

# Researching portfolio content strategies

*How can I make the portfolio easily  
maintainable when adding new  
content?*



Date	:	26-09-2023
Version	:	1.1
Status	:	Definitive
Author	:	Luc Swinkels



## Version

Version	Date	Author(s)	Amendments	Status
0.1	15-09-2023	Luc Swinkels	First draft	Draft
1.0	18-09-2023	Luc Swinkels	First definitive version	Definitive
1.1	26-09-2023	Luc Swinkels	Added expert interview	Definitive

## Table of contents

1. Context .....	4
2. Method .....	5
2.1 Literature study .....	5
2.2 Expert interview .....	5
3. Results .....	6
3.1 Literature study .....	6
3.2 Expert interview .....	7
4. Conclusion.....	8
5. Literature .....	9



## 1. Context

The goal of this document is to help me build my portfolio by answering one of my sub questions:

*“How can I make the portfolio easily maintainable when adding new content?”*

This means I need to figure out what content structures are available for my portfolio, and which one fits me best.

## 2. Method

To answer this research question, I am going to use multiple library research CMD methods.

### 2.1 Literature study

I will be browsing the internet for resources on content structures for blog posts or portfolios, what their pros and cons are, and ultimately what would fit best with my type of project (building a portfolio).



### 2.2 Expert interview

I will be conducting an expert interview with Bardt van der Dennen, who is a teacher at FHICT with a background in full-stack development. Having worked with JavaScript frameworks before, he is an ideal candidate to give me insight in ways to write dynamic content using these frameworks.



## 3. Results

### 3.1 Literature study

#### 3.1.1 Which content structures are there?

While I wasn't able to find a curated list in an article, I was able to find structures that kept coming back in my search results, such as:

- Using a full CMS instead of React/Next
- Using a headless CMS alongside React/Next
- Using markdown with Next
- Using markdown and MDX with Next (combine markdown and JSX)
- Using static pages

I have decided to learn more about the 3 options that I find best suited for me, being a headless CMS, markdown with MDX, or static pages.

#### 3.1.2 Using a (headless) CMS

A headless CMS has tons of pros like separating front-end from back-end, being able to write content without technical knowledge of the app, and omnichannel connectivity (being able to send your data to multiple sources) (*Holmes, 2022*).

Using a headless CMS would be the only option to pick from if the content was being maintained by someone with no technical experience. However, since I am the only one working on this project and have no intentions of letting other people write content for me in the future, I am hesitant on using a headless CMS because it would mean I would have to integrate a whole CMS system into my app.

This option would be best suited for non-technical users to be able to add content. It also keeps content separated from your application which could be beneficial.

#### 3.1.3 Using markdown / mdx

Next.js recommends using markdown (with mdx) to write things like blog posts (*Configuring: MDX*, n.d.), which can be compared to portfolio products in a way. It uses markdown to write content and popular packages like MDX allows for integration with JSX based frameworks, by using JSX components inside of markdown files.

This option would be best suited for blogs or content pages that keep a predefined lay-out and content structure that isn't subject to change.

#### 3.1.4 Using static content pages

Static content pages allow for the most freedom as you are able to decide on any lay-out and content, that isn't locked behind a certain structure like markdown. It also means you can use it out of the box without any extra packages.

This option would be best suited towards content pages that have dynamic lay-outs and aren't just massive blobs of text (so more visually oriented content), although requires a bit more effort than markdown because you have to use JSX components.

## 3.2 Expert interview

I conducted an expert interview with Bardt van der Dennen, teacher at FHICT with a background in full-stack development, where I asked him the following questions to gain insight in which method would suit me best:

**Out of these 3 content structure technologies, headless CMS, markdown, static content pages, which do you have personal experience with or have you heard of?**

- I have used all 3 before.

**(If having worked with markdown/mdx) Do you think writing product page content on my portfolio in markdown would be a good idea considering I plan on having dynamic lay-outs based on the content?**

- Lay-outs and styling in markdown are very limited, which could be a big con if you are planning on having multiple lay-outs for products.

**Which benefits are there to using a headless CMS for a portfolio like this? (simple product pages)**

- The separation of front-end and data is very nice for the organization of your app, and allows for a lot of flexibility for future content changes.

**Which of the 3 options do you think are best for future scalability and why?**

- Headless CMS, because of the separation of data and front-end.

**In terms of the user, do you think any of these options are more user-friendly than others (i.e. loading speeds) or do you feel like they only impact development experience?**

- Indirectly, if you can push more content to the user in a quicker way, the end user might visit your page more often for updates. In terms of performance, it doesn't really impact it.

**If you were in my shoes, and you had no bias or experience towards either of these 3, which one would you pick for this portfolio?**

- I would pick a headless CMS, because of the flexibility and the separation of data and front-end. It's also a very useful technology to learn as tons of companies are using it for content management in the modern development world.



## 4. Conclusion

Initially, I chose to use static content pages by writing static HTML pages with JSX inside of Next, because this allows me to change lay-outs dynamically depending on the content that I'm writing. My portfolio will consist of different types of products, some of them will have different lay-outs depending on images and text so I prefer having that type of freedom when writing my content. It also helps that I don't need to learn any new syntaxes, or learn how to use extra dependencies like I would have to when adding a CMS or an MDX package to my portfolio.

However, after my expert interview with Bardt, it became clear to me that using a headless CMS is a great way to organize and manage content, offering more flexibility and dynamic content changes by separating the data with the front-end, and it doesn't require too much implementation.

If I had a portfolio product lay-out where I knew every product was going to be able to use the same kind of lay-out, I probably would have chosen markdown with MDX.



## 5. Literature

1. *Configuring: MDX*. (n.d.). Next.js. <https://nextjs.org/docs/pages/building-your-application/configuring/mdx>
2. Holmes, J. (2022). The benefits of headless CMSes. *Sanity.io*.  
<https://www.sanity.io/headless-cms/headless-cms-benefits>