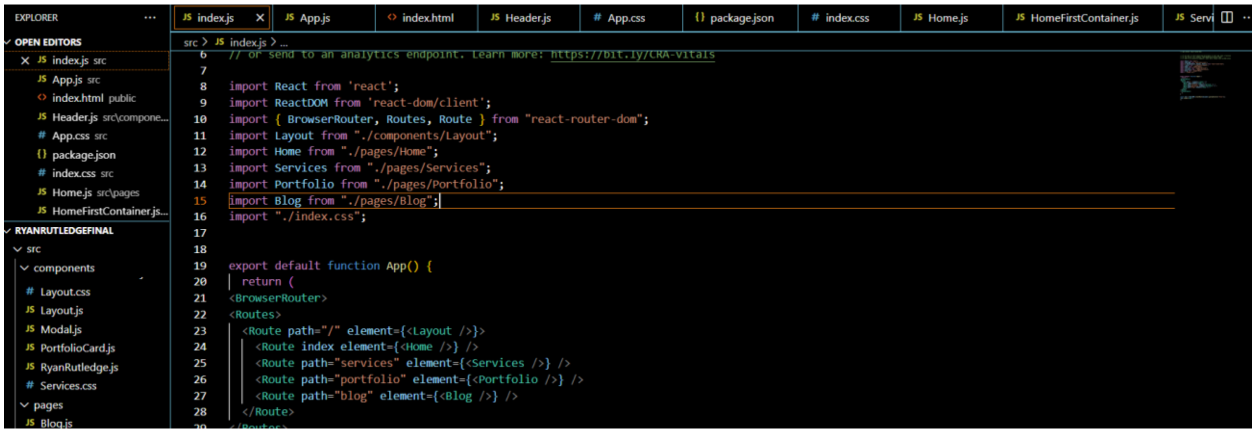
CIT 31300 Final Project

Ryan Rutledge

How I setup Page Routing:



```
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

import React from 'react';
import ReactDOM from 'react-dom/client';
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Layout from "./components/Layout";
import Home from "./pages/Home";
import Services from "./pages/Services";
import Portfolio from "./pages/Portfolio";
import Blog from "./pages/Blog";
import "./index.css";


export default function App() {
  return (
<BrowserRouter>
<Routes>
  <Route path="/" element={<Layout />}>
    <Route index element={<Home />} />
    <Route path="services" element={<Services />} />
    <Route path="portfolio" element={<Portfolio />} />
    <Route path="blog" element={<Blog />} />
  </Route>
```

In the Index.js, I imported React from react to make sure that we are receiving the necessary react application for react to work in our project. We will then import the ReactDom from React-Dom/client and the Browser Router, Routes and Route from react-router-dom in order to host multiple pages from a single source that can navigate to pages based on their element path.

Now we will need to import the pages that we plan to display on our application, in this case our Layout for Nav, Footer, and Header. Services, Home, Portfolio and Blog for our individual pages, and Index.css for formatting and our future tailwindcss.

We then setup our Routes using the Browser Router, Routes, Route function that we previously imported, and we begin planting our pages within each route with each pages specific element.

