

Front End

Class 15

October 6, 2021

Agenda

- Kahoot
- Review
- Studio

Single-Page Application Routing

Multiple Pages In Single-Page Apps

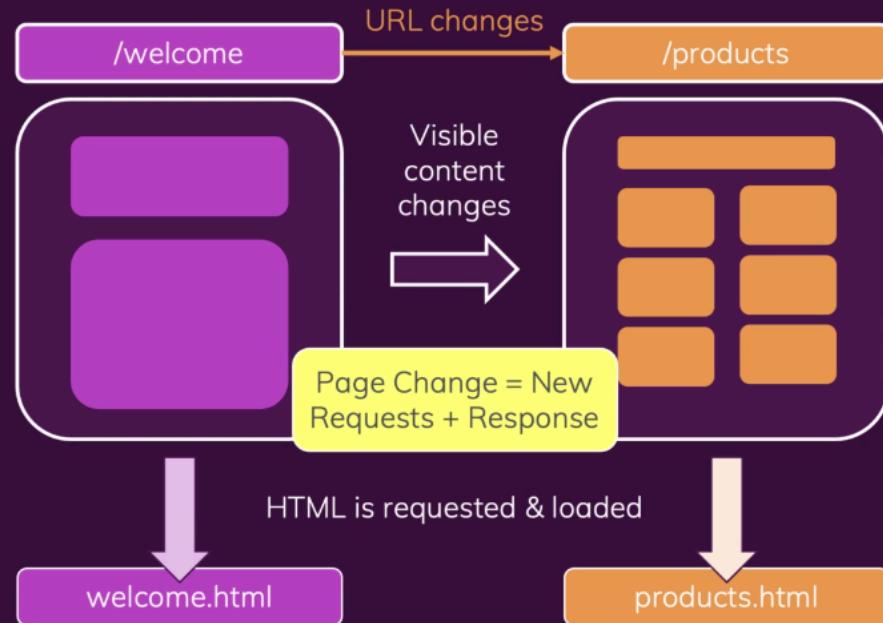


Module Content

What is Client-Side “Routing”?

Using React-Router

Multi-Page Routing





Building SPAs

When building complex user interfaces, we typically build **Single Page Applications (SPAs)**

Only one initial HTML request & response

Page (URL) changes are then handled by client-side (React) code

Changes the visible content without fetching a new HTML file

```
react-complete-guide $ npm install react-router-dom  
((.....)) .. idealTree:react-complete-guide: sill idealTree buildDeps
```

<https://www.npmjs.com/package/react-router>

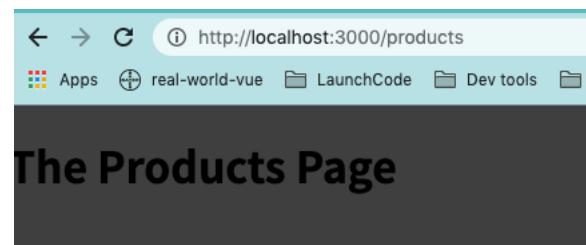
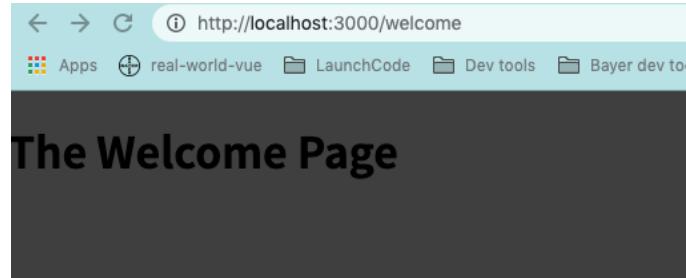
```
src > js App.js > ...
1   import { Route } from 'react-router-dom';
2   import Products from './components/Products';
3   import Welcome from './components/Welcome';
4   function App() {
5     return (
6       <div>
7         <Route path="/welcome">
8           <Welcome />
9         </Route>
10        <Route path="/products">
11          <Products />
12        </Route>
13      </div>
14    );
15  }
16
17  export default App;
18  |
```

```
import ReactDOM from 'react-dom';
import { BrowserRouter } from 'react-router-dom';
import './index.css';
import App from './App';

ReactDOM.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>,
  document.getElementById('root')
);
```

```
1 const Products = () => {
2   return <h1>The Products Page</h1>;
3 };
4 export default Products;
5
```

```
const Welcome = () => {
  return <h1>The Welcome Page</h1>;
};
export default Welcome;
```



Working with Links

```
const MainHeader = () => {
  return (
    <header>
      <nav>
        <ul>
          <li>
            <a href="/welcome">Welcome</a>
          </li>
          <li>
            <a href="/products">Products</a>
          </li>
        </ul>
      </nav>
    </header>
  );
};

export default MainHeader;
```

```
import { Route } from 'react-router-dom';
import MainHeader from './components/MainHeader';
import Products from './components/pages/Products';
import Welcome from './components/pages/Welcome';
function App() {
  return (
    <div>
      <MainHeader />
      <Route path="/welcome">
        <Welcome />
      </Route>
      <Route path="/products">
        <Products />
      </Route>
    </div>
  );
}

export default App;
```

Responsive ▾ 1840 × 442 79% ▾ No throttling ▾

- Welcome
- Products

The Welcome Page

Elements Sources Console Network Performance Memory Security Lighthouse Application Augury Components Profiler

Preserve log Disable cache No throttling

Request	Status	Type	Initiator	Size	Time	Waterfall
welcome.js	200	script	welcome	7.4 kB	10 ms	
0.chunk.js	200	script	welcome	433 kB	101 ms	
main.chunk.js	200	script	welcome	3.8 kB	10 ms	
react_devtools_backend.js	200	script	injectGlobalHook.js:1664	476 kB	13 ms	
css2?family=Noto+Sans+JP:wght@400;700&display=swap	200	stylesheet	welcome	61.0 kB	65 ms	
sockjs-node	101	websocket	webpackHotDevClient.js:60	0 B	Pending	
-F62fjtqLzI2JPCgQBnw7HFowwII2lcnk-AFfrgQrvWXpdFg3KXxAMsKMbdN.119.woff2	200	font	css2?family=Noto+Sans+JP:wght@400;700&display=...	51.2 kB	48 ms	
-F62fjtqLzI2JPCgQBnw7HFowwII2lcnk-AFfrgQrvWXpdFg3KXxAMsKMbdN.118.woff2	200	font	css2?family=Noto+Sans+JP:wght@400;700&display=...	13.5 kB	25 ms	
-F6pfjtqLzI2JPCgQBnw7HFQei0q131nj-pXANNwpfqCt9pay6XBdsAJNhVEwQ.119.woff2	200	font	css2?family=Noto+Sans+JP:wght@400;700&display=...	52.8 kB	47 ms	
-F6pfjtqLzI2JPCgQBnw7HFQei0q131nj-pXANNwpfqCt9pay6XBdsAJNhVEwQ.118.woff2	200	font	css2?family=Noto+Sans+JP:wght@400;700&display=...	13.6 kB	33 ms	
ng-validate.js	200	script	content-script.js:1	127 kB	3 ms	
manifest.json	200	manifest	Other	815 B	3 ms	
logo192.png	200	png	Other	5.6 kB	6 ms	

1.2 MB transferred | 2.9 MB resources | Finish: 383 ms | DOMContentLoaded: 288 ms | Load: 362 ms

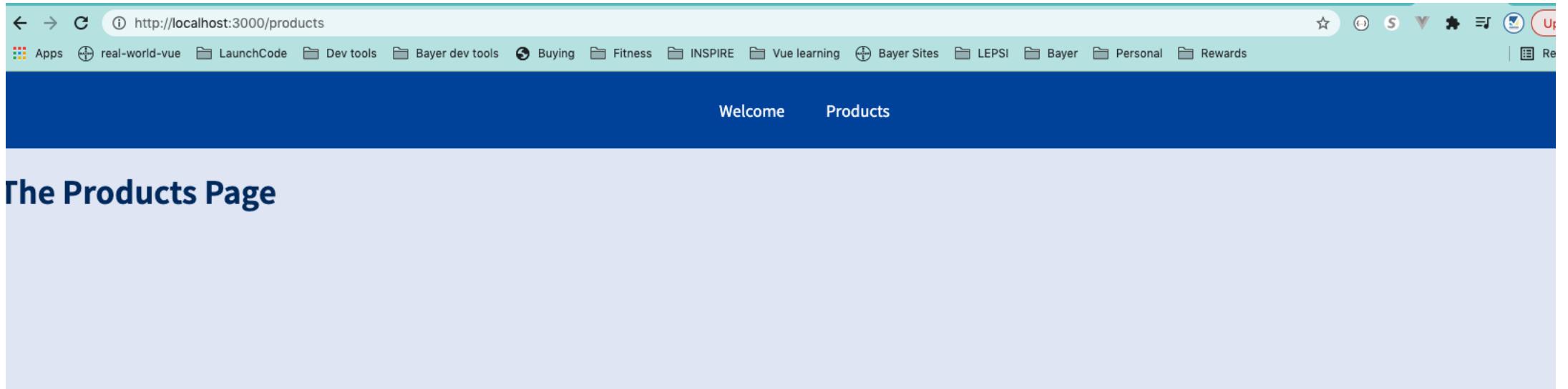
```
import { Link } from 'react-router-dom';
const MainHeader = () => {
  return (
    <header>
      <nav>
        <ul>
          <li>
            <Link to="/welcome">Welcome</Link>
          </li>
          <li>
            <Link to="/products">Products</Link>
          </li>
        </ul>
      </nav>
    </header>
  );
}
export default MainHeader;
```

React router listens for clicks to the links and create the fake navigation for us.
Does not spin. New requests are not said and don't lose state.

```
import { Link } from 'react-router-dom';
import classes from './MainHeader.module.css';

const MainHeader = () => {
  return (
    <header className={classes.header}>
      <nav>
        <ul>
          <li>
            <Link to="/welcome">Welcome</Link>
          </li>
          <li>
            <Link to="/products">Products</Link>
          </li>
        </ul>
      </nav>
    </header>
  );
};

export default MainHeader;
```

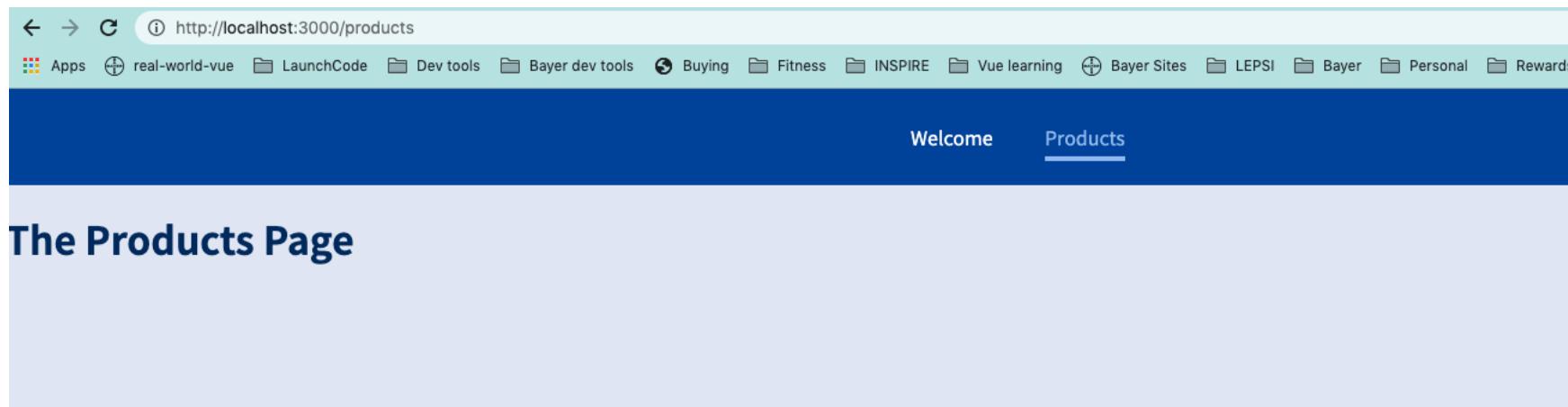


The Products Page

```
import { NavLink } from 'react-router-dom';
import classes from './MainHeader.module.css';

const MainHeader = () => {
  return (
    <header className={classes.header}>
      <nav>
        <ul>
          <li>
            <NavLink activeClassName={classes.active} to="/welcome">
              Welcome
            </NavLink>
          </li>
          <li>
            <NavLink activeClassName={classes.active} to="/products">
              Products
            </NavLink>
          </li>
        </ul>
      </nav>
    </header>
  );
};

export default MainHeader;
```

