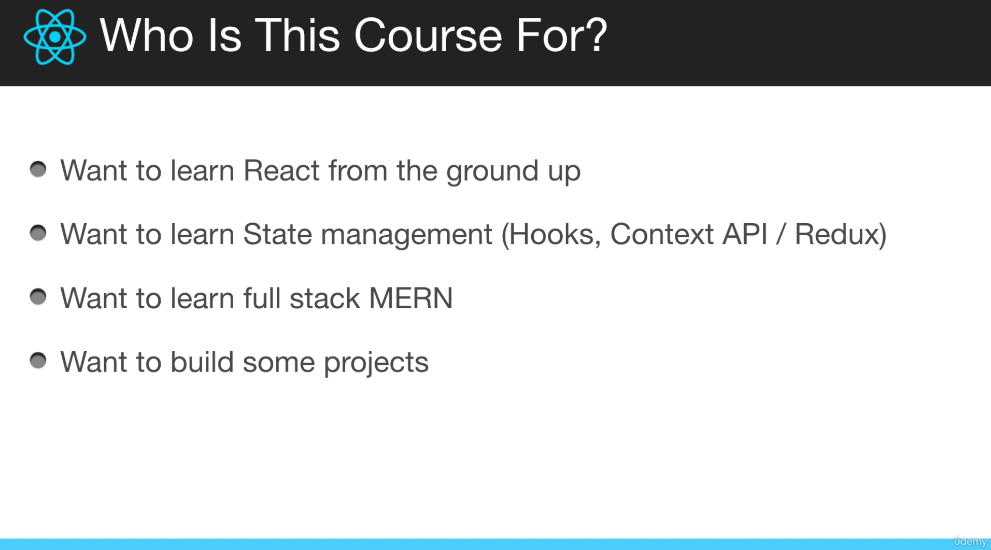
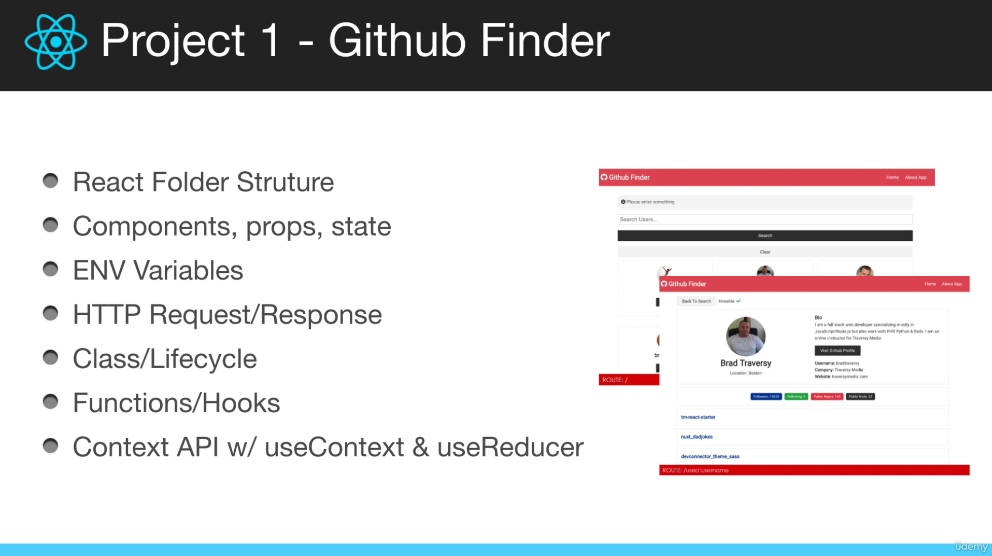
