

# *Epic Global School*

## *[School Automation System]*

### Abstract

The School Automation System aims to integrate smart technologies into educational institutions for enhanced functionality and efficiency. This system includes features such as RFID-based attendance tracking, automatic lighting control, and garden pump automation using soil moisture sensors. The project leverages ESP8266 microcontrollers to enable remote operations through smartphones.

---

## 1. Introduction

- **Objective:** To create a cost-effective and efficient automation system for schools to improve daily operations and reduce manual intervention.
  - **Scope:**
    - Automating attendance using RFID cards.
    - Automatic and manual control of school corridor and classroom lights.
    - Garden pump automation based on soil moisture levels, with manual override via smartphone.
- 

## 2. System Design

### 2.1 Components Used

1. **Microcontroller:** ESP8266 for remote control and connectivity.
2. **RFID System:** For attendance tracking and real-time display.
3. **Sensors:**
  - Soil Moisture Sensor for garden automation.
  - Light sensors (if applicable) for automatic mode in classrooms.
4. **Actuators:**
  - Relays for switching lights and pumps.
5. **Display:** LCD/LED screens for showing attendance.
6. **Power Supply:** 5V DC for microcontrollers and sensors.

## 2.2 System Architecture

- **RFID Attendance:**
    - RFID cards are scanned, and the data is transmitted to a server/display for real-time attendance tracking.
  - **Light Control:**
    - **Automatic Mode:** Lights are controlled via a smartphone app using the ESP8266 module.
    - **Manual Override:** Lights can also be manually turned on/off.
  - **Garden Pump:**
    - Soil moisture below a set threshold activates the pump.
    - Users can also activate the pump manually via the smartphone app.
- 

## 3. Implementation

### 3.1 RFID-Based Attendance System

- Setup an RFID reader at the entrance of classrooms or corridors.
- Integrate the reader with a microcontroller to process attendance data.
- Display attendance on screens in real-time.

### 3.2 Light Control System

- Connect ESP8266 modules to lights via relays.
- Develop a mobile application interface for remote control.
- Configure automatic light modes triggered by specific conditions.

### 3.3 Garden Pump Automation

- Install soil moisture sensors in the garden.
  - Program the system to monitor moisture levels and activate the pump when levels fall below the threshold.
  - Integrate manual pump control through the smartphone app.
- 

## 4. Results and Discussion

- **Efficiency:**
  - Automated processes reduced manual work.
  - Attendance tracking became faster and more accurate.

- **Cost Effectiveness:** The system utilized affordable components and minimized energy waste.
  - **User Experience:** Teachers and staff reported ease of use and convenience.
  - **Limitations:**
    - Internet dependency for remote control.
    - RFID range and interference issues.
- 

## 5. Conclusion

The School Automation System successfully demonstrated the integration of smart technologies to automate routine tasks in schools. The combination of RFID, ESP8266, and sensors offered significant improvements in attendance management, energy conservation, and garden maintenance.

---

## 6. Future Work

- Expand the RFID system for visitor tracking.
  - Incorporate motion sensors for enhanced light control.
  - Develop a more robust app interface for easier usability.
  - Explore renewable energy options to power the system.
- 

## 7. References

- Include datasheets for RFID readers, ESP8266, and sensors used.
  - Cite tutorials and resources for ESP8266 programming and app development.
- 

