RFID-Based Authentication System with Real-time database Server Connectivity

Made by – Jaydeep Gupta

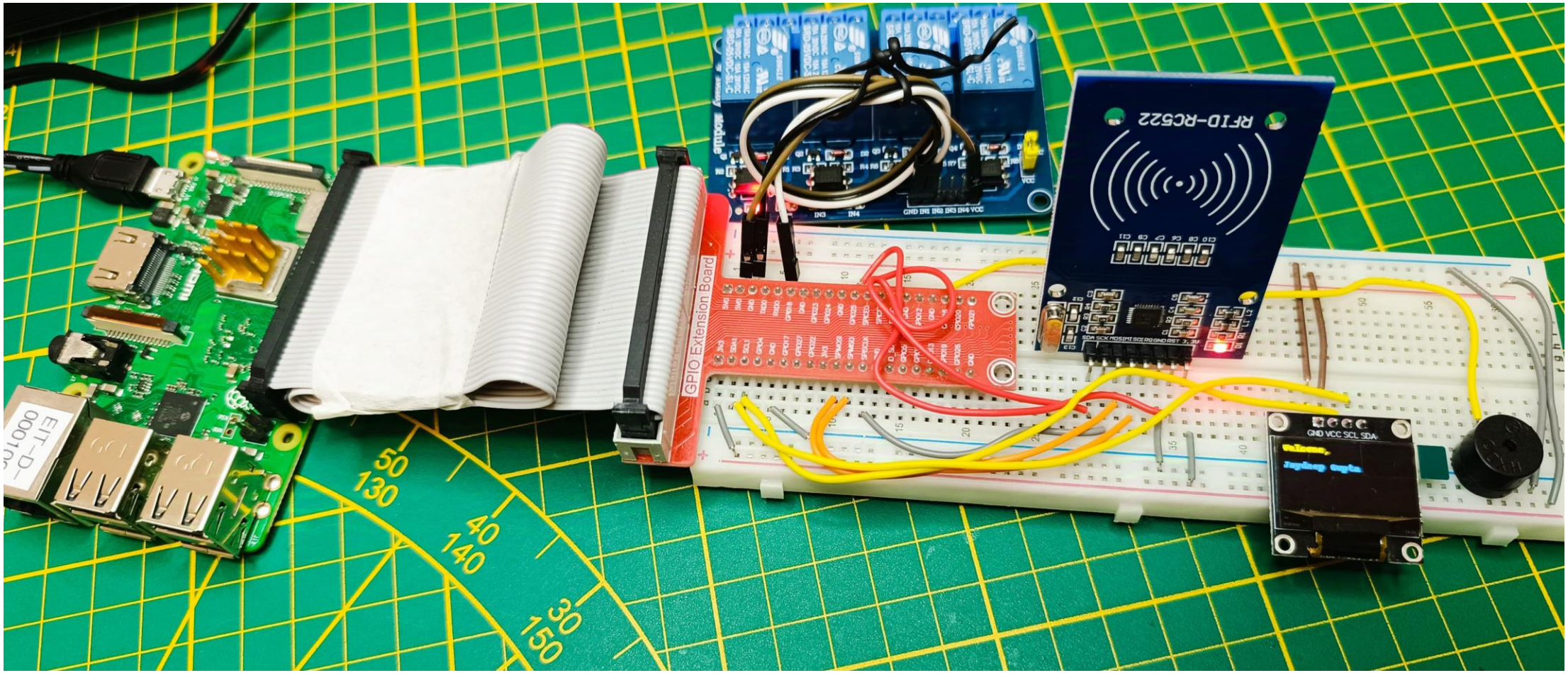
Subject- Applied Network and Bus Technology for Automation & Control
2024 summer semester

Course of study- Masters in Commercial Vehicle Technology

University-RPTU Kaiserslautern

Guided by Prof. Dr.-Ing. Christian De Schryver

How it looks



Introduction

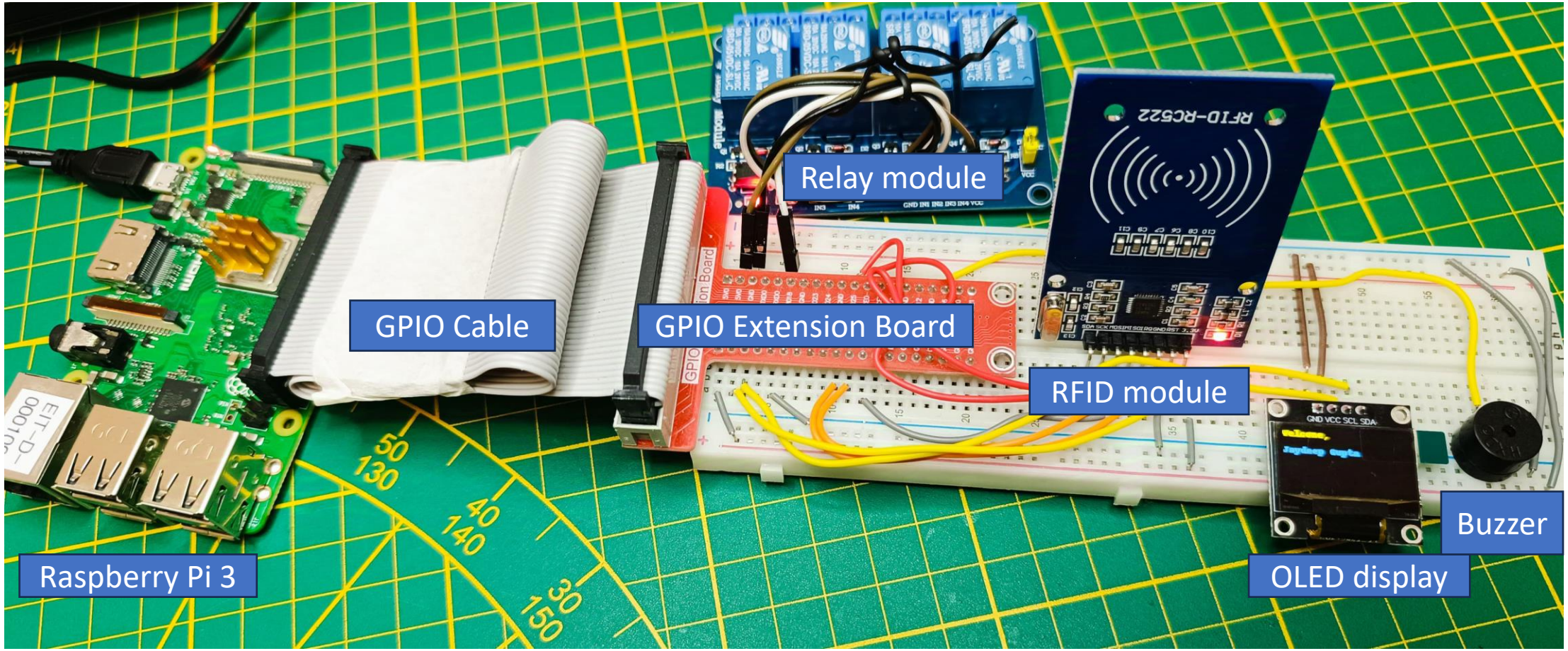
- Overview of the project:
  - This project integrates RFID technology with real time server-based authentication.
  - It uses Raspberry Pi as a computer to control a relay and buzzer based on access permissions.
- Real –world applications:
  - Can be used for door locks, turnstiles, and attendance systems.

Task and Motivation

- **Task**:
  To develop an RFID-based authentication system that uses RFID cards for access control, integrated with an OLED display for feedback and connected to a real-time database server for logging and verification.
- **Motivation**:
  Access control systems are essential in securing restricted areas. Combining RFID technology with real-time server connectivity enhances security and allows for centralized monitoring. This project provides hands-on experience with IoT technologies, embedded systems, and real-time communication, aligning with modern access control needs.

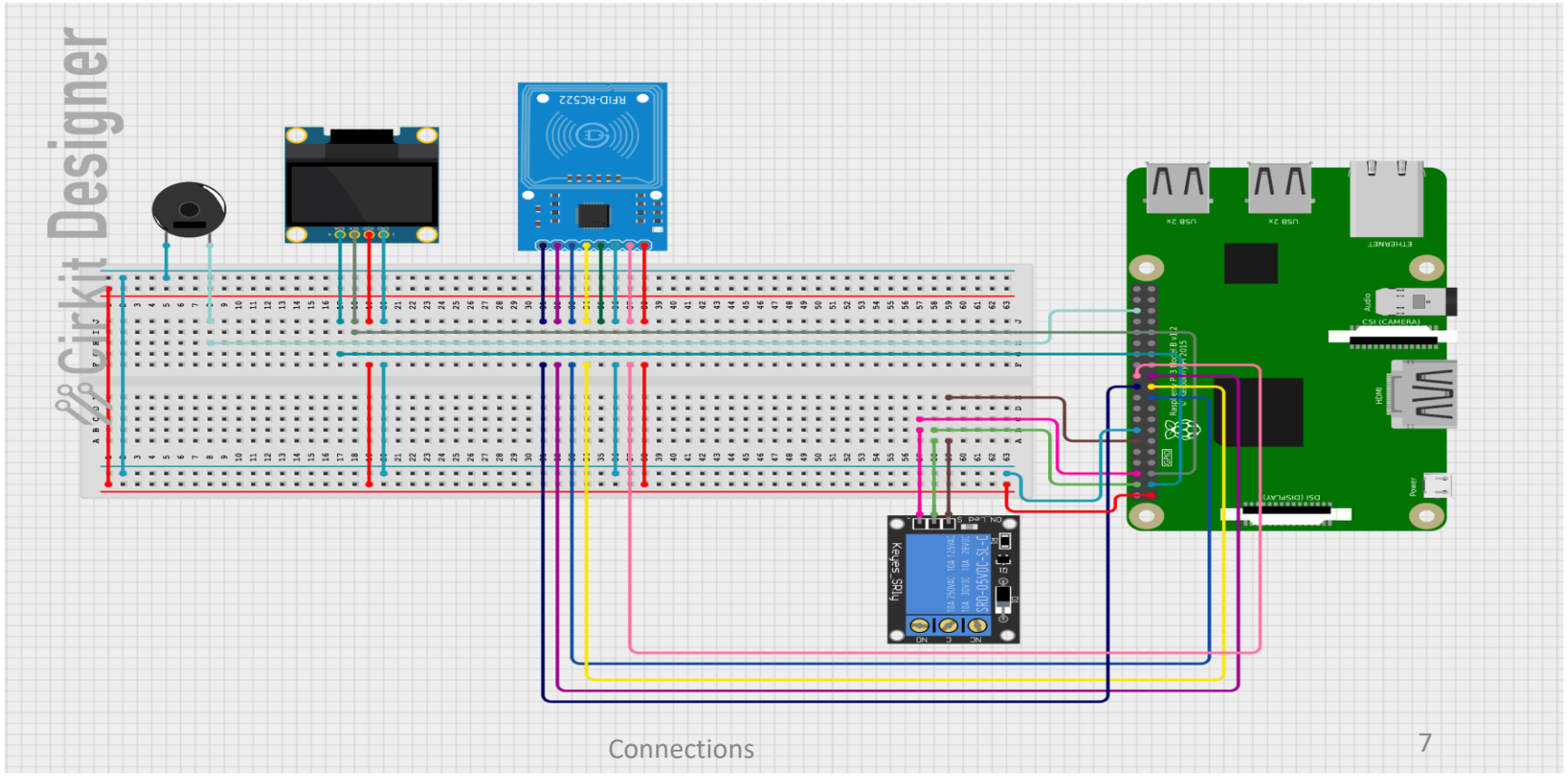
Hardware Used

- List of components:
  - Raspberry Pi 3
    - Video Capture card(to use Raspberry Pi board in OBS software)
    - Keyboard
    - Mouse
    - Power supply
  - RFID reader (MFRC522)
  - Relay module
  - Buzzer
  - OLED display (SSD1306)
  - GPIO Extension Board with cable
  - Breadboard, jumper wires, hookup wires

