

Digital NSW Work Task - Carousel Component

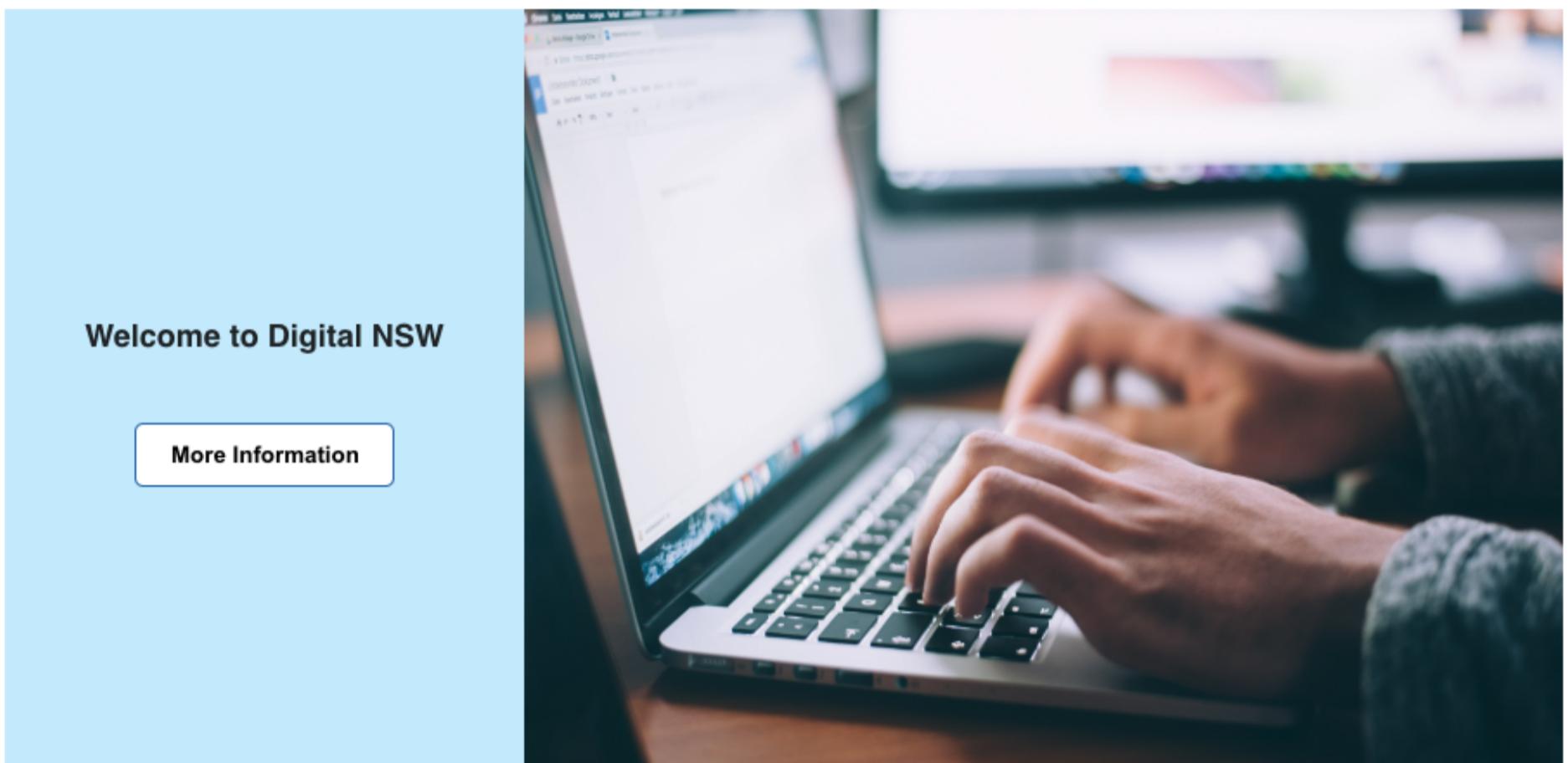


Table of Content

1. [Planning and building](#)

1.1 [Evaluating the need for carousel component](#)

1.2 [Planning for building the component](#)

2. [Proposed Design](#)

2.1 [Theme styles](#)

2.2 [Autoplay](#)

2.3 [Play speed](#)

2.4 [Content layout](#)

2.5 [Responsive design](#)

3. [Accessibility](#)

4. [Testing](#)

5. [Storybook](#)

6. [Installation](#)

Planning and building

Evaluating the need for carousel component

Upon reviewing the existing design system components, the carousel component is missing in the current design system. Recognizing the potential to enhance user experience, I have conceived the idea of developing this component. By introducing a carousel feature, we aim to infuse the digital NSW website with a sense of dynamism, visual appeal, and enhanced user interactions.

