IoT Embedded Developer Assignment

STEP 1:

- Connecting ESP32 wifi module to API server.
- IP address will be extracted .

```
/*****************************************************************/

#include <WiFi.h>

#include <WiFiClientSecure.h>

char ssid[] = "SSID";

char password[] = "PASS";

WiFiClientSecure client;  // For HTTPS requests

#define TEST_HOST "https://fastag-internal.parkzap.com/account/mockable_test/" //Given URL

#define TEST_HOST_FINGERPRINT "db d5 80 39 39 26 68 31 ed 58 07 a5 31 af ec 3d 3a 1b 53 d1" // connecting finger print to get more accurate url.

void setup() {

Serial.begin(115200); // read how many bits/sec.

 WiFi.mode(WIFI_STA); // connect wifi

 WiFi.disconnect();

 delay(100);

 Serial.print("Connecting Wifi: ");

 Serial.println(ssid);

 WiFi.begin(ssid, password);

 while (WiFi.status() != WL_CONNECTED) {

  Serial.print(".");
```

```
  delay(500);
 }
 Serial.println("");
 Serial.println("WiFi connected");
 Serial.println("IP address: ");
 IPAddress ip = WiFi.localIP();
 Serial.println(ip);
 client.setFingerprint(TEST_HOST_FINGERPRINT);
makeHTTPRequest();
}

void makeHTTPRequest() {
 if (!client.connect(TEST_HOST, 443))
 {
  Serial.println(F("Connection failed"));
  return;
 }
 yield();
 client.print(F("GET ")); // Send HTTP request
 client.print("https://fastag-internal.parkzap.com/account/mockable_test/");
 client.println(F(" HTTP/1.1"));
 //Headers
 client.print(F("Host: "));
 client.println(TEST_HOST);
```

```
client.println(F("Cache-Control: no-cache"));
if (client.println() == 0)
{
  Serial.println(F("Failed to send request"));
  return;
}
// Check HTTP status
char status[32] = {0};
client.readBytesUntil('\r', status, sizeof(status));
if (strcmp(status, "HTTP/1.1 200 OK") != 0)
{
  Serial.print(F("Unexpected response: "));
  Serial.println(status);
  return;
}
// Skip HTTP headers
char endOfHeaders[] = "\r\n\r\n";
if (!client.find(endOfHeaders))
{
  Serial.println(F("Invalid response"));
  return;
}
while (client.available() && client.peek() != '{')
{
```

```
  char c = 0;
  client.readBytes(&c, 1);
  Serial.print(c);
  Serial.println("BAD");
 }
 while (client.available()) {
  char c = 0;
  client.readBytes(&c, 1);
  Serial.print(c);
 }
}
```

Step 2:

- Connect ESP32 microcontroller.
- Fetching data from server after generating the IP address.
- Equating the API key.

NOTE:

I didn’t have microcontroller sir. After that I can’t generate the output sir.