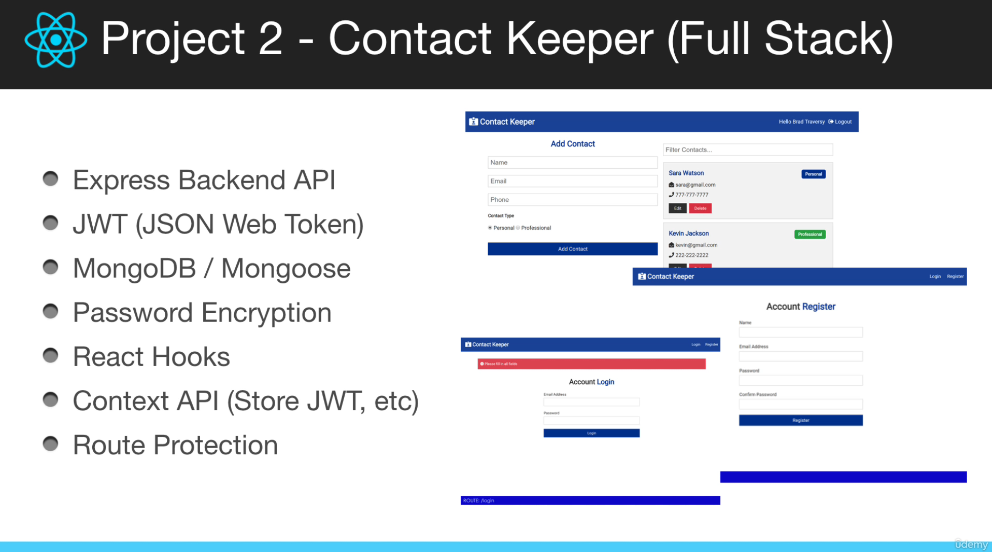
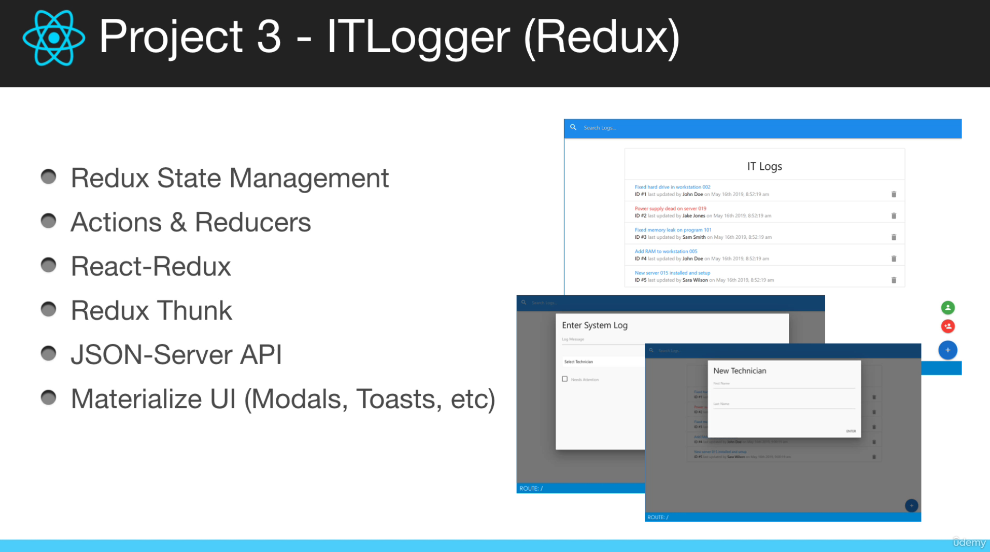
**SLIDES**

