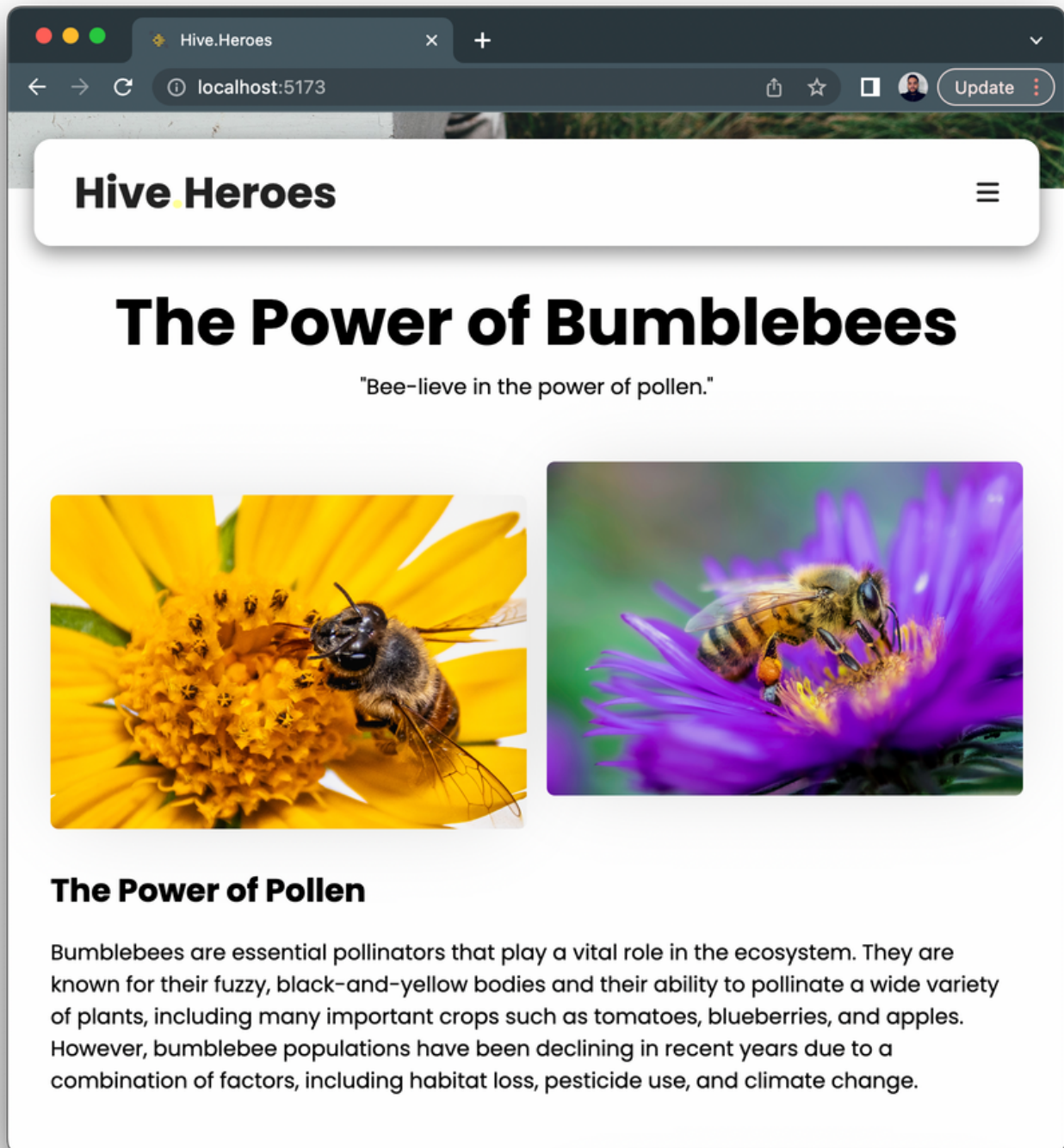


# Hive.Heroes

## Case Study



## Overview & Goal:

**Hive Heroes is a mock-up pseudo website, designed to be tailored to those who love bumblebees and have a mission to protect them, like bumblebee conservations. The website features different pages of what a Home page, About page, Services page, and a Contact page could look like. The main focus of this website was to display a portion of layout and design skills, while majority of this project was to show how dynamic and reusable React components can be. Using Reacts dynamic and reusable components, I was able to make multiple pages on this website with minimal coding involved, without having to make adjustments to each individual page.**

