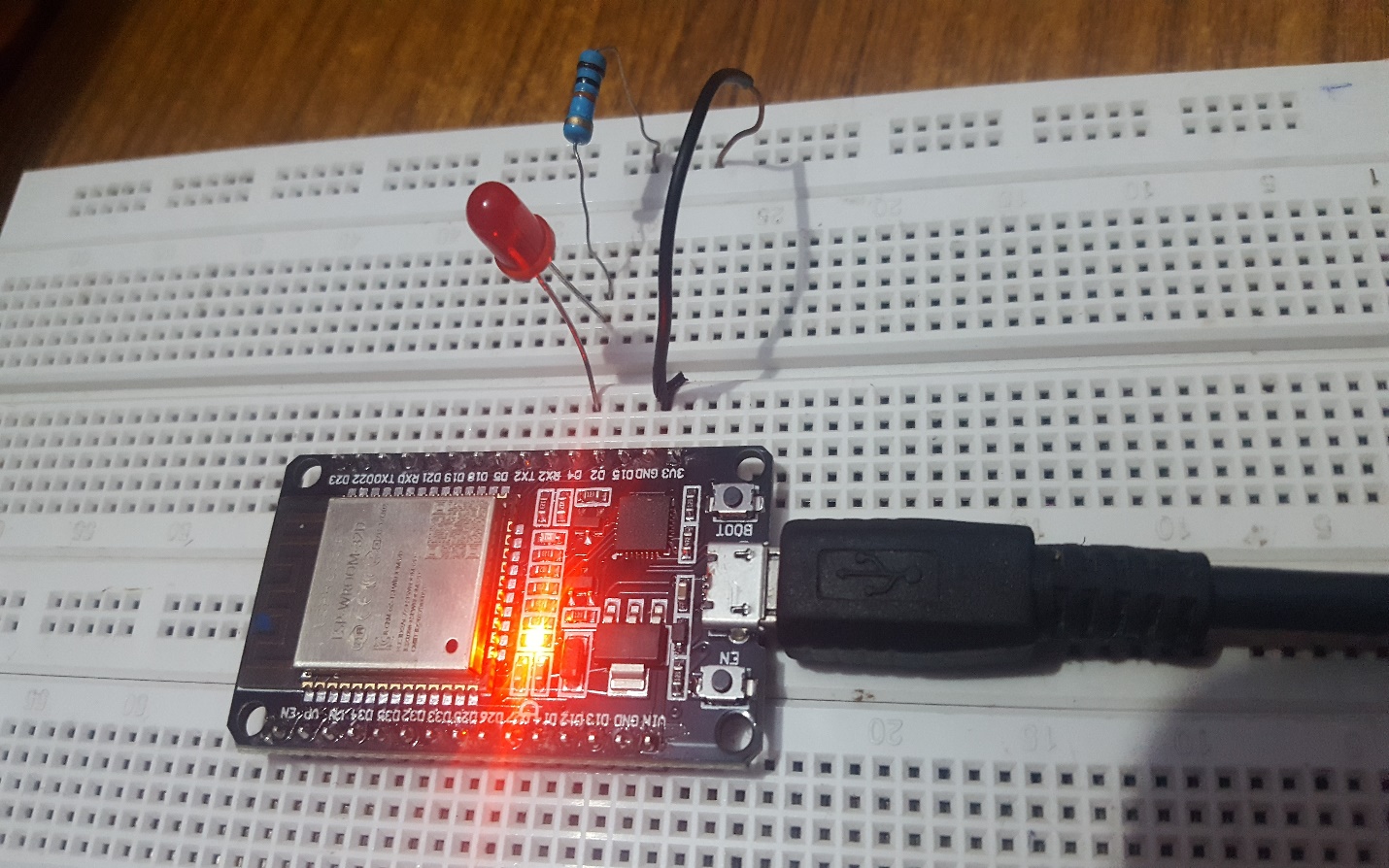
|  |
| --- |
| #include <WiFi.h>  #include <IOXhop\_FirebaseESP32.h>  #define FIREBASE\_HOST "ledcontrol-62944.firebaseio.com/"  #define FIREBASE\_AUTH "OxzrkO3madFgyyp4CNRCco0dS4ZLzXmtUG88SJ5k"  #define WIFI\_SSID "PinkPanther"  #define WIFI\_PASSWORD "AFA123E185"  String fireStatus = ""; // led status received from firebase  int led = 4;  void setup() {  Serial.begin(9600);  delay(1000);  pinMode(4, OUTPUT);  WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); //try to connect with wifi  Serial.print("Connecting