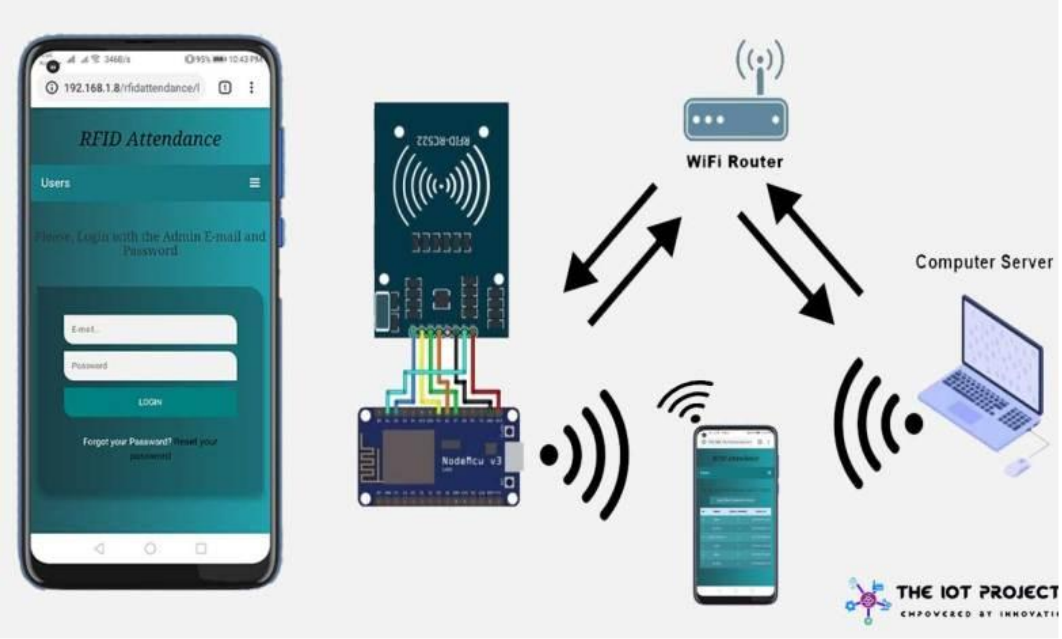
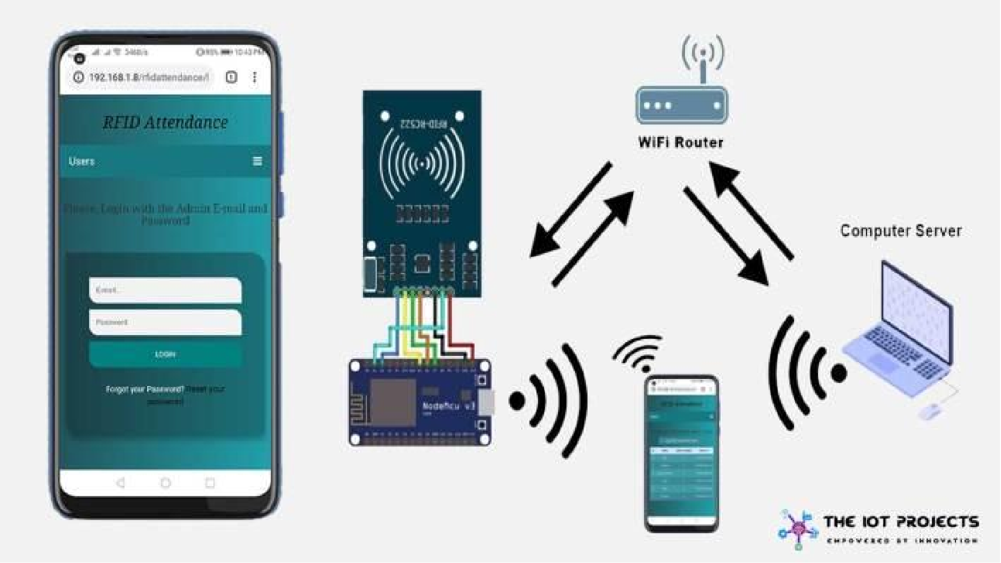
RFID Based Attendance System Using NodeMCU with PHP Web App



RFID Based Attendance System Using NodeMCU with PHP Web App

Overview

The RFID based attendance system is developed using PHP, CSS, and Javascript. An RFID Based Attendance System Using NodeMCU is a modern attendance system. Hence, is a very interesting project. It can be useful in different places like schools, Colleges, industry and private organizations to register the attendance of students, teachers, employees, etc. to tabulate monthly/daily working hours automatically. When the person with the correct RFID card swipes his/her RFID tag, His/Her arrival time will be stored in system Log. Usually, when the same person swipes his/her RFID tag again, the system will save it as his/her leaving time.



RFID Based Attendance System Using NodeMCU with PHP Web App

The RFID attendance system is developed with the IoT platform. We have used NodeMCU ESP8266 development board with MF-RC522 Module to send the card UID to the PHP Web app and store data into the website database. Basically, the admin plays an important role in the management of this system.

Also Read: IoT based RFID Attendance System Using ESP32, OLED Display, and RFID Module

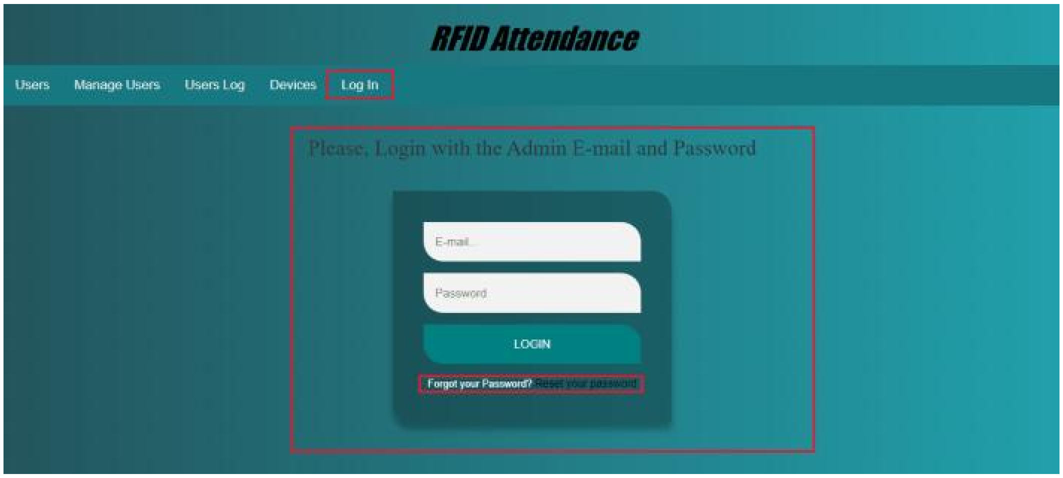
- Home Automation with MIT App Inventor and ESP8266
- IoT based Silent Intruder Alarm using Arduino
- Temperature Controlled Home Automation using Arduino
- IoT Based RFID Smart Door Lock System Using NodeMCU ESp8266
- IoT Based Flood Monitoring System Using NodeMCU & Thingspeak

Features:

```
-Secured Login System
-Admin Panel
-User Entry Log
-User Management System
-Multiple Device Management System
-Filter the Log Data by Date, Time-In, Time-Out, and Department
-Export those filtered data to Excel
```

