# ENSF 381
# Full Stack Web Development

## Lecture 21: Props & Router

Slides: Ahmad Abdellatif, PhD
Instructor: Novarun Deb, PhD

UNIVERSITY OF CALGARY

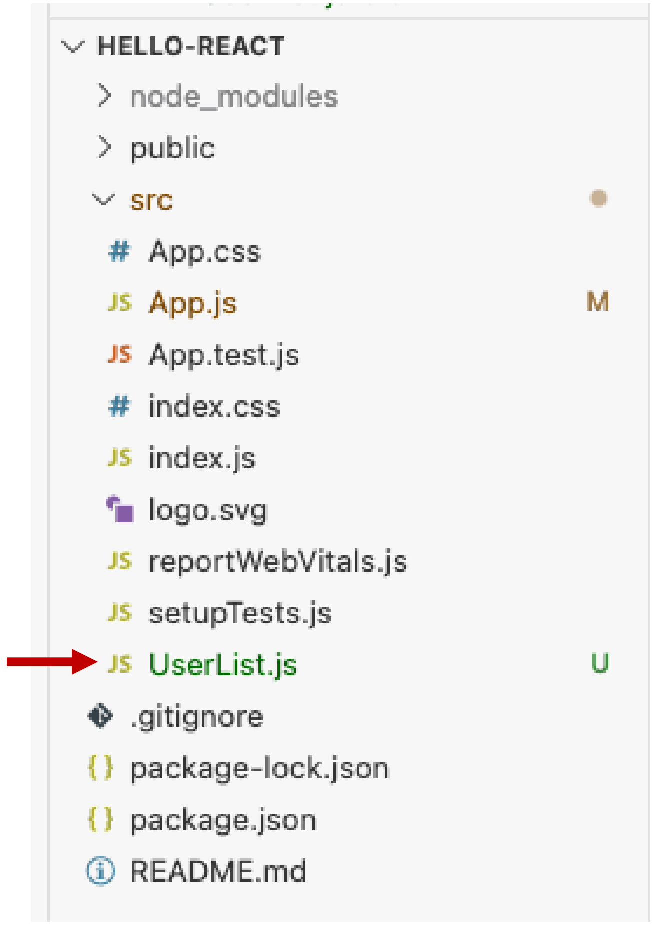
## Outline

- Recap.
- Props.
- Events.
- Router.

# Recap - Modularization

- We need to modularize the code to enhance its maintainability and readability.

- Create the list in a separate file:



