UI libraries

Which UI library (if any) should be used to develop the front-end?

Available product analysis

Date	:	16-04-2024
Version	:	1.0
Status	:	Definitive
Author	:	Luc Swinkels

Version history

Version	Date	Author(s)	Amendments	Status
0.1	09-04-2024	Luc Swinkels	First draft	Draft
1.0	16-04-2024	Luc Swinkels	First definitive version	Definitive

Table of contents

Context

The goal of this research is to answer the following research question:

“Which UI library (if any) should be used to develop the front-end?

Answering this question means I have to figure out which libraries are available, fit my criteria, and offer the best integration for this project.

To do this, I will be using the available product analysis (Vogel, n.d.) research method.

Methods

Available product analysis

For this research, I will be using the available product analysis (Vogel, n.d.) method, as there are lots of different options available online.

Results

Before choosing a UI library, it is important to set criteria for a library, as there are tons of options nowadays, ranging from complete design systems to simple copy and paste components.

The only libraries or components I would consider have to adhere to the following criteria that I based on my own experience, the project's needs, and the stakeholders needs, with the most important criteria at the top:

Default styling
Styling should be based on TailwindCSS, as my project will be styled with TailwindCSS, and I don't want multiple ways of styling throughout the application as it will turn into messy code. The reason I have opted for TailwindCSS is due to it being the main choice for styling applications within Moonly, as well as having a great personal experience with it. It also directly ties in to one of the project's key requirements, which is to develop a product/prototype that can be easily scaled or used for other projects. By using TailwindCSS, I can copy and paste components and maintain styles across projects, without having to create a whole design system for each project.

Compatibility
The library should be compatible with Hydrogen (React), as my project will be built with Hydrogen.

Pricing
The library should be free, as there are tons of free options available and I don't have a budget to spend on UI library subscriptions.

Customizability
I don't want to be locked into someone else's design. I should be able to customise these components to fit in with the branding and style of my project.

Dependencies
A library shouldn't be dependent on a lot of (unmaintained) packages. I want to make sure that I can still use these components in many years from now without issues, instead of having to migrate all my components to ones that do work.

Accessibility
Basic HTML semantic rules should be followed for accessibility, and it would be a plus to have components that put a lot of effort into making sure each component is accessible, so that I don't have to worry about it.

Selection of components
There should be a wide range of basic components that can be used throughout the project, meaning the library shouldn't be completely new and unfinished.

After ruling out a lot of libraries that did not meet this criteria, I came up with the following list based on reading through documentation, personal experience, peer experience, and online articles (Gerchev, 2023) (Sahan, 2023) (Dumont, 2024):

- **shadcn/ui** (Shadcn, n.d.)
  - shadcn/ui is a very new but exciting and popular library, used by companies such as Vercel. It focuses on creating highly customizable, quality, accessible components. shadcn/ui's styling is a bit more opinionated than some others, and offers lots of interactivity for their components.
- **daisyUI** (*daisyUI — Tailwind CSS Components (Version 4 Update Is Here)*, n.d.)

  - daisyUI is one of the longest standing libraries in this list, and also widely popular. It offers a lot of components with a fairly generic base style, leaving a lot of room for customization.
- **HyperUI** (*Free Open Source Tailwind CSS Components | HyperUI*, n.d.)
  - HyperUI is a fairly generic component library that offers lots of flat components.
- **Aceternity** (*Aceternity UI*, n.d.)
  - Aceternity is a lesser-known UI library that consists of more abstract components that you wouldn't often see in other libraries. This makes it a possible addition or alternative for creating abstract and custom sections.
- **Flowbite** (*Flowbite - Build Websites Even Faster With Components on Top of Tailwind CSS*, n.d.)
  - Flowbite is a fairly generic component library that offers lots of flat but interactive components.
- **Preline UI** (*Preline UI, Crafted With Tailwind CSS*, n.d.)
  - Preline UI is a lesser-known component library that provides a lot of flat components, but also lots of sections where they combine components so you can quickly find examples and tweak them as you like.
- **RippleUI** (*RippleUI | TailwindCSS Components*, n.d.)
  - RippleUI is a library that offers lots of flat, defaultly styled components that feels quite similar to Google's material design.
- **Sira** (*Home - Sira*, n.d.)
  - Sira is a library that offers simple components by using their own pre-defined class names based on Tailwind's existing classes, removing the need to type long tailwind classes.
- **Mamba UI** (*Mamba UI*, n.d.)
  - Mamba UI is a library that offers pre-built components and sections with very minimal styling, focusing on layouts and sizes. Their components don't offer a lot of functionality but act as a good baseline to build upon.
- **Meraki UI** (*Meraki UI Tailwind CSS Components*, n.d.)
  - Meraki UI offers lots of cleanly styled components and sections with lots of variants, focusing on style instead of interactivity.
- **Pagedone** (*Pagedone*, n.d.)
  - Pagedone UI offers a lot of aesthetically pleasing generic components and sections. They offer lots of built-in styling and are most suited to copy and paste as whole sections. While their free version offers a good amount of components, some are locked behind their paid version.
- **Kitwind** (Kitwind, n.d.)
  - Kitwind UI is a UI kit that offers mostly "sections" and groups of components instead of singular components.