Functionality Performed by users

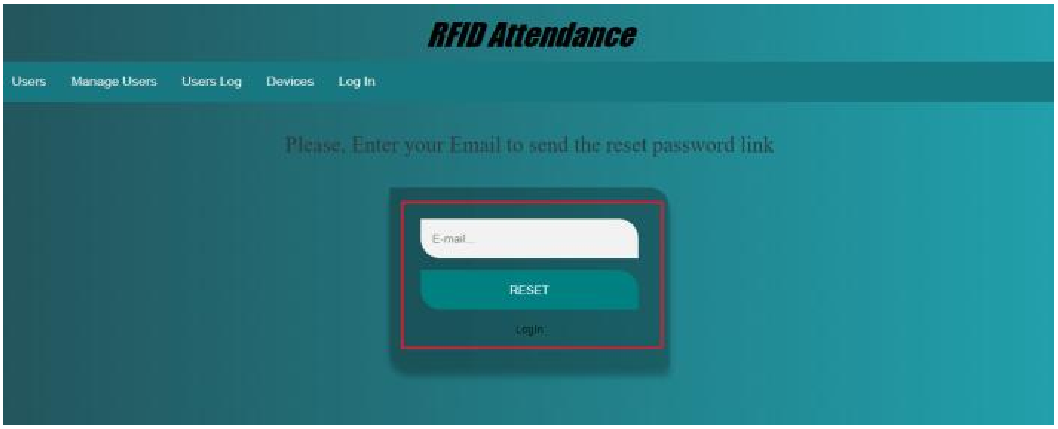
Now, talking about the features of the RFID based attendance system using NodeMCU.The home page displays the admin login page. Unless You login to the system, you won't be able to browse other available options. Hence the system is secured. Basically, the design of the RFID attendance system project is pretty simple. Hence the user won't find any difficulties while enrolling his/her attendance. The user needs to swipe his/her card or a keychain to maintain attendance that includes the entry time as well. The major functions provided to admin are mentioned below:



Please, Login with the Admin E-mail and Password

Admin Panel Login System

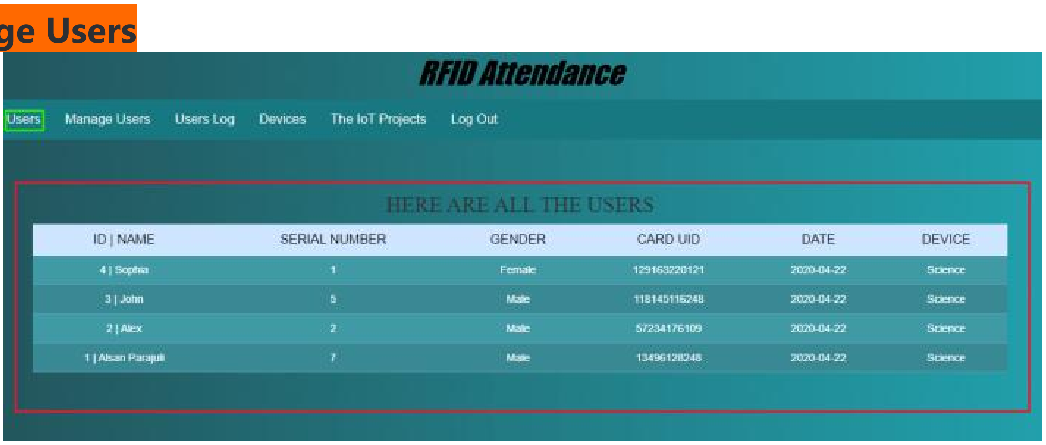
- Admin Login/Logout System
- Forgot Password for admin
- Edit and Update admin profile



Please, Enter your Email to send the reset password link

Reset Admin Account Password

Manage Users

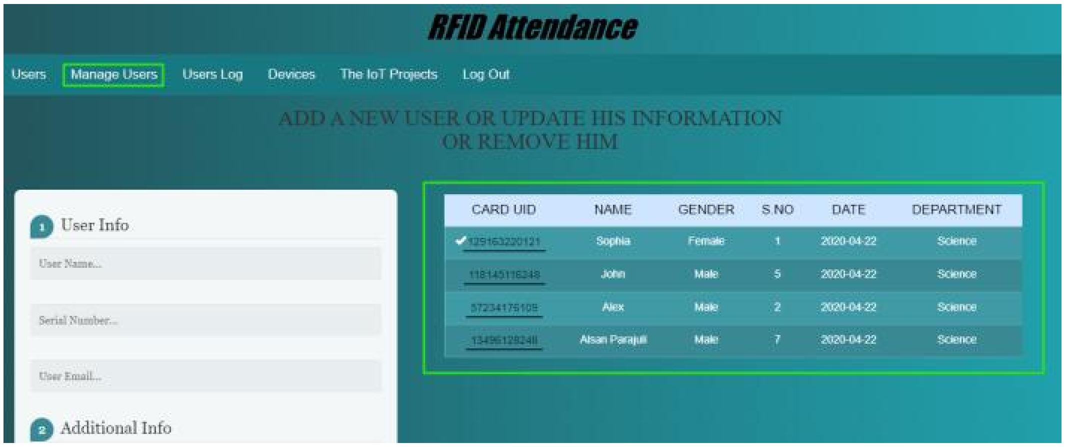


HERE ARE ALL THE USERS

ID | NAME	SERIAL NUMBER	GENDER	CARD UID	DATE	DEVICE
4 | Sophia	1	Female	129163220121	2020-04-22	Science
3 | John	5	Male	118145116248	2020-04-22	Science
2 | Alex	2	Male	57234176109	2020-04-22	Science
1 | Alsan Parajuli	7	Male	13496128248	2020-04-22	Science

View and Manage Users

- View users
- Add New User
- Edit and update the existing users
- Remove Users

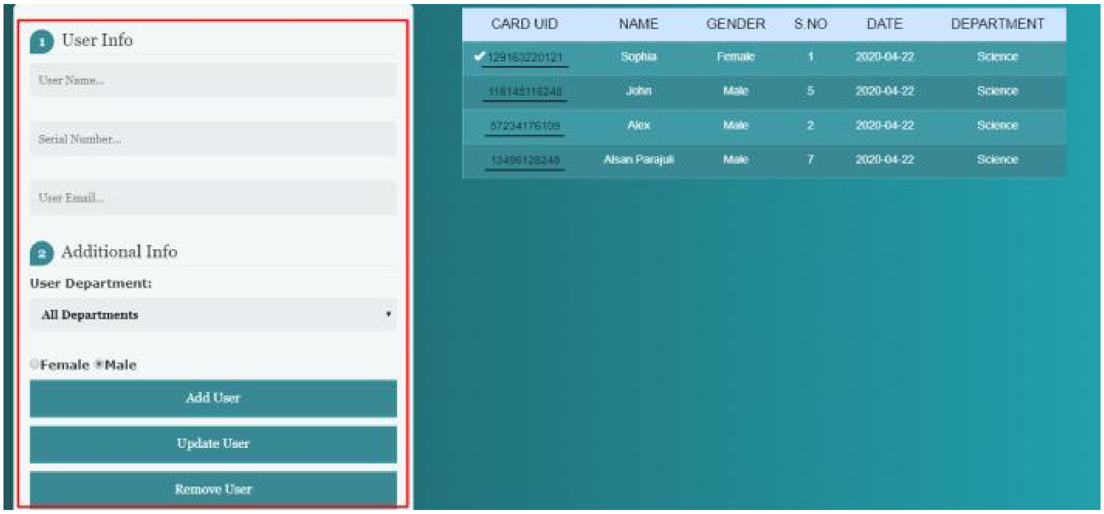


ADD A NEW USER OR UPDATE HIS INFORMATION OR REMOVE HIM

CARD UID	NAME	GENDER	S.NO	DATE	DEPARTMENT
✓ 129163220121	Sophia	Female	1	2020-04-22	Science
118145116248	John	Male	5	2020-04-22	Science
57234176109	Alex	Male	2	2020-04-22	Science
13496128248	Alsan Parajuli	Male	7	2020-04-22	Science

User Management System

From the admin panel, the admin can enroll new users, update and remove users from the user management system. Further, the admin can view all the attendance records.



1 User Info

User Name...

Serial Number...

User Email...

