Nicol – Personal Portfolio Project (Client-Style Brief)

Owner: Nicol
Mentor: Ken
Project Type: Portfolio website — shipped in stages (V1 now, V2 after Front End Libraries cert)
Repo: (create a new GitHub repo named `nicol-portfolio`)
Deploy: GitHub Pages (V1), later migrate to a React build for V2

---

1) Objective

Create a clean, fast, accessible personal portfolio that: - Introduces who you are (short bio + photo/avatar). - Showcases FreeCodeCamp projects with clear links and short descriptions. - Lets people contact you easily. - Demonstrates basic JS interactivity now (V1), and modern front-end libraries later (V2).

---

2) Delivery Plan (Progression)

V1 (Immediately after JavaScript Algorithms cert): - **Stack:** HTML5, CSS3, vanilla JS - **Pages/Sections:** Home (hero), Projects, About, Contact (simple form or mailto link) - **Enhancements (2–3 required):** see §4 - **Deploy:** GitHub Pages

V2 (After Front End Development Libraries cert): - **Stack:** React + a styling approach (Bootstrap or Sass) - **Upgrades:** Componentized layout, better styling, optional dark mode, smoother animations, improved projects grid, SEO/meta polish. - **Deploy:** React build to GitHub Pages or Netlify.

(Optional V3 later: add blog/notes, simple CI for build checks, Lighthouse 90+ targets.)

---

3) V1 Deliverables

1. Public GitHub repo with a **README.md** that includes:
2. Project purpose, stack, how to run locally, and deployment link.
3. Live site via **GitHub Pages**.
4. Four core sections:
5. **Hero:** name, short one-line value statement, call-to-action button to Projects.
6. **Projects:** grid/cards of FCC projects (title, 1–2 sentence summary, tech used, link to live/demo & code).
7. **About:** short bio (2–3 sentences) + skills list (HTML, CSS, JS; mark what you're currently learning).
8. **Contact:** mailto link or simple form (non-backend; validate in JS, then `mailto:` or link to your email/LinkedIn).
9. **2–3 vanilla JS enhancements** (see next section) implemented cleanly.

10. Basic accessibility & performance passes (see §6).

---

4) V1 JavaScript Enhancements (choose 2–3)

- **Filter projects by tag/tech** (buttons to show All / HTML / CSS / JS).
- **Dynamic Projects List** sourced from a small local `projects.json` (or inline JS array) so adding projects doesn't require editing HTML.
- **"Copy link to clipboard"** button for each project.
- **Theme toggle (light/dark)** persisted with `localStorage`.
- **Form validation** for Contact (check empty fields, email pattern with regex).

> **Required:** Include your **FreeCodeCamp project links** (live + code). At least 4 items to start. This can be the data backing your dynamic Projects list.

---

5) Content Requirements

- **Bio:** 2–3 sentences; keep it positive and factual.
- **Avatar/photo:** Optional; keep file size small.
- **Project cards:** title, short description (what problem, what you learned), tags (e.g., `html`, `css`, `js`).
- **Contact:** mailto link or a basic non-backend form with client-side validation.

---

6) Design, Accessibility & Performance

- **Layout:** mobile-first; responsive grid for projects.
- **Typography/Color:** 1–2 fonts max; 1 primary color + neutral shades; ensure contrast (use WebAIM Contrast Checker).
- **A11y:** semantic HTML (landmarks: header/main/footer/nav), alt text for images, focus styles visible, aria-labels where helpful.
- **Performance:** images ≤ 200KB, use `loading="lazy"` for project images; avoid heavy libraries in V1.

Acceptance checks (V1): - Pages render well on mobile (375px), tablet (~768px), laptop (≥1280px). - All project links work. - JS enhancements function without errors (check devtools console). - Lighthouse: Performance ≥ 80, Accessibility ≥ 90, Best Practices ≥ 80, SEO ≥ 80.

---

7) Git/GitHub Workflow (simple & safe)

Setup (once): 1. Install Git, set name/email: `git config --global user.name "Nicol"` `git config --global user.email "you@example.com"` 2. Create repo on GitHub: `nicol-portfolio`. Clone it locally.

Branching model: - `main` = always deployable. - For each feature/fix, create a branch: `feature/projects-grid`, `feat/theme-toggle`, `fix/typo-readme`.

Routine: 1. `git checkout -b feature/<name>` 2. Do the work, test locally. 3. Commit small and often (present tense): - `git add .` - `git commit -m "add projects grid and card layout"` 4. Push and open a Pull Request (PR) to `main`. 5. Self-review using the checklist (below), then merge.

PR Self-Review Checklist: - Does it build/run locally without console errors? - Is the HTML semantic and valid? - Are styles responsive at common breakpoints? - Are JS enhancements accessible (keyboard/ARIA) and resilient (null checks)? - Are links working? Spelling checked?

Avoid Git overload: - Keep commits small; one idea per commit. - Don't rewrite history; if stuck, make a new branch and try again. - Use GitHub Desktop if the CLI feels heavy (optional).

---

8) Deployment (V1)

1. In GitHub repo → **Settings → Pages** → Source: `main` / `/root` → Save.
2. Ensure entry file is `index.html` at repo root.
3. Wait 1–2 minutes for the site URL to go live. Add the URL to README.

---

9) V2 Upgrade (after Front End Libraries)

- **Rewrite in React** (Vite or Create React App).
- Convert sections to components: `Header`, `ProjectCard`, `ProjectsGrid`, `About`, `Contact`, `ThemeToggle`.
- **Styling:** Bootstrap or Sass; ensure consistent spacing and a system for colors/sizes.
- **Data:** move projects data to a JSON file or fetch from GitHub API/Gist.
- **Enhancements:**
- Dark mode (context + `localStorage`).
- Animations (CSS transitions or a lightweight lib).
- Accessibility review and keyboard testing.
- **Deploy:** build to GitHub Pages or Netlify; update README and links.

Acceptance checks (V2): Lighthouse targets P≥ 90, A≥ 95, BP≥ 90, SEO≥ 90. No console warnings in production build.

---

10) Timeline & Expectations

- **Start:** Begin **immediately after** completing the JavaScript Algorithms cert.

- **V1 Duration Target:** 1–2 weeks of steady, part-time effort.
- **Review:** Open a PR when V1 is live; share repo + Pages link for mentor review.
- **V2 Trigger:** Begin once the Front End Development Libraries cert is completed (React/Bootstrap/Sass learned).

11) Support & References

- **MDN Web Docs:** HTML, CSS, JS references and examples.
- **FreeCodeCamp Forum:** project help and examples.
- **Lighthouse:** Chrome DevTools → Audits (check performance & accessibility).
- **Contrast Checker:** WebAIM Contrast.

12) Kickoff Note to Nicol

Nicol, you'll treat this like a real client build: follow the brief, ship a clean V1, then iterate after your next cert. Keep scope tight, commit small, and ask questions when blocked. Start as soon as you finish your JavaScript Algorithms certification.

You've got this. Ship V1, then we level it up.