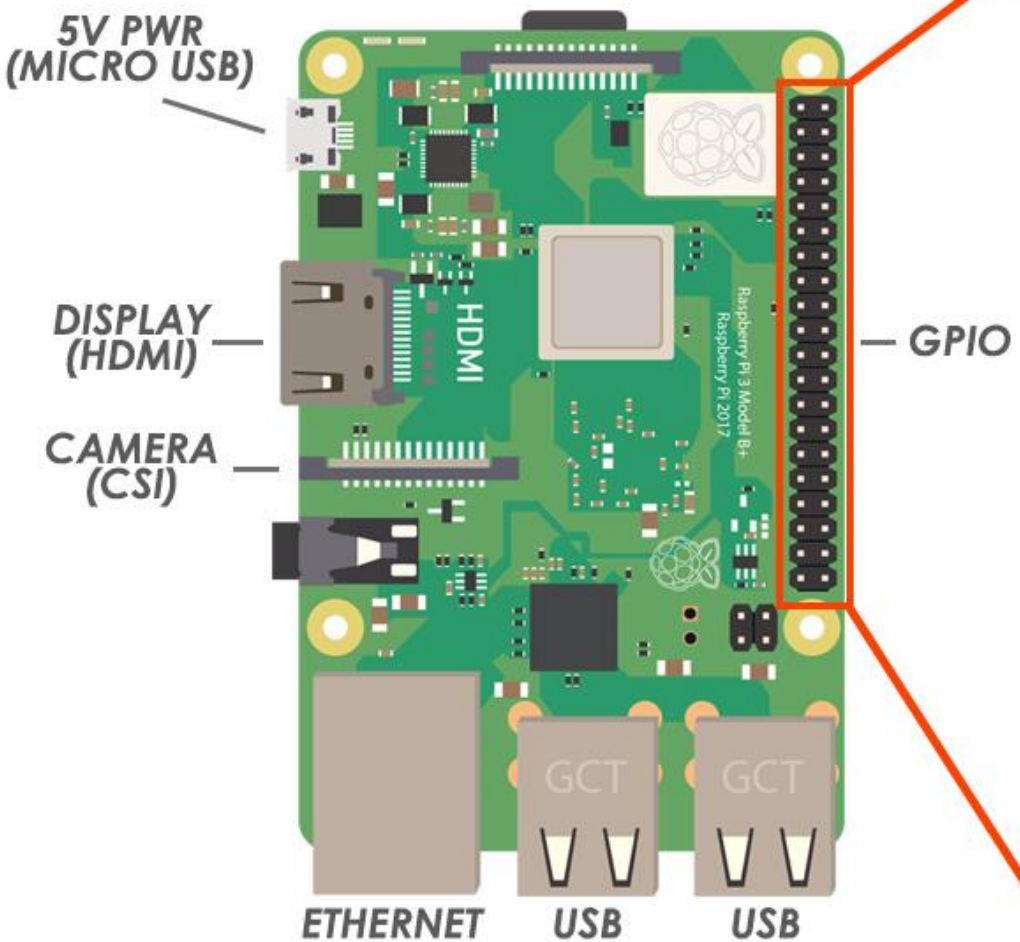

Relay module
GPIO Cable
GPIO Extension Board
RFID module
Buzzer
OLED display
Raspberry Pi 3

Set Up the Hardware

1. **Connect the RC522 RFID Reader**:
    1. Use jumper wires to connect the RFID reader to the Raspberry Pi's GPIO pins:
        1. SDA → GPIO 8 (Pin 24)
        2. SCK → GPIO 11 (Pin 23)
        3. MOSI → GPIO 10 (Pin 19)
        4. MISO → GPIO 9 (Pin 21)
        5. GND → Ground (Pin 6)
        6. RST → GPIO 25 (Pin 22)
        7. 3.3V → 3.3V (Pin 1)
2. **Connect the OLED Display**:
    1. Connect the I2C pins:
        1. VCC → 3.3V (Pin 1)
        2. GND → Ground (Pin 6)
        3. SCL → GPIO 3 (Pin 5)
        4. SDA → GPIO 2 (Pin 3)
3. **Connect the Relay Module**:
    1. Signal → GPIO 18 (Pin 12)
    2. VCC → 5V (Pin 2)
    3. GND → GND (Pin 6)
4. **Connect the Buzzer (Optional)**:
    1. Signal → GPIO 18 (Pin 12)
    2. GND → Ground (Pin 6)



Connections



GPIO

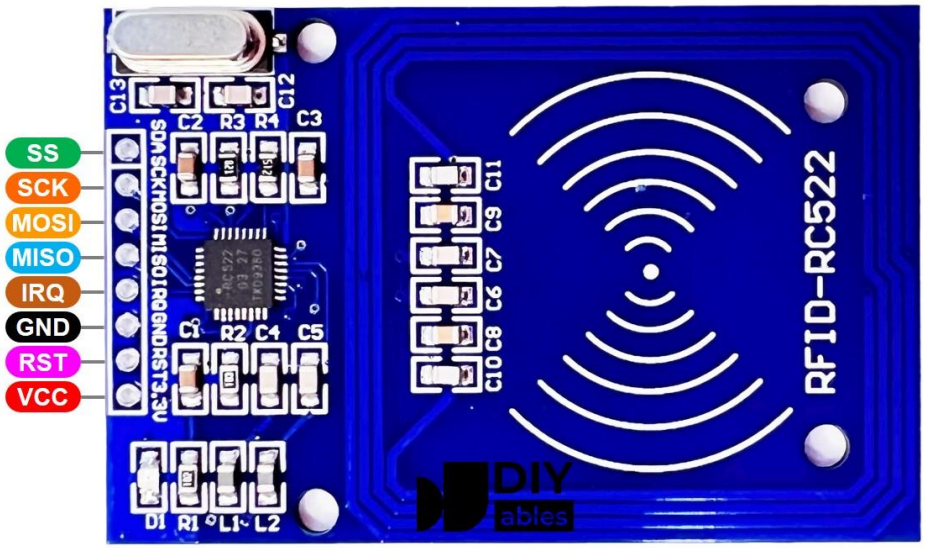
YoungWonks

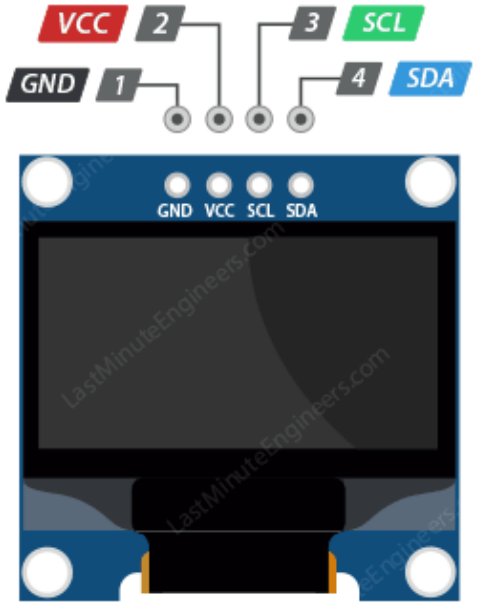
Software

- Thonny with Python(Programming Language)
- MySQL (Database)
- Raspbian OS (Legacy 32-bit)
- Cron (to run .py file on every reboot)
  - @reboot sleep 15 && python /home/pi/Python/Rfid_project6,5.py
  - @reboot- cmd use to auto start .py file
  - Sleep 15- cmd delays .py file with 15 secs before running to allow proper connection of wifi and server

Protocols

- SPI (Serial Peripheral Interface) for the RFID Reader
- I2C (Inter-Integrated Circuit) for the OLED Display
- HTTP/HTTPS for Database Server Communication

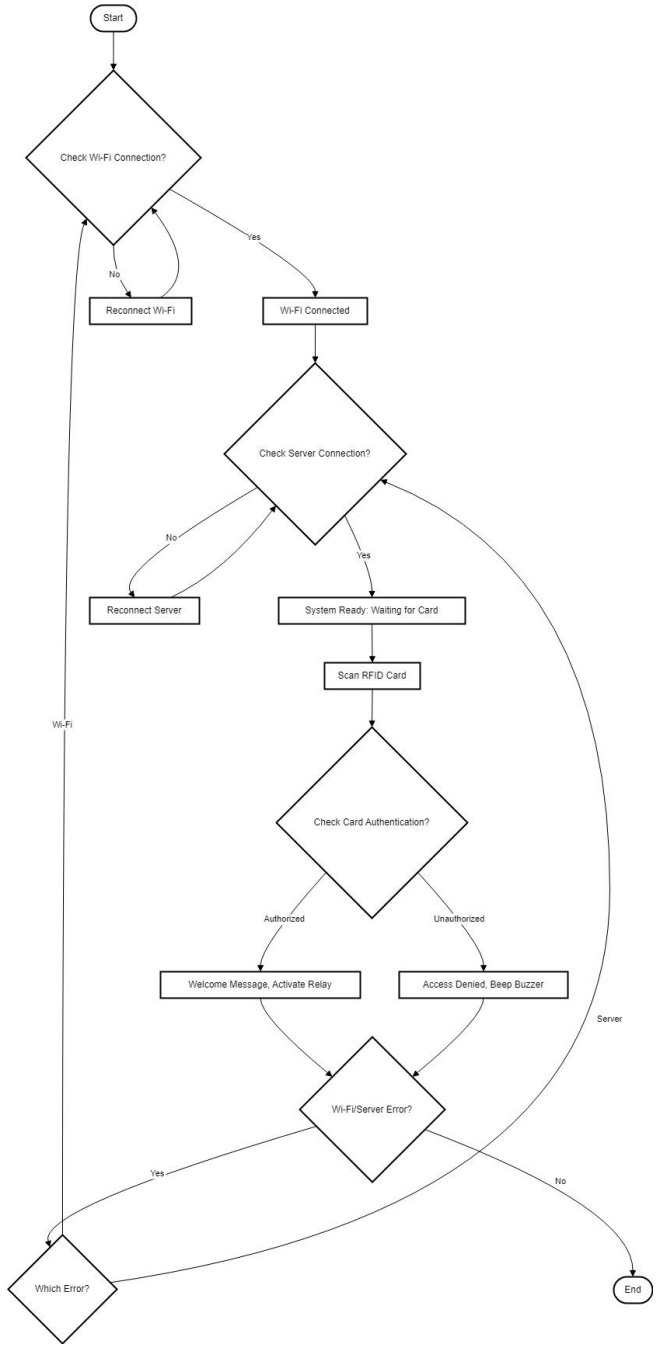




I2C OLED Display Pinout

Last Minute ENGINEERS.com

Workflow


Start
Check Wi-Fi Connection?
Yes
No
Reconnect Wi-Fi
Wi-Fi Connected
Check Server Connection?
No
Yes
Reconnect Server
System Ready: Waiting for Card
Scan RFID Card
Wi-Fi
Check Card Authentication?
Authorized
Unauthorized
Welcome Message, Activate Relay
Access Denied, Beep Buzzer
Server
Wi-Fi/Server Error?
Yes
No
Which Error?
End

Server setup

1. **Download MySQL**:
    - Go to MySQL Community Downloads and download the **MySQL Installer**.
2. **Install MySQL Server**:
    - Run the installer and choose the setup type (e.g., Developer Default).
    - Select **MySQL Server** and **MySQL Workbench** during the installation.
    - Configure the MySQL server with:
        - **Root Password**: Set a strong password for the root user.
        - **Port**: Ensure MySQL is set to run on port 3306.
3. **Test MySQL Installation**:
    - Open **MySQL Workbench** or **MySQL Command Line Client**.
    - Log in using the root credentials:

Code in MySQL Workbench

- CREATE DATABASE turnstile_auth;
- CREATE USER 'raspberrypi'@'%' IDENTIFIED BY 'your_password’;
- GRANT ALL PRIVILEGES ON turnstile_auth.* TO 'raspberrypi'@'%’;
- FLUSH PRIVILEGES;

- USE turnstile_auth;
- CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  uid VARCHAR(255) NOT NULL,
  name VARCHAR(100) NOT NULL
-);
- INSERT INTO users (uid, name) VALUES('508184438054', 'Jaydeep Gupta'),('374694568577', 'Dharmesh');

Edit my.ini Using Command Prompt

1.Open Command Prompt as Administrator:

- Press Win + R, type cmd, and press Ctrl + Shift + Enter to open Command Prompt with administrator privileges.

2.Navigate to the my.ini Directory:

- Use the cd (change directory) command to go to the directory containing the my.ini file. For example:

```
cmd
cd "C:\Program Files\MySQL\MySQL Server X.X\"
```

3.Open my.ini in Notepad:

- Type the following to open my.ini in Notepad for editing:

```
cmd
notepad my.ini
```

- This will launch Notepad with the my.ini file open.

4.Edit the bind-address Setting:

- Locate the [mysqld] section in the file.
- Add or update the line:

```
text
bind-address = 0.0.0.0
```

- This change allows MySQL to accept connections from any IP address.

5.Save and Close the File:

- Press Ctrl + S in Notepad to save the changes.
- Close Notepad.

Create Inbound Rule

•Open **Windows Defender Firewall**:
•Go to **Control Panel > System and Security > Windows Defender Firewall > Advanced Settings**.
•Check **Inbound Rules**:
•In the left-hand pane, click **Inbound Rules**.
•In the right-hand pane, look for rules that include **Port** under the **Protocol and Ports** column.
•Filter by Port:
•Scroll through the list or use the **Filter** option to find rules that involve port 3306.
•Review Rule Details:
•Double-click a rule to open its properties and view:
    - **Name**: Name of the rule.
    - **Enabled**: Whether the rule is active.
    - **Protocol and Ports**: Specific port or protocol the rule applies to.

Setting a Static IP Address

Steps:

1.Open Network Settings:

- Go to **Control Panel > Network and Sharing Center > Change Adapter Settings**.

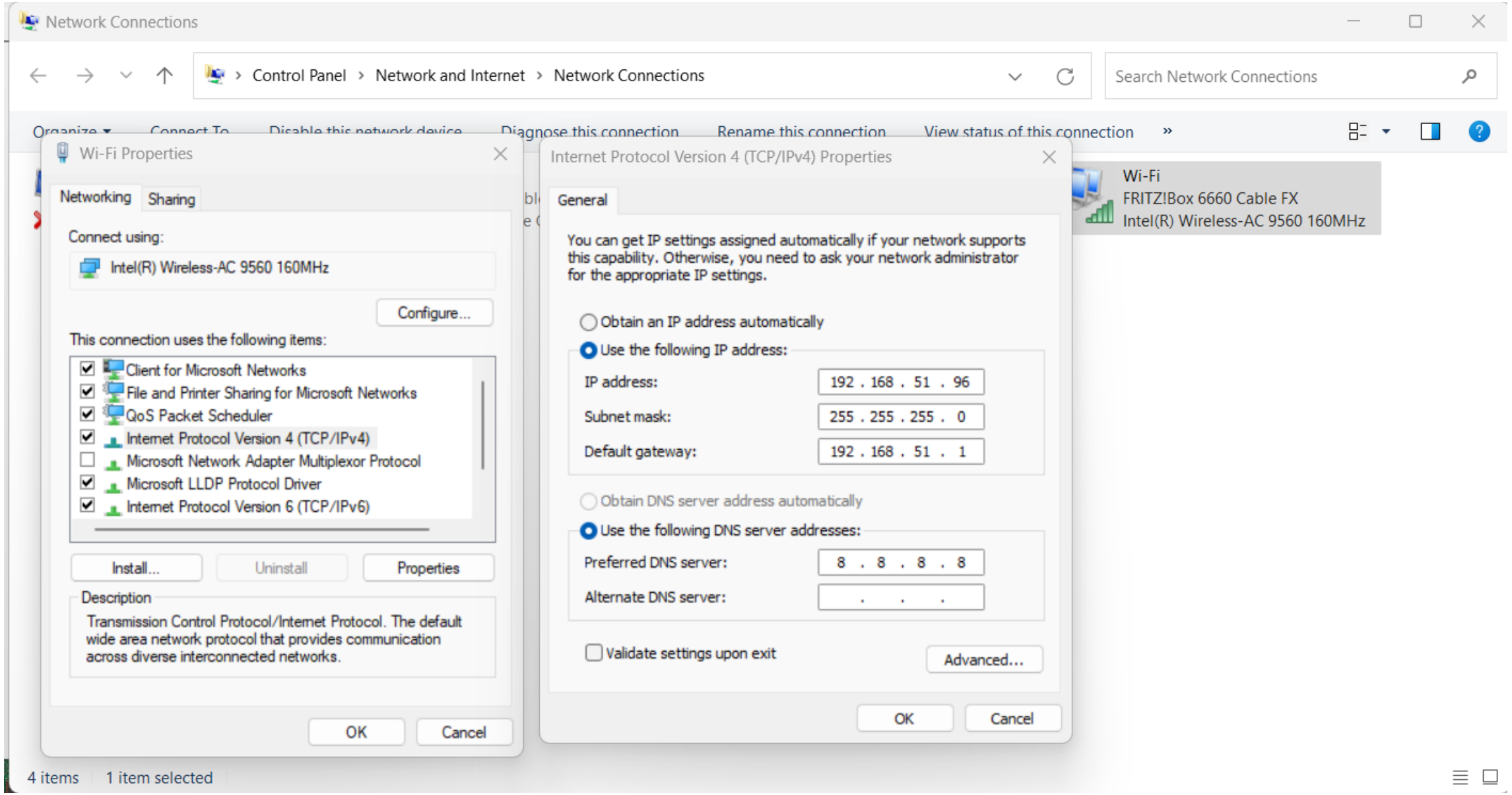
2.Select the Network Adapter:

- Right-click on your active network connection (e.g., Wi-Fi) and choose **Properties**.

3.Configure IPv4 Settings:

- Select **Internet Protocol Version 4 (TCP/IPv4)** and click **Properties**.
- Choose **Use the following IP address** and enter:
  - **IP Address**: An address in the same range as the router (e.g., 192.168.51.96).
  - **Subnet Mask**: Typically 255.255.255.0.
  - **Default Gateway**: The router's IP (e.g., 192.168.51.1).
- Enter a **Preferred DNS Server** (e.g., 8.8.8.8 for Google DNS).

*NOTE:- using static IP address will restrict certain websites to browse in your server computer


Network Connections
Control Panel › Network and Internet › Network Connections
Search Network Connections
Organize
Connect To
Disable this network device
Diagnose this connection
Rename this connection
View status of this connection
Wi-Fi
FRITZ!Box 6660 Cable FX
Intel(R) Wireless-AC 9560 160MHz
Wi-Fi Properties
Networking
Sharing
Connect using:
Intel(R) Wireless-AC 9560 160MHz
Configure...
This connection uses the following items:
Client for Microsoft Networks
File and Printer Sharing for Microsoft Networks
QoS Packet Scheduler
Internet Protocol Version 4 (TCP/IPv4)
Microsoft Network Adapter Multiplexor Protocol
Microsoft LLDP Protocol Driver
Internet Protocol Version 6 (TCP/IPv6)
Install...
Uninstall
Properties
Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
OK
Cancel
Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 192 . 168 . 51 . 96
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 192 . 168 . 51 . 1
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server: 8 . 8 . 8 . 8
Alternate DNS server:
Validate settings upon exit
Advanced...
OK
Cancel
4 items
1 item selected

Prepare Your Raspberry Pi

Step 1.1: Install the OS

1.Download Raspberry Pi OS:

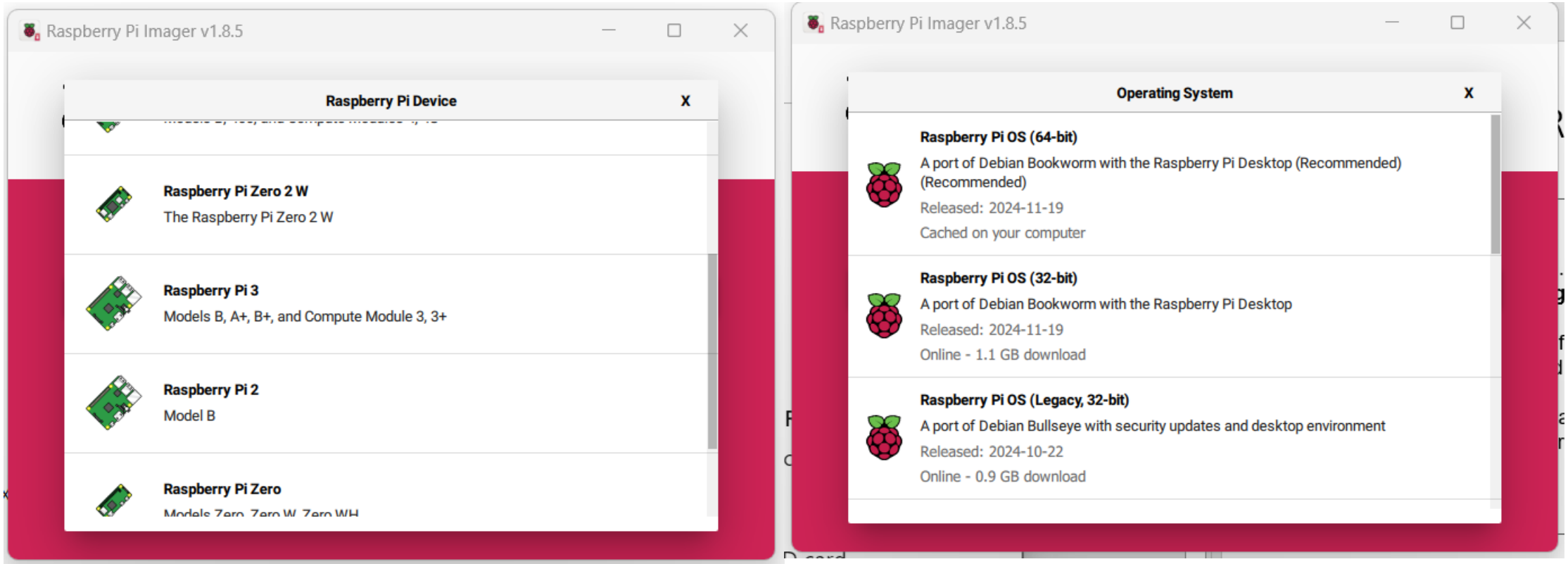
- Visit Raspberry Pi Downloads.
- Download **Raspberry Pi Imager** and use it to install **Raspberry Pi OS with Desktop** on your SD card

2.Write the OS to the SD Card:

- Use **Raspberry Pi Imager** to flash the OS onto the SD card.
- Optionally configure Wi-Fi and SSH during the setup (advanced options).

3.Boot the Raspberry Pi:

- Insert the SD card into your Raspberry Pi and power it on.
- If using a monitor and keyboard, log in with:
  - Username: pi
  - Password: raspberry



Select Raspberry Pi 3→Legacy, 32-bit

Step 1.2: Update the System

Run the following commands to update the Raspberry Pi's packages:

bash

sudo apt update sudo apt upgrade -y

Step 1.3: Enable Necessary Interfaces

1.Open the Raspberry Pi configuration tool:

bash

sudo raspi-config

2.Enable:

- **I2C** (for the OLED display).
- **SPI** (for the RFID reader).
- **SSH** (for remote access).

3.Save changes and reboot the Raspberry Pi:

bash

sudo reboot

3. Install Required Software

Step 3.1: Install Python and Libraries

1.Install required Python libraries:

bash

pip install mfrc522

pip install adafruit-circuitpython-ssd1306

pip install pillow

pip install mysql-connector-python

Step 3.2: Test I2C and SPI

1.Check I2C Devices:

bash

sudo i2cdetect -y 1

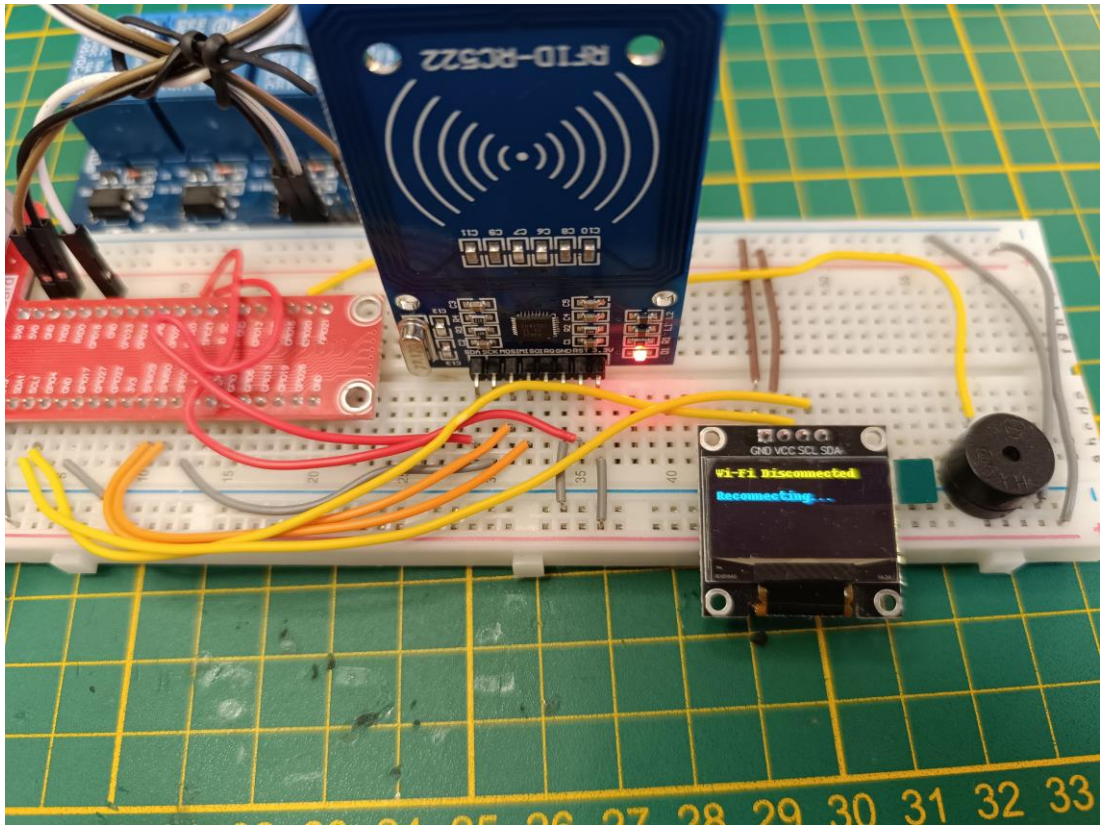
- Verify that the OLED display's address appears (e.g., 0x3C).

2.Check SPI Devices:

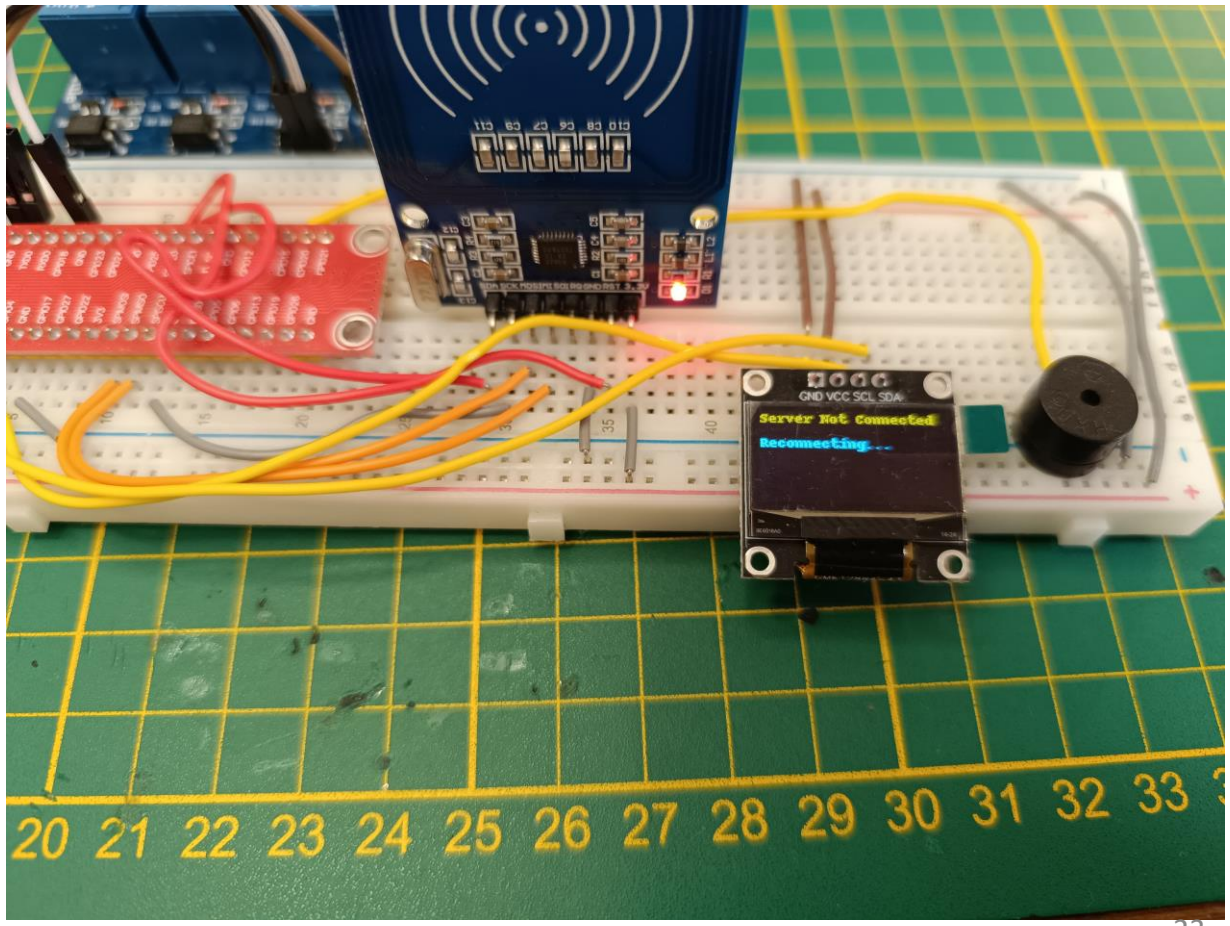
- The RC522 should work without additional SPI checks if wired correctly.

