14.employee

Objectives

- **Explain the need and Benefits of React Context API**
- **Working with createContext()**
- **List the types of Router Components**

In this hands-on lab, you will learn how to:

- **Create a context to be used by child components**
- **Create a provider and consumer of the context**

Prerequisites

The following is required to complete this hands-on lab:

- **Node.js**
- **NPM**
- **Visual Studio Code**

Notes

Estimated time to complete this lab: 30 minutes.

Developers of Apps Centric Solutions have created an employee management application which supports light and dark themes for the buttons. The current solution uses the react state and props to provide the theme name to be used from App component to Employee List component and from there to Employee Card component. Quality assurance team analyzed the solutions and found the technique being used to be a substandard one. React architect

suggested to use the react context API to share the theme name with nested child components instead of passing them down using props from the parent component.

You are assigned the task of converting the application form props only to React Context API.

Application can be downloaded from below

1. **Unzip the application and open it using VS Code**
2. **Go to terminal and execute *npm install* command to restore all the node modules**

Figure 1: Restore node modules

3. **Run the application once to see the output. Use npm start command.**

Figure 2: Starting application

4. **Explore the components present in App.js, EmployeesList.js and EmployeeCard.js files.**
5. **Create a new file with the name as ThemeContext.js. Define a new context in the file with the name as ThemeContext and assign it a default value of 'light' and export it as default form the module.**
6. **Open App component present in App.js file.**
    a. **Import the ThemeContext in App component.**

b. **Define the theme context provider to be the entire JSX of the App component.**

c. **Assign the value for the theme provider from the state of the component.**

d. **Modify the call to EmployeeList component so that theme name is no longer passed as props.**

7. **Go to EmployeeList component present in EmployeeList.js file and modify it so that theme name is not passed explicitly to its child component.**

8. **Go to EmployeeCard component inside EmployeeCard.js file**

    a. **Import the ThemeContext into the component file**

    b. **Retrieve the value of the context with the help of useContext() and store it in a variable**

    c. **Use the variable to pass the className for the buttons.**

CODE:

App.js

```
import React from 'react';
import ThemeContext from './ThemeContext';          // Import the context
import EmployeeList from './components/EmployeeList'; // Import child component

class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      theme: 'dark' // or 'light'
    };
```

```
  }

  render() {
    return (
      <ThemeContext.Provider value={this.state.theme}>
        <div className={`app-container ${this.state.theme}`}>
          <h1>Employee Management App</h1>
          <EmployeeList />
        </div>
      </ThemeContext.Provider>
    );
  }
}

export default App;
```

EmployeