APPENDIX II : ARDUINO IDE CODE

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecure.h>
//----------------------------------------Include the DHT Library
#include "DHT.h"

//---------------------------------------
#define DHTPin 14
//-->Defines the type of DHT sensor used, in this project the sensor used is DHT22.
#define DHTTYPE DHT22
// -->Defining an On Board LED, used for indicators when the process of connecting to a Wi-Fi router
#define ON_Board_LED 2
// -->Defining an On Board LED, used for process indicator
#define ON_Board_pro_LED 16

DHT dht(DHTPin, DHTTYPE); //--> Initialize DHT sensor, DHT dht(Pin_used, Type_of_DHT_Sensor);

//--> Replace with your network credentials
const char* ssid = "Zombie!"; //--> Your Wi-Fi name or SSID.
const char* password = "Z79<7d73"; //--> Your Wi-Fi password.
//---------------------------------------

//--> REPLACE with your Domain name and URL path or IP address with path
```

```
const char* serverName = "http://iotespml.000webhostapp.com/post-esp-data.php";
//----------------------------------------Host & httpsPort
const char* host = "script.google.com";
const int httpsPort = 443;
//----------------------------------------
WiFiClientSecure client; //--> Create a WiFiClientSecure object.

//--> spreadsheet script ID
String GAS_ID = "AKfycbyiJmhklamogUkDRrKcIZWP80GHF5x1ZiZcI75HpSPVzlnKhJxn";
//----------------------------------------Digital fingerprint
const char* fingerprint = "46 B2 C3 44 9C 59 09 8B 01 B6 F8 BD 4C FB 00 74 91 2F EF F6";
//----------------------------------------
//================================================== void setup()
void setup() {
  Serial.begin(115200);
  delay(500);

  dht.begin();  //--> Start reading DHT22 sensor
  delay(500);

  WiFi.begin(ssid, password);
  Serial.println("");

  pinMode(ON_Board_LED,OUTPUT); //--> On Board LED port Direction output
  pinMode(ON_Board_pro_LED,OUTPUT); //--> On Board LED port Direction output
  digitalWrite(ON_Board_pro_LED, HIGH); //--> Turn off Led On Board
```

```
  digitalWrite(ON_Board_LED, HIGH); //--> Turn off Led On Board


  //----------------------------------------Wait for connection
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    //----------------------Make the On Board Flashing LED on the process of connecting to the Wi-Fi router.
    digitalWrite(ON_Board_LED, LOW);
    delay(500);
    digitalWrite(ON_Board_LED, HIGH);
    delay(500);
    //----------------------------------------
  }
  //----------------------------------------
 //--> Turn off the On Board LED when it is connected to the Wi-Fi router.
  digitalWrite(ON_Board_LED, HIGH);
  //-->If successfully connected to the Wi-Fi router,
 //-->the IP Address that will be visited is displayed in the serial monitor
  Serial.println("");
  Serial.print("Successfully connected to : ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  //----------------------------------------
  client.setInsecure();
}
```

```
//================================================================ void loop()
void loop() {

 // -->Reading temperature or humidity takes about 250 milliseconds!
 // -->Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
 delay(2000);

 //----------------------Make the On Board Flashing LED on the process of connecting to the wifi router.
 digitalWrite(ON_Board_pro_LED, LOW);
 //----------------------------------------

 int h = dht.readHumidity();
 int t = dht.readTemperature();  // Read temperature as Celsius (the default)

 // Check if any reads failed and exit early (to try again).
 if (isnan(h) || isnan(t) || t > 200 || h > 200) {
  Serial.println("Failed to read from DHT sensor !");
  delay(500);
  return;
 }
 String Temp = "Temperature : " + String(t) + " °C";
 String Hum = "Humidity : " + String(h) + " %";
 Serial.println(Temp);
 Serial.println(Hum);
```

```
  //--> Calls the sendData Subroutine for googlesheet
  sendData_to_mysql(t,h);

  //--> Calls the sendData Subroutine for googlesheet
  sendData_to_google(t, h);

  //----------------------------------------Make the On Board Flashing LED OFF the process.
  digitalWrite(ON_Board_pro_LED, HIGH);
  //----------------------------------------
  //Send an HTTP POST request every 2 seconds
  delay(1000);
}

//================================================================================
=========
// Subroutine for sending data to MySQL for website
void sendData_to_mysql(float temp, int hum){
  delay(250);
  String string_temperature =  String(temp);
  //String string_temperature =  String(temp, DEC);
  String string_humidity =  String(hum, DEC);
  String apiKeyValue = "tPmAT5Ab3j7F9";
  String sensorName = "DHT22";
  String sensorLocation = "My Room";

  //Check WiFi connection status
  if(WiFi.status()== WL_CONNECTED){
```

```
  HTTPClient http;

  // Your Domain name with URL path or IP address with path
  http.begin(serverName);

  // Specify content-type header
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");