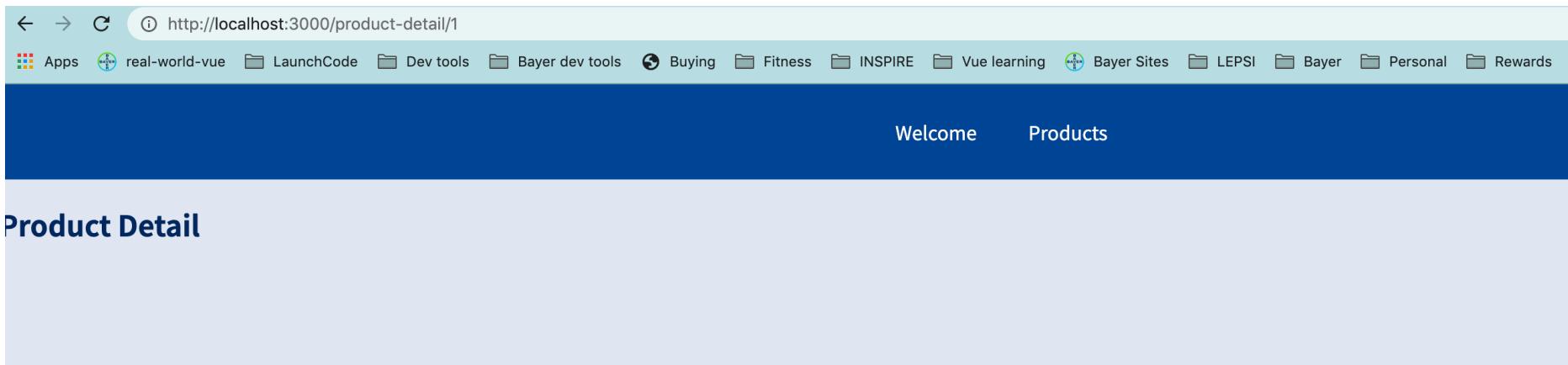


Dynamic Routes

```
return (
  <div>
    <MainHeader />
    <Route path="/welcome">
      <Welcome />
    </Route>
    <Route path="/products">
      <Products />
    </Route>
    <Route path="/product-detail/:productId">
      <ProductDetail />
    </Route>
  </div>
);
```

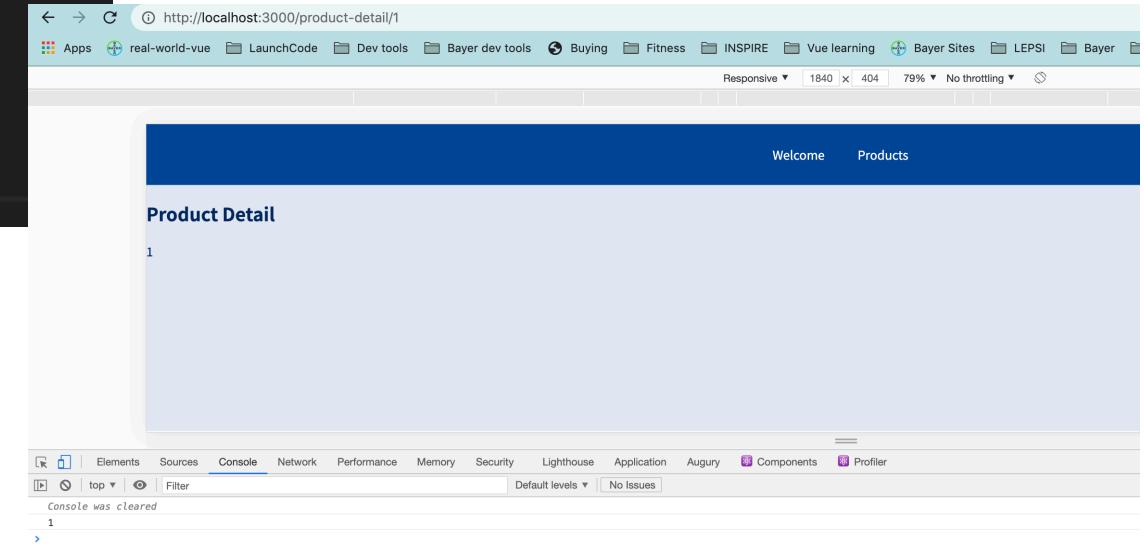
Dynamic Routes

```
1 const ProductDetail = () => {
2   return (
3     <section>
4       <h2>Product Detail</h2>
5     </section>
6   );
7 }
8 export default ProductDetail;
9 |
```



Dynamic Routes

```
import { useParams } from 'react-router';
const ProductDetail = () => {
  const params = useParams();
  console.log(params.productId);
  return (
    <section>
      <h2>Product Detail</h2>
      <p>{params.productId}</p>
    </section>
  );
}
export default ProductDetail;
```



Dynamic Routes

```
import { Link } from 'react-router-dom';
const Products = () => {
  return (
    <section>
      <h1>The Products Page</h1>
      <ul>
        <li>
          <Link to="/products/p1">A Book</Link>
        </li>
        <li>
          <Link to="/products/p2">A Carpet</Link>
        </li>
        <li>
          <Link to="/products/p3">An Online Course</Link>
        </li>
      </ul>
    </section>
  );
}

export default Products;
```

```
import { Route } from 'react-router-dom';
import MainHeader from './components/MainHeader';
import ProductDetail from './components/pages/ProductDetail';
import Products from './components/pages/Products';
import Welcome from './components/pages/Welcome';
function App() {
  return (
    <div>
      <MainHeader />
      <Route path="/welcome">
        <Welcome />
      </Route>
      <Route path="/products">
        <Products />
      </Route>
      <Route path="/products/:productId">
        <ProductDetail />
      </Route>
    </div>
  );
}

export default App;
```

Welcome Products

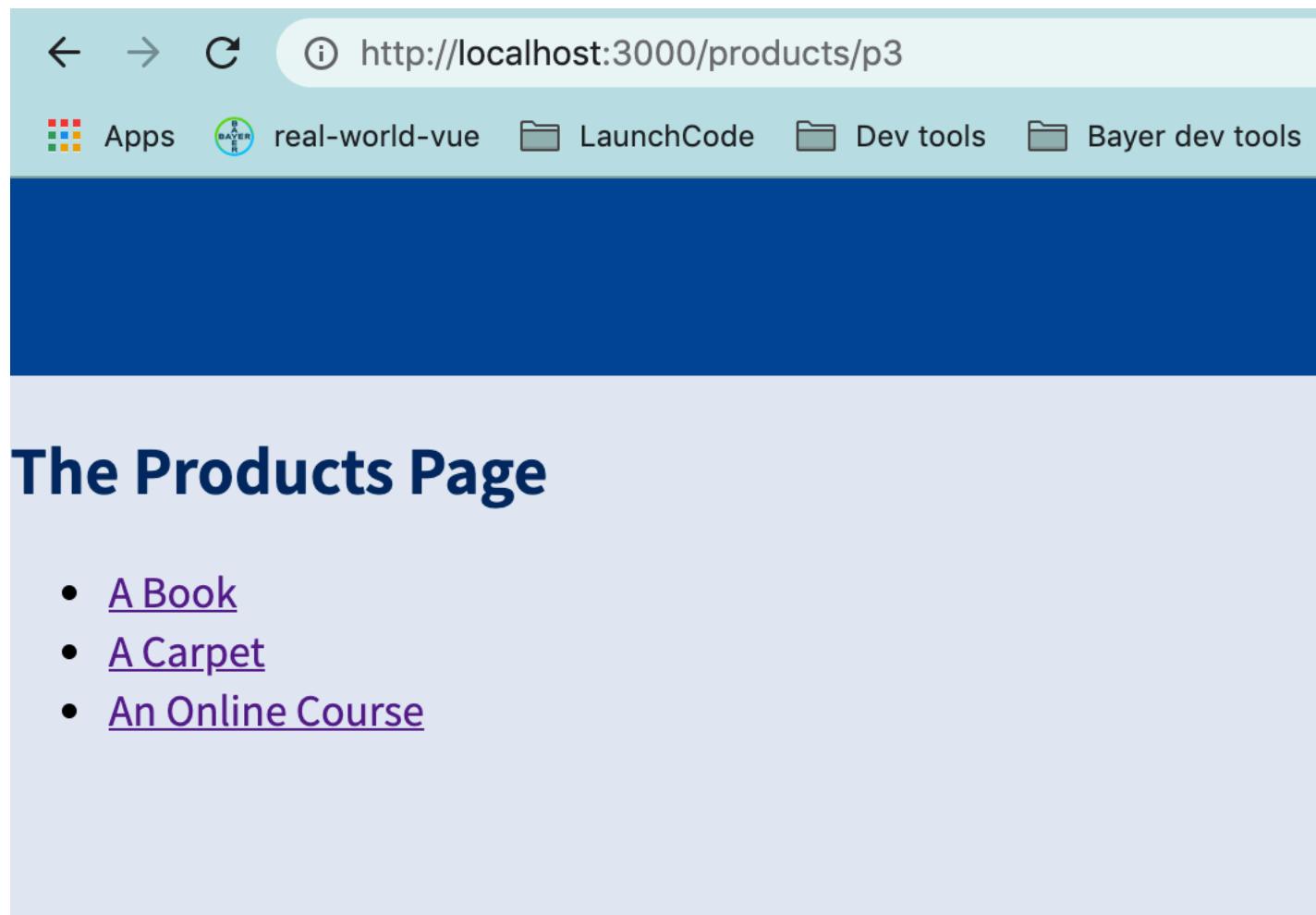
The Products Page

- [A Book](#)
- [A Carpet](#)
- [An Online Course](#)

Product Detail

Dynamic Routes

```
import { Route, Switch } from 'react-router-dom';
import MainHeader from './components/MainHeader';
import ProductDetail from './components/pages/ProductDetail';
import Products from './components/pages/Products';
import Welcome from './components/pages/Welcome';
function App() {
  return (
    <div>
      <MainHeader />
      <Switch>
        <Route path="/welcome">
          <Welcome />
        </Route>
        <Route path="/products">
          <Products />
        </Route>
        <Route path="/products/:productId">
          <ProductDetail />
        </Route>
      </Switch>
    </div>
  );
}
export default App;
```

