

# Personal Networking Assistant for Conference Meetup

Name: Shailesh Agrawal

PRN: 22010348

Roll no.: 311006

Div/Batch: A/A1

Sr. no	Index	Pg. no
1.	Introduction	2
2.	Objective	2
3.	Hardware Requirements	2
4.	Software Requirements	2
5.	Block Diagram	5
6.	Procedure	5
7.	Output	11
8.	Conclusion	14
9.	Reference	15

## 1. INTRODUCTION

PNA is a Personal Networking Assistant for conference meetings. It is an hardware and a software based application. It is very useful during the meeting when somebody wants to exchange his/her own information then this device comes into picture. So basically in big conference events or meetups many people participates or we can say are invited so far, most of the people there did not know the identity or profile of other. Before going to start communication with that people we should know about his position and profession,

To make this thing easy and simple our PNA is framed. Where we will circulate a google form at the time of invitation with a RFID card with his/her name on it and form a database of collected personal information by the google forms. Now at the entrance of conference hall there will be a scanning machine where the person will scan the provided RFID card, as soon as the scan take place that person's profile will be display on the provided server. So looking at the person's name on the card we can search that person on server provided and we will get to know about his profile and if we want we can save it by taking snapshot so we can contact then in future also.

## 2. OBJECTIVE

- To create database from google form and linked it with the UID of RFID tag
- To interface MFRC522 reader to ESP8266 which connect the hardware to the hosted server
- Hosting MySQL database on localhost server with including frontend as per the requirements
- Displaying registered data on the hosted server after canning RFID card on the reader
- Counting of number of displayed data entry for attendance purpose

## 3 HARDWARE REQUIREMENTS

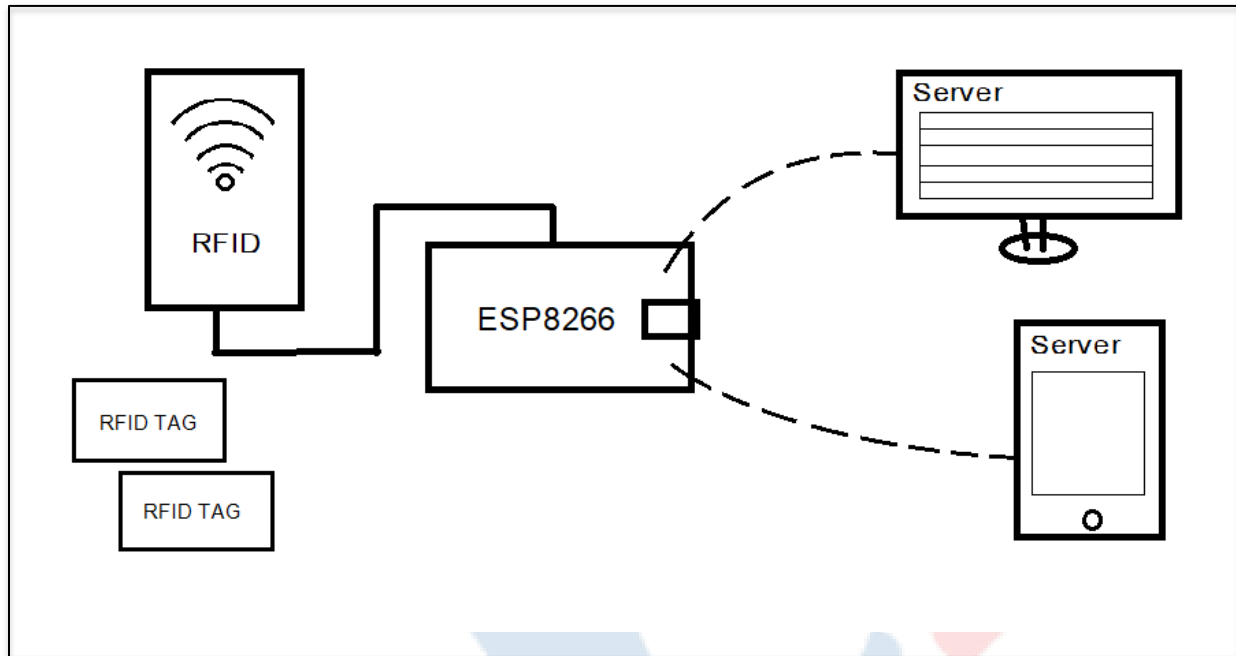
- NodeMCU ESP8266
- MFRC522 RFID Reader
- RFID Tags
- Jumper wires
- USB to micro-USB Cable
- Breadboard

## 4. SOFTWARE REQUIREMENTS

- Arduino IDE
- Cloud Platform App
- XAMPP server
- PHP Source Code

- RFID-RC522 Library
- NodeMcu ESP8266 Library and Board Manager

## 5. BLOCK DIAGRAM

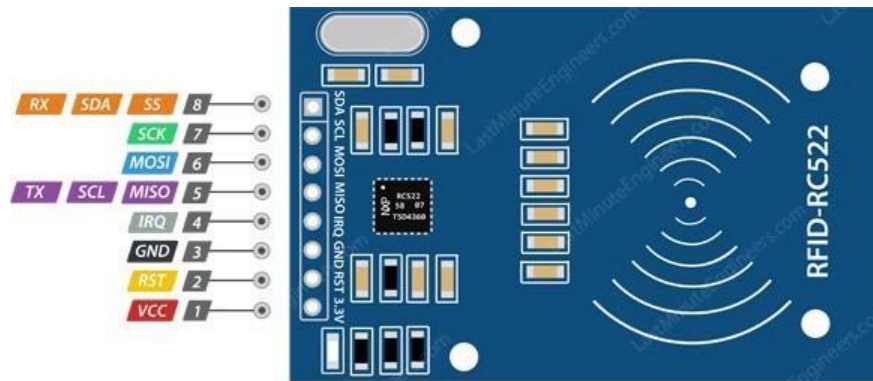


## 6. PROCEDURE

### A. MFRC522 RFID interfacing with NodeMCU

- RFID: Radio-Frequency identification is the use of radio waves to read and capture information stored on a tag attached to an object. A tag can be read from up to several feet away and does not need to be within direct line-of-sight of the reader to be tracked. This is the advantage over Bar-code.

RFID reader: A RFID reader is a device used to gather information from an RFID tag, which is used to track individual objects. Radio waves are used to transfer data from the tag to a reader.



The simple specifications of RC522 module from left to right first pins are as follows:

#### Pin Name Details

1. 3.3V +3.3V Power Supply
2. RST Reset
3. GND Ground Pin
4. IRO Not Connected
5. MISO Serial Communication
6. MOSI Serial Communication
7. SCK TX/RX with ESP8266
8. SDA TX/RX with ESP8266

**Passive tag:** A passive tag is an RFID tag that does not contain a battery, the power is supplied by the reader. When radio waves from the reader are encountered by a passive rfid tag, the coiled antenna within the tag forms a magnetic field. The tag draws power from it.



Features:

Module Name: MF522-ED

Working current : 13—26mA/ DC 3.3V

Standby current : 10-13mA/DC 3.3V

Sleeping current : <80uA

Peak current : <30mA

Working frequency : 13.56MHz

Card reading distance : 0~60mm (mifare1 card)

Protocol : SPI

Data communication speed : Maximum 10Mbit/s

Card types supported : mifare1 S50、mifare1 S70、Mifare UltraLight、mifare Pro、mifare Desfire

Dimension : 40mm×60mm

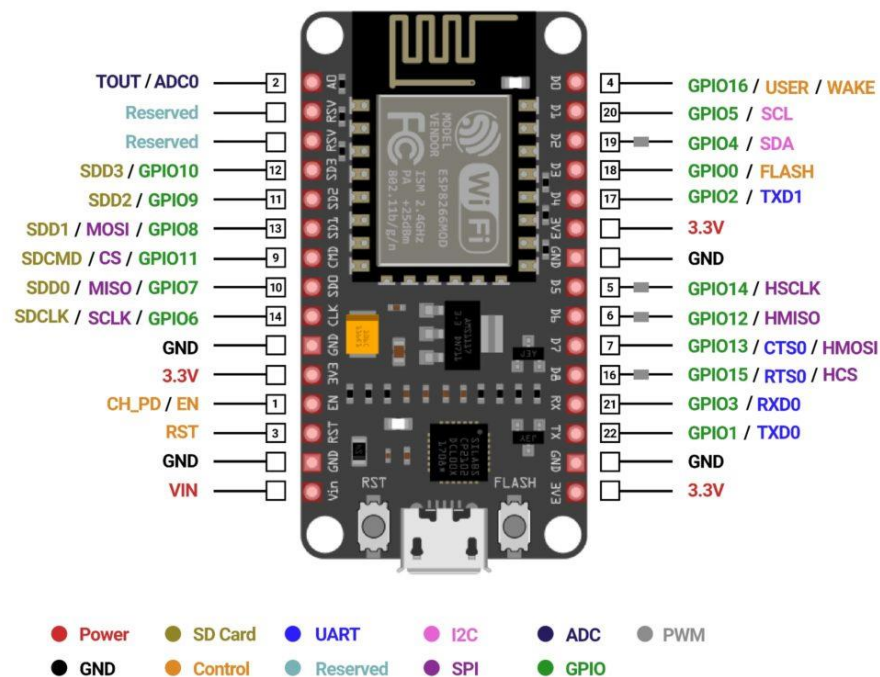
Working temperature : -20—80 degree

Storage temperature : -40—85 degree

Humidity : relevant humidity 5%—95%

Max SPI speed: 10Mbit/s

- b. ESP8266 NodeMCU: NodeMCU is an open-source firmware and development kit that helps you to prototype or build IoT products. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The firmware uses the Lua scripting language. It is based on the eLua project and built on the Espressif Non-OS SDK for ESP8266.