Planning for building the component

Four main objects for Carousel component is to achieve:

- Extensibility
- Consistency
- User-friendly
- Responsive

For extensibility: In developing this component, my intention was to create a versatile and expandable solution. The component should provide future growth based on systems design principle. The component should has the quality of being able to add new functionality and capabilities.

For consistency: To ensure a cohesive visual experience, the carousel need to seamlessly integrate with the existing website styles. By harmonizing background colors, font styles, and layout, the component would match the styles of the exisiting website, and it should looks coherent across the other different elements. Then it will deliver a unified and high-quality user experience.

For user-friendly: Catering to users of varying ages and abilities, including seniors and individuals with visual or hearing impairments, the carousel was designed with the objective of simplicity. It was crucial to strike a balance between functionality and ease of use, enabling effortless navigation and interaction for all user groups.

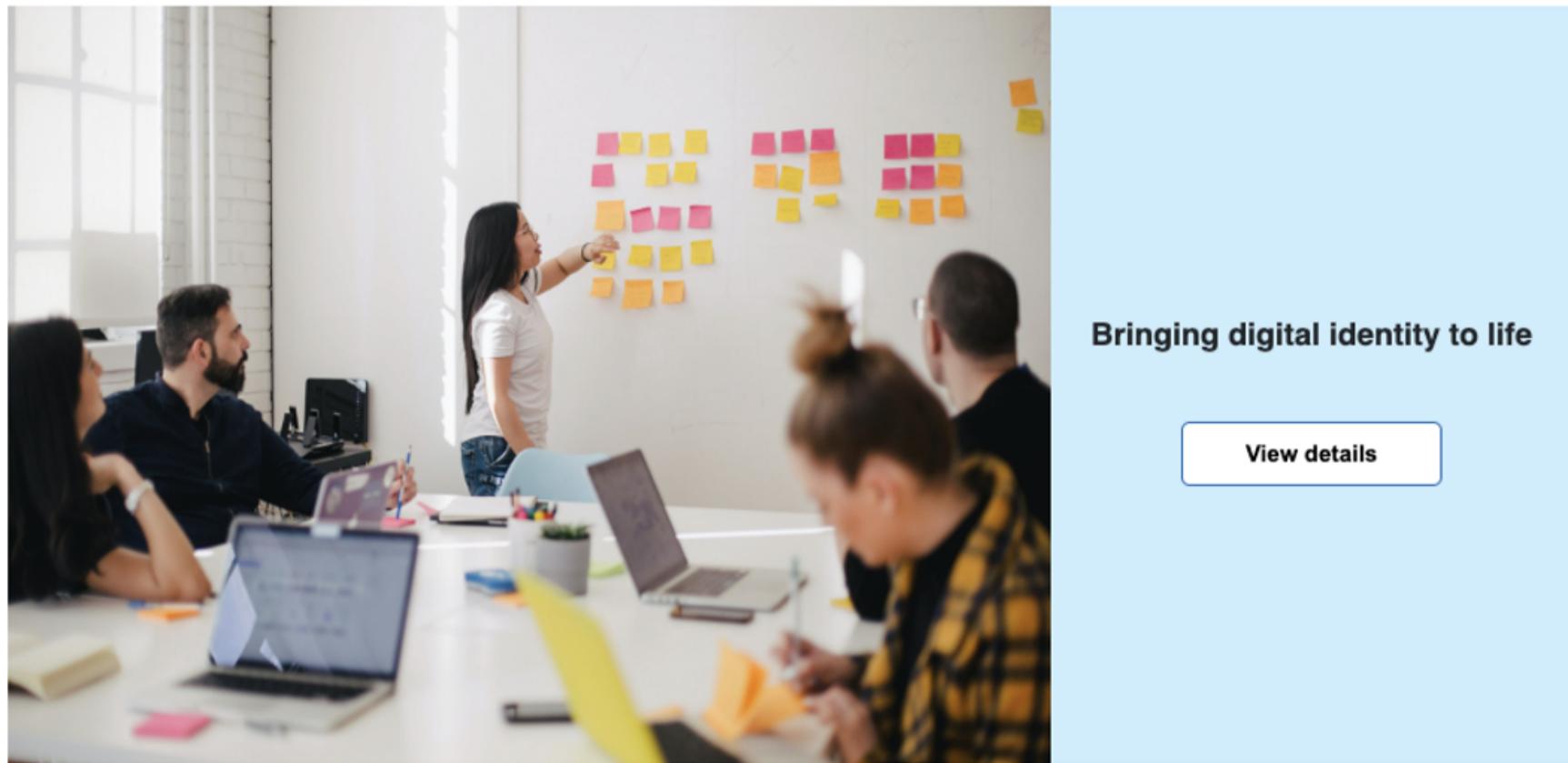
For Responsive: Recognizing the diverse range of devices utilized by website visitors, I prioritized responsive design principles. By ensuring the carousel's adaptability across different screen sizes, such as mobile, tablet, and laptop, users could enjoy a consistently optimal experience regardless of their chosen device.

