```
//---------------------------------------Including the libraries.
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <MFRC522.h>
#include <WiFi.h>
#include <HTTPClient.h>
#include <vector>


#define SS_PIN  5
#define RST_PIN 4
#define BTN_PIN 15

const char* ssid = "Atchyuth Kalla";
const char* password = "00000000";




void http_Req(String str_modes, String str_uid);
String getValue(String data, char separator, int index);
int getUID();
void byteArray_to_string(byte array[], unsigned int len, char buffer[]);

String Web_App_URL = "https://script.google.com/macros/s/AKfycbyktSZdylSqJACPWeeeKbWfeBbXqXmpkMPmyH7wQJD1h5NPNOe2Ht_0rpC76OxJ9sA3oA/exec";

String reg_Info = "";
String atc_Info = "";
String atc_Name = "";
String atc_Date = "";
String atc_Time_In = "";
String atc_Time_Out = "";

int lcdColumns = 16;
int lcdRows = 2;

int readsuccess;
char str[32] = "";
String UID_Result = "--------";
String modes = "atc";

LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(115200);
  Serial.println();
  delay(1000);

  pinMode(BTN_PIN, INPUT_PULLUP);

  lcd.init();
  lcd.backlight();

  lcd.clear();

  delay(500);

  SPI.begin();
  mfrc522.PCD_Init();

  delay(500);


  lcd.setCursor( 0, 0);
```

```
    lcd.print("Google  Sheets");
    lcd.setCursor( 0, 1);
    lcd.print("AttendanceSystem");

    delay(3000);
    lcd.clear();

    Serial.println();
    Serial.println("-------------");
    Serial.println("WIFI mode : STA");
    WiFi.mode(WIFI_STA);
    Serial.println("-------------");

    Serial.println();
    Serial.println("------------");
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

    int connecting_process_timed_out = 20;
    connecting_process_timed_out = connecting_process_timed_out * 2;
    while (WiFi.status() != WL_CONNECTED) {
      Serial.print(".");
      lcd.setCursor( 0, 0);
      lcd.print("Connecting WIFI");
      delay(250);
      lcd.clear();
      delay(250);

      if (connecting_process_timed_out > 0) connecting_process_timed_out--;
      if (connecting_process_timed_out == 0) {
        delay(1000);
        ESP.restart();
      }
    }

    Serial.println();
    Serial.println("WiFi connected");
    Serial.println("------------");

    lcd.clear();
    lcd.setCursor( 0, 0);
    lcd.print("WiFi connected");
    delay(2000);
    lcd.clear();
    delay(500);

  }


  void loop() {
    int BTN_State = digitalRead(BTN_PIN);

    if (BTN_State == LOW) {
      lcd.clear();
      if (modes == "atc") {
        modes = "reg";
      } else if (modes == "reg") {
        modes = "atc";
      }
      delay(500);
    }

    readsuccess = getUID();

    if (modes == "atc") {

      lcd.setCursor(0, 0);
      lcd.print("ATTENDANCE");
```

```
      lcd.setCursor( 0, 1);
      lcd.print("Please tap card");



      if (readsuccess) {
        lcd.clear();
        delay(500);
        lcd.setCursor( 0, 0);
        lcd.print("Getting UID");
        lcd.setCursor( 0, 1);
        lcd.print("Successfully");
        delay(1000);

        http_Req(modes, UID_Result);
      }
    }

    if (modes == "reg") {
      lcd.setCursor( 0, 0);
      lcd.print("REGISTRATION");
      lcd.setCursor( 0, 1);
      lcd.print("Tap your card");

      if (readsuccess) {
        lcd.clear();
        delay(500);
        lcd.setCursor( 0, 0);
        lcd.print("Getting UID");
        lcd.setCursor( 0, 1);
        lcd.print("Successfully");

        delay(1000);

        http_Req(modes, UID_Result);
      }
    }

    delay(10);
  }


  void http_Req(String str_modes, String str_uid) {
    if (WiFi.status() == WL_CONNECTED) {
      String http_req_url = Web_App_URL + "?sts=" + str_modes + "&uid=" + str_uid;

      Serial.println();
      Serial.println("-------------");
      Serial.println("Sending request to Google Sheets...");
      Serial.print("URL : ");
      Serial.println(http_req_url);

      HTTPClient http;
      http.begin(http_req_url);

      int httpCode = http.GET();
      Serial.print("HTTP Status Code : ");
      Serial.println(httpCode);

      String payload;
      if (httpCode == 30) {
        String newUrl = http.header("Location");
        Serial.println("Redirecting to: " + newUrl);

        http.end(); // close current connection

        // follow the redirect
        http.begin(newUrl);
        httpCode = http.GET();
```

```
206         Serial.print("HTTP Status Code (after redirect) : ");
207         Serial.println(httpCode);
208       }
209
210       if (httpCode > 0) {
211         payload = http.getString();
212         Serial.println("Payload : " + payload);
213       }
214
215       Serial.println("-------------");
216       http.end();
217
218       // The rest of your code remains unchanged...