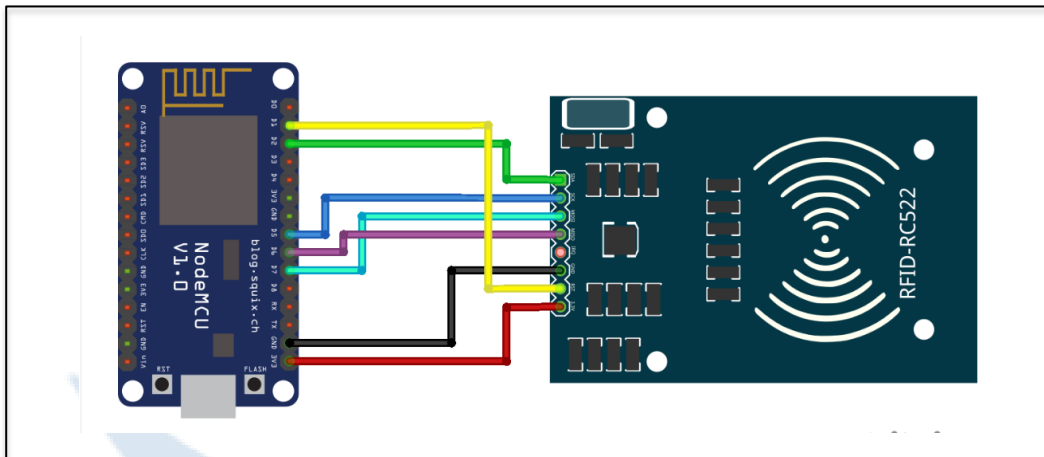
NodeMCU ESP8266 Specifications & Features:

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

c. Pin wiring:

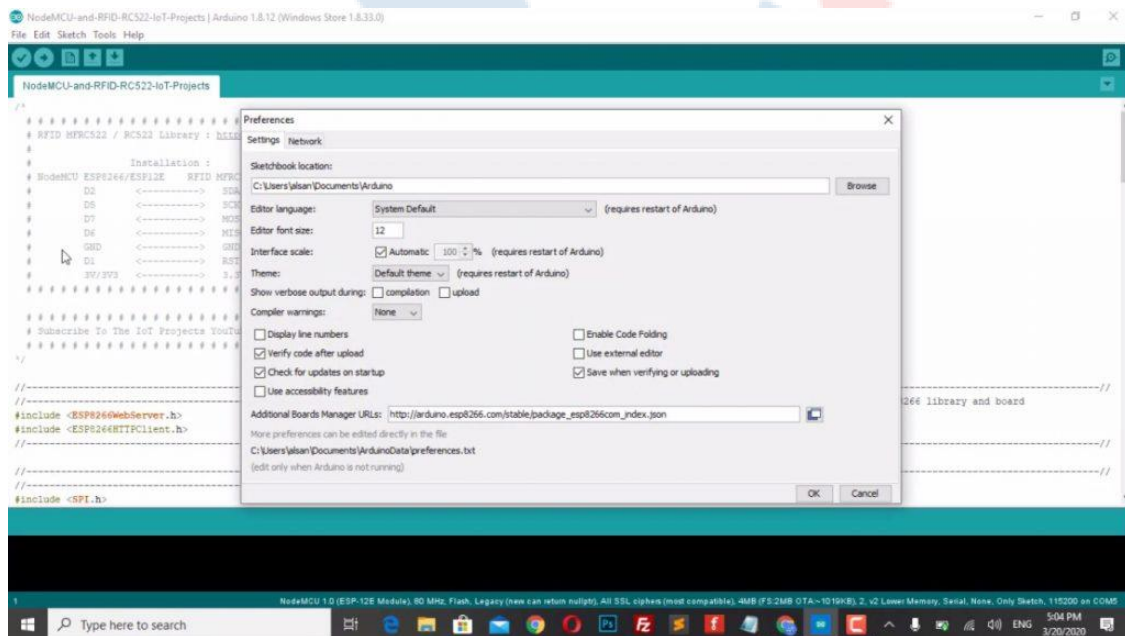
NodeMCU	MFRC522
D4 (GPIO2)	SDK (SS)
D5 (GPIO14)	SCK
D6 (GPIO13)	MOSI
D7 (GPIO12)	MISO (SCL)
	IRQ
GND	GND
D3 (GPIO0)	RST (Flash)
3v	3v

d. Interfacing diagram:

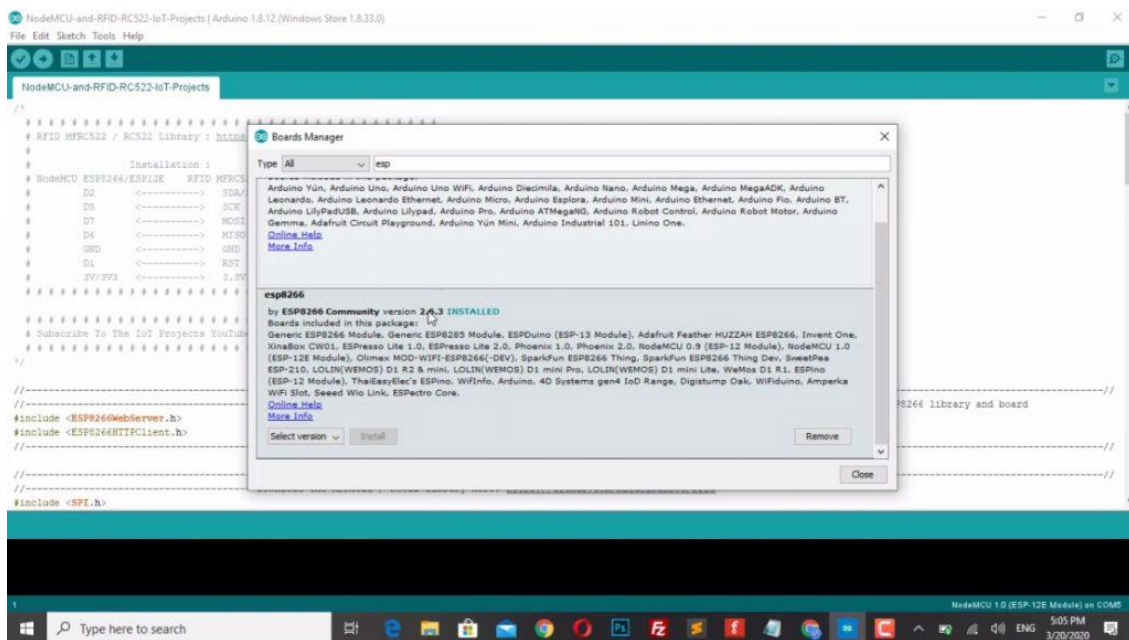


B. Setting up Arduino IDE for ESP8266 Board:

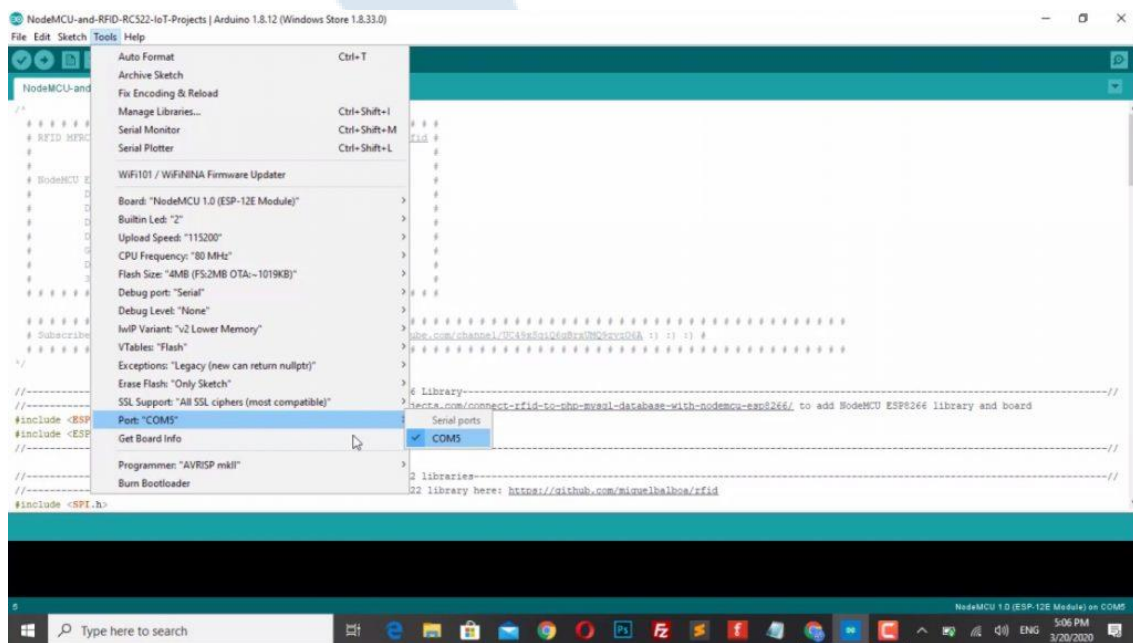
- We need to add the esp8266 board to our Arduino IDE. Open up your IDE then go to “File -> Preferences”
- Paste [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) in additional board manager URLs.



