Ultrasonic sensor data collection using Arduino and MYSQL

Capturing data from sensors

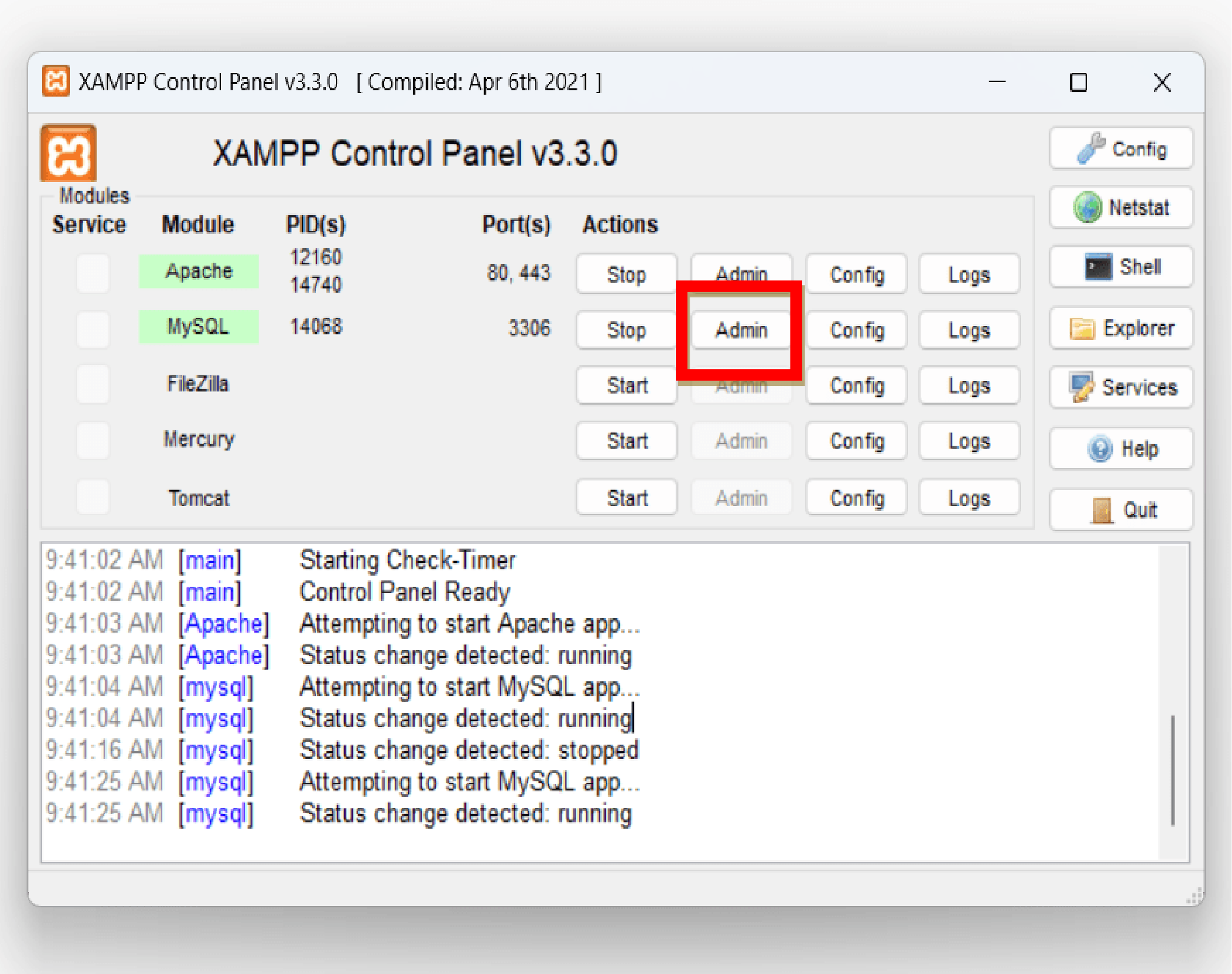
The output of a device that recognizes and reacts to an input of any kind from the external world is called sensor data. The output can be used to direct a process, supply information to the end user, or serve as input for another system.

Steps to follow:

1. MYSQL Database Installation Using XAMPP

XAMPP is installed by using the following link by installing the zip file:
[Download XAMPP (apachefriends.org)](https://www.apachefriends.org)
Once the XAMPP is installed do the following:

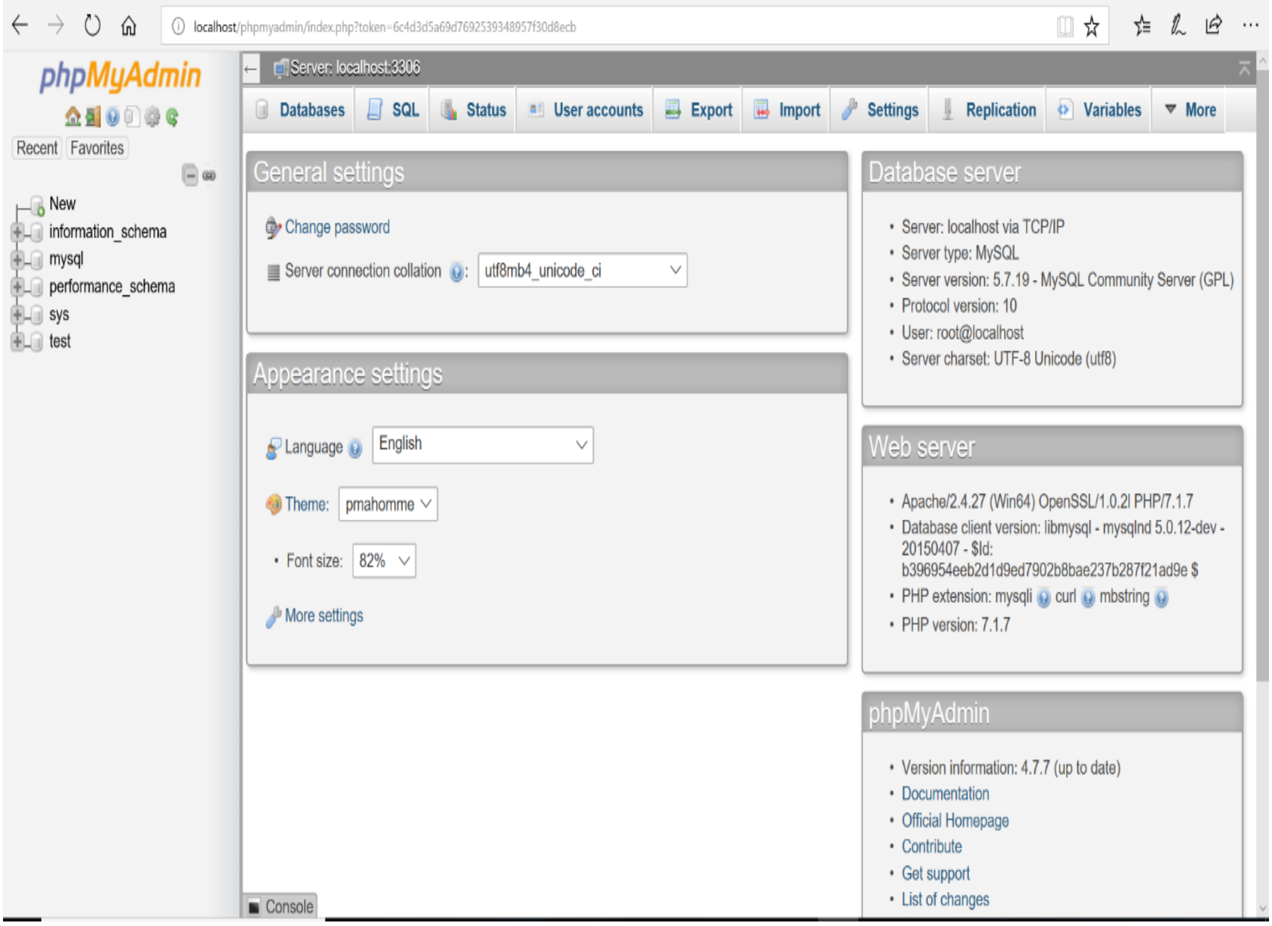
a. **Start** the Apache and MySQL
b. Click on **Admin** to open MySQL IDE