**Section 1: Course Introduction**

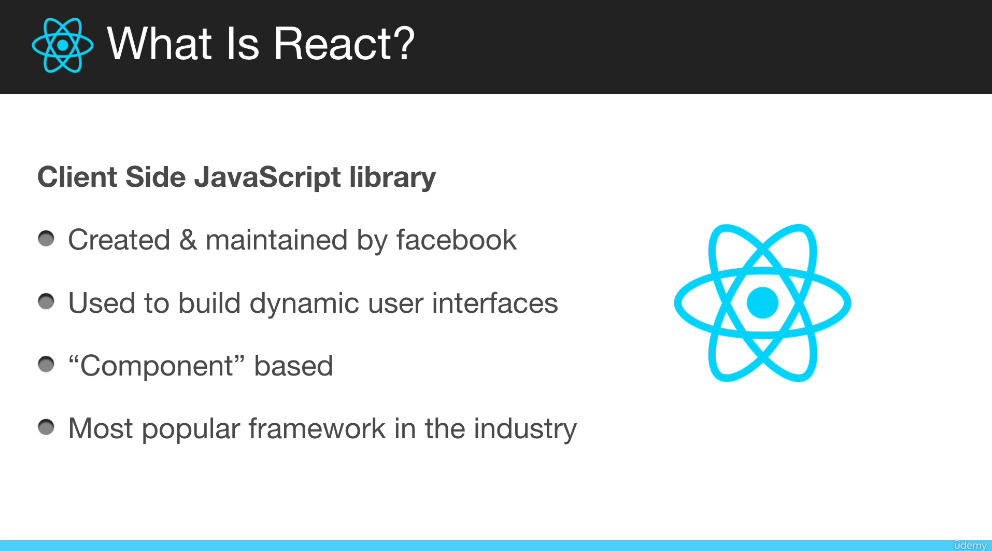
****

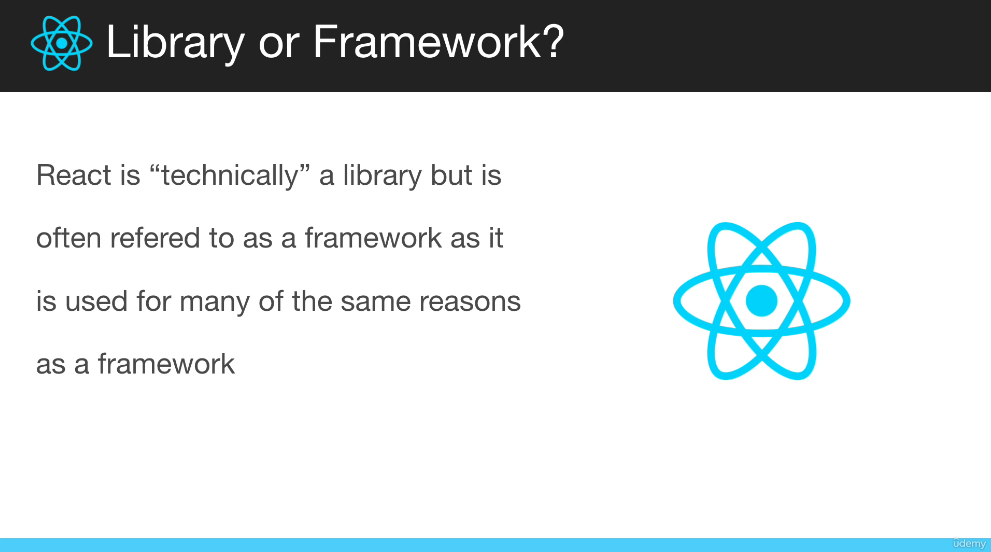
****

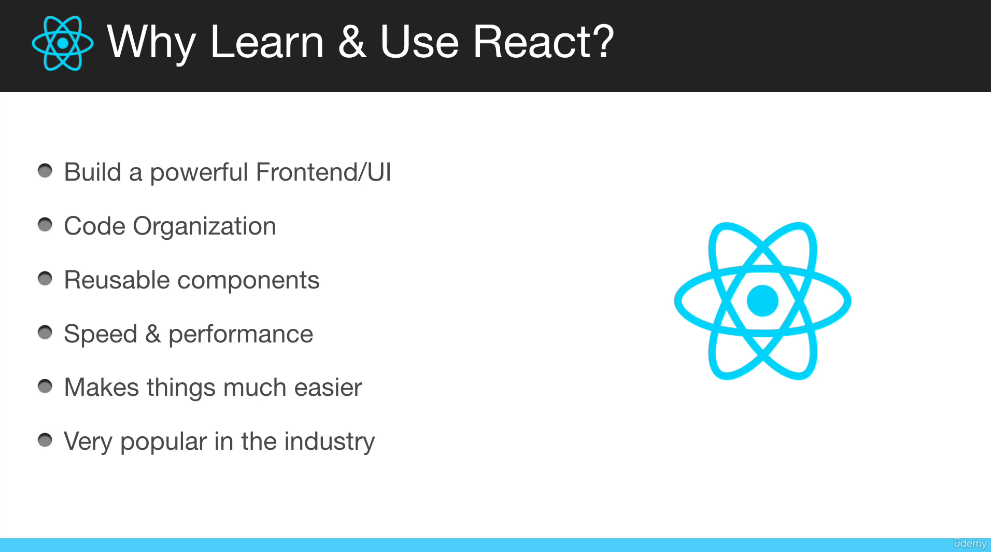
****

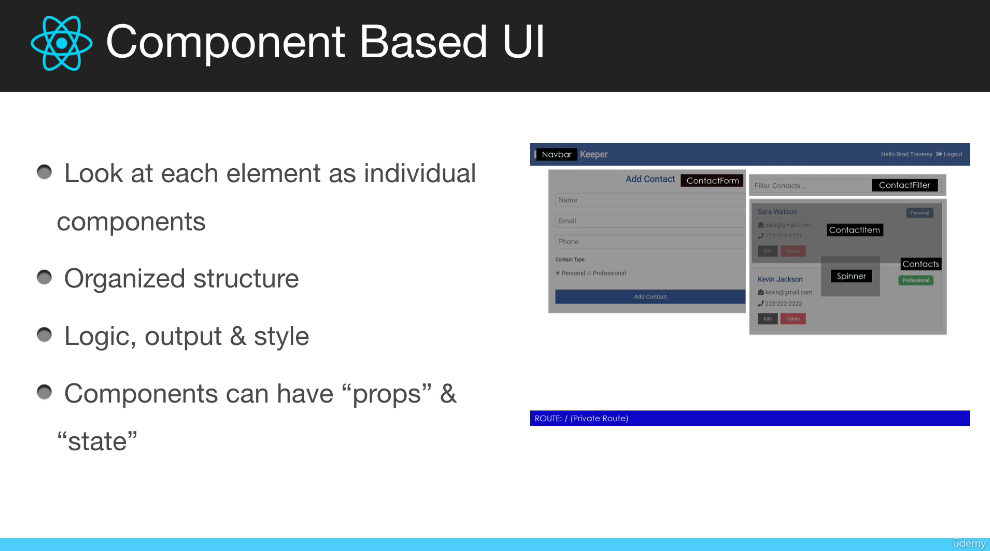
