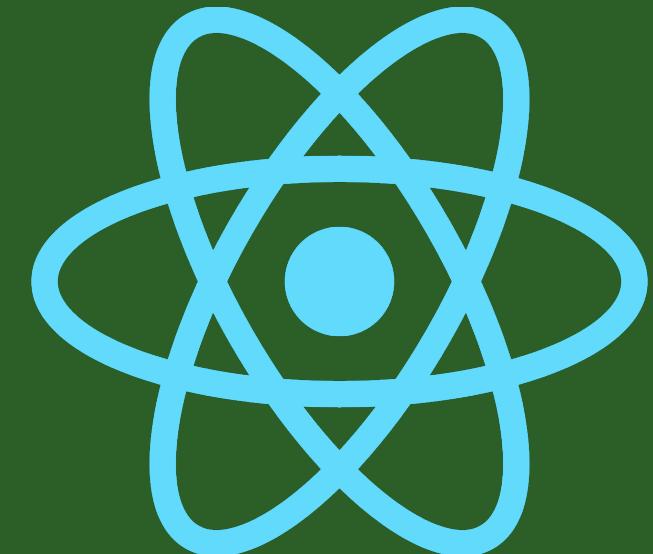
A detailed green circuit board graphic is positioned on the left side of the slide, showing various components like resistors, capacitors, and inductors.

REACT.JS



REACT WORKSHOP - SOHAIL HANIF

[HTTPS://GITHUB.COM/SOHAILHANIF/REACT](https://github.com/sohailhanif/react)

OBJECTIVES

Upon completion of this lesson, you should know:

- NPM / Dev Environment Setup
- React.JS
- Components
- Props
- State
- Navigation
- Bootstrap
- API calls

WEB FRAMEWORKS

- Collection of modules, packages, and libraries that are designed to help speed up development of applications
- Provide a standard way to build and deploy web application
- Automate overhead of commonly done tasks in web development
- Used for both static websites and dynamic web applications
- Security, scalability, robustness, integrations

FRONTEND WEB FRAMEWORKS

- Run on the client browser
- Display data to user
- Accept user input
- User interaction and experience
- Send data to server
- React, Angular, Vue



BACKEND WEB FRAMEWORKS

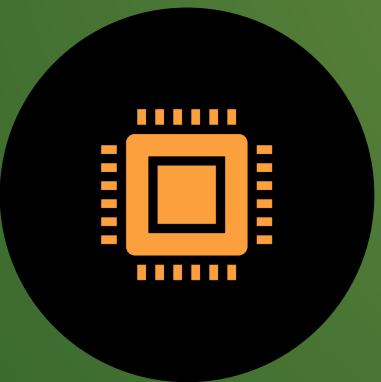
- Run on the server
- Accept requests and data from client
- Define and send pages
- Processing of data
- Long term storage of data



NODE PACKAGE MANAGEMENT

- Install 3rd party packages
- Frameworks, libraries, tools
- Packages are downloaded into the `node_modules` folder
- Dependencies and requirements listed in `package.json` file
- Packages can be install locally per project or globally on system
- Multiple package managers including `npm`, `bower`, and `jspm`

