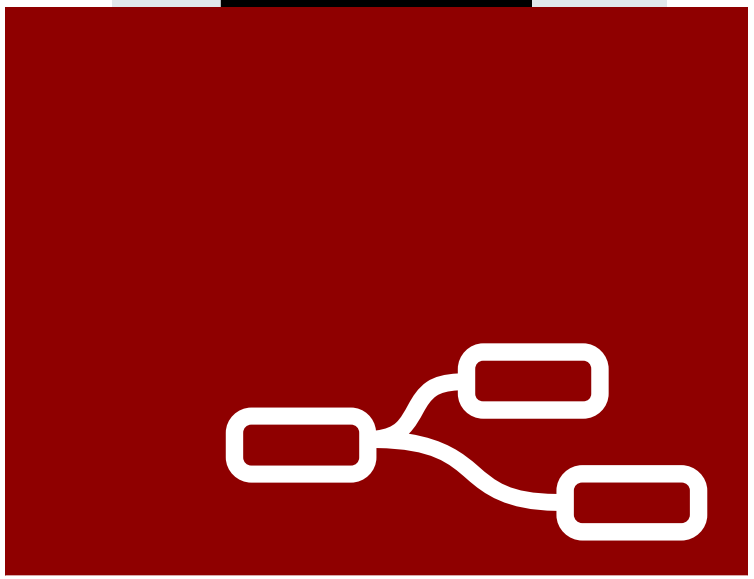
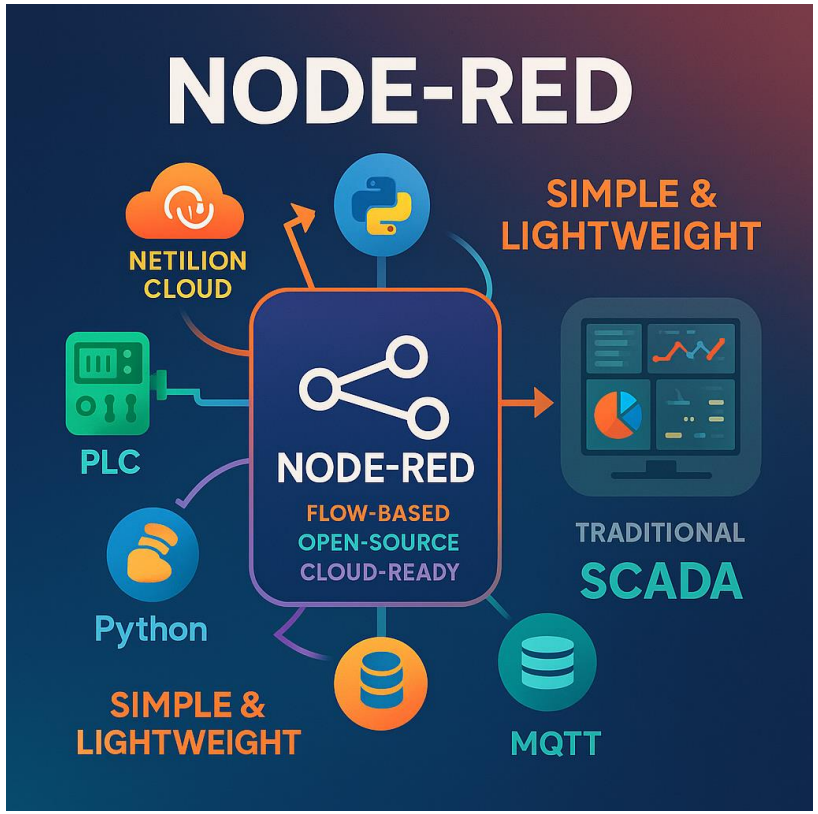
Node-RED





- visual programming tool for wiring together hardware devices, APIs and online services.
- Used in our project to control, monitor and visualize the Smart Parking System.
- Plays a key role in connecting PLC, Python, Database and MQTT.

Why Node-RED is Better Than WinCC (SCADA) for This Project

Free & Open Source

Node-RED is completely free, while WinCC requires expensive licenses.

User-Friendly Interface

Node-RED offers a simple, drag-and-drop flow editor. WinCC is more complex and needs special training.

Web & Mobile Accessibility

Node-RED dashboards work in any browser, even on mobile. WinCC requires additional setup for remote acce.