to ");  Serial.print(WIFI\_SSID);  while (WiFi.status() != WL\_CONNECTED) {  Serial.print(".");  delay(500);  }  Serial.println();  Serial.print("Connected to ");  Serial.println(WIFI\_SSID);  Serial.print("IP Address is : ");  Serial.println(WiFi.localIP()); //print local IP address  Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH); // connect to firebase  Firebase.setString("LED\_STATUS", "OFF"); //send initial string of led status  }  void loop() {  fireStatus = Firebase.getString("LED\_STATUS"); // get led status input from firebase  if (fireStatus == "ON") { // compare the input of led status received from firebase  Serial.println("Led Turned ON");  digitalWrite(4, HIGH); // make output led ON  }  else if (fireStatus == "OFF") { // compare the input of led status received from firebase  Serial.println("Led Turned OFF");  digitalWrite(4, LOW); // make output led OFF  }  else {  Serial.println("Wrong Credential! Please send ON/OFF");  }  } |

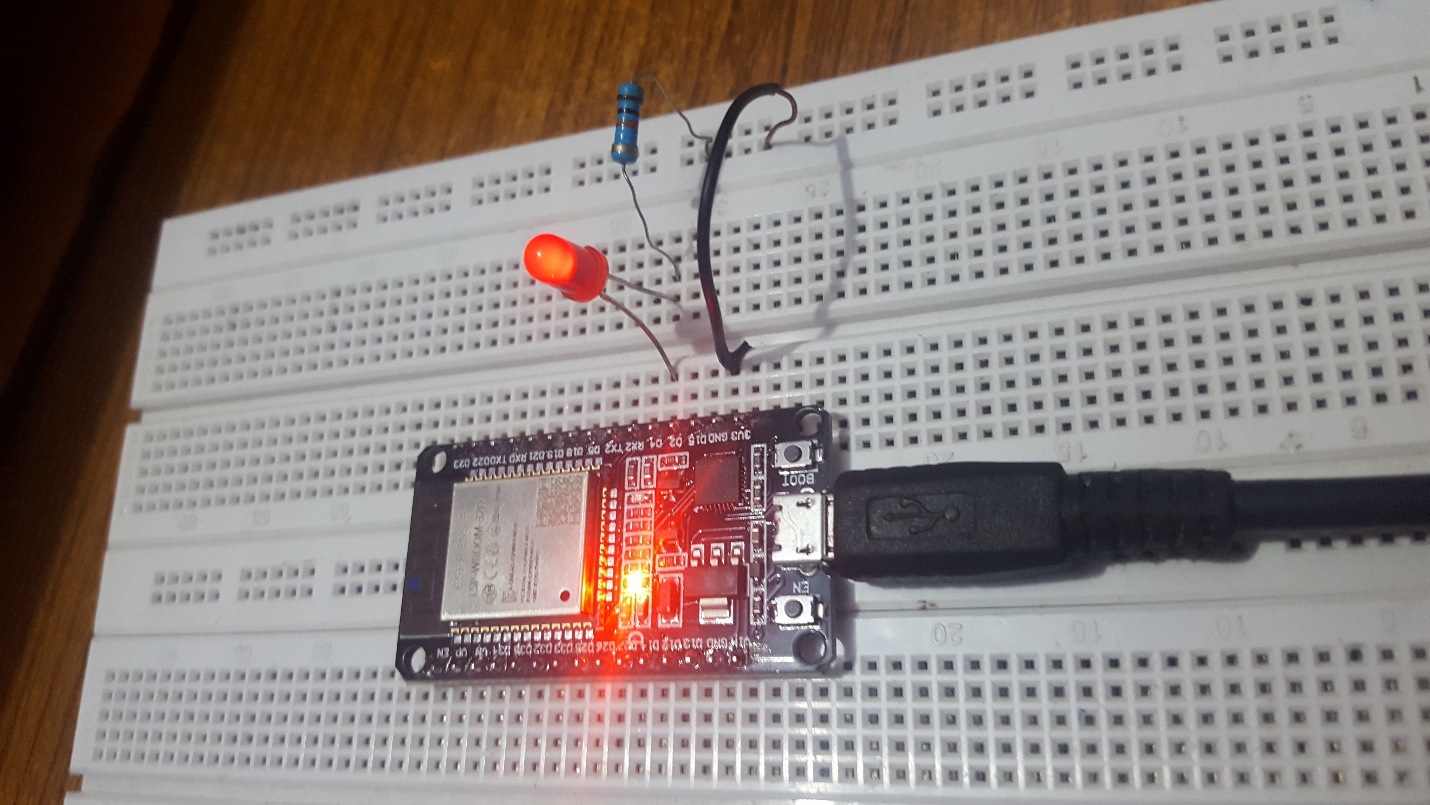










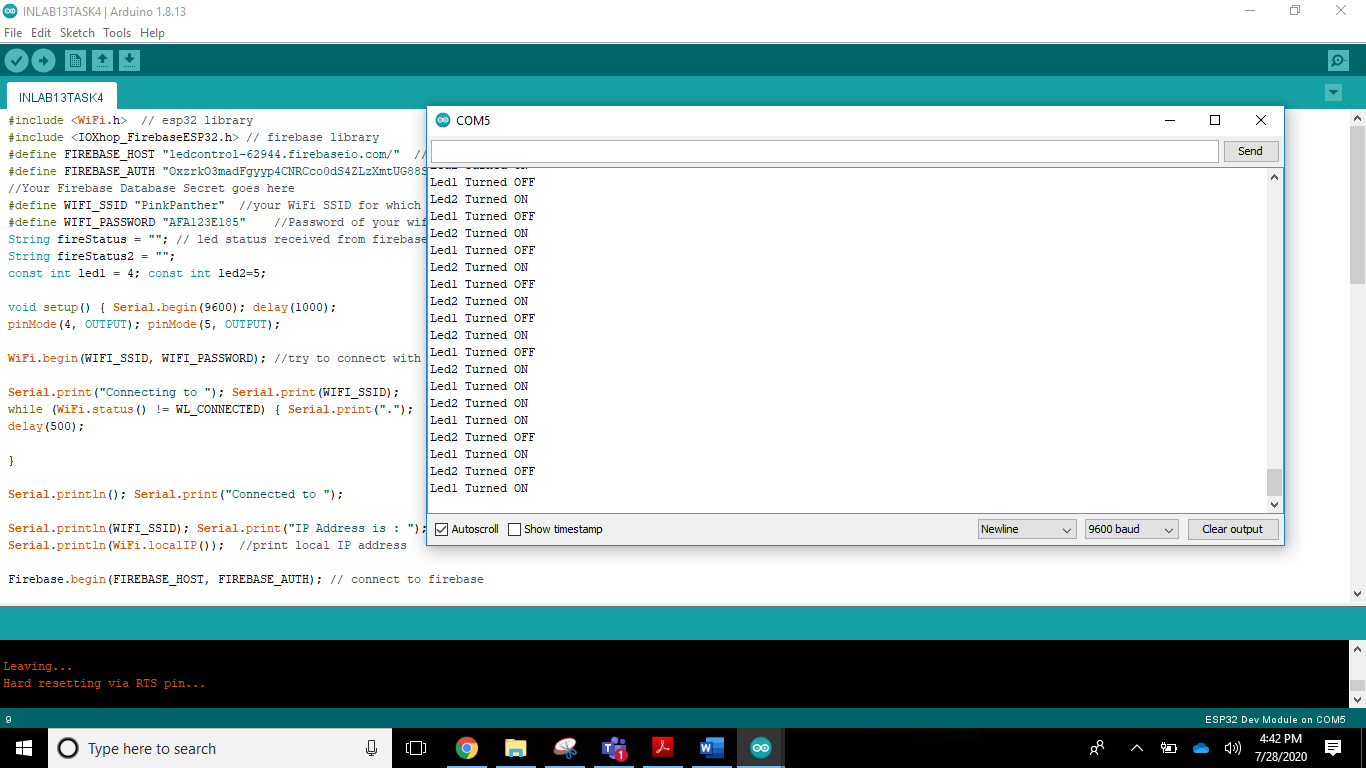


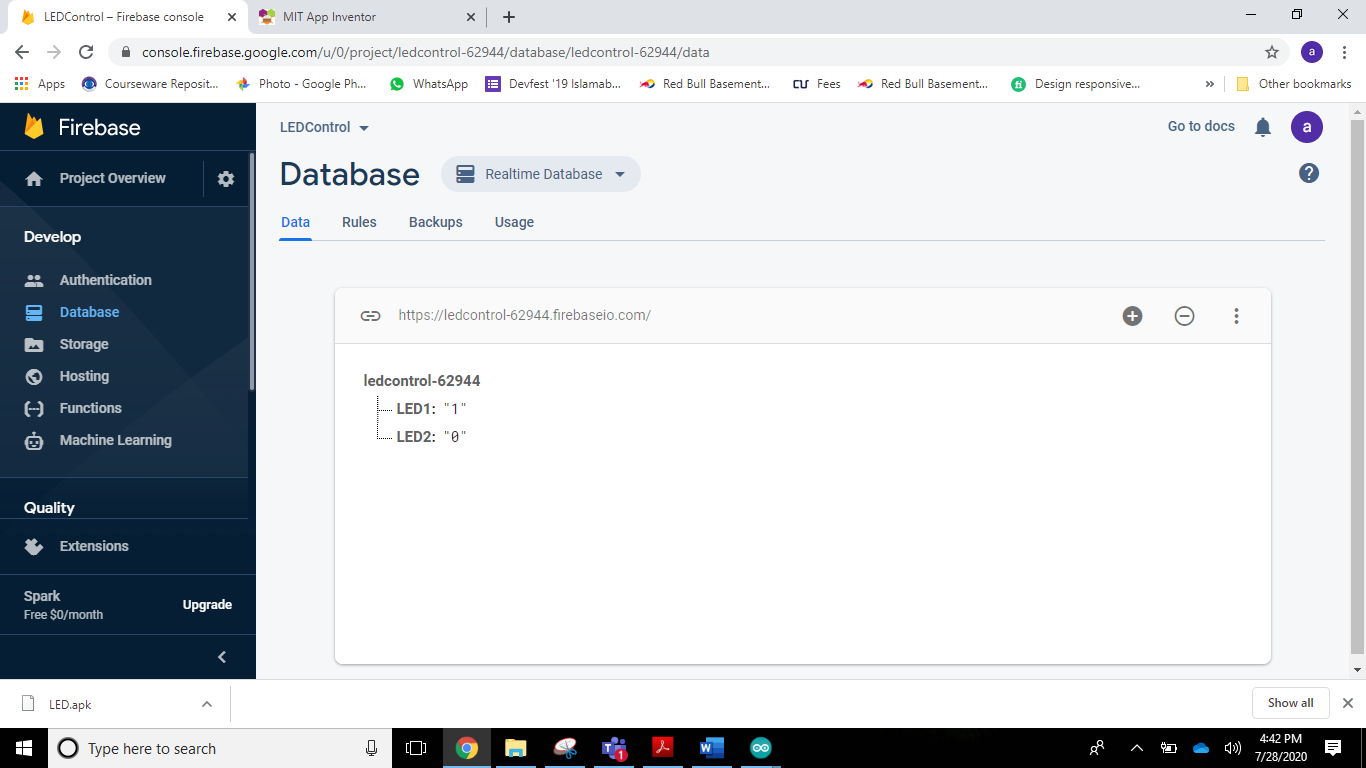