NODE MODULES



CORE MODULES THAT ARE
COMMON IN MOST APPS



3RD PARTY MODULES THAT
CAN BE INSTALLED THROUGH
PACKAGE MANAGERS



CUSTOM MODULES CREATED
AND EXPORTED THROUGH
LOCAL FILES

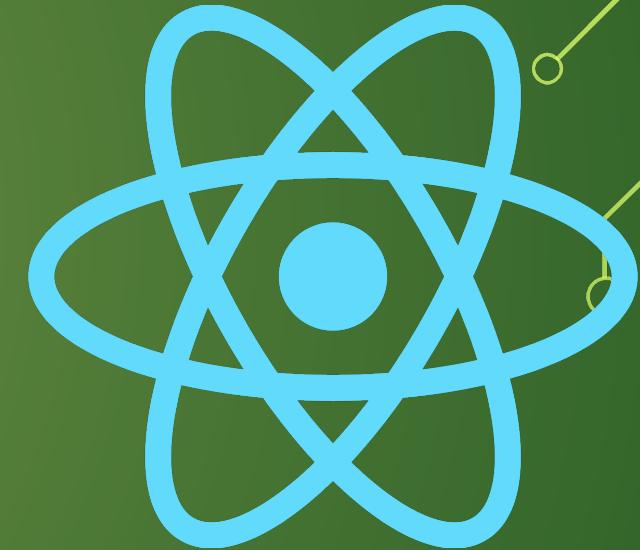


INSTALLING NODE JS

CAN DOWNLOAD FOR ALL PLATFORMS FROM [HTTPS://NODEJS.ORG/EN/](https://nodejs.org/en/)

REACT.JS

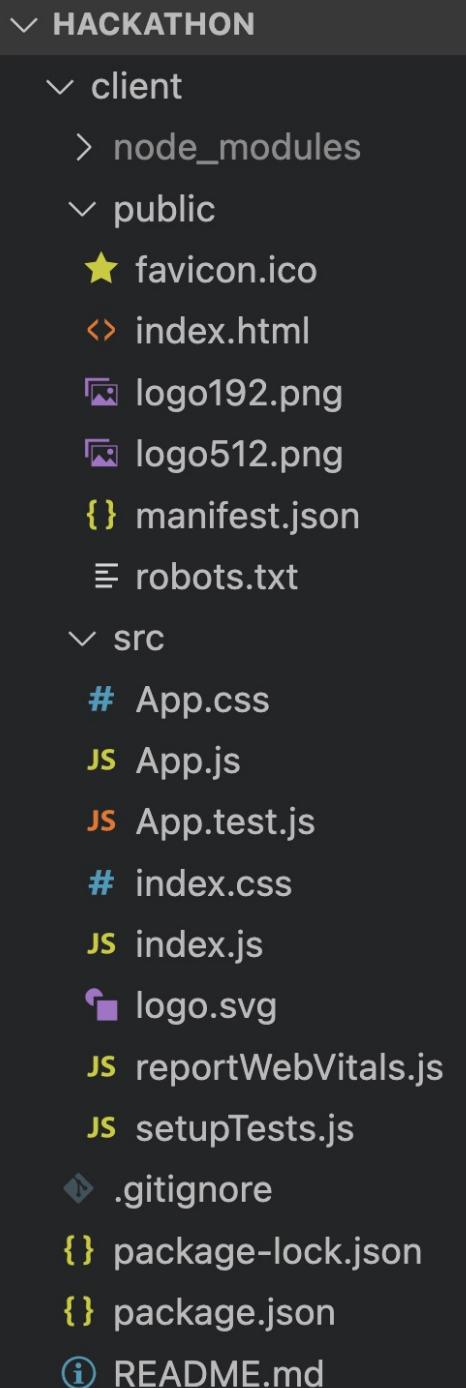
- React.js is a frontend framework
- Responsible for rendering content to the user, collecting user input, and communicating with the backend
- React is based on reusable components and makes it easy to create interactive UIs and manage the state of an application
- `npm install create-react-app`





REACT.JS ENVIRONMENT SETUP

- Create folder called Hackathon and open in VSCode
- Open terminal in VSCode (Terminal > New terminal)
 - npm install create-react-app
 - npx create-react-app client
 - cd client
 - npm run start
- Can access React app through browser at localhost:3000



REACT.JS COMPONENTS

- Split the UI into independent, reusable pieces
- Accept arbitrary inputs (called “props”) and return React elements describing what should appear on the screen.
- Can have both class and function components
- Can manage their own state and handle user events
- Uses JSX which is a syntax extension to JavaScript
- Can store in a folder called components in the src folder

REACT.JS PROPS

- Props (short for "properties") are a way to pass data from a parent component to a child component
- Can be accessed inside the component using `this.props`
- Used to customize the behavior and appearance of a component
- Make a component more reusable by allowing it to be configured in different ways depending on the context in which it is used
- Props are immutable