Easy Integration

Node-RED works seamlessly with Python, MQTT, databases, and cloud platforms. WinCC has limitited third-party support.

Flexible for IoT

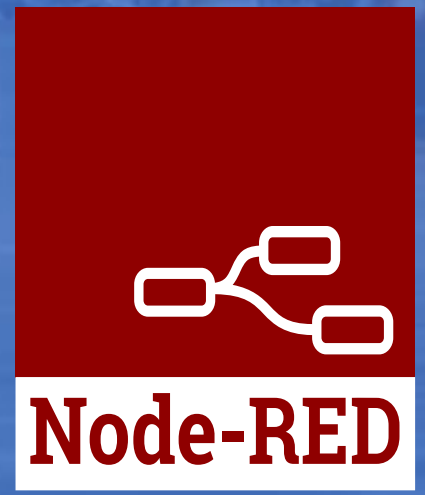
Node-RED is designed with IoT in mind—ideal for smart, connected systems. WinCC is mainly built for traditional industrial environments

Lightweight Hosting

Node-RED can run on small devices like Raspberry Pi. WinCC usually needs an industrial-grade PC

Node-RED Accessibility

- Node-RED can be accessed from any device on the local network ,whether it's a PC, tablet, or phone ;through a simple web browser.
- Once hosted (e.g. on a Raspberry Pi or local server) , it needs no extra software and runs smoothly even on low-power devices.
- This makes monitoring and control easy and flexible from anywhere in the system.



Node-RED Architecture in Our System

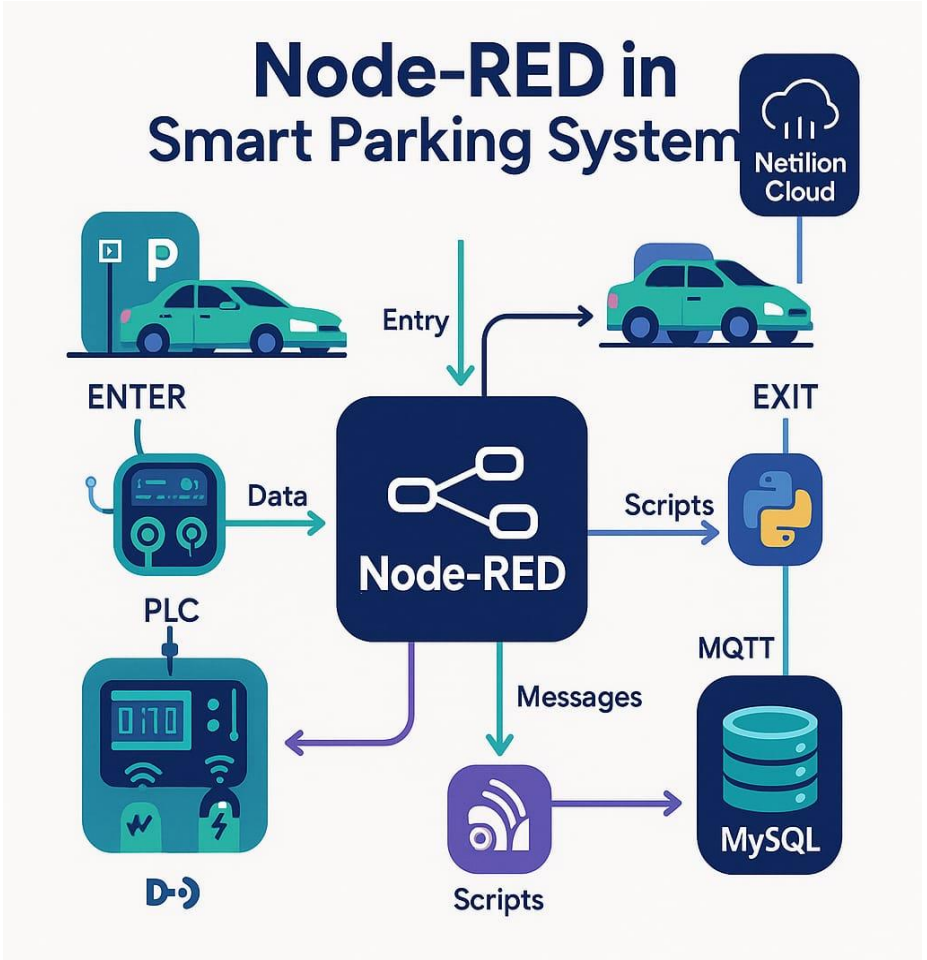
Node-RED → the central hub that connects

