

TRANSLATING THE NJ DESIGN SYSTEM INTO WEB COMPONENTS

State of New Jersey – Office of Innovation | Resident Experience Initiative

Elizabeth (Lizzy) Zhang, Software Engineering Fellow | *Brown University, Computer Science & Biology*

Keywords:

front-end, web components, design system, user research

Summary:

Lizzy helped update and standardize the [New Jersey Web Design System](#) (NJWDS), leading the engineering effort to convert NJWDS components from plain HTML into web components. She conducted user research with engineers to understand their needs for a web component library. Incorporating the results of her research, Lizzy created and published an initial version of the **njwds-components** package on NPM using **Stencil** and **TypeScript**. She also created a demo repository with examples and documentation of how to use the package across different frameworks. Lizzy's work sets the foundation for a library that will enable developers to use the design system more effectively, increasing their capacity to create consistent, accessible surfaces for New Jersey digital properties and services.

SOFTWARE ENGINEERING FELLOWSHIP SUMMER 2024

Resident Experience

State of New Jersey – Office of Innovation

Anne Loverso — Engineering Co-Director

coding it forward >



ELIZABETH ZHANG

Brown University
Computer Science and Biology

ROADMAP

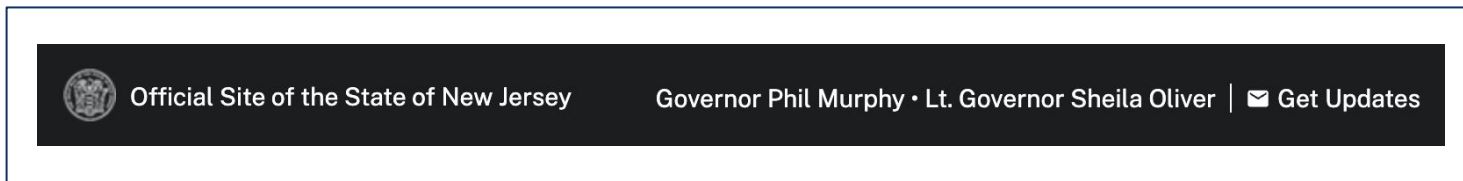
1. Translating the NJ Web Design System (NJWDS) into Web Components
2. Home Experience: Developing an Eviction Informational Intake Tool

TRANSLATING THE NJ WEB DESIGN SYSTEM INTO WEB COMPONENTS

BACKGROUND

- New Jersey Web Design System (NJWDS)
- Extension of the [US Web Design System](#)
- Components and theming for NJ digital products and services
- Used in multiple projects across Business Experience and Resident Experience
 - e.x. Innovation Site, Business Navigator/MyAccount, Unemployment Insurance
- Efforts to update and standardize design system
 - Engineering: translate NJWDS components from plain HTML to web components

Interface element



Plain HTML

```
<section class="nj-banner" aria-label="Official government
website">
  <header class="nj-banner__header">
    <div class="grid-container">
      <div class="nj-banner__inner">
        <div class="grid-col-auto">
          
          ...
        </div>
      </div>
    </div>
  </header>
</section>
```

Web component

```
<njwds-banner></njwds-banner>
```

WEB COMPONENTS

- Browser feature that allows developers to create reusable custom HTML elements
 - Native elements: `<button>`, `<heading>`, `<p>`
 - Custom element: `<njwds-button type="secondary">`
- Web components offer a better developer experience:
 - Ease of use
 - Enforce correct usage
 - Deliver updates automatically

USER RESEARCH: PUBLISHING

Goals

1. Understand how NJWDS is **currently used** by engineers in the office.
2. Understand the most useful ways for engineers to **publish, import, and contribute** NJWDS web components.

Methods: 30 minute interviews with three engineers experienced with NJWDS

- Across four projects on both Business Experience and Resident Experience

SYNTHESIS STEP 1: GROUP BY INTERVIEW

User Interviews

Engineer 1

Context

Training engineer
Engineer 1 (Age 34)
Intermediate software
engineer with 10 years of
experience in web
development, primarily
React and Node.js.
Recently transitioned
from a senior role to a
senior role.