Proposed Design

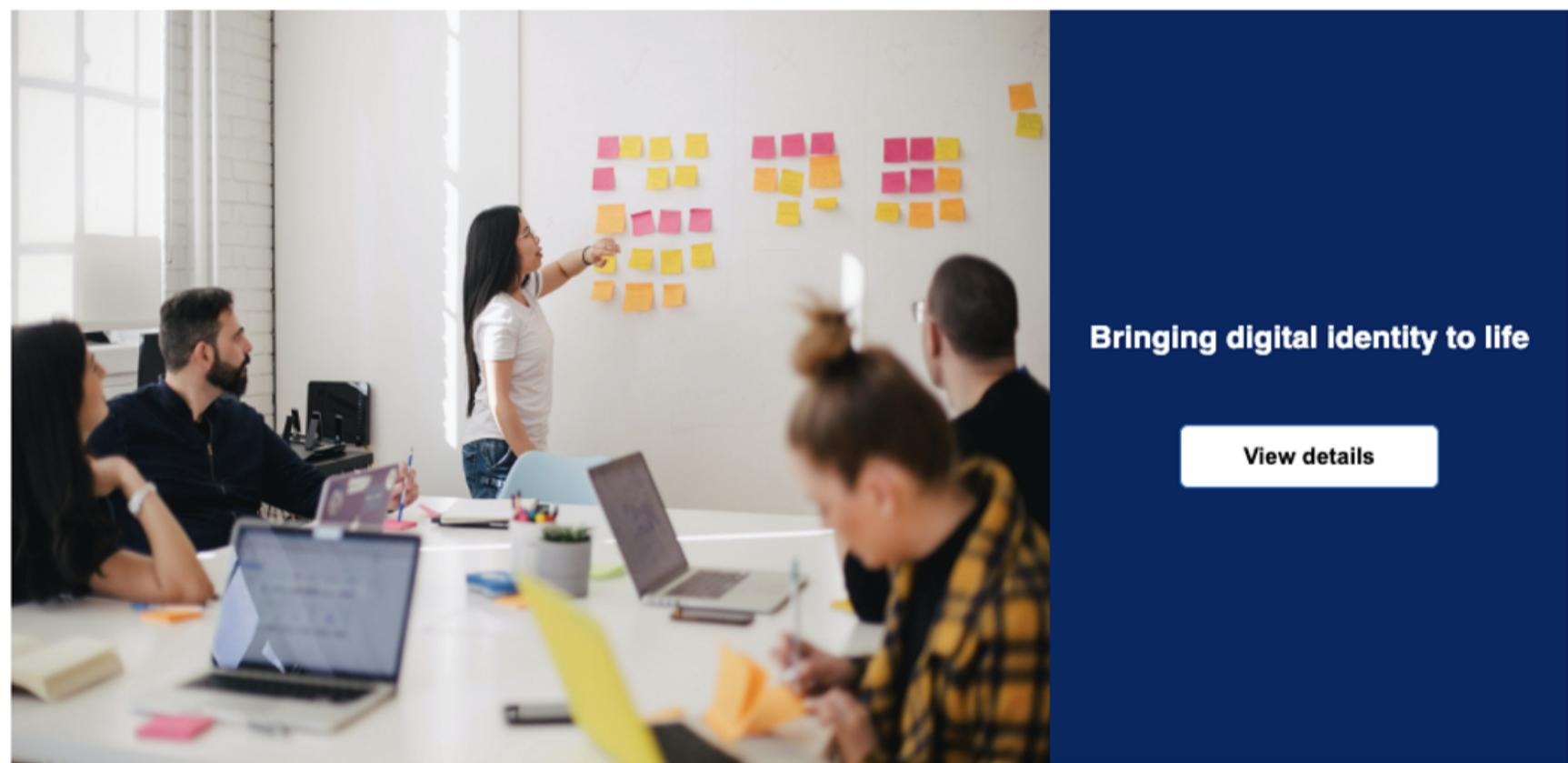
Theme styles

In an effort to enhance user satisfaction, I introduced multiple theme variations for the carousel component, including dark and light modes. Acknowledging that some users may experience eye strain with a light background, the inclusion of a dark theme caters to their preferences. This flexibility empowers users to switch between themes based on their individual comfort and visual preferences.

Light Theme



Dark Theme



Autoplay

To infuse dynamism into the website, I incorporated autoplay functionality within the carousel. However, based on the WACG guidelines, reference on [WCAG 2.1 Guidelines Explained with Examples](#)

Some users might experience distraction or nausea from animated or moving content. If scrolling through a page causing elements to move or scrolling with backgrounds that move at a different rate than the foreground, there must be an option for the user to turn these off or the user may enable reduce motion preferences.



Play speed

To adhere to WCAG standards, I implemented play speed control, enabling users to adjust the speed at which carousel slides transition. Based on the reference on [Web Content Accessibility checklist](#)

For any auto-updating information that (1) starts automatically and (2) is presented in parallel with other content, there is a mechanism for the user to pause, stop, or hide it or to control the frequency of the update unless the auto-updating is part of an activity where it is essential.

Therefore, by inputting desired intervals, users can customize the timing to their preference. The default setting of 2 seconds per slide can be easily modified, empowering users to optimize their carousel experience.