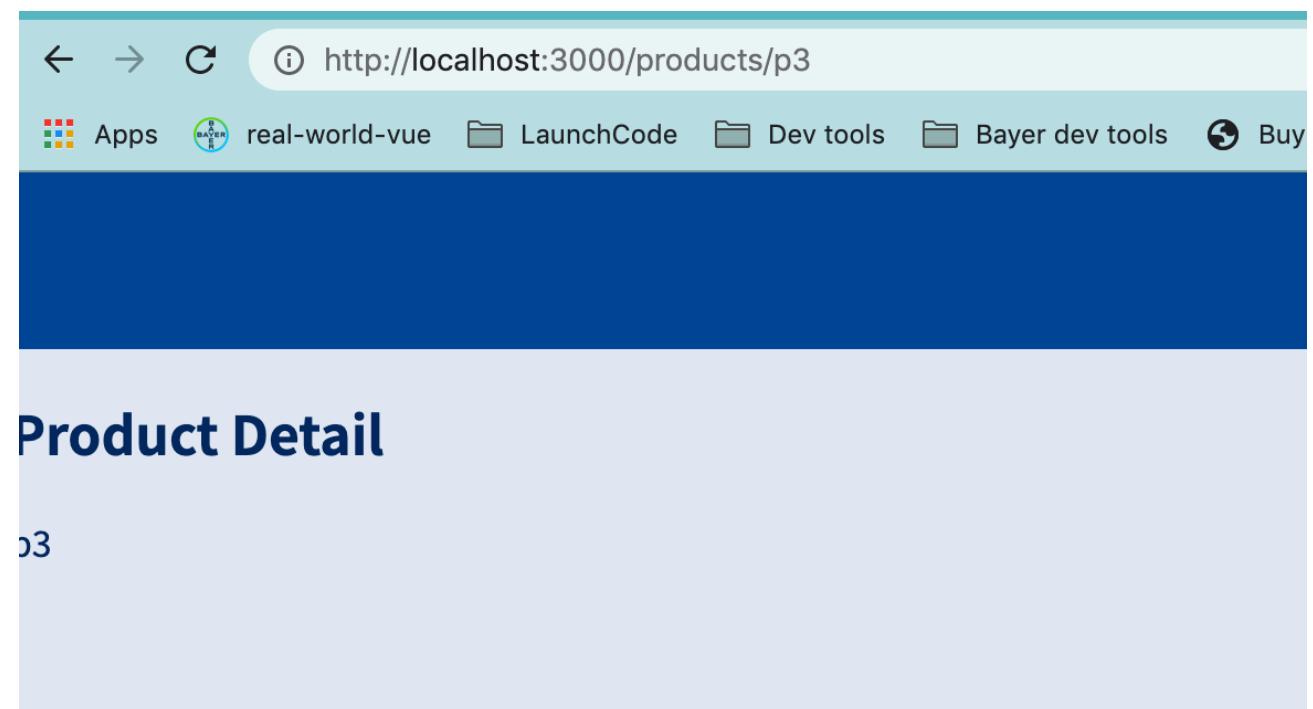


Dynamic Routes

Can change the order of the routes to order that's desired or use the exact prop.

```
import { Route, Switch } from 'react-router-dom';
import MainHeader from './components/MainHeader';
import ProductDetail from './components/pages/ProductDetail';
import Products from './components/pages/Products';
import Welcome from './components/pages/Welcome';
function App() {
  return (
    <div>
      <MainHeader />
      <Switch>
        <Route path="/welcome">
          <Welcome />
        </Route>
        <Route path="/products" exact>
          <Products />
        </Route>
        <Route path="/products/:productId">
          <ProductDetail />
        </Route>
      </Switch>
    </div>
  );
}

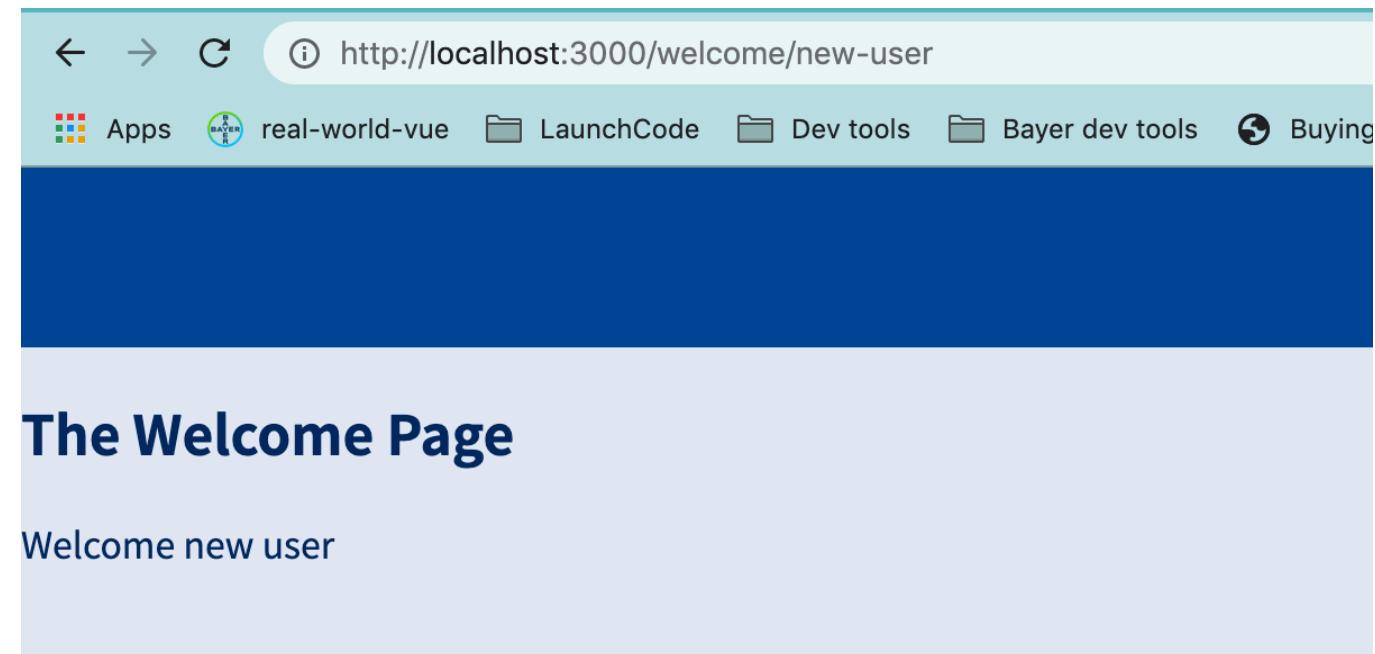
export default App;
```



Nested Routes

Requirement may have route inside pages.
Could have user-based page in welcome page.

```
import { Route } from 'react-router-dom';
const Welcome = () => {
  return (
    <section>
      <h1>The Welcome Page</h1>
      <Route path="/welcome/new-user">
        <p>Welcome new user</p>
      </Route>
    </section>
  );
}
export default Welcome;
```

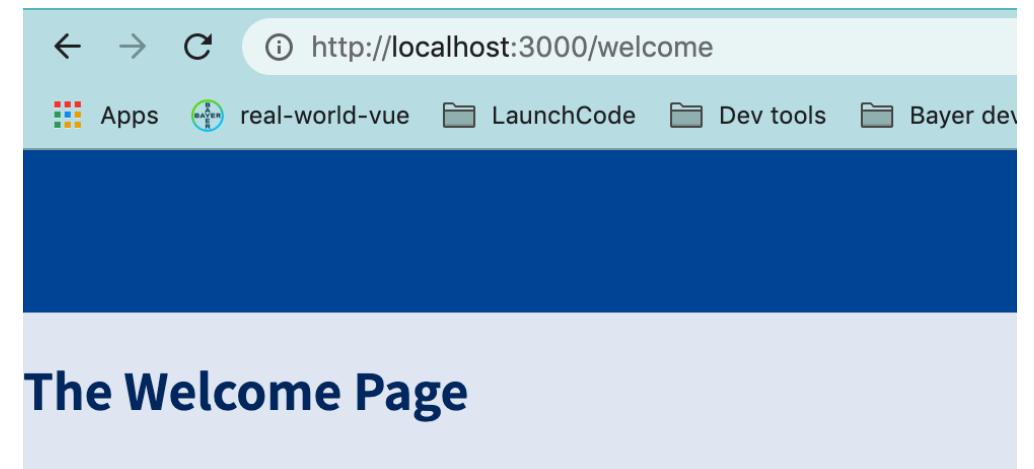


Redirecting the user

Define a route for /

```
import { Route, Switch, Redirect } from 'react-router-dom';
import MainHeader from './components/MainHeader';
import ProductDetail from './components/pages/ProductDetail';
import Products from './components/pages/Products';
import Welcome from './components/pages/Welcome';
function App() {
  return (
    <div>
      <MainHeader />
      <Switch>
        <Route path="/" exact>
          <Redirect to="/welcome" />
        </Route>
        <Route path="/welcome">
          <Welcome />
        </Route>
        <Route path="/products" exact>
          <Products />
        </Route>
        <Route path="/products/:productId">
          <ProductDetail />
        </Route>
      </Switch>
    </div>
  );
}

export default App;
```



Project

```
import { Route, Switch, Redirect } from 'react-router-dom';
import QuoteList from './components/quotes/QuoteList';
import QuoteItem from './components/quotes/QuoteItem';
import QuoteForm from './components/quotes/QuoteForm';
function App() {
  return (
    <Switch>
      <Route path="/" exact>
        <Redirect to="/quotes" />
      </Route>
      <Route path="/quotes">
        <QuoteList />
      </Route>
      <Route path="/quotes/:quoteId">
        <QuoteItem />
      </Route>
      <Route path="/new-quote">
        <QuoteForm />
      </Route>
    </Switch>
  );
}

export default App;
```



```
import { Fragment } from 'react';

import QuoteItem from './QuoteItem';
import classes from './QuoteList.module.css';

const QuoteList = (props) => {
  return (
    <Fragment>
      <h1>Quotes</h1>
      <ul className={classes.list}>
        {props.quotes.map((quote) => (
          <QuoteItem
            key={quote.id}
            id={quote.id}
            author={quote.author}
            text={quote.text}
          />
        ))}
      </ul> *
    </Fragment>
  );
};

export default QuoteList;
```

```
import classes from './QuoteItem.module.css';

const QuoteItem = (props) => [
  return (
    <h1>Quote Details</h1>
    // <li className={classes.item}>
    //   <figure>
    //     <blockquote>
    //       <p>{props.text}</p>
    //     </blockquote>
    //     <figcaption>{props.author}</figcaption>
    //   </figure>
    //   <a className='btn'>
    //     View Fullscreen
    //   </a>
    // </li>
  );
];

export default QuoteItem;
```