### Task 4

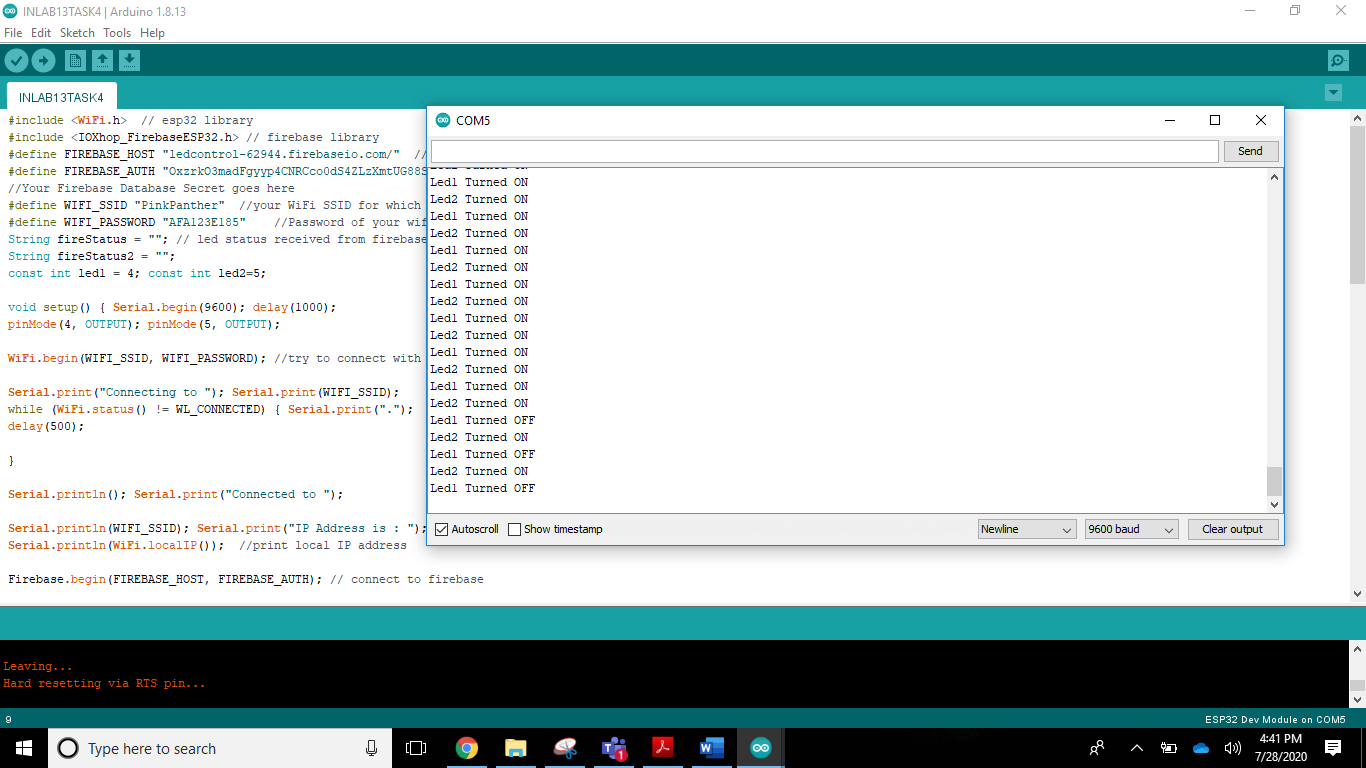
**LED control using Google Firebase, ESP32 and Mobile App**

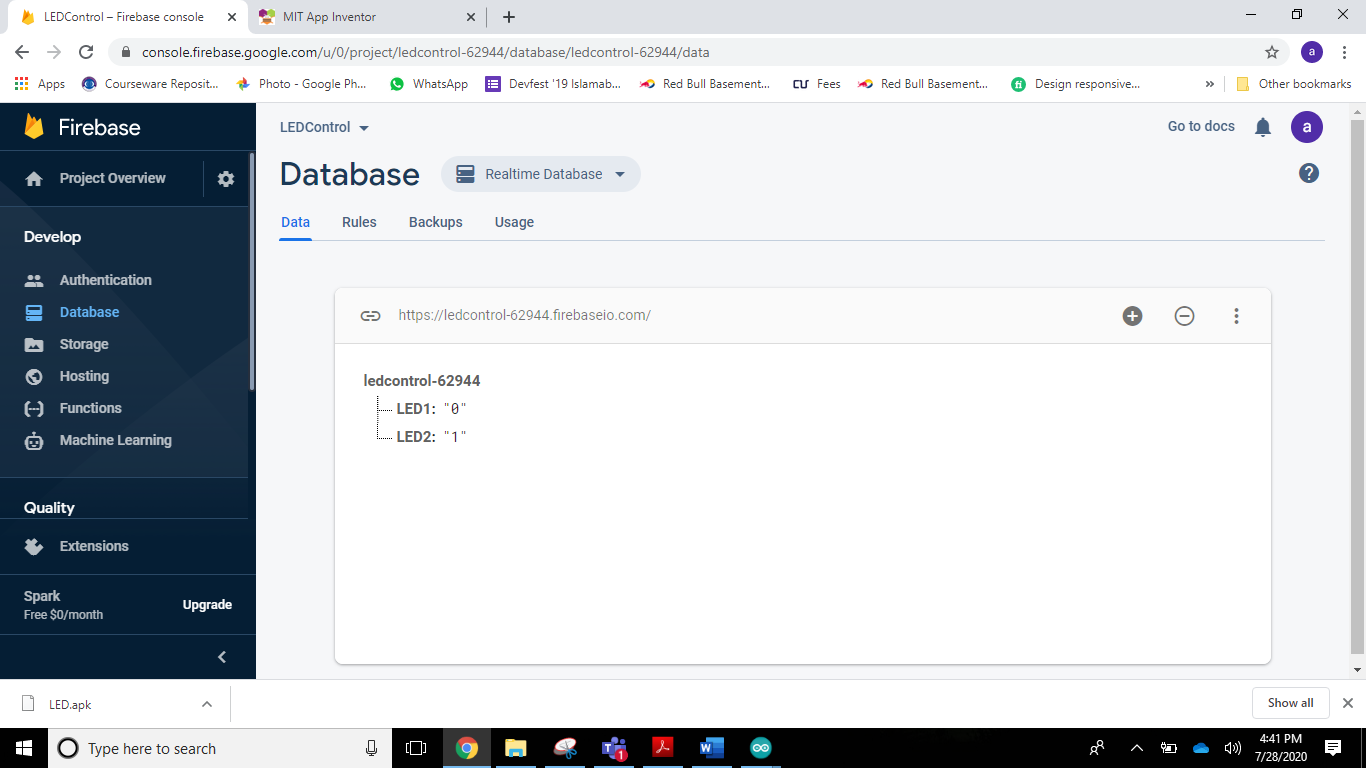
|  |
| --- |
| #include <WiFi.h> // esp32 library#include <IOXhop\_FirebaseESP32.h> // firebase library#define FIREBASE\_HOST "ledcontrol-62944.firebaseio.com/" //Your Firebase //Project URL goes here without "http:" , "\" and "/"#define FIREBASE\_AUTH "OxzrkO3madFgyyp4CNRCco0dS4ZLzXmtUG88SJ5k"//Your Firebase Database Secret goes here#define WIFI\_SSID "PinkPanther" //your WiFi SSID for which yout NodeMCU connects#define