2 Additional Info

User Department:

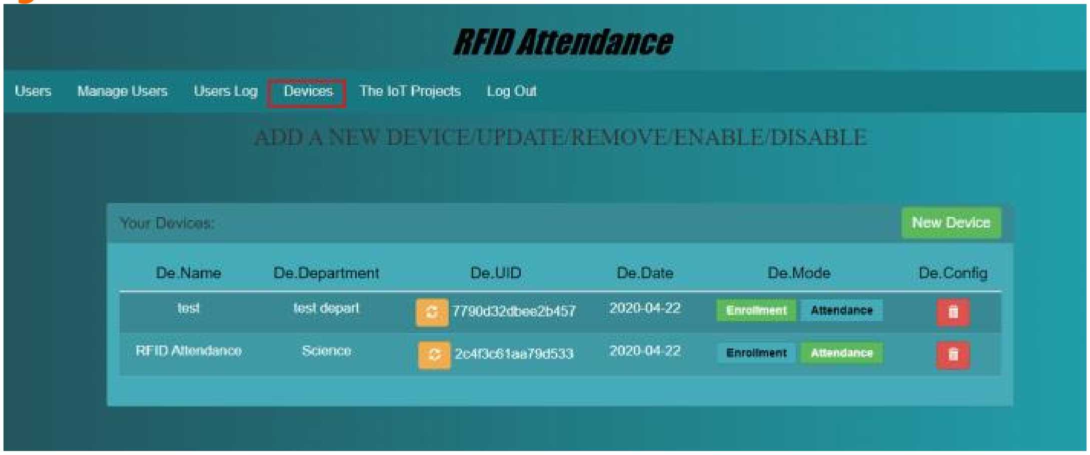
All Departments

○Female ●Male

CARD UID	NAME	GENDER	S.NO	DATE	DEPARTMENT
✓ 129163220121	Sophia	Female	1	2020-04-22	Science
118145116248	John	Male	5	2020-04-22	Science
57234176109	Alex	Male	2	2020-04-22	Science
13496128248	Alsan Parajuli	Male	7	2020-04-22	Science

Enroll New Users/Update/Remove

Manage Device



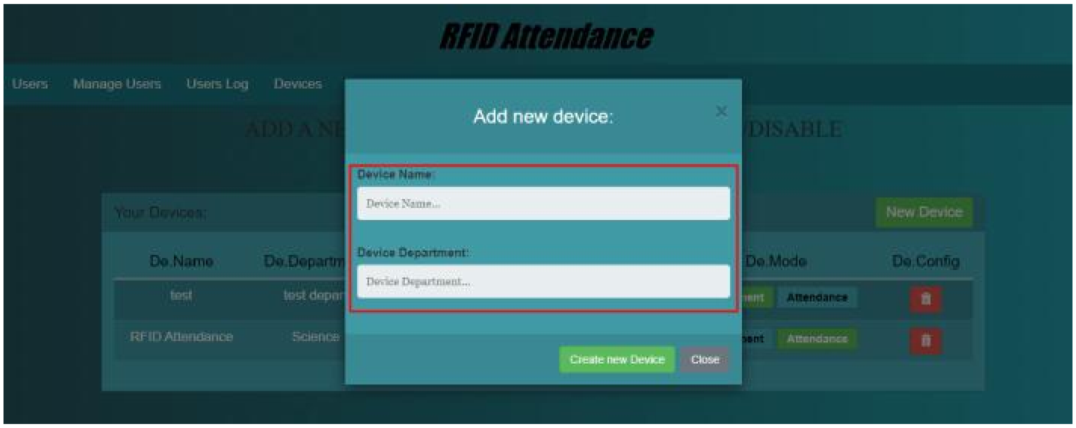
ADD A NEW DEVICE/UPDATE/REMOVE/ENABLE/DISABLE

Your Devices:

De.Name	De.Department	De.UID	De.Date	De.Mode	De.Config
test	test depart	7790d32dbee2b457	2020-04-22	Enrollment Attendance	
RFID Attendance	Science	2c4f3c61aa79d533	2020-04-22	Enrollment Attendance	

Device Management System

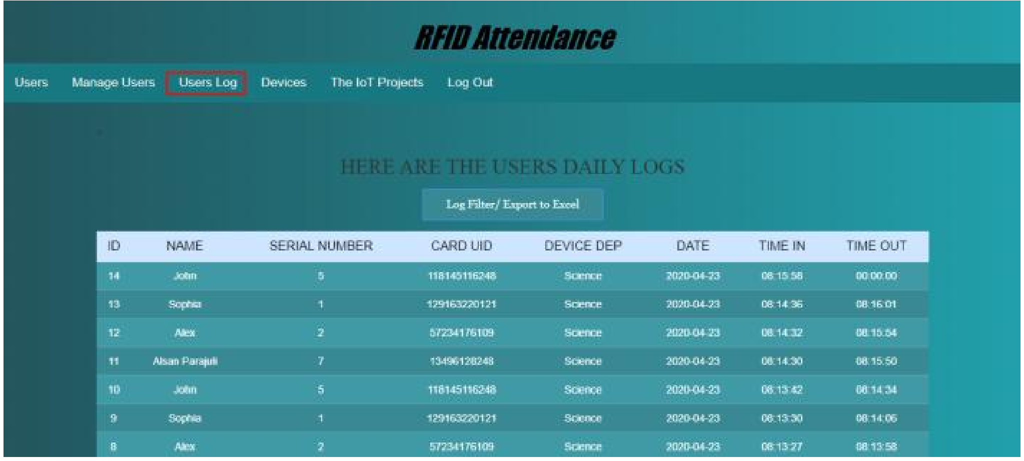
- Add new device
- Update existing device
- Delete device
- Update New token to the device
- Change the device mode (Enrollment mode: to register new users to the system, Attendance Mode: To record attendance of registered users)



Add New RFID Scanner Device

Actually, from the devices section admin can add a new device, update the device, and remove the device. To add a new device you need to enter a device name and its department. Furthermore, you can also update the device token from the device UID Section.

View Users Log

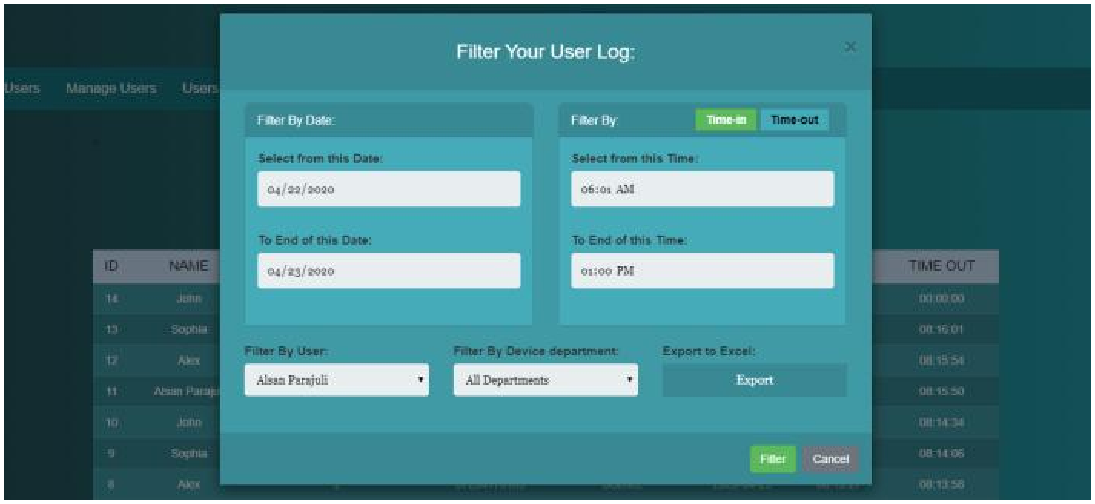


HERE ARE THE USERS DAILY LOGS

ID	NAME	SERIAL NUMBER	CARD UID	DEVICE DEP	DATE	TIME IN	TIME OUT
14	John	5	118145116248	Science	2020-04-23	08:15:58	00:00:00
13	Sophia	1	129163220121	Science	2020-04-23	08:14:36	08:16:01
12	Alex	2	57234176109	Science	2020-04-23	08:14:32	08:15:54
11	Alsan Parajuli	7	13496128248	Science	2020-04-23	08:14:30	08:15:50
10	John	5	118145116248	Science	2020-04-23	08:13:42	08:14:34
9	Sophia	1	129163220121	Science	2020-04-23	08:13:30	08:14:06
8	Alex	2	57234176109	Science	2020-04-23	08:13:27	08:13:58

View Users log on RFID Based Attenance System

From the user log menu, you can navigate to all the user’s logs data. You can view their arrival and Leaving time as well. Furthermore, It has more functions to filter your logs by user, date, arrival time, leaving time, and filter by different departments, etc. Hence, you can also export those data to excel.