**A React component is a piece of independent and reusable code, similar to JavaScript functions. Components are the building blocks of a React application, and they are created and rendered based on requirements. In React, a component either holds data (has a state), or receives data through props. This website specifically will make use of props in a data file, while the components will hold the values that can be changed around without having to touch the main components or route pages themselves.**

## Goals:

- **Learn more about Reacts dynamic components using props.**
- **Showcase the ability to learn and grow in new skills.**
- **Showcase to small businesses and other institutions that I could design a website for their goals and vision.**

## Role:

- **Full-Stack Web Developer**

## Tools:

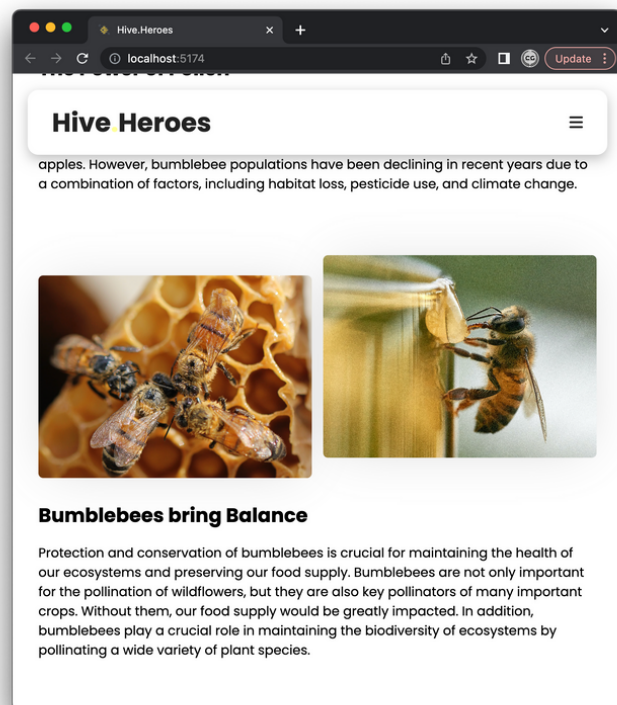
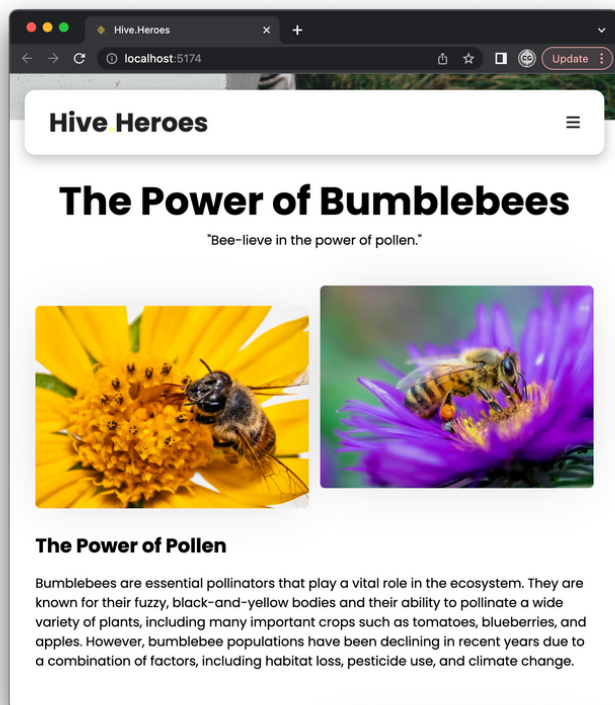
- **React**
- **React Components**
- **Props**

## Approach:

A React component is a piece of independent and reusable code, similar to JavaScript functions. Components are the building blocks of a React application, and they are created and rendered based on requirements. In React, a component either holds data (has a state), or receives data through props. This website specifically will make use of props in a data file, while the components will hold the values that can be changed around without having to touch the main components or route pages themselves.

Props are a short form for properties, and they simply refer to the internal data of a React component. Props can be passed as attributes to components in JSX, and we receive the props as arguments within the component, just like parameters and arguments passed to a function in JavaScript. Passing props to components is an excellent way to pass values around in an application dynamically.

