Dev Journal:

GitHub Repo: https://github.com/quinnwillett/QWPortfolio

Deployed Site: https://quinnwillett.github.io/QWPortfolio

Session 1: File Creation, Structure & Navigation

I started by creating the main structure of the site:
index.html, about.html, previousworks.html, projects.html, and contact.html.

I knew I wanted a clean, consistent navbar across every page, so I built a reusable navigation section and copied it into each file.

The biggest challenge early on was keeping spacing consistent. Some page headers felt too close to the nav, others too far. I solved this by adjusting the global CSS for margins and padding until everything felt balanced.

Next, I moved onto the home page. I built a two-column hero layout: text on the left and my profile image on the right. I added some funny stats, but the coolest feature I added here was the live timer showing how long I've been alive.

This was something I didn't learn in class, and building it required breaking down time into days, hours, minutes, and seconds. The full logic is in script.js

At first the timer didn't appear until a couple seconds had passed, which made it look broken. I fixed that by calling updateAliveCounter() right after defining the function.

Writing My Own CSS Theme
I then focused on styling. I didn't want to use any premade theme but I didn't know what would look good, so I looked at some other websites to gain inspiration. I decided to go with a cool retro purple gradient aesthetic. Kind of like being in a lava lamp, which is the inspiration of the shape in the home page.

This gradient and button styles became the main color of the website, such as in styles.css

```
.cta {
   background: linear-gradient(135deg, #6a3bff, #cf3bff);
}
```

Session 2: Contact Page: Web Form + Sliding Social Sidebar
Web3Forms Contact Form

I added a working contact form using Web3Forms, which lets the form submit directly to email without backend code. I styled the form using CSS Grid. The code is in contact.html

Sliding Socials Tab

This was another feature not taught in class. I wanted a sidebar that expands when clicked. I built it using CSS transitions and a small JavaScript toggle:

```
label.addEventListener("click", () => {
   tab.classList.toggle("open");
});
```

At one point the icons didn’t show because they inherited the wrong color, so I added fill="currentColor" to the SVGs to fix it.

Session 3: About Page & Media Tabs

This session was one of the most fun because it was the first time I intentionally built features we did not learn in class, which the assignment requires.

I added image “carousels” for my hobbies. Each one only shows one image now, but the structure allows for expansion later. The layout comes from about.html

Favorite Media, Interactive Tabs

The media tabs were the part I was most excited about. When you click a button (Movies, Games, TV, Albums), the content updates without reloading the page.

I wrote all the tab-switching logic in JavaScript:

```
tabButtons.forEach(btn => {
   btn.addEventListener("click", () => {
      document.querySelector(".tab-btn.active")?.classList.remove("active");
      btn.addClass("active");
      document.querySelector(".tab-panel.active")?.classList.remove("active");
      document.getElementById(btn.dataset.tab).classList.add("active");
   });
});
```

At first, nothing happened when I clicked the tabs. I realized I hadn’t included the <script> block in my html file. Even after this it still wouldn't work. After some debugging I realized I forgot to add the .active class to the default content panel. Once I added that, everything worked.

Session 4, Writing Previous Works & Projects Pages

I wanted the Previous Works page to feel like a mini résumé. I manually wrote the content, formatted everything with HTML, and added styling for certain headings to match my design and color palette. All the content came from me, no generator or template.

On the Projects page, I built two project showcases using large gradient backgrounds and text-boxes that overlap them. Getting the positioning right required some trial and error because I used position:absolute. After adjusting spacing, everything sat correctly.

Session 5, Final Polishing & Deployment

The last part of the process was reviewing every page for consistency, cleaning up spacing issues, and making sure everything worked

Outside Sources Used

I kept outside references minimal and only used them for specific questions:

Documentation

MDN Web Docs - checking syntax for classList, timers, event listeners
https://developer.mozilla.org

How to Create Tabs with Vanilla JavaScript - webcrunch
https://www.youtube.com/watch?v=rSEgjpMZXIc

Web3Forms Guide - setting up the form endpoint
https://web3forms.com

How To Hover With JavaScript
https://www.youtube.com/watch?v=1xEfqgElMsY

Inspiration

Looked at general color palettes and layout ideas on Dribbble (no code copied). Also looked at other fun website design ideas to see which aesthetic I wanted to go with.