  // Prepare your HTTP POST request data
  String httpRequestData = "api_key=" + apiKeyValue + "&sensor=" + sensorName + "&location=" +
sensorLocation   + "&value1=" + String(string_temperature) + "&value2=" + String(string_humidity) +
"";

  Serial.print("httpRequestData: ");
  Serial.println(httpRequestData);

  // Send HTTP POST request
  int httpResponseCode = http.POST(httpRequestData);

  if (httpResponseCode>0) {
     Serial.print("HTTP Response code: ");
     Serial.println(httpResponseCode);
   }
   else {
     Serial.print("Error code: ");
     Serial.println(httpResponseCode);
   }
```

```
    // Free resources
    http.end();
  }
  else {
    Serial.println("WiFi Disconnected");
  }
}
//================================================================================
=========
// Subroutine for sending data to Google Sheets
void sendData_to_google(float tem, int hum) {
  delay(250);
  Serial.println("==========");
  Serial.print("connecting to google");
  Serial.println(host);
  //----------------------------------------Connect to Google host
  if (!client.connect(host, httpsPort)) {
    Serial.println("connection failed");
    return;
  }
  //----------------------------------------
  //----------------------------------------Verify fingerprint
  if (client.verify(fingerprint, host)) {
    Serial.println("certificate matches");
  } else {
    Serial.println("certificate doesn't match");
  }
```

```
  //---------------------------------------

  //---------------------------------------Processing data and sending data
  String string_temperature  =  String(tem);
  String string_humidity  =  String(hum, DEC);
  String url = "/macros/s/" + GAS_ID + "/exec?temperature=" + string_temperature + "&humidity=" +
string_humidity;
  Serial.print("requesting URL: ");
  Serial.println(url);

  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
       "Host: " + host + "\r\n" +
       "User-Agent: BuildFailureDetectorESP8266\r\n" +
       "Connection: close\r\n\r\n");

  Serial.println("request sent");
  //---------------------------------------

  //---------------------------------------Checking whether the data was sent successfully or not
  while (client.connected()) {
    String line = client.readStringUntil('\n');
    if (line == "\r") {
      Serial.println("headers received");
      break;
    }
  }
  String line = client.readStringUntil('\n');
```

```
  if (line.startsWith("{\"state\":\"success\"")) {
    Serial.println("esp8266/Arduino CI successfull!");
  } else {
    Serial.println("esp8266/Arduino CI has failed");
  }
  Serial.print("reply was : ");
  Serial.println(line);
  Serial.println("closing connection");
  Serial.println("==========");
  Serial.println();
  //----------------------------------------
}
//=======================================================================================
```

APPENDIX III : WEBSITE CODE – I (DESIGN)

```
<?php
$servername = "localhost";
$username = "id17789025_espml8266";
$password = "s#G)dtu*j|y0tRMZ";
$dbname = "id17789025_espml";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, value1, value2 FROM SensorData";
/*select items to display from the sensordata table in the data base*/

if ($result = $conn->query($sql)) {

    /* select all the weekly tasks from the table googlechart */
    $rows_temp = array();
    $table_temp = array();
    $table_temp['cols'] = array(

    // Labels for your chart, these represent the column titles.
```

```
array('label' => 'id', 'type' => 'number'),
array('label' => 'Temperature', 'type' => 'number')
);

$rows_hum = array();
$table_hum = array();
$table_hum['cols'] = array(

// Labels for your chart, these represent the column titles.
array('label' => 'id', 'type' => 'number'),
array('label' => 'Humidity', 'type' => 'number')
);

/* Extract the information from $result */
foreach($result as $r) {

  $temp = array();
  $hum = array();

// The following line will be used to slice the line chart
  $temp[] = array('v' => (int) $r['id']);
  $hum[] = array('v' => (int) $r['id']);


  // Values of the each slice
  $temp[] = array('v' => (int) $r['value1']);
```

```
    $hum[] = array('v' => (int) $r['value2']);

    $rows_temp[] = array('c' => $temp);
    $rows_hum[] = array('c' => $hum);
  }

  $table_temp['rows'] = $rows_temp;
  $table_hum['rows'] = $rows_hum;

  // convert data into JSON format
  $jsonTable_temp = json_encode($table_temp);
  $jsonTable_hum = json_encode($table_hum);

  //echo $jsonTable;
  }

 $conn->close();
?>

<html>
 <head>
  <meta http-equiv="refresh" content="5" >
        <title> Sensor Data </title>
        <style>
     table {
       border-collapse: collapse;
```

```
      width: 90%;
    }

    th, td {
      text-align: center;
      padding: 8px;
    }

    tr:nth-child(even){background-color: #f2f2f2}

    th {
      background-color: #04AA6D;
      color: white;
    }
  </style>

  <!--Load the Ajax API-->
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
  <script type="text/javascript">