Take note of the layout you see below, which is on our Home page of our application. Both of them consisting two images, a header, and a paragraph displaying different content.



```

3 import Cards from "../components/Cards";
4 import { NavBarView } from "../components/Navbar";
5 import Footer from "../components/Footer";
6 import HomeImg from "../assets/bee8.jpg";
7
8 function Home() {
9   return (
10     <
11       <NavbarView />
12       <Hero
13         cName='hero'
14         heroImg={HomeImg}
15         title='Join the movement and become a HiveHero today!'
16         text='Hiving a sweet-life, one flower at a time.'
17         buttonText='Learn More'
18         url='/'
19         btnClass='show'
20       />
21       <Info />
22       <Cards />
23       <Footer />
24     </>
25   )
26 }

```

This is the code snippet for our Home page. The **<Info /> component** is what's shown above. This component consists of two images, a header, and a paragraph.

The two pieces of code below show how amazing React's dynamic components can be used. The left image is our **<Info />** component that sits in our Home Page. The right image is our **<InfoData />** component that sits outside our Home Page but is utilized within our **<Info />** component. If you noticed, our **<InfoData />** is using props to call the data found within our **<Info />** component. This means we can apply CSS styling in **<InfoData />** and the stylings will follow no matter how you change the properties of the **<InfoData />** inside the **<Info />** component. This component is dynamic because you can change the properties found in **<InfoData />** function that sits in the **<Info />** component without having to mess with the main data or layout. You can even reuse the **<InfoData />** function multiple times, like seen below, and the data file keeps the original design and data.

```

react-bees-site > src > components > Info.jsx > Info
1 import Bumblebee1 from "../assets/bee1.jpg";
2 import Bumblebee2 from "../assets/bee2.jpg";
3 import Bumblebee3 from "../assets/bee9.jpg";
4 import Bumblebee4 from "../assets/bee10.jpg";
5 import InfoData from "./InfoData";
6 import "../InfoStyles.css";
7
8 const Info = () => {
9   return (
10     <div className="Info">
11       <h1>The Power of Bumblebees</h1>
12       <p>"Bee-lieve in the power of pollen."</p>
13
14       <InfoData
15         className="first-info"
16         headings="The Power of Pollen"
17         text="Bumblebees are essential pollinators that play a vital role in the ecosystem. They are known for their fuzzy, black-and-yellow bodies and their ability to pollinate a wide variety of plants, including many important crops such as tomatoes, blueberries, and apples. However, bumblebee populations have been declining in recent years due to a combination of factors, including habitat loss, pesticide use, and climate change."
18         img1={Bumblebee3}
19         img2={Bumblebee4}
20       />
21
22       <InfoData
23         className="first-info-reverse"
24         heading="Bumblebees bring Balance"
25         text="Protection and conservation of bumblebees is crucial for maintaining the health of our ecosystems and preserving our food supply. Bumblebees are not only important for the pollination of wildflowers, but they are also key pollinators of many important crops. Without them, our food supply would be greatly impacted. In addition, bumblebees play a crucial role in maintaining the biodiversity of ecosystems by pollinating a wide variety of plant species."
26         img1={Bumblebee1}
27         img2={Bumblebee2}
28       />
29     </div>
30   );
31 }
32
33 export default Info;

```

```

react-bees-site > src > components > InfoData.jsx > InfoData >
1 import { Component } from 'react';
2 import Bumblebee3 from "../assets/bee9.jpg";
3 import Bumblebee4 from "../assets/bee10.jpg";
4 import "../InfoStyles.css";
5
6 class InfoData extends Component {
7   render() {
8     return (
9       <div className={this.props.className}>
10         <div className="des-text">
11           <h2>{this.props.heading}</h2>
12           <p>{this.props.text}</p>
13         </div>
14
15         <div className="image">
16           <img alt="img" src={this.props.img1}></img>
17           <img alt="img" src={this.props.img2}></img>
18         </div>
19       </div>
20     );
21   }
22 }
23
24 export default InfoData;

```

## Conclusion:

**The entire application of Hive Heroes has utilized all of its pages to consist of these reusable dynamic components using props and calling the data found within its properties. This makes the application extremely simple to make changes and additions without making any touches to the data files and main component, also making the application easily scalable.**

**I learned valuable lessons during this project that I can take with me on any future project. Learning exactly how dynamic react component can be, and wrapping my brain around using props were the difficult parts of this project, but I caught on very quickly and it began to make a lot more sense after each component I used it.**