Filter Your User Log:

Filter By Date:

Select from this Date: 04/22/2020

To End of this Date: 04/23/2020

Filter By: Time-in

Select from this Time: 06:01 AM

To End of this Time: 01:00 PM

Filter By User: Alsan Parajuli

Filter By Device department: All Departments

Export to Excel:

Filter Users Log to export

Hardware Components Required

S.N	COMPONENTS NAME	DESCRIPTION	QUANTITY	amazon
1	NodeMCU	ESP8266 12E Board	1	https://amzn.to/3mTuL95
2	RFID Module	RFID-RC522 Module	1	https://amzn.to/3pOPpb4
3	Jumper Wires	Male to Male Jumper Wires	4	https://amzn.to/2JWSR44
4	Breadboard	Solderless Breadboard MIni	1	https://amzn.to/3n33uRT

Software Required with Download Links

- **Arduino IDE**
- **XAMPP server**
- **PHP Source Code**
- **RFID-RC522 Library**
- **NodeMcu ESP8266 Library and Board Manager**

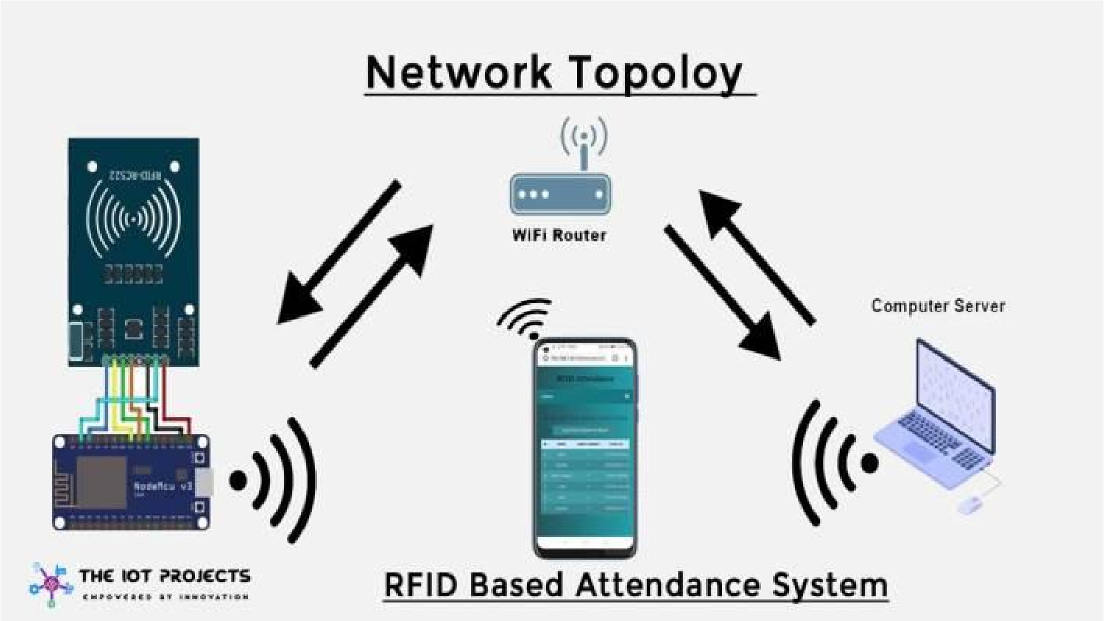
Technology Used in RFID attendance System Using NodeMCU

Here we are listing all the major required software components for this project to develop the user interface.

1. **Embedded C**: All the hardware interface coding has been done in embedded C Language.
2. **PHP**: All the Logics and web interface is written in PHP Language.
3. **MYSQL**: For database, MYSQL Database is used.
4. **Apache 2**: To run the web interface over the Apache 2 server.
5. **Javascript and CSS**: To Style web Interface with animation and validate the form.

RFID Based Attendance System Network Topology

This is the simple network topology designed to run this project over a Local Area Network. The Project Device and Computer server are connected to the same network through WiFi.



RFID Based Attendance System Network Topology

Now before getting started with this project, let’s learn What is RFID? with its specifications and Features.

RFID RC522 Module

What is RFID RC522 Module?

At first, let’s learn a little bit about RFID. RFID is the short form of Radio Frequency Identification. RFID modules use electromagnetic fields for transferring data between the card and the reader. Different RFID tags are attached to objects like Keychain, cards, etc. and whenever we place that object in front of the RFID reader, the reader reads that tags. The next benefit of RFID is that it doesn’t require to be in a straight line to get detected. Unlike a barcode, in RFID there’s no such restriction. So, here are some features of RFID RC522.

RFID MF-RC522 Module

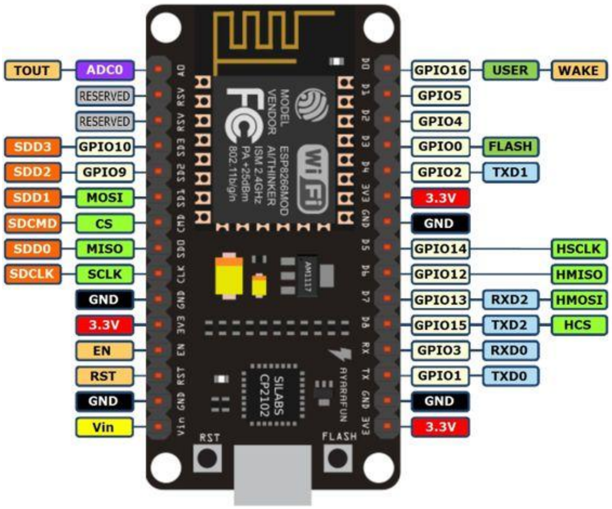
Features:

```
- 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
```

