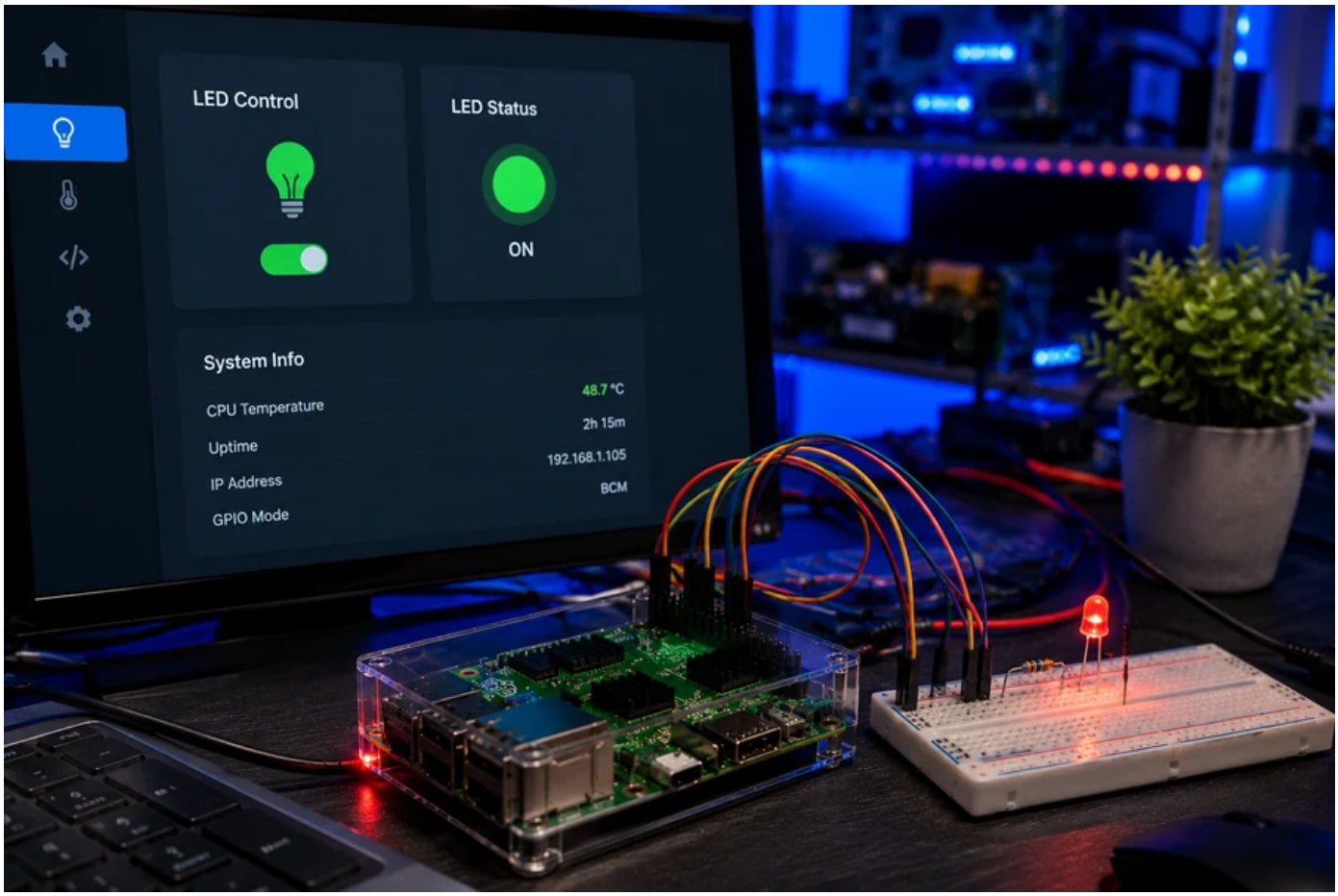
WolfieWeb Raspberry Pi Flask Web Dashboard Lab Manual

Super detailed GPIO LED wiring, Flask install, dashboard code, network testing, and upgrade instructions.



What You Will Build

This project turns the Raspberry Pi into a small local web server. A browser opens a Flask dashboard and controls a GPIO-connected LED. This is the foundation for browser-based robotics controls, sensor dashboards, and local automation panels.

