ASSIGNMENT 4 FINAL PROJECT

Personal Portfolio Web Application

A professional, responsive showcase of development skills featuring API integrations, dynamic theming, and modern UI architecture.

PRESENTED BY
Aleen AlQarni

DATE
December 5, 2025

OVERVIEW

Introduction

This project represents the culmination of Assignment 4, synthesizing design principles, interactivity, and API integration into a single professional identity platform.

Personal Portfolio Web App

My Objective

To design and deploy a responsive portfolio that effectively highlights technical skills, hosts project demos, and provides clear contact avenues for potential employers.

Why I Built It

To move beyond static templates and build a custom solution that demonstrates mastery of DOM manipulation, fetch APIs, and modern CSS architecture.

Problems Solved

Eliminates the need for sending multiple links by centralizing work into one URL and offering an organized, filterable view of all assignments.

Motivation

Driven by the desire to establish a strong professional identity, showcase growth, and create a sandbox for experimenting with new UI/UX ideas.

PORTFOLIO ARCHITECTURE

Project Overview

A modern web application built with a focus on clean aesthetics, responsive behavior, and robust data integration.

Dynamic User Experience

Features a fluid interface with smooth transitions, interactive states, and a clean, responsive layout that adapts seamlessly to any device size.

Data-Driven Components

Project data and content are separated from the view layer, allowing for dynamic rendering and easy updates through structured JSON objects.

API Integrations

Live connection to external services including the GitHub API for repository data and third-party APIs for engaging visual content.

Professional Design

Designed with accessibility and aesthetics in mind, utilizing modern CSS variables for theming and a consistent visual hierarchy.

100%	2+	ES6+
RESPONSIVE	LIVE APIS	MODERN JS

CORE CAPABILITIES

Key Features

Smart Greeting

Personalized welcome message that adapts based on the user's local time of day (Morning, Afternoon, Evening).

Session Timer

Real-time counter tracking visit duration, demonstrating `setInterval` logic and dynamic DOM updates.

Theme Engine

Interactive color randomizer that instantly updates CSS variables across the entire application state.

Project Filtering

Advanced sorting and filtering logic allowing users to organize projects by date, relevance, or tech stack.

GitHub Sync

Live REST API integration fetching repository stats, descriptions, and languages directly from source.

Random Cat API

A delightful micro-interaction fetching random cat images to demonstrate async/await fetch patterns.

Form Logic

Contact form with real-time input validation, error handling states, and success feedback messages.

Mobile Drawer

Responsive navigation that transforms into a touch-friendly side drawer on smaller viewports.

SYSTEM DESIGN

Technical Architecture

Core Stack

HTML5
Semantic Markup

CSS3
Grid & Flexbox

JavaScript
ES6+ Modular Logic

UI Layer

Responsive Layout

Utilizes CSS Grid for main architectural structure and Flexbox for component alignment, ensuring seamless adaptability across devices.

Style Layer

Dynamic Theming

Powered by CSS Variables (Custom Properties), allowing for instant global color updates via JavaScript manipulation without page reloads.

Storage Layer

LocalStorage

Implements browser-based persistence to save user preferences and session data, maintaining state across page refreshes.

Data Layer

REST Integrations

Asynchronous fetch() calls to external endpoints (GitHub, Cat API) with robust error handling and loading states.

LIVE CONNECTIONS

API Integrations

GitHub API

Repository Synchronization

- Fetches latest repo data dynamically
- Displays name, description, & languages
- Formats 'Last Updated' timestamps

fetchGithub.js

```
const getRepos = async () => {
  const res = await fetch('api.github.com/...');
  const data = await res.json();
  displayRepos(data);
};
```

The Cat API

Visual Micro-interactions

- Random photo generation on button click
- Handles loading states gracefully
- Robust error handling for failed requests

Preview Component

FETCHING CAT...

PROBLEM SOLVING

Challenges & Solutions

Mobile Alignment

Navigation elements were overlapping on small screens and ignoring safe-area insets on modern phones.

Flexbox Utilities

Implemented comprehensive Flexbox controls and `env(safe-area-inset-bottom)` alongside rigorous breakpoint testing.

API Rate Limits

GitHub and Cat API requests would occasionally fail or hit rate limits during rapid testing, breaking the UI.

Graceful Fallbacks

Added robust try/catch blocks, user-friendly error messages, and basic caching to minimize unnecessary fetch calls.

Complex Sorting

Filtering and sorting simultaneously created conflicting states where the project list would render incorrectly.

Stable Logic

Refactored to use pure functions and a centralized state object, ensuring filters apply to the source data cleanly.

Fluid Layouts

Maintaining consistent visual hierarchy and spacing across tablet and desktop viewports proved difficult.

Grid Utilities

Utilized CSS Grid for macro-layout and fluid typography (`clamp()`) to ensure proportional scaling on all screens.

DEVELOPMENT ASSISTANT

AI Integration

Code Debugging

Identified tricky race conditions in API fetches and suggested cleaner state management patterns for complex filter logic.

UX Enhancements

Proposed micro-animations for loading states and refined color contrast ratios to ensure better accessibility standards.

Documentation

Assisted in structuring README files and generating clear JSDoc comments for modular utility functions.

Feature Ideation

Suggested edge-case scenarios for the contact form validation and creative ideas for the 404 error page.

Ethical Use

"AI was utilized as a powerful support tool to accelerate learning, not as a replacement for understanding."

KEY PRINCIPLES

- Code verification & review
- Understanding logic flow
- Debugging own implementations

Academic Integrity Maintained

LIVE WALKTHROUGH

Demo Features

presentation_mode: active

01

Responsive Nav

- ✓ Desktop Header
- ✓ Mobile Drawer / Hamburger

02

Personalization

- ✓ Time-based Greeting
- ✓ Color Theme Switcher

03

Project Logic

- ✓ Category Filtering
- ✓ Date/Name Sorting

04

GitHub Data

- ✓ Fetch Repositories
- ✓ Live Update Stats

05

Async Actions

- ✓ Random Photo Fetch
- ✓ Loading Spinner State

06

Contact Form

- ✓ Input Validation
- ✓ Success Feedback

Roadmap

Future Improvements

Backend Integration

Implementing a Node.js & Express environment to handle real message delivery and database storage using MongoDB.

Priority **HIGH**

Advanced UX

Enhancing interactivity with scroll-triggered animations (AOS), smooth page transitions, and skeleton loading states.

Priority **MED**

Content Expansion

Adding deep-dive case studies for flagship projects and a technical blog section to showcase writing skills.

Priority **MED**

Accessibility

Achieving WCAG 2.1 AA compliance through comprehensive ARIA labels, improved contrast ratios, and keyboard navigation.

Priority **HIGH**

INFRASTRUCTURE • EXPERIENCE CONTENT INCLUSIVITY

PROJECT COMPLETE

Requirements Met

Fulfilled all coursework criteria including responsive design, API integration, and interactive DOM manipulation.

Skills Gained

Deepened practical knowledge of RESTful APIs, asynchronous JavaScript, and modern CSS layout techniques.

Future Ready

Established a scalable, modular codebase that is ready for backend integration and professional deployment.

Thank You

AA
Aleen AlQarni
Frontend Developer