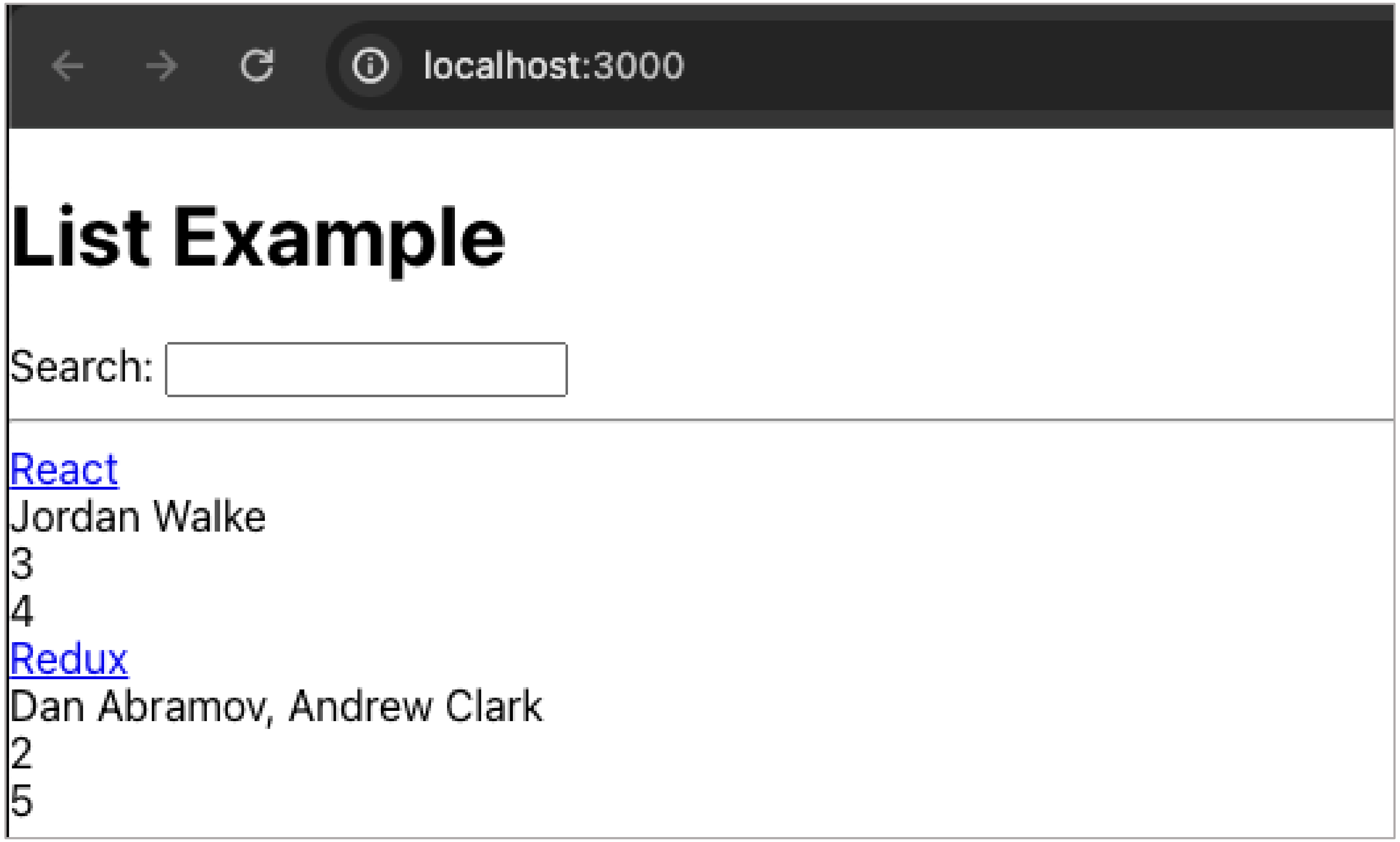
# Recap - Modularization (UserList component)

```
import React from 'react';
const list = [
  {
  title: 'React',
  url: 'https://reactjs.org/',
  author: 'Jordan Walke',
  num_comments: 3,
  points: 4,
  objectID: 0,
  },
  {
  title: 'Redux',
  url: 'https://redux.js.org/',
  author: 'Dan Abramov, Andrew Clark',
  num_comments: 2,
  points: 5,
  objectID: 1,
  },
  ];

function UserList() {
return list.map(function(item) {
  return (
  <div id={item.objectID}>
  <div>
  <a href={item.url}>{item.title}</a>
  </div>
  <div>{item.author}</div>
  <div>{item.num_comments}</div>
  <div>{item.points}</div>
  </div>
);
});
}
export default UserList;
```

## Recap - Modularization (App component)

```
import React from 'react';
import UserList from './UserList';


function App() {
return(
  <div>
  <h1>List Example</h1>
  <label htmlFor="search">Search: </label>
  <input id="search" type="text" />
  <hr />

<UserList />

</div>
);
}

export default App;
```

# Recap - Modularization



localhost:3000

List Example

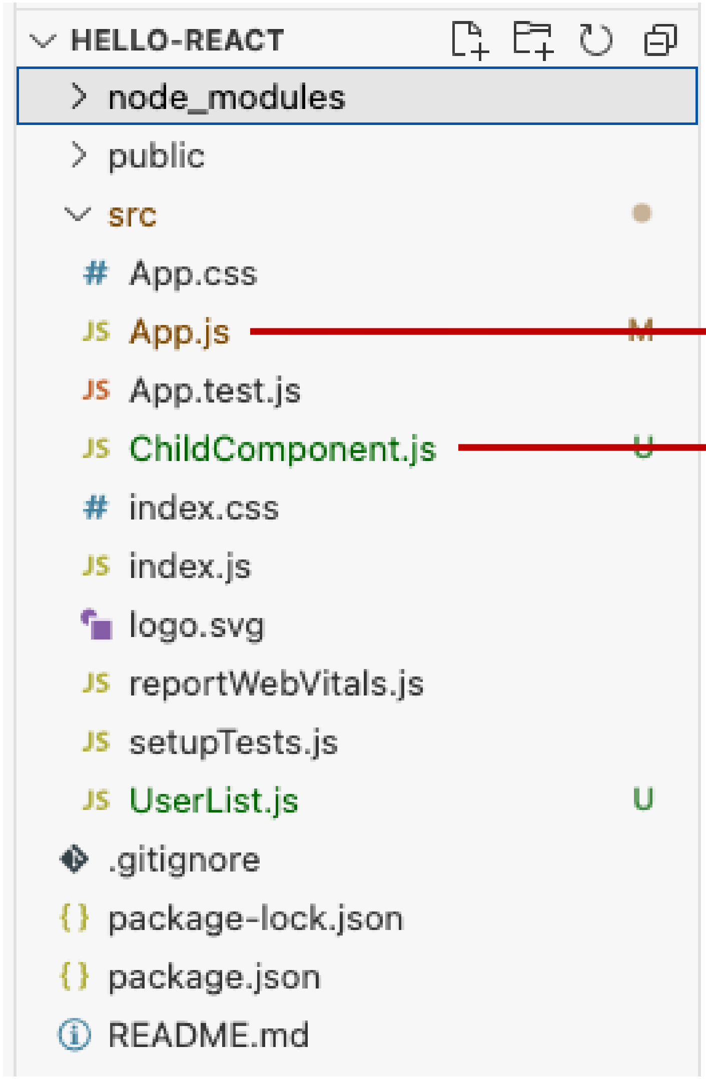
Search:

React
Jordan Walke
3
4
Redux
Dan Abramov, Andrew Clark
2
5

## Customizing component behavior and appearance with Props

- Pass data from a parent component to a child component.
- A set of arguments that are passed to a React component.
- These arguments are **similar to parameters in a function**, providing a way to customize the behavior and appearance of a component.
- **Props are read-only**: components can not modify the props they receive; they are considered immutable.
- "props" is an abbreviation for "properties".

# Props – Example



HELLO-REACT
- node_modules
- public
- src
  - App.css
  - App.js M
  - App.test.js
  - ChildComponent.js U
  - index.css
  - index.js
  - logo.svg
  - reportWebVitals.js
  - setupTests.js
  - UserList.js U
- .gitignore
- package-lock.json
- package.json
- README.md

Parent component

Child component

## Props – Example (App Component)

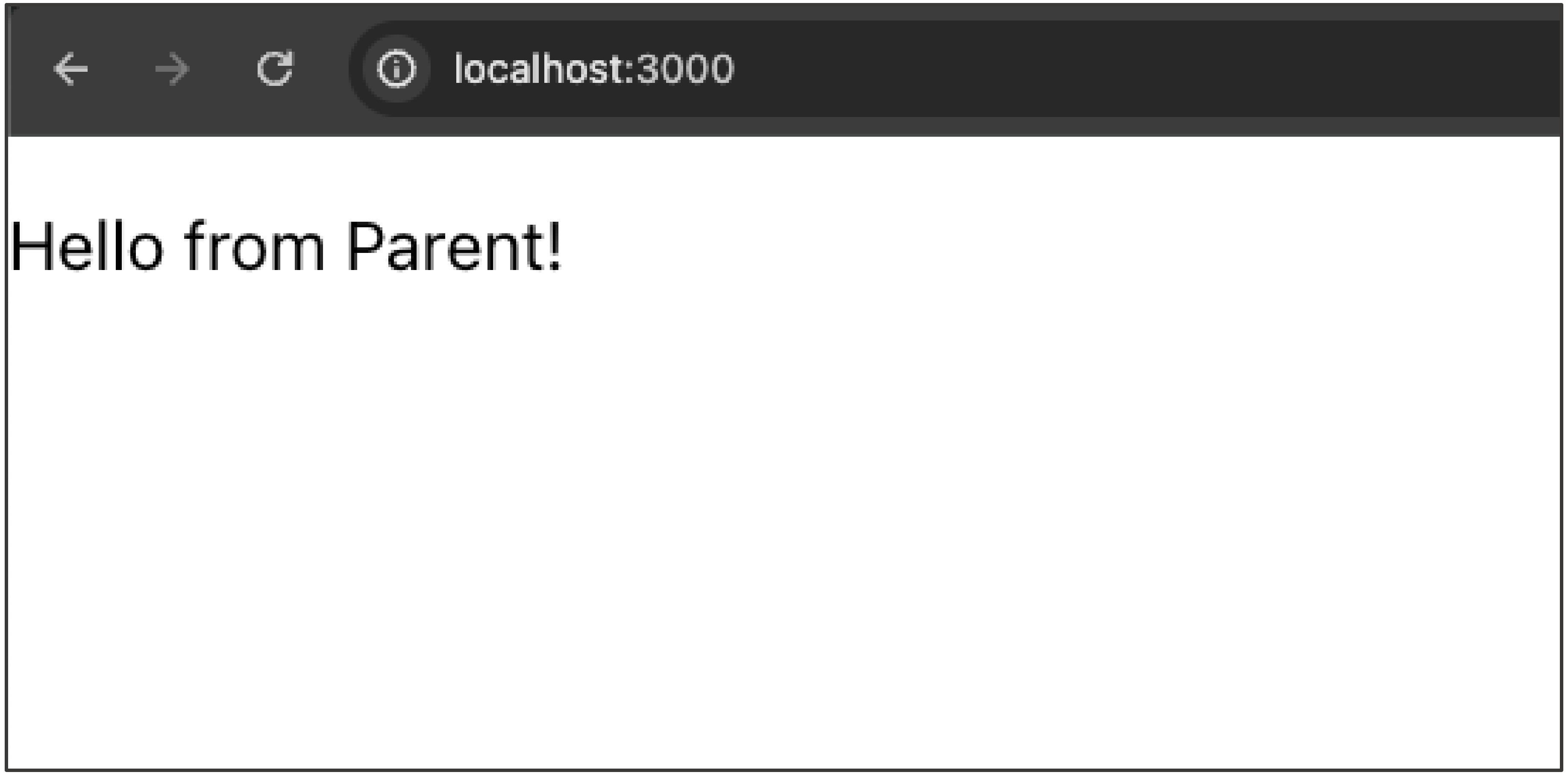
```
import React from 'react';
import ChildComponent from './ChildComponent.js';

function App() {
  const dataToPass = "Hello from Parent!";

  return (
    <div>
      <ChildComponent message={dataToPass} />
    </div>
  );
}

export default App;
```

