

EXERCISE:8

SECURED COMMUNICATION FROM MICROCONTROLLER TO BACKEND

PHP CODE

```
<?php
class lpg {
    public $link = "";
    function __construct($lpgConcentration) {
        $this->connect();
        $this->storeInDB($lpgConcentration);
    }
    function connect() {
        // Connecting to the database
        $this->link = mysqli_connect('localhost', 'root', '') or die('Cannot connect to the DB');
        mysqli_select_db($this->link, 'lpg') or die('Cannot select the DB'); // Your database is 'lpg'
    }
    function storeInDB($lpgConcentration) {
        // Insert LPG data into the database
        $query = "INSERT INTO sensor (value) VALUES ('" . $lpgConcentration . "')";
        $result = mysqli_query($this->link, $query) or die('Errant query: ' . $query);
    }
}

// Check if LPG concentration is passed in the URL
if ($_GET['lpgConcentration'] != '') {
    // Create new lpg object and store data
    $lpg = new lpg($_GET['lpgConcentration']);
}
?>
```

ARDUINO CODE:

```
#include <SPI.h>

#include <Ethernet.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // Setting MAC Address

// Pin for the LPG sensor (Assume it's connected to A0)

#define LPG_PIN A0

char server[] = "172.16.8.25"; // Server IP

IPAddress ip(172,16,0,0);

EthernetClient client;

/* Setup for Ethernet and RFID */

void setup() {

  Serial.begin(9600);

  if (Ethernet.begin(mac) == 0) {

    Serial.println("Failed to configure Ethernet using DHCP");

    Ethernet.begin(mac, ip);

  }

  delay(1000);

}

/* Infinite Loop */

void loop() {

  int lpgSensorValue = analogRead(LPG_PIN); // Read LPG sensor value

  float lpgConcentration = map(lpgSensorValue, 0, 1023, 0, 100); // Mapping the sensor value to a
percentage (optional)

  Sending_To_phpmyadmindatabase(lpgConcentration); // Sending LPG data to database

  delay(3000); // Interval

}

void Sending_To_phpmyadmindatabase(float lpgConcentration) { // Connecting with MySQL

  if (client.connect(server, 80)) {

    Serial.println("connected");

    // Make a HTTP request:
```

```

Serial.print("GET /testcode/lpg_sensor.php?lpgConcentration=");

client.print("GET /testcode/lpg_sensor.php?lpgConcentration="); // Your URL

Serial.println(lpgConcentration);

client.print(lpgConcentration);

client.print(" "); // SPACE BEFORE HTTP/1.1

client.print("HTTP/1.1");

client.println();

client.println("Host: 172.16.10.148");

client.println("Connection: close");

client.println();

} else {

    // If connection to the server failed:

    Serial.println("connection failed");

}

}

```

ARDUINO OUTPUT:

```

sketch_sep27a.ino
27 float lpgConcentration = map(lpgSensorValue, 0, 1023, 0, 100); // Mapping the sensor value to a percentage (optional)
28
29 Sending_To_phpmyadmindatabase(lpgConcentration); // Sending LPG data to database
30 delay(3000); // Interval
31 }
32
33 void Sending_To_phpmyadmindatabase(float lpgConcentration) { // Connecting with MySQL

```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Mega or Mega 2560 on COM3')