Test the Project

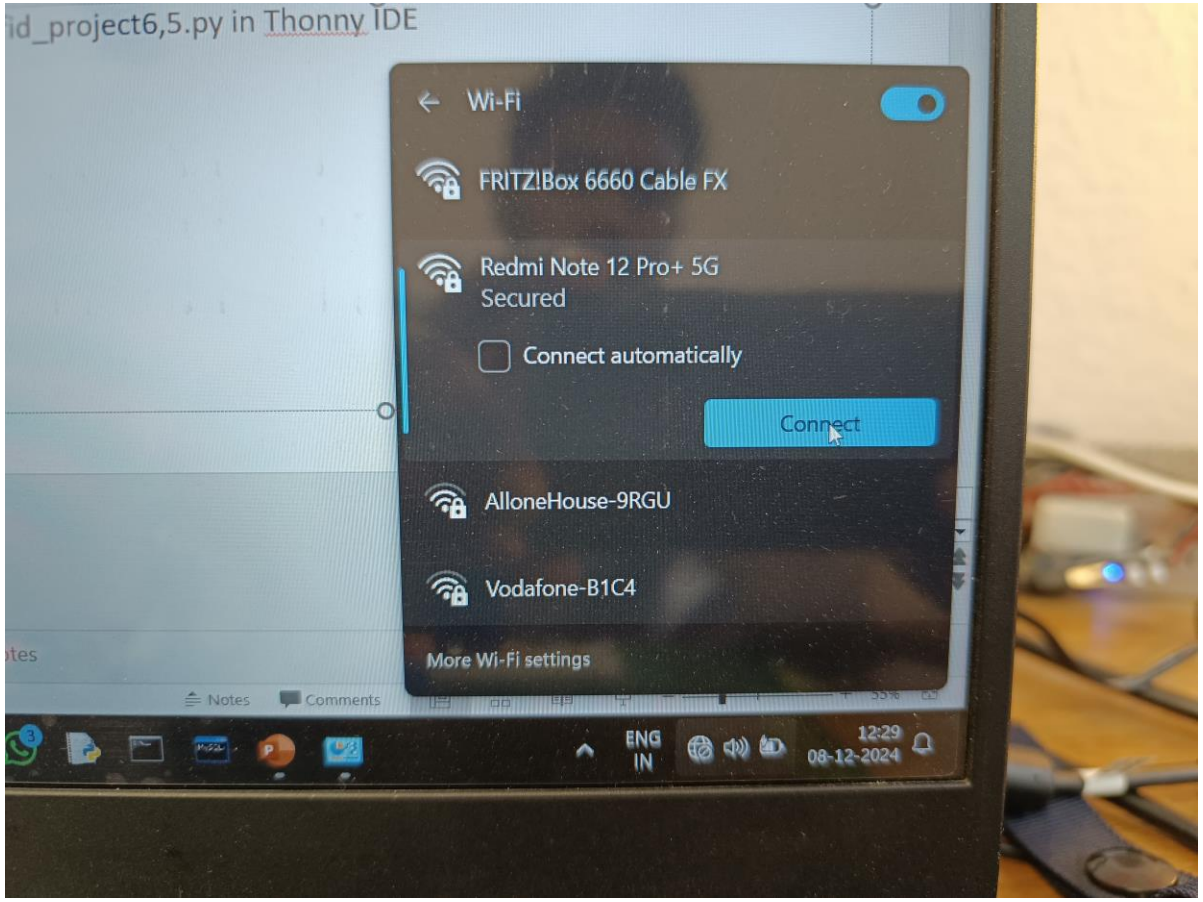
- Run Rfid_project6,5.py in Thonny IDE on Raspberry Pi
- First look, When wifi is not connected



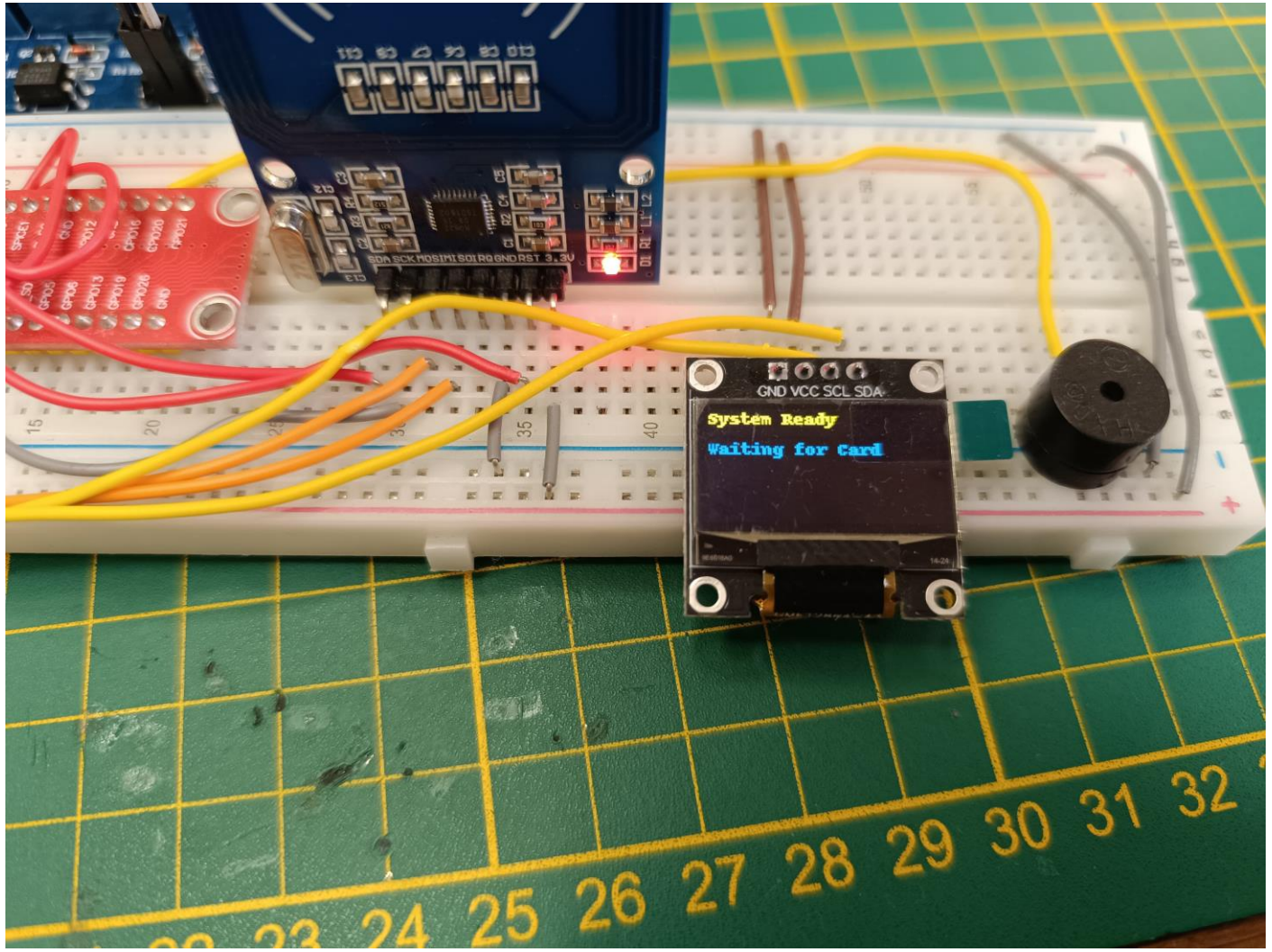
After connecting to wifi, It will display “server not connected”



- Connect your server where you have installed and made database in MySQL. In my case in my laptop.
- Connect server with the same wifi as with Raspberry Pi is connected so that both devices would be on same subnet.