Pass the 'message' from the parent component (App) to the child component (ChildComponent).

## Props – Example (ChildComponent)

```
import React from 'react';

function ChildComponent (props) {
    return (
        <div>
            <p>{props.message}</p>
        </div>
            );
        };

export default ChildComponent;
```

## Props – Example



```
Hello from Parent!
```

## Passing multiple parameters using props – Example (App component)

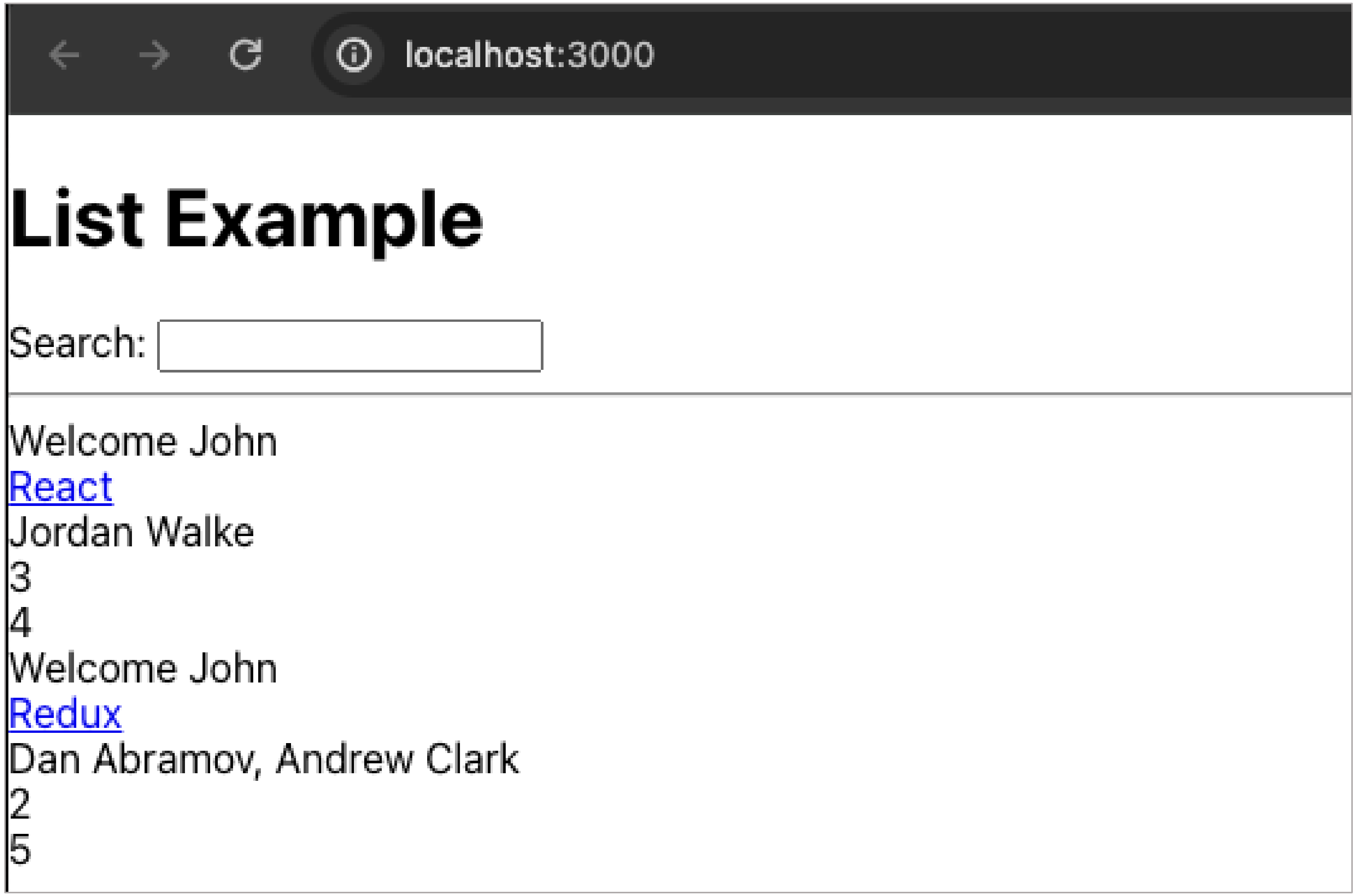
```
import React from 'react';
import UserList from './UserList';
const list = [
{
title: 'React',
url: 'https://reactjs.org/',
author: 'Jordan Walke',
num_comments: 3,
points: 4,
objectID: 0,
},
{
title: 'Redux',
url: 'https://redux.js.org/',
author: 'Dan Abramov, Andrew Clark',
num_comments: 2,
points: 5,
objectID: 1,
}];
function App() {
return(
  <div>
  <h1> List Example </h1>
  <label htmlFor="search">Search: </label>
  <input id="search" type="text" />
  <hr />
<UserList list= {list} name="John"/>
</div>
);
}
export default App;
```