Specifications & Pin Details

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
```



NodeMcu ESP8266 Pinouts

Interfacing RFID RC522 with NodeMcu ESP8266 Module

NodeMCU ESP8266/ESP12E RFID MFRC522 / RC522

D2 <———-> SDA/SS

D5 <———-> SCK

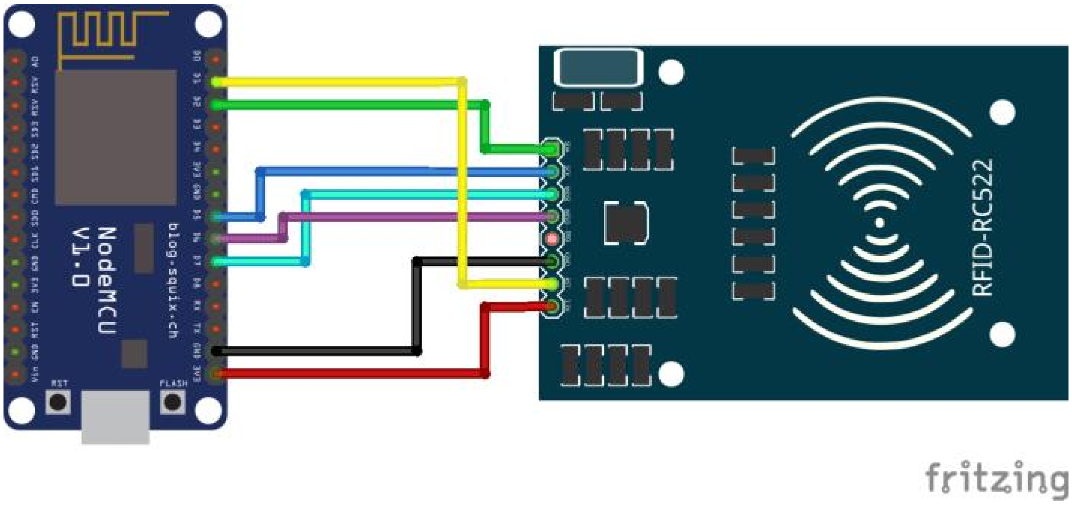
D7 <———-> MOSI

D6 <———-> MISO

GND <———-> GND

D1 <———-> RST

3V/3V3 <———-> 3.3V



Interfacing RFID RC522 with NodeMcu ESP8266 Module

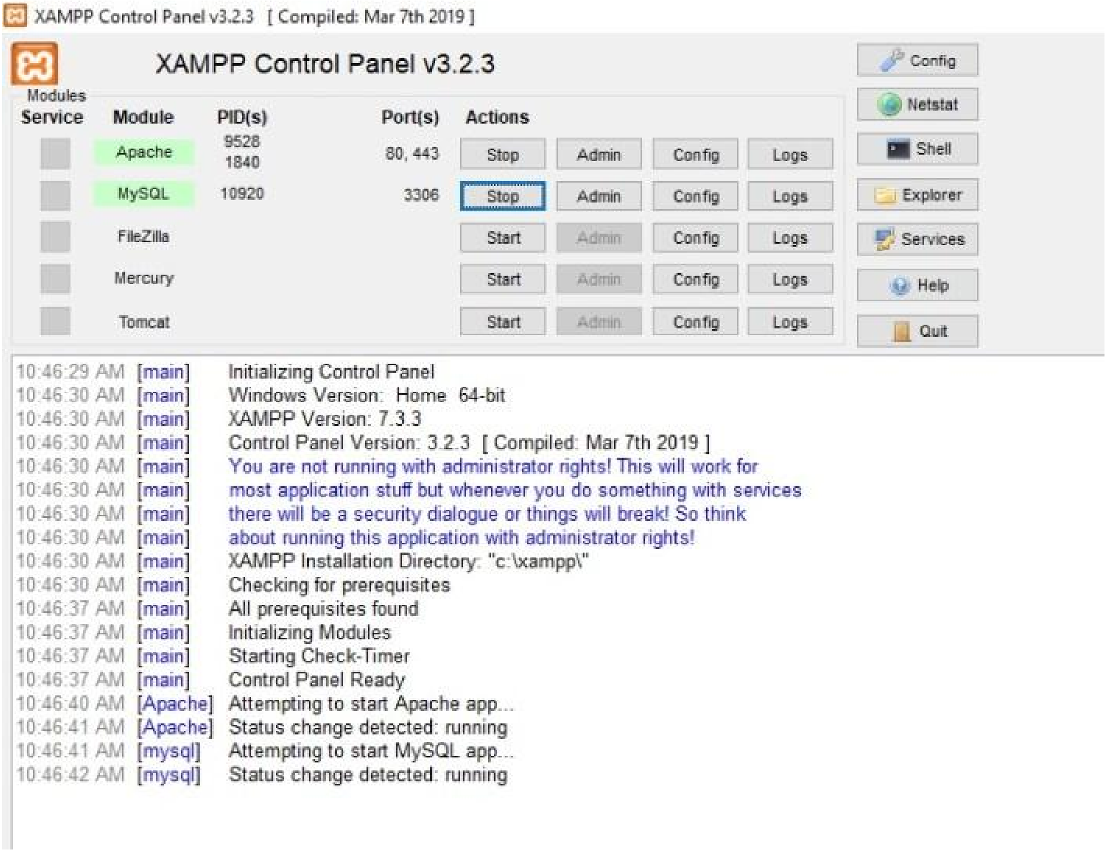
How to run a server?

Actually, we can Deploy this PHP Web App on Hosting Server as well. But, you will need Public IP from Your Broadband connections to send data to the hosting server.

However, we are using the XAMPP server. It can be used both in Windows and Linux. Basically, Ubuntu users can use LAMP it's better than XAMPP. But, I am using Windows so I am going to use the XAMPP server. So you can download the XAMPP server from this link.

Deploying our PHP Web App

- Start your Mysql and Apache from XAMPP Control Panel.



XAMPP Control Panel v3.2.3 [Compiled: Mar 7th 2019]

XAMPP Control Panel v3.2.3

Service	Module	PID(s)	Port(s)	Actions			
	Apache	9528 1840	80, 443	Stop	Admin	Config	Logs
	MySQL	10920	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

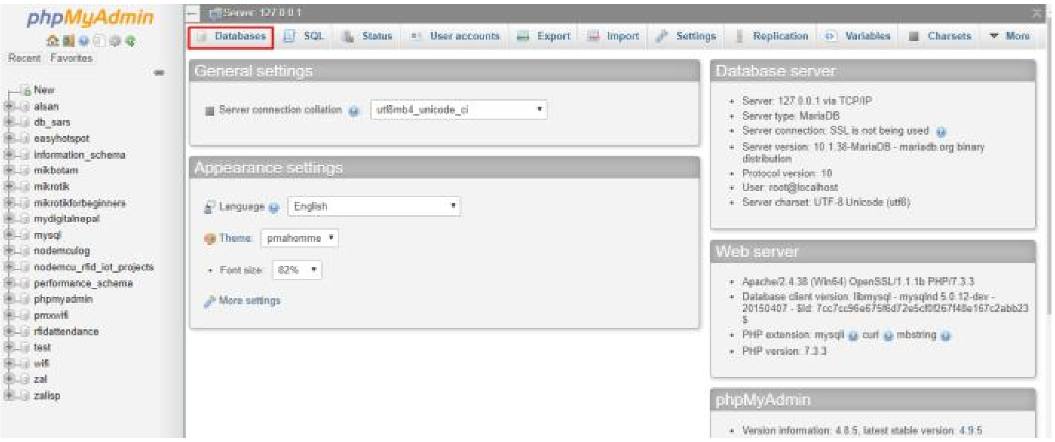
```
10:46:29 AM  [main]    Initializing Control Panel
10:46:30 AM  [main]    Windows Version:  Home  64-bit
10:46:30 AM  [main]    XAMPP Version: 7.3.3
10:46:30 AM  [main]    Control Panel Version: 3.2.3  [ Compiled: Mar 7th 2019 ]
10:46:30 AM  [main]    You are not running with administrator rights! This will work for
10:46:30 AM  [main]    most application stuff but whenever you do something with services
10:46:30 AM  [main]    there will be a security dialogue or things will break! So think
10:46:30 AM  [main]    about running this application with administrator rights!
10:46:30 AM  [main]    XAMPP Installation Directory: "c:\xampp\"
10:46:30 AM  [main]    Checking for prerequisites
10:46:37 AM  [main]    All prerequisites found
10:46:37 AM  [main]    Initializing Modules
10:46:37 AM  [main]    Starting Check-Timer
10:46:37 AM  [main]    Control Panel Ready
10:46:40 AM  [Apache]  Attempting to start Apache app...
10:46:41 AM  [Apache]  Status change detected: running
10:46:41 AM  [mysql]   Attempting to start MySQL app...
10:46:42 AM  [mysql]   Status change detected: running
```

- Download the PHP web app and the Arduino IDE Program code.
- Extract the file.
- Copy the rfidattendance folder (Main Project folder).

- Paste the folder in C:\>xampp/htdocs/ folder.

Now let’s connect the database to the RFID attendance system.

- Open the browser and go to the URL “http://localhost/phpmyadmin/”
- Then click on the database tab.



Server connection collation: utf8mb4_unicode_ci

Language: English

Theme: pmahomme

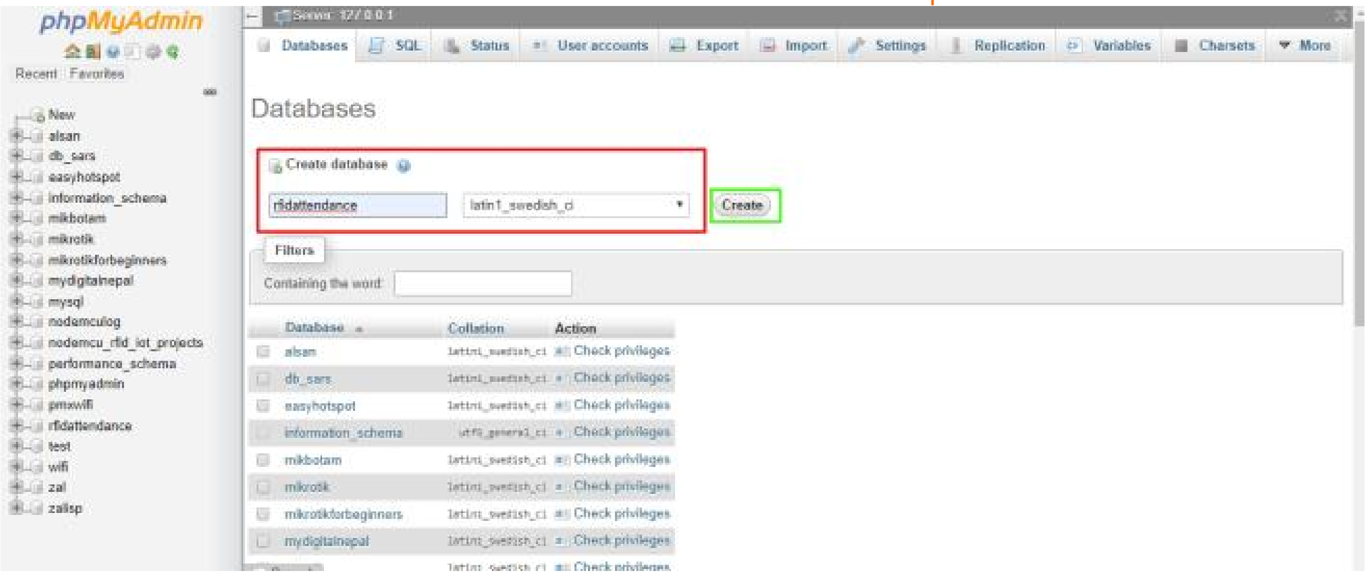
Font size: 82%

- Server: 127.0.0.1 via TCP/IP
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.1.38-MariaDB - mariadb.org binary distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

- Apache/2.4.38 (Win64) OpenSSL/1.1.1b PHP/7.3.3
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - $Id: 7cc7cc96e675f6d72e5cf0f267f48e167c2abb23 $
- PHP extension: mysqli curl mbstring
- PHP version: 7.3.3

- Version information: 4.8.5, latest stable version: 4.9.5

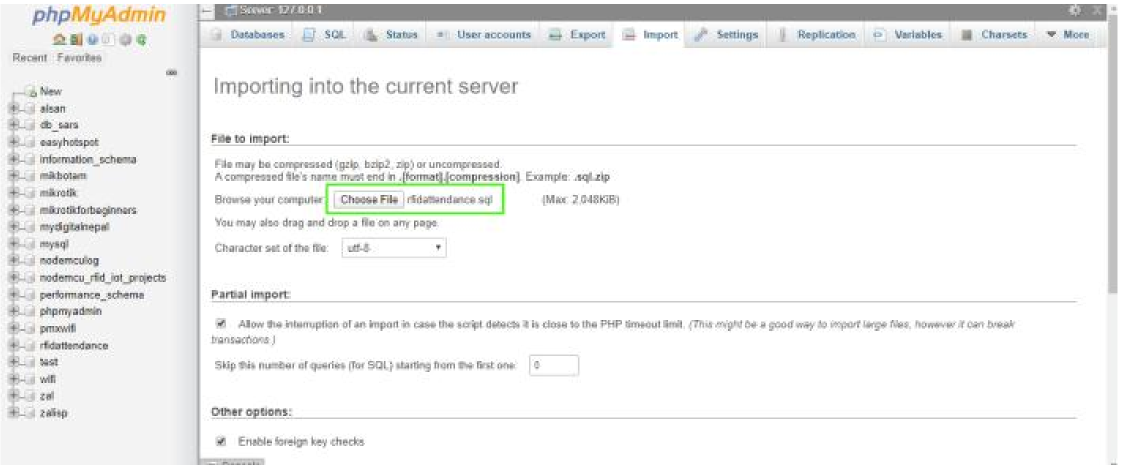
- Create a database name “rfidattendance” and then click on the import tab.



rfidattendance latin1_swedish_ci

Database	Collation	Action
alsan	latin1_swedish_ci	Check privileges
db_sars	latin1_swedish_ci	Check privileges
easyhotspot	latin1_swedish_ci	Check privileges
information_schema	utf8_general_ci	Check privileges
mikbotam	latin1_swedish_ci	Check privileges
mikrotik	latin1_swedish_ci	Check privileges
mikrotikforbeginners	latin1_swedish_ci	Check privileges
mydigitalnepal	latin1_swedish_ci	Check privileges
	latin1_swedish_ci	Check privileges

- Click on the browse file and then choose the “rfidattendance.sql” file located inside the rfidattendance folder.