Styling

Uses a design system
based on a set of
design tokens. The
design system is
maintained by a
design team and
used by all developers.

Importing Web Components

Using a library to
import web components
from a third-party
library. The library
provides a set of
web components that
can be imported into
the application.

Importing React Components

Imports a set of
React components
from a third-party
library. The library
provides a set of
React components that
can be imported into
the application.

Publishing

Uses a build system
to build the application
and a package manager
to publish the
application to a
package registry.

Contributing Components

Would be interested in
contributing to a
library. The library
provides a set of
web components that
can be imported into
the application.

Would be interested in
contributing to a
library. The library
provides a set of
web components that
can be imported into
the application.

Engineer 2

Context

Senior developer
Engineer 2 (Age 32)
Senior developer with
12 years of experience
in web development,
primarily React and
Node.js.

Styling

Uses a design system
based on a set of
design tokens. The
design system is
maintained by a
design team and
used by all developers.

Indexability

Imports a set of
React components
from a third-party
library. The library
provides a set of
React components that
can be imported into
the application.

Importing Web Components

Using a library to
import web components
from a third-party
library. The library
provides a set of
web components that
can be imported into
the application.

Importing React Components

Imports a set of
React components
from a third-party
library. The library
provides a set of
React components that
can be imported into
the application.

Publishing

Uses a build system
to build the application
and a package manager
to publish the
application to a
package registry.

Documentation

Would be interested in
contributing to a
library. The library
provides a set of
web components that
can be imported into
the application.

Contributing Components

Would be interested in
contributing to a
library. The library
provides a set of
web components that
can be imported into
the application.

Additional Feedback

Would be interested in
contributing to a
library. The library
provides a set of
web components that
can be imported into
the application.

Engineer 3

Context

Senior developer
Engineer 3 (Age 35)
Senior developer with
13 years of experience
in web development,
primarily React and
Node.js.

Styling

Uses a design system
based on a set of
design tokens. The
design system is
maintained by a
design team and
used by all developers.

Indexability

Imports a set of
React components
from a third-party
library. The library
provides a set of
React components that
can be imported into
the application.

Importing Web Components

Using a library to
import web components
from a third-party
library. The library
provides a set of
web components that
can be imported into
the application.

Importing React Components

Imports a set of
React components
from a third-party
library. The library
provides a set of
React components that
can be imported into
the application.

Publishing

Uses a build system
to build the application
and a package manager
to publish the
application to a
package registry.

Documentation

Would be interested in
contributing to a
library. The library
provides a set of
web components that
can be imported into
the application.

Contributing Components

Would be interested in
contributing to a
library. The library
provides a set of
web components that
can be imported into
the application.

Additional Feedback

Would be interested in
contributing to a
library. The library
provides a set of
web components that
can be imported into
the application.

SYNTHESIS STEP 2: GROUP BY THEME

Themes

ResX and BizX projects use NJWDS differently.

While ResX project frequently consumes NJWDS components, BizX mostly uses NJWDS for the styling utilities.

"90% of the time we're using components directly (on Innovation website and UI claims status application)"

Copying NJWDS component HTML is starting point for both Innovation website and UI mod. Especially for Innovation site (no no-actor designer), goes to NJWDS component page first. If can't find what they're looking for, then goes to USWDS.

Business Navigator/ MyAccount mostly uses NJWDS for styling

Innovation website uses both styling classes and components.

Style clashes are a present but manageable issue.

Style clashes are a fairly common issue that projects have managed using CSS overrides or custom classes. However, styling concerns have blocked Business Navigator from upgrading NJWDS versions.