## Project Team Member Name:

1: Parul Bajpai { Team Leader }

2: Naman Pratap Singh { Team Leader }

3: Akshay Bajpai { Team Leader , Key Contributor }

4: Aishwarya Mishra { Team Leader }

5: Vaibhavi Singh { Team Leader }

6: Dileep Gupta

7: Suryansh Singh

8: Ritesh Singh

9: Anushu Verma

10: Anmolpreet Singh

11: Yash Shukal

12: Siddharth Shukla

13: Parth Shukla

14: Aradya Gupta

15: Tejaswini

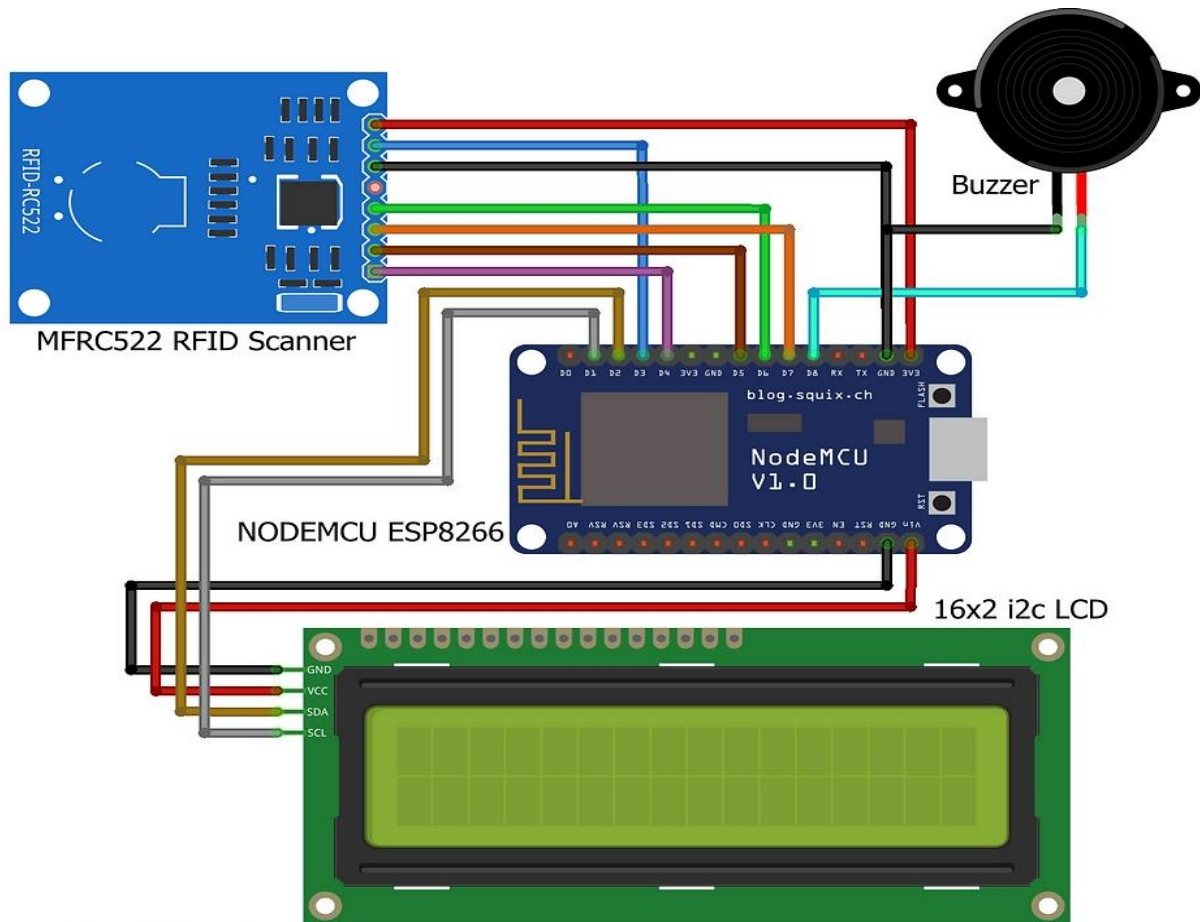
17: Annavi Patel

16: Anjana Mishra

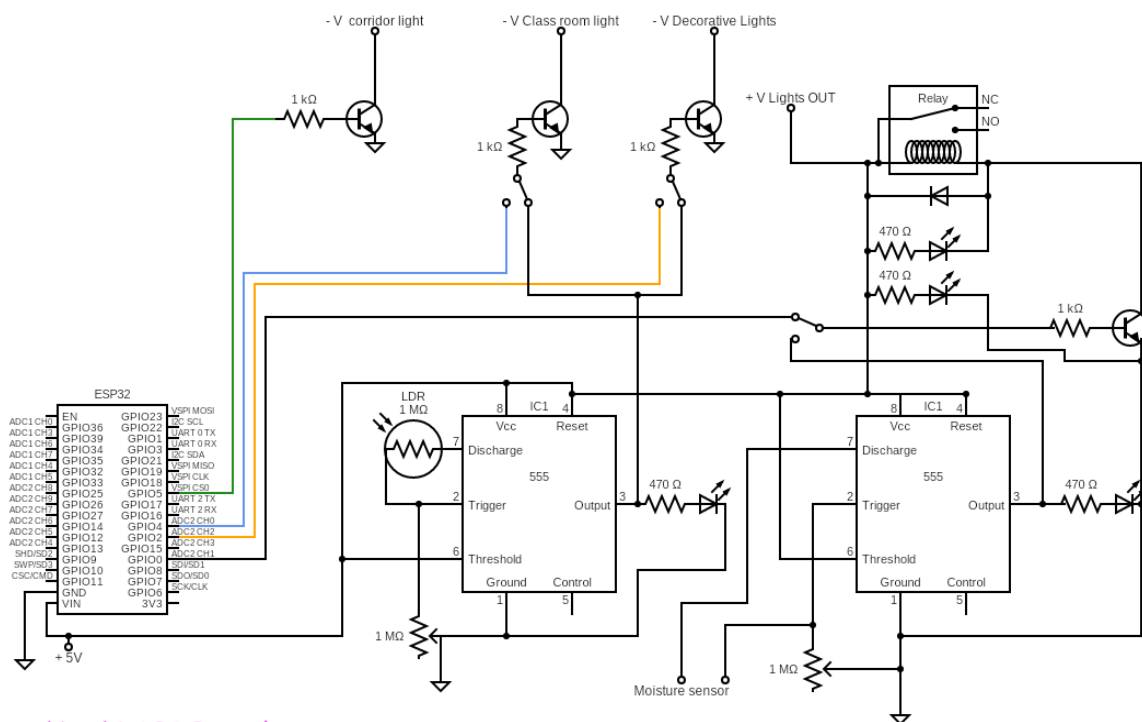
18: Priya

19: Anuriya

## Circuit Diagrams :



anubhavb4123@gmail.com



anubhavb4123@gmail.com

## ***Code Snippets for esp8266***

```
#include <ESP8266WiFi.h>
```

```
// Define GPIO pins for the lights (adjust if necessary  
for your ESP8266 board)
```

```
const int light1 = D1; // GPIO5
```

```
const int light2 = D2; // GPIO4
```

```
const int light3 = D3; // GPIO0
```

```
const int light4 = D4; // GPIO2
```

```
// Set up the Access Point credentials
```

```
const char* ssid = "#Epic Global School_Control";
```

```
const char* password = "22446688"; // Minimum 8  
characters
```

```
// Create an instance of the WiFi server on port 80
```

```
WiFiServer server(80);
```

```
// HTML content for the control page