Name	Description	Default	Control
intervalInSeconds	number	2	2

Content layout

Flexibility in content layout was a paramount consideration during development. Users now have the freedom to choose whether the image is positioned on the left or right side of the carousel. This feature accommodates individual reading preferences, with some users favoring text placement on the left-hand side and others on the right-hand side.

Image on left position:



Welcome to Digital NSW

More Information

< ⏪ ⏩ >

Show code

A screenshot of a digital interface showing a person's hands typing on a laptop keyboard. To the right of the image is a blue sidebar with the text "Welcome to Digital NSW" and a "More Information" button. Below the image are navigation arrows and a "Show code" button.

Name	Description	Default	Control
intervalInSeconds	number	2	2
imagePosition	string	"right"	<input checked="" type="radio"/> left <input type="radio"/> right

Image on right position:

Welcome to Digital NSW

More Information

< ⏪ ⏹ ⏺ >

Show code

Name	Description	Default	Control
intervalInSeconds	number	2	<input type="text" value="2"/>
imagePosition	string	"right"	<input type="radio"/> left <input checked="" type="radio"/> right

Responsive design

Recognizing the importance of accessibility, I dedicated effort to ensure responsive design across various screen sizes. By implementing the component's responsiveness, users can access and enjoy the carousel on devices of all types, ranging from mobiles and tablets to laptops. This comprehensive approach ensures an optimal user experience, particularly for individuals with low vision who frequently utilize mobile devices. The following figure is the overview of the component across different screen size.