At the end we will be placing

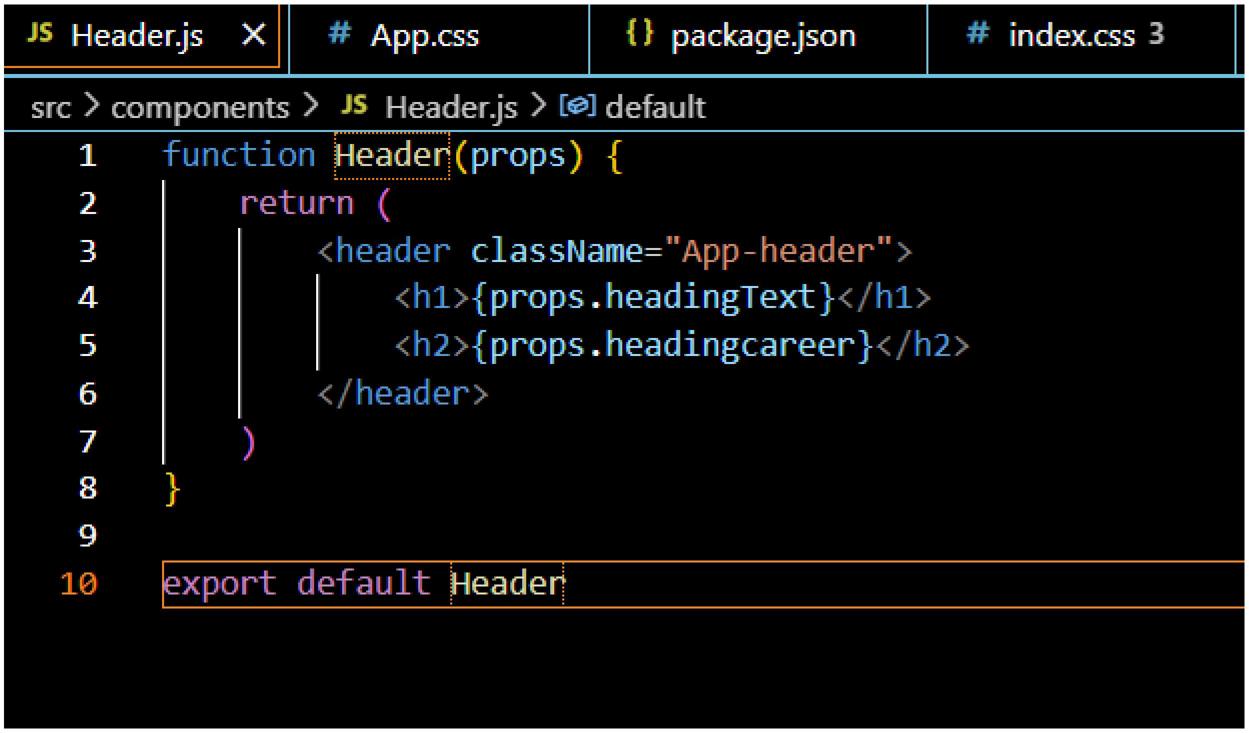
const root = ReactDOM.createRoot(document.querySelector('#root'));

root.render(<App />

);

In order to display the page for our application.

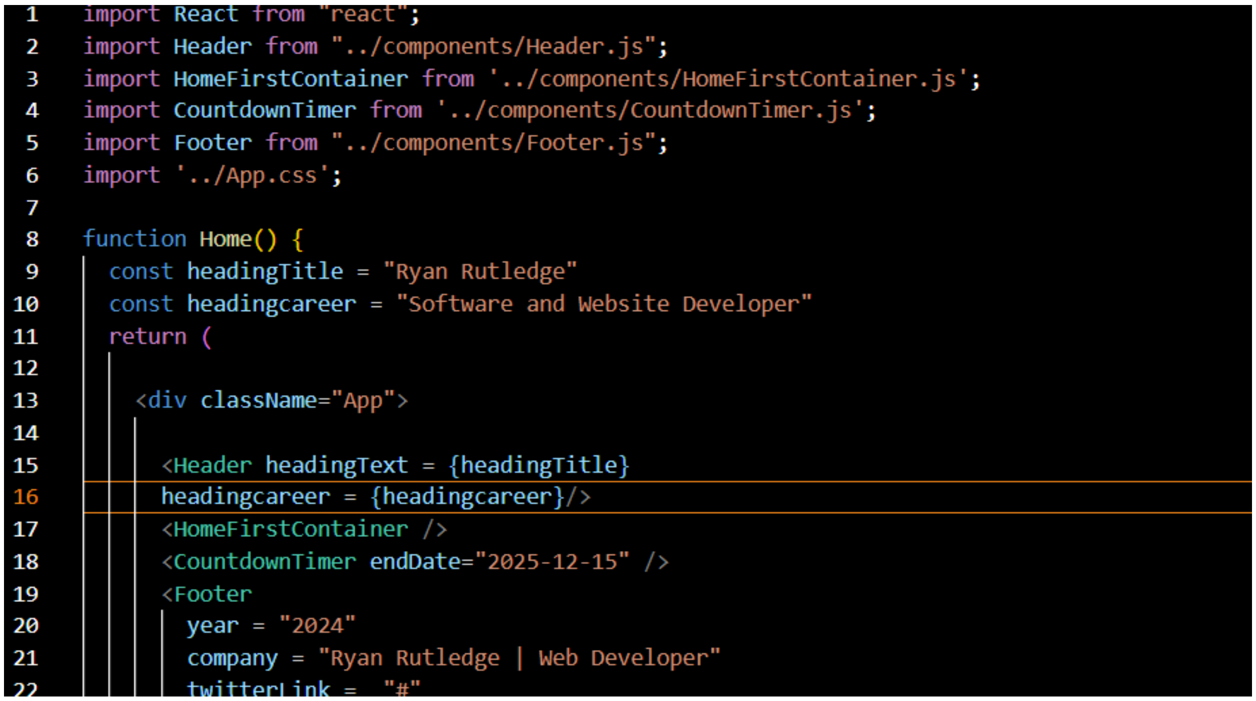
How to pass data via props:



JS Header.js × | # App.css | {} package.json | # index.css 3

src > components > JS Header.js > default

```
function Header(props) {
    return (
        <header className="App-header">
            <h1>{props.headingText}</h1>
            <h2>{props.headingcareer}</h2>
        </header>
    )
}

export default Header
```

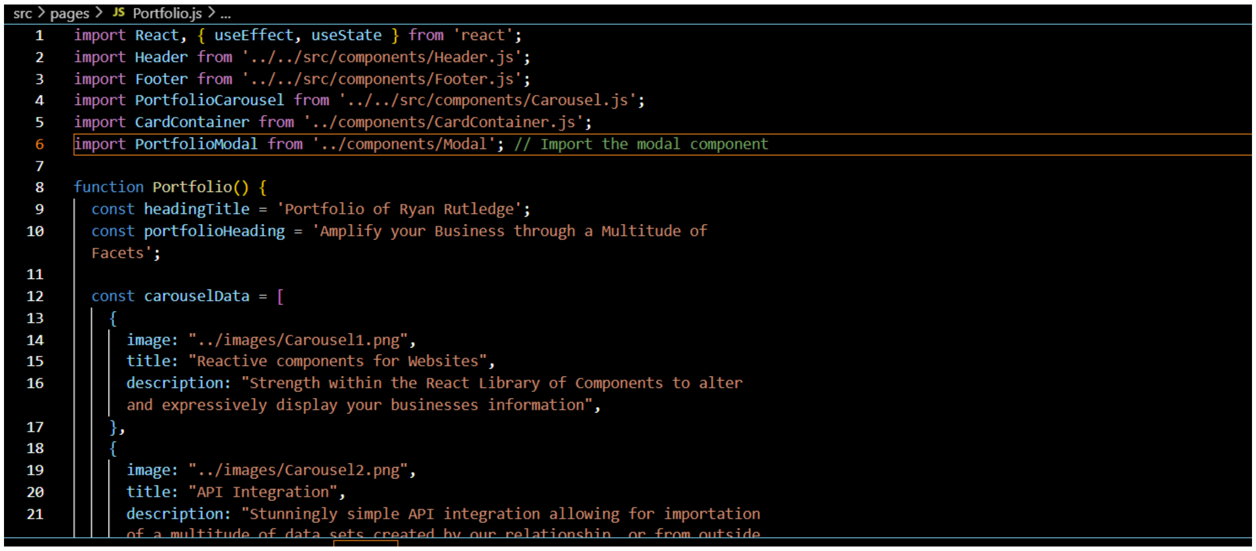


```
import React from "react";
import Header from "../components/Header.js";
import HomeFirstContainer from '../components/HomeFirstContainer.js';
import CountdownTimer from '../components/CountdownTimer.js';
import Footer from "../components/Footer.js";
import '../App.css';

function Home() {
  const headingTitle = "Ryan Rutledge"
  const headingcareer = "Software and Website Developer"
  return (

    <div className="App">

      <Header headingText = {headingTitle}
      headingcareer = {headingcareer}/>
      <HomeFirstContainer />
      <CountdownTimer endDate="2025-12-15" />
      <Footer
        year = "2024"
        company = "Ryan Rutledge | Web Developer"
        twitterLink = "#"
```