# Passing multiple parameters using props – Example (UserList component)

```
import React from 'react';

function UserList(props) {

return props.list.map(function(item) {
  return (

  <div id={item.objectID}>
  <div> Welcome {props.name}</div>
  <div>
  <a href={item.url}>{item.title}</a>
  </div>
  <div>{item.author}</div>
  <div>{item.num_comments}</div>
  <div>{item.points}</div>
  </div>
);
});
}
export default UserList;
```

Src: The Road to React: The React.js with Hooks in JavaScript Book

# Props – Example 2



localhost:3000

List Example

Search:

```
Welcome John
React
Jordan Walke
3
4
Welcome John
Redux
Dan Abramov, Andrew Clark
2
5
```

# Events in React

- Actions or occurrences that can be detected and handled by React components.

- React has the same events as HTML.

- React abstracts the browser's native events, providing **a consistent way** to handle interactions in the applications.

- Here are some common events:
  - onClick
  - onChange
  - onMouseOver, onMouseOut
  - onLoad

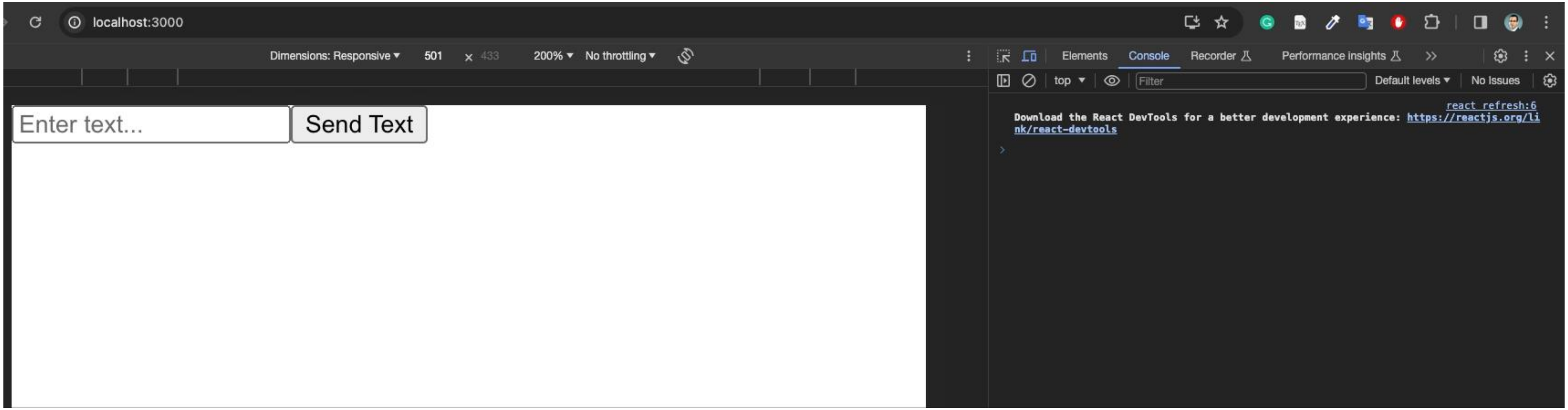
# Events - Example

```
import React from 'react';

function handleClick() {
console.log(document.getElementById("name_input_field").value);
};

function handleChange(event) {
  console.log(event.target.value);
  };

function App() {
return (
  <div>
    <input
      id="name_input_field"
      type="text"
      placeholder="Enter text..."
      onChange={handleChange}
    />
    <button onClick={handleClick}>Send Text</button>
  </div>
);}
export default App;
```