REACT.JS COMPONENTS AND PROPS

The image shows a dark-themed code editor interface. On the left is an Explorer sidebar titled "EXPLORER" showing a project structure under "HACKATHON". The main area is a code editor tab titled "JS Person.js" with the following content:

```
client > src > components > JS Person.js > ...
1 import React from 'react';
2
3 class Person extends React.Component {
4   static defaultProps = {
5     name: 'John Smith',
6     age: 30
7   };
8   render() {
9     return (
10       <div>
11         <p>My name is {this.props.name}</p>
12         <p>My age is {this.props.age}</p>
13       </div>
14     );
15   }
16 }
17
18 export default Person
```

REACT.JS COMPONENTS AND PROPS

JS App.js X

client > src > JS App.js > ...
You, 1 second ago | 1 author (You)

```
1 import './App.css';
2
3 import Person from './components/Person';
4
5 function App() {
6   return (
7     <div className="App">
8       <header className="App-header">
9         <Person/>
10        <Person name="Jane Smith" age={25}/>
11       </header>
12     </div>
13   );
14 }
15
16 export default App;
```

My name is John Smith

My age is 30

My name is Jane Smith

My age is 25

REACT.JS COMPONENTS AND PROPS

client > src > components > JS People.js > ...

```
1 import React from 'react';
2 import Person from './Person.js';
3
4 class People extends React.Component {
5   render() {
6     return (
7       <div>
8         <Person />
9         <Person name="Drake" age={20} />
10        <Person name="Alice" age={23} />
11        <Person name="Jason" age={17} />
12        <Person name="Kevin" age={19} />
13     </div>
14   )
15 }
16
17
18 export default People
```

client > src > JS App.js > ...

You, 1 minute ago | 1 author (You)

```
1 import './App.css';
2
3 import Person from './components/Person';
4 import People from './components/People';
5
6 function App() {
7   return (
8     <div className="App">
9       <header className="App-header">
10         <Person/>
11         <Person name="Jane Smith" age={25}>/>
12       <People/>
13     </header>
14   );
15 }
16
17
18
19 export default App;
```

REACT.JS STATE

- Component's state is an object that holds data that can change over time and affect the component's behavior and rendering
- State is managed by the component itself, and it can be updated using the `setState()` method
- The component will re-render whenever its state changes
- Component's state should only be modified by the component itself, and not directly from outside the component

REACT.JS STATE

JS Click.js X

```
client > src > components > JS Click.js > ...
1  import React from 'react';
2
3  class Click extends React.Component {
4    constructor(props) {
5      super(props);
6      this.state = { count: props.initialClicks || 0 };
7      this.handleClick = this.handleClick.bind(this);
8    }
9
10   handleClick() {
11     this.setState({ count: this.state.count + 1 });
12   }
13
14   render() {
15     return (
16       <div>
17         <button onClick={this.handleClick}>Click me</button>
18         <p>You have clicked the button {this.state.count} times</p>
19       </div>
20     );
21   }
22 }
23 export default Click;
24
```

JS App.js X

```
client > src > JS App.js > ...
You, 1 minute ago | 1 author (You)
1
2
3  import './App.css';
4
5  import Person from './components/Person';
6  import People from './components/People';
7  import Click from './components/Click';
8
9  function App() {
10    return (
11      <div className="App">
12        <header className="App-header">
13          <Person/>
14          <Person name="Jane Smith" age={25}/>
15
16          <People/>
17
18          <Click/>
19          <Click initialClicks={100}/>
20        </header>
21      </div>
22    );
23  export default App;
```

REACT.JS BOOTSTRAP

- Bootstrap is a popular open-source CSS framework that provides a set of pre-defined CSS classes for styling common user interface elements
- React Bootstrap provides a collection of React components that implement the same styling and behavior as the Bootstrap components, but can be easily integrated into a React application using JSX
- `npm install bootstrap react-bootstrap`

```
client > src > JS App.js > ...
You, 4 seconds ago | 1 author (You)
1 import './App.css';
2 import 'bootstrap/dist/css/bootstrap.css';
```

