easy news and updates system

Devon Xavier Dalrymple

Created: 2023-01-27

Last modified: 2023-05-19 17:07:00

Digital Portfolio

software Design journal (all-in-one)

Table of Contents

[1 Table of Figures iv](#_Toc135408576)

[2 Table of Tables v](#_Toc135408577)

[3 Introduction 1](#_Toc135408578)

[3.1 About this Project 1](#_Toc135408579)

[3.2 Replacing the Old 1](#_Toc135408580)

[3.3 About this Document 1](#_Toc135408581)

[3.4 All-in-One Clarification 1](#_Toc135408582)

[3.5 Project Information 1](#_Toc135408583)

[3.5.1 Start Date 1](#_Toc135408584)

[3.5.2 Due Date 1](#_Toc135408585)

[3.5.3 Budget 2](#_Toc135408586)

[3.5.4 Project Requirements Clarification 2](#_Toc135408587)

[3.5.5 Project Risks 2](#_Toc135408588)

[4 Data Dictionary 2](#_Toc135408589)

[5 Competitor-Type Analysis 3](#_Toc135408590)

[5.1 Note 3](#_Toc135408591)

[5.2 Matt Farley’s Portfolio 3](#_Toc135408592)

[5.2.1 Link 3](#_Toc135408593)

[5.2.2 Notes 3](#_Toc135408594)

[5.3 Eina Onting’s Portfolio 4](#_Toc135408595)

[5.3.1 Link 4](#_Toc135408596)

[5.3.2 Notes 4](#_Toc135408597)

[5.4 Fabian Irsara’s Portfolio 4](#_Toc135408598)

[5.4.1 Link 4](#_Toc135408599)

[5.4.2 Notes 4](#_Toc135408600)

[5.5 Diane Laidlaw’s Portfolio 5](#_Toc135408601)

[5.5.1 Link 5](#_Toc135408602)

[5.5.2 Notes 5](#_Toc135408603)

[5.6 Adham Dannaway’s Portfolio 5](#_Toc135408604)

[5.6.1 Link 5](#_Toc135408605)

[5.6.2 Notes 6](#_Toc135408606)

[5.7 Competitors Ranked (Ignoring Experience) 6](#_Toc135408607)

[6 System Requirements 7](#_Toc135408608)

[6.1 Functional 7](#_Toc135408609)

[6.1.1 Front Page with Brief Introduction and Navigation Links 7](#_Toc135408610)

[6.1.2 OAuth Login to Connect to the API 7](#_Toc135408611)

[6.1.3 Use SSL for Production Server 7](#_Toc135408612)

[6.1.4 About Me Page with Personal and Professional 7](#_Toc135408613)

[6.1.5 Resume Page 7](#_Toc135408614)

[6.1.6 Previous Projects Page 7](#_Toc135408615)

[6.1.7 Integrated Side Project Showcase Page 7](#_Toc135408616)

[6.1.8 References (May Require User Login) Page 7](#_Toc135408617)

[6.1.9 Skills Page 7](#_Toc135408618)

[6.1.10 Certifications Page 7](#_Toc135408619)

[6.1.11 Other Credentials Page 7](#_Toc135408620)

[6.1.12 Portfolio Updates Page 7](#_Toc135408621)

[6.2 Nonfunctional 8](#_Toc135408622)

[6.2.1 WCAG 2.1 AA Compliance 8](#_Toc135408623)

[6.2.2 Custom Logo 8](#_Toc135408624)

[6.2.3 Custom Favicon 8](#_Toc135408625)

[6.2.4 Professional Code Documentation 8](#_Toc135408626)

[7 Nice-to-Haves 8](#_Toc135408627)

[7.1 Labyrinth of Devon (v3.0) 8](#_Toc135408628)

[7.1.1 Why 8](#_Toc135408629)

[7.1.2 Self-Applied Constraints 9](#_Toc135408630)

[7.1.3 Prerequisites 9](#_Toc135408631)