WIFI\_PASSWORD "AFA123E185" //Password of your wifi networkString fireStatus = ""; // led status received from firebaseString fireStatus2 = "";const int led1 = 4; const int led2=5;void setup() { Serial.begin(9600); delay(1000);pinMode(4, OUTPUT); pinMode(5, OUTPUT);WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); //try to connect with wifiSerial.print("Connecting to "); Serial.print(WIFI\_SSID);while (WiFi.status() != WL\_CONNECTED) { Serial.print(".");delay(500);}Serial.println(); Serial.print("Connected to ");Serial.println(WIFI\_SSID); Serial.print("IP Address is : ");Serial.println(WiFi.localIP()); //print local IP addressFirebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH); // connect to firebaseFirebase.setString("LED1", "0"); //send initial string of led statusFirebase.setString("LED2", "0");}void loop() {fireStatus = Firebase.getString("LED1"); // get led status input from firebaseif (fireStatus == "1") { // compare the input of led status received from firebaseSerial.println("Led1 Turned ON");digitalWrite(4, HIGH); // make output led ON}else if (fireStatus == "0") { // compare the input of led status received from firebaseSerial.println("Led1 Turned OFF");digitalWrite(4, LOW); // make output led OFF}else {Serial.println("Wrong Credential for LED1! Please send ON/OFF");}fireStatus2 = Firebase.getString("LED2");if (fireStatus2 == "1") { // compare the input of led status received from firebaseSerial.println("Led2 Turned ON");digitalWrite(5, HIGH); // make output led ON}else if (fireStatus2 == "0") { // compare the input of led status received from firebaseSerial.println("Led2 Turned OFF");digitalWrite(5, LOW); // make output led OFF}else {Serial.println("Wrong Credential for LED2! Please send ON/OFF");}} |

