Personal Project

Research - How to make a portfolio scalable?

Juliën Verheijen
Fontys Hogeschool ICT
Advanced Media Design

Table of Contents

Context

This document describes all the research that answers the following sub-question " How to make a portfolio scalable?". This research provides more clarity about how the portfolio can be expanded, whether a framework is the best solution and which framework is best used.

Research Method

Literature Study

Quickly and easily search for useful information from reliable sources. This allows choices to be made and substantiated with facts.

Expert Interview

An expert can put you on track when you enter a new domain or field of expertise. The expert can point out sources, give you a sense of direction or point to common pitfalls.

(CMD Methods et al., 2015)

Result

The following questions must be answered first, to answer the research question ("How to make a portfolio scalable?").

- *Am I going to use a framework?*
- *If yes? Which framework should I use?*
- *Is a framework the best way to make it scalable for this project?*

Am I going to use a framework?
The article by Pekarsky (2020) explains the advantages and disadvantages of using a framework.

Advantages:

1. **Maintainability:** Breaking up your app into reusable, standalone components makes it easier to make quick changes that don't impact the rest of the application.

2. **Separation of concerns:** Modern framework design encourages a maintainable, modular architecture and allows your front-end developers to focus on what they do best: taking data and displaying it to users in an intuitive and efficient way.

3. **Speed:** Boilerplate code aimed at addressing common problems makes it easier for you to get your app up and running; component-based design makes it quicker to develop.

4. **Collaboration:** Since frameworks often follow similar design patterns, it's easier for developers who are new to your codebase to develop and maintain your app.

5. **Community:** Popular frameworks have a community of people around them with dedicated tutorials, forums, meetups, and generally supportive developers you can seek help from.

Disadvantages:

1. **Abstracted, overhead code:** Until you're thoroughly familiar with it, framework code is a black box. It can be hard to discern how much of the code is helpful to your application and frustrating to fix bugs resulting from code you're not familiar with.

2. **Learning curve:** Learning to use a framework effectively takes time. To be productive, you need to understand the syntax, tooling, *and* philosophy behind how a framework functions. For

projects where speed is essential, learning a brand new technology might not be the best use of your time.

3. **Overkill for smaller projects:** If you're looking to deploy a static site or a site where every component is unique, you might not need the power and overhead of a full-fledged framework. It might still be helpful to implement a minimal framework or even library—we'll discuss these in the next section.

4. **Setup:** Setting up and customizing a framework to your specific use case takes time. If speed is essential, go with what you know, or what your development team is comfortable with.

5. **Strong opinions:** An opinionated framework may feel constricting, and its design principles may clash with yours. Make sure you research the specific framework you're implementing. If you prefer to build from scratch, go with your own solution.

6. **Ecosystem evolution:** The JavaScript framework ecosystem is famously volatile. The hottest framework of today may not be popular next year, and with this shift, developers and support may be hard to find.

   (Pekarsky, 2020)

So, there are a lot of advantages and disadvantages. Many of the same components will be used to develop the portfolio. These will also be continuously supplemented with new content. It will only take some time at the start to learn to understand the new code. According to the article, there is a lot of information about this online. That's why it's better to start with this during school and gain experience in using frameworks so that I can start working more quickly after studying.

Which framework should I use?
An article by Karczewski (2023) explains the most popular frameworks of this year. I compared the advantages and disadvantages of the 5 most popular ones and determined which one would best suit this project.

React

+ Pro's

- Frequent updating;
- Virtual DOM enables high-speed operations in the document;
- You can combine it with many other JS libraries;
- Allows writing components without classes;
- Easy migration between different versions;
- Reusable code components;
- Good for beginners.

- Con's

- Due to fast updates and many frameworks that complement React, it's challenging to browse proper documentation;
- The complexity of learning JSX syntax.

The most popular framework for a reason. As a result, the most information can be found about it. Good for beginners but also complex to learn.

Vue

+ Pro's

- Tiny and fast;
- Friendly for beginners;
- Detailed documentation;
- Simple syntax;
- Two-way data binding.