219       String sts_Res = getValue(payload, ',', 0);
220
221       if (sts_Res == "OK") {
222         if (str_modes == "atc") {
223           atc_Info = getValue(payload, ',', 1);
224
225           if (atc_Info == "TI_Successful" || atc_Info == "TO_Successful") {
226             atc_Name = getValue(payload, ',', 2);
227             atc_Date = getValue(payload, ',', 3);
228             atc_Time_In = getValue(payload, ',', 4);
229
230             if (atc_Info == "TO_Successful") {
231               atc_Time_Out = getValue(payload, ',', 5);
232             }
233
234             int name_Length = atc_Name.length();
235             int pos = 0;
236             if (name_Length > 0 && name_Length <= lcdColumns) {
237               pos = map(name_Length, 1, lcdColumns, 0, (lcdColumns / 2) - 1);
238               pos = ((lcdColumns / 2) - 1) - pos;
239             } else if (name_Length > lcdColumns) {
240               atc_Name = atc_Name.substring(0, lcdColumns);
241             }
242
243             lcd.clear();
244             delay(500);
245
246             lcd.setCursor( 0, 0);
247
248             lcd.print("IN: ");
249             lcd.print(atc_Time_In);
250             lcd.setCursor( 0, 1);
251             lcd.print("OUT: ");
252             if (atc_Info == "TO_Successful") {
253               lcd.print(atc_Time_Out);
254             }
255             delay(5000);
256             lcd.clear();
257             delay(500);
258           } else if (atc_Info == "atcInf01") {
259             lcd.clear();
260             delay(500);
261             lcd.setCursor( 0, 0);
262             lcd.print("Attendance");
263             lcd.setCursor( 0, 1);
264             lcd.print("Completed");
265             delay(5000);
266             lcd.clear();
267             delay(500);
268           } else if (atc_Info == "atcErr01") {
269             lcd.clear();
270             delay(500);
271             lcd.setCursor( 0, 0);
272             lcd.print("Error!");
273             lcd.setCursor( 8, 0);
274             lcd.print("Card not");
```

```
            lcd.setCursor( 0, 1);
            lcd.print("registered");
            delay(5000);
            lcd.clear();
            delay(500);
          }

          atc_Info = "";
          atc_Name = "";
          atc_Date = "";
          atc_Time_In = "";
          atc_Time_Out = "";
        }

        if (str_modes == "reg") {
          reg_Info = getValue(payload, ',', 1);

          if (reg_Info == "R_Successful") {
            lcd.clear();
            delay(500);
            lcd.setCursor( 0, 0);
            lcd.print("Registration");
            lcd.setCursor( 0, 1);
            lcd.print("Successful");
            delay(5000);
            lcd.clear();
            delay(500);
          } else if (reg_Info == "regErr01") {
            lcd.clear();
            delay(500);
            lcd.setCursor( 0, 0);
            lcd.print("Error!");
            lcd.setCursor( 8, 0);
            lcd.print("Already");
            lcd.setCursor( 0, 1);
            lcd.print("Registered");
            delay(5000);
            lcd.clear();
            delay(500);
          }

          reg_Info = "";
        }
      }
    } else {
      lcd.clear();
      delay(500);
      lcd.setCursor( 0, 0);
      lcd.print("Error!");
      lcd.setCursor( 0, 1);
      lcd.print("WiFi disconnected");
      delay(3000);
      lcd.clear();
      delay(500);
    }
  }

  String getValue(String data, char separator, int index) {
    int found = 0;
    int strIndex[] = { 0, -1 };
    int maxIndex = data.length() - 1;

    for (int i = 0; i <= maxIndex && found <= index; i++) {
      if (data.charAt(i) == separator || i == maxIndex) {
        found++;
        strIndex[0] = strIndex[1] + 1;
        strIndex[1] = (i == maxIndex) ? i + 1 : i;
      }
    }
```

```
344     return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
345   }
346   void byteArray_to_string(byte array[], unsigned int len, char buffer[]) {
347     for (unsigned int i = 0; i < len; i++) {
348       byte nib1 = (array[i] >> 4) & 0x0F;
349       byte nib2 = (array[i] >> 0) & 0x0F;
350       buffer[i * 2 + 0] = nib1 < 0xA ? '0' + nib1 : 'A' + nib1 - 0xA;
351       buffer[i * 2 + 1] = nib2 < 0xA ? '0' + nib2 : 'A' + nib2 - 0xA;
352     }
353     buffer[len * 2] = '\0';
354   }
355   int getUID() {
356     if (!mfrc522.PICC_IsNewCardPresent()) {
357       return 0;
358     }
359     if (!mfrc522.PICC_ReadCardSerial()) {
360       return 0;
361     }
362 
363     byteArray_to_string(mfrc522.uid.uidByte, mfrc522.uid.size, str);
364     UID_Result = str;
365 
366     mfrc522.PICC_HaltA();
367     mfrc522.PCD_StopCrypto1();
368 
369     return 1;
370     }
```