[7.2 Day-Based Messages 9](#_Toc135408632)

[7.3 Card/Table Listing Switch 10](#_Toc135408633)

[7.4 Professor Testimonials 10](#_Toc135408634)

[7.5 Caravan Card Game 10](#_Toc135408635)

[7.5.1 What? 10](#_Toc135408636)

[7.5.2 Why? 10](#_Toc135408637)

[7.5.3 Rules 10](#_Toc135408638)

[7.5.4 Computer Player Structuring 10](#_Toc135408639)

[7.6 Extension Transpiler 11](#_Toc135408640)

[7.6.1 What? 11](#_Toc135408641)

[7.6.2 Why? 11](#_Toc135408642)

[8 Frontend Design 11](#_Toc135408643)

[8.1 Built With 11](#_Toc135408644)

[8.2 Color Theme 12](#_Toc135408645)

[8.3 Font Ideas 12](#_Toc135408646)

[8.3.1 First Font-Type Set 12](#_Toc135408647)

[8.3.2 Second Font-Type Set 12](#_Toc135408648)

[8.3.3 Third Font-Type Set 13](#_Toc135408649)

[9 Backend Design 13](#_Toc135408650)

[9.1 Built With 13](#_Toc135408651)

[10 Database Design 13](#_Toc135408652)

[10.1 Built With 13](#_Toc135408653)

[11 Deployment Design 13](#_Toc135408654)

[12 Test Plan 13](#_Toc135408655)

[13 Secrets Configuration 13](#_Toc135408656)

[13.1 About 13](#_Toc135408657)

[13.2 Identity Server Certificate Generation 14](#_Toc135408658)

[14 Running the application 14](#_Toc135408659)

[15 Lessons Learned 14](#_Toc135408660)

[15.1 JSON Serialization and Camel Casing 14](#_Toc135408661)

[16 Other Links/Guides for Reference 14](#_Toc135408662)

# Table of Figures

[Figure 1: Color Theme 12](#_Toc135408663)

# Table of Tables

[Table 1: Competitor Analysis Table 6](#_Toc135408664)

# Introduction

## About this Project

This project is to build a digital portfolio, befitting a back-end leaning entry-to-junior-level Web developer. Particularly this should show of some ASP.NET and Angular skills as these are the employment opportunities, I am most interested in. This has become more necessary now that graduation is coming and there is a lack of job offers.

## Replacing the Old

This system will replace a Google Web site that I made back in 2017. The previous site was made before I learned to program and is now embarrassing to have it still listed as my digital portfolio for a Web developer in 2023. The old is linked currently on LinkedIn and is potentially detrimental to how hirable I appear.

## About this Document

This document is a design journal, which means it includes various things that I have jotted down or diagrammed in order to design and implement the project. It will be made freely available on GitHub as a way to show off my design and technical writing skills when the digital portfolio goes online.

Since this is a personal project needing completion in a few months, with relatively little time available to it, many things such as a detailed set of CRC cards and class ADTs may not be included. It is not that I lack the knowledge to do them, but their gain in this small project and limited time availability make them less desirable.

## All-in-One Clarification

Normally, I would have requirements, competitive analysis, test planning, etc., in different documents with more details included. Instead, all of this will be lumped together with less detail into this design journal. This will make it easier to jump to different information, reduce duplication of information, and make the project simpler.

## Project Information

### Start Date

2023-01-27

### Due Date

2023-04-05

### Budget

The budget for this project is $150 for the first two years that this project is deployed, ignoring the cost of labor to develop, test, and deploy.

### Project Requirements Clarification

The planned budget for this project is $150 and it is meant to be used for hosting fees (if applicable), purchasing a domain name, acquiring SSL, buying domain protection, and potentially other items, such as an email with the domain name and cloud storage.

The due date is approximately one month from the date of finals completion. It is set this way to allow time in April to market myself towards prospecting employers. It is also this way since time in April will need to be focused on completing finals, especially the capstone project.

### Project Risks

#### Classwork is Overwhelming

It is quite possible with this being my final semester that classwork will become very difficult and may distract me from personal project development. In response, the end date has been set a month from graduation and a time block for 45 minutes from Monday to Friday has been allocated each week to work on this project and others.

If time is left after classwork is completed on schedule for a week, 2/3 of the backup time blocks will be allocated towards personal project development.

#### High Cost of Deployment

The budget of $150 may be too much for an extensive deployment. In response, the budget may be lowered by foregoing on niceties such as a domain email, using free SSL providers, and self-hosting the deployment.

#### Data Loss

It is possible that data loss could hinder the project’s completion. In response, each time work concludes on this project for the day, all changes will be pushed up to the develop branch on GitHub.

# Data Dictionary

1. Caravan Route – Caravan has three routes with caravans that the two players must compete for control over (get the buy), both players can build their side of the route to try and reach between 21 and 26
2. RNG – Random Generation, relating usually to luck-based features of a game such as chance encounters or randomized placement of items and enemies.

# Competitor-Type Analysis

## Note

These competitors are seen as highly recommended by this source: <https://arc.dev/developer-blog/web-developer-portfolio/>

It is not expected that this portfolio will outdo theirs. These developers have been in the industry for a longer time and have more highlights in their metaphorical resume that can be used. However, by assessing what was done well, this can be done better than it otherwise might of.

## Matt Farley’s Portfolio

### Link

<https://mattfarley.ca/>

### Notes

1. Navbar is very light and has no displayed outline.
2. Navbar has a button titled ‘Say Hello’ which redirects to the contact page.
3. Contact page does not show information but has a form to insert name, email, and message.
4. Mentorship page has a humorous title and subtitle, “I haven’t met you, and this is crazy…But if you need a mentor, then email me maybe.
5. Mentorship page shows people he has helped as a simple map image with profile pictures placed within the map pins.
6. Design is soft and friendly, with purple as the theme color.
7. Outlines for contents are hidden and outlines for some are structured to produce a floating effect of the main content.
   1. This is enhanced by the dividing colors for the page background altering with the content in the middle.
8. A beautiful list tile view is used on his front page to divide ‘Designer’, ‘Frontend Developer’, and ‘Mentor’ into separate categories which details elements like languages known, experience, tools used, etc.
9. Content is very intentionally displayed where the background changes between white and the theme color in certain areas to help give the appearance of the content floating on the page.
10. Funnily enough, his page happens to be using BulmaCSS as well.
11. Copyright on the footer writes out as ‘©twentytwentythree.’

## Eina Onting’s Portfolio

### Link

<https://www.eina.ca/>

### Notes

1. Basic contact information is visible as soon as the front page loads when rendered on desktop
2. ‘too bright?’ and ‘too dark?’ are displayed on the navbar as a way to toggle between light and dark mode
3. Love the dark mode color that features a creamy peachy pink outline and soft brown combo
4. The dark mode text is less white and more of a light brown which does wonders for the page and its creamy feel
5. Theme is very friendly, soft, upbeat, and also seemingly, tasty?
6. Light mode features a nice blue for the headings but I personally dislike how much it reminds me of news websites
7. Navbar items are lowercase and underlined with that same peachy pink
8. Link to a CSS animated drawing shown on front page that is easily seen due to using that same underline used for the navbar

## Fabian Irsara’s Portfolio

### Link

<https://fabianirsara.com/>

### Notes

1. Outline around the page is large and nice, however, the bottom outline, being as thick as the rest of the page’s outline, seems to make the page look a bit worse.
2. The desktop view uses a hamburger menu, which was placed to the right and is animated
3. Desktop view opens with a large circular profile picture and the heading, “Cheers, I am Fabian”
4. Emojis are placed around the page and give it a very friendly look
5. Important details listed on the front page are given the theme color
6. A day of the week message is displayed towards the bottom of the introduction and changes the adjective for the day every few seconds using an animation that quickly types out the adjective, one letter at a time
7. There are hidden hyperlinks that change color and write themselves out again on hover, but the letter writing out animation seems to make them appear as if they are a bit buggy.
   1. It may simply be that it is changing to fast.
   2. It may also be better if the letters in the animation where simply invisible as the animation played and then rendered in, which would prevent the introduction paragraphs from jumping on the page as much
8. Pixels page look great and has background images for each list item tab show on hover, while being performant

## Diane Laidlaw’s Portfolio

### Link

<https://thedesigncreative.co.uk/>

### Notes

1. Unlike the earlier examples, this page makes use of frequent light gray outlines that divide dark mode content
2. Theme makes use of near black with the main colors being shades of brown
3. Page seems very professional and formal
4. The logo for the page gives off the feeling that Diane is some sort of valuable designer product
5. Columns are given a list tile sort of look and page content uses a container centered on the page
6. Makes use of an oval for her picture
7. Projects page looks amazing showing the same or similar web pages for different projects grouped up as cards that have a tablet, phone, laptop, and desktop render all present and consistent
8. Gives off the appearance of having a lot of experience and desiring a notably higher developer salary

## Adham Dannaway’s Portfolio

### Link

<https://www.adhamdannaway.com/>

### Notes

1. Portfolio page has 3 columns if small cards with images and summary text that showcase different designs he has done for various types of projects
2. Pages seem really slow and the animation between them seems to increase the sluggish feel
3. Theme is mostly black and white with some grayscale elements. Images add color and pop out from the page
4. Front page has a transition between his web development and his design areas of expertise with a hero that has an inverted slider that reacts to the cursors position
5. Pages seem to drain more memory than it should and seem to not be performant

## Competitors Ranked (Ignoring Experience)

To not give an unfair advantage, experience listed is ignored, only the site structure matters.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Person | Color/Font Theme | Call-to-Actions | Features | General Viewing Experience |
| Farley | 5/5 ★ | 5/5 ★ | 4/5 ★ | 4/5 ★ |
| Onting | 5/5 ★ | 4/5 ★ | 3/5 ★ | 4/5 ★ |
| Irsara | 3/5 ★ | 3/5 ★ | 4/5 ★ | ?/5 ★ |
| Laidlaw | 5/5 ★ | 5/5 ★ | 5/5 ★ | 5/5 ★ |
| Dannaway | 3/5 ★ | 2/5 ★ | 3/5 ★ | 2/5 ★ |

Table : Competitor Analysis Table

# System Requirements

## Functional

### Front Page with Brief Introduction and Navigation Links

### OAuth Login to Connect to the API

### Use SSL for Production Server

### About Me Page with Personal and Professional

### Resume Page

### Previous Projects Page

### Integrated Side Project Showcase Page

### References (May Require User Login) Page

### Skills Page

#### Early Notes

1. A doughnut chart distribution of frameworks to projects and another for languages to projects could look really nice at the top of the page

#### Formal

### Certifications Page

### Other Credentials Page

### Portfolio Updates Page

#### What?

This would be a page for me to post recent updates about new skills, changes to the site, talk about new technologies, etc. It would function sort of like a short summary blog in a way. Where blogs may be given pages, this would be meant for content from one to ten paragraphs in length with maybe attached images/files.

This is my planned alternative as I am not currently interested in maintaining a blog of any scale, these updates could be days to months apart.

#### Formal

## Nonfunctional

### WCAG 2.1 AA Compliance

### Custom Logo

### Custom Favicon

### Professional Code Documentation

# Nice-to-Haves

## Labyrinth of Devon (v3.0)

### Why

The Labyrinth of Devon was my submission for my first and second semesters of college. It is based around the World of Zuul project from the BlueJ books. It is a simple text based game that in the second class when from pure command line to featuring some graphical elements as the text-based game was played.

To make this digital portfolio more interesting and as a semi-separate personal project, I want to remake a version of Labyrinth of Devon. The idea being that I could see how much that I have changed since my freshman year.