REACT.JS BOOTSTRAP

JS Click.js

```
client > src > components > JS Click.js > ...
1 import React from 'react';
2 import { Button } from 'react-bootstrap';
3
4 class Click extends React.Component {
5   constructor(props) {
6     super(props);
7     this.state = { count: props.initialClicks || 0 };
8     this.handleClick = this.handleClick.bind(this);
9   }
10
11 handleClick() {
12   this.setState({ count: this.state.count + 1 });
13 }
14
15 render() {
16   return (
17     <div>
18       <Button onClick={this.handleClick} variant="primary">Click Me</Button>
19       <p>You have clicked the button {this.state.count} times</p>
20     </div>
21   );
22 }
23
24 export default Click;
```

JS Person.js

```
client > src > components > JS Person.js > ...
1 import React from 'react';
2 import { Alert } from 'react-bootstrap';
3
4 class Person extends React.Component {
5   static defaultProps = {
6     name: 'John Smith',
7     age: 30
8   };
9   render() {
10     return (
11       <div>
12         <Alert variant="primary">
13           <p>My name is {this.props.name}</p>
14           <p>My age is {this.props.age}</p>
15         </Alert>
16       </div>
17     );
18   }
19
20
21
22 export default Person
```

My name is Drake

My age is 30

My name is Alice

My age is 23

My name is Jason

My age is 17

My name is Kevin

My age is 19

Click Me

You have clicked the button 1 times

Click Me

You have clicked the button 102 times

REACT.JS ROUTER

- React Router is a popular library for building client-side routing in React applications
- Allows you to define a set of URL paths that map to different components in your application
- Different parts of the app can be displayed to the user depending on the current URL
- Can store different pages in a pages folder under src
- `npm install react-router-dom`

REACT.JS ROUTER

HACKATHON

client

> node_modules

> public

< src

components

JS Click.js

JS Navigation.js

JS People.js

JS Person.js

pages

JS Click.js

JS Error.js

JS Home.js

JS People.js

App.css

JS App.js

JS App.test.js

index.css

JS index.js

logo.svg

JS reportWebVitals.js

JS setupTests.js

↳ .gitignore

{ } package-lock.json

{ } package.json

ⓘ README.md

> OUTLINE

> TIMELINE

client > src > JS App.js > ...

```
You, 33 seconds ago | 1 author (You)  
1 import './App.css';  
2 import 'bootstrap/dist/css/bootstrap.css';  
3  
4 import Navigation from './components/Navigation';  
5  
6 import {  
7   BrowserRouter as Router,  
8   Routes,  
9   Route  
} from 'react-router-dom';  
10  
11  
12 import ClickPage from './pages/Click';  
13 import PeoplePage from './pages/People';  
14 import Error from './pages/Error';  
15 import Home from './pages/Home';  
16  
17  
18 function App() {  
19   return (  
20     <div className="App">  
21       <header className="App-header">  
22         <Navigation/>  
23         <Router>  
24           <Routes>  
25             <Route path="/" element={<Home />} />  
26             <Route path="/home" element={<Home />} />  
27             <Route path="/people" element={<PeoplePage />} />  
28             <Route path="/click" element={<ClickPage />} />  
29             <Route path="*" element={<Error />} />  
30           </Routes>  
31         </Router>  
32       </header>  
33     </div>  
34   );  
35 }
```

client > src > components > JS Navigation.js > ...

```
1  import React from 'react';  
2  import { Container } from 'react-bootstrap';  
3  import Nav from 'react-bootstrap/Nav';  
4  import Navbar from 'react-bootstrap/Navbar';  
5  
6  const Navigation = () => {  
7    return (  
8      <Container fluid>  
9        <Navbar bg="primary" variant="dark" fixed="top">  
10          <Container>  
11            <Navbar.Brand href="home">Workshop</Navbar.Brand>  
12            <Nav className="me-auto">  
13              <Nav.Link href="home">Home</Nav.Link>  
14              <Nav.Link href="people">People</Nav.Link>  
15              <Nav.Link href="click">Click Counter</Nav.Link>  
16            </Nav>  
17          </Container>  
18        </Navbar>  
19    </Container>  
20  );  
21}  
22  
23  export default Navigation;
```