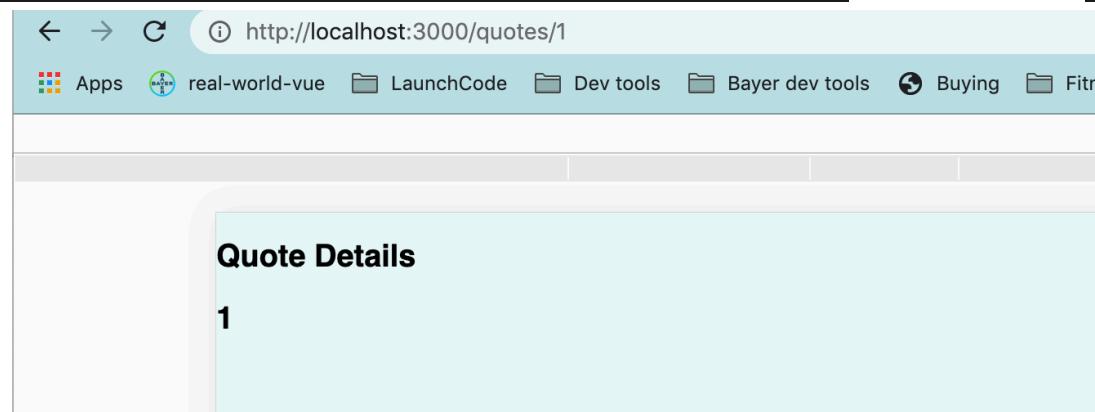
Project

```
import { Route, Switch, Redirect } from 'react-router-dom';
import QuoteList from './components/quotes/QuoteList';
import QuoteItem from './components/quotes/QuoteItem';
import QuoteForm from './components/quotes/QuoteForm';
function App() {
  return (
    <Switch>
      <Route path="/" exact>
        <Redirect to="/quotes" />
      </Route>
      <Route path="/quotes" exact>
        | <QuoteList />
      </Route>
      <Route path="/quotes/:quoteId">
        | <QuoteItem />
      </Route>
      <Route path="/new-quote">
        | <QuoteForm />
      </Route>
    </Switch>
  );
}

export default App;
```

```
import classes from './QuoteItem.module.css';
import { useParams } from 'react-router';
const QuoteItem = (props) => {
  const params = useParams();
  return (
    <section>
      <h1>Quote Details</h1>
      <h1>{params.quoteId}</h1>
    </section>
    // <li className={classes.item}>
    //   <figure>
    //     <blockquote>
    //       <p>{props.text}</p>
    //     </blockquote>
    //     <figcaption>{props.author}</figcaption>
    //   </figure>
    //   <a className='btn'>
    //     View Fullscreen
    //   </a>
    // </li>
  );
}

export default QuoteItem;
```



Project

```
import { quotes } from './quoteList';
import classes from './QuoteItem.module.css';
import { useParams, Route } from 'react-router-dom';
import CommentsList from '../comments/CommentsList';
const QuoteItem = (props) => {
  const params = useParams();
  return (
    <section>
      <h1>Quote Details</h1>
      <h1>{params.quoteId}</h1>
      <Route path={`/quotes/${params.quoteId}/comments`}*>
        <CommentsList />
      </Route>
    </section>
    // <li className={classes.item}>
    //   <figure>
    //     <blockquote>
    //       <p>{props.text}</p>
    //     </blockquote>
    //     <figcaption>{props.author}</figcaption>
    //   </figure>
    //   <a className='btn'>
    //     View Fullscreen
    //   </a>
    // </li>
  );
}

export default QuoteItem;
```

```
import classes from './QuoteItem.module.css';
import { useParams, Route } from 'react-router-dom';
import CommentsList from '../comments/CommentsList';
const QuoteItem = (props) => {
  const params = useParams();
  return (
    <section>
      <h1>Quote Details</h1>
      <h1>{params.quoteId}</h1>
      <Route path="/quotes/:quoteId/comments">
        <CommentsList />
      </Route>
    </section>
    // <li className={classes.item}>
    //   <figure>
    //     <blockquote>
    //       <p>{props.text}</p>
    //     </blockquote>
    //     <figcaption>{props.author}</figcaption>
    //   </figure>
    //   <a className='btn'>
    //     View Fullscreen
    //   </a>
    // </li>
  );
}

export default QuoteItem;
```

Project

The screenshot shows a browser window with two tabs open:

- Top Tab:** URL: <http://localhost:3000/quotes/1>. The title is "Quote Details". The content displays the number "1".
- Bottom Tab:** URL: <http://localhost:3000/quotes/1/comments>. The title is "Comments". A modal dialog box is displayed, showing a `TypeError: Cannot read property 'map' of undefined` error. The error occurred at `src/components/comments/CommentsList.js:6`. The code snippet is:

```
3 | const CommentsList = (props) => {
4 |   return (
> 5 |     <ul className={classes.comments}>
6 |       {props.comments.map((comment) => (
7 |         <CommentItem key={comment.id} text={comment.text} />
8 |       ))}
9 |   )
```

A yellow bar at the bottom right of the modal indicates "17 stack frames were collapsed". Below the modal, the `Module.<anonymous>` section of the DevTools shows:

```
4 | import './index.css';
5 | import App from './App';
```

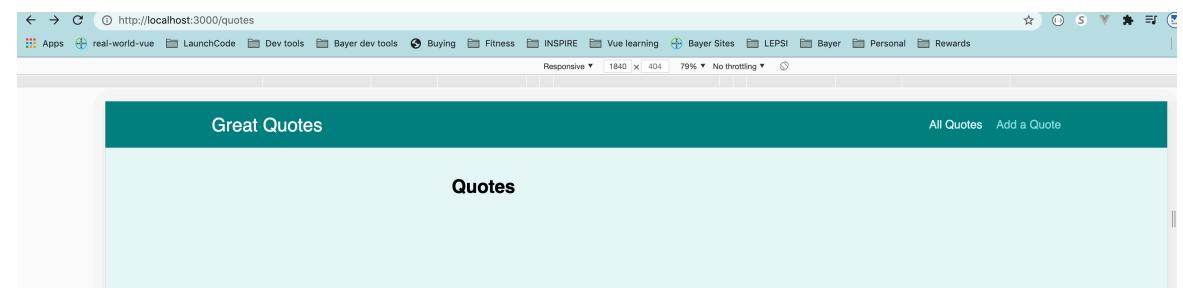
Project – Adding Layout wrapper

```
import { NavLink } from 'react-router-dom';
import classes from './MainNavigation.module.css';
const MainNavigation = () => {
  return (
    <header className={classes.header}>
      <div className={classes.logo}>Great Quotes</div>
      <nav className={classes.nav}>
        <ul>
          <li>
            <NavLink to="/quotes" activeClassName={classes.active}>
              All Quotes
            </NavLink>
          </li>
          <li>
            <NavLink to="/new-quote" activeClassName={classes.active}>
              Add a Quote
            </NavLink>
          </li>
        </ul>
      </nav>
    </header>
  );
};

export default MainNavigation;
```

```
import classes from './Layout.module.css';
import MainNavigation from './MainNavigation';
const Layout = (props) => {
  return (
    <>
      <MainNavigation />
      <main className={classes.main}>{props.children}</main>
    </>
  );
};

export default Layout;
```

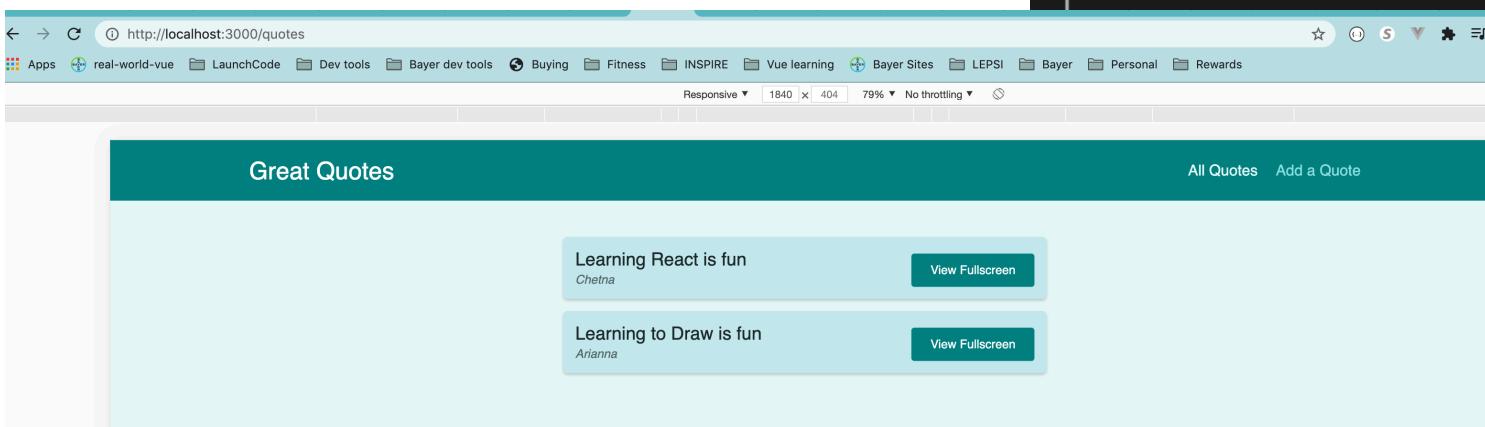


Project – Adding Dummy Data

```
1 import QuoteList from '../quotes/QuoteList';
2
3 const AllQuotes = () => [
4   const DUMMY_QUOTES = [
5     { id: 'q1', author: 'Chetna', text: 'Learning React is fun' },
6     { id: 'q2', author: 'Arianna', text: 'Learning to Draw is fun' },
7   ];
8
9   return <QuoteList quotes={DUMMY_QUOTES} />;
10 ];
11
12 export default AllQuotes;
13
```

```
import { useParams, Route } from 'react-router-dom';
import CommentsList from '../comments/CommentsList';
const QuoteDetail = () => {
  const params = useParams();
  return (
    <section>
      <h1>Quote Details</h1>
      <h1>{params.quoteId}</h1>
      <Route path="/quotes/:quoteId/comments">
        <CommentsList />
      </Route>
    </section>
  );
};

export default QuoteDetail;
```



Project – Adding Dummy Data

```
import QuoteForm from '../quotes/QuoteForm';

const NewQuote = () => {
  const addQuoteHandler = (quoteData) => {
    console.log(quoteData);
  };
  return <QuoteForm onAddQuote={addQuoteHandler} />;
};

export default NewQuote;
```

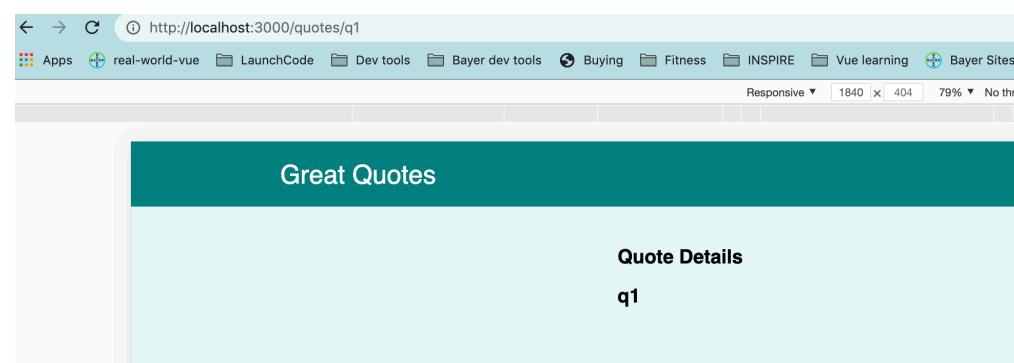
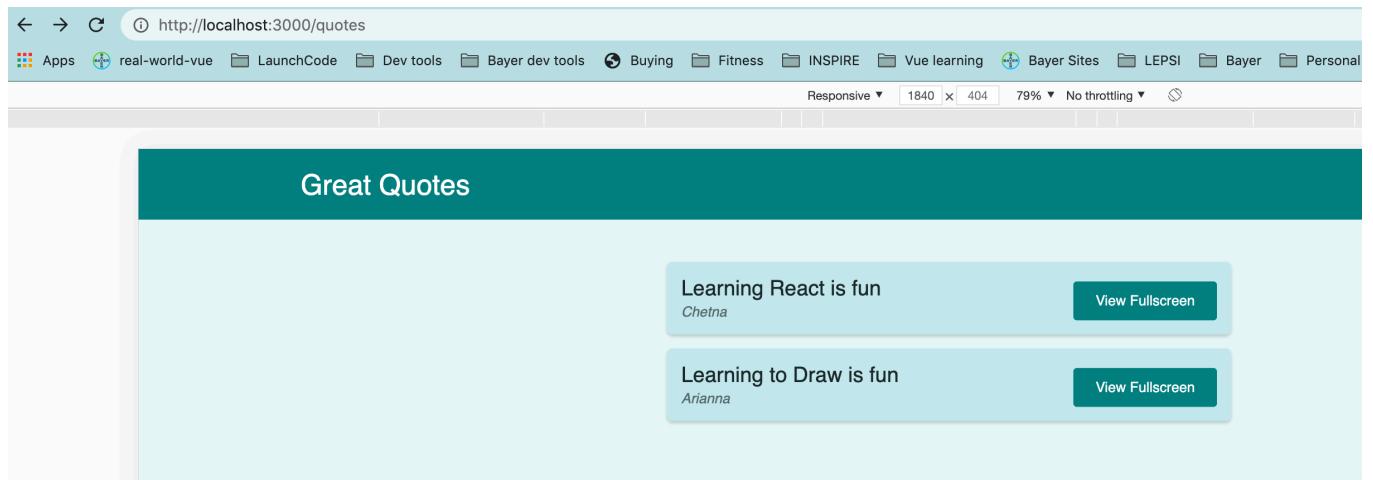
The screenshot shows a browser window with the URL `http://localhost:3000/new-quote`. The page title is "Great Quotes". On the right, there is a form with fields for "Author" (containing "aaa") and "Text" (containing "ssddd"). Below the form is a "Add Quote" button. At the bottom of the page, there is a developer tools console tab labeled "Console". The console output shows several ESLint errors:

- `src/App.js`: Line 2:8: 'QuoteList' is defined but never used no-unused-vars
- `src/App.js`: Line 3:8: 'QuoteItem' is defined but never used no-unused-vars
- `src/App.js`: Line 4:8: 'QuoteForm' is defined but never used no-unused-vars
- `src/components/pages/NewQuote.js`: Line 1:10: 'Fragment' is defined but never used no-unused-vars
- `src/components/quotes/QuoteItem.js`: Line 11:7: The href attribute is required for an anchor to be keyboard accessible. Provide a valid, navigable address as the href value. If you cannot provide an href, but still need the element to resemble a link, use a button and ch more: <https://github.com/evcohen/eslint-plugin-jsx-ally/blob/master/docs/rules/anchor-is-valid.md> jsx-ally/anchor-is-valid
- `src/components/quotes/QuoteItem.js`: Line 1:10: 'Fragment' is defined but never used no-unused-vars
- `src/components/quotes/QuoteItem.js`: Line 3:8: 'QuoteItem' is defined but never used no-unused-vars
- `src/components/quotes/QuoteItem.js`: Line 4:8: 'QuoteForm' is defined but never used no-unused-vars
- `src/components/quotes/QuoteItem.js`: Line 11:7: The href attribute is required for an anchor to be keyboard accessible. Provide a valid, navigable address as the href value. If you cannot provide an href, but still need the element to resemble a link, use a button and ch more: <https://github.com/evcohen/eslint-plugin-jsx-ally/blob/master/docs/rules/anchor-is-valid.md> jsx-ally/anchor-is-valid

Project – Outputting Data

```
import classes from './QuoteItem.module.css';
import { Link } from 'react-router-dom';
const QuoteItem = (props) => {
  return (
    <li className={classes.item}>
      <figure>
        <blockquote>
          <p>{props.text}</p>
        </blockquote>
        <figcaption>{props.author}</figcaption>
      </figure>
      <Link className="btn" to={`/quotes/${props.id}`}>
        | View Fullscreen
      </Link>
    </li>
  );
};

export default QuoteItem;
```

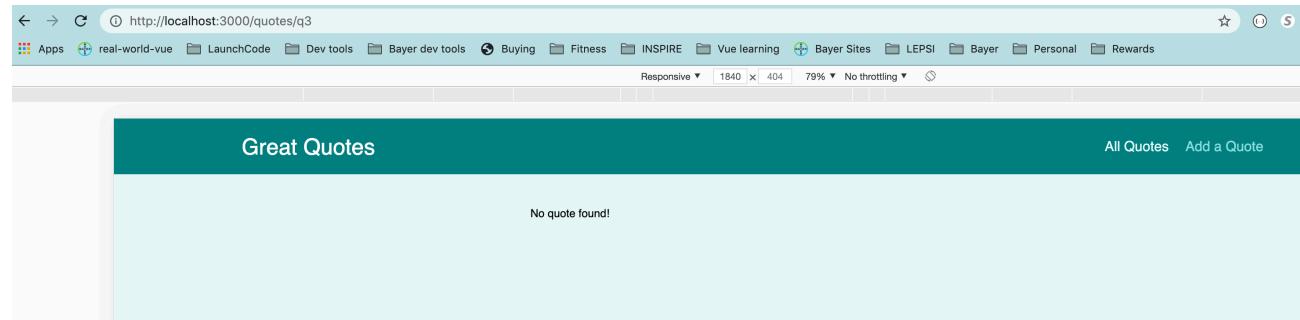
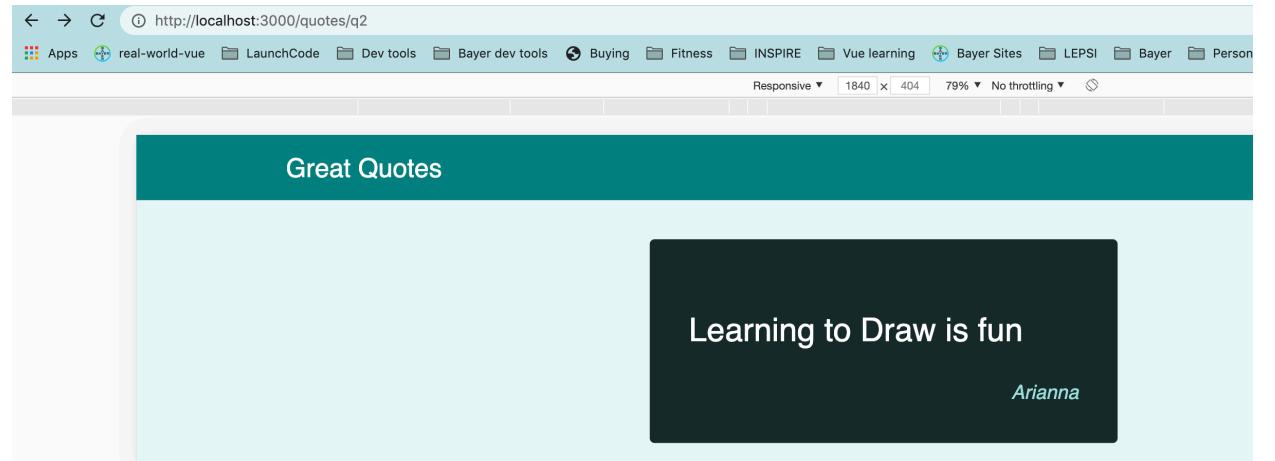


Project – Outputting Data

```
import { useParams, Route } from 'react-router-dom';
import CommentsList from '../comments/CommentsList';
import HighLightedQuote from '../quotes/HighlightedQuote';
const DUMMY_QUOTES = [
  { id: 'q1', author: 'Chetna', text: 'Learning React is fun' },
  { id: 'q2', author: 'Arianna', text: 'Learning to Draw is fun' },
];
const QuoteDetail = () => {
  const params = useParams();

  const quote = DUMMY_QUOTES.find((quote) => quote.id === params.quoteId);
  if (!quote) {
    return <p>No quote found!</p>;
  }
  return (
    <section>
      <HighLightedQuote text={quote.text} author={quote.author} />
      <Route path="/quotes/:quoteId/comments">
        <CommentsList />
      </Route>
    </section>
  );
};

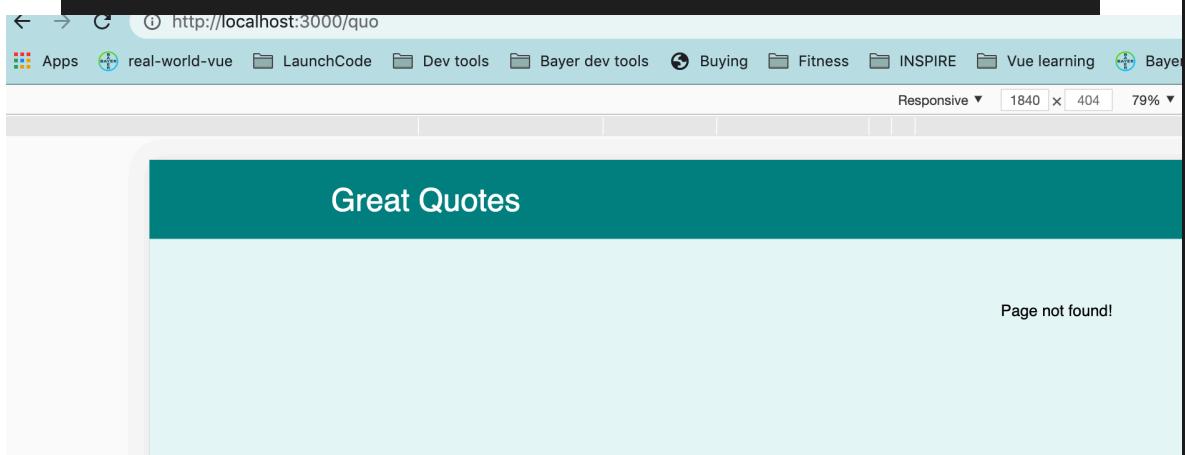
export default QuoteDetail;
```



Project – Implementing a not found page. 404 page

```
const NotFound = () => {
  return (
    <div className="centered">
      <p>Page not found!</p>
    </div>
  );
}

export default NotFound;
```



```
import { Route, Switch, Redirect } from 'react-router-dom';
import Layout from './components/layout/Layout';
import AllQuotes from './components/pages/AllQuotes';
import QuoteDetail from './components/pages/QuoteDetail';
import NewQuote from './components/pages/NewQuote';
import NotFound from './components/pages/NotFound';

function App() {
  return (
    <Layout>
      <Switch>
        <Route path="/" exact>
          <Redirect to="/quotes" />
        </Route>
        <Route path="/quotes" exact>
          <AllQuotes />
        </Route>
        <Route path="/quotes/:quoteId">
          <QuoteDetail />
        </Route>
        <Route path="/new-quote">
          <NewQuote />
        </Route>
        <Route path="*">
          <NotFound />
        </Route>
      </Switch>
    </Layout>
  );
}
```