PLC (Modbus)
for sensors and control

Python Scripts
for user validation and car identification

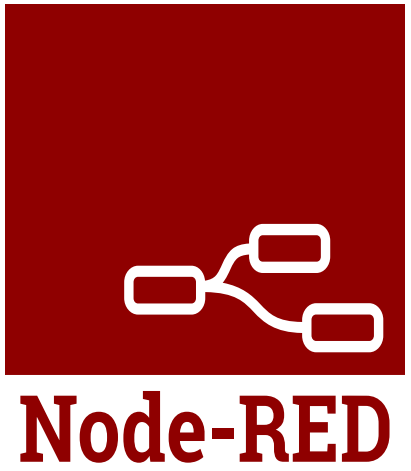
MySQL Database
for storing user data and car info

MQTT Broker
for lightweight communication

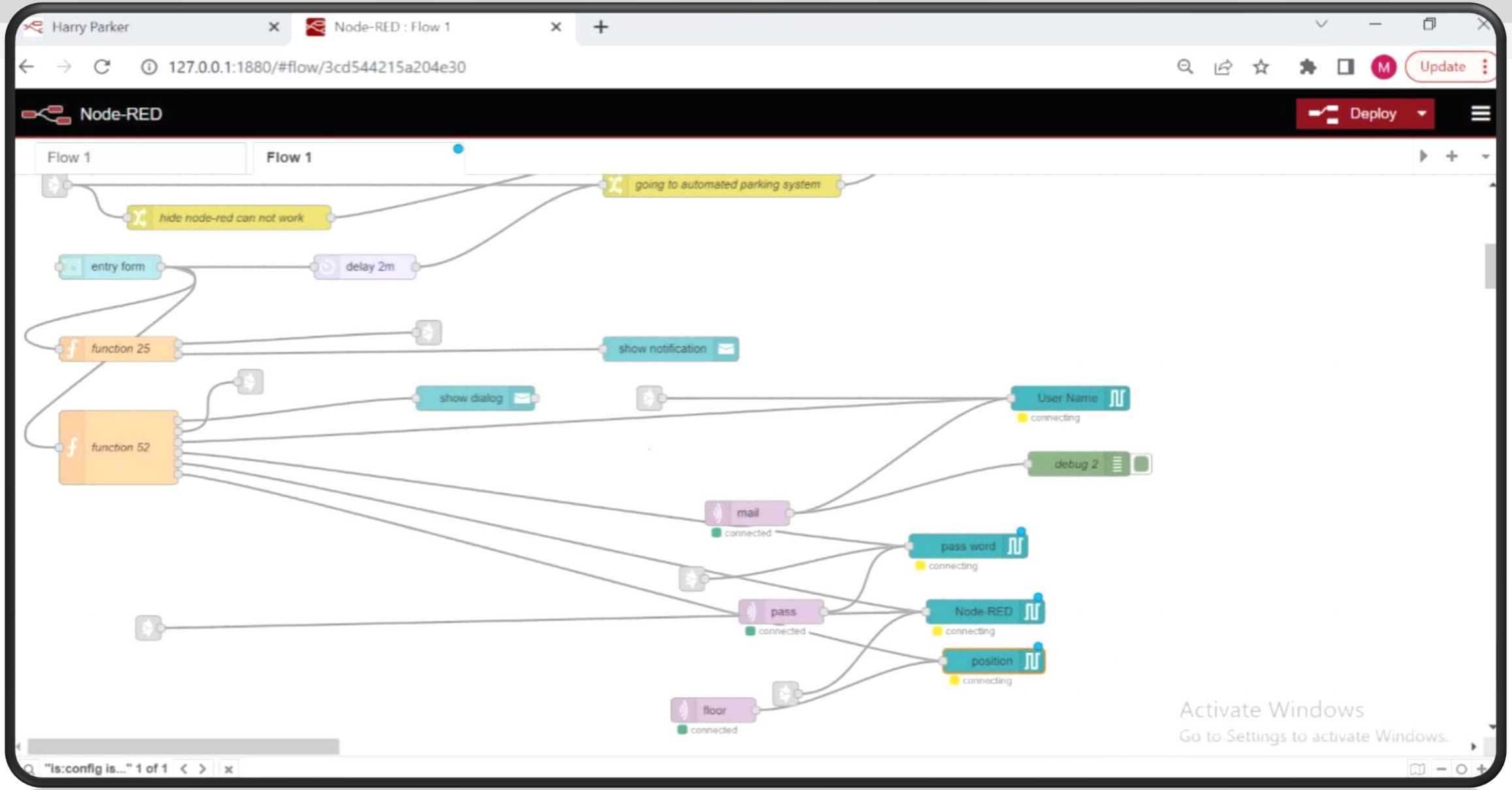


Node Types Used