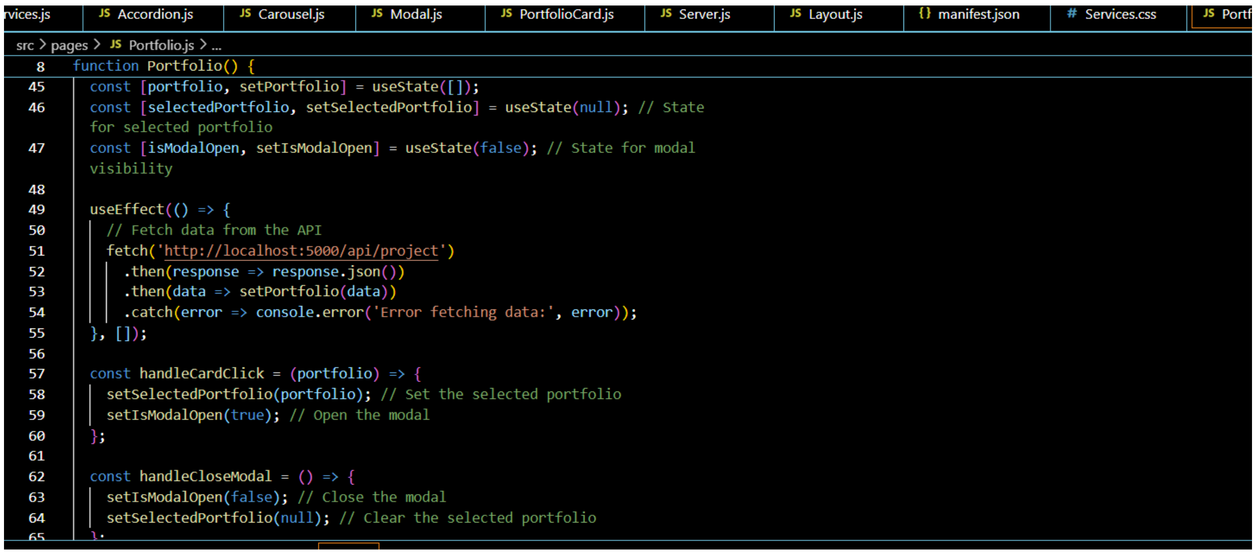
We utilize props in this project through our Header.js component, where we return a specified header for every page, but as you go to each page, we will be changing the props information that is being passed. We will be entering a headingText and headingCareer as our Title and Subtitle on each page.

How we established a React Hook and Why?

In this project we utilized the React Hook to include a Modal on the Cards that show some of the Branding and Web Development project that I have done in the years at Purdue University.


src > pages > Portfolio.js > ...

```
import React, { useEffect, useState } from 'react';
import Header from '../../src/components/Header.js';
import Footer from '../../src/components/Footer.js';
import PortfolioCarousel from '../../src/components/Carousel.js';
import CardContainer from '../components/CardContainer.js';
import PortfolioModal from '../components/Modal'; // Import the modal component

function Portfolio() {
  const headingTitle = 'Portfolio of Ryan Rutledge';
  const portfolioHeading = 'Amplify your Business through a Multitude of
  Facets';

  const carouselData = [
    {
      image: "../images/Carousel1.png",
      title: "Reactive components for Websites",
      description: "Strength within the React Library of Components to alter
      and expressively display your businesses information",
    },
    {
      image: "../images/Carousel2.png",
      title: "API Integration",
      description: "Stunningly simple API integration allowing for importation
      of a multitude of data sets created by our relationship, or from outside
```


rvices.js | Accordion.js | Carousel.js | Modal.js | PortfolioCard.js | Server.js | Layout.js | manifest.json | Services.css | Portf

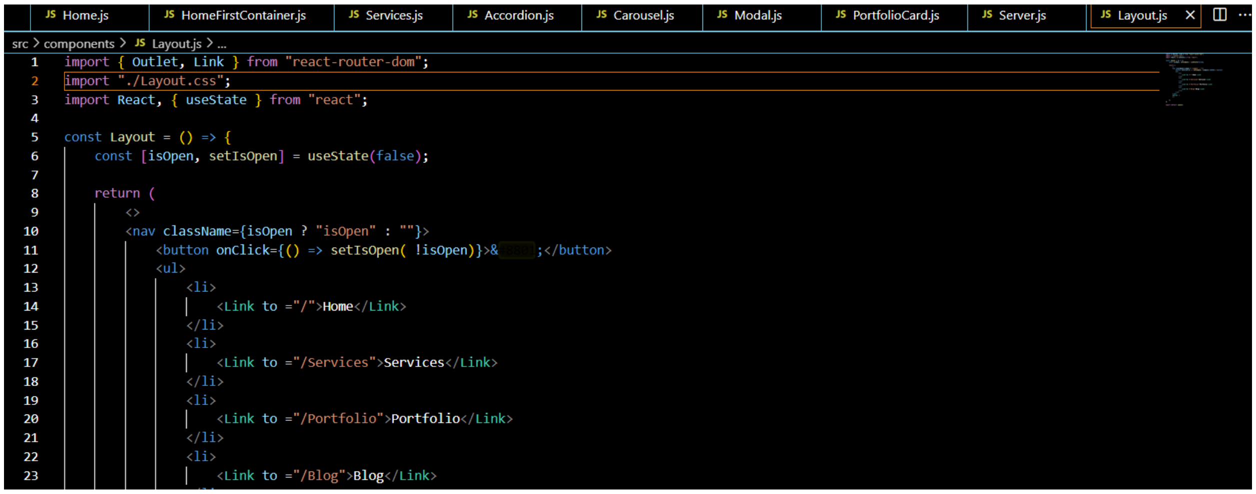
src > pages > Portfolio.js > ...

```
function Portfolio() {
  const [portfolio, setPortfolio] = useState([]);
  const [selectedPortfolio, setSelectedPortfolio] = useState(null); // State
  for selected portfolio
  const [isModalOpen, setIsModalOpen] = useState(false); // State for modal
  visibility

  useEffect(() => {
    // Fetch data from the API
    fetch('http://localhost:5000/api/project')
      .then(response => response.json())
      .then(data => setPortfolio(data))
      .catch(error => console.error('Error fetching data:', error));
  }, []);

  const handleCardClick = (portfolio) => {
    setSelectedPortfolio(portfolio); // Set the selected portfolio
    setIsModalOpen(true); // Open the modal
  };

  const handleCloseModal = () => {
    setIsModalOpen(false); // Close the modal
    setSelectedPortfolio(null); // Clear the selected portfolio
  };
```