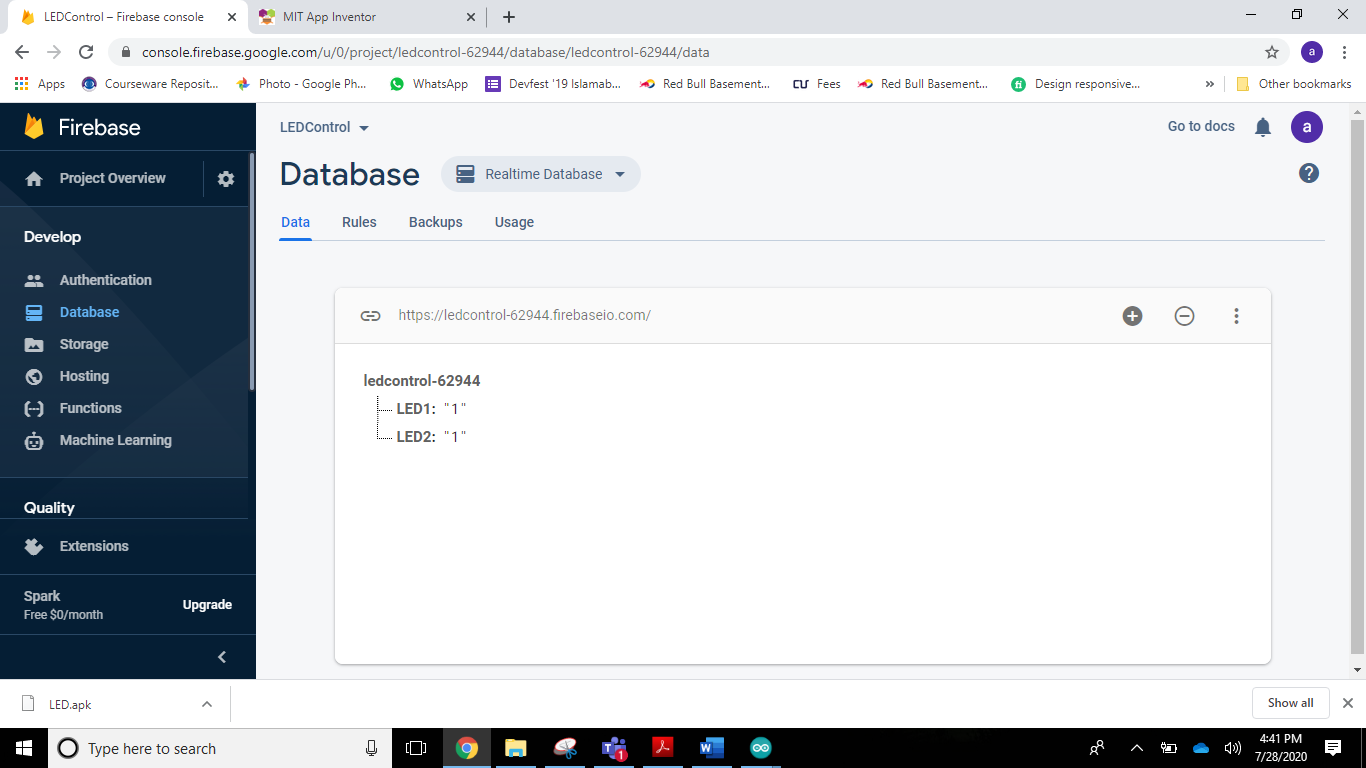
# 

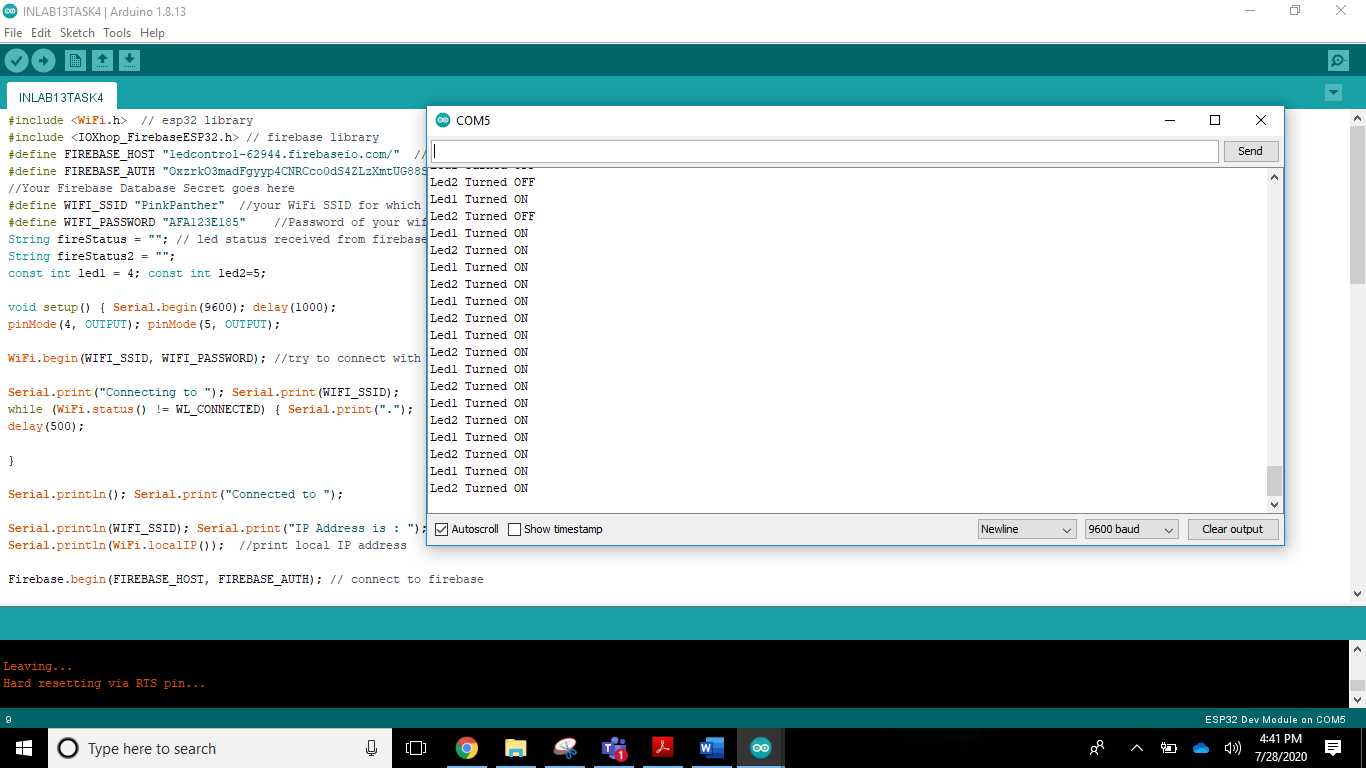


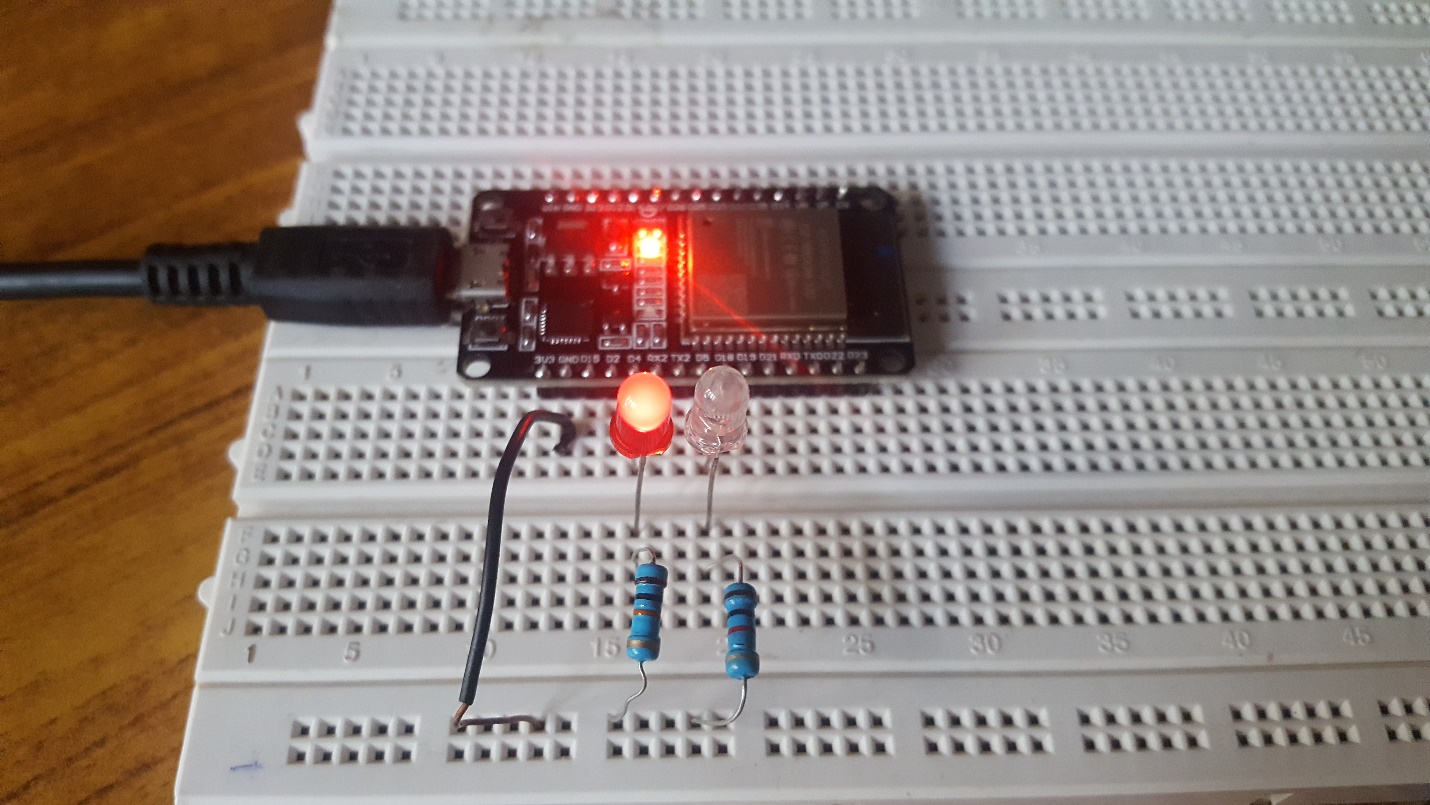


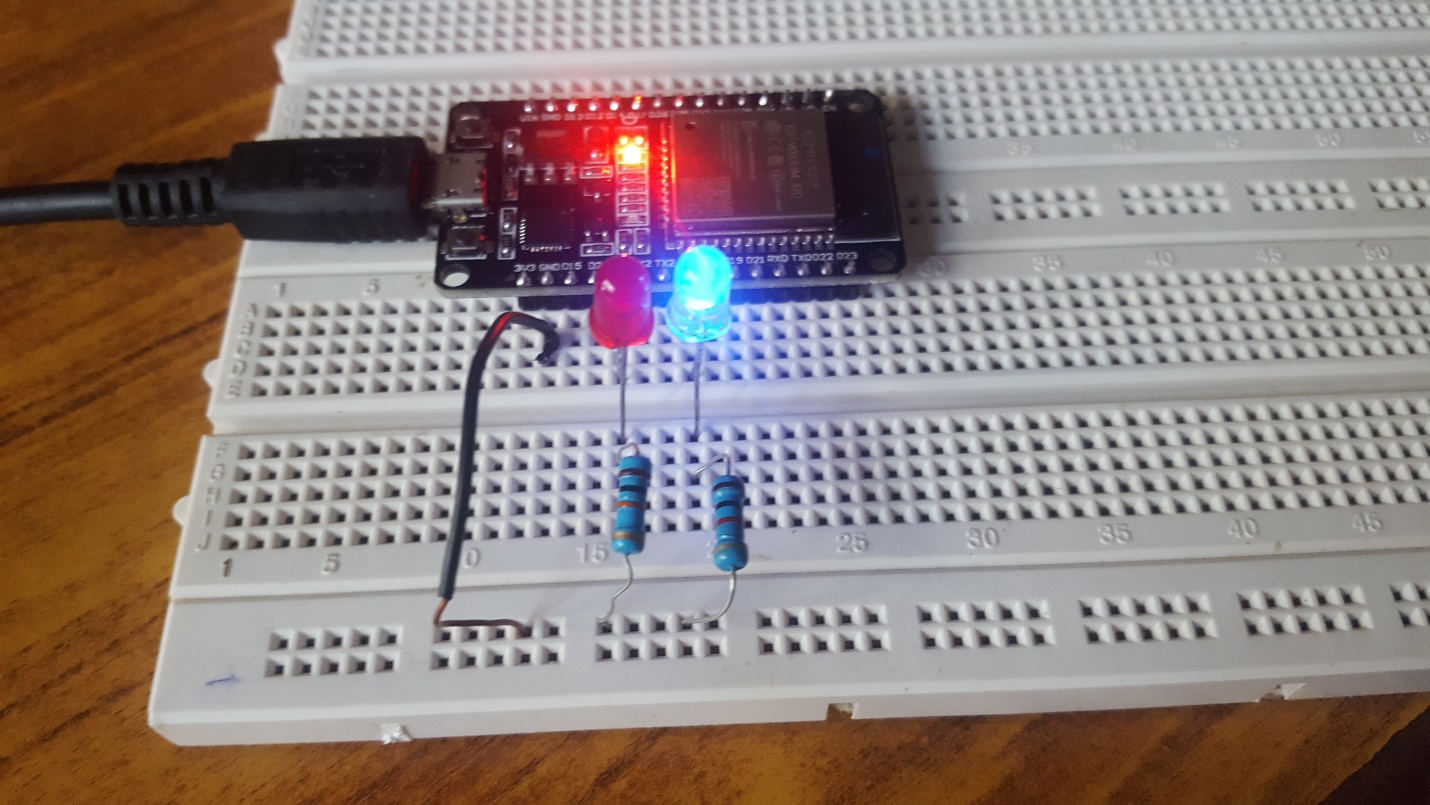
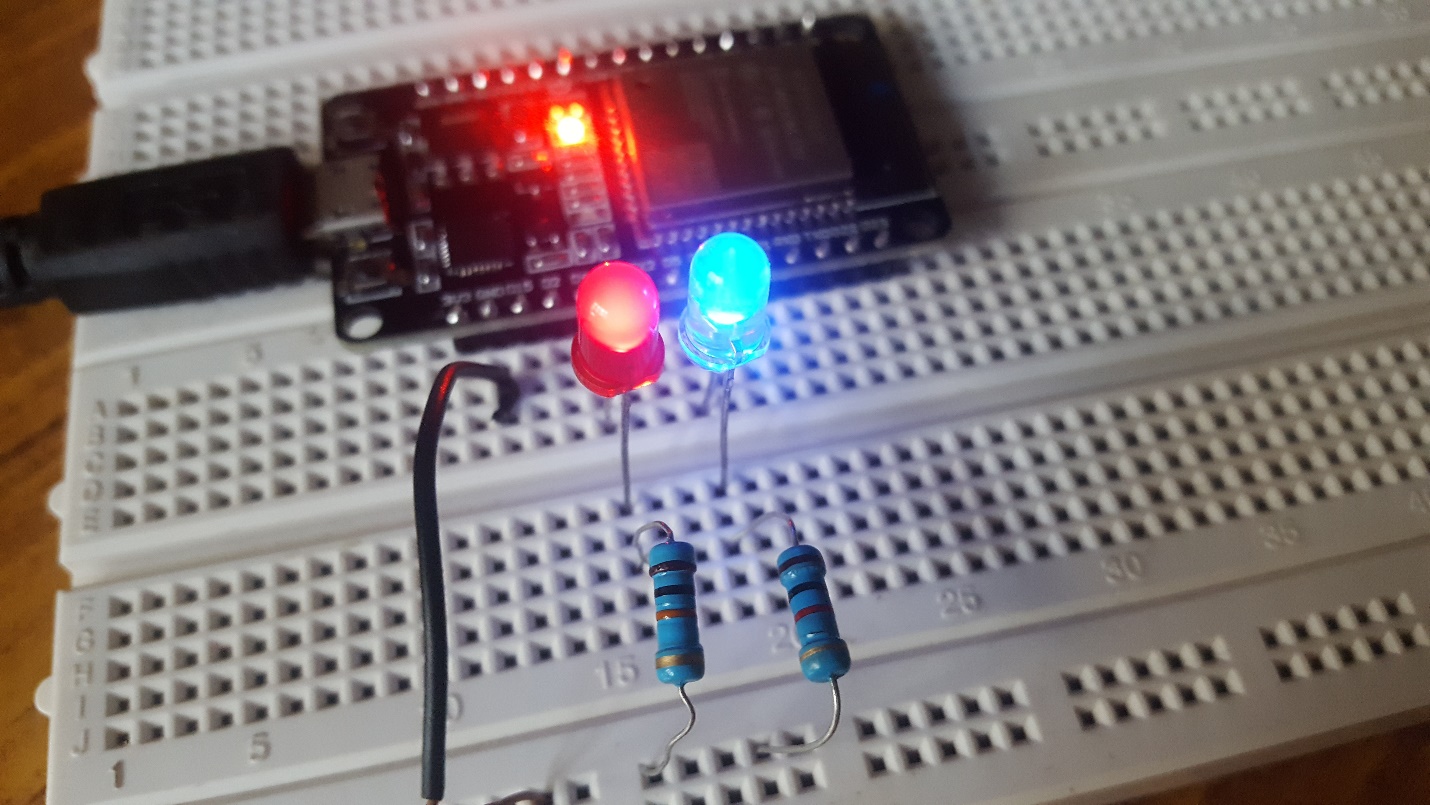










.com

# Conclusion:

|  |
| --- |
| In this lab we learnt how to make an Android Application using MIT App Inventer, connect it to the database and read the data from that database using and ESP32. We also learnt how to make a Real Time Database on Google Firebase. |