

How Design Can Lead Frontend Development



Mirha Masala · September 2022

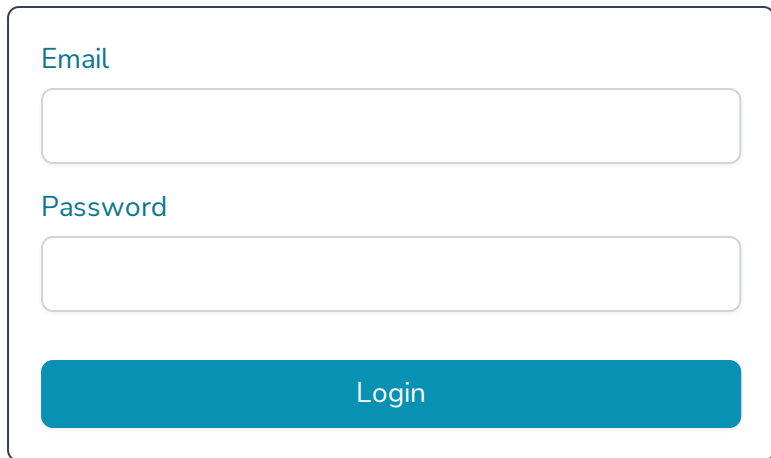
What I Often Find Missing in Designs

- Named components (and named frames and grouped layers)

Mostly, a *design system*.

- Pre-defined layouts
- Variants and different states for components
- Unhappy path
- Annotation of interactions
- One sheet for icons, logos, and components
- Consistency for colors, font sizes, and spacings (and plenty of each)
- Clean SVGs and other assets
- Clear versioning

Thinking in Components



Email

Password

Login

Creating Components

1. Draw boxes around every component (and subcomponent).
2. Give them all names.

How do you know what should be its own component?

- One such technique is the *single responsibility principle*, that is, a component should ideally only do one thing.
- If it [a component] ends up growing, it should be decomposed into smaller subcomponents.

Source: Thinking in React

Form Split Into Components I

Email

Password

Login

```
1   <!-- 1. FormLabel -->
2   <FormLabel />
```

```
1   <!-- 2. FormInput -->
2   <FormInput />
```

```
1   <!-- 3. FormSubmitButton -->
2   <FormSubmitButton />
```

```
1   <!-- 4. FormField -->
2   <FormField />
```

```
1   <!-- 5. FormLayout -->
2   <FormLayout />
```

Form Split Into Components II

Email

Password

Login

```
1  <FormLayout>
2    <FormField>
3      <FormLabel />
4      <FormInput />
5    </FormField>
6    <FormField>
7      <FormLabel />
8      <FormInput />
9    </FormField>
10   <FormSubmitButton />
11 </FormLayout>
```

Form Folder Structure

Not unlike how one may sort files and folders on their computer (if you're *Sheldon* like me)

```
1  my-project/  
2    |  
3    └─ components/  
4        │  
4        │   ├── FormField  
5        │   ├── FormInput  
6        │   ├── FormLabel  
7        │   ├── FormLayout  
8        │   └── FormSubmitButton
```



Benefits

If designers and frontend developers work together to create this level of structure:

- There'd be visual consistency
- Features could be developed faster
- Code could be kept DRY (Don't repeat yourself)
 - Which in turn would help to keep the codebase more manageable
- New features could be developed faster
- Design, Product, and Development can point to a single-ish source of truth and use the same naming for features, pages, etc.
- Collaboration would be much more easygoing and fun
- Motivation for the project could be kept high

Design Systems

- [Carbon](#)
- [Evergreen](#)
- [Pluralsight](#)
- [Primer React](#)
- [Zendesk](#)

And many more at [styleguide.io](#).

More

- [Atomic Design](#) - On creating design systems with atoms, molecules, organisms, templates and pages.
- [BEM](#) - A naming convention
- [Storybook Showcase](#)
- [WordPress Gutenberg](#) + [WordPress Patterns](#)

Resources

- Thinking in Components
- Thinking in React
- Thinking in React: The 2020 version
- Scaling Your Vue Components for Mid-Large Size Apps