- Go to “Tools -> Board -> Boards Manager” search and install esp8266.



- Go to Tools -> Port and select the COM Port of your NodeMCU. To check what port your NodeMcu is plugged in, go to Device Manager -> Ports (COM & LPT).



- Now select NodeMcu 1.0 (ESP-12E Module) by clicking Tools -> Board. Scroll down until you find it.



- Install the MFRC522 library to your Arduino Libraries folder –[Download RFID Library](#)

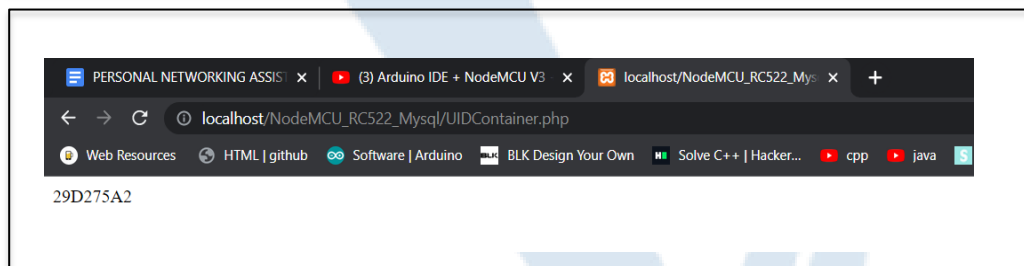
```

NodeMCU_V3_RFID_RC522_DB.ino
114   for(int i=0;i<4;i++){
115       readcard[i]=mfc522.uidByte[i]; //storing the UID of the tag in readcard
116       array_to_string(readcard, 4, str);
117       StrUID = str;
118   }
119   mfc522.PICC_HaltA();
120   return 1;
121 }
122 //-----
123
124 //-----Procedure to change the result of reading an array UID into a string-----
125 void array_to_string(byte array[], unsigned int len, char buffer[]) {
126     for (unsigned int i = 0; i < len; i++)
127     {
128         byte nib1 = (array[i] >> 4) & 0x0F;
129         byte nib2 = (array[i] >> 0) & 0x0F;
130         buffer[i*2+0] = nib1 < 0xA ? '0' + nib1 : 'A' + nib1 - 0xA;
131         buffer[i*2+1] = nib2 < 0xA ? '0' + nib2 : 'A' + nib2 - 0xA;
132     }
133     buffer[len*2] = '\0';
134 }
135 //-----
  
```

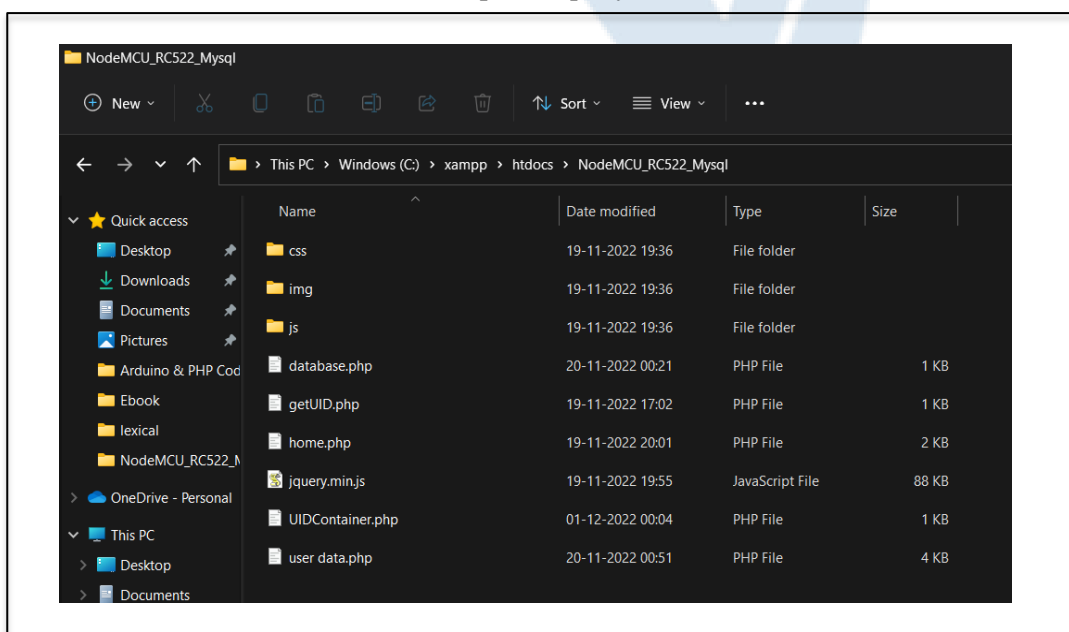
Output Serial Monitor

Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM3')