REACT.JS ROUTER

client > src > pages > **JS** Home.js > ...

```
1 import React from "react";
2
3 function Home() {
4   return <h1>Home</h1>
5 }
6
7 export default Home;
```

client > src > pages > **JS** Error.js > ...

```
1 import React from "react";
2
3 function Error() {
4   return <div>Error</div>
5 }
6
7 export default Error;
```

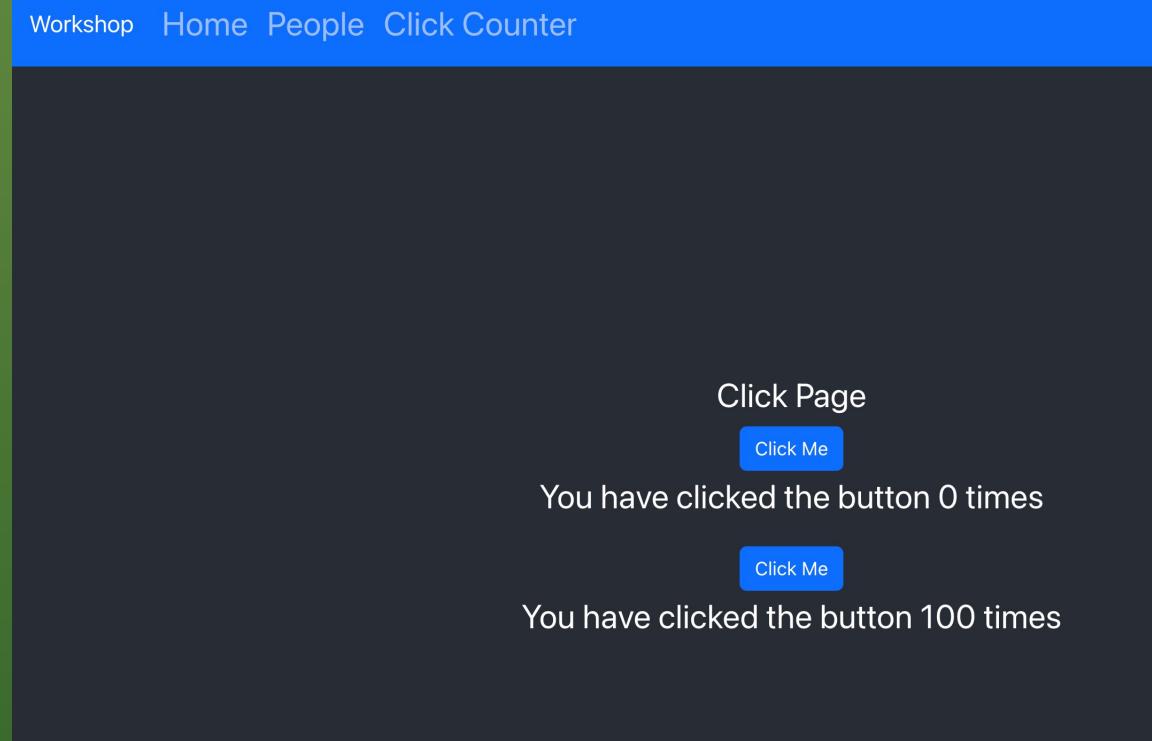
client > src > pages > **JS** People.js > ...

```
1 import React from "react";
2 import Person from '../components/Person';
3 import People from '../components/People';
4
5 function PeoplePage() {
6   return <div>
7     <h1>People Page</h1>
8     <Person/>
9     <Person name="Jane Smith" age={25}>
10    <People/>
11  </div>
12}
13
14 export default PeoplePage;
```