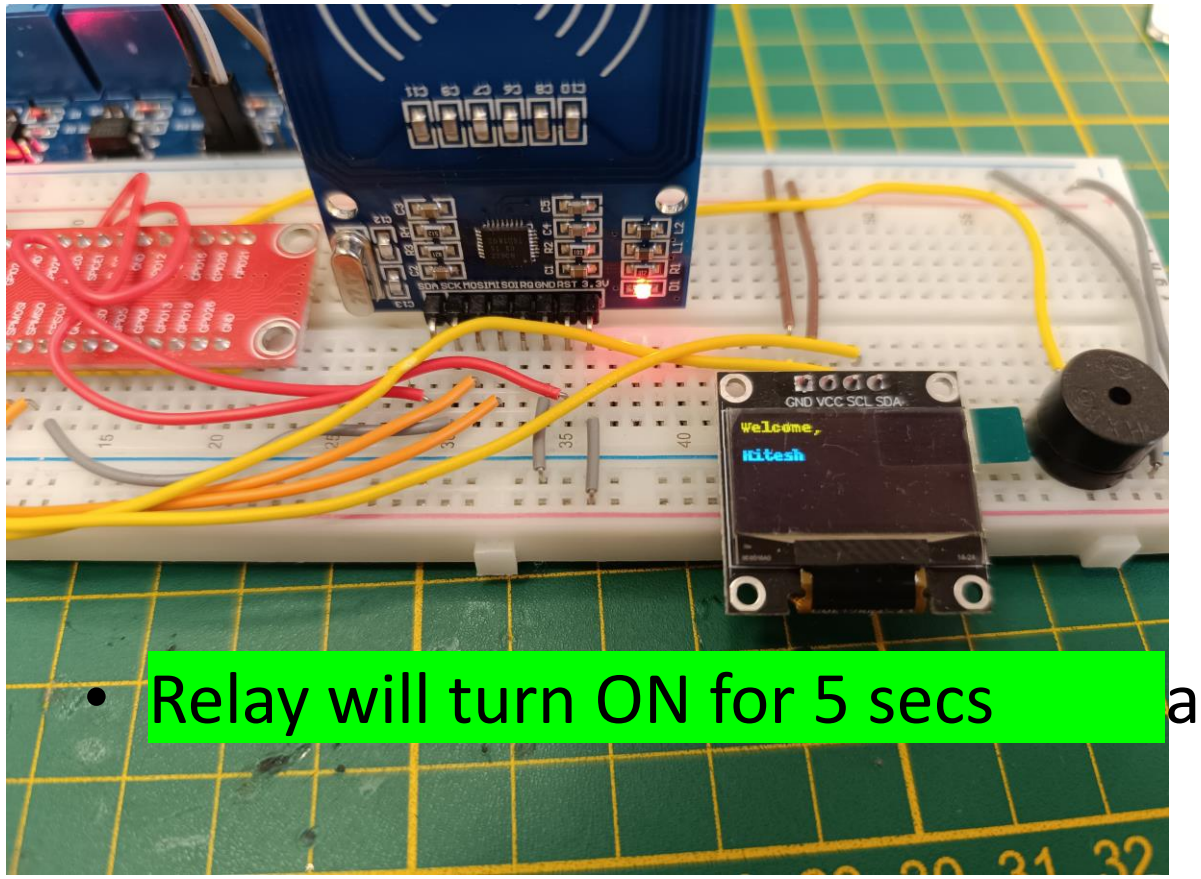


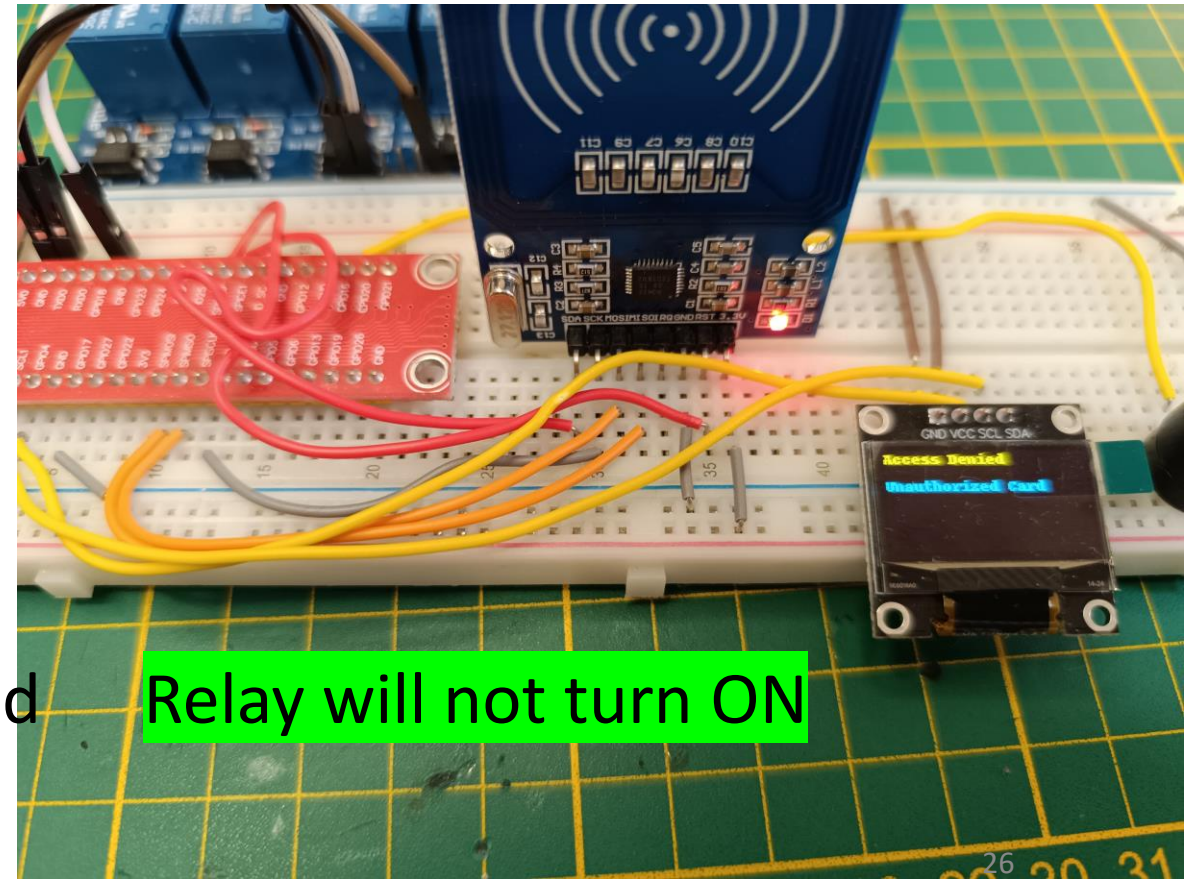
- Now the display will show “System Ready Waiting for Card”



Place card on reader

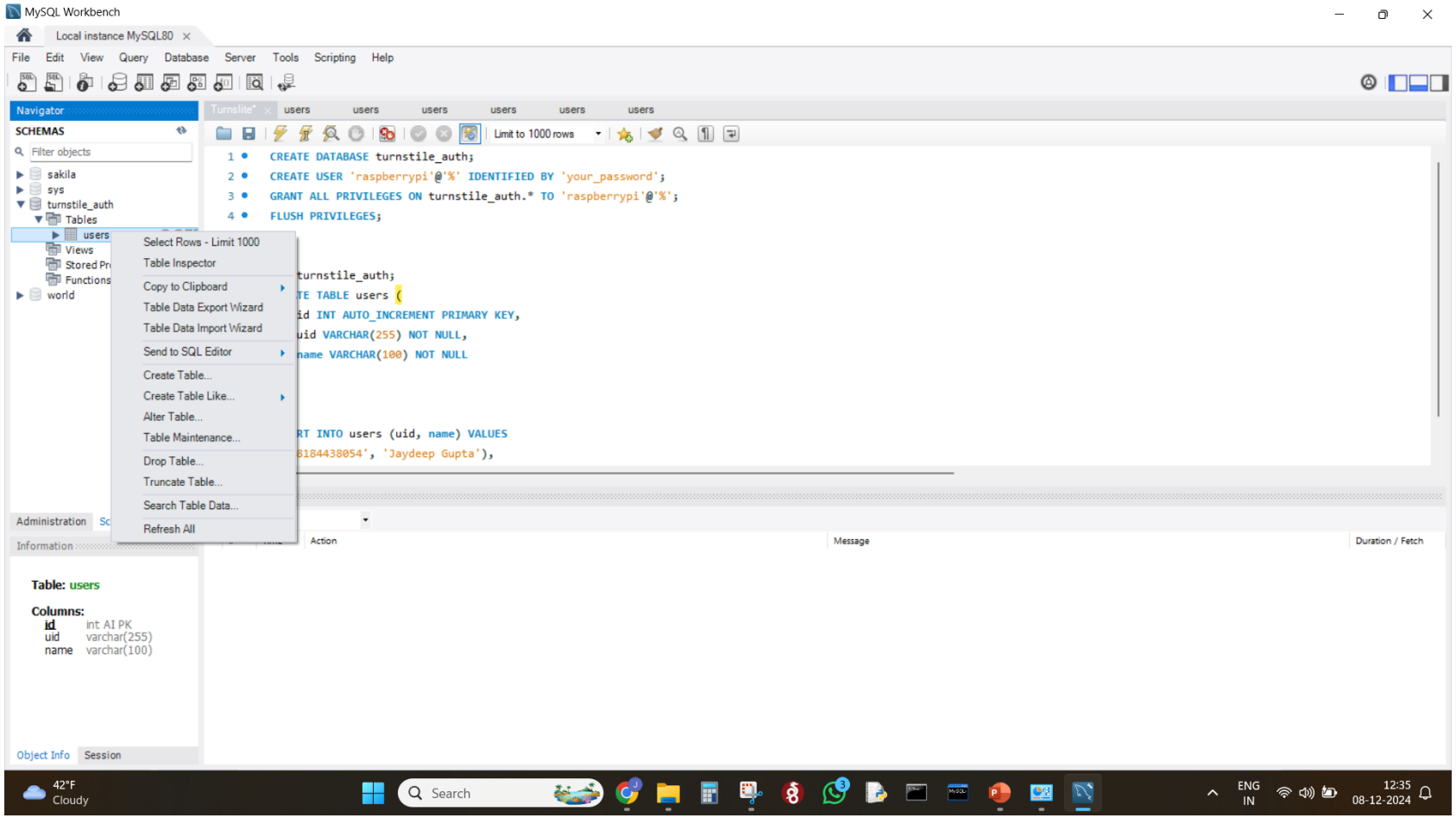
- Saved in database and Not saved in database