Importing into the current server

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in .[format].[compression]. Example: .sql.zip

Browse your computer: Choose File rfidattendance.sql (Max: 2,048KiB)

You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

☑ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

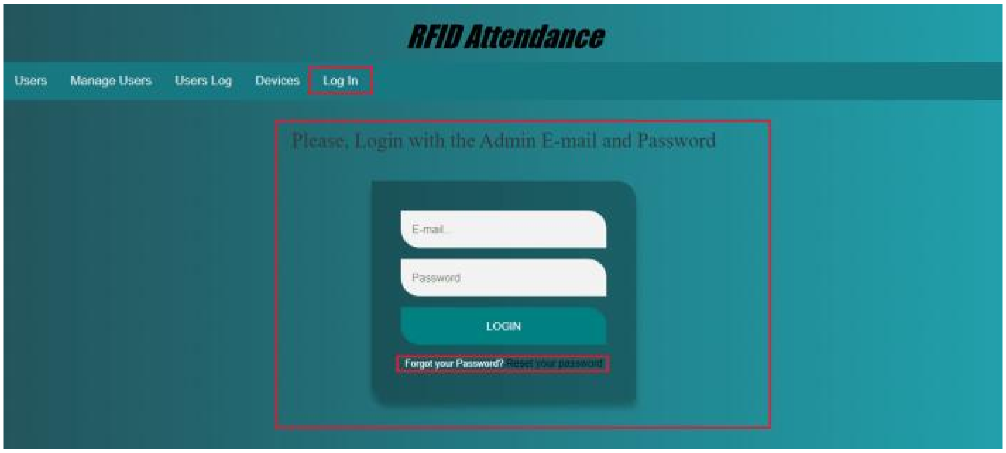
Skip this number of queries (for SQL) starting from the first one: 0

Other options:

☑ Enable foreign key checks

- Click on Go.

After creating a database, open a browser and go to the URL "http://localhost/rfidattendance/". Now, the admin login panel will be displayed.



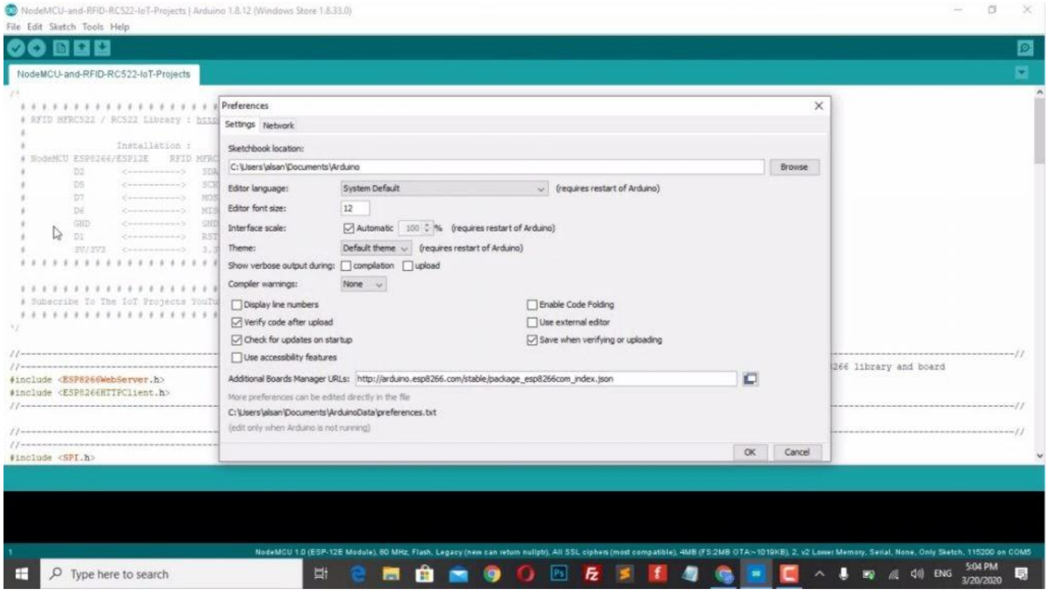
Please, Login with the Admin E-mail and Password

Admin Panel Login System

- Enter the admin email and password to enter the system.
- The default admin credentials is: email= admin@gmail.com, and password= 123.

Setting Up Arduino IDE for NodeMCU ESP8266

We need to add the esp8266 board to our Arduino IDE. Open up your IDE then go to "File -> Preferences" or simply hit "Ctrl + comma".


NodeMCU-and-RFID-RC522-IoT-Projects | Arduino 1.8.12 (Windows Store 1.8.33.0)
File Edit Sketch Tools Help
NodeMCU-and-RFID-RC522-IoT-Projects
Type here to search

Preferences

Settings | Network

Sketchbook location: C:\Users\alsan\Documents\Arduino [Browse]

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

□ Display line numbers □ Enable Code Folding

☑ Verify code after upload □ Use external editor

☑ Check for updates on startup ☑ Save when verifying or uploading

□ Use accessibility features

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file

C:\Users\alsan\Documents\ArduinoData\preferences.txt

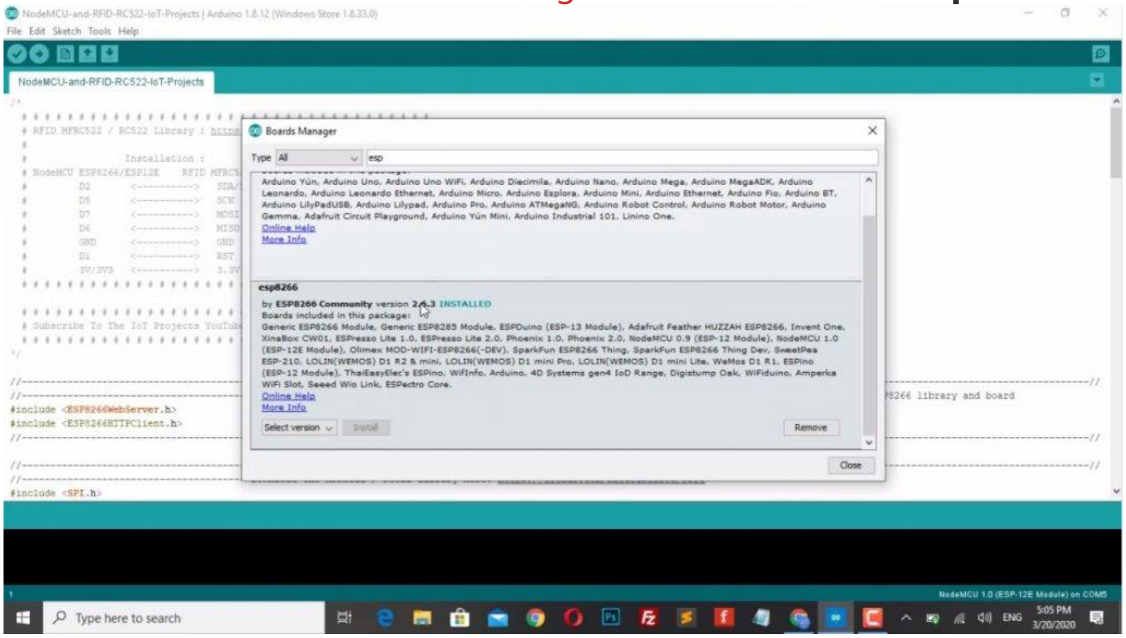
(edit only when Arduino is not running)

[OK] [Cancel]

NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Legacy (new can return nullptr), All SSL ciphers (most compatible), 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Serial, None, Only Sketch, 115200 on COM5

Paste http://arduino.esp8266.com/stable/package_esp8266com_index.json in additional board manager URLs.

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


NodeMCU-and-RFID-RC522-IoT-Projects | Arduino 1.8.12 (Windows Store 1.8.33.0)
File Edit Sketch Tools Help
NodeMCU-and-RFID-RC522-IoT-Projects
Type here to search

Boards Manager

Type All | esp

Arduino Yún, Arduino Uno, Arduino Uno WiFi, Arduino Diecimila, Arduino Nano, Arduino Mega, Arduino MegaADK, Arduino Leonardo, Arduino Leonardo Ethernet, Arduino Micro, Arduino Esplora, Arduino Mini, Arduino Ethernet, Arduino Fio, Arduino BT, Arduino LilyPadUSB, Arduino Lilypad, Arduino Pro, Arduino ATMegaNG, Arduino Robot Control, Arduino Robot Motor, Arduino Gemma, Adafruit Circuit Playground, Arduino Yún Mini, Arduino Industrial 101, Linino One.
Online Help
More Info

esp8266
by **ESP8266 Community** version **2.6.3** INSTALLED
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online Help
More Info

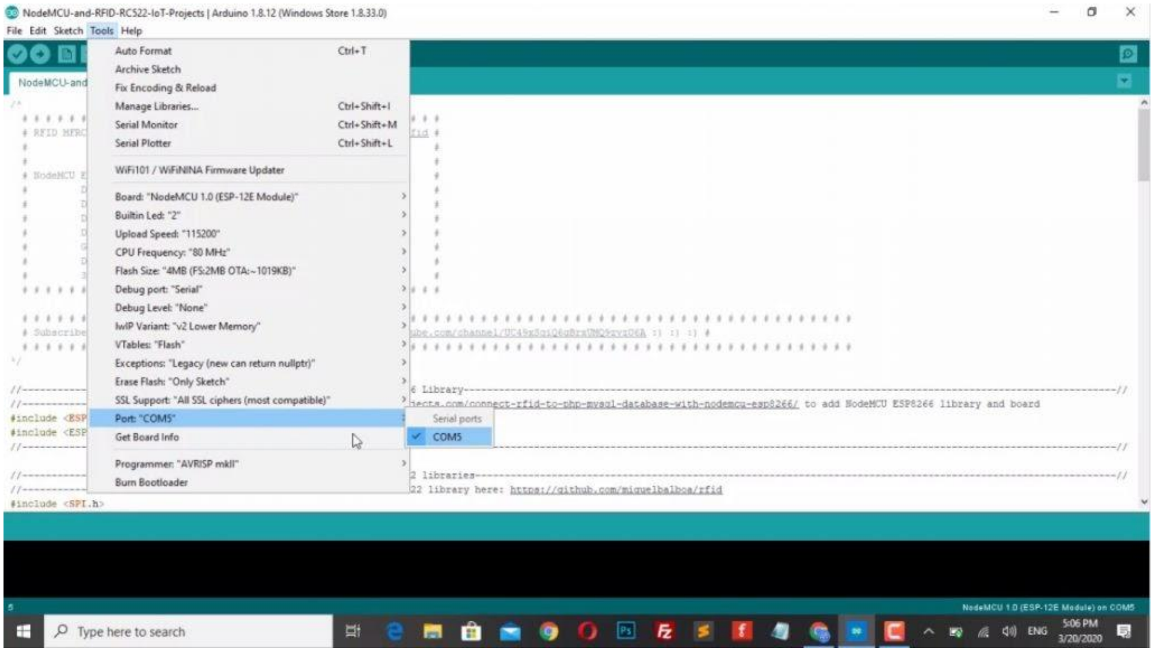
[Select version] [Install] [Remove]

[Close]

NodeMCU 1.0 (ESP-12E Module) on COM5

Selecting Port and Board

Go to Tools -> Port and select the COM Port of your NodeMCU. If you are not sure what port your NodeMcu is plugged in, go to Device Manager -> Ports (COM & LPT).



NodeMCU-and-RFID-RC522-IoT-Projects | Arduino 1.8.12 (Windows Store 1.8.33.0)
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Burn Bootloader
Serial ports
Type here to search

Board: "NodeMCU 1.0 (ESP-12E Module)"
Builtin Led: "2"
Upload Speed: "115200"
CPU Frequency: "80 MHz"
Flash Size: "4MB (FS:2MB OTA:~1019KB)"
Debug port: "Serial"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Exceptions: "Legacy (new can return nullptr)"
Erase Flash: "Only Sketch"
SSL Support: "All SSL ciphers (most compatible)"
Port: "COM5"
✓ COM5
Programmer: "AVRISP mkII"

NodeMCU 1.0 (ESP-12E Module) on COM5

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

Program/Sketch Code