```

```
const char webpage[] PROGMEM = R"rawliteral(
<!DOCTYPE html>
<html>
<head>
  <title>ESP8266 Light Control</title>
  <meta name="viewport" content="width=device-
width, initial-scale=1">
  <style>
    button {
      font-size: 24px;
      margin: 10px;
      padding: 20px;
      color: DodgerBlue;
    }
    h1 {
      color: Tomato;
    }
    h2 {
      color: orange;
    }
  </style>
</head>
<body>
  <div>
    <h1>Light Control</h1>
    <h2>ESP8266</h2>
    <div>
      <button>Turn On</button>
      <button>Turn Off</button>
    </div>
  </div>
</body>
</html>
)
```

```
</style>
</head>
<body>
  <h1>#Epic Global School {Controls}:</h1>
  <button onclick="location.href='/on1'">Corridor
Light ON</button>
  <button onclick="location.href='/off1'">Corridor
Light OFF</button><br>
  <button onclick="location.href='/on2'">Class Room
Light ON</button>
  <button onclick="location.href='/off2'">Class Room
Light OFF</button><br>
  <button onclick="location.href='/on3'">Water Pump
ON</button>
  <button onclick="location.href='/off3'">Water Pump
OFF</button><br>
  <button onclick="location.href='/on4'">Decorative
Lights ON</button>
  <button onclick="location.href='/off4'">Decorative
Lights OFF</button>
  <h1>Project Team Member Name:</h1>
```