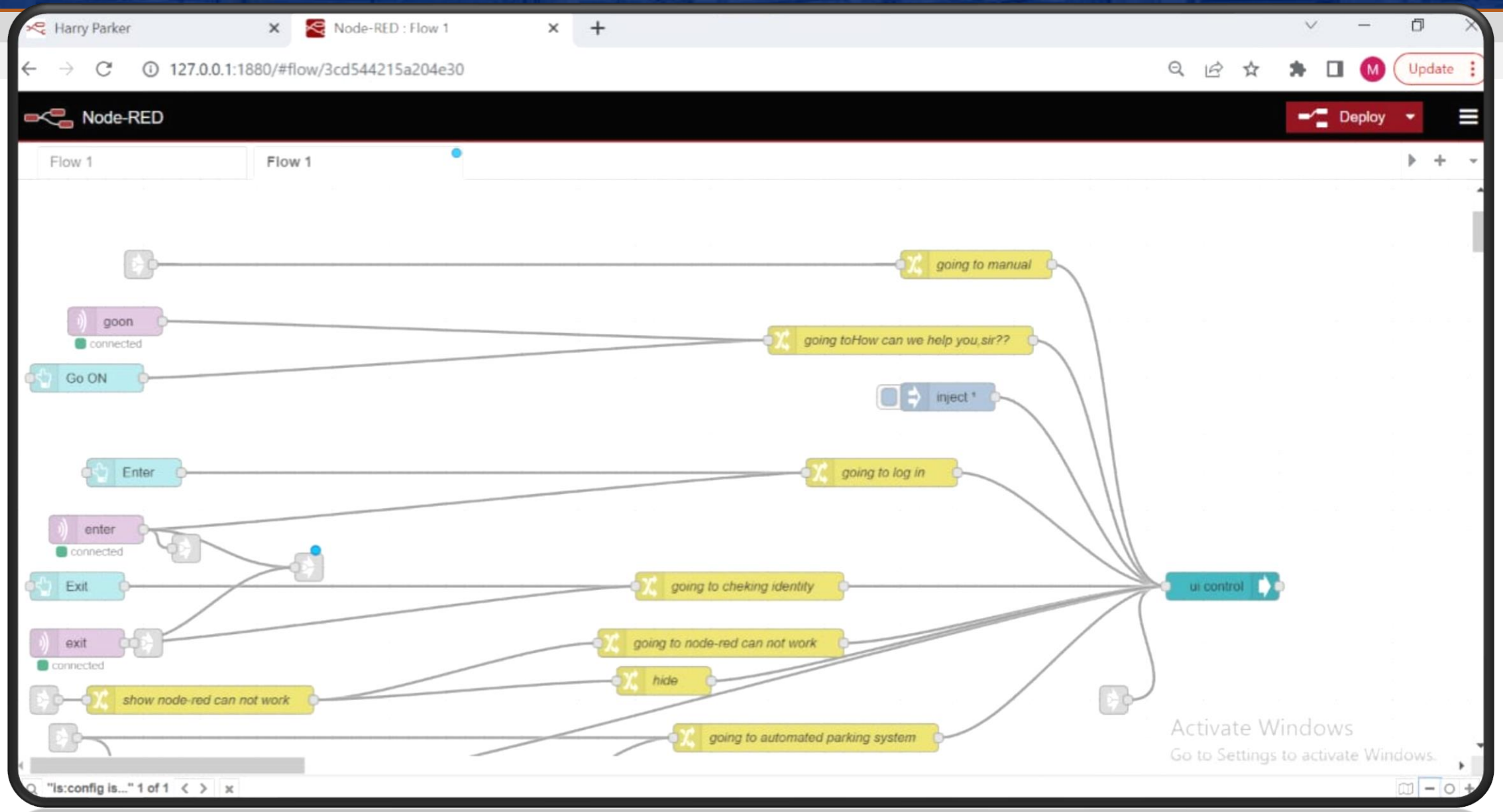
1. S7 in-read, S7 out-write → Connect with PLC mqtt
2. mqtt out → For internal messaging mysql → Interact with user database
3. ui_input, ui_text, ui_button → Build user interface (login, car status)
4. function, switch, debug → Handle flow logic