****

**What is React?**









**Section 2: Project 1 Start [Github Finder] & React Setup**

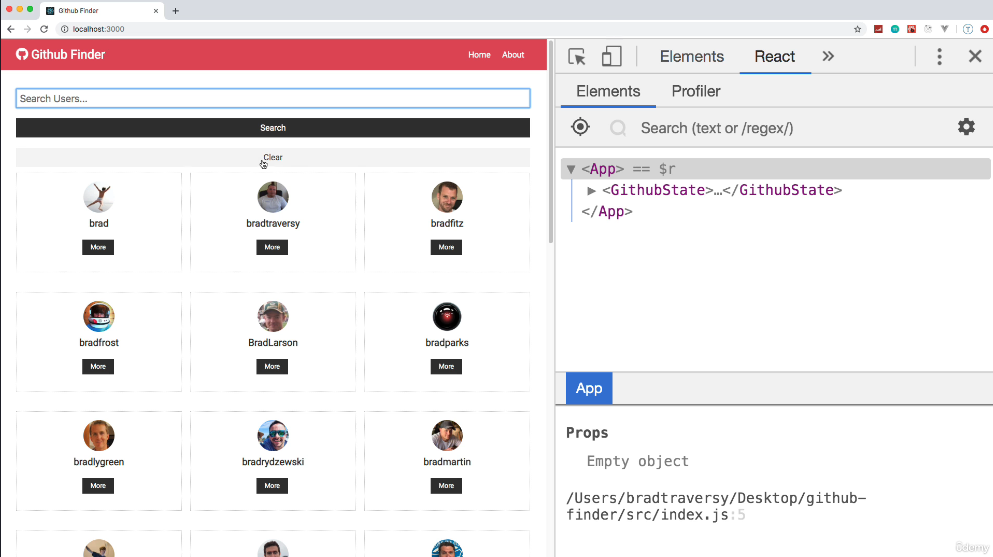
**Project Introduction**

1) We are going to start off using class components and stateless functional components.