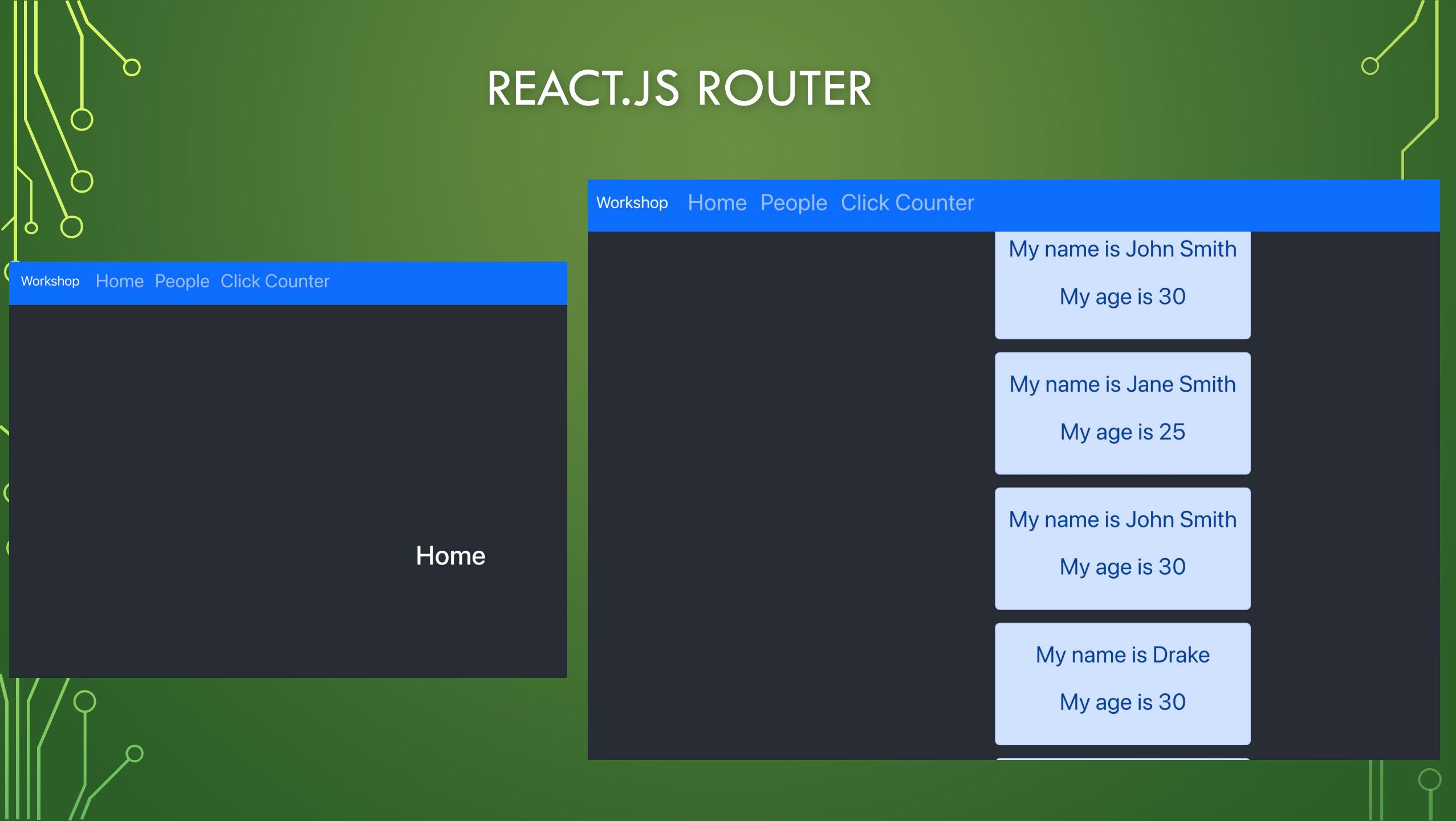
REACT.JS ROUTER

client > src > pages > **JS** Click.js > ...

```
1 import React from "react";
2 import Click from '../components/Click';
3
4 function ClickPage() {
5   return <div>
6     Click Page
7     <Click/>
8     <Click initialClicks={100}/>
9   </div>
10 }
11
12 export default ClickPage;
```



