PROJECT 0 — (GITHUB)
SETTING UP YOUR CSCI 4208 PORTFOLIO REPOSITORY

PROJECT GOAL

Build a public “portfolio + project board” you’ll use all semester to (1) list and link your labs/projects/capstone, (2) track work with GitHub Issues and a Project (v2) board, and (3) use light automation (GitHub Actions) and GitHub Pages for client-side deployments.

LEARNING OBJECTIVES

- Practice project management in GitHub (Issues, labels, milestones, boards)
- Use git/GitHub workflows (forks, commits, pushes)
- Run and interpret GitHub Actions (workflows)
- Deploy client-side web apps to GitHub Pages (no server required)

PREREQUISITES

- A GitHub account
- Basic git knowledge (commit/push) or comfort using the GitHub web UI

WHY WE’RE USING GITHUB’S TOOLS & SERVICES

GitHub’s Repositories, Issues, Project boards, Actions, and Pages are industry-standard. You’ll finish the course with a public artifact (portfolio + board) you can include in résumés and interviews that demonstrates planning, execution, and delivery.

TOOLS & SERVICES WE’LL TOUCH

- GitHub Repositories ·
- GitHub Actions ·
- GitHub Issues ·
- GitHub Project (v2)
- Views (Roadmap, Table, Board) ·
- GitHub Pages

SECTION A — GITHUB REPOSITORIES (CREATE YOURS)

A1. Fork the template

1. Open the template: https://github.com/scalemailted/csci4208-portfolio-2025
2. Click **Fork**. Name your repo exactly: **csci4208-portfolio-2025**.

A2. Turn on required features (in your fork)

• Settings → General → Features → check **Issues**
• Open the **Actions** tab → click **Enable** (first time only)

A3. (Recommended) Turn on GitHub Pages now

• Settings → Pages → "Deploy from branch" → Branch: **main**, Folder: **/**
• Your site (after a minute):
`https://<username>.github.io/csci4208-portfolio-2025/`

A4. Required settings — checklist

□ Issues: ON (Settings → General → Features → Issues)
□ Actions: Enabled (open Actions at least once)
□ Pages: ON (Settings → Pages → main / /)
□ After you make a Project board later, set Project → Settings → **Make public**

A5. Repository layout (where things go)

```
/
├─ index.html (your portfolio landing page; add links here)
├─ labs/ (each lab in its own folder: labs/lab-01/, labs/lab-02/, …)
├─ projects/ (pointer folders: projects/project-01/README.md → external project repo + live URL)
├─ capstone/ (pointer folders per milestone: capstone/capstone-01/README.md → team repo + board)
└─ _docs/ (this handout / course how-to notes)
```

Notes:

• **Labs:** submit source here; if static, use `index.html` as entry (lab document will specify).
• **Projects:** code lives in a **separate public repo**; in this repo you provide a pointer README + demo steps.
• **Capstone:** team repo + team Project board; each milestone folder here contains a pointer README + links.

SECTION B — GITHUB ACTIONS (AUTOMATION)

B1. Seed / Update Roadmap Issues (Action #1)

What it does:
Pulls the instructor roadmap and **creates/updates Issues** with labels and milestones (Modules).

How to run:
Actions → "Seed / Update Roadmap Issues" → Run workflow

When to run:
After the instructor posts changes, or allow the nightly schedule (if enabled) to run.

B2. Update README Submission Table (Action #2)

What it does:
Rewrites a compact table in your README showing **Item | Key | Status** based on Issue **labels** (or open/closed).

How to run:
Actions → "Update README Submission Table" → Run workflow

Also runs:
nightly (if scheduled), after seeding, on issue changes (opened/edited/labeled/closed), and on pushes to

- `labs/**`,
- `projects/**`,
- `capstone/**`,
- `index.html`.

***Tip**: If a job "does nothing," open its run logs—logs explain what changed or why nothing changed.*

SECTION C — GITHUB ISSUES (YOUR SINGLE SOURCE OF TRUTH)

C1. What Issues are (and your role)
Every roadmap item (lab, project, capstone milestone) is an **Issue**. It includes an **Acceptance** checklist you must satisfy before asking for approval.

C2. Issue properties (what you'll see)
- **Title:** assignment name (e.g., "Lab-02 — CSS: Crash Course")
- **Labels:**
  – Type: `lab`, `project`, `capstone`
  – Key: `lab-01`, `project-01`, `capstone-01` (**do not change**; updates match by key)
  – Tracks: `Frontend`, `Backend`, `DB`, `Security` (apply all that fit)
  – Status: `reviewed`, `ready-for-approval`, `blocked`, `completed`, `approved`, `deployed`
  – Parent tag: `task` (added automatically)
- **Milestone:** the Module (used for grouping on the board)
- **Checklist:** deliverables to satisfy (tick all before requesting approval)

C3. Changing state & labels
- **Open/Close**: use the button near the title. Keep open while working.
- **Labels**: right sidebar → **Labels** (apply/remove status & track labels).