Project – Implementing imperative navigation

Success or Navigate somewhere else when add a new quote

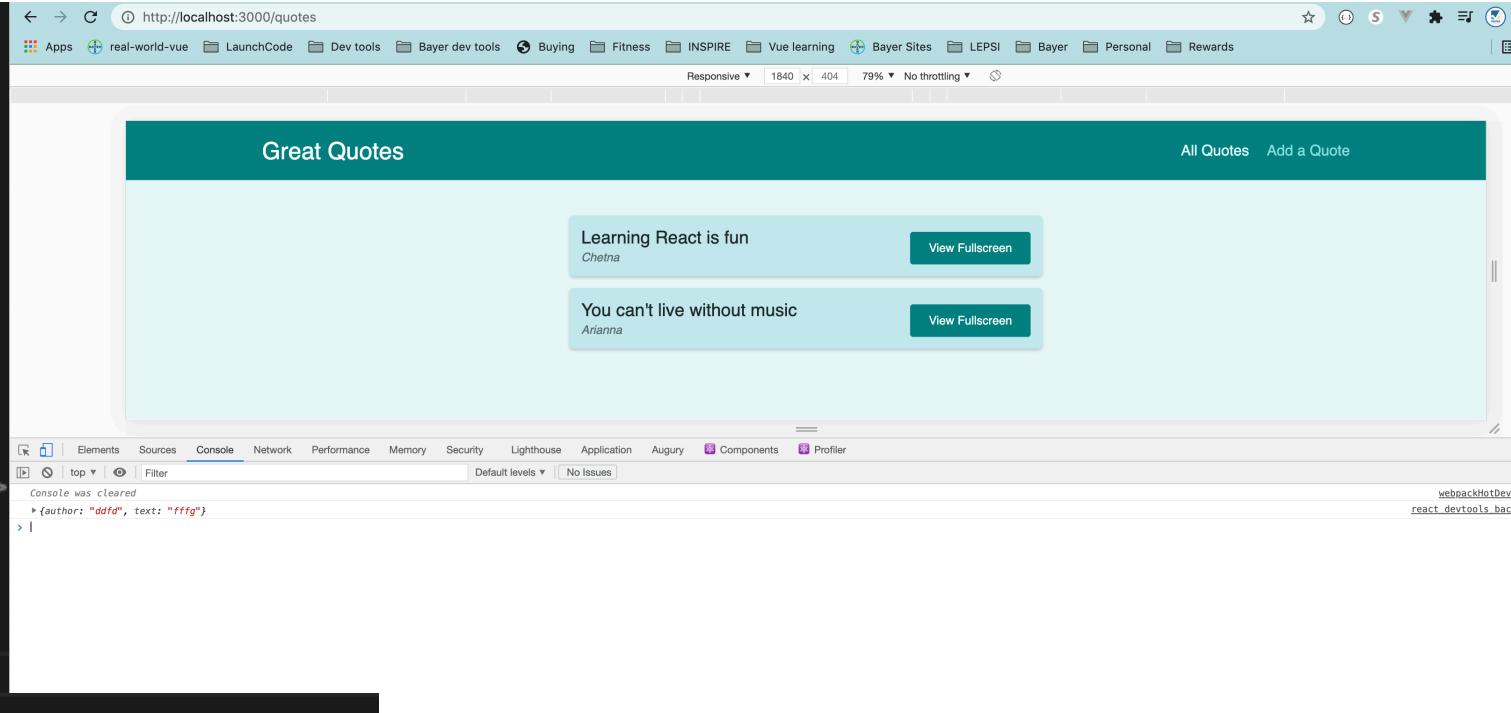
```
import QuoteForm from '../quotes/QuoteForm';
import { useHistory } from 'react-router-dom';

const NewQuote = () => {
  const history = useHistory();

  const addQuoteHandler = (quoteData) => {
    console.log(quoteData);
    history.push('/quotes');
  };

  return <QuoteForm onAddQuote={addQuoteHandler} />
};

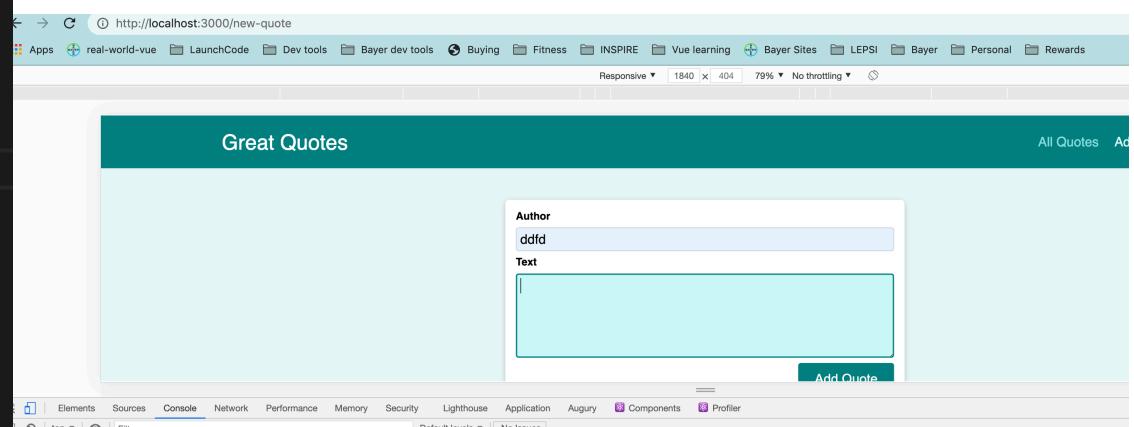
export default NewQuote;
```



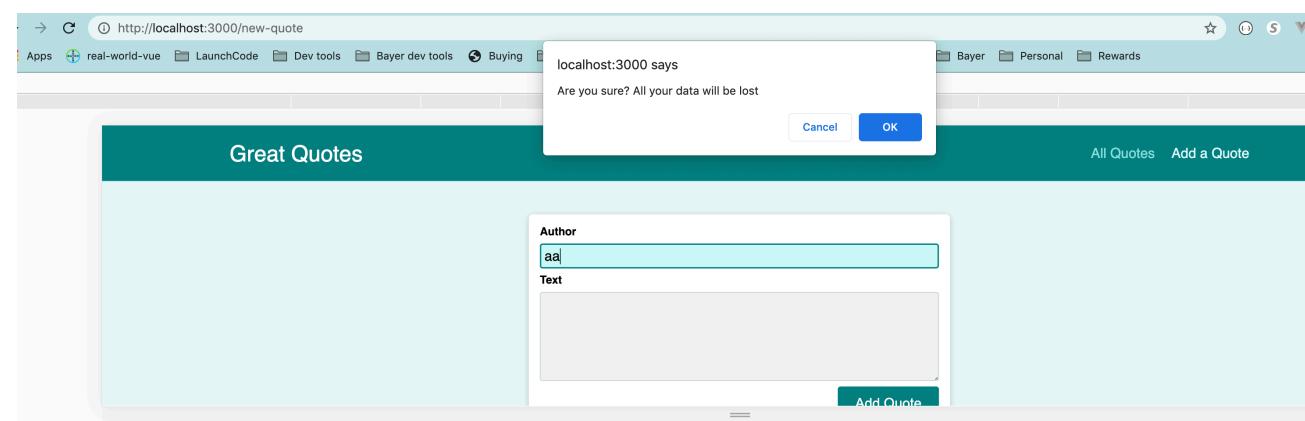
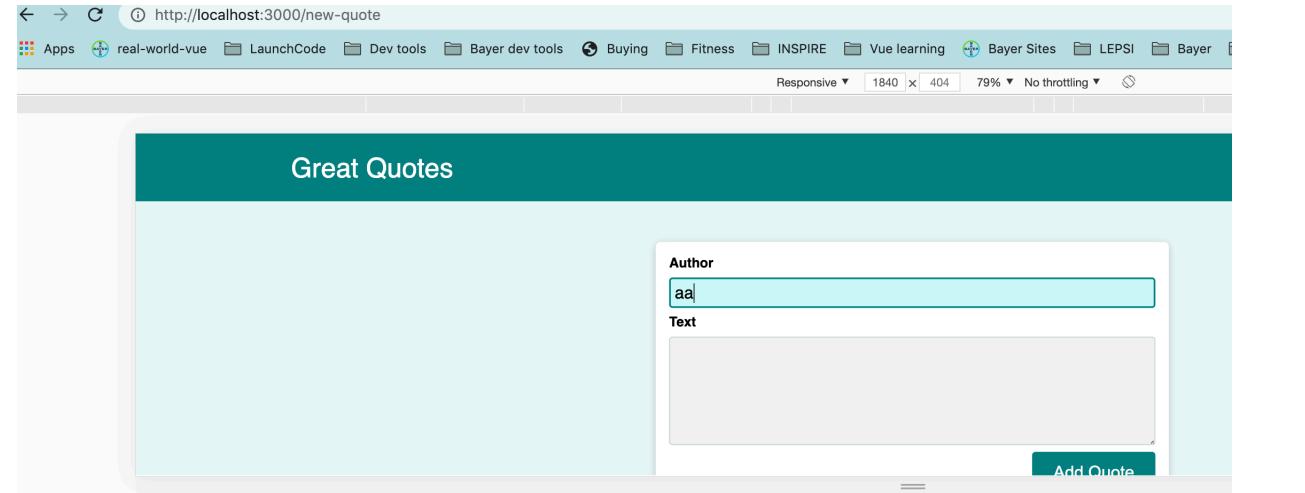
Project – Preventing some unwanted Route transitions

```
const formFocusHandler = () => {
  console.log('Focus!');
};

return (
  <Card>
    <form
      onFocus={formFocusHandler}
      className={classes.form}
      onSubmit={submitFormHandler}
    >
      {props.isLoading && (
        <div className={classes.loading}>
          <LoadingSpinner />
        </div>
      )}
    
```



Project – Preventing some unwanted Route transitions



```
const enteredAuthor = authorInputRef.current.value;
const enteredText = textInputRef.current.value;

// optional: Could validate here

props.onAddQuote({ author: enteredAuthor, text: enteredText });

}

const formFocusHandler = () => {
  console.log('Focus!');
  setIsEntering(true);
};

const finishEnteringHandler = () => {
  setIsEntering(false);
};

return (
  <>
    <Prompt
      when={isEntering}
      message={({location}) => 'Are you sure? All your data will be lost'}
    />
    <Card>
      <form
        onFocus={formFocusHandler}
        className={classes.form}
        onSubmit={submitFormHandler}
      >
        {props.isLoading && (
          <div className={classes.loading}>
            <LoadingSpinner />
          </div>
        )}
      </form>
      <div className={classes.control}>
        <label htmlFor="author">Author</label>
        <input type="text" id="author" ref={authorInputRef} />
      </div>
      <div className={classes.control}>
        <label htmlFor="text">Text</label>
        <textarea id="text" rows="5" ref={textInputRef}></textarea>
      </div>
      <div className={classes.actions}>
        <button onClick={finishEnteringHandler} className="btn">
          Add Quote
        </button>
      </div>
    </Card>
  </>
)
```

Project – Working with Query Parameters

Logic to sort components

Set and use query parameters to save the sorting

Saved in URL

If URL is used without query parameter, it is default sorted.

