

ASSIGNMENT SUBMISSION

Personal Portfolio Web Application

A fully responsive, API-integrated showcase of development skills.

Project Overview

- ✓ **Production Ready:** A polished, professional application representing the final stage of the coursework.
- 📦 **Full Stack Features:** Includes complex state management, API integration, and interactive UI components.
- 🔗 **Culmination:** Integrates all concepts learned from Assignments 1–4 into a single coherent product.



Personal Motivation



Professional Use

Build a real-world asset to demonstrate skills to future employers.



Skill Mastery

Learn to structure, debug, and polish a complete web application.



Problem Solving

Combine creativity with technical knowledge in a practical scenario.

Key Features & Interactivity



Theme System: Dark/Light toggle saved in localStorage.



User Logic: Greeting, visitor counter, and session timer.



GitHub API: Live project fetching with sorting & filtering.



Search: Real-time filtering by language and name.



UX Polish: Loading skeletons and error handling states.



Responsive: Collapsible details and mobile-friendly layout.

Code Architecture



HTML5

Semantic layout ensuring accessibility and proper section organization.



CSS3

Component-based styling, CSS variables for theming, and smooth animations.



JavaScript

Modular logic blocks, async API fetch functions, and event listeners.

AI-Assisted Development



Accelerating Workflow

AI tools were utilized to enhance code quality and solve complex logic hurdles:

- 🛠 Debugging async/await API issues.
- ⚡ Optimizing performance (Lazy loading).
- 🔗 Refining documentation structure.

Deep Dive: Challenges

⚠️ Hardest Challenges

- Combining sort, filter, and search into one clean pipeline.
- Managing GitHub API rate limits.
- Handling multiple UI states (Loading, Error, Empty).
- Ensuring state persistence across sessions.

🔧 How They Were Solved

- Broke logic into small, modular functions.
- Implemented blocks with fallback UI.
- Used for persistent state.
- Leveraged AI for rapid troubleshooting.

Key Lessons Learned

01

Modular Code

Clean JavaScript structure is crucial for scalability.

02

Real Data

Handling external APIs requires robust error checking.

03

State Mgmt

Managing UI state is the core of modern web apps.

Future Roadmap

Backend

Node.js for real
contact forms.



Auth

User dashboard &
login.



UI/UX

Advanced sorting
animations.



Scale

Convert to React.js
App.



Project Outcomes

- ✓ Fully polished, production-ready portfolio.
- ✓ Demonstrated advanced problem-solving.
- ✓ Built a professional asset for career growth.