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
□	Apache	12160 14740	80, 443	Stop	Admin	Config	Logs
□	MySQL	14068	3306	Stop	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
9:41:02 AM  [main]    Starting Check-Timer
9:41:02 AM  [main]    Control Panel Ready
9:41:03 AM  [Apache]  Attempting to start Apache app...
9:41:03 AM  [Apache]  Status change detected: running
9:41:04 AM  [mysql]   Attempting to start MySQL app...
9:41:04 AM  [mysql]   Status change detected: running
9:41:16 AM  [mysql]   Status change detected: stopped
9:41:25 AM  [mysql]   Attempting to start MySQL app...
9:41:25 AM  [mysql]   Status change detected: running
```

By clicking on admin page of MYSQL, PHPMYADMIN (phpMyAdmin is a free and open-source administration tool for MySQL and MariaDB as a portable web application written primarily in PHP) is opened successfully as shown below.



General settings

Server connection collation: utf8mb4_unicode_ci

Appearance settings

Language: English

Theme: pmahomme

Font size: 82%

Database server

- Server: localhost via TCP/IP
- Server type: MySQL
- Server version: 5.7.19 - MySQL Community Server (GPL)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.27 (Win64) OpenSSL/1.0.2l PHP/7.1.7
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - $Id: b396954eeb2d1d9ed7902b8bae237b287f21ad9e $
- PHP extension: mysqli curl mbstring
- PHP version: 7.1.7

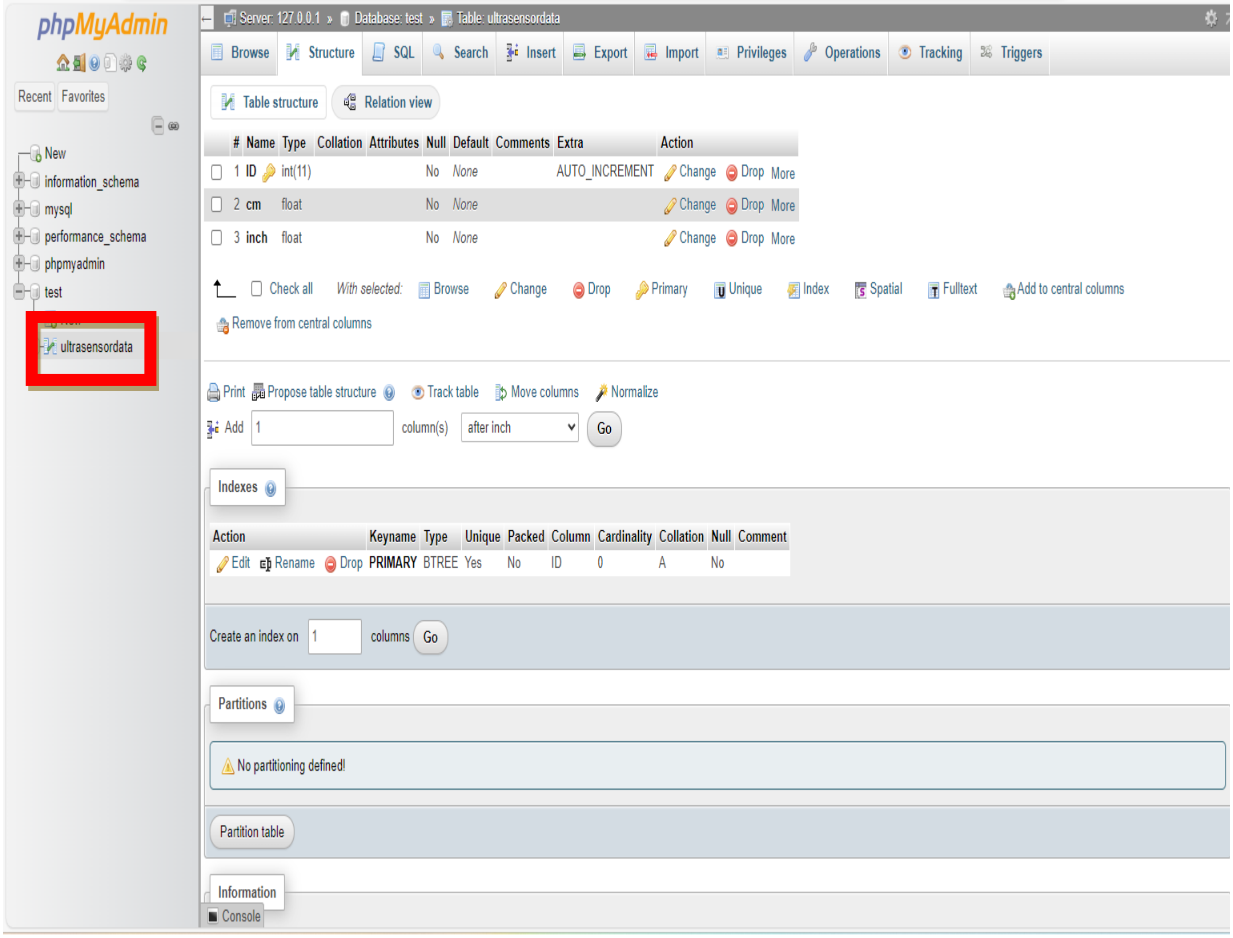
phpMyAdmin

- Version information: 4.7.7 (up to date)
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes

After installing phpMyAdmin, needs to create the following:

a. Database[test] and
b. A table [distance] inside the database [test]
c. Create columns [inside table distance] has to be created to store the data from the sensor.

For ultrasonic senor experiment,



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	ID	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	cm	float			No	*None*			Change Drop More
□	3	inch	float			No	*None*			Change Drop More

Add 1 column(s) after inch

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	ID	0	A	No	

Create an index on 1 columns

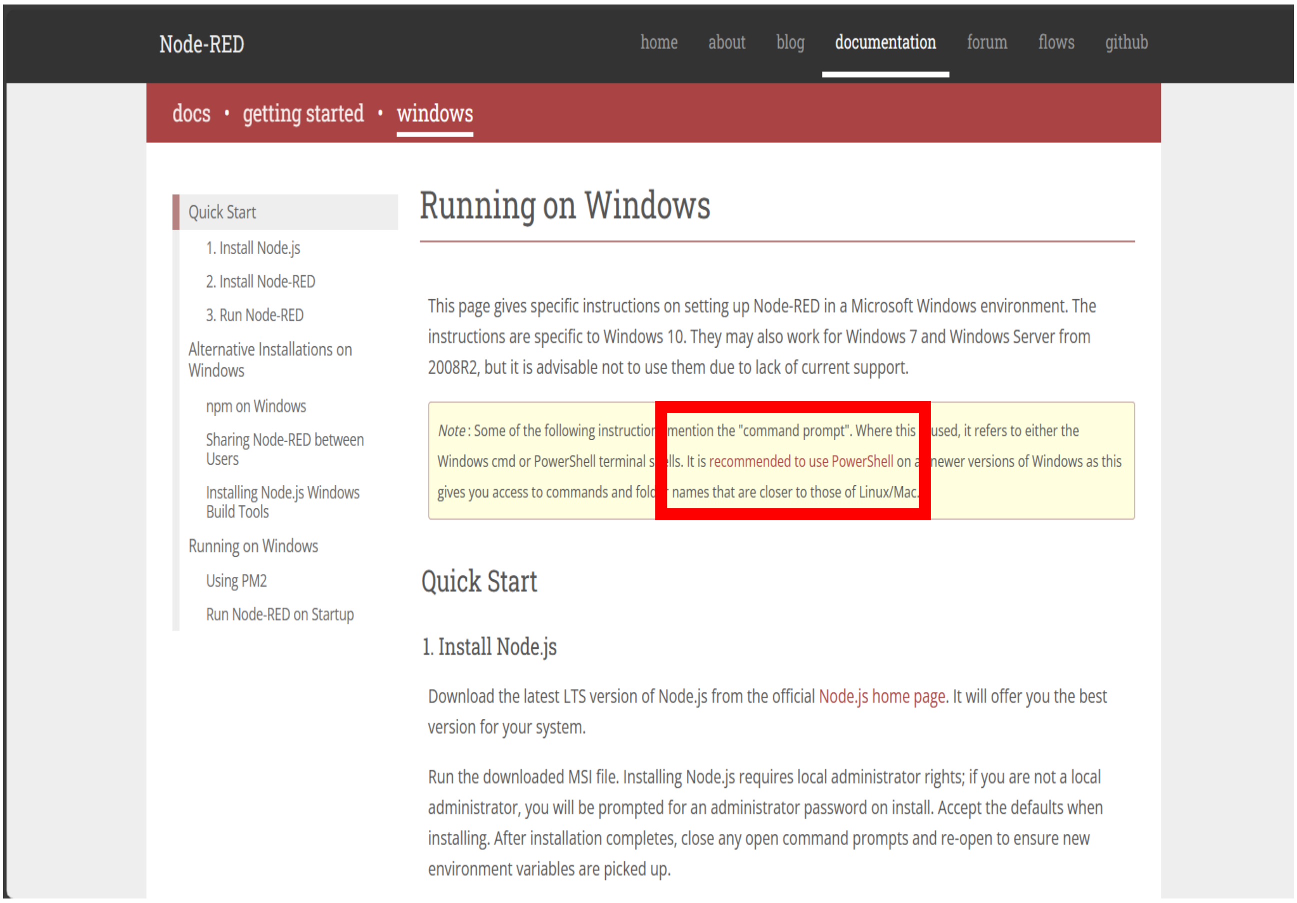
No partitioning defined!

2. Node Red Installation

Node-RED is the flow-based, low-code development tool for visual programming, developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things.

Node-RED is built on top of Node.js.
So, Node.js has to be installed before Node-RED to make the NODE-RED work.
Before Installing Node-Red, Node-Red is installed using the link:
Running on Windows : Node-RED (nodered.org)

The below page supports with the commands, which is to be followed for the installation.



Running on Windows

This page gives specific instructions on setting up Node-RED in a Microsoft Windows environment. The instructions are specific to Windows 10. They may also work for Windows 7 and Windows Server from 2008R2, but it is advisable not to use them due to lack of current support.

Note : Some of the following instructions mention the "command prompt". Where this is used, it refers to either the Windows cmd or PowerShell terminal shells. It is recommended to use PowerShell on all newer versions of Windows as this gives you access to commands and folder names that are closer to those of Linux/Mac.

Quick Start

1. Install Node.js

Download the latest LTS version of Node.js from the official Node.js home page. It will offer you the best version for your system.

Run the downloaded MSI file. Installing Node.js requires local administrator rights; if you are not a local administrator, you will be prompted for an administrator password on install. Accept the defaults when installing. After installation completes, close any open command prompts and re-open to ensure new environment variables are picked up.

3. The highlighted link in the image is clicked to download the Node.js.

Once installed, open a command prompt and run the following command in the command prompt to ensure Node.js and npm are installed correctly.

a. Using cmd - `node --version && npm --version`

Need to receive back output that looks similar to:

```
v18.15.0
9.5.0
```

Install Node-RED

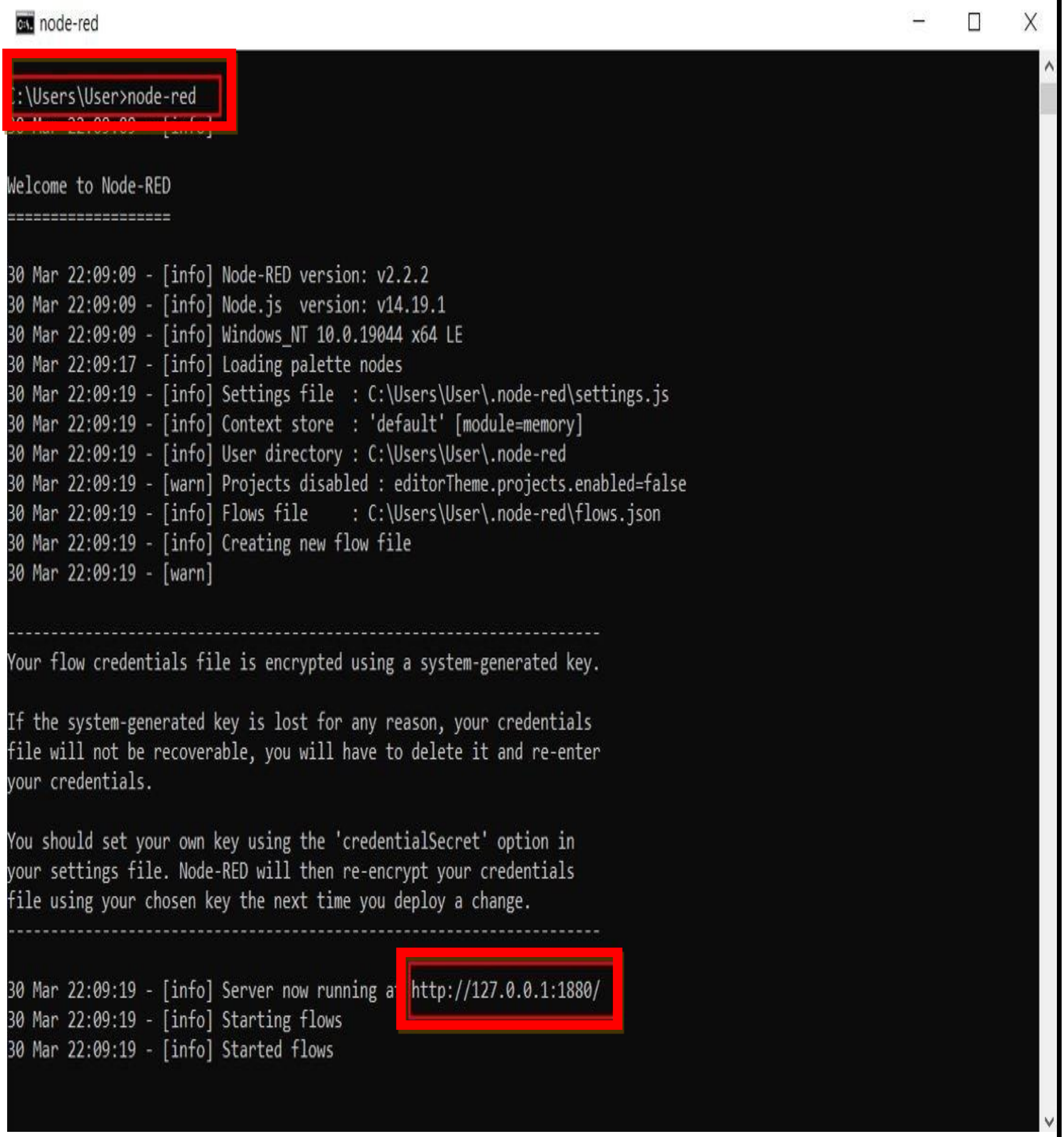
Installing Node-RED as a global module adds the command `node-red` to your system path.

b. Execute the following at the command prompt:

```
npm install -g --unsafe-perm node-red
```

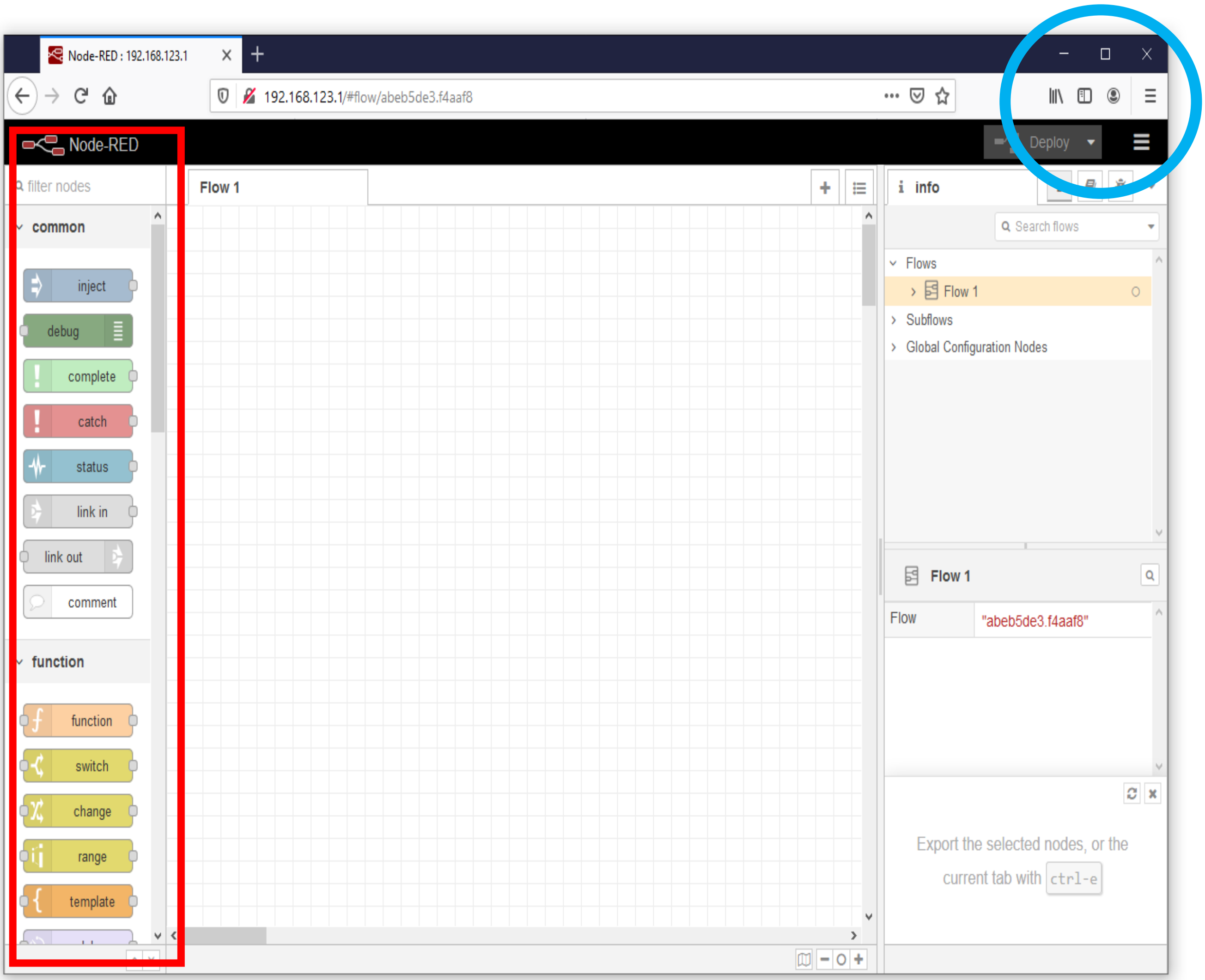
Run Node-RED

c. Once installed, now ready to run **node-red.**



```
:\Users\User>node-red
30 Mar 22:09:09 - [info]

Welcome to Node-RED
===================

30 Mar 22:09:09 - [info] Node-RED version: v2.2.2
30 Mar 22:09:09 - [info] Node.js  version: v14.19.1
30 Mar 22:09:09 - [info] Windows_NT 10.0.19044 x64 LE
30 Mar 22:09:17 - [info] Loading palette nodes
30 Mar 22:09:19 - [info] Settings file  : C:\Users\User\.node-red\settings.js
30 Mar 22:09:19 - [info] Context store  : 'default' [module=memory]
30 Mar 22:09:19 - [info] User directory : C:\Users\User\.node-red
30 Mar 22:09:19 - [warn] Projects disabled : editorTheme.projects.enabled=false
30 Mar 22:09:19 - [info] Flows file     : C:\Users\User\.node-red\flows.json
30 Mar 22:09:19 - [info] Creating new flow file
30 Mar 22:09:19 - [warn]

