

SWE363 – Assignment 4 Presentation

Abdulmajeed Aljuhaymi

December 2025



Project Overview

What I Built & Why



Personal Portfolio

A professional showcase of my projects, skills, and journey as a Software Engineering student.



Accessibility First

Built with WCAG guidelines — semantic HTML, ARIA labels, keyboard navigation.



Zero Dependencies

Pure HTML, CSS, and JavaScript — no frameworks, deployable anywhere.



AI-Enhanced

Leveraged AI tools for code generation, accessibility tips, and improvement.



Personal Motivation

- Create a professional presence to showcase my work to recruiters and collaborators
- Apply web development skills learned throughout SWE363 to a real-world project
- Explore modern accessibility practices and inclusive design
- Build something I can continue to improve and use beyond this course
- Experiment with AI-assisted development workflow

✨ Key Features



Dynamic Theming

Light/Dark mode with localStorage persistence.



Project Gallery

Category filters, live search, empty states.



GitHub Integration

Live API feed with sorting and offline fallbacks.



Learning Spotlight

API-driven quotes with graceful error handling.



AI Accessibility Coach

ChatGPT-authored tips with attribution.



Contact Form

Inline validation, ARIA live regions, mailto.



Technical Architecture

HTML5

Semantic Markup



CSS3

Custom Properties



Vanilla JS

ES2020+ Modules



APIs

GitHub + Quotable

CSS Grid & Flexbox

CSS Custom Properties

LocalStorage

Fetch API

ARIA Attributes

prefers-reduced-motion

Lazy Loading



AI Integration

How I Used AI Throughout Development

ChatGPT (Codex CLI)

Primary code assistant

ChatGPT (Web)

Content & copy drafting

- **Code Generation:** Scaffolding, API integration, animation utilities
- **Debugging:** Resolving CORS issues, state persistence bugs
- **Accessibility:** ARIA patterns, live regions, focus management
- **Documentation:** Technical docs and usage reports

All AI outputs were reviewed, customized, and documented in `docs/ai-usage-report.md`



Challenges & Solutions

Challenge: GitHub API Rate Limits

Free tier limits requests, causing errors during testing.

Solution: Smart Caching

LocalStorage caching, optional PAT support, and curated fallback repos.

Challenge: State Persistence

User preferences resetting on page reload.

Solution: localStorage Architecture

Consistent key-value storage for theme, filters, and preferences.



Lessons Learned

- **CSS Custom Properties** enable powerful theme systems without duplicating selectors
- **Progressive Enhancement** ensures functionality even with JavaScript disabled
- **Graceful Degradation** for API failures keeps the user experience smooth
- **Accessibility** isn't an afterthought — it should be built in from day one
- **AI is a Tool** — it accelerates development but requires human review
- **Documentation** matters — both for others and for your future self



Outcomes & Future Plans

Rubric Area	Achievement
Functionality	All features implemented and working
Code Quality	Clean, modular, well-documented code
Accessibility	ARIA labels, keyboard nav, reduced-motion
AI Integration	Documented, responsible, effective use



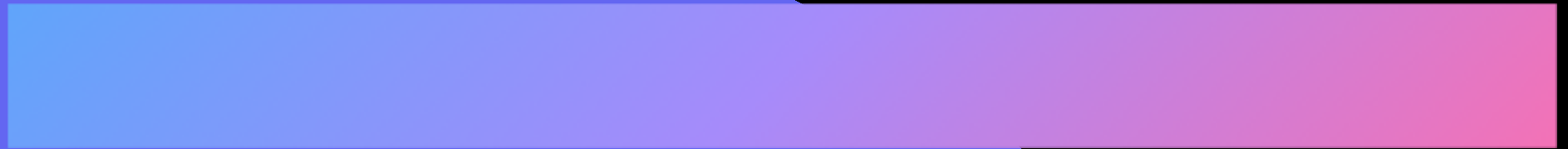
v1.1: PWA features & offline support



v1.2: Multi-language support



v2.0: Backend & real contact form



Questions?

GitHub

Email

Live Demo: Open `index.html` in your browser