Has experienced styling clashes, usually around colors. Solution: custom override using an important flag or creating another style with higher priority

No styling clashes on Innovation website

MyAccount (uses SASS, MUI, direct styling, NJWDS/USWDS). Currently overriding some classes. Frowned to version 0.3.0 and afraid to update because of layout shifts/style changes

Mostly no styling clashes, usually working with NJWDS in mind.

Indexability is important to some projects using NJWDS.

For some projects, having indexable component content is crucial. For others, it doesn't matter at all, either because it's a client-side application or it doesn't need to show up in search results.

Indexability not necessary for Business Navigator since it's client-side application.

Would want 100% indexability for Innovation site. Want to ensure dynamic content can be put on the website while maintaining SEO.

SEO/Indexability not a concern for UI project.

Indexability matters more on Innovation Site. Can't think of a specific example of indexing content (vs. page discovery). But the more options you give the more they'll want it.

Engineers would rather have a package with all React components vs. an individual package for each component.

Engineers expressed a preference for a single package due to convenience, as projects using NJWDS are likely to use more than one component.

Probably easier to have all React components in the same package, assuming using more than one component

Importing React components in unified vs. individual package should matter with tree shaking in modern bundlers. Still a better pattern for engineers to import specific components from package to show where it's coming from.

Not sure, but will almost always be using more than one component, so would probably want to install all components in one package. As NJWDS gets bigger, could have additional "TAC" packages for specific components (e.g. feedback widget)

Engineers are willing to do the required setup to use React components, as long as it's well documented.

Calling the `defineCustomElements` function and (Next.js specific) wrapping components was not seen as an undue burden on using NJWDS React components.

Setup required for using React components in Next.js doesn't seem like too much work.

To decide whether extra setup of React components is worth it has to be better understood trade-off between using React vs. web components. Seems like a one-time learning curve. If it's well documented, it shouldn't be too much of a headache.

Engineers want documentation that is easy to navigate and contains examples, particularly of how different components work together.

"A very basic example of js form steps" but that is fully fleshed out of the different components working together. Would be the number one thing that would increase our adoption of NJWDS components beyond just the utility classes"

Would be helpful to have mirror of USWDS documentation for NJWDS docs. Eliminates need to have to look at two different places - can better document when we've made changes to USWDS.

"If there are more architectural changes needed, having example documentation at least in our most commonly used environments... would be nice to have"

USER RESEARCH: DESIGN DECISIONS

- 13 themes from 3 interviews → 8 design decisions...
 - **Publish** - two packages: “base NJWDS” + “NJWDS Components”
 - **Import** - from both local project files and remote server
 - **Contribute** - clear governance and process for adding components

NJWDS: ENGINEERING WORK

- Proved feasibility of [Stencil](#)
 - Software library for building web components
 - Offers integration with React framework
- Created two GitHub repositories
 - Component library (njwds-components) published on NPM
 - Alert component
 - Demo repository
 - Four examples: vanilla HTML, React, Next.js, Astro
- Established developer workflow for implementing components

NJWDS: NEXT STEPS

1. **Foundation for njwds-components library**
2. Translate initial web components
 - Partnership with Home Experience
3. Release v1 for engineers office-wide (and beyond!)

HOME EXPERIENCE: DEVELOPING AN EVICTIION INFORMATIONAL INTAKE TOOL

BACKGROUND

Create a tool for New Jerseyans facing eviction

1. Provides informational resources to help navigate the eviction process
2. Connects people directly to a Resource Navigator in their county

Innovation starts here

Eviction Protection

Outcome

Residents facing eviction have a direct and simple path to connect to assistance

Resident-initiated intake for CEDD agencies

Intake Status Tracking

Outcome

Residents facing eviction can find clear information about what they should do at each stage of seeking assistance