<h2 style="color:blue;">1: Parul Bajpai { Team Leader }</h2>

<h2 style="color:blue;">2: Naman Pratap Singh { Team Leader }</h2>

<h2 style="color:blue;"> 3: Akshay Bajpai { Team Leader , Key Contributor } </h2>

<h2 style="color:blue;">4: Aishwarya Mishra { Team Leader }</h2>

<h2 style="color:blue;">5: Vaibhavi Singh { Team Leader }</h2>

<h2>6: Dileep Gupta </h2>

<h2>7: Suryansh Singh</h2>

<h2>8: Ritesh Singh</h2>

<h2>9: Anushu Verma</h2>

<h2>10: Anmolpreet Singh </h2>

<h2>11: Yash Shukal</h2>

<h2>12: Siddharth Shukla</h2>

<h2>13: Parth Shukla</h2>

<h2>14: Aradya Gupta</h2>

<h2>15: Tejaswini </h2>

<h2>16: Anjana Mishra</h2>

<h2>17: Annavi Patel</h2>

<h2>18: Priya </h2>

<h2>19: Anuriya</h2>

<h2

style="color:red;">.....

.....END.....</h2>

</body>

</html>

)rawliteral";

void setup() {

Serial.begin(115200);

// Initialize GPIO pins as OUTPUT

pinMode(light1, OUTPUT);

pinMode(light2, OUTPUT);

pinMode(light3, OUTPUT);

pinMode(light4, OUTPUT);

// Turn off all lights initially

digitalWrite(light1, LOW);

```
digitalWrite(light2, LOW);  
digitalWrite(light3, LOW);  
digitalWrite(light4, LOW);
```

```
// Start the Access Point  
WiFi.softAP(ssid, password);  
Serial.println("Access Point started!");  
Serial.print("IP Address: ");  
Serial.println(WiFi.softAPIP());
```

```
// Start the server  
server.begin();  
}
```

```
void loop() {  
  // Check if a client is connected  
  WiFiClient client = server.available();  
  if (!client) {  
    return;  
  }
```

```
// Wait for the client to send a request
String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();
```

```
// Handle the requests to control lights
if (request.indexOf("/on1") != -1) {
    digitalWrite(light1, HIGH);
} else if (request.indexOf("/off1") != -1) {
    digitalWrite(light1, LOW);
}

if (request.indexOf("/on2") != -1) {
    digitalWrite(light2, HIGH);
} else if (request.indexOf("/off2") != -1) {
    digitalWrite(light2, LOW);
}

if (request.indexOf("/on3") != -1) {
    digitalWrite(light3, HIGH);
} else if (request.indexOf("/off3") != -1) {
```

```
    digitalWrite(light3, LOW);
}
if (request.indexOf("/on4") != -1) {
    digitalWrite(light4, HIGH);
} else if (request.indexOf("/off4") != -1) {
    digitalWrite(light4, LOW);
}

// Send the response
client.print("HTTP/1.1 200 OK\r\nContent-Type:
text/html\r\n\r\n");
client.print(webpage);

// Close the connection
client.stop();
}
```

