Cat Feeder Application
User Guide

Cat Feeder Capstone Group
Penn State University
in collaboration with AstroLabe LLC

May 6, 2025

1 Getting Started

1.1 Quick Start

1. **Plug in the feeder.** The device boots and starts its built-in Wi-Fi hotspot named **ClaudeNet**.
2. **Connect to *ClaudeNet*.** Join the hotspot with WPA2 password `StrongPassword123`.
3. **Open the web app.** In a browser, go to `http://10.42.0.1:3000`. The Cat Feeder dashboard should load within a few seconds.

1.2 Scheduling Feeds

- **Hourly grid.** The home page shows a 24-hour grid. Tick a checkbox to schedule a feed at that hour.
- **Custom times.** Click *Add Time*, choose an hour and minute, then press *Save*. Multiple custom times are allowed.
- **Persistence.** Schedules save instantly and survive power cycles.

1.3 Feeding History

The *History* tab logs every dispense event with its timestamp and portion size. Use the date-range filter to narrow results.

1.4 Camera & Audio

- **Live camera.** Select *Camera* to watch a real-time preview.
- **Sound cues.** Press *Play Sound* to trigger a chime through the onboard speaker. A chime also sounds automatically whenever food dispenses.

1.5 Instant Dispense

Press the *Dispense Now* button (lower-right) for an immediate feed. The event appears in *History* and plays the standard chime.

1.6 Restarting the Feeder

If the dashboard becomes unresponsive or you lose Wi-Fi, unplug the feeder, wait five seconds, and plug it back in. The hotspot re-appears in ~30 s.

2 Advanced Use

2.1 SSH Access

1. Make sure your laptop is connected to **ClaudeNet** or the same LAN.
2. Open a terminal and connect via mDNS: `$ ssh CatFeeder@raspberrypi.local` If mDNS fails, use the hotspot IP (`10.42.0.1`) or the DHCP address shown in the dashboard footer.
3. Default username is `pi`; enter the password you set while imaging.
4. For key-based login, generate and copy your key:

2.2 Common SSH Issues

"Host key changed" warning Users see this after re-flashing the SD card. The exact banner:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.
Please contact your system administrator.
Add correct host key in /Users/<you>/.ssh/known_hosts to get rid of this message.
Offending ED25519 key in /Users/<you>/.ssh/known_hosts:25
Host key for raspberrypi.local has changed and you have requested strict checking.
Host key verification failed.
```

```
Run the following:
$ ssh-keygen -R raspberrypi.local
or
$ ssh-keygen -R 10.42.0.1
```

Timeouts Confirm the hotspot is active and that your firewall allows outbound SSH (port 22).

2.3 Troubleshooting

- **Cannot load app:** Check you are connected to **ClaudeNet**. Restart if necessary (subsection 1.6).
- **Lost schedule settings:** Ensure you clicked *Save*. If issues persist, verify SD-card space via SSH or restart (subsection 1.6).
- **Camera not streaming:** Refresh the tab or ensure another client is not holding the stream.

3 Codebase Overview

The project is split into a TypeScript/Next.js web frontend and several lightweight Flask micro-services that handle hardware interaction. All paths below are relative to the repository root `catfeeder/`.

3.1 Frontend (Next.js 14)

- `src/app/page.tsx` – top-level layout: displays the banner and decides which component tree to render for the current route.
- `src/components/controlsTest.tsx` – UI for feed scheduling, instant dispense, camera preview, and sound controls. Nearly all React state-logic for the dashboard lives here.
- `src/app/api/py/brightness/[brightness]/route.ts` – an API route that routes `PUT` requests from the React UI to the motor-control Flask service (see below).
- Static assets (icons, styles) live under `public/`.

3.2 Hardware Services (Flask)

Each hardware domain runs in its own Python virtual environment so that library dependencies (e.g. `gpiozero`, `flask`) stay isolated.

`camera_service/` Streams MJPEG frames from the CSI camera. Exposes `/stream` and `/snapshot` endpoints used by the React *Camera* tab.

`audio_service/` Plays short mp3 files over the speaker.

`gpio/` Contains `main.py`, a tiny Flask app that turns the stepper motor to dispense kibble. Requests arrive from the Next.js API route listed above.

Lifecycle At boot, `start-hotspot.service` ensures a fallback network. Once the OS finishes loading, `catfeeder.service` launches the production build from `.next/`. If you push new code:

1. Pull changes, rebuild, and restart the service (subsection 3.5).
2. Logs are viewable with `sudo journalctl -u catfeeder.service -f`.

3.3 Inter-Service Flow

1. React component dispatches `fetch('/api/py/brightness/200',{method:'PUT'})`.
2. Next.js API route forwards the call to `gpio/main.py` on the Pi using `fetch` (server-side).
3. `gpio/` toggles GPIO pins via `gpiozero.LGPIOFactory` and replies `200 OK`.
4. The API route returns success to the browser, and UI state updates (portion history).

3.4 Redeploying the Web App (systemd)

After pushing new frontend code, reload the feeder and confirm the update:

3.5 Redeploying the Web App (systemd)

After pushing new frontend code, reload the feeder and confirm the update:

1. **Reboot the device**
   Unplug the Raspberry Pi for five˜seconds, plug it back in, **or** run
   `sudo reboot`

2. **Reconnect to the hotspot**
   Wait about 30˜s for **ClaudeNet** to reappear, then join it with
   the WPA2 password `StrongPassword123`.

3. **Verify the update**
   Point your browser to `http://10.42.0.1:3000`. The dashboard should show
   the latest changes.

4. **If the update did not appear, redeploy manually**

   (a) SSH in: `ssh CatFeeder@10.42.0.1`

   (b) Remove the old build: `rm -rf .next`

   (c) Build a fresh bundle: `npm run build`

   (d) Restart the service: `sudo systemctl restart catfeeder.service`

   Reload `http://10.42.0.1:3000`; the changes should now be live.