```
//*******************************libraries********************************
//RFID-----------------------------
#include <SPI.h>
#include <MFRC522.h>
//NodeMCU--------------------------
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
//************************************************************************
#define SS_PIN  D2  //D2
#define RST_PIN D1  //D1
//************************************************************************
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
//************************************************************************
/* Set these to your desired credentials. */
const char *ssid = "Alsan Air WiFi 4";
const char *password = "11122235122@kap1";
const char* device_token  = "2c4f3c61aa79d533";
//************************************************************************
String URL = "http://192.168.1.8/rfidattendance/getdata.php"; //computer IP or the
server domain
String getData, Link;
String OldCardID = "";
unsigned long previousMillis = 0;
//*************************************************************************
void setup() {
  delay(1000);
  Serial.begin(115200);
  SPI.begin();  // Init SPI bus
  mfrc522.PCD_Init(); // Init MFRC522 card
  //---------------------------------------------
  connectToWiFi();
}
//*************************************************************************
void loop() {
  //check if there's a connection to Wi-Fi or not
  if(!WiFi.isConnected()){
    connectToWiFi();    //Retry to connect to Wi-Fi
  }
  //---------------------------------------------
  if (millis() - previousMillis >= 15000) {
    previousMillis = millis();
    OldCardID="";
  }
  delay(50);
  //---------------------------------------------
  //look for new card
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;//got to start of loop if there is no card present
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;//if read card serial(0) returns 1, the uid struct contians the ID of the
read card.
  }
  String CardID ="";
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    CardID += mfrc522.uid.uidByte[i];
  }
  //---------------------------------------------
  if( CardID == OldCardID ){
    return;
  }
  else{
    OldCardID = CardID;
  }
  //---------------------------------------------
//  Serial.println(CardID);
  SendCardID(CardID);
  delay(1000);
}
//************send the Card UID to the website*************
void SendCardID( String Card_uid ){
  Serial.println("Sending the Card ID");
  if(WiFi.isConnected()){
    HTTPClient http;    //Declare object of class HTTPClient
    //GET Data
    getData = "?card_uid=" + String(Card_uid) + "&device_token=" +
String(device_token); // Add the Card ID to the GET array in order to send it
    //GET methode
    Link = URL + getData;
    http.begin(Link); //initiate HTTP request   //Specify content-type header