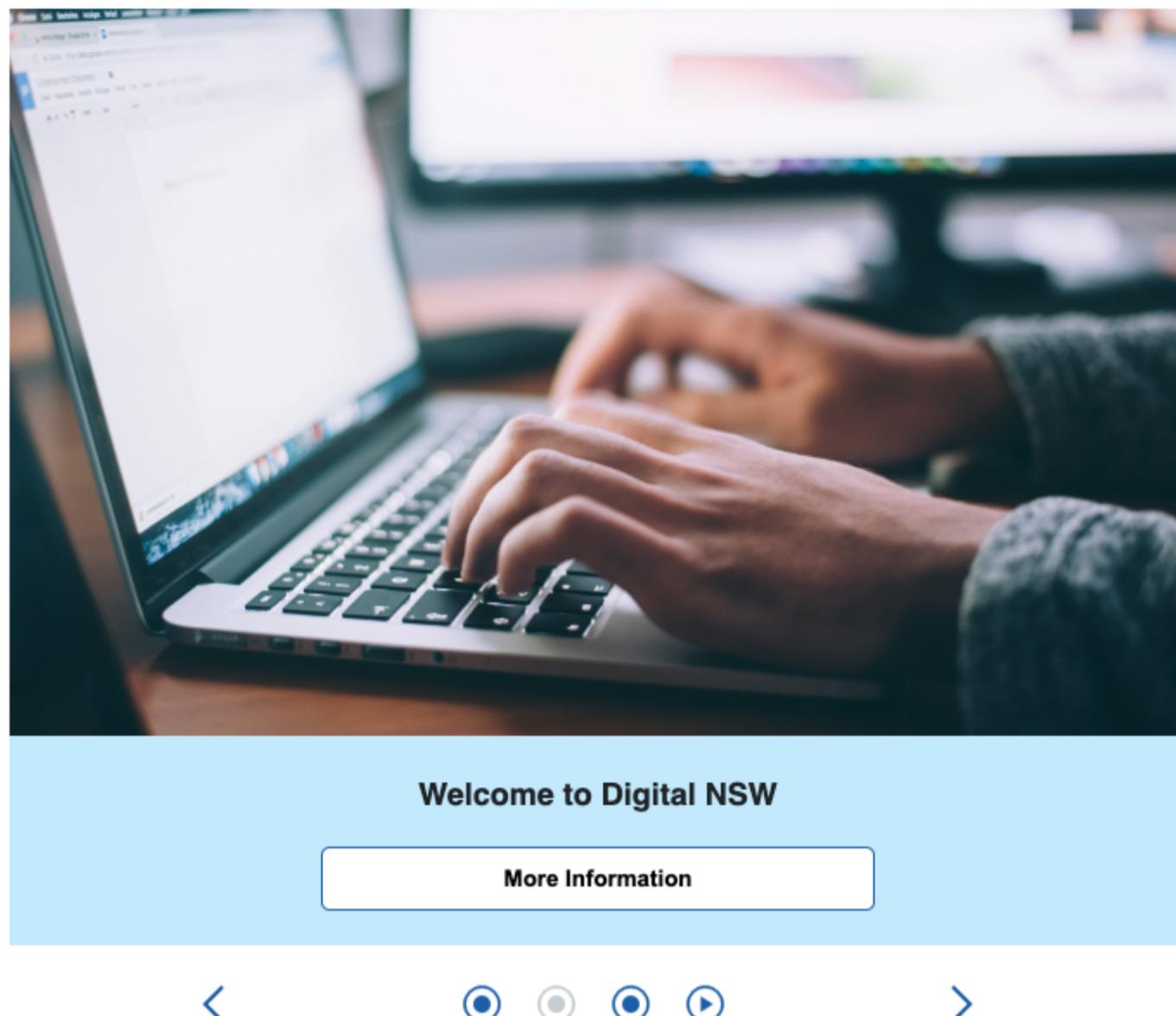
Laptop view

Welcome to Digital NSW

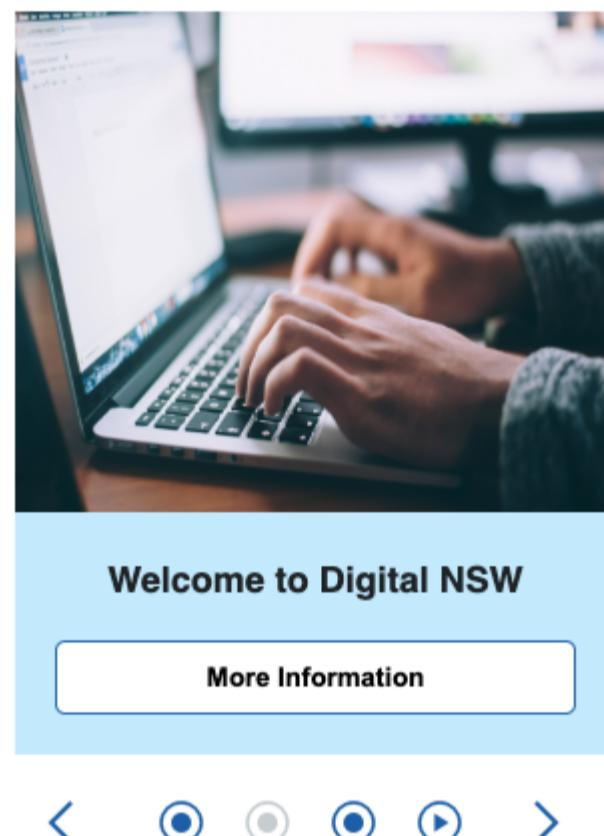
More Information

< ⏪ ⏹ ⏺ >

Tablet view



Mobile view



Accessibility

Screen reader

To enhance the accessibility for this application, I have included aria labels for some of the elements. Using ARIA labels is crucial for screen readers to provide accurate and meaningful information to users with visual impairments. According to the WCAG 2.1 accessibility standard, ARIA (Accessible Rich Internet Applications) is a set of attributes that can be added to HTML elements to provide additional information about their behavior and purpose.

reference on [WCAG 2.1 Web Content Accessibility Guidelines](#)

Assistive technologies that are important in the context of this document include, the screen readers, which are used by people who are blind to read textual information through synthesized speech or braille;

With the help of aria labels, we can provide a more inclusive and accessible web experience for individuals.

Keyboard control

I have also implemented the 'onKeyDown' event method to ensure the keyboard control and improving accessibility on web pages, as emphasized by the WCAG accessibility standard. Keyboard control refers to the ability for users to navigate and interact with a website using only their keyboard, without relying on a mouse or other pointing device. By providing robust keyboard control, websites become more accessible to individuals who may have difficulty using a mouse or other pointing device.

Testing

To run the test:

```
npm run test
```

I used React testing library and Jest to test the component, The testing cases includes:

- application should render carousel component
- component should render carousel indicators
- component should render play/pause auto play button
- component should render back to previous slide and go to next side arrow buttons
- component should render next slide when user click right arrow button
- component should render previous slide when user click the left arrow button

The following screenshot shows the test cases for this component:

```
PASS  src/components/carousel/Carousel.test.tsx
Carousel component