  // Load the Visualization API and the piechart package.
  google.load('visualization', '1', {'packages':['corechart']});

  // Set a callback to run when the Google Visualization API is loaded.
```

```
google.setOnLoadCallback(drawChart_temp);
google.setOnLoadCallback(drawChart_hum);

function drawChart_temp() {
  // Create our data table out of JSON data loaded from server.
  var data = new google.visualization.DataTable(<?=$jsonTable_temp?>);
  var options = {
     title: 'Temperature Data',
     borderColor: "red",
     hAxis: {title: 'Serial Number'},
     vAxis: {title: 'Temperature ( Celcius )'},
     width: 650,
     height: 300,
     legend: 'none'
    };
  // Instantiate and draw our chart, passing in some options.
  // Do not forget to check your div ID
  var chart = new google.visualization.LineChart(document.getElementById('temp_chart'));
  chart.draw(data, options);
}

function drawChart_hum() {
  // Create our data table out of JSON data loaded from server.
  var data = new google.visualization.DataTable(<?=$jsonTable_hum?>);
  var options = {
     title: 'Humidity Data',
```

```
      borderColor: "blue",
      hAxis: {title: 'Serial Number'},
      vAxis: {title: 'Humidity ( % )'},
      width: 650,
      height: 300,
      legend: 'none'
    };
   // Instantiate and draw our chart, passing in some options.
   // Do not forget to check your div ID
   var chart = new google.visualization.LineChart(document.getElementById('hum_chart'));
   chart.draw(data, options);
 }
 </script>
</head>

<body style="background-color:#FAF0DC; text-align:center">
 <h1>SENSOR DATA</h1>
 <table class="columns">
    <!--this is the div that will hold the line chart-->
    <tr>
       <td><div id="temp_chart" style="border: 1px solid #ccc"></div></td>
       <td><div id="hum_chart" style="border: 1px solid #ccc"></div></td>
    </tr>
 </table>
```

```
<?php
$servername = "localhost";
$username = "id17789025_espml8266";
$password = "s#G)dtu*j|y0tRMZ";
$dbname = "id17789025_espml";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, sensor, location, value1, value2, reading_time FROM SensorData ORDER BY id ASC";
/*select items to display from the sensordata table in the data base*/
echo '<table cellspacing="10" cellpadding="7" style="border: 1px solid black; margin-left:auto; margin-right:auto;">
      <tr>
        <th>ID</th>
        <th>Date & Time</th>
        <th>Sensor</thh>
        <th>Location</th>
        <th>Temperature ( &deg;C )</th>
        <th>Humidity ( &#37; )</th>
       <!-- <th>Pressure</th> -->
      </tr>';
```

```
if ($result = $conn->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        $row_id = $row["id"];
        $row_reading_time = $row["reading_time"];
        $row_sensor = $row["sensor"];
        $row_location = $row["location"];
        $row_value1 = $row["value1"];
        $row_value2 = $row["value2"];
        // timezone to - 5 hour (you can change 1 to any number)
        $row_reading_time = date("m-d-Y H:i:s", strtotime("$row_reading_time - 5 hours"));
        echo '<tr>
                <td>' . $row_id . '</td>
                <td>' . $row_reading_time . '</td>
                <td>' . $row_sensor . '</td>
                <td>' . $row_location . '</td>
                <td>' . $row_value1 . ' &deg;C</td>
                <td>' . $row_value2 . ' &#37;</td>
              </tr>';
    }
    $result->free();
}
$conn->close();
?>
  </body>
</html>
```

APPENDIX IV : WEBSITE CODE (POST DATA)

```
<?php
$servername = "localhost";
$username = "id17789025_espml8266";
$password = "s#G)dtu*j|y0tRMZ";
$dbname = "id17789025_espml";

// Keep this API Key value to be compatible with the ESP32 code provided in the project page.
// If you change this value, the ESP32 sketch needs to match
$api_key_value = "tPmAT5Ab3j7F9";

$api_key= $sensor = $location = $value1 = $value2 = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $api_key = test_input($_POST["api_key"]);
    if($api_key == $api_key_value) {
        $sensor = test_input($_POST["sensor"]);
        $location = test_input($_POST["location"]);
        $value1 = test_input($_POST["value1"]);
        $value2 = test_input($_POST["value2"]);

        // Create connection
        $conn = new mysqli($servername, $username, $password, $dbname);
        // Check connection
```

```
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        $sql = "INSERT INTO SensorData (sensor, location, value1, value2)
        VALUES ('" . $sensor . "', '" . $location . "', '" . $value1 . "', '" . $value2 . "')";

        if ($conn->query($sql) === TRUE) {
            echo "New record created successfully";
        }
        else {
            echo "Error: " . $sql . "<br>" . $conn->error;
        }

        $conn->close();
    }
    else {
        echo "Wrong API Key provided.";
    }

}
else {
    echo "No data posted with HTTP POST.";
}
```

```
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
```