This project runs a Flask-based web UI for controlling a paintball turret (Pan/Tilt + Fire system).
The UI handles button controls, title, and animation. It's designed to run locally on a Raspberry Pi and be expanded to control GPIO pins for motor and firing functions.

## 1️⃣ Install Requirements on Raspberry Pi

Open a terminal on the Pi and run:

**sudo apt update**
**sudo apt install python3 python3-pip git -y**

Then install Flask:

**pip3 install flask**

## 2️⃣ Clone or Download the Repository

If you're using GitHub:

**git clone https://github.com/<your-username>/<repo-name>.git**
**cd <repo-name>**

Or if someone sends you a ZIP:

- Download and extract it
- Open a terminal in the extracted folder

## 3️⃣ Run the UI

Make sure you're in the project directory (the same folder as **app.py**), then run:

**python3 app.py**

If it runs successfully, you'll see something like:

*** Running on http://0.0.0.0:5000/**

Then, on the Pi's web browser (or another device on the same network), go to:

**http://<pi-ip-address>:5000**

Tip: Run **hostname -I** on the Pi to find its IP address.

## 4️⃣ Modify the UI

**To change how it looks:**

- Open static/style.css for colors, sizes, animations
- Open templates/index.html to change layout or add buttons

**To change what the buttons *do*:**

- Open app.py

- Find the route for each button, like this:

```
@app.route("/fire")
def fire():
    print("FIRE pressed")
    return "Fire command sent"
```

You can replace the **print()** with GPIO logic later, like:

```
import RPi.GPIO as GPIO
GPIO.output(FIRE_PIN, GPIO.HIGH)
```

(That should only be done after confirming wiring & safety.)

## 5️⃣ Stop the Server

When you're done, stop the app with:

**CTRL + C**

## 6️⃣ Optional: Run Automatically on Boot

If you want the UI to start every time the Pi boots:

1. Edit the crontab:

```
crontab -e
```

2.

3. Add this line at the bottom:

```
@reboot cd /home/pi/<repo-name> && python3 app.py &
```

4.

## 7️⃣ GPIO Integration Notes

For later:

- GPIO logic goes in app.py under each Flask route
- All pin mappings and signal timing are detailed in the provided hardware notes
- Make sure to add GPIO.setmode(GPIO.BCM) and proper cleanup calls when you integrate motors

## 8️⃣ Folder Structure

```
paintball-ui/
│
├── app.py              # Flask server
├── static/
│   └── style.css       # Styles, colors, animations
├── templates/
│   └── index.html      # UI layout
├── hardware_pins.txt   # Wiring map and notes
└── README.md           # This guide
```