THE UID OF THE SCANNED CARD IS : 29D275A2



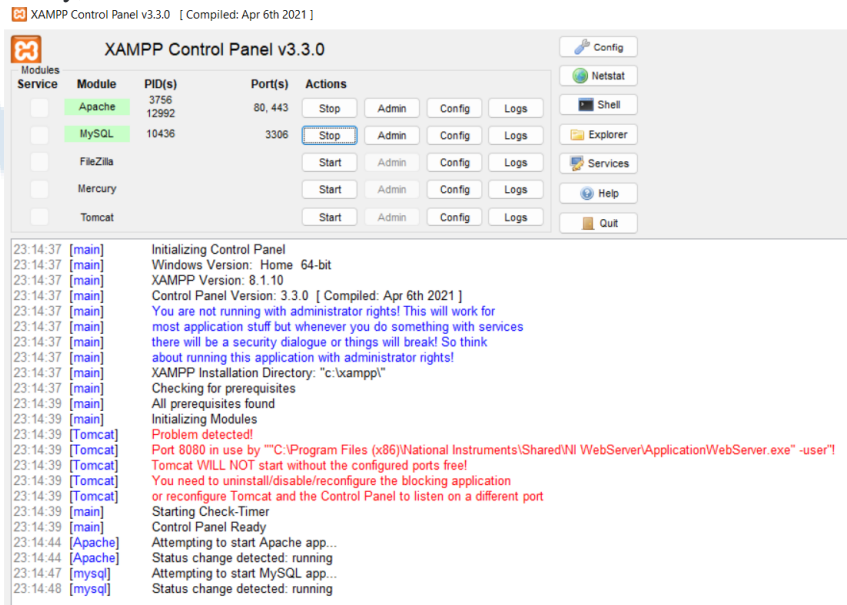
- Download Bootstrap and JQuery



### C. Creating DataBase Using MySQL

Make NodeMCU send Id data from RFID RC522 and create PHP file on the server to receive the data

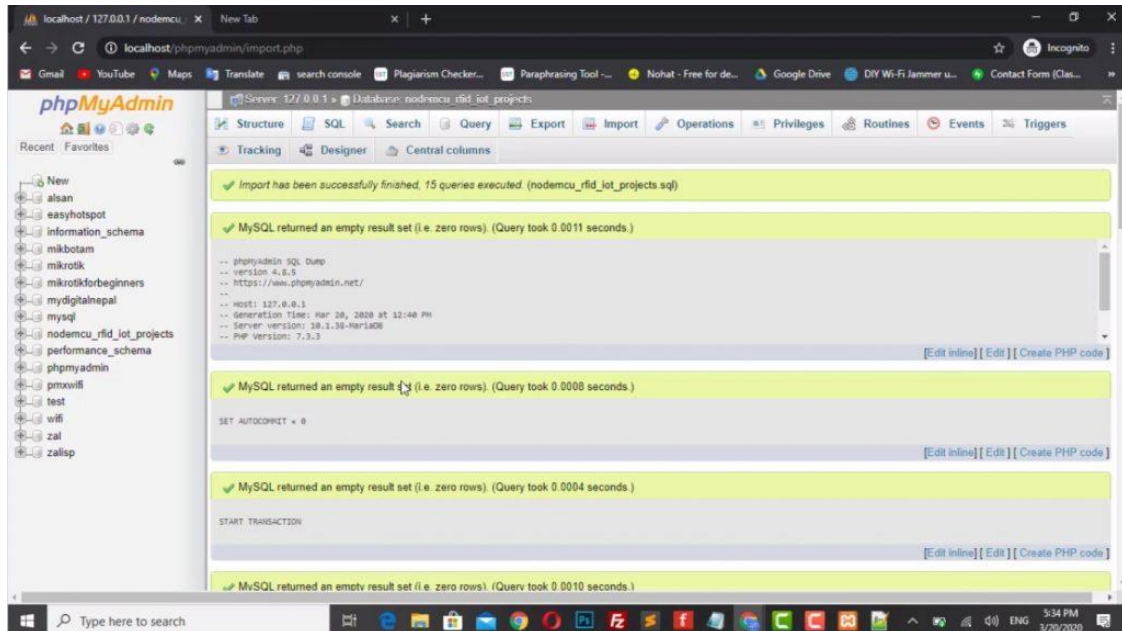
However, we are using the XAMPP server it can be used both in Windows and Linux  
Basically, Ubuntu users use LAMP it's better than X



- Starting Mysql and Apache from XAMPP Control Panel.
- Now there need some database files that will link and will be hosted to our local servers so they are,
  1. database.php
  2. getUID.php
  3. home.php
  4. insertDB.php
  5. nodemcu\_rfid\_iot\_projects.sql
  6. read tag user data.php
  7. read tag.php
  8. registration.php
  9. UIDContainer.php
  10. user data delete page.php
  11. user data edit page.php
  12. user data edit tb.php
  13. user data. .phpAll of this php files are available in this given link, [PHP Source Code](#)

- Copy-paste the RFID PHP Source Code folder in your htdocs folder.

