**Internet of Things**

**Lab Report 10**

**Hafiz Ahmad**

**19l-1316**

**Section-7A2**

**Create APIs and fetch data online using ESP8266 NodeMCU**

**INTRODUCTION:**

A well-liked development board based on the ESP8266 is the NodeMCU.In addition to the ESP8266 SoC-containing ESP12 module, it has a USB connector and breadboard-friendly pins to facilitate project testing and development on the ESP8266.Memory and processor upgrades:The clock speed of NodeMCu is 80 MHz, and it has 4 MB of flash memory.IoT-ready built-in TCP/IP stack:The NodeMCU has a Wifi connection and can use Wifi to connect to the internet.It is ideal for Internet of Things applications.In normal operation mode, an ESP8266 module typically consumes 70mA, so if we run it continuously in normal mode, it will only run for one and a half days.

**OBJECTIVES:**

To learn about to Create APIs and fetch data online using ESP8266 NodeMCU.

**Lab code:**

#include <ESP8266WiFi.h>

const char\* ssid = "EE"; // write your wifi name

const char\* password = "ee123456"; // write your wifi password

const char\* host = "api.thingspeak.com";

int value = 1;

void setup() {

Serial.begin(115200);

delay(10);

// We start by connecting to a WiFi network

Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}//end setup

void loop() {

delay(5000);

Serial.print("Connecting to ");

Serial.println(host);

// Use WiFiClient class to create TCP connections

WiFiClient Client;

const int httpPort = 80;

if (!Client.connect(host, httpPort)) {

Serial.println("Connection failed");

return;

}

// We now create a URI for the request

String url = "/apps/thinghttp/send\_request?api\_key=UFJO4VUZAB5BCCBF"; //paste url of your api

Serial.print("Requesting URL: ");

Serial.println(host + url);

Serial.println(String("TRY: ") + value + ".");

// This will send the request to the server

Client.print(String("GET ") + url +"&header=false" + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" + "Connection: close\r\n\r\n");

delay(2000); //change delay to 2000

// Read all the lines of the reply from server and print them to Serial

while(Client.available()){

String line = Client.readStringUntil('\r');

Serial.println(line);

}

Serial.println("");

Serial.println(String("Try nr. ") + value + " is finished.");

Serial.println("Waiting for next try...");

Serial.println("");

value = value + 1;

delay(20000);

} 

Graphical user interface, text, application, email

Description automatically generated

**Application:**

* NodeMCU is an open source platform based on ESP8266 which can connect objects and let data transfer using the Wi-Fi protocol. In addition, by providing some of the most important features of microcontrollers such as GPIO, PWM, ADC, and etc, it can solve many of the project's needs alone.
* The Smart security devices, including surveillance cameras and smart locks.
* Smart energy devices, including HVACs and thermostats.
* Smart industrial devices, including Programmable Logic Controllers (PLCs)
* Smart medical devices, including wearable health monitors.

**Issues:**

we never find any issue regarding this lab.

**Conclusion:**

In this lab we perform An application programming point of interaction (Programming interface) is a figuring point of interaction which characterizes associations between different programming go-betweens.It specifies the kinds of calls and requests that can be made, how to make them, the data formats that should be used, the rules to follow, and other details.It can also provide mechanisms for extension, allowing users to modify and enhance the functionality that is already there.An API can be completely unique to a component or based on an industry standard to guarantee interoperability.APIs make modular programming possible by hiding information. This lets users use the interface without having to use the implementation.