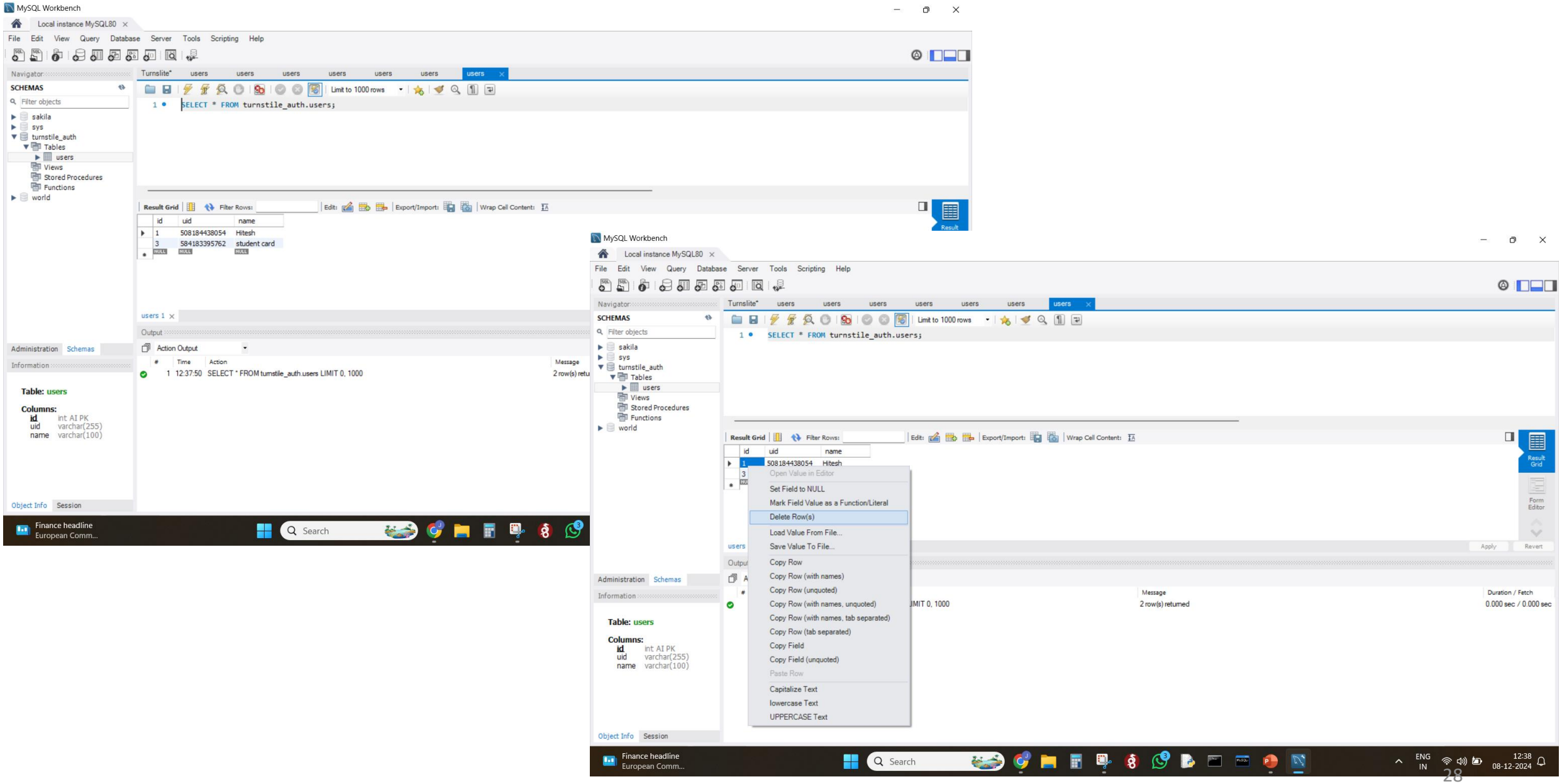




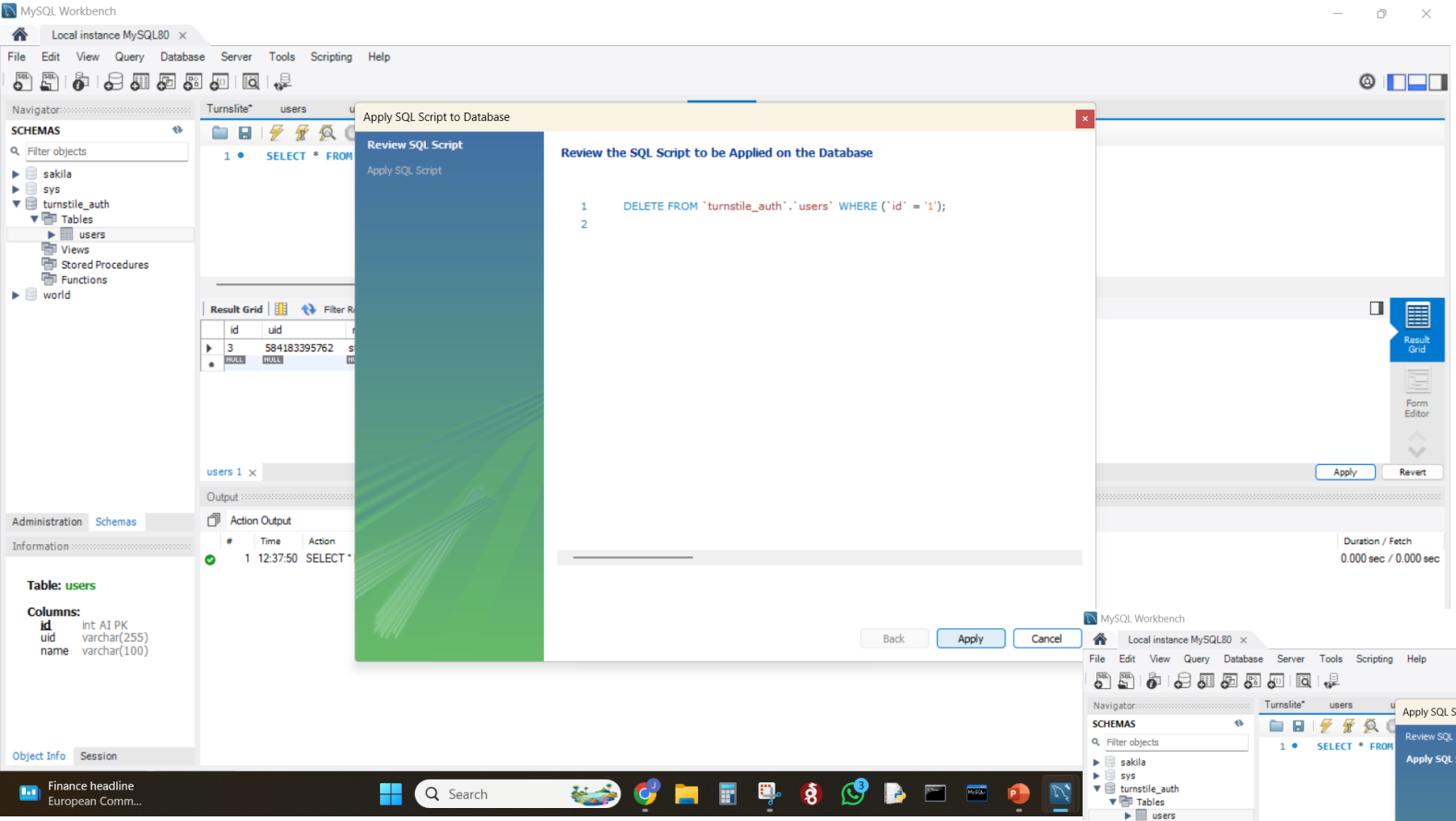
- Relay will turn ON for 5 secs and Relay will not turn ON