Node Types Used

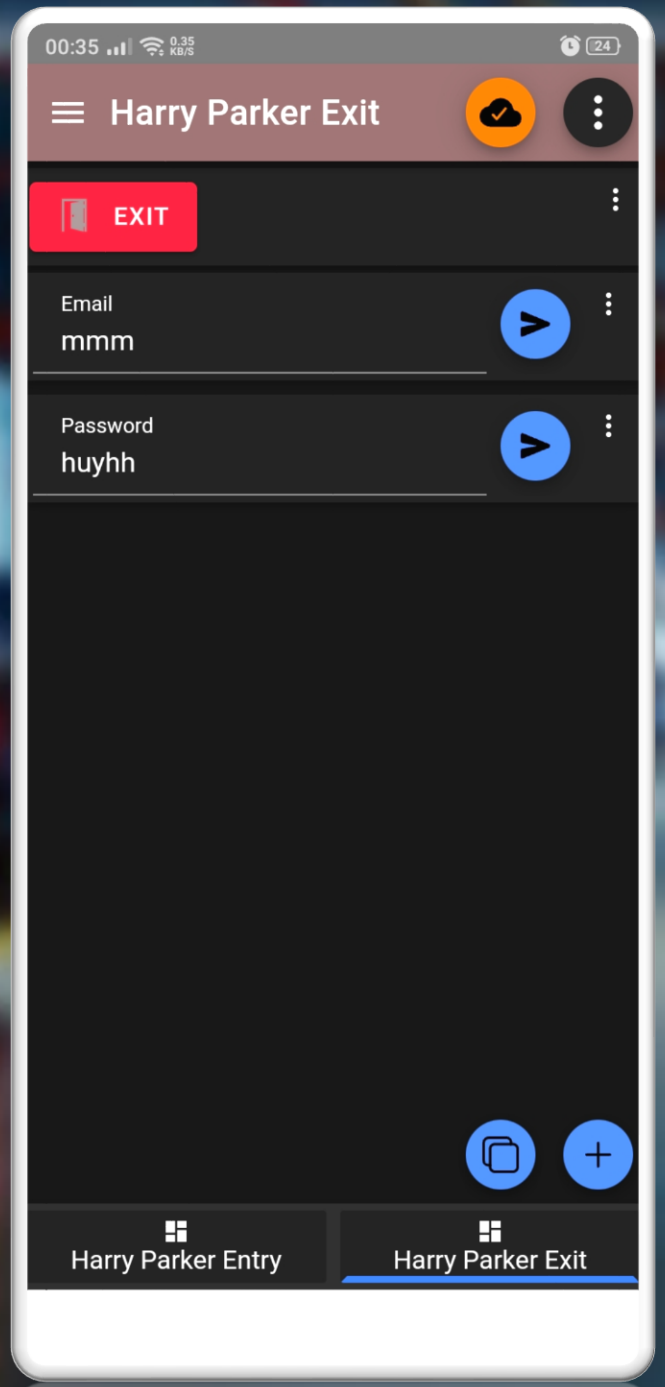


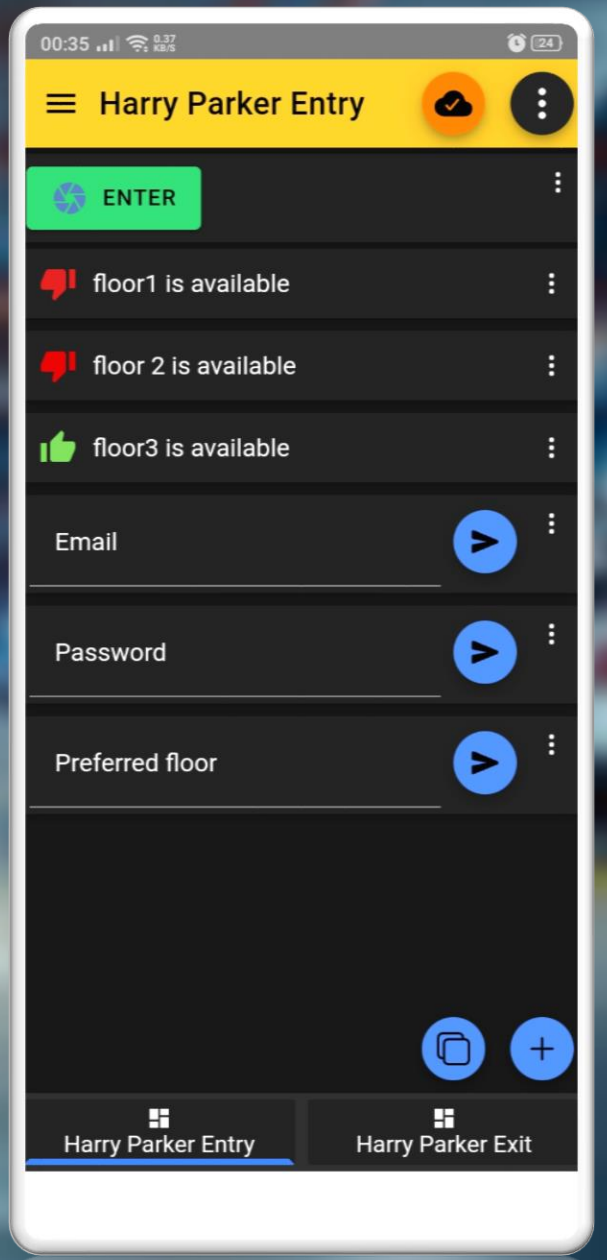
Node Types Used



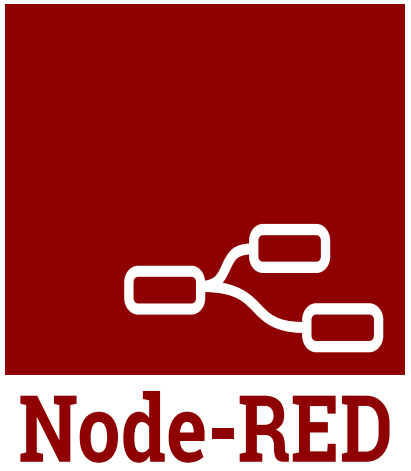
Node-RED, MQTT, and IoT

- Node-RED serves as the central hub in our IoT system , using MQTT for fast and lightweight communication between devices like PLCs, sensors and dashboards.
- This setup allows real-time data exchange, smart automation, and easy system scaling
- Making it ideal for modern industrial applications.


00:35
Harry Parker Exit
EXIT
Email
mmm
Password
huyhh
Harry Parker Entry
Harry Parker Exit


00:35
Harry Parker Entry
ENTER
floor1 is available
floor 2 is available
floor3 is available
Email
Password
Preferred floor
Harry Parker Entry
Harry Parker Exit

Example Dashboard Explanation



1.User enters username and password on dashboard

2. Node-RED sends this data to PLC → Python reads it

3. Python script checks the database

4.If match found → Node-RED identifies the car and sends movement command to the PLC

5. Gate opens automatically

Node-RED : Flow 1 | Harry Parker

localhost:1880/ui/#!/5?socketid=SwyJuBOWBA3EKSGIAAAB

Update

vertical manual control

rel execute

rel speed

rel distance

home execute

horizontal manual control

rel execute

rel speed

rel distance

home execute

manual

start and reset

restart

reset

Advantages of Node-RED in Industry 4.0

- Cloud-friendly and IoT-ready
- Can be monitored and controlled remotely
- Integrates well with AI and other modern tool
- Ideal for smart systems like parking, factories and building automation

Node-RED

Challenges and Solutions

PLC communication latency

- optimized with proper Modbus config

Dashboard delay

- fixed by managing UI refresh cycles

Conclusion & Impact

- Node-RED made the system flexible, visual, and easy to manage
- Allowed us to quickly develop, test, and deploy features.
- Central piece in integrating all components into a smart, realtime system.

Node-RED