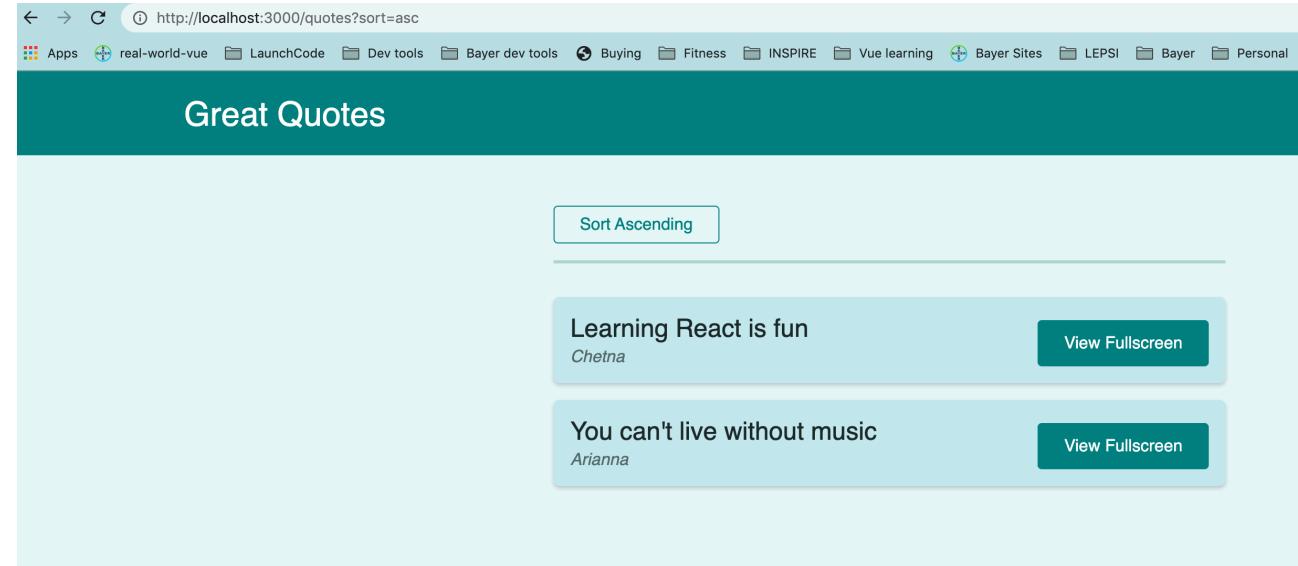
`useHistory` provides access to history object, which is an object that allows us to access and change a url

```
import { Fragment } from 'react';
import { useHistory } from 'react-router-dom';

import QuoteItem from './QuoteItem';
import classes from './QuoteList.module.css';

const QuoteList = (props) => {
  const history = useHistory();
  const changeSortingHandler = () => {
    history.push('/quotes?sort=asc');
  };
  return (
    <Fragment>
      <div className={classes.sorting}>
        <button onClick={changeSortingHandler}>Sort Ascending</button>
      </div>
      <ul className={classes.list}>
        {props.quotes.map((quote) => (
          <QuoteItem
            key={quote.id}
            id={quote.id}
            author={quote.author}
            text={quote.text}
          />
        ))}
      </ul>
    </Fragment>
  );
};

export default QuoteList;
```



Project – Working with Query Parameters

Read query parameter value

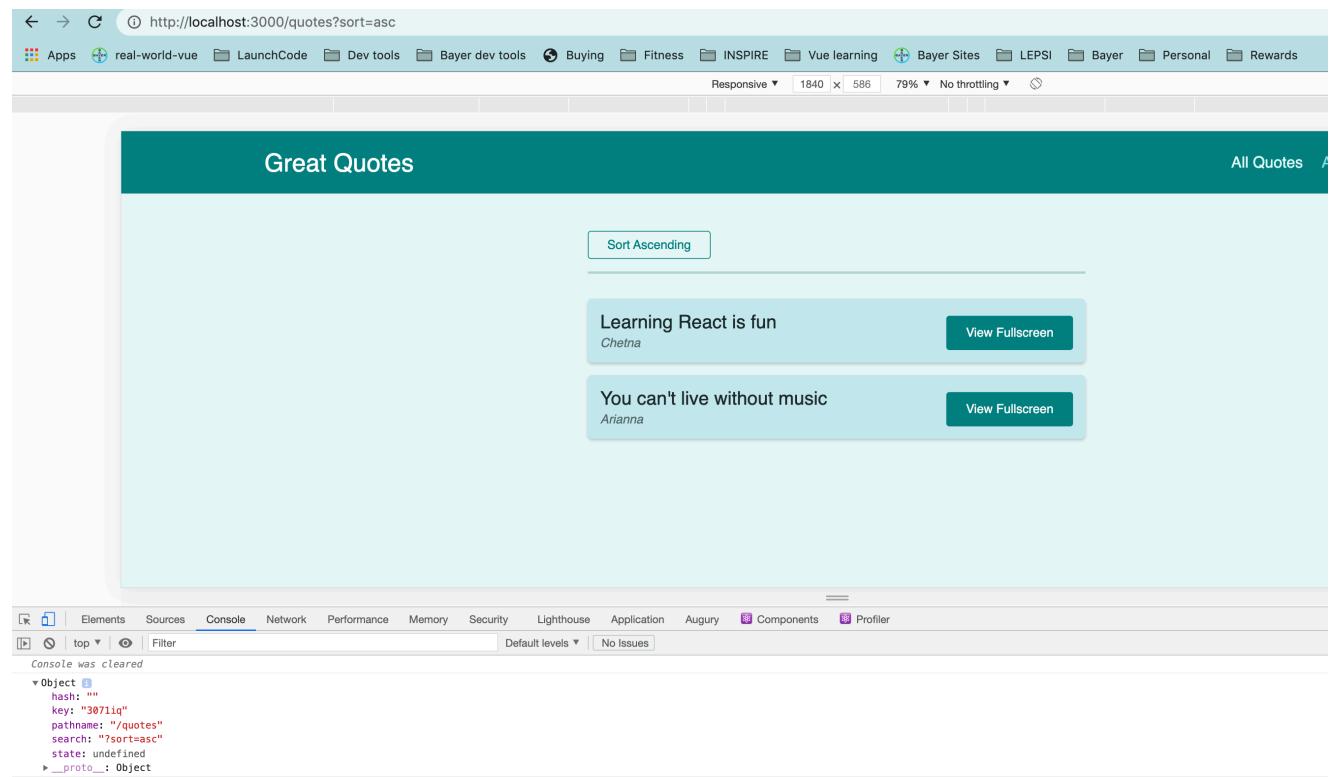
useLocation hook helps us provide access to location object that provides information on the loaded page and url.

```
import { Fragment } from 'react';
import { useHistory, useLocation } from 'react-router-dom';

import QuoteItem from './QuoteItem';
import classes from './QuoteList.module.css';

const QuoteList = (props) => {
  const history = useHistory();
  const location = useLocation();
  console.log(location);
  const changeSortingHandler = () => {
    history.push('/quotes?sort=asc');
  };
  return (
    <Fragment>
      <div className={classes.sorting}>
        <button onClick={changeSortingHandler}>Sort Ascending</button>
      </div>
      <ul className={classes.list}>
        {props.quotes.map((quote) => (
          <QuoteItem
            key={quote.id}
            id={quote.id}
            author={quote.author}
            text={quote.text}
          />
        ))}
      </ul>
    </Fragment>
  );
}

export default QuoteList;
```



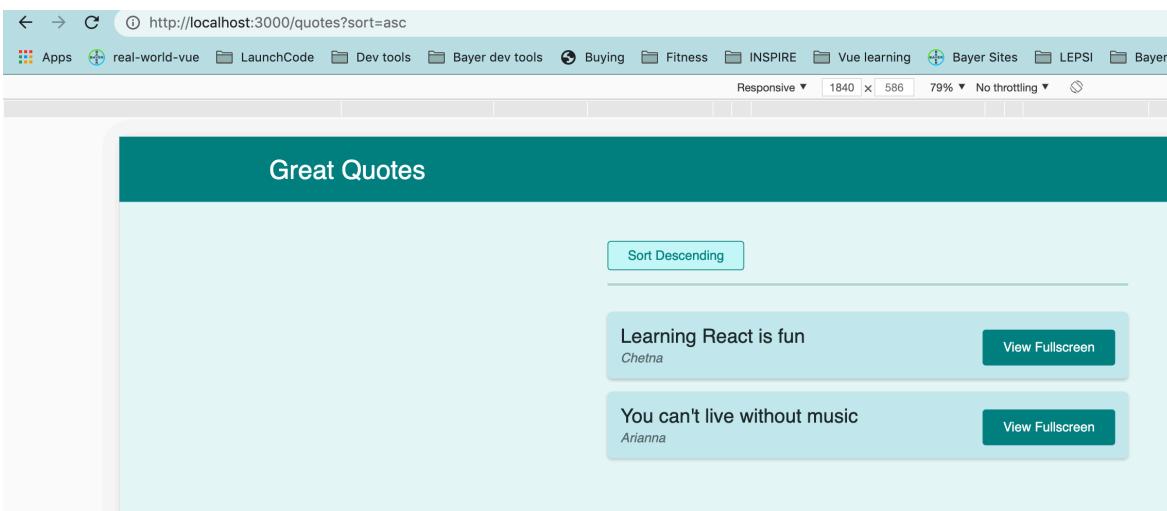
Project – Working with Query Parameters

Read query parameter value

useLocation hook helps us provide access to location object that provides information on the loaded page and url.

JS constructor URLSearchParams

Pass in location.search provides object



```
import { Fragment } from 'react';
import { useHistory, useLocation } from 'react-router-dom';

import QuoteItem from './QuoteItem';
import classes from './QuoteList.module.css';

const QuoteList = (props) => {
  const history = useHistory();
  const location = useLocation();
  const queryParams = new URLSearchParams(location.search);
  const isSortingAscending = queryParams.get('sort') === 'asc';

  const changeSortingHandler = () => {
    history.push('/quotes?sort=' + (isSortingAscending ? 'desc' : 'asc'));
  };

  return (
    <Fragment>
      <div className={classes.sorting}>
        <button onClick={changeSortingHandler}>
          Sort {isSortingAscending ? 'Descending' : 'Acsending'}
        </button>
      </div>
      <ul className={classes.list}>
        {props.quotes.map((quote) => (
          <QuoteItem
            key={quote.id}
            id={quote.id}
            author={quote.author}
            text={quote.text}
          />
        ))}
      </ul>
    </Fragment>
  );
};

export default QuoteList;
```

Project – Working with Query Parameters

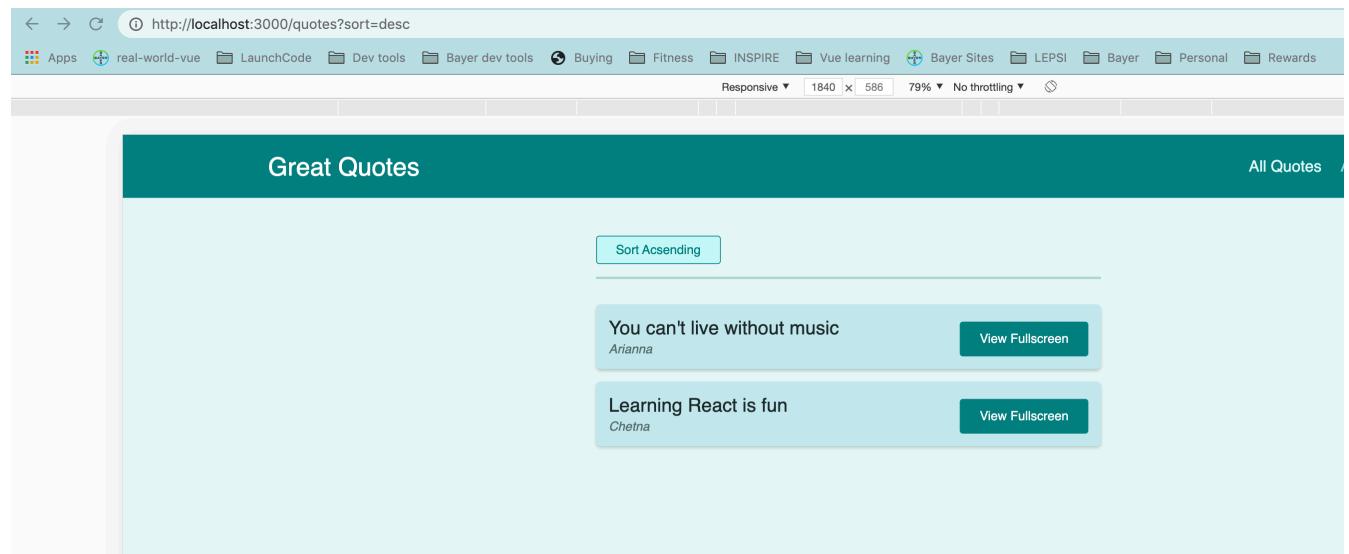
```
import { Fragment } from 'react';
import { useHistory, useLocation } from 'react-router-dom';

import QuoteItem from './QuoteItem';
import classes from './QuoteList.module.css';

const sortQuotes = (quotes, ascending) => {
  return quotes.sort((quoteA, quoteB) => {
    if (ascending) {
      return quoteA.id > quoteB.id ? 1 : -1;
    } else {
      return quoteA.id < quoteB.id ? 1 : -1;
    }
  });
};

const QuoteList = (props) => {
  const history = useHistory();
  const location = useLocation();
  const queryParams = new URLSearchParams(location.search);
  const isSortingAscending = queryParams.get('sort') === 'asc';
  const sortedQuotes = sortQuotes(props.quotes, isSortingAscending);
  const changeSortingHandler = () => const isSortingAscending: boolean
    history.push('/quotes?sort=' + (isSortingAscending ? 'desc' : 'asc'));
};

return (
  <Fragment>
    <div className={classes.sorting}>
      <button onClick={changeSortingHandler}>
        Sort {isSortingAscending ? 'Descending' : 'Ascending'}
      </button>
    </div>
    <ul className={classes.list}>
      {sortedQuotes.map((quote) => (
        <QuoteItem
          key={quote.id}
          id={quote.id}
          author={quote.author}
          text={quote.text}
        />
      ))}
    </ul>
  </Fragment>
);
```



Project – Getting Creative with Nested Routes.