The previous version was built using Java and AWT/Swing. It featured a four-floor text-based dungeon experience. It had randomization of damage that took the min and max damage values from the weapon and character strength and reduced it by any defense modifiers. The player and enemies had a chance to dodge. The player could gain experience and level up. There were weapons, armor, and other items that could be equipped. Each turn allowed one command to be performed. If the enemies were not in the same room of the player on that floor of the dungeon, they could move to an adjacent room at random and surprise attack the player in the same round. The dungeon had three floors of enemies, items, and random room generation. On the final floor was what was meant to be a mini boss out of three choices. My theme to outdo the other students taking the course between two classes was to add randomization everywhere: four layouts, with a few rooms randomly selected for the floor’s list, and a random allotment of monsters and items. This gave my game, something that only one other student could boast of at the time, different experiences each time it was played.

However, the code was unoptimized, had boring lore elements, lacked depth, had a poor class system, and used heavy copy and paste. If I wanted 20 health potions for a floor, I had 20 lines of code do the adding of health potions. The GUI in the second version was mediocre and had no images for what occurred in text. The class system was meant to be Warrior, Rogue, Mage, and Peasant. The peasant class worked as expected since they were meant to be a challenge to do since they lacked bonuses. The mage was useless but stronger than the peasant since the magic system was never implemented. The rogue had no special abilities but was stronger than the mage and had better dodge chances. This led to an unintentional difficulty system based on class chosen (which due to time constraints, I left in because it was a “feature”).

How have I changed over the years? What could be done so much better, even with the time tacked on by making it Web friendly? This is a fun little experiment I want to do.

### Self-Applied Constraints

1. 24-hour limit, timed.
   1. If I take a short break for less than 30 minutes, the time will be left to run since I tend to continue thinking about a project on break
   2. If I stop for the day or long period, the time will stop
2. Six hours will be dedicated to any initial set up of third-party libraries along with designing the game
3. 18 hours will partially be used for design but, mostly, it will be dedicated to implementing the game on the site
4. Third-party libraries are allowed but cannot be overused
5. It must stay true to the mostly text-based nature of the original two versions
6. It must be on its project page with a description of what was used and what it is about
7. It should feature at least five floors
8. It should have more lore elements
9. It should have a working-class system
10. It should have the RNG features that the original two versions had
11. It should have magic elements

### Prerequisites

This cannot be started until previous projects have been listed and detailed on other pages and 1/3 of the requirements for the portfolio are met.

## Day-Based Messages

This was used with one of the examples and provides a dynamic and positive statement about the day. If there is a good place for it that is not doing well with being empty, this could be implemented.

## Card/Table Listing Switch

This may only fit certain pages, but allowing the view of data to be changed could help to give off a vibe of being more knowledgeable for doing something that is relatively simple

## Professor Testimonials

NOTE: Perhaps include a letter from each if they are willing

## Caravan Card Game

### What?

This is a card game that may have originated with the game Fallout: New Vegas. The main premise is simple, sell 2 caravan routes by having a higher bid than the opponent but no more than 26 and no less than 21 points for that route.

### Why?

This is a very enjoyable card game that does not have as many implementations as others online. This makes it more unique compared to other digital portfolios. It requires some form of AI player and has more complexity than a simple game like Blackjack.

This card game would show off skills related to online game mechanics, simple conditional ‘AI’, some creativity, etc.

### Rules

<https://www.reddit.com/r/cardgames/comments/97c7g2/caravan_card_game_in_reallife_detailed_rules/>

### Computer Player Structuring

Below items will be organized by difficulty and may require tweaking to function as intended

#### Difficulties

1. Easy – 33% chance to make a bad move (if possible)
2. Normal – 20% chance to make a bad move (if possible)
3. Hard – 10% chance to make a bad move (if possible)

#### Decks Used

Each additional card will have a random suit.