- Creating a new Database named RFID in your PHPMyAdmin.
- Import the nodemcu\_rfid\_iot\_projects.sql file. Or else to import an SQL file in PHPMyAdmin, simply click on your database, click “Import -> Choose File -> Click nodemcu\_rfid\_iot\_projects.sql -> Open -> Go”



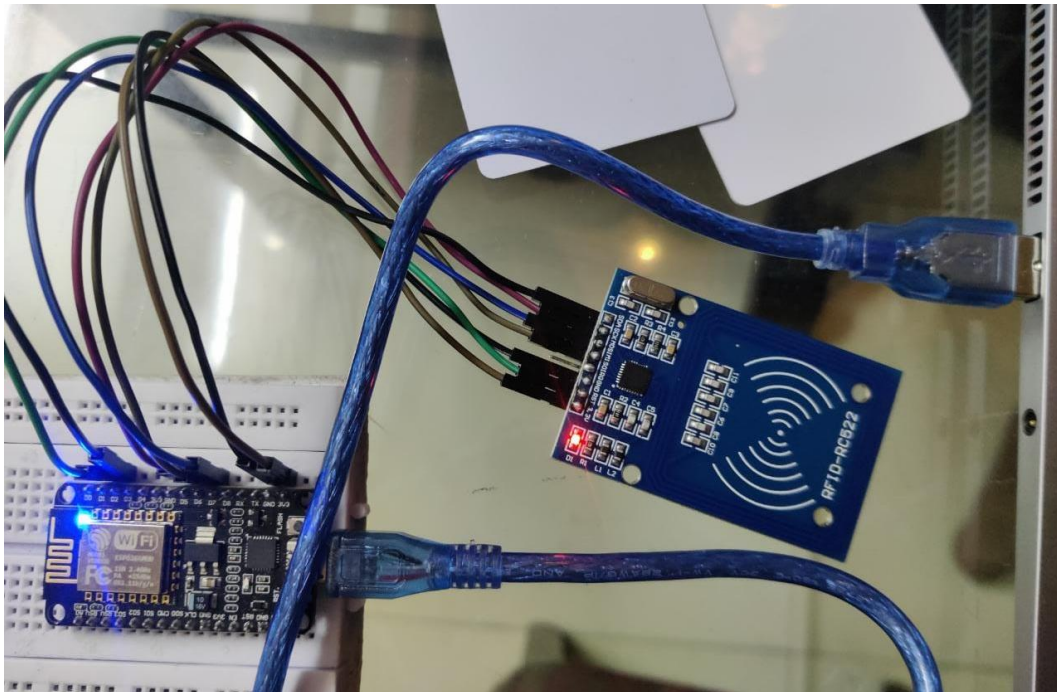
## 7. OUTPUT

Following video is an output of our PNA. Where 1<sup>st</sup> the UID of a card is generated and then using that UID an entry of name Shailesh Agrawal is Registered which can be seen in user data page of our server and we can also edit or delete the registered entry. Then after scan the same card we can see the data of shailesh and trupty in Read tag id section.

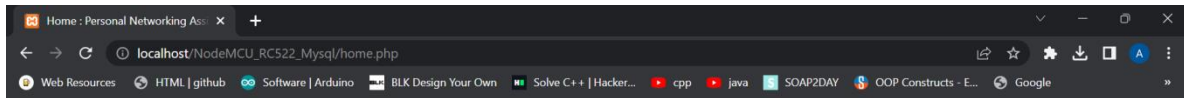


pna.mp4

Mentioned below are the snapshot of hardware serial monitor and hosted server,



```
NodeMCU_V3_RFID_RC522_DB | Arduino IDE 2.0.3
File Edit Sketch Tools Help
NodeMCU 1.0 (ESP-12E...)
NodeMCU_V3_RFID_RC522_DB.ino
32
33 #define SS_PIN D2 //--> SDA / SS is connected to pinout D2
34 #define RST_PIN D1 //--> RST is connected to pinout D1
35 MFRC522 mfrc522(SS_PIN, RST_PIN); //--> Create MFRC522 instance.
36
37 #define ON_Board_LED 2 //--> Defining an On Board LED, used for indicators when the process of connecting to a wifi router
38
39 //-----SSID and Password of your WiFi router-----
40 const char* ssid = "S.S.S.A";
41 const char* password = "air666691";
42 //-----
43
Output Serial Monitor x
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM3') New Line 115200 baud
THE UID OF THE SCANNED CARD IS : 29D275A2
200
THE UID OF THE SCANNED CARD IS : 64ED3622
200
THE UID OF THE SCANNED CARD IS : 64ED3622
200
THE UID OF THE SCANNED CARD IS : 29D275A2
200
Ln 40, Col 26 UTF-8 NodeMCU 1.0 (ESP-12E Module) on COM3 2
```