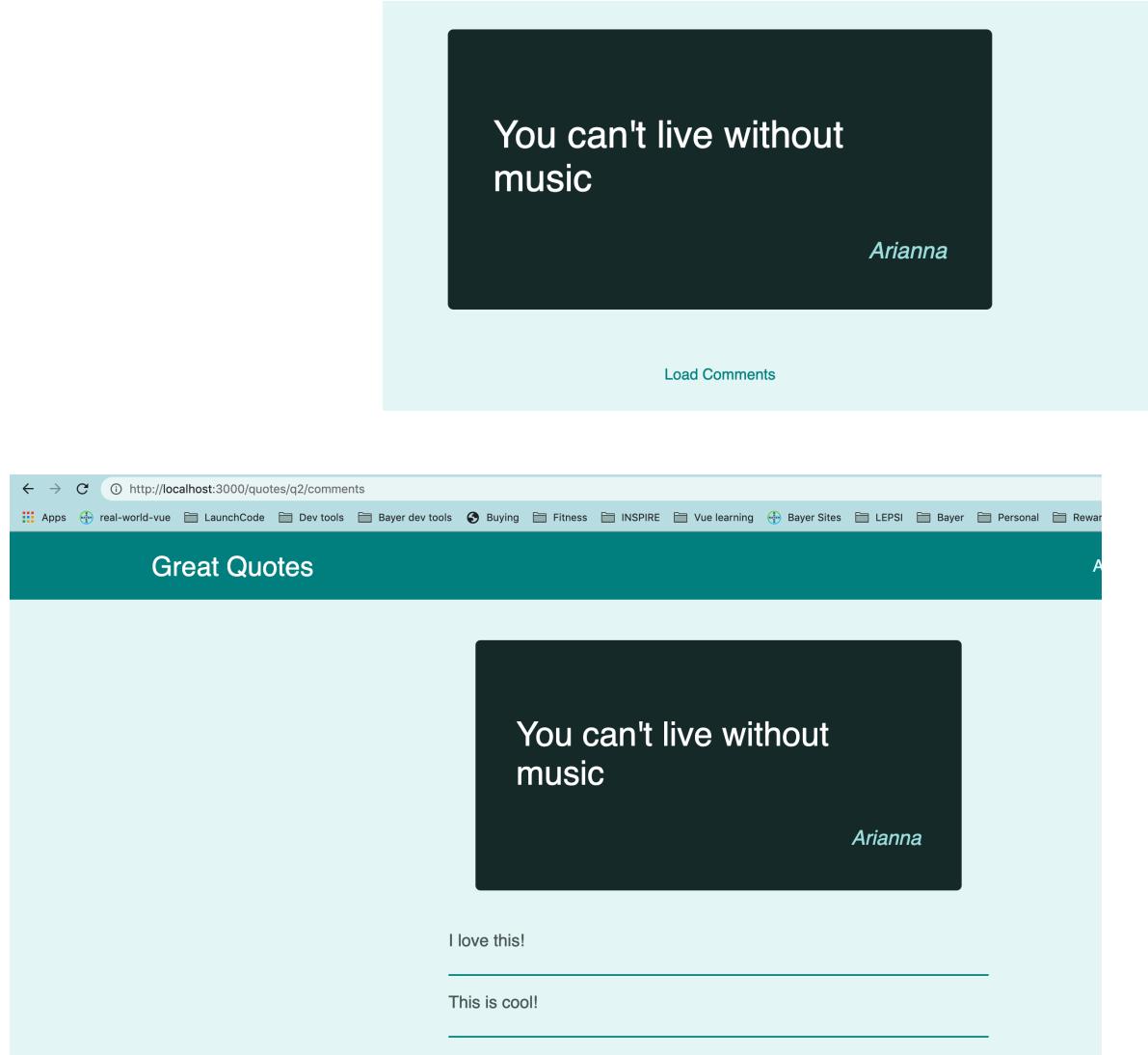
Comments path gets added to quote Detail. Can be a link instead.

```
import { useParams, Route, Link } from 'react-router-dom';
import CommentsList from '../comments/CommentsList';
import HighLightedQuote from '../quotes/HighlightedQuote';
const DUMMY_QUOTES = [
  { id: 'q1', author: 'Chetna', text: 'Learning React is fun' },
  { id: 'q2', author: 'Arianna', text: "You can't live without music" },
];

const DUMMY_COMMENTS = [
  { id: 'c1', text: 'I love this!' },
  { id: 'c2', text: 'This is cool!' },
];
const QuoteDetail = () => {
  const params = useParams();

  const quote = DUMMY_QUOTES.find((quote) => quote.id === params.quoteId);
  if (!quote) {
    return <p>No quote found!</p>;
  }
  return (
    <section>
      <HighLightedQuote text={quote.text} author={quote.author} />
      <Route path={`/quotes/${params.quoteId}/comments`} exact>
        <div className="centered">
          <Link className="btn--flat" to={`/quotes/${params.quoteId}/comments`}>
            Load Comments
          </Link>
        </div>
      </Route>
      <Route path="/quotes/:quoteId/comments">
        <CommentsList comments={DUMMY_COMMENTS} />
      </Route>
    </section>
  );
}

export default QuoteDetail;
```



Project – Writing more flexible routing code.

useMatchRoute has more data from react router

```
Line 14:9:  'match' is assigned a value but never used  no-unused-vars
  ▼Object 1
    isExact: true
    ▶ params: {quoteId: "q2"}
      path: "/quotes/:quoteId"
      url: "/quotes/q2/"
    ▶ __proto__: Object
  ▶ {path: "/quotes/:quoteId", url: "/quotes/q2", isExact: false, params: {...}}
```

>

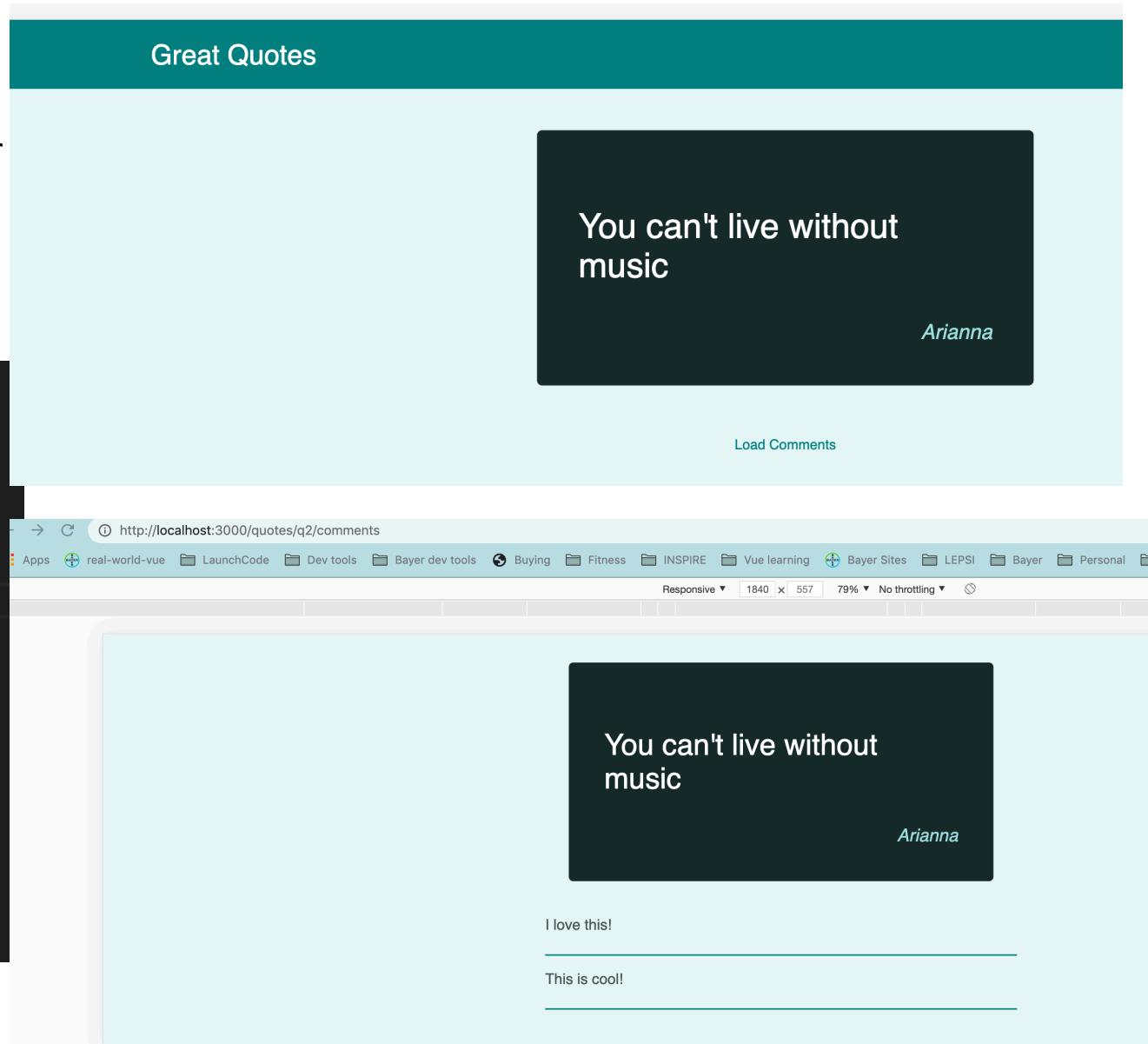
```
const match = useRouteMatch();
const params = useParams();

console.log(match);
```

Project – Writing more flexible routing code.

useMatchRoute has more data from react router

```
return (
  <section>
    <HighLightedQuote text={quote.text} author={quote.author} />
    <Route path={match.path} exact>
      <div className="centered">
        <Link className="btn--flat" to={`${match.url}/comments`}>
          Load Comments
        </Link>
      </div>
    </Route>
    <Route path={`${match.path}comments`}>
      <CommentsList comments={DUMMY_COMMENTS} />
    </Route>
  </section>
);
```



Project – Sending and getting Quote data via http

Project – Adding the comments features

Studio

<https://www.edureka.co/blog/interview-questions/react-interview-questions/>