We call the functions without ().

Src: The Road to React: The React.js with Hooks in JavaScript Book

# Events - Example



```
react_refresh:6
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
```

## React Router

- A popular library for handling navigation and routing in React applications.
- It enables the creation of **single-page applications** by allowing developers to define different "routes" within their application and rendering the appropriate components based on the current URL.
- We need to install the React Router Library*:

```
npm install react-router-dom
```

***Ensure that you are located at the root of your React project.**

## Router - Syntax

```
import React from 'react';
import { BrowserRouter, Routes, Route } from 'react-router-dom';

function App() {
  return (
   <BrowserRouter>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
   </BrowserRouter>
  );
};

export default App;
```

This component is the root of the routing configurations.

Used to define your application's routes and contains the individual route configurations.

Specifies a particular route and the component to be rendered when that route is matched.

## Router - Syntax

- Path: The path attribute specifies the URL path for which this route should be active.

- Element: The element attribute specifies the React element that should be rendered when this route is matched.

## Router - Example

In this example, we will have three pages:

- Home Page
- About Us Page
- Contact Us Page

## Router – Example (App)

```
import React from 'react';
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import Home from './Home';
import AboutUs from './AboutUs';
import ContactUs from './ContactUs';

function App() {
return(
<BrowserRouter>
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/About" element={<AboutUs />} />
  <Route path="/ContactUs" element={<ContactUs />} />
</Routes>
</BrowserRouter>
);
}

export default App;
```

## Router – Example (Home)

```
import React from 'react';

function Home() {
  return (
   <div>
     <h1>Welcome to the home page!</h1>
     <a href='/'> Home </a>
     <a href='/About'> About Us</a>
     <a href='/ContactUs'> Contact Us</a>
   </div>
  );
};

export default Home;
```

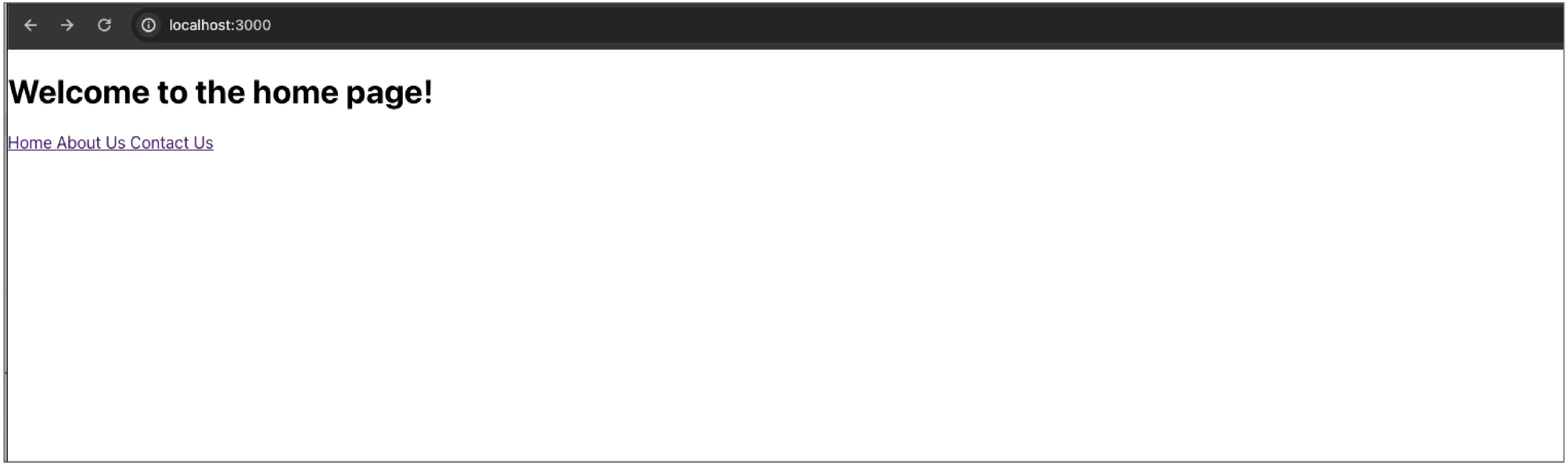
## Router – Example (AboutUs)

```
import React from 'react';

function AboutUs() {
  return
   <h1>
     This is About Us page!
   </h1>;
};

export default AboutUs;
```