While there are lots more options, I have opted to limit my choices to the most popular ones. Before starting this research, I only knew of shadcn/ui and daisyUI, which are very solid and popular options. However, I am surprised by the amount of robust components that other libraries offer.

These libraries generally consist of copy and pasteable components made with Tailwind classes, which means, if they don't bring their own opinionated styling for things like colours and fonts, I could use multiple of these at once.

While all of these libraries allow for some form of customisation by passing Tailwind classes to components, shadcn/ui is one of the only libraries that allows you to install components through a CLI and directly place it into your code, giving you access to the entire source code with no limitations. Libraries like Flowbite offer a middle ground where you import a component, and are able to pass various options to their component to adjust it.

Conclusion

Since my design is rather straightforward and inspired by existing web shops, I want to use a library that has a focus on accessibility, good practices and good interactivity. It has to be based on Tailwind, should be free, and I should be able to directly alter the component to fit my own designs.

I have chosen to use shadcn/ui as my component library because it gives me the most freedom and flexibility over my components, as well as being very well designed in terms of interactivity and accessibility. While the default styling for shadcn/ui components might not always fit my designs, due to having the complete freedom over the component, I can adjust styles where needed and keep the interactivity aspect of the component.

Sources

1. Vogel, J. (n.d.). ICT Research Methods — Methods Pack for research in ICT. ICT Research Methods. https://ictresearchmethods.nl/library/available-product-analysis/
2. Gerchev, I. (2023, December 20). 10 best Tailwind CSS component libraries - LogRocket Blog. LogRocket Blog. https://blog.logrocket.com/10-best-tailwind-css-component-libraries/
3. Sahan, Y. F. (2023, October 18). 11 Best Tailwind CSS Component Libraries in 2023 - Landing Page Tips - medium. Medium. https://medium.com/landing-page-tips/11-best-tailwind-css-component-libraries-in-2023-9c5383a6dd0e
4. Dumont, D. (2024, January 31). You are using Tailwind CSS and looking to speed up the integration of blocks into your site or web app ? This page is for you. We have selected the best Tailwind CSS UI component kits. https://suncel.io/ressources/top/top-12-best-tailwind-ui-components-kits
5. Shadcn. (n.d.). shadcn/ui. Shadcn/Ui. https://ui.shadcn.com/
6. daisyUI — Tailwind CSS Components (version 4 update is here). (n.d.). https://daisyui.com/
7. Free open source Tailwind CSS Components | HyperUI. (n.d.). HyperUI. https://www.hyperui.dev/
8. Aceternity UI. (n.d.). https://ui.aceternity.com/
9. Flowbite - Build websites even faster with components on top of Tailwind CSS. (n.d.). https://flowbite.com/
10. Preline UI, crafted with Tailwind CSS. (n.d.). Preline. https://preline.co/index.html
11. RippleUI | TailwindCSS Components. (n.d.). https://www.ripple-ui.com/
12. Home - Sira. (n.d.). https://sira.riazer.com/
13. Mamba UI. (n.d.). Mamba UI. https://mambaui.com/
14. Meraki UI Tailwind CSS Components. (n.d.). https://merakiui.com/
15. Pagedone. (n.d.). https://pagedone.io/
16. Kitwind. (n.d.). Coming Soon. https://kitwind.io/