

Portfolio 2 — Reflection (Hilde-Kathrine Ljosland Vatne)

Student

- **Name:** Hilde-Kathrine Ljosland Vatne
 - **GitHub:** (<https://github.com/Entav78>)
 - **Portfolio site:** (<https://app.netlify.com/projects/portfolio2hildevatne/overview>)
-

Projects Overview

1. **CSS Frameworks** — Vite + vanilla JS with Bootstrap.
2. **JavaScript Frameworks (HildeShop)** — React + TypeScript + Tailwind.
3. **Semester Project 2 (TailFinder, resit)** — React + TypeScript + Tailwind + Zustand.

Each project includes:

- Live site
 - Public GitHub repo with updated README
 - One or more improvements completed for Portfolio 2
-

1) CSS Frameworks

Live: [Live](#)

Repo: [css-frameworks-portfolio2](#)

README: [README.md](#)

Goal & Scope (short)

- Practice using a CSS framework to layout a small social app (Noroff Social API).
- Focus on card grids, off-canvas nav, and a consistent single-post view.

What I improved for Portfolio 2

- Uniform card media sizing on Home/Profile pages (consistent grid heights).
- Simplified/cleaned nav; pointer cursor for all interactive elements.
- Single-post page: constrained hero image with aspect-ratio and lazy loading.
- Public routing build setup for Netlify (multi-page inputs in Vite + 200.html/_redirects).

Commits (with links):

- UI polish: unify card image sizes (home/profile), fix profile posts grid + header layout, add single-post responsive media, clean `styles.css` (scoped rules), add pointer cursors, and simplify Navigation.
[35a56c6](#)

- feat(ui): bootstrap card grid + profile layout.
[1fec737](#)
- feat(auth): skin login/register with Bootstrap cards + spacing.
[e68576b](#)

Reflection (what I learned / trade-offs)

- Bootstrap is fast for layout but opinionated; I prefer Tailwind for long-term projects.
 - The biggest visual win came from constraining **aspect-ratio** on card images.
 - Multi-page Vite requires explicit inputs and Netlify fallbacks.
-

2) JavaScript Frameworks — HildeShop

Live: [Live](#)

Repo: [Repo](#)

README: [README](#)

Goal & Scope (short)

- Ecommerce storefront consuming Noroff API; search/sort, product detail, cart, checkout success, contact.

What I improved for Portfolio 2

- **Reliable routing on Netlify** using `public/_redirects` (`/* /index.html 200`).
- **React context split:** separated `CartContext` hooks from `CartProvider` component to remove Fast Refresh warnings and simplify tests.
- Tests updated to wrap with `CartProvider` only where needed.

Commit links (evidence):

- **Add Netlify SPA fallback (`_redirects`) / deep-link refresh**
 - Initial redirect: [70172d6](#)
 - Follow-up fallback chore: [cd7253c](#)
 - Asset URL fix to keep SPA fallback working: [8a71432](#)
- **Split cart context/provider + test updates**
 - Commits touching the `src/context` folder (introducing `cart.ts` + `CartProvider.tsx`):
<https://github.com/Entav78/jsfw-2025-hilde/commits/main/src/context>
 - Test update after the split (import path fix): [00fc14a](#)

Reflection

- The context split improved DX and testing isolation.
 - Keeping SPA routing robust on hosts like Netlify is essential for deep links and refresh.
-

3) Semester Project 2 — TailFinder (resit)

Live: [Live](#) **Repo:** [Repo](#) **README:** [README](#)

Goal & Scope (short)

- Pet adoption app with auth, filters, paginated lists, image “reveal all” accessibility control.

What I improved for Portfolio 2

- **SEO & crawling:** added `robots.txt` (allow all) + `sitemap.xml` and `<link rel="sitemap">`.
- **Meta polish:** `<meta name="description">` + Open Graph/Twitter tags.
- Lighthouse SEO moved to **100** (Perf/Access/Best Practices remain high).

Commit links (evidence):

- **Add robots.txt, sitemap.xml, SPA rewrites, meta description** → [79d172a](#)
- **Update README** → [872089c](#)

Reflection

- Small SEO files (robots/sitemap) + meta description make a big difference.
 - Accessibility feature (“Reveal All Images”) remains a core UX choice.
-

Portfolio Site Notes

Live:

Repo:

Requirements checklist

- ☒ Homepage with three teaser cards.
- ☒ Thumbnails optimized (≤ 200 KB).
- ☒ Teaser text ≤ 120 –130 chars.
- ☒ Each teaser links to an **article page**.
- ☒ Article page includes: title + copy link, hero image + caption, live site link (new tab), GitHub README link (new tab), main content.
- ☒ Personal GitHub profile link is public.
- ☒ README files updated for all three projects.

Tech line (welcome section)

React · TypeScript · Tailwind CSS · Bootstrap · Vite · Netlify · REST APIs

Submission (Moodle)

Include:

1. **Portfolio website link (live)**
2. **Personal GitHub profile**
3. **Links to all three repos**

4. **Links to all three live projects**
 5. **This PDF reflection document**
 6. **5-minute video presentation**
-

Presentation Script (outline)

1. **Welcome** — Who I am and what I build (React/TS, accessibility & performance focus).
2. **Portfolio tour** — Home page; tools line; quick card → article flow.
3. **Project 1** — Goal, one visual improvement (uniform cards), single-post hero tweak, quick demo.
4. **Project 2** — SPA routing fix on Netlify; context split; quick demo of cart & tests note.
5. **Project 3** — SEO + meta/robots/sitemap; accessibility “reveal images”; show Lighthouse.
6. **Wrap** — Where to find code, what I’d improve next, thanks.