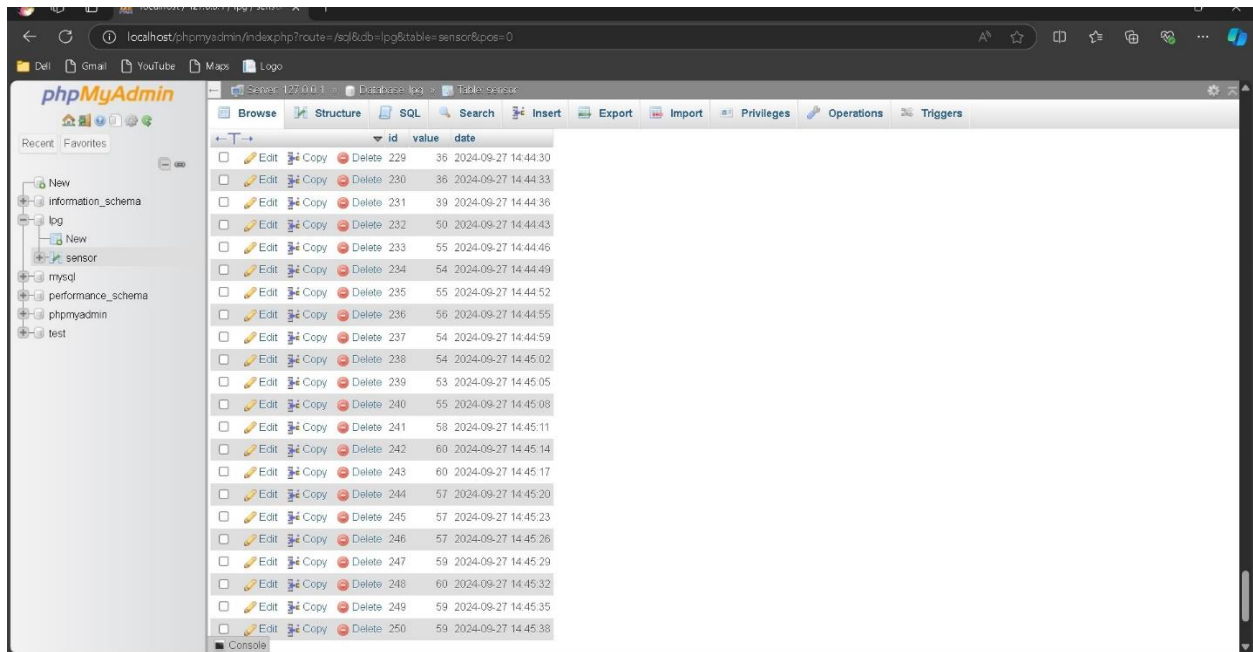
```

connected
GET /testcode/lpg_sensor.php?lpgConcentration=37.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=37.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=37.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=37.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=37.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=37.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=37.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=37.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=37.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=37.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=36.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=36.00
connected
GET /testcode/lpg_sensor.php?lpgConcentration=36.00

```

Ln 41, Col 48 Arduino Mega or Mega 2560 on COM3

DATABASE OUTPUT:



The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database structure with folders for 'information_schema', 'log', 'mysql', 'performance_schema', 'phpmyadmin', and 'test'. The 'log' folder is expanded, showing a 'sensor' table. The main panel shows the 'sensor' table with columns 'id', 'value', and 'date'. The table contains 25 rows of data, each with a checkbox, 'Edit', 'Copy', and 'Delete' icons, and a 'value' column. The 'date' column shows timestamps from 2024-09-27 14:44:30 to 2024-09-27 14:45:38. The bottom of the interface shows a 'Console' tab.

	id	value	date
<input type="checkbox"/>	229	35	2024-09-27 14:44:30
<input type="checkbox"/>	230	36	2024-09-27 14:44:33
<input type="checkbox"/>	231	39	2024-09-27 14:44:36
<input type="checkbox"/>	232	50	2024-09-27 14:44:43
<input type="checkbox"/>	233	55	2024-09-27 14:44:46
<input type="checkbox"/>	234	54	2024-09-27 14:44:49
<input type="checkbox"/>	235	55	2024-09-27 14:44:52
<input type="checkbox"/>	236	56	2024-09-27 14:44:55
<input type="checkbox"/>	237	54	2024-09-27 14:44:59
<input type="checkbox"/>	238	54	2024-09-27 14:45:02
<input type="checkbox"/>	239	53	2024-09-27 14:45:05
<input type="checkbox"/>	240	55	2024-09-27 14:45:08
<input type="checkbox"/>	241	58	2024-09-27 14:45:11
<input type="checkbox"/>	242	60	2024-09-27 14:45:14
<input type="checkbox"/>	243	60	2024-09-27 14:45:17
<input type="checkbox"/>	244	57	2024-09-27 14:45:20
<input type="checkbox"/>	245	57	2024-09-27 14:45:23
<input type="checkbox"/>	246	57	2024-09-27 14:45:26
<input type="checkbox"/>	247	59	2024-09-27 14:45:29
<input type="checkbox"/>	248	60	2024-09-27 14:45:32
<input type="checkbox"/>	249	59	2024-09-27 14:45:35
<input type="checkbox"/>	250	59	2024-09-27 14:45:38