## Router – Example (ContactUs)

```
import React from 'react';

function ContactUs() {
  return
    <h1>
      For any question, please contact us at: info@info.com
    </h1>;
};

export default ContactUs;
```

## Router – Example



localhost:3000

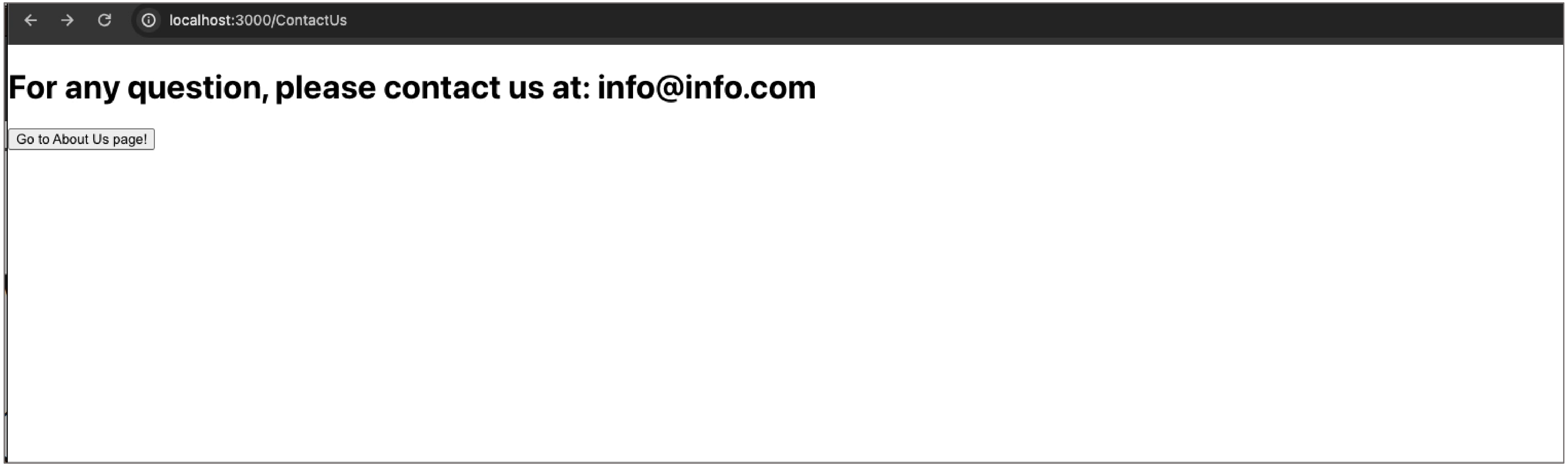
Welcome to the home page!

Home About Us Contact Us

# Router – Navigate to another page using Event (ContactUs)

```
import React from 'react';
import {useNavigate } from 'react-router-dom';

function ContactUs() {

const navigate = useNavigate(); // This hook is used for programmatic navigation in a React application

function handleButtonClick(){
  navigate("/About")
}
  return (
    <div>
      <h1>For any question, please contact us at: info@info.com</h1>
      <button onClick={handleButtonClick}>Go to About Us page!</button>
    </div>
  );
};

export default ContactUs;
```