## ***Code Snippets for Arduino nano that controls NFC***

```
#include <SPI.h>
```

```
#include <MFRC522.h>
```

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
#define SS_PIN 10
```

```
#define RST_PIN 9
```

```
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create  
MFRC522 instance
```

```
// LED pins
```

```
#define GREEN_LED_PIN 3
```

```
#define RED_LED_PIN 4
```

```
// Valid UIDs of students
```

```
byte validUIDs[][4] = {
```

```
    {0x73, 0x6A, 0x4C, 0x28}, // UID of student 1
```

```
{0xCE, 0x56, 0x4C, 0x22}, // UID of student 2
{0xDE, 0xB2, 0xA2, 0xA1}, // UID of student 3
{0xFE, 0xFE, 0x36, 0x22}, // UID of student 4
};
```

```
// Student names
```

```
const char* studentNames[] = {
    "Anubhav Bajpai", // Name of student 1
    "Akshay Bajpai", // Name of student 2
    "Vaibhavi Singh", // Name of student 3
    "Aishwarya Mishra" // Name of student 4
};
```

```
// LCD I2C display
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2); // Adjust the
address and size according to your display
```

```
void setup() {
```

```
    Serial.begin(9600);    // Initialize serial
communication
```

```
    SPI.begin();          // Initiate SPI bus
```

```
mfr522.PCD_Init();    // Initiate MFRC522
lcd.init();           // Initialize the LCD display
lcd.backlight();      // Turn on the backlight
lcd.clear();
lcd.setCursor(2, 0);
lcd.print("Scan ID card");
Serial.println("Ready to read RFID cards");

// Set LED pins as output
pinMode(GREEN_LED_PIN, OUTPUT);
pinMode(RED_LED_PIN, OUTPUT);

// Turn off both LEDs initially
digitalWrite(GREEN_LED_PIN, LOW);
digitalWrite(RED_LED_PIN, LOW);
}

void loop() {
    // Look for new cards
    if (mfr522.PICC_IsNewCardPresent()) {
```



```
if (mfrc522.PICC_ReadCardSerial()) {  
    // Show UID on serial monitor  
    Serial.print("UID tag: ");  
    for (byte i = 0; i < mfrc522.uid.size; i++) {  
        Serial.print("0x");  
        if (mfrc522.uid.uidByte[i] < 0x10) Serial.print("0");  
        Serial.print(mfrc522.uid.uidByte[i], HEX);  
        if (i < mfrc522.uid.size - 1) Serial.print(", ");  
    }  
    Serial.println();  
    Serial.print("UID Number: ");  
    String content = "";  
    for (byte i = 0; i < mfrc522.uid.size; i++) {  
        content.concat(String(mfrc522.uid.uidByte[i] <  
0x10 ? "0" : ""));  
        content.concat(String(mfrc522.uid.uidByte[i],  
HEX));  
    }  
    content.toUpperCase();  
    Serial.println(content);  
}
```

```
// Check if the UID matches any of the valid UIDs
bool uidMatched = false;
int studentIndex = -1;

for (int i = 0; i < sizeof(validUIDs) /
sizeof(validUIDs[0]); i++) {

    if (memcmp(mfrc522.uid.uidByte, validUIDs[i],
mfrc522.uid.size) == 0) {

        uidMatched = true;
        studentIndex = i;
        break;
    }
}

// Perform actions based on UID match
if (uidMatched) {











    digitalWrite(GREEN_LED_PIN, HIGH);
    digitalWrite(RED_LED_PIN, LOW);
    updateAttendance(studentIndex + 1, "Present");
    delay(3000); // Keep the green LED on for 3
seconds











    digitalWrite(GREEN_LED_PIN, LOW);
```












```
} else {  
    digitalWrite(GREEN_LED_PIN, LOW);  
    digitalWrite(RED_LED_PIN, HIGH);  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("ID doesn't match");  
    delay(2000); // Keep the red LED on for 2 seconds  
    digitalWrite(RED_LED_PIN, LOW);  
}  
lcd.clear();  
lcd.setCursor(2, 0);  
lcd.print("Scan ID card");  
  
    delay(1000);  
}  
mfrc522.PICC_HaltA(); // Stop reading  
mfrc522.PCD_StopCrypto1(); // Stop encryption on  
PCD  
}  
}
```

```
// Function to update attendance in the database
void updateAttendance(int studentID, const char*
status) {
    // Display the name of the student on the LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(studentNames[studentID - 1]); // -1 to adjust
for array indexing
    lcd.setCursor(0, 1); // Set cursor to the second line
    lcd.print("~");
    lcd.setCursor(2, 1); // Set cursor position after 4 pixels
    lcd.print(status);
}
```

*Component List :-*

ITEMS	HSN CODE	UNIT PRICE	TAX	QTY	ITEM TOTAL
 <div>           ESP8266 NodeMCU CP2102 WiFi Development Board            SKU: QC0167         </div>	8542	<div>Rs.</div> <div><del>232.00</del></div> <div>Rs.</div> <div>220.40</div>	Rs. 39.66	1	Rs. 220.40
 <div>           Nano R3 CH340 Chip Development Board - Compatible with Arduino - Soldered (Without Cable)            SKU: QC2027         </div>	8542	<div>Rs.</div> <div><del>186.00</del></div> <div>Rs.</div> <div>176.70</div>	Rs. 31.80	1	Rs. 176.70
 <div>           Female to Female Connecting Wires / Jumper Wires (Set of 10)            SKU: QC0408         </div>	8542	<div>Rs. 14.00</div> <div>Rs. 13.30</div>	Rs. 2.38	1	Rs. 13.30
 <div>           Male to Female connecting wires / Jumper wires (set of 10)            SKU: QC0407         </div>	8542	<div>Rs. 15.00</div> <div>Rs. 14.25</div>	Rs. 2.56	1	Rs. 14.25
 <div>           Male to Male connecting wires / Jumper wires (set of 10)            SKU: QC0389         </div>	8542	<div>Rs. 15.00</div> <div>Rs. 14.25</div>	Rs. 2.56	1	Rs. 14.25
 <div>           Arduino Nano Cable (Blue/Black)            SKU: QC0092         </div>	8542	<div>Rs. 30.00</div> <div>Rs. 28.50</div>	Rs. 5.13	1	Rs. 28.50
 <div>           General Purpose Universal PCB for Prototype (15x10)cm            SKU: QC0425         </div>	8542	<div>Rs. 25.00</div> <div>Rs. 23.75</div>	Rs. 4.28	1	Rs. 23.75
 <div>           40 Pin Straight Male Header Pins - Berg Strips            SKU: QC0293         </div>	8542	<div>Rs. 8.00</div> <div>Rs. 7.60</div>	Rs. 1.37	1	Rs. 7.60
 <div>           40 Pin Straight Female Header Pins - Berg Strips            SKU: QC0294         </div>	8542	<div>Rs. 9.00</div> <div>Rs. 8.55</div>	Rs. 1.54	1	Rs. 8.55
 <div>           USB to micro-USB Cable for NodeMCU and Raspberry Pi            SKU: QC0009         </div>	8542	<div>Rs. 50.00</div> <div>Rs. 30.00</div>	Rs. 5.40	1	Rs. 30.00

	2x3 inch Single Side Copper Plate Perf Board for PCB Prototype / Dotted Board / General Purpose PCB / Zero PCB SKU: QC0580	8542	<del>Rs. 13.00</del> Rs. 12.35	Rs. 2.23	2	Rs. 24.70