---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

30 Mar 22:09:19 - [info] Server now running at http://127.0.0.1:1880/
30 Mar 22:09:19 - [info] Starting flows
30 Mar 22:09:19 - [info] Started flows
```

d. After running the node-red command in command prompt, the server's name is used to open the home page of node red. (press ctrl+enter). Will opens the Node-RED IDE.

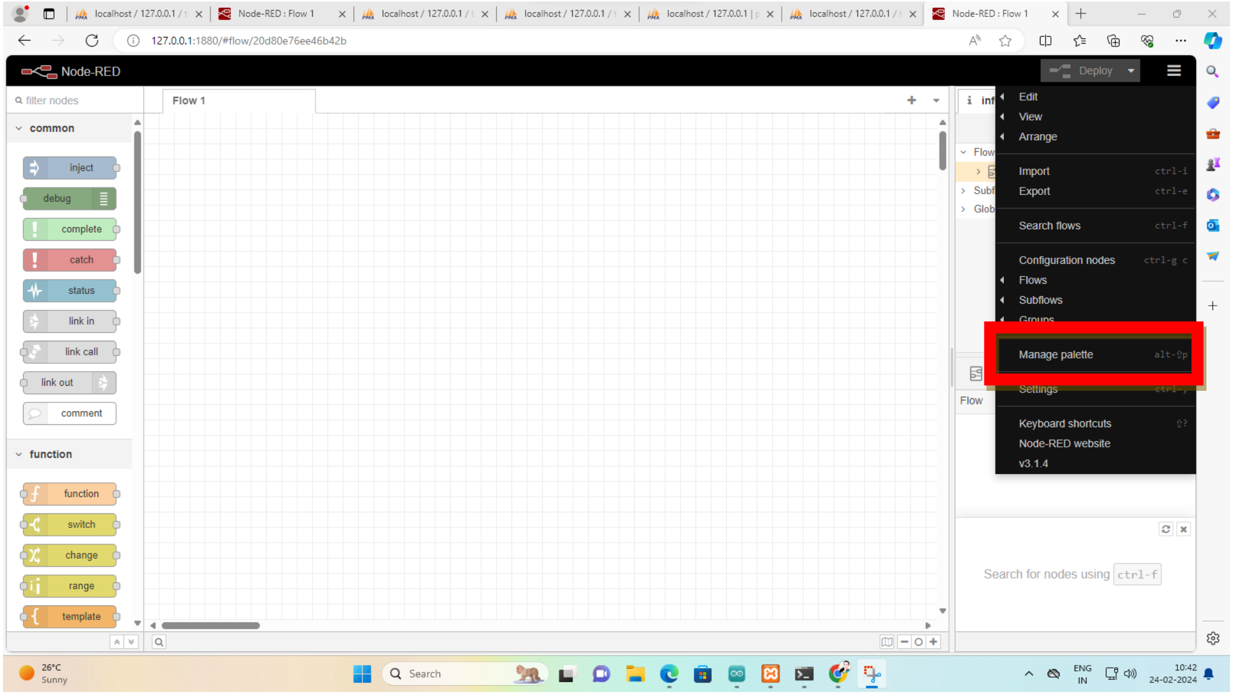


Home Page of Node-Red

As shown in the **red coloured rectangle**, the **Pallets** are used to drag and drop in the flow which is used to transfer the data from Arduino to MySQL.

If the required pallet is not present, then that particular pallet can be installed easily by clicking on the **3 horizontal lines** in the top right corner of the node-red and selecting Manage pallets option as shown in the **blue circle**.

Aim is to transfer the data from the **sensor to MySQL** so that the data is stored permanently in the database.

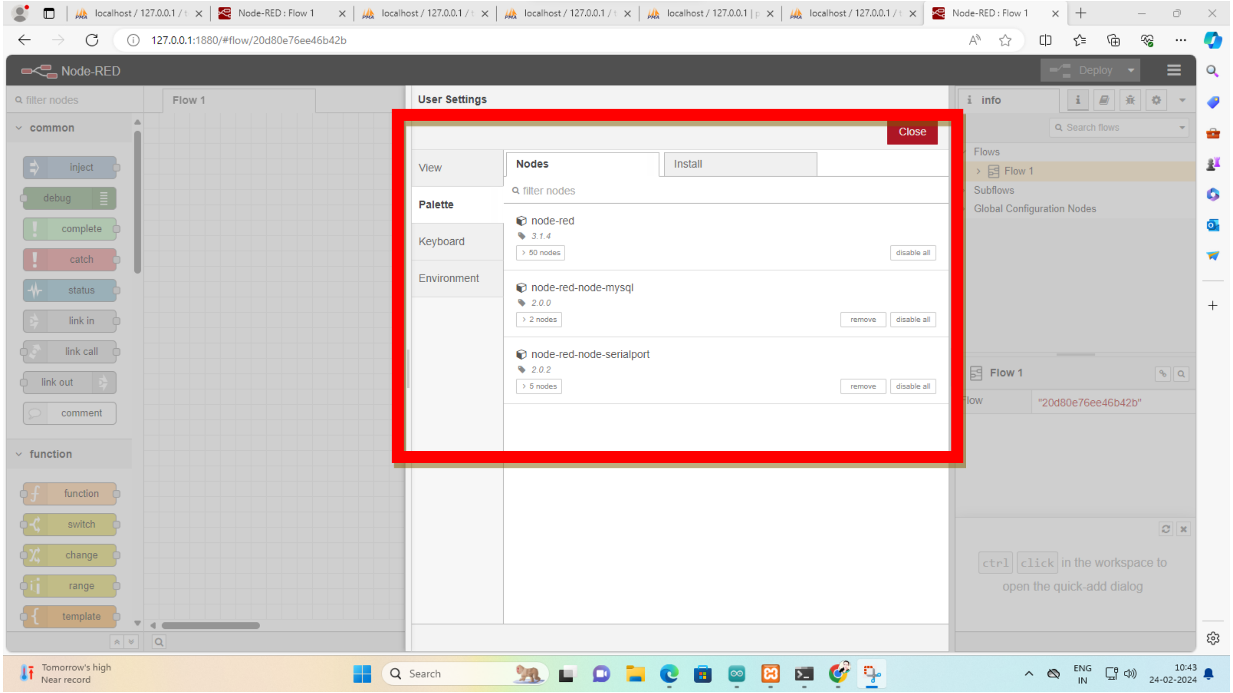


a. Install the following pallets:

i) install node-red

ii) node-red-node-mysql

iii) node-red-node-serialport



4. **Example application** we are taking the code of **Ultrasonic sensor** as example and trying to store the data generated by the sensor in the MySQL database.

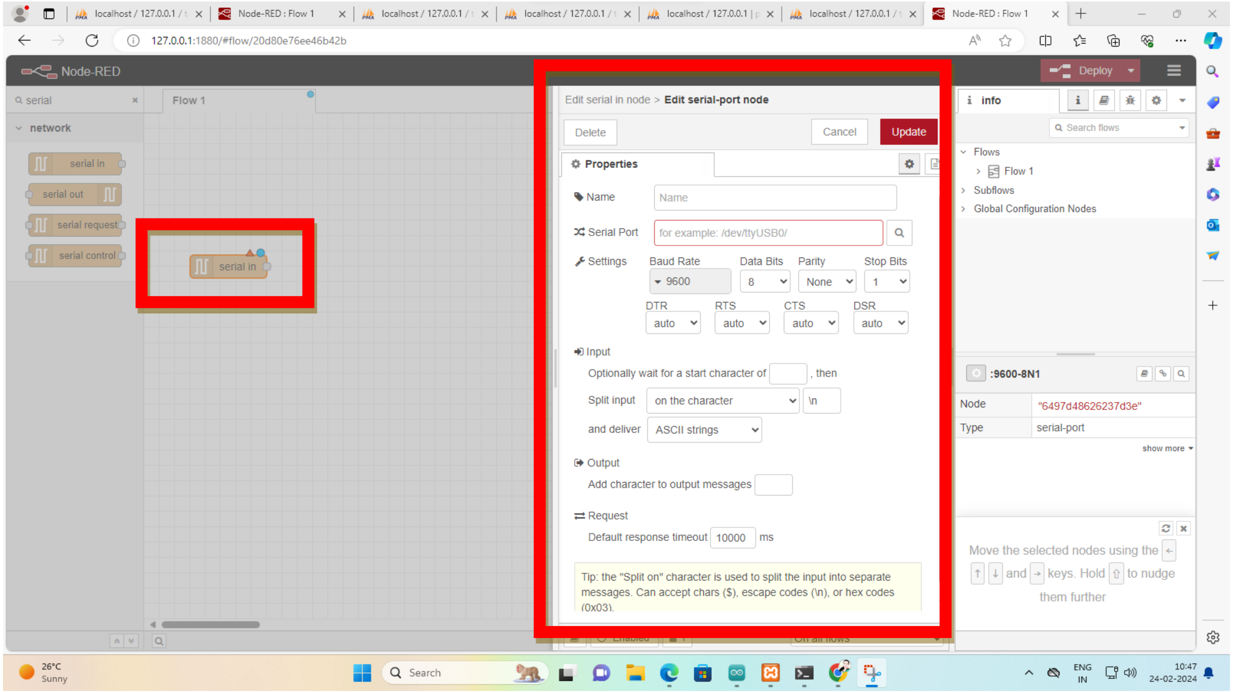
   a. Code of Ultrasonic Sensor to be written in the Arduino IDE:

```
const int trigPin = 7;
const int echoPin = 8;
void setup() {
Serial.begin(9600);
pinMode(trigPin,OUTPUT);
pinMode(echoPin,INPUT);
}
 void loop() {
  long duration, inches, cm;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);
// The data has to be fed in this format so that it will be stored in MySQL
  Serial.print("{\"inches\":");
  Serial.print(inches);
// The data has to be fed in this format so that it will be stored in MySQL
  Serial.print(",\"cm\":");.
  Serial.print(cm);
  Serial.println("}")
  delay(1000);
}
long microsecondsToInches(long microseconds) {
  return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
}
```

5. Establishing the connection Application output to Database

We need four different pallet to achieve this connection and they are:

a. The first pallet to be dragged from node-red is **Serial In.** Which is used to specify the port used by Arduino UNO. As given in the figure below (red rectangle), by double clicking on the **Serial In Pallet**. A dialog box is opened which is used to specify the **baud value (9600) and COM number(which ever the port identified by Arduino IDE)**. And finally press **Done** button.



Edit serial in node > Edit serial-port node

Delete Cancel Update

Properties

Name: Name

Serial Port: for example: /dev/ttyUSB0/

Settings

Baud Rate	Data Bits	Parity	Stop Bits
9600	8	None	1

DTR	RTS	CTS	DSR
auto	auto	auto	auto

Input

Optionally wait for a start character of , then

Split input on the character \n

and deliver ASCII strings

Output

