Rayan's Quest

Interactive Portfolio Web Application

SWE363 - Final Project Presentation

Rayan Alamri | 202247900

December 2025

Introduction

Project Overview

A **dual-experience portfolio** combining traditional web development with an innovative RPG game layer.

1. Traditional Portfolio

Standard sections, responsive design, API integrations

2. RPG Game Mode

Unlock portfolio content through interactive gameplay

Objectives

- ⚔ Build a complete, professional portfolio website
- ⚔ Demonstrate HTML, CSS, and JavaScript proficiency
- ⚔ Add creative interactivity through gamification
- ⚔ Integrate external APIs and persist user preferences

Introduction

Personal Motivation

Why This Approach?

- Wanted to showcase both **practical skills** and **creativity**
- Traditional portfolios demonstrate competence
- The game layer demonstrates innovation and ambition
- Combines everything learned in SWE363

Target Audience

Recruiters

Seeking technical skills

Developers

Appreciating creative projects

Technical Demo

Portfolio Features

Core Sections

Section	Features
About Me	Bio, education, interests
Projects	Expandable cards, filters
Skills	Categorized technical skills
Contact	Form validation, feedback

Enhancements

Feature	Implementation
Theme Toggle	Dark/Light mode (localStorage)
Advice API	Live quotes (Advice Slip API)
GitHub Feed	Latest repos (GitHub API)
Responsive	Mobile, tablet, desktop

Technical Demo

Key Web Features

Responsive Design

- Mobile-first CSS
- Breakpoints at 768px/480px
- Flexbox & Grid layouts

Accessibility

- Semantic HTML5
- ARIA labels & live regions
- prefers-reduced-motion

State Persistence

- Theme saved to localStorage
- Visitor name remembered
- Login state toggles

Technical Demo

API Integrations

Advice Slip API

```
fetch('https://api.adviceslip.com/advice')
  .then(res => res.json())
  .then(data => displayAdvice(data));
```

- ⚔️ Loading / Success / Error states
- ⚔️ Manual refresh button
- ⚔️ Cache-busting with timestamps

GitHub API

Fetches latest public repositories

Displays repo name, description, language

Live refresh capability

Technical Demo

Innovative Feature: RPG Game Mode

Transform portfolio browsing into an **interactive adventure**

Game Features

- 2D explorable village map with NPCs
- 4 quests that unlock portfolio sections
- Combat system with XP progression
- Mini-games and collectibles

Why It Works

- Engages visitors longer than static pages
- Demonstrates advanced JavaScript skills
- Memorable and shareable experience

Technical Demo

Project Architecture

File Structure

```
assignment-4/
├── index.html       # Single page
├── css/
│   ├── style.css    # Portfolio
│   └── game.css     # RPG UI
├── js/
│   ├── script.js    # Portfolio Logic
│   └── game.js      # Game Engine
├── assets/images/   # Screenshots
└── docs/            # Documentation
```

Tech Stack

HTML5 - Semantic markup, Canvas

CSS3 - Custom props, animations

Vanilla JS - ES6 classes, Fetch API

No external libraries or frameworks

Technical Demo

AI Integration

Tools Used

ChatGPT: Portfolio foundation, validation, API patterns

Claude Code: Game engine, quest system, documentation

Area	AI Contribution
Form Validation	Email confirmation, error messaging
API Integration	Fetch patterns, error handling
Game Architecture	State management, collision detection
Documentation	Structure and formatting

My Contributions: Design, Styling, Content, Final Implementation.

Technical Deep Dive

Challenges & Solutions

1. Dual Experience

Problem: Shared content, two UIs.

Solution: Shared content data model, separate presentation layers.

2. Validation UX

Problem: Intrusive errors.

Solution: Blur/input handlers with dynamic ARIA updates.

3. API Errors

Problem: Network failures.

Solution: Robust Loading, Success, Error states with retry.

Technical Deep Dive

Innovative Solutions

Theme System

CSS custom properties.

RPG-themed palette.

Smooth transitions.

Responsive

Adapts mobile to desktop.

Game UI simplifies on small screens.

Touch-friendly.

No Dependencies

Pure Vanilla JS.

No frameworks.

Deep understanding.

Technical Deep Dive

Lessons Learned

Web Dev

- CSS variables simplify theming
- localStorage improves UX
- Semantic HTML matters

API Integration

- Handle all states (load/fail)
- Cache-busting is crucial
- Rate limits exist

Project Mgmt

- Start with core features
- Test incrementally
- Docs save time

Conclusion

Project Outcomes

Achievements

- ✅ Complete, responsive portfolio
- ✅ Working API integrations
- ✅ Accessible design (ARIA)
- ✅ Innovative RPG game layer
- ✅ Professional documentation

Technical Metrics

500+
Lines of JS (Portfolio)

2500+
Lines of JS (Game)

2
External APIs

100%
Vanilla Code

Conclusion

Future Improvements

Portfolio

Backend for contact form

Blog with markdown

Project filtering

Game

Save/Load progress

Sound effects & music

Mobile touch controls

Deployment

CI/CD pipeline

Performance monitoring

Analytics integration

Live Demo

Demonstration Outline

1. Traditional View

- Navigate sections
- Toggle dark/light theme
- Show API feeds
- Form validation

2. RPG Game Mode

- Start the adventure
- Quest & unlock system
- Combat & progression

Thank You

Rayan's Quest

Live Demo: rayanassignment4.netlify.app

GitHub: github.com/Rayan-Alamri/assignment-4

Documentation: /docs folder

Questions?