- Con's

- Lack of plugins and language barriers (lots of Chinese content);
- Relatively new and developed by private individuals;
- Limited community of developers (but growing);
- Limited applicability for more extensive projects;
- No solid company behind it.

I used Vue before in a previous project. But certainly, don't know everything about this yet. Still, I would prefer to get acquainted with a new framework.

Svelte

+ Pro's

- One of the fastest front-end frameworks with quick responsiveness;
- Component-based architecture with minimal code;
- Lightweight, simple, uses existing JS libraries;
- No virtual DOM;
- SEO optimized.

- Con's

- Immature community and limited ecosystem;
- Lack of supporting materials, limited tooling;
- Certain doubts about scalability and coding nuances.

It is beneficial for front-end beginners as it uses simple syntax, does not require DOM manipulations, and ensures that the browser loads as little as possible, which speeds up your web product.

Angular

+ Pro's

- Component-based architecture;
- Two-way data binding;

- Directives and dependency injection functions;
- Highly testable / reusable / manageable applications;
- Improved server performance;
- Strong community, good training materials, etc.;
- Supported by Google.

- Con's

- Difficult for beginners and overwhelming for smaller teams;
- Limited SEO capabilities;
- Bloated code and large.

Angular is considered one of the most complex front-end frameworks. It can be challenging to learn on your own from scratch.

JQuery

+ Pro's

- Beginner-friendly;
- Convenient to work with;
- Compatible with all major web browsers;
- A large selection of plugins;
- A strong and well-developed community;
- Provides multiple tools for DOM manipulations;
- SEO optimized.

- Con's

- Huge library to import;
- Apps built with it can be a bit slower than native apps;
- DOM APIs are considered outdated;
- Lacks a data layer.

JQuery is the oldest on this list, therefore it is widely available and easy to work with.

There is a wide choice for choosing a framework, the main requirement is that it is not too complex and that there is enough information about it. Because learning this framework takes extra time, I chose the most popular option, which is the most valuable to put on my resume. That's why React is the best choice in this case.

Is a framework the best way to make it scalable for this project?
In a conversation with teacher Bardt van der Dennen I found out that using a framework will not be the best solution for this project. Bardt says that it takes a lot of time to learn a new framework and I don't have that much time to learn it completely in one semester and directly implement it to the portfolio. Especially if you don't master JavaScript completely. He recommends me to use only Vanilla JavaScript. You can still repeat the same objects without copying the same code, so this makes it easy to adjust and is less error prone. He also advised me to use JSON the to put all the data in. This makes it clearer, and I only have to modify the JSON file and not search through the code.

Conclusion

Frameworks can have major advantages, allowing objects that repeat more often in the portfolio to be created faster and better, and they can also be adjusted more easily. The biggest disadvantage is that it takes more time to learn and understand how these frameworks work. For this reason, I chose not to use a framework and to realise this with Vanilla JavaScript only.

Learning Outcomes

Learning outcome 5: Investigative problem solving

This research document uses the CMD research methods. As a result, conclusions have been drawn that provide answers to the sub-questions. these answers help answer the main question.

Learning outcome 3: Software design and realization

By investing in which framework will suit the best for this project.

Literature

CMD Methods, Van Turnhout, K., Jacobs, M., Kamp, I., Mulholland, C., Neuman, A., Rouwhorst, S., & Van Vlies, L. (2015). *CMD methods*. Accessed September 14, 2023, from https://cmdmethods.nl/

Karczewski, D. (2023, 19 juni). Best Frontend Frameworks. *What Are The Best Frontend Frameworks To Use In 2023?* Accessed September 14, 2023, from https://www.ideamotive.co/blog/best-frontend-frameworks

Pekarsky, M. (2020, 3 februari). Does your web app need a front-end framework? - Stack Overflow. Accessed September 14, 2023, from https://stackoverflow.blog/2020/02/03/is-it-time-for-a-front-end-framework/