Important GPIO Rule

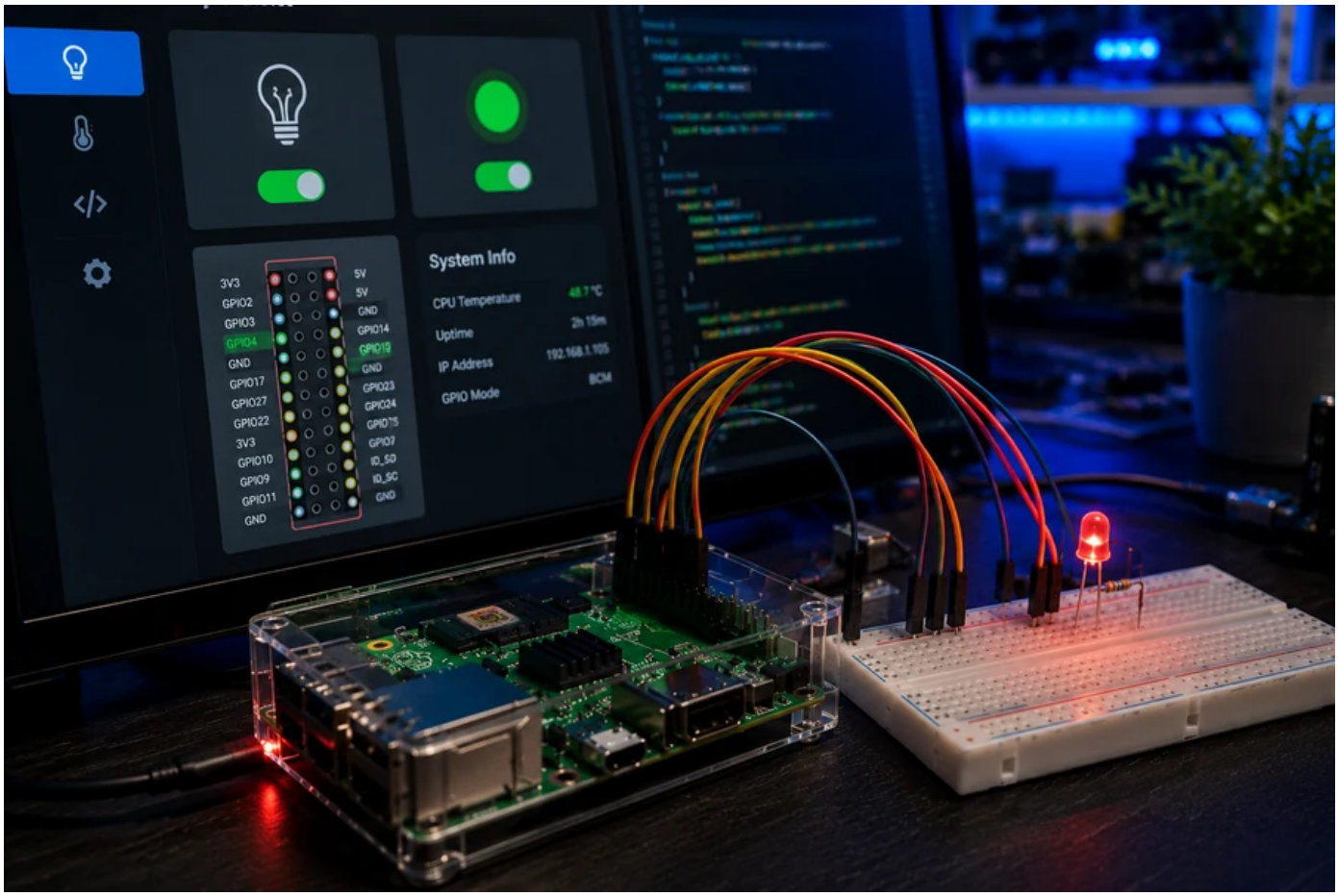
Always use a resistor with an LED. A Raspberry Pi GPIO pin should not drive an LED directly. Use a 220Ω or 330Ω resistor between GPIO17 and the LED long leg.

Parts Needed

Part	Purpose	Notes
Raspberry Pi 4 or 5	Runs Flask dashboard	Must be on the same network as the browser device.
LED	GPIO output indicator	Long leg is positive/anode.

220Ω or 330Ω resistor	Limits LED current	Required to protect the GPIO pin.
Breadboard	Temporary wiring	Keeps the LED circuit clean.
Jumper wires	GPIO connections	Use female-to-male wires for Pi header.

GPIO LED Wiring



The ZIP includes an editable SVG diagram: images/flask_web_dashboard_gpio_led_diagram.svg.

Connection	Wire To	Purpose
GPIO17 / physical pin 11	Resistor then LED long leg	Controls the LED from Flask.
LED short leg	Ground / physical pin 6	Completes the circuit.
Browser	http://PI-IP-ADDRESS:5000	Opens the dashboard from another device.

Install Flask

```
sudo apt update
sudo apt install python3-pip -y
pip3 install flask gpiozero
```

Find the Pi IP Address

```
hostname -I
```

Flask Dashboard Script

```
from flask import Flask, redirect
from gpiozero import LED
from datetime import datetime

app = Flask(__name__)
led = LED(17)

@app.route("/")
def home():
    state = "ON" if led.value else "OFF"
    now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    return f'''
    <h1>WolfieWeb Pi Dashboard</h1>
    <p>LED status: <strong>{state}</strong></p>
    <p>Last refresh: {now}</p>
    <a href="/on">Turn LED ON</a> | <a href="/off">Turn LED OFF</a>
    '''

@app.route("/on")
def on():
    led.on()
    return redirect("/")

@app.route("/off")
def off():
    led.off()
    return redirect("/")

app.run(host="0.0.0.0", port=5000)
```

Save as app.py and run with python3 app.py. Then open http://PI-IP-ADDRESS:5000 from a phone, tablet, or desktop on the same network.

Troubleshooting

Problem	Likely Cause	Fix
Dashboard opens only on Pi	Flask bound to localhost	Use host='0.0.0.0'.
LED does not change	Wiring wrong or wrong GPIO number	Use GPIO17, physical pin 11.
Page cannot be reached	Wrong IP, firewall, or different Wi-Fi	Run hostname -I and stay on same network.
LED stays dim	Wrong resistor placement	Put resistor in series with LED.

Upgrade Path

After this works, add sensor readings, servo controls, MQTT status cards, OpenCV latest-capture previews, or SQLite charts. This dashboard can become the front panel for every advanced tutorial on the page.