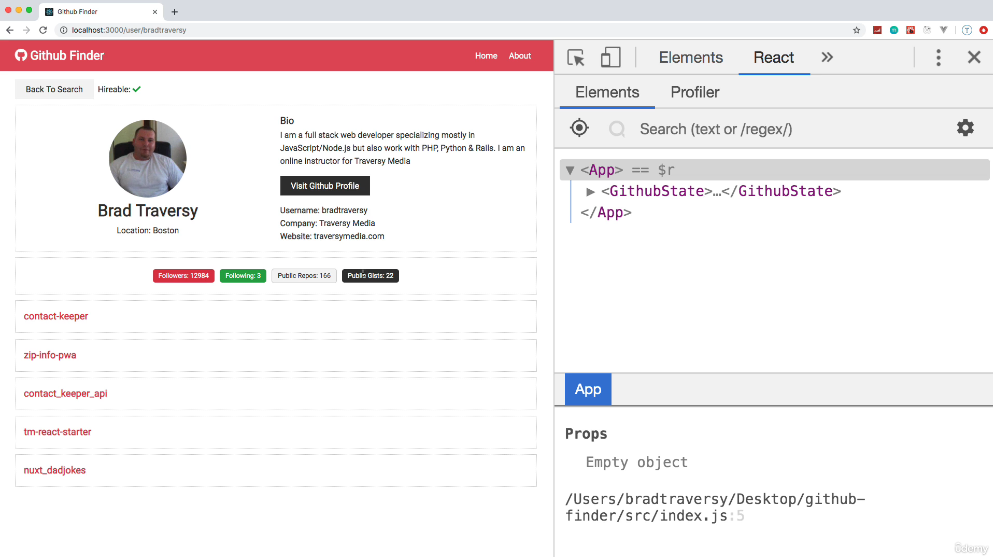
2) Then we are going to refactor the app to all functional components using hooks.

3) We are going to implement the app level state using the context API, the useReducer hook, and the useContext hook.

4) We are going to use the GitHub public API and search for users.



We use react-router to redirect here:



**Code & Resources**

The final code for this project can be found in this Github repo:

<https://github.com/bradtraversy/github-finder>

Docs for the Github API:

<https://developer.github.com/v3/>

To Register Your Github App & Get Keys:

<https://github.com/settings/applications/new>

API Endpoints We Will Be Working With:

<https://api.github.com/users>

<https://api.github.com/search/users?q=brad>

<https://api.github.com/users/bradtraversy>

<https://api.github.com/users/bradtraversy/repos>

**Use this solution when using npx create-react-app, and it doesn’t work.**

npm install ajv@^6.12.6 [ajv-keywords@^3.5.2](mailto:ajv-keywords@%5e3.5.2)

**When running the app cloned from the Brad’s repository, I had a problem when running the app. This solved it:**

1. Run the app from the cmd, and run this command:

$env:NODE\_OPTIONS="--openssl-legacy-provider"

npm start

**SPA:** Single Page Application framework. Everything is routed through a single file (public/index.html)

**index.js**

In the old syntax of React, we had the following, in index.js

ReactDom.render(<App />, document.getElementById(‘root’));

But now, we have this:

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);

**CRA Cleanup & Prepare**

How to write a functional component like a class component:

We have this functional component:

function App() {

  return (

    <div className="App">

      <h1>Hello from React</h1>

    </div>

  );

}

export default App;

This is how it looks if we turn it into a class component:

class App extends Component{

  render() {

    return (

      <div className="App">

        <h1>Hello from React</h1>

      </div>

    );

  }

}

export default App;