Add character to output messages

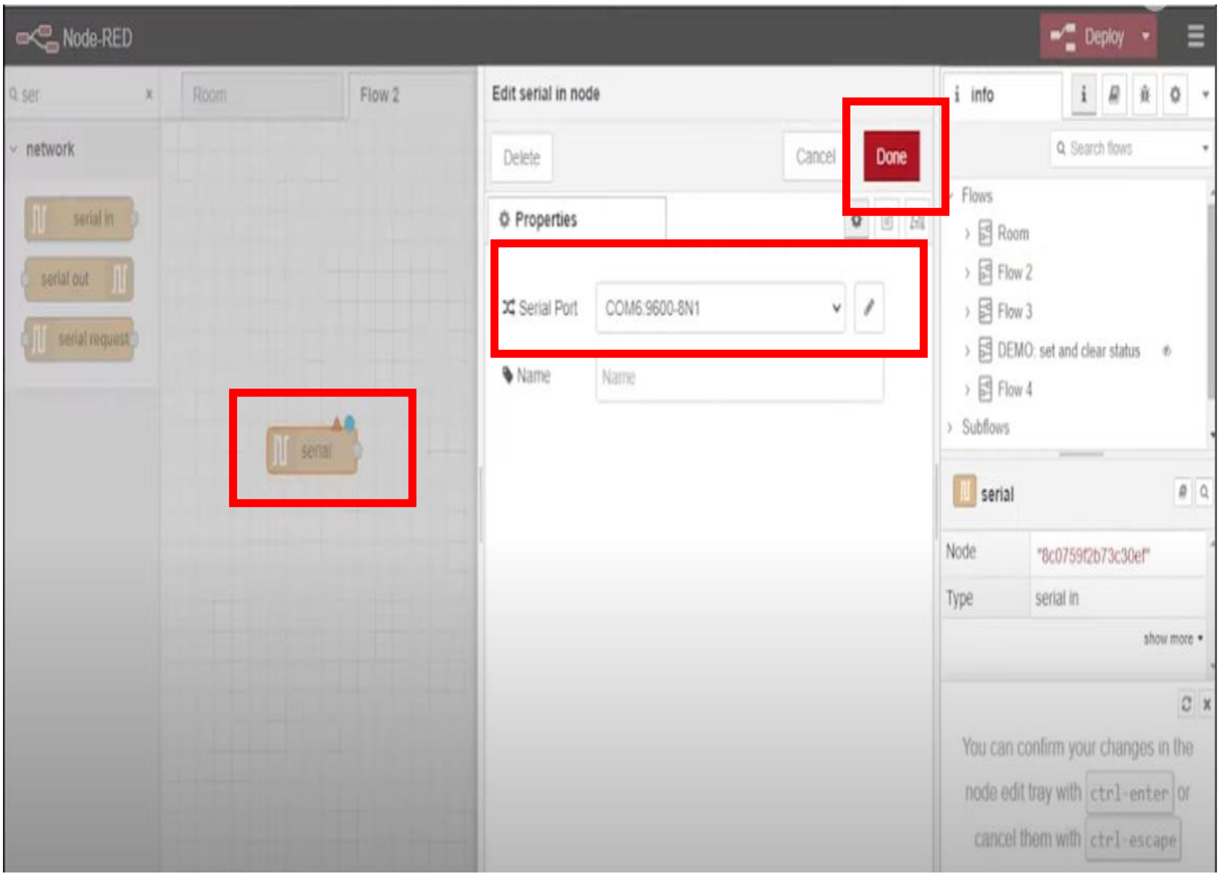
Request

Default response timeout 10000 ms

Tip: the "Split on" character is used to split the input into separate messages. Can accept chars ($), escape codes (\n), or hex codes (0x03).

:9600-8N1

Node	"6497d48626237d3e"
Type	serial-port



Edit serial in node

Delete Cancel Done

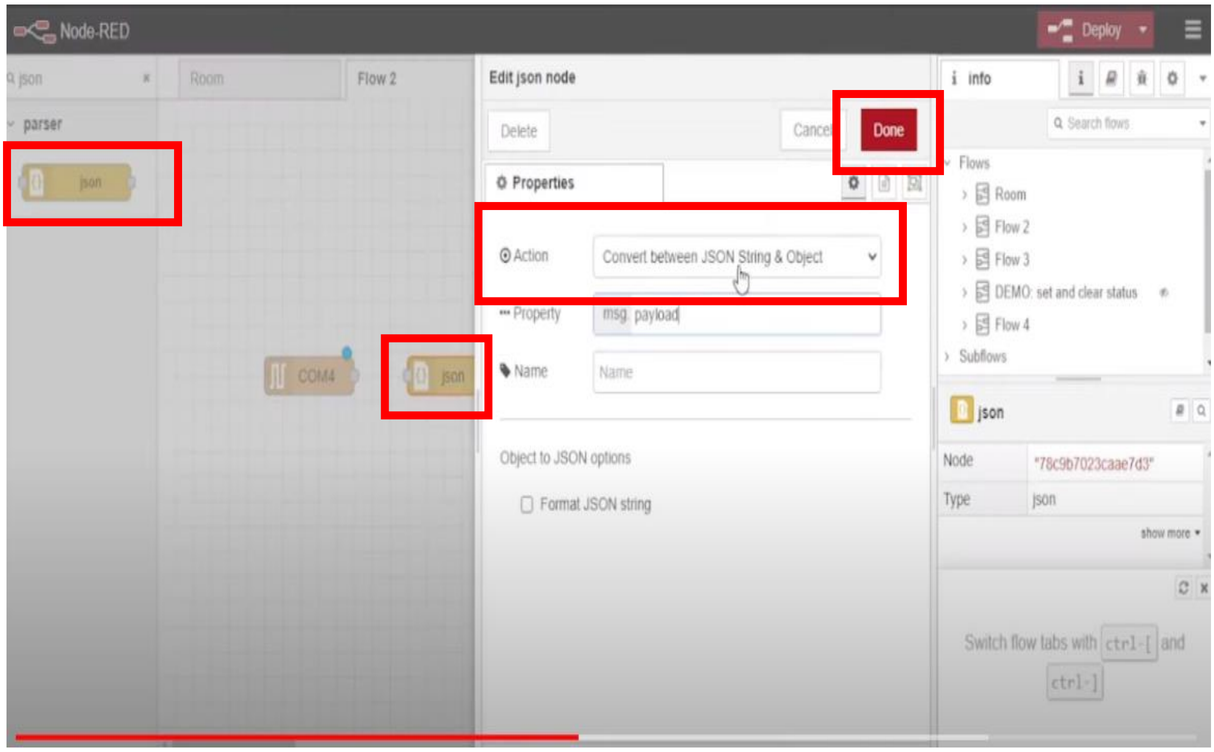
Properties

Serial Port: COM6:9600-8N1

Name: Name

Node	"8c0759f2b73c30ef"
Type	serial in

b. The second pallet to be dragged is **JSON,** and double click on the pallet, Where the action to be selected as **Convert between JSON String and Object**.



Edit json node

Action: Convert between JSON String & Object

Property: msg. payload

Name: Name

Object to JSON options

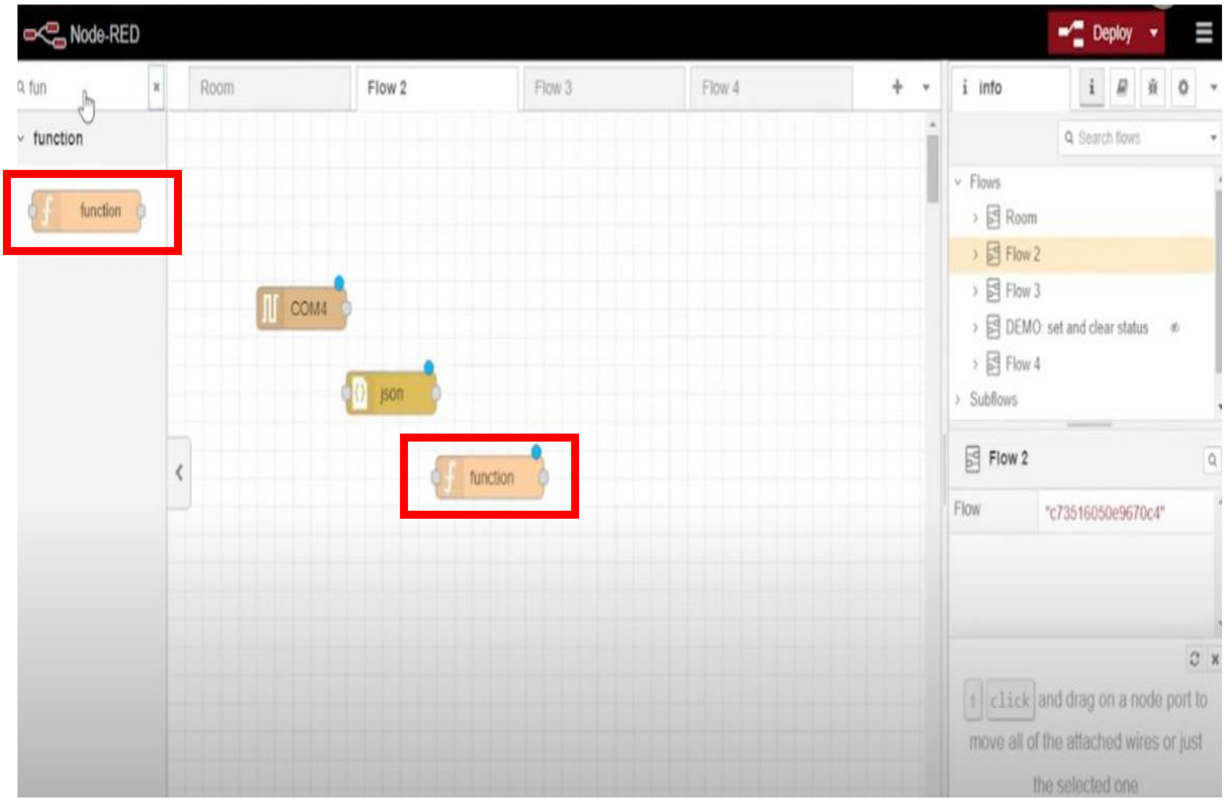
□ Format JSON string

Node: "78c9b7023caae7d3"

Type: json

Switch flow tabs with ctrl-[and ctrl-]

c. The third pallet to be selected is **function** where the below code has to be written.



↕ click and drag on a node port to move all of the attached wires or just the selected one

i) Drag and drop the function pallent

ii) Double click on the pallet

iii) Write the code on the function pallet to initialize the data values from the sensor readings of the application

iv) Write the SQL Query to insert the sensor generated values to the database for future use.

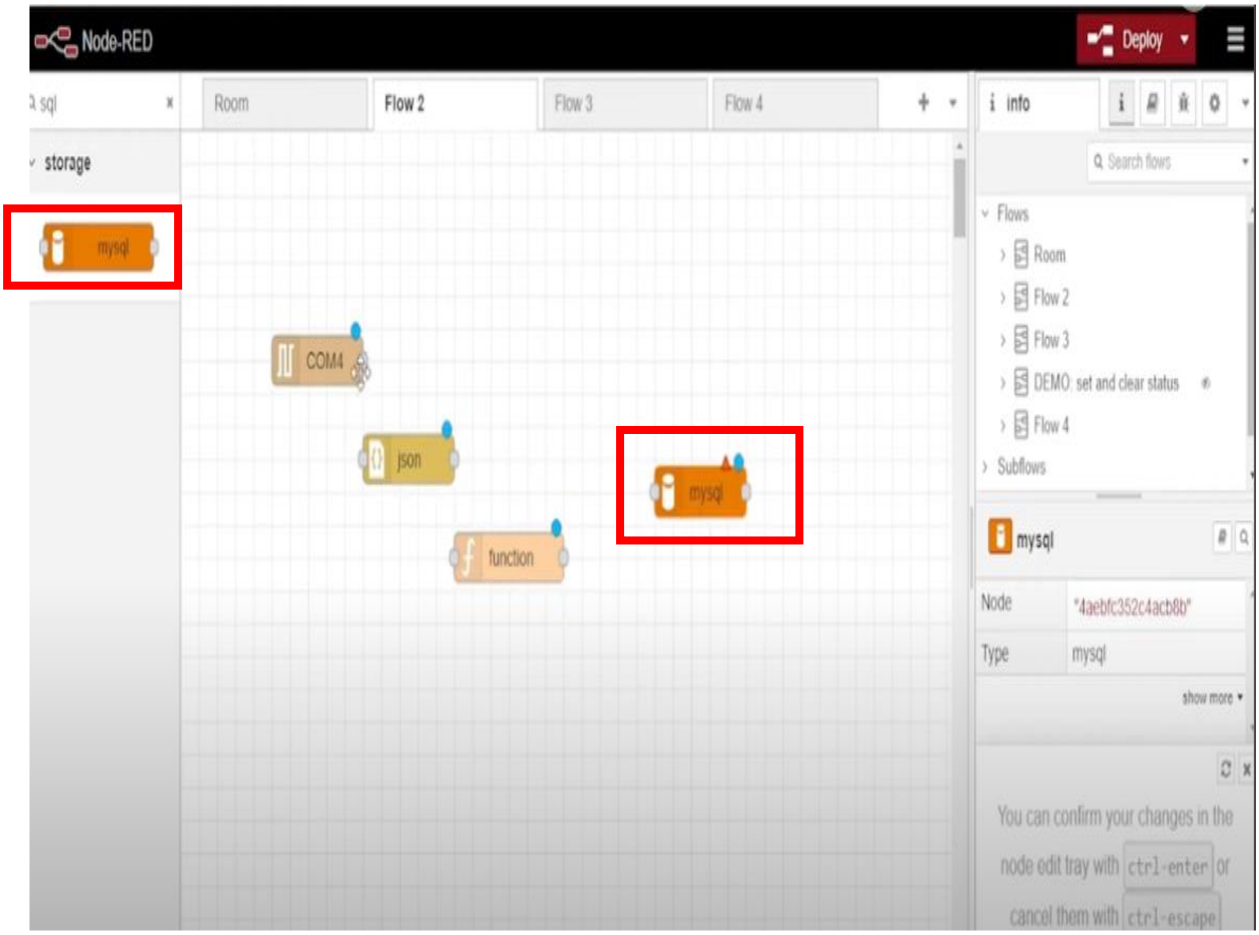
Note:

The Arduino UNO program variables and the MySQL table column names should be same

Code to be written in function pallet:

```
var value = JSON.parse(JSON.stringify(msg.payload)); //converting string to
                                                      // JSON object
value = msg;
var sensor1 = msg.payload.inches; //adding value to the payload
var sensor2 = msg.payload.cm;     //adding value to the payload
msg.payload = [sensor1, sensor2]; //adding value to the payload
msg.topic = 'INSERT INTO distance(inches,cm) values (?,?);'; //query to insert
return msg;
```

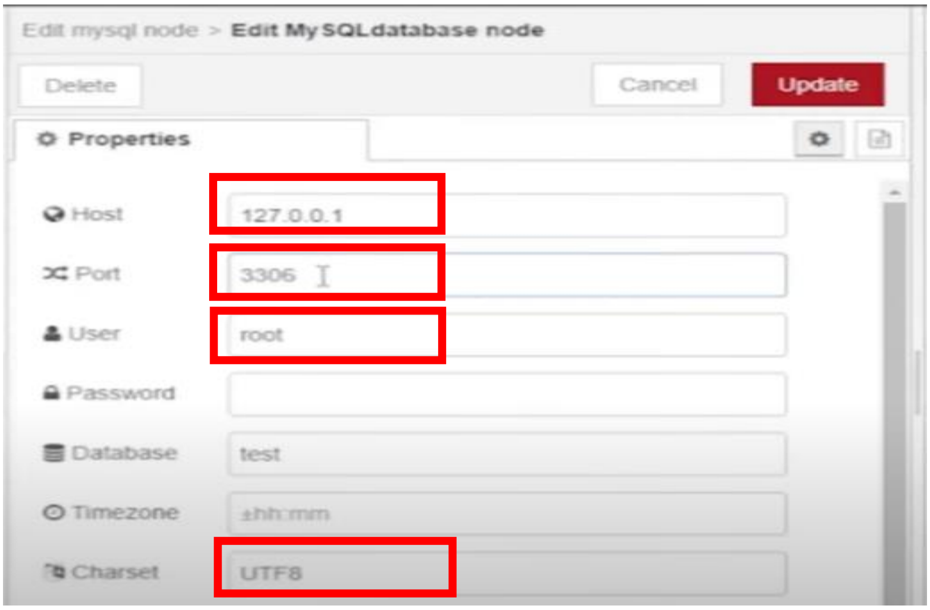
d. The fourth pallet is **mysql to include.** Which has to be installed in **manage pallets option**(if not available). The name of the pallet is ***node-red-node-mysql.***



Node "4aebfc352c4acb8b"
Type mysql

You can confirm your changes in the node edit tray with ctrl-enter or cancel them with ctrl-escape

i) Doulble click on the mysql pallet to configure.
ii) The **database name** (test) has to be mentioned
iii) Enter **root** information
iv) Enter the the port number **3306** of xampp
v) Checkout default charset as Charset as UTF-8
vi) Click on **update** button



Host 127.0.0.1

Port 3306

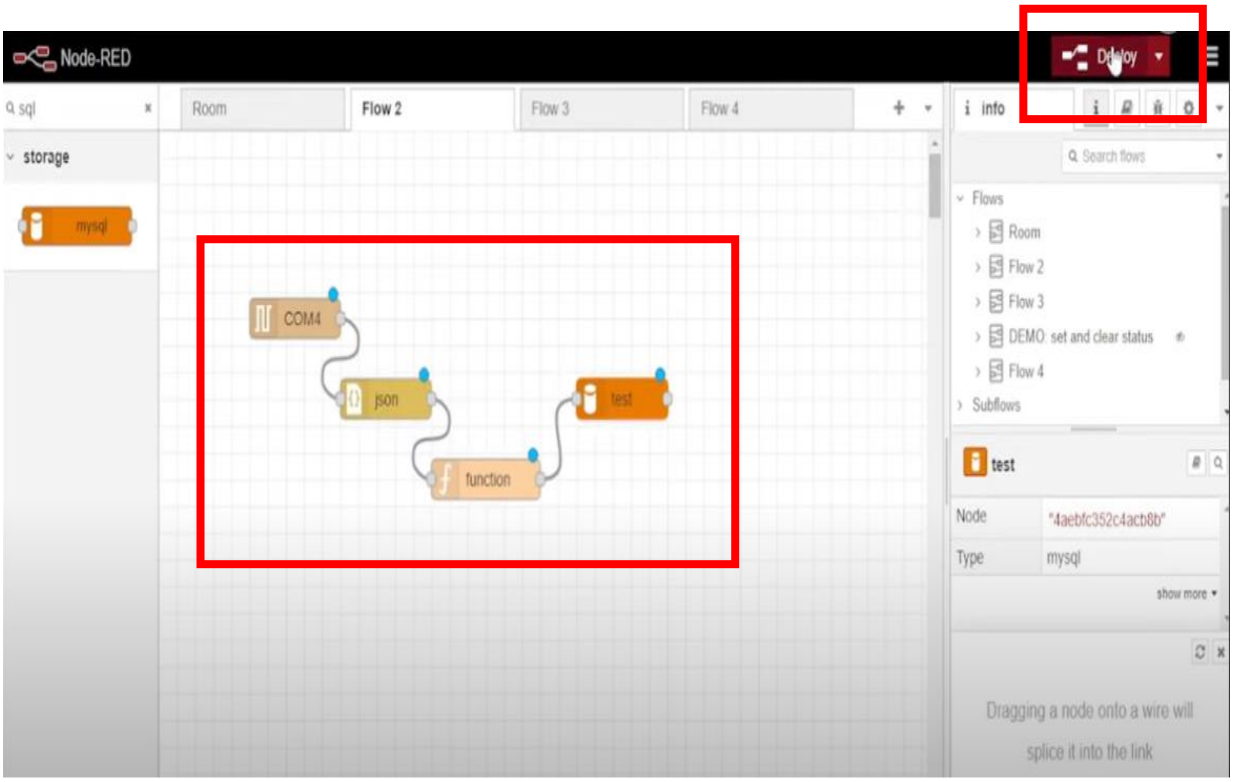
User root

Password

Database test

Timezone ±hh:mm

Charset UTF8



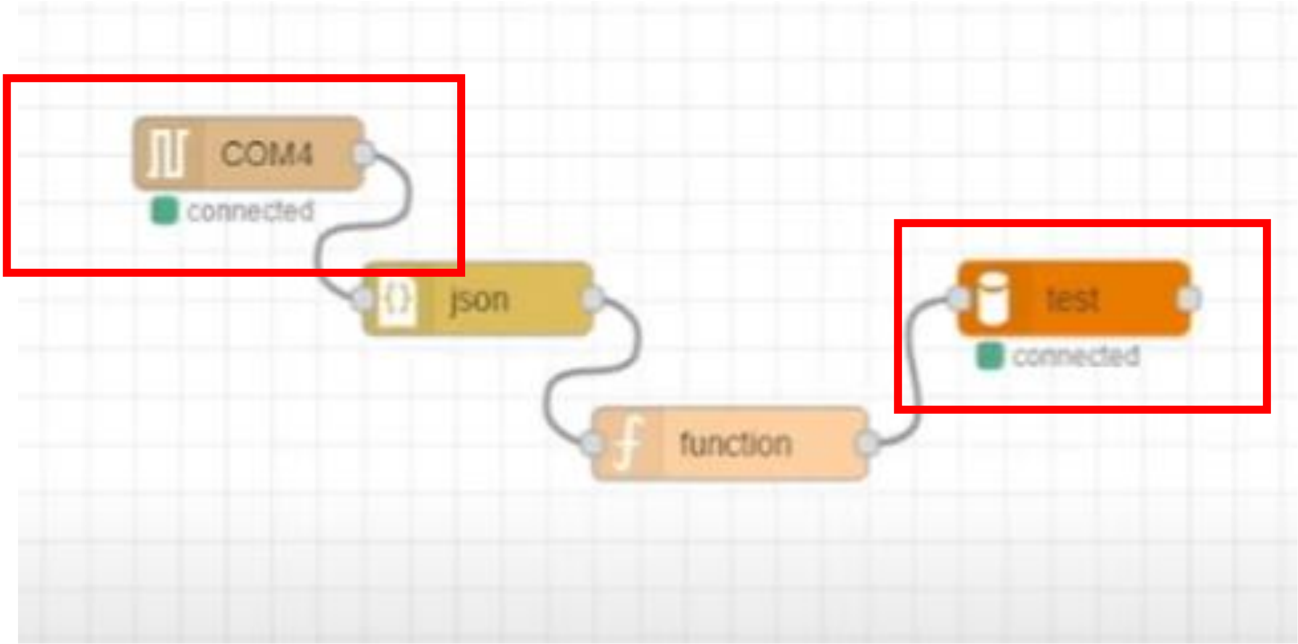
test

Node	"4aebfc352c4acb8b"
Type	mysql

e. After considering all the pallets, these pallets are to get interconnected automatically, as shown in the figure above. Finally **deploy button** is used to transfer the data.

6. Verification of the connection with MySQL

After deploying both the **COM pallet** and **Mysql pallet** should show as **connected**. Which is shown in the figure below, then the data shall transfer successfully from Arduino circuit to MySql. Also we should get **OK** at the end to confirm for the successful connection establishment from sensor application with MySQL.


COM4
connected
json
function
test
connected