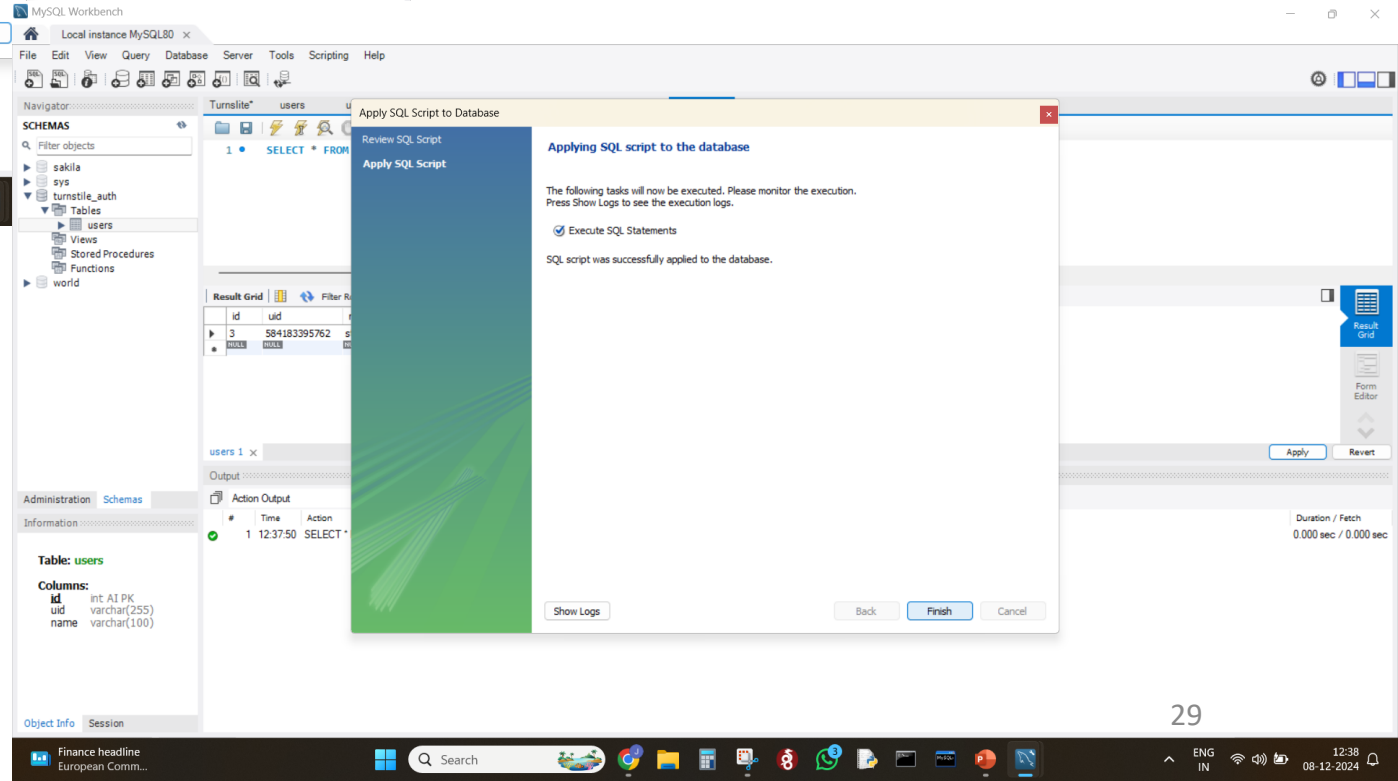
Removing/ Adding UID in database

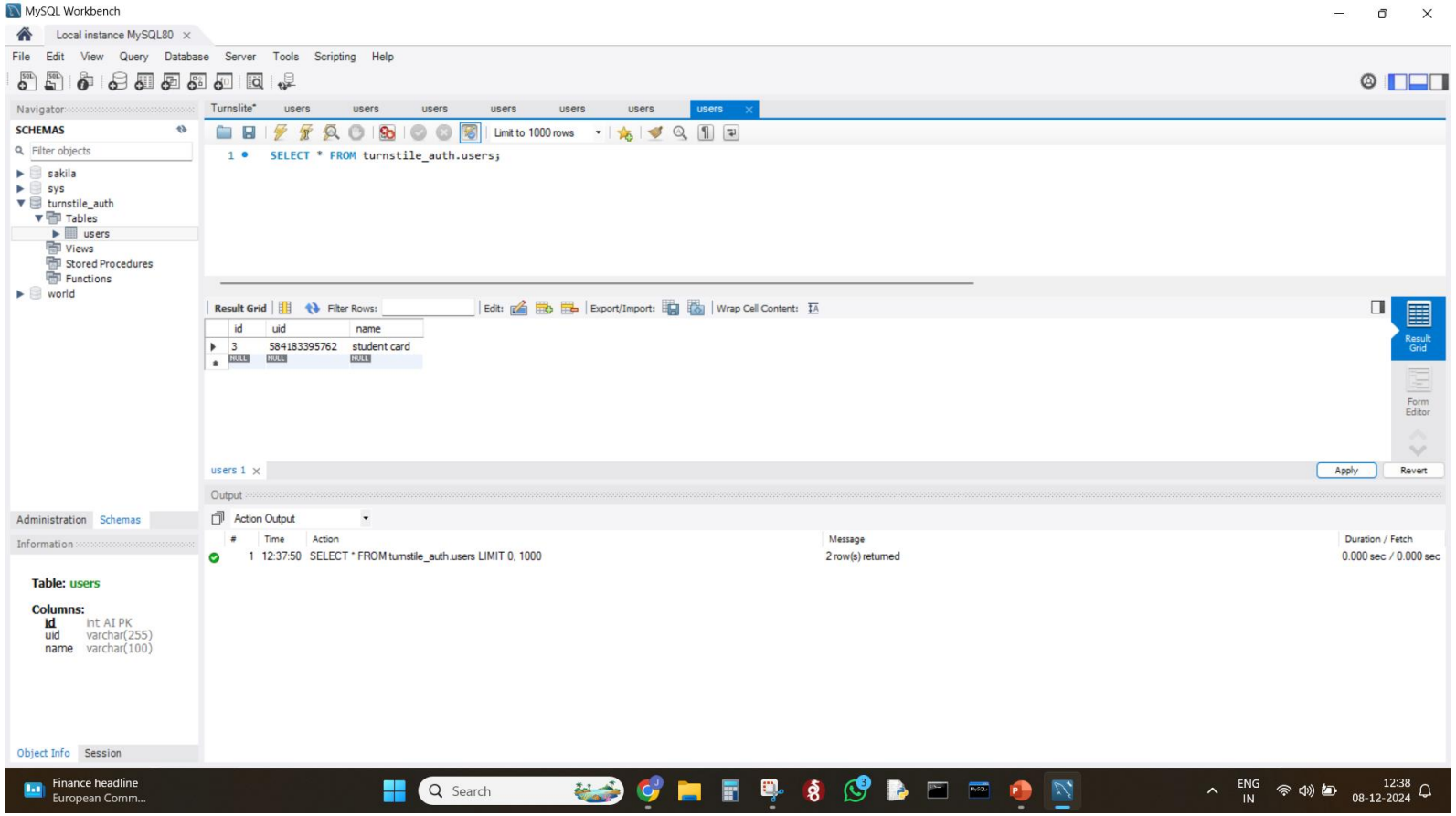




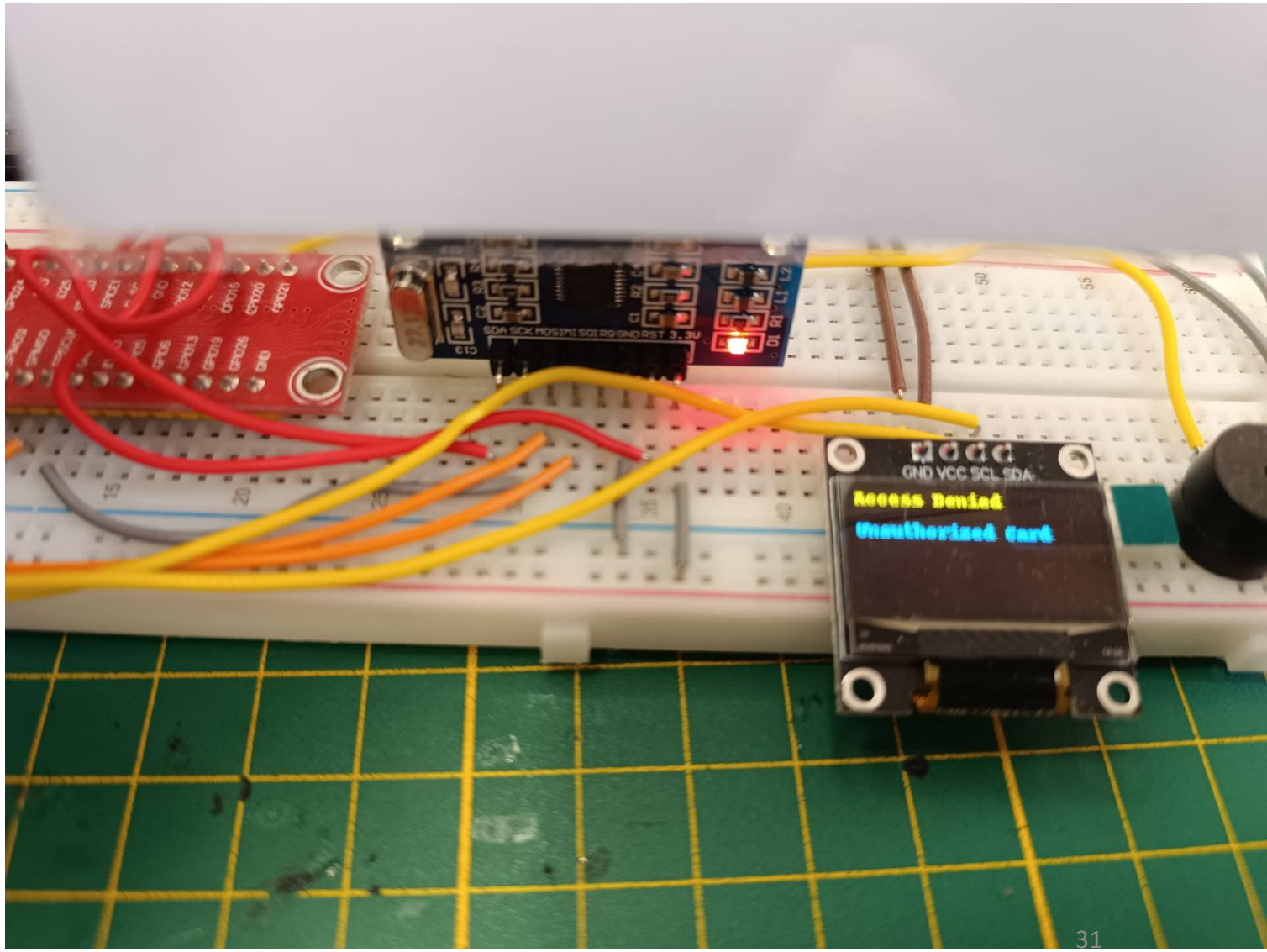
Removing UID from Database


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sakila
sys
turnstile_auth
Tables
users
Views
Stored Procedures
Functions
world
Administration Schemas
Information
Table: users
Columns:
id int AI PK
uid varchar(255)
name varchar(100)
Object Info Session
Turnslite* users
1 SELECT * FROM
Result Grid
id uid
3 584183395762
users 1
Output
Action Output
Time Action
1 12:37:50 SELECT *
Result Grid
Form Editor
Apply Revert
Duration / Fetch
0.000 sec / 0.000 sec
Apply SQL Script to Database
Review SQL Script
Apply SQL Script
Review the SQL Script to be Applied on the Database
1 DELETE FROM `turnstile_auth`.`users` WHERE (`id` = '1');
2
Back Apply Cancel
Finance headline
European Comm...
Search


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sakila
sys
turnstile_auth
Tables
users
Views
Stored Procedures
Functions
world
Administration Schemas
Information
Table: users
Columns:
id int AI PK
uid varchar(255)
name varchar(100)
Object Info Session
Turnslite* users
1 SELECT * FROM
Result Grid
id uid
3 584183395762
users 1
Output
Action Output
Time Action
1 12:37:50 SELECT *
Result Grid
Form Editor
Apply Revert
Duration / Fetch
0.000 sec / 0.000 sec
Apply SQL Script to Database
Review SQL Script
Apply SQL Script
Applying SQL script to the database
The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.
Execute SQL Statements
SQL script was successfully applied to the database.
Show Logs
Back Finish Cancel
Finance headline
European Comm...
Search
ENG
IN
12:38
08-12-2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sakila
sys
turnstile_auth
Tables
users
Views
Stored Procedures
Functions
world
Turnstile*
users
Limit to 1000 rows
1 • SELECT * FROM turnstile_auth.users;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
id uid name
3 584183395762 student card
NULL NULL NULL
Result Grid
Form Editor
users 1
Apply
Revert
Output
Action Output
Time Action Message Duration / Fetch
1 12:37:50 SELECT * FROM turnstile_auth.users LIMIT 0, 1000 2 row(s) returned 0.000 sec / 0.000 sec
Administration Schemas
Information
Table: users
Columns:
id int AI PK
uid varchar(255)
name varchar(100)
Object Info Session
Finance headline
European Comm...
Search
ENG
IN
12:38
08-12-2024

- Now try that card which we have remove from database. It will show “Access Denied Unauthorised Card”



6. Set Up Auto-Run on every Reboot

To automatically run the project on boot:

1.Edit the crontab:

bash

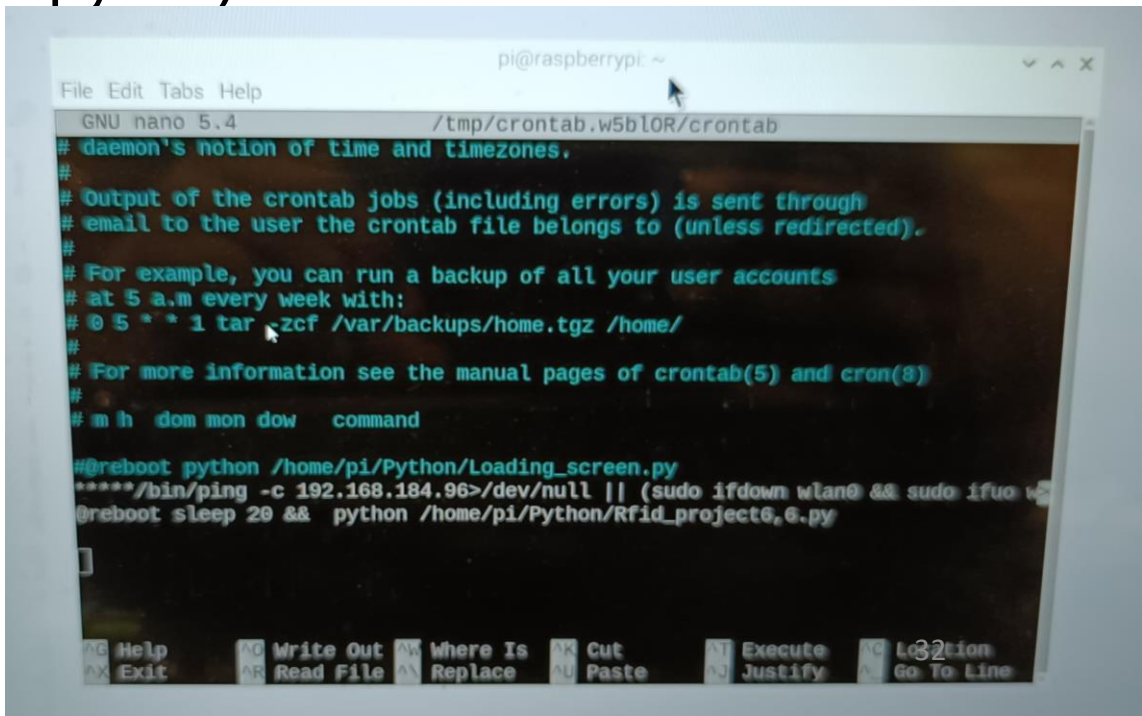
crontab -e

2.Add the following line at the end:

bash

@reboot sleep 20 && python /path/to/rfid_project6,5.py

(replace */path/to* with your actual location of .py file)



2. What Worked Well?

- **Hardware Integration**:
  - The RFID reader, relay module, and OLED display worked seamlessly with the Raspberry Pi, showcasing reliable performance in authentication tasks.
- **Database Management**:
  - MySQL effectively handled the storage and retrieval of user data, enabling accurate validation of RFID cards.
- **System Functionality**:
  - The relay control provided a robust mechanism for actuating external devices (like a door lock) based on authentication.
- **Connectivity**:
  - The Python script's integration with the MySQL server demonstrated stable real-time communication.
- **Ease of Debugging**:
  - Testing individual components (e.g., relay, RFID reader) made troubleshooting manageable.

3. What Could Have Been Done Better?

- **Security Enhancements**: Encrypting the connection between the Raspberry Pi and MySQL server (e.g., SSL/TLS) to improve security.
- Making a web interface for adding and removing users from database
- Using another Raspberry Pi to make separate device which differentiates owned cards with there name and new card displays UID on screen to be save in database.

4. Was My Time Estimation/Planning Realistic?

- **Partially Realistic**: Initial time estimates for hardware assembly and basic functionality were accurate.
- However, unforeseen challenges, such as resolving connectivity issues with the server and configuring the firewall, required more time than anticipated.
- Debugging and integrating all components into a unified system took longer than expected, especially handling MySQL remote access and configuring the relay module.

5. What Else Did I Learn Through the Project?

- **Technical Knowledge**:
- Gained a deeper understanding of:
  - GPIO programming on the Raspberry Pi.
  - Configuring MySQL for remote access.
  - Managing network configurations (e.g., static IP setup, firewall rules).
- **Problem-Solving**:
- Learned how to debug complex systems by isolating hardware and software components during testing.
- **Time Management**:
- Understood the importance of allocating extra time for unexpected issues, especially in multi-component projects.
- **Collaboration**:
- Realized the value of documenting each step for better communication during presentations or collaborations.
- **Practical Application**:
- Learned how theoretical concepts (e.g., database design, network communication) translate into practical, real-world systems.

- For a complete explanation of this project watch this video:- https://youtu.be/Sv5qsHx23_U
- Get Python code for this project:-

Thanks to all for reading till the end