    int httpCode = http.GET();   //Send the request
    String payload = http.getString();    //Get the response payload

//    Serial.println(Link);   //Print HTTP return code
    Serial.println(httpCode);   //Print HTTP return code
    Serial.println(Card_uid);     //Print Card ID
    Serial.println(payload);    //Print request response payload

    if (httpCode == 200) {
      if (payload.substring(0, 5) == "login") {
        String user_name = payload.substring(5);
    //  Serial.println(user_name);

      }
      else if (payload.substring(0, 6) == "logout") {
        String user_name = payload.substring(6);
    //  Serial.println(user_name);

      }
      else if (payload == "succesful") {

      }
      else if (payload == "available") {

      }
      delay(100);
      http.end();  //Close connection
    }
  }
}
//********************connect to the WiFi******************
void connectToWiFi(){
    WiFi.mode(WIFI_OFF);        //Prevents reconnection issue (taking too long to
connect)
    delay(1000);
    WiFi.mode(WIFI_STA);
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("");
    Serial.println("Connected");
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());  //IP address assigned to your ESP

    delay(1000);
}
//=======================================================================
```

Program Code Explanation

Include RFID Library

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

Include NodeMCU ESP8266 Library files.

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
```

Define RFID MF-RC522 Module Pin.

```
#define SS_PIN  D2  //D2
#define RST_PIN D1  //D1
```

Create MFRC522 instance.

```
MFRC522 mfrc522(SS_PIN, RST_PIN);
```

Set your desired WiFi credentials.

```
const char *ssid = "Alsan Air WiFi 4";
const char *password = "11122235122@kap1";
```

Copy the Device token from devices section and paste it over here.

```
const char* device_token  = "2c4f3c61aa79d533";
```

Place your computer IP or the server domain.

```
String URL = "http://192.168.1.8/rfidattendance/getdata.php";
```

Init SPI bus and MFRC522 card

```
SPI.begin();
mfrc522.PCD_Init();
```

check if there’s a connection to Wi-Fi or not.

```
if(!WiFi.isConnected())
```

Retry to connect to Wi-Fi

```
 connectToWiFi();
```

look for new card

```
if ( ! mfrc522.PICC_IsNewCardPresent()) {
```

go to start of loop if there is no card present

```
if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
```

Select one of the cards. If read card serial(0) returns 1, the UID struct contains the ID of the read card.

```
if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
```

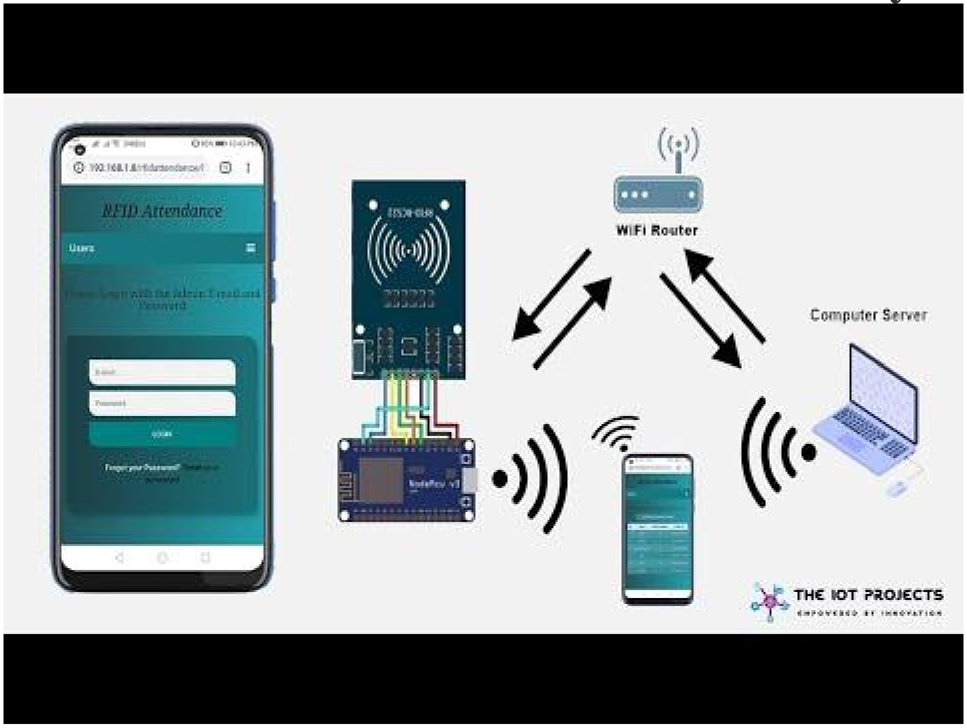
Send the Card UID to the website.

```
void SendCardID( String Card_uid ){
  Serial.println("Sending the Card ID");
  if(WiFi.isConnected()){
    HTTPClient http;    //Declare object of class HTTPClient
    //GET Data
    getData = "?card_uid=" + String(Card_uid) + "&device_token=" +
String(device_token); // Add the Card ID to the GET array in order to send it
    //GET methode
    Link = URL + getData;
    http.begin(Link); //initiate HTTP request   //Specify content-type header

    int httpCode = http.GET();   //Send the request
    String payload = http.getString();    //Get the response payload

//    Serial.println(Link);   //Print HTTP return code
    Serial.println(httpCode);   //Print HTTP return code
    Serial.println(Card_uid);     //Print Card ID
    Serial.println(payload);    //Print request response payload

    if (httpCode == 200) {
      if (payload.substring(0, 5) == "login") {
        String user_name = payload.substring(5);
    //  Serial.println(user_name);

      }
      else if (payload.substring(0, 6) == "logout") {
        String user_name = payload.substring(6);
    //  Serial.println(user_name);

      }
      else if (payload == "succesful") {

      }
      else if (payload == "available") {

      }
      delay(100);
      http.end();  //Close connection
    }
  }
}
```

connect to the WiFi

```
void connectToWiFi(){
    WiFi.mode(WIFI_OFF);        //Prevents reconnection issue (taking too long to
connect)
    delay(1000);
    WiFi.mode(WIFI_STA);
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("");
    Serial.println("Connected");

    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());  //IP address assigned to your ESP

    delay(1000);
}
```

Video Tutorial of RFID Based Attendance System



Thank You so much Electronics Tech Channel for this awesome Project. Please help him to grow his channel. Your one subscribe can motivate him to create more projects like this on future.

https://theiotprojects.com/rfid-based-attendance-system-using-nodemcu/

https://www.youtube.com/watch?v=SKxY2qjOPYQ