C4. Searching & filtering (Issues tab)
- Open labs: `is:issue is:open label:lab`
- Module 1 items: `is:issue label:module-01`
- Ready for approval: `is:issue label:ready-for-approval`

SECTION D — GITHUB PROJECT BOARD

D1. Create & connect your Project board

1. **New Project** → choose **Board** view

2. Project **Settings** → **Workflows** → **Auto-add to project**
   – Repositories: select **this** portfolio repo
   – Filter: `is:issue` → toggle **ON**

3. **Workflows** → **Item added:** set **Status = Todo** (or similar)

4. In the view: **Column field = Status**, save as **Board** view

5. Optional: **Group by = Milestone** (rows per Module), save as **Roadmap** view

6. Optional: switch to **Table** view (filter/sort), save as **Table** view

7. Project **Settings** → **Make public** (shareable link for résumé/reviews)

D2. How to use the views:

- **Board (kanban):** daily flow (Todo → In progress → Done)
- **Roadmap (Milestone rows):** see items by Module
- **Table:** filter/sort by labels, milestones, assignees

Note: *Our README status table uses* ***Issue labels*** *(token-free), not the board's internal Status.*

SECTION E — GITHUB PAGES (DEPLOYING NO-SERVER WEB APPS)

E1. What qualifies

Client-side apps: HTML/CSS/JS only (no custom server). Typical for static labs and early projects.

E2. Enable Pages (once per repo)

- Settings → **Pages** → "Deploy from branch" → Branch: **main**, Folder: **/**
- Wait ~1 minute; visit `https://<username>.github.io/csci4208-portfolio-2025/`

E3. Place your entry file(s)

- **Labs:** `labs/<lab-key>/index.html` (link it from the root `index.html`)
- **Projects (static):** deploy in the **external project repo** to its own Pages URL, or include a static build and link to it.
- Use **relative paths** for assets (`./css/style.css`, `./img/...`). Case matters.

E4. Verify & link

- Open your Pages URL and confirm it loads.
- Add the live link to the Issue and apply **`deployed`**.
- Add the live link to your root `index.html` portfolio page.

E5. Common pitfalls (and fixes)

- **404 after enabling Pages** → wait; ensure there's an `index.html` at the target path
- **Asset 404s** → fix capitalization; keep `./` relative paths
- **Single-page frameworks** → avoid custom client routing for this course unless instructed

---

SECTION F — DAILY WORKFLOW LOOP (WHAT YOU'LL REPEAT)

1. Pick an Issue from your board
2. Do the work in `labs/<key>/` or in your external project repo
3. Commit/push; update the Issue checklist + apply a **status label**
4. Re-run the **README Submission Table** workflow (or let it auto-trigger)
5. Move the card on the **Board** (e.g., Todo → In progress → Done)

TROUBLESHOOTING QUICK LIST

• "Issues disabled" → Settings → General → Features → enable **Issues**
• "Actions didn't change anything" → open run logs (often the file already matched)
• Board empty → Project → Workflows → **Auto-add** ON; use **Add items** once to backfill; or re-run the seeder
• Pages 404 → ensure Pages is enabled, target has `index.html`, wait a minute, fix asset paths/case

---

CHECKLIST TO TURN IN WITH PROJECT 0

□ Forked repo: **csci4208-portfolio-2025**
□ Features enabled: **Issues**, **Actions**, **Pages**
□ Project (v2) board created, auto-add ON, **public**
□ Seeded roadmap issues appear (with labels/milestones)
□ README status table shows items (Item | Key | Status)
□ Root `index.html` exists with at least one link (lab/project)
□ Submitted the three links (repo, board, Pages)

GRADING RUBRIC (PROJECT 0)

Part 1: [Repos]	Fork template; set description; update README; correct layout	[20%]
Part 2: [Actions]	Run seeder + README table actions; verify logs/changes	[20%]
Part 3: [Issues]	Keys/labels correct; Acceptance checklist used; filters shown	[20%]
Part 4: [Project]	Project: Auto-add ON; Board/Roadmap/Table views; public link	[20%]
Part 5: [Pages]	GitHub Pages enabled; root index.html; at least one live link	[20%]

DELIVERABLES

- Portfolio repo link: `https://github.com/<username>/csci4208-portfolio-2025`
- Public Project (v2) board link
- Live portfolio Pages URL

Make sure your board is **public** and your links are visible in your repo About or README.

SUBMISSION

- Post the link to your completed forked project in the Discord server in the GitHub channel