Responsive Informational Guide focused on Eviction

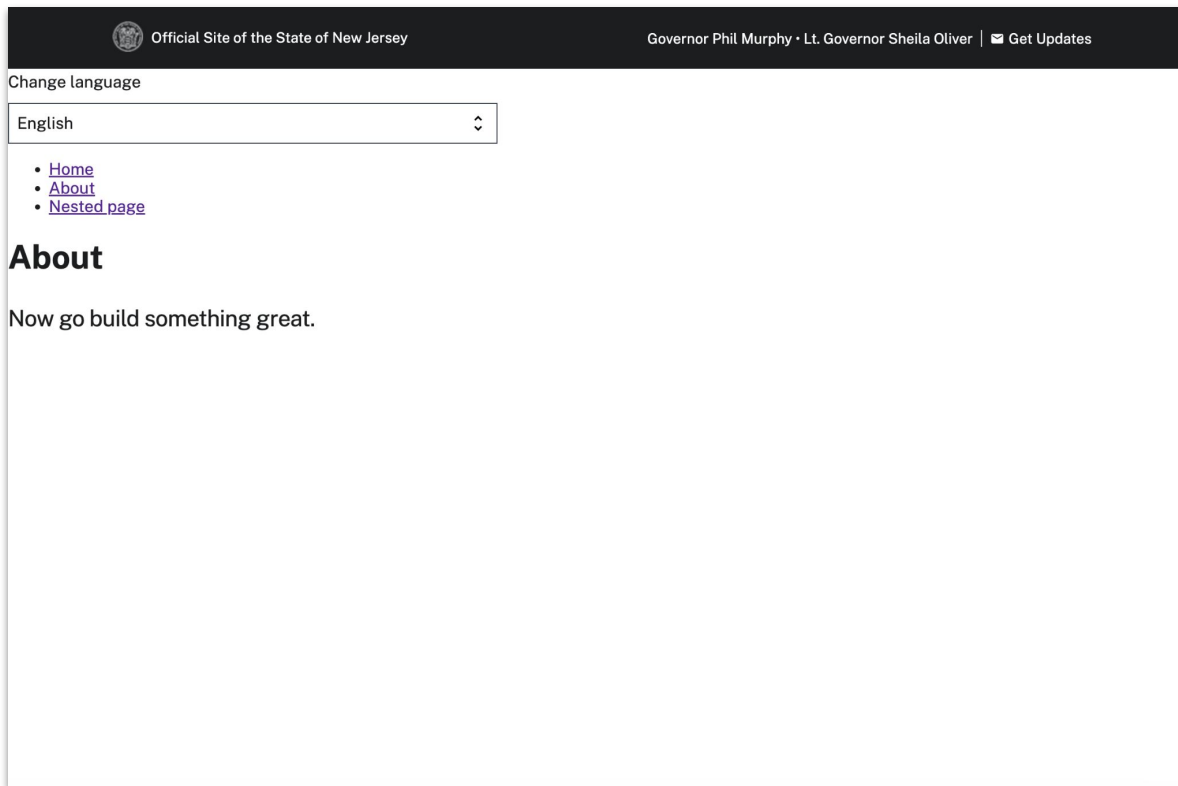
Iterate through several versions



TECHNICAL EXPLORATION

- Compared three frameworks for static sites: Next.js, Gatsby, Astro
- Starter application for each framework
 - Basic layout using NJWDS
 - Internationalization
 - Switch between English & Spanish
 - Localize URL Names (“en/about” → “/es/quienes-somos”)
 - Save language preference

EXAMPLE: STARTER APPLICATION



RECOMMENDING A SOLUTION

- Technical approach document for the Data Director at the Department of Community Affairs (DCA)
- Three main criteria:
 1. Internationalization functionality
 2. Long-term maintainability by DCA
 3. Stability & long-term support
- Final recommendation: Astro → accepted!

STARTING DEVELOPMENT

- Helped build and test the language selector component
- Detects and saves user's browser language preference
- **Next steps:** continue building intake form for second round of user testing

English Español

Apply for eviction help

Based on your answers, we'll connect you with a local expert (resource navigator) who can help you find financial and legal aid. We worked with people who experienced eviction to make these questions less stressful.

**THANK YOU SO
MUCH!**