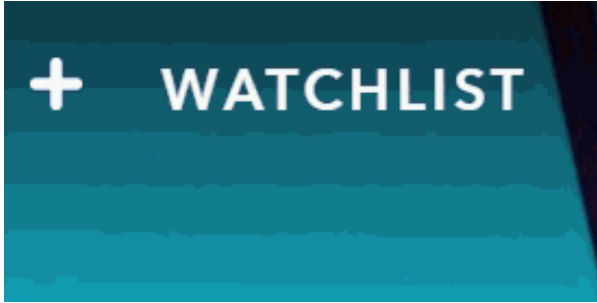


6. Animating the Menu



element.

This is a simple three part component. It has an icon, text and an animated hover

Let's break it down...

NavBtn.js contains an `img` & `span` tag simply wrapped in an `<a>` tag. We pass in the route, icon and text to make it reusable.

AppBar.js simply takes a number of these elements (defined in a constant array at the top of the file see `NavBtns Array` example) and uses a `map` to place them inside a `div` tag that lays them out horizontally.

The magic happens in `navBtn.css`

You can see from code that the icon & text are inside of the component. The animated line however is not. This element is entirely in CSS and requires us to:

- Draw a Line
- Position the line under the element and fit it to the content
- Animate the line on hover.

What is `::before`?

This is a simple CSS function that allows us to modify the CSS before the render (there is the complimentary `::after` too). The only requirement here is **content**, this value simply adds the given content before (or after) the element (we put an empty string here as we don't use it for this purpose)

For example **content**: `"(-)"` would prefix every element `(-)`. This can be especially useful on the `::after` tag to place common suffixes like `:"` without having to add them in code.

Drawing a Line:

CSS isn't a drawing tool and doesn't have any option to simply draw a line so we need to get creative. If you take a look at CSS line 26, we utilise the **`border-radius`** & **`height`** attributes to effectively draw a flat box.

Positioning:

We utilise **`absolute`** positioning for this, which means we take the anchor (usually top-left corner) and we move the element from that position. We **`left`** align move the **`bottom`** by 6px to bring the element below the text. We then assign **`width:auto`** to set the **`width`** equal to the element width which is set by the text, this ensures the line stretches from beginning to end of the text.

Animating:

NavBtn.js

```
import { useNavigate } from
'react-router-dom';
import styles from './navBtn.
module.css';
const NavBtn = ({ route, icon,
text }) => {
  const navigate =
useNavigate();
  return (
    <a className={styles.
navBtn} onClick={() => navigate
(route)}>
      <img src={icon} alt=
{text} />
      <span>{text}</span>
    </a>
  );
};export default NavBtn;
```

AppBar.js - NavBtns Array

```
const navBtns = [
  {
    route: routes.home,
    icon: `${routes.icons}/home-
icon.svg`,
    text: 'Home',
  },
  {
    route: `${routes.category}
/series`,
    icon: `${routes.icons}
/series-icon.svg`,
    text: 'Series',
  },
  {
    route: `${routes.category}
/original`,
    icon: `${routes.icons}
/original-icon.svg`,
    text: 'Originals',
  }
]
```

There are a few ways to animate in CSS, in this case we are using the basic CSS **transition** value. The transition property takes a property to animate over a period of time (read [here](#) for more). In our case we use the **all** keyword to take any transitionable property (not all properties can animate) that appears in both the before and the corresponding end value (hover on line 42). We then add the **span: hover::before** to define the end values. This completes our transition and simply scales the X axis from 0-1 and the opacity from 0-1.

Tip:

If you are working with animations, get the setup correct before adding in the transition elements. This will allow you to see what you are doing before changing things such as scale & opacity which confuse things!

Useful Resources:

[Border CSS Generator](#) - Because working out these values by hand is a waste of time.

AppBar.js Render

```
//App Bar Render Function
// ... code here
<div className={styles.navBtns}>
  {navBtns.map((item) => (
    <NavBtn key={item.text} route=
    {item.route} icon={item.icon}
    text={item.text} /> ))}
</div>
// ... code here
```

NavBtn.css

```
.navBtn {
  display: flex;
  align-items: center;
  padding: 0 12px;
  cursor: pointer;
}

.navBtn > img {
  height: 20px;
  min-width: 20px;
  width: 20px;
  z-index: auto;
}

.navBtn > span {
  color: rgb(249, 249, 249);
  font-size: 13px;
  letter-spacing: 1.42px;
  line-height: 1.08;
  padding: 2px 0px;
  white-space: nowrap;
  position: relative;
  text-transform: uppercase;
}

.navBtn > span::before {
  background-color: rgb(249,
249, 249);
  border-radius: 0px 0px 4px
4px;
  bottom: -6px;
  left: 0px;
  right: 0px;
  height: 2px;
  content: '';
  position: absolute;
  opacity: 0;
  transform-origin: left
center;
  transform: scaleX(0);
  transition: all 250ms cubic-
bezier(0.25, 0.46, 0.45, 0.94)
0s;
  width: auto;
}

.navBtn > span: hover::before {
  transform: scaleX(1);
  visibility: visible;
  opacity: 1 !important;
}
```