REACT.JS ROUTER



REACT.JS API CALL

client > src > **JS** App.js > ...

```
16 import WikipediaPage from './pages/Wikipedia';
17
18
19 function App() {
20   return (
21     <div className="App">
22       <header className="App-header">
23         <Navigation/>
24         <Router>
25           <Routes>
26             <Route path="/" element={<Home />} />
27             <Route path="/home" element={<Home />} />
28             <Route path="/people" element={<PeoplePage />} />
29             <Route path="/click" element={<ClickPage />} />
30             <Route path="/wikipedia" element={<WikipediaPage />} />
31             <Route path="*" element={<Error />} />
32           </Routes>
33         </Router>
34       </header>
35     </div>
36   );
37 }
```

client > src > pages > **JS** Wikipedia.js > ...

```
1 import React from "react";
2 import Wikipedia from '../components/Wikipedia';
3
4 function WikipediaPage() {
5   return <div>
6   <Wikipedia/>
7   </div>
8 }
9
10 export default WikipediaPage;
```

components > **JS** Navigation.js > [🔗] default

```
<Nav.Link href="home">Home</Nav.Link>
<Nav.Link href="people">People</Nav.Link>
<Nav.Link href="click">Click Counter</Nav.Link>
<Nav.Link href="wikipedia">Wikipedia</Nav.Link>
</Nav>
</Container>
```

REACT.JS API CALL

```
client > src > components > JS Wikipedia.js > Wikipedia
```

```
1 import React, { useState, useEffect } from 'react';
2 import axios from 'axios';
3
4 function Wikipedia() {
5   const [page, setPage] = useState({});
6   const [search, setSearch] = useState('');
7   const [error, setError] = useState(null);
8
9   useEffect(() => {
10     async function fetchData() {
11       try {
12         const response = await axios.get(
13           `https://en.wikipedia.org/w/api.php?action=query&format=json&origin=*&prop=extracts&exintro=&explaintext=&titles=${search}`);
14         setPage(response.data.query.pages);
15       } catch (err) {
16         setError(err);
17       }
18     }
19     if (search) {
20       fetchData();
21     }
22   }, [search]);
23
24   if (error) {
25     return <p>{error.message}</p>;
26   }
}
```

- Axios is a library used to make HTTP requests
- npm install axios

```
client > src > components > JS Wikipedia.js > ...
```

```
28   return (
29     <div>
30       <h2>Wikipedia</h2>
31       <form>
32         <label>
33           Search:
34           <input type="text" value={search} onChange={e => setSearch(e.target.value)} />
35         </label>
36         <button type="submit" onClick={e => e.preventDefault()}>Search</button>
37       </form>
38       {Object.values(page).map((p) => (
39         <div key={p.pageid}>
40           <h3>{p.title}</h3>
41           <p>{p.extract}</p>
42         </div>
43       )));
44     </div>
45   );
46 }
47
48 export default Wikipedia;
```

REACT.JS API CALL

Workshop Home People Click Counter Wikipedia

Wikipedia

Search:

JavaScript

JavaScript (), often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O. JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js. Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

SUMMARY

In this lesson we learned:

- NPM / Dev Environment Setup
- React.JS
- Components
- Navigation
- Props
- State
- Bootstrap
- API calls

REFERENCES

- <https://reactjs.org/docs/getting-started.html>
- <https://reactjs.org/docs/components-and-props.html>
- <https://react-bootstrap.github.io/components/alerts>
- https://www.w3schools.com/REACT/react_components.asp
- <https://github.com/SohailHanif/React>