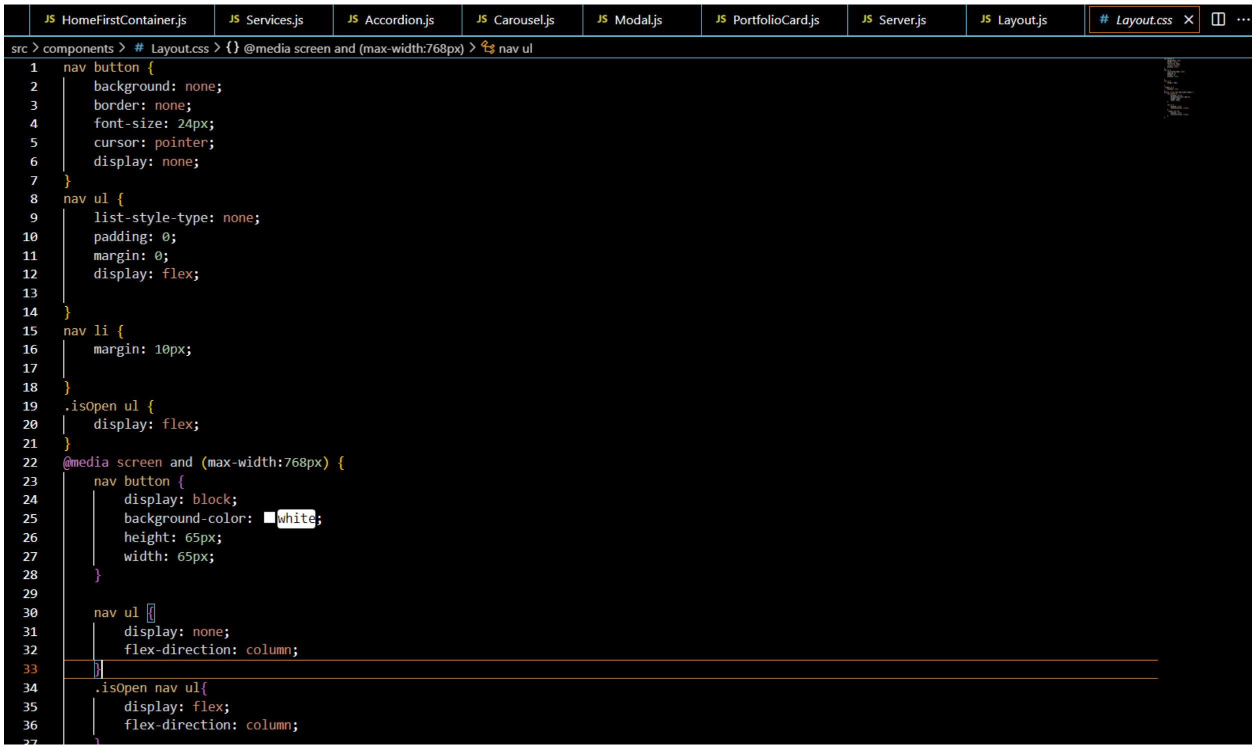
We utilize the useEffect and useState in the Portfolio.js Page for creating Modals that will be activated upon clicking on each identified card that is displayed. We have the connection between the API that we have created and the Portfolio Page and Corresponding Components. This allows us to pass data from our API into both Modal and Page. The Modal allows us to freeze any other portions of the page until the Modal is closed out. The information that is passed into the Modal allows the user to click a link and go directly to the github page of each (web) project.

How I implemented the Hamburger Menu

We implemented a Hamburger menu through our Layout.js and Layout.css files that are allowing media queries to stage at specified screenwidths. Once the screen width falls below the specified pixels, the hamburger menu will be displayed through the React-router-dom imported modules. We utilize the Outlet and link to create navigation and Usestate from React to create when the hamburger menu will be visible and when it will be opened upon clicking.



```
import { Outlet, Link } from "react-router-dom";
import "./Layout.css";
import React, { useState } from "react";

const Layout = () => {
    const [isOpen, setIsOpen] = useState(false);

    return (
        <>
        <nav className={isOpen ? "isOpen" : ""}>
            <button onClick={() => setIsOpen( !isOpen)}>&#8801;</button>
            <ul>
                <li>
                    <Link to ="/">Home</Link>
                </li>
                <li>
                    <Link to ="/Services">Services</Link>
                </li>
                <li>
                    <Link to ="/Portfolio">Portfolio</Link>
                </li>
                <li>
                    <Link to ="/Blog">Blog</Link>
```



```
nav button {
    background: none;
    border: none;
    font-size: 24px;
    cursor: pointer;
    display: none;
}
nav ul {
    list-style-type: none;
    padding: 0;
    margin: 0;
    display: flex;

}
nav li {
    margin: 10px;

}
.isOpen ul {
    display: flex;
}
@media screen and (max-width:768px) {
    nav button {
        display: block;
        background-color: white;
        height: 65px;
        width: 65px;
    }

    nav ul {
        display: none;
        flex-direction: column;
    }
    .isOpen nav ul{
        display: flex;
        flex-direction: column;
```

How components are Related

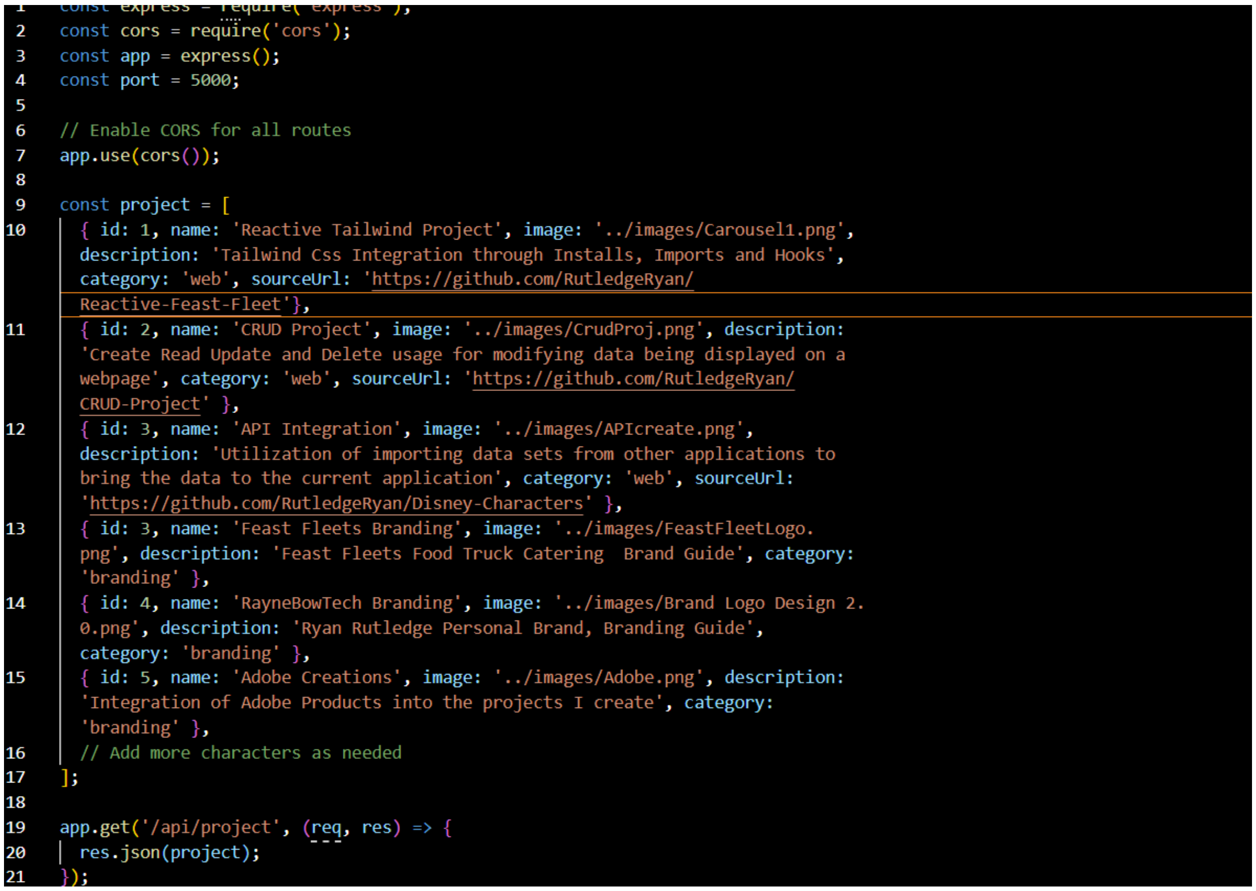
I created components to create sections within each page, and to bring in specialized components that will display data such as the accordion, carousel or cards that are in the project. We also have our header, footer, and Layout components within this section. We utilize the accordion component in the Services page to display all the different forms of digital creation that I have learned and am able to provide for clients. We utilize the Cards and Accordion to display specific projects and give descriptions of the main components that were the focal points of that specific project.

Styling and Difficulties

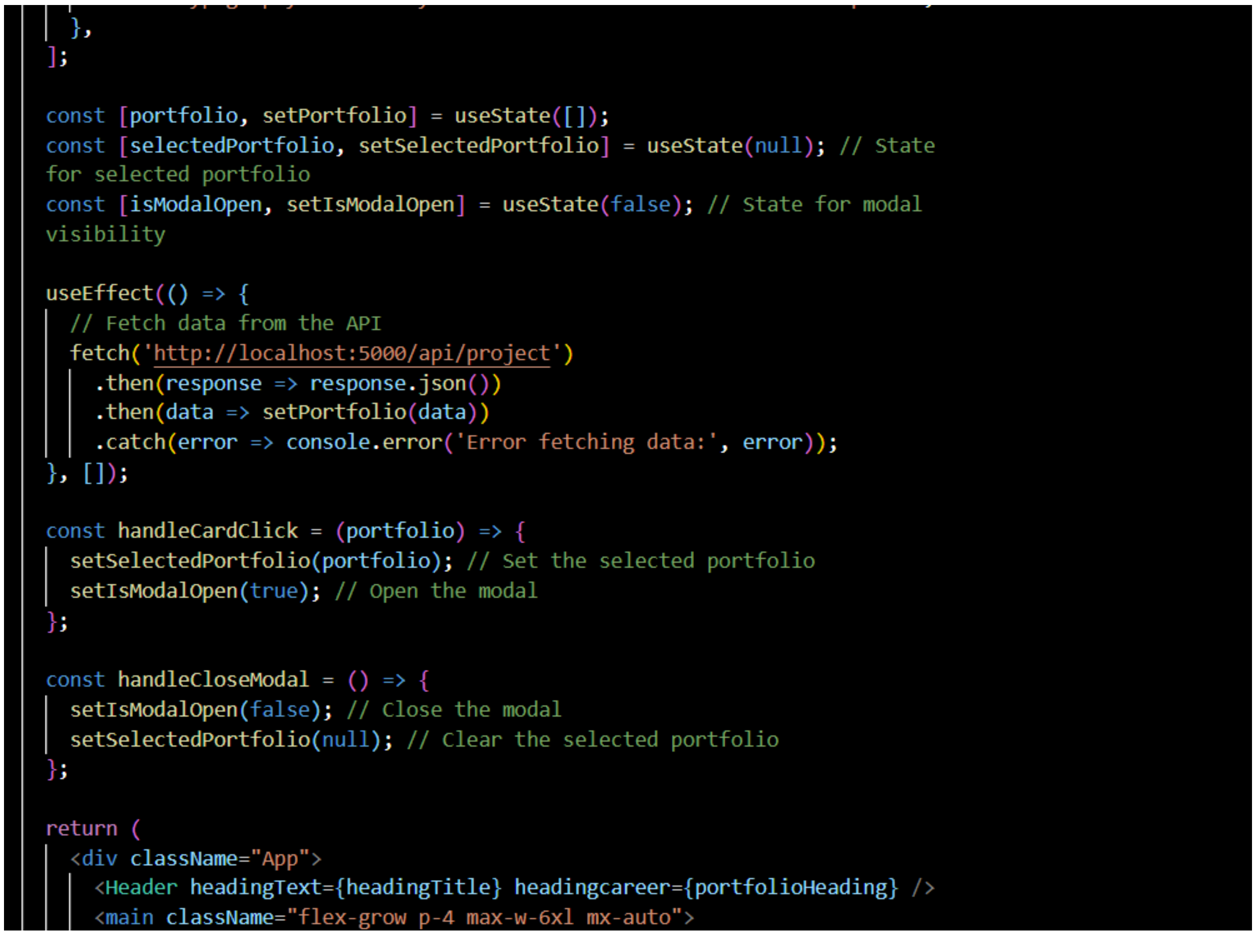
I unfortunately ran into a huge issue with utilizing my typical css styling techniques and the overwriting that my Tailwindcss created within my application. Prior to bringing in my Tailwindcss imports in my index.js file, my css for each page or component would alter the information that I wanted in the correct styling. Once I imported the Tailwindcss, my css files no longer created styling for my components and pages and the Tailwindcss was applied to all pages and components. This was a large speedbump in my project, and I eventually decided that I would need to only utilize Tailwindcss in order to complete more of the requirements of the project. This was a difficult decision, but I need my accordion, cards, and carousel to function correctly for completion.

