Week 1 Lab — Setup & Git (Extended Quick Guide)

Goal: Get your tools ready, make a tiny semantic web page, and practice Git/GitHub with a pull request.

What you'll submit: repo link, PR link, 2 screenshots, short reflection.

0) What you'll finish (deliverables)

- GitHub repo named ucu-dip-webdev-2025-lab01--
- A merged PR from a feature branch → main
- Files: index.html, .gitignore, README.md
- 2 screenshots: (1) PR page on GitHub (2) Chrome DevTools → Network tab showing 200 for the HTML file
- Short reflection (100–150 words) in the README

1) Install & verify tools (10–20 min)

Node.js (LTS):

```
Windows (PowerShell): winget install OpenJS.NodeJS.LTS
macOS (Homebrew): brew install node
Ubuntu/Debian: sudo apt-get update && sudo apt-get install -y nodejs npm
Verify:
  node -v
  npm -v
```

Git:

```
Windows: winget install Git.Git
macOS: brew install git
Ubuntu/Debian: sudo apt-get install -y git
Verify: git --version
```

VS Code: install from the website or use winget (Windows). Extensions: Prettier, Live Server (optional), GitLens (optional).

2) Configure Git (once, any OS)

```
git config --global user.name "Your Name"
git config --global user.email "you@example.com"
git config --global init.defaultBranch main
# Windows only (line endings):
git config --global core.autocrlf true
# macOS/Linux:
git config --global core.autocrlf input
```

GitHub account: create one if you don't have it. If Git asks for a password on push over HTTPS, use a Personal Access Token (PAT).

3) Start your project (15–20 min)

```
mkdir lab01-hello-web && cd lab01-hello-web
```

.gitignore (starter)

```
# Node/npm
node_modules/
.npmrc
# Logs
*.log
npm-debug.log*
# OS files
.DS_Store
Thumbs.db
# Env files
.env
```

index.html (simple, semantic)

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>UCU Noticeboard — Lab 01</title>
</head>
<body>
  <header>
    <h1>UCU Noticeboard</h1>
    <nav aria-label="primary">
      <a href="#news">News</a>
      <a href="#events">Events</a>
      <a href="#contact">Contact</a>
    </nav>
  </header>
  <main>
    <section id="news" aria-labelledby="news-title">
      <h2 id="news-title">Latest News</h2>
      <article>
        <h3>Welcome to Web & Mobile Development</h3>
        <p>This is your first semantic HTML page. Keep it simple and clear.</p>
      </article>
    </section>

    <section id="events" aria-labelledby="events-title">
      <h2 id="events-title">Upcoming Events</h2>
      <ul>
        <li>Orientation — Wed 10:00–12:00</li>
        <li>Git Workshop — Fri 14:00–16:00</li>
      </ul>
    </section>

    <section id="contact" aria-labelledby="contact-title">
      <h2 id="contact-title">Contact Us</h2>
      <form>
        <label for="email">Email</label>
        <input id="email" type="email" placeholder="you@example.com" required />
        <button type="submit">Send</button>
      </form>
    </section>
  </main>
  <footer><small>© 2025 Uganda Christian University</small></footer>
```

```
</body>
</html>
```

README.md (template)

```
# Lab 01 — Environment Setup & Git Workflow

## Tasks
- [x] Install Node LTS, Git, VS Code
- [x] Configure Git user and default branch
- [x] Initialize repo and create semantic `index.html`
- [x] Create `.gitignore` and `README.md`
- [x] Open a Pull Request from a feature branch

## How to run
Open `index.html` in a browser (or use VS Code Live Server).

## Screenshots
1) PR page  2) DevTools → Network tab (HTML request shows 200)

## Reflection (100–150 words)
- What is an HTTP request/response?
- Which status code did you see, and what does it mean?
- One thing you learned about Git today.
```

4) First commit → push to GitHub (10 min)

```
git init
git add .
git commit -m "chore: scaffold lab01 with index.html, README, .gitignore"
# create an empty GitHub repo named:
# ucu-dip-webdev-2025-lab01-<firstname>-<studentID>
git remote add origin https://github.com/<you>/ucu-dip-webdev-2025-lab01-<firstname>-<studentID>.g
git branch -M main
git push -u origin main
```

5) Create a feature branch & open a PR (15–20 min)

```
git checkout -b feature/add-news-item
# make a tiny change: add a <li> in 'Upcoming Events', or a line in the article
git add index.html
git commit -m "feat: add first news item to noticeboard"
git push -u origin feature/add-news-item
```

On GitHub: open a Pull Request (base=main, compare=feature/add-news-item), add a short description and a simple checklist, then Merge.

6) Quick web check (5–10 min)

Open your page in Chrome → Right-click → Inspect → Network. Check the document row (HTML) shows Status 200. Disable cache while DevTools is open, then refresh. Take a screenshot.

Submission summary

- Repo link & PR link
- Files present: index.html, .gitignore, README.md
- Two screenshots in README (PR page + Network tab)
- Reflection added to README

Grading (4 pts)

- Tools installed & versions shown — 1.0
- Proper Git workflow (branch → PR → merge) — 1.5
- Semantic HTML + basic a11y (labels/alt) — 1.0
- Clean commits + .gitignore + README — 0.5

Common fixes

- Auth failed on push: use a GitHub Personal Access Token as HTTPS password.
- Wrong default branch: git branch -M main then push again.
- Remote already exists: git remote -v → git remote remove origin → re-add.
- Live preview: use VS Code Live Server.

Mini cheat-sheet

Good commit messages: feat: add contact form | fix: correct header semantics | chore: update README

Status codes: 200 OK, 301/302 redirect, 404 not found, 500 server error

Semantics: header, nav, main, section, article, footer; add alt on images; label every input

Optional bonus (5–10 min)

- Publish via GitHub Pages (Settings → Pages → from main).
- Validate HTML and fix one warning.