  Component Rendering
    ✓ it should render carousel component (67 ms)
    ✓ it should render carousel indicators (62 ms)
    ✓ it should render play/pause auto play button (24 ms)
    ✓ it should render back to previous slide and go to next side arrow buttons (37 ms)
    ✓ it should render next slide when user click right arrow button (40 ms)
    ✓ it should render previous slide when user click the left arrow button (34 ms)

Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        2.462 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.□
```

Storybook

To view the component in storybook:

```
npm run storybook
```

To provide a better overview of different variations of the component, I have implemented the Storybook. Storybook allows developers and designers to showcase different variations of a component in an isolated and interactive environment. It provides a dedicated space to view and explore various states and styles of the component.

The following picture shows an overview of the Storybook docs and canvas:

The screenshot shows the Storybook interface. At the top, there's a navigation bar with icons for search, refresh, and file operations. Below it is a sidebar with a tree view containing 'Carousel' (selected), 'Docs', and 'Carousel'. A search bar says 'Find components /'. The main area displays a carousel component. It features a photograph of a team working in an office, with a woman pointing at sticky notes on a wall. To the right of the image is the text 'Bringing digital identity to life' and a 'View details' button. Below the image are navigation arrows and a circular progress indicator. At the bottom is a controls panel titled 'Controls (4)' with tabs for 'Actions' and 'Interactions'. It lists four properties: 'intervalInSeconds' (set to 2), 'imagePosition' (set to 'left'), 'style' (set to 'light'), and 'carouselItems*' (which is expanded to show three items, each with four keys). There's also a 'RAW' button.

Installation

This application requires [npm](#) to run.

Please install the dependencies and start the application.

```
npm install  
npm start
```

To start Storybook.

```
npm run stroybook
```

Contact

Please contact: sophiesun.au@gmail.com