	BC547 NPN Transistor SKU: QC0368	8542	<del>Rs. 3.00</del> Rs. 2.85	Rs. 0.51	5	Rs. 14.25
	1K ohm, 1/4 Watt Resistor with 1% tolerance (Pack of 10) SKU: QC0626	8542	<del>Rs. 8.00</del> Rs. 7.60	Rs. 1.37	1	Rs. 7.60
	330 ohm, 1/4 Watt Resistor with 1% tolerance (Pack of 10) SKU: QC0622	8542	<del>Rs. 8.00</del> Rs. 7.60	Rs. 1.37	1	Rs. 7.60
	LDR Sensor (Light Dependent Resistor) - 5mm SKU: QC0273	8542	<del>Rs. 5.00</del> Rs. 4.75	Rs. 0.86	4	Rs. 19.00
	555 Timer IC (LM555 / NE555 / SE555) SKU: QC0065	8542	<del>Rs. 8.00</del> Rs. 7.60	Rs. 1.37	2	Rs. 15.20
	1M Ohm Trimpot Potentiometer (Variable Resistor) SKU: QC0590	8542	<del>Rs. 7.00</del> Rs. 6.65	Rs. 1.20	1	Rs. 6.65
	470 ohm, 1/4 Watt Resistor with 5% tolerance (Pack of 10) SKU: QC0180	8542	<del>Rs. 5.00</del> Rs. 4.75	Rs. 0.86	1	Rs. 4.75
	5V 600mA DC-DC Step-up Booster Module for single cell battery SKU: QC0530	8542	<del>Rs. 32.00</del> Rs. 30.40	Rs. 5.47	1	Rs. 30.40
	SPDT Miniature Slide Switch SKU: QC0328	8542	<del>Rs. 7.00</del> Rs. 6.65	Rs. 1.20	3	Rs. 19.95
				Discount SAVE5	-Rs. 72.85	

	Small Breadboard / Mini Solderless Board - 400 points SKU: QC0441	8542	<del>Rs. 41.00</del> Rs. 38.95	Rs. 7.01	2	Rs. 77.90
	16x2 (1602) LCD Display (Blue Backlight) SKU: QC0140	8542	<del>Rs. 100.00</del> Rs. 95.00	Rs. 17.10	1	Rs. 95.00
	I2C Serial Interface LCD Adapter Module SKU: QC0142	8542	<del>Rs. 47.00</del> Rs. 44.65	Rs. 8.04	1	Rs. 44.65
	TP4056 Battery Charging Protection Module (Type C) SKU: QC0875	8542	<del>Rs. 21.00</del> Rs. 19.95	Rs. 3.59	1	Rs. 19.95
	18650 Li-ion 2500mAh Rechargeable Battery Copy SKU: QC0763	8542	<del>Rs. 63.00</del> Rs. 59.85	Rs. 10.78	2	Rs. 119.70
	36AWG Multi Strand Wire 14/36 (Green - 1mtr) SKU: QC0900	8542	<del>Rs. 8.00</del> Rs. 7.60	Rs. 1.37	3	Rs. 22.80
	36AWG Multi Strand Wire 7/0.0052 (Blue - 1mtr) SKU: QC0898	8542	<del>Rs. 8.00</del> Rs. 7.60	Rs. 1.37	3	Rs. 22.80
	36AWG Multi Strand Wire 14/36 (Gray - 1mtr) SKU: QC0901	8542	<del>Rs. 5.00</del> Rs. 4.75	Rs. 0.86	2	Rs. 9.50
	36AWG Multi Strand Wire 14/36 (Black - 1mtr) SKU: QC0902	8542	<del>Rs. 8.00</del> Rs. 7.60	Rs. 1.37	2	Rs. 15.20
	RC522 RFID 13.56MHZ Reader Writer Module / MFRC-522 reader with tag SKU: QC0400	8542	<del>Rs. 92.00</del> Rs. 87.40	Rs. 15.73	1	Rs. 87.40
	RFID Card / RFID Tag (13.56MHz RF range) SKU: QC0010	8542	<del>Rs. 14.00</del> Rs. 13.30	Rs. 2.39	3	Rs. 39.90

[anubhavb4123@gmail.com](mailto:anubhavb4123@gmail.com)