EXTRA FEATURES

In this application I wanted to be able to create my own API to bring data into the components of my Portfolio Page. I create a Server.js File that I used Express to create a JSON style data set. I created the JSON data within the Server.js File, and began setting up the data on a separate localhost from the main project, and referenced the localhost API I created to bring data into the Portfolio Components through references the data sets.



```javascript
const express = require('express');
const cors = require('cors');
const app = express();
const port = 5000;

// Enable CORS for all routes
app.use(cors());

const project = [
  { id: 1, name: 'Reactive Tailwind Project', image: '../images/Carousel1.png',
  description: 'Tailwind Css Integration through Installs, Imports and Hooks',
  category: 'web', sourceUrl: 'https://github.com/RutledgeRyan/
  Reactive-Feast-Fleet'},
  { id: 2, name: 'CRUD Project', image: '../images/CrudProj.png', description:
  'Create Read Update and Delete usage for modifying data being displayed on a
  webpage', category: 'web', sourceUrl: 'https://github.com/RutledgeRyan/
  CRUD-Project' },
  { id: 3, name: 'API Integration', image: '../images/APIcreate.png',
  description: 'Utilization of importing data sets from other applications to
  bring the data to the current application', category: 'web', sourceUrl:
  'https://github.com/RutledgeRyan/Disney-Characters' },
  { id: 3, name: 'Feast Fleets Branding', image: '../images/FeastFleetLogo.
  png', description: 'Feast Fleets Food Truck Catering  Brand Guide', category:
  'branding' },
  { id: 4, name: 'RayneBowTech Branding', image: '../images/Brand Logo Design 2.
  0.png', description: 'Ryan Rutledge Personal Brand, Branding Guide',
  category: 'branding' },
  { id: 5, name: 'Adobe Creations', image: '../images/Adobe.png', description:
  'Integration of Adobe Products into the projects I create', category:
  'branding' },
  // Add more characters as needed
];

app.get('/api/project', (req, res) => {
  res.json(project);
});
```



```
    },
  ];

  const [portfolio, setPortfolio] = useState([]);
  const [selectedPortfolio, setSelectedPortfolio] = useState(null); // State
  for selected portfolio
  const [isModalOpen, setIsModalOpen] = useState(false); // State for modal
  visibility

  useEffect(() => {
    // Fetch data from the API
    fetch('http://localhost:5000/api/project')
      .then(response => response.json())
      .then(data => setPortfolio(data))
      .catch(error => console.error('Error fetching data:', error));
  }, []);

  const handleCardClick = (portfolio) => {
    setSelectedPortfolio(portfolio); // Set the selected portfolio
    setIsModalOpen(true); // Open the modal
  };

  const handleCloseModal = () => {
    setIsModalOpen(false); // Close the modal
    setSelectedPortfolio(null); // Clear the selected portfolio
  };

  return (
    <div className="App">
      <Header headingText={headingTitle} headingcareer={portfolioHeading} />
      <main className="flex-grow p-4 max-w-6xl mx-auto">
```