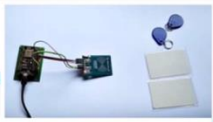
## Personal Networking Assistant



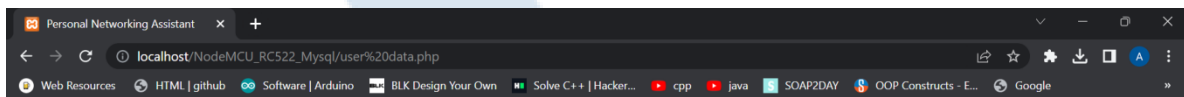
### Personal Networking Assistant

**Registration Form**  
ID:   
Name:   
Gender:   
Email Address:   
Mobile Number:

**User Data**  
ID: 866080F8  
Name: The IoT Projects  
Gender: Male  
Email: ask.theiotprojects@gmail.com  
Mobile Number: 9800988978



Name	ID	Gender	Email	Mobile Number	Action
John	789174F5	Male	john@gmail.com	23456789	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
The IoT Projects	866080F8	Male	ask.theiotprojects@gmail.com	9800988978	<input type="button" value="Edit"/> <input type="button" value="Delete"/>



## Personal Networking Assistant



### User Data Table

Name	ID	Gender	Email	Mobile Number	Action
Shailesh Agrawal	64ED3622	Male	Shaileshhagrawal@gmail.com	9130618111	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Trupti Gajhas	29D275A2	Female	trupti.22010424@vit.ac.in	9788377662	<input type="button" value="Edit"/> <input type="button" value="Delete"/>



Registration : Personal Networki

localhost/NodeMCU\_RC522\_Mysql/registration.php

Web Resources HTML | github Software | Arduino BLK Design Your Own Solve C++ | Hacker... cpp java SOAP2DAY OOP Constructs - E... Google

**Personal Networking Assistant**

Home User Data **Registration** Read Tag ID

**Registration Form**

ID Please Scan your Card / Key

Name

Gender Male

Email Address

Mobile Number

Save

Read Tag : Personal Networking

localhost/NodeMCU\_RC522\_Mysql/read%20tag.php

Web Resources HTML | github Software | Arduino BLK Design Your Own Solve C++ | Hacker... cpp java SOAP2DAY OOP Constructs - E... Google

**Personal Networking Assistant**

Home User Data Registration **Read Tag ID**

**Please Scan Tag to Display ID or User Data**

User Data	
ID	: 29D275A2
Name	: Trupti Gajhas
Gender	: Female
Email	: trupti.22010424@vilit.ac.in
Mobile Number	: 9788377662

## 8. CONCLUSION

Finally we have successfully hosted our local server application provided by localhost with including backend like MySQL, PHP linking Database and frontend like HTML and CSS as a software end and also proper interfacing MFRC522 with ESP8266 Wi-Fi module by defining the proper ports and pins and link the hardware with database by providing personal hotspot to ESP. after implementation of hardware code uploaded through Arduino IDE platform by including all required libraries for reading and writing RFID to the getting of UID of the cards. We also get the count of participants for the conference which is present there, and it is done by taking a count of the entries on our server

## 9. REFERENCE

1. <https://www.atlantis-press.com/article/25836428.pdf>
2. [http://www.cisterlabs.pt/docs/assessing\\_the\\_esp8266\\_wifi\\_module\\_for\\_the\\_internet\\_of\\_things/1438/view.pdf](http://www.cisterlabs.pt/docs/assessing_the_esp8266_wifi_module_for_the_internet_of_things/1438/view.pdf)
3. <https://circuits4you.com/2018/10/03/rfid-reader-rc522-interface-with-nodemcu-using-arduino-ide/>
4. <https://cityos-air.readme.io/docs/esp8266-nodemcu>

The logo for Vishwakarma Institutes features a large, stylized 'V' shape. The left side of the 'V' is a solid light blue triangle. The right side is composed of two overlapping shapes: a light blue chevron pointing downwards and a light red shape above it that resembles a flame or a stylized bird's head. Below the 'V' shape, the words 'VISHWAKARMA' and 'INSTITUTES' are written in a light blue, sans-serif, all-caps font, stacked on two lines.

VISHWAKARMA  
INSTITUTES