1. Easy – standard 54 card deck with Jokers
2. Normal – standard 54 card deck and 1 extra Joker, King, Queen, Jack, and Ace
3. Hard – standard 54 card deck and 2-3 extra Jokers, Kings, Queens, Jacks, and Aces

## Extension Transpiler

### What?

The extension transpiler was a large part of my senior capstone. It used a custom script and could convert the script into Dart source code. The use case was for school messenger apps (School Messenger Application project) to support user-created extensions.

Resulting source code is meant to follow the Triple Segmented State package pattern. This could mean from one to three files depending on the type of extension and its requirements with tracking state.

### Why?

This extension transpiler was left close to being finished after a large amount of research, design, and implementation time was assigned to it. This would make it so that work did not get thrown away with the completion of the senior capstone class. It would also showcase an area of development skills related to transpiler design and may serve as quite the showcase for the digital portfolio.

# Frontend Design

## Built With

1. TypeScript
2. Angular
3. PrimeNG
4. FontAwesome
5. Bulma
6. SCSS files

## Color Theme

Chart, treemap chart

Description automatically generated

Figure : Color Theme

This was selected because it was colorful and went well with the predefined theme that Bulma uses for its components. In fact, Color 2 is the primary color used for components in Bulma. The other colors were selected based on the second color.

This color theme seems to give way to a colorful and friendly design. I’ve decided to use a buttery white for the background color



Figure : Background Color #F1EBDA

The current choice for the text color is a sort of velvety red, which stands out from the pink navbar and background nicely



Figure : Text Color #662222

## Font Ideas

### First Font-Type Set

#### Heading

Archivo Narrow

#### Subheading

Fondamento

#### Body

Cambay

### Second Font-Type Set – Likely Choice

#### Heading

Archivo Black

#### Subheading

Hind Guntur

#### Body

Imprima

### Third Font-Type Set

#### Heading

PT Sans

#### Subheading

Changa

#### Body

Molengo

## Page Animations

Page animations will occur based on the starting and end menu button location on the page (in relation to desktop view)

If the page to route to is already selected or under the same dropdown, then play no animation.

If the page to route to is left of the current menu item, then animate the new page coming in from the left and pushing the old page out to the right.

If the page to route to is right of the current menu item, then animate the new page coming in from the right and pushing the old page out to the left.

# Backend Design

## Built With

1. ASP.NET 6
2. Duende IdentityServer 6E
3. Bootstrap – May be changed to Bulma later on

## Routes

### Frontend

1. /

### Swagger

1. /api/
2. /api/swagger/

### API

Versioning will be used for most paths on the API

1. /api/Authentication/\*
2. /api/UserInfo/\*
3. /api/v\*/\*

# Database Design

## Built With

1. PostgreSQL
2. PgAdmin – Developer Side
3. Fly Postgres – Production Side

# Deployment Design

# Test Plan

# Secrets Configuration

## About

This section is for all of secrets needed in environment or secrets files for the application to downloaded onto a machine and run without issue. Only the key will be supplied for these.

## Identity Server Certificate Generation

<https://stackoverflow.com/questions/35880187/how-would-i-generate-the-identity-server-signing-certificate>

# Running the application

dotnet publish -c Release -o published

dotnet published/Portfolio.dll

New-SelfSignedCertificate -Type Custom -Subject "CN=PortfolioId" -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.3") -KeyUsage DigitalSignature -KeyAlgorithm RSA -KeyLength 2048 -CertStoreLocation "Cert:\LocalMachine\My"

# Lessons Learned

## JSON Serialization and Camel Casing

The default project for Angular w/ ASP.NET Core 6 is not configured correctly, at least from what was seen. The deserialization of JSON is case sensitive when done by the http client for Angular. The pascal case used by .NET Core caused issues with deserialization, this caused the table on the testing ‘Fetch Data’ page to not have data in its rows (although the rows did get created).

Additionally, Angular/TypeScript does not notify when an array object is not actually holding the desired type of data. There are no errors thrown when this occurs and functionality just seemingly breaks quietly.

# Other Links/Guides for Reference