

Afterhours - Course Outline

Session: Afterhours Spring '26

Location: ACUD MACHT NEU – School of Machines, Veteranenstraße 21, 10119 Berlin

Instructor: Sarah Grant

Office Hours: [Mondays + Fridays 10:00-12:00](#)

Contact: teaching@chootka.com

Description

This course explores how to develop your own digital tools - from self-hosted services to custom scripts and alternative infrastructure. Moving beyond being passive users of corporate platforms, we'll learn to set up our own servers, networks, AI systems, and communication tools. Questions around control of data, software, hardware and infrastructure will be tied to larger themes of autonomy, resilience, and community self-sufficiency. An ethos of artistry and activism will be infused throughout as students develop their own toolkit of custom tools, self-hosted services, and alternative infrastructure. This course is for you if you want to understand how the systems you rely on actually work, explore the creative potential of these technologies in your practice, or gain the skills to build and maintain alternatives that serve your communities. No previous experience is necessary, although a willingness to go behind the GUI and get your hands dirty with a computer console will help you get the most out of this course. Plan for approximately 8 hours per week including class time, readings, and project development.

Learning Objectives

- Gain foundational command line skills and systems administration knowledge
- Learn to set up and manage your own web servers and self-hosted applications
- Understand how to run your own local AI/LLM infrastructure
- Build alternative communication networks (mesh, peer-to-peer, radio)
- How to identify which tools serve your practice and assemble them into your own personalized toolkit

Core Technical Areas

- Command line interface and basic scripting
- Self-hosted web servers and applications (Yunohost)
- Local knowledge management (Obsidian)
- Self-hosted LLMs (Ollama) and integration strategies
- Mesh networking (Meshtastic)
- Peer-to-peer protocols (IPFS)
- Radio communication and alternative networks
- Embedded computing with Raspberry Pi

- Additional tools and topics based on student needs and collective curiosity

Structure & Teaching Methodology

This will be a 12-week course consisting of slides, class discussions, and hands-on workshoping supplemented by optional readings. By the end of the program, everyone will have worked toward developing a collection of self-hosted tools and infrastructure that complements their practice. Together we will each develop a personalized toolkit that may consist of servers, networks, AI systems, communication tools, and new ways of thinking about digital creativity and autonomy.

Class time will consist of both discussions and hands-on workshop sessions. Lectures will provide context on movements toward technological self-sufficiency and the politics of infrastructure. Workshop sessions will serve as practical introductions to the software and hardware needed for building our own systems. Outside of class, we will have suggested readings to help deepen our understanding of concepts learned in class.

Throughout the course, students will have the opportunity to ask questions, share ideas, and receive feedback on their work as we progress. Ultimately the goal is to gain both the technical skills and conceptual framework to build, maintain, and share digital tools that assist us in our creative and activist practices.

12-Week Semester Plan

Weeks 1-2: Command Line Basics

Topics:

- Introduction to self-hosted infrastructure;
- terminal navigation and file system;
- permissions and environment variables;
- bash scripting fundamentals;
- automation with cron;
- debugging in the terminal.

Readings: TBD

Goal: Understand how to control systems without a GUI and automate repetitive tasks.

Deliverable:

A collection of bash scripts that automate something in your workflow (backup, file organization, system monitoring).

Weeks 3-4: Self-Hosted Web Server

Topics:

- Server concepts and architecture;

- Yunohost installation and configuration;
- DNS basics; deploying web applications;
- user management and permissions;
- backup and basic security.

Readings: TBD

Goal: Run your own server and deploy applications you control.

Deliverable:

A working Yunohost server hosting at least one application (blog, wiki, file share, or portfolio site).

Weeks 5-6: Mesh Networking & Off-Grid Communication

Topics:

- LoRa radio fundamentals;
- Meshtastic setup and configuration;
- mesh network topology;
- MQTT protocol; bridging mesh messages to your web server;
- designing for resilience.

Readings: TBD

Goal: Build communication infrastructure that doesn't depend on the internet.

Deliverable:

A Meshtastic node that sends sensor data or messages to your web server via MQTT.

Weeks 7-8: Self-Hosted AI

Topics:

- Local LLM concepts;
- Ollama installation and model management;
- running inference locally; API integration;
- connecting LLM to web interfaces; prompt design for personal tools.

Readings: TBD

Goal: Run AI on your own hardware without cloud dependencies.

Deliverable:

A locally-hosted LLM accessible via web UI or integrated into your server (e.g., chatbot, writing assistant, or custom tool).

Weeks 9-10: Peer-to-Peer & Local-First Tools

Topics:

- Decentralization concepts;
- IPFS installation and content addressing;
- pinning and persistence strategies;

- Obsidian for knowledge management;
- local-first philosophy;
- sync without cloud services.

Readings: TBD

Goal: Distribute content and manage knowledge without centralized platforms.

Deliverable:

A personal knowledge base in Obsidian plus content published and accessible via IPFS.

Weeks 11-12: Integration & Networked Autonomy

Topics:

- Raspberry Pi deployment;
- connecting systems together;
- documentation practices;
- maintenance and sustainability;
- conceptualizing autonomy through infrastructure.

Readings: TBD

Goal: Assemble a personal toolkit and articulate its role in your practice.

Deliverable:

Final Project: An integrated system using at least three technologies from the course (e.g., a Raspberry Pi running a web server that receives mesh network data and queries a local LLM, or an IPFS-hosted site updated via Obsidian and announced over Meshtastic).

Required Materials

- Laptop
- Raspberry Pi 4+
- 32GB MicroSD card
- Enclosure
- Power supply
- Ethernet cable
- Ethernet - USB adapter
- Heltec v3 ESP32
- USB-C cable (for Heltec)