# Router – Navigate to another page using Event (ContactUs)



For any question, please contact us at: info@info.com

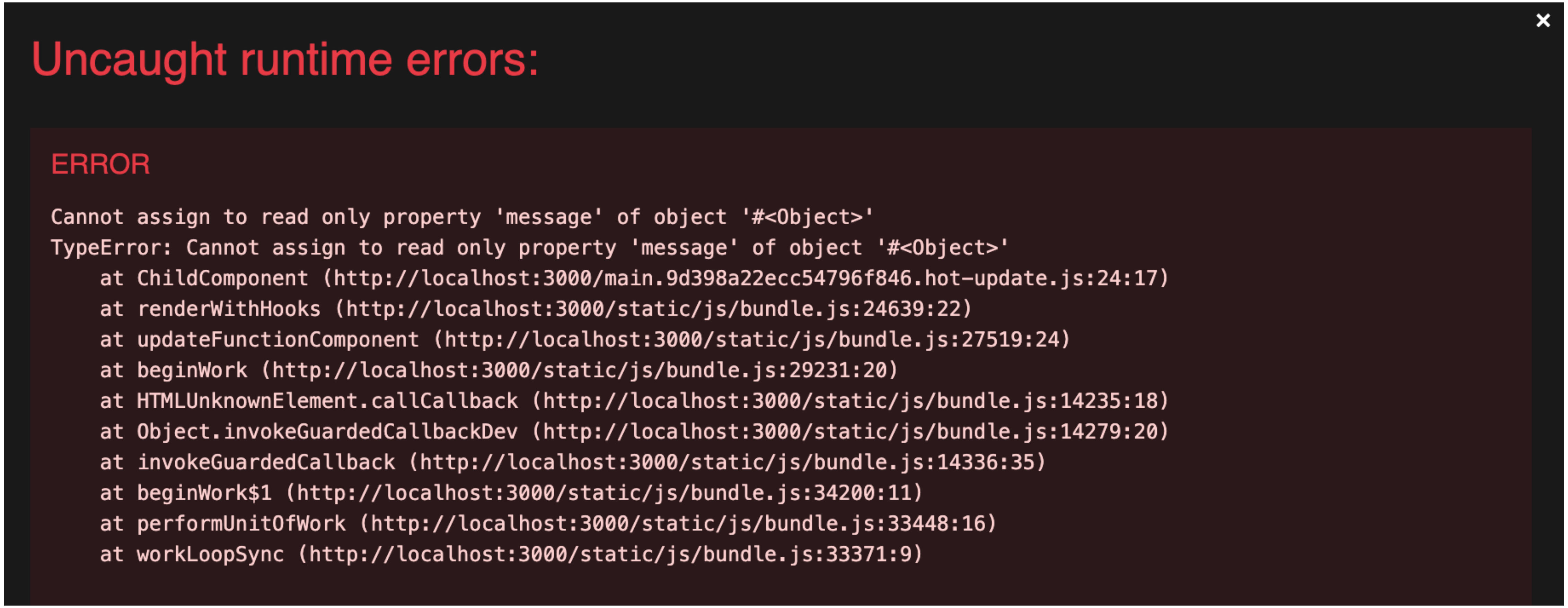
## What is the output?

```
import React from 'react';
import ChildComponent from './ChildComponent.js';

function App() {
  const dataToPass = "Hello from Parent!";

  return (
    <div>
      <ChildComponent message={dataToPass} />
    </div>
  );
}

export default App;
```

## What is the output?

```
import React from 'react';

function ChildComponent (props) {
props.message = 'Hello ENSF381'
    return (
        <div>
            <p>{props.message}</p>
        </div>
            );
        };

export default ChildComponent;
```

## What is the output?



Uncaught runtime errors:

ERROR

```
Cannot assign to read only property 'message' of object '#<Object>'
TypeError: Cannot assign to read only property 'message' of object '#<Object>'
    at ChildComponent (http://localhost:3000/main.9d398a22ecc54796f846.hot-update.js:24:17)
    at renderWithHooks (http://localhost:3000/static/js/bundle.js:24639:22)
    at updateFunctionComponent (http://localhost:3000/static/js/bundle.js:27519:24)
    at beginWork (http://localhost:3000/static/js/bundle.js:29231:20)
    at HTMLUnknownElement.callCallback (http://localhost:3000/static/js/bundle.js:14235:18)
    at Object.invokeGuardedCallbackDev (http://localhost:3000/static/js/bundle.js:14279:20)
    at invokeGuardedCallback (http://localhost:3000/static/js/bundle.js:14336:35)
    at beginWork$1 (http://localhost:3000/static/js/bundle.js:34200:11)
    at performUnitOfWork (http://localhost:3000/static/js/bundle.js:33448:16)
    at workLoopSync (http://localhost:3000/static/js/bundle.js:33371:9)
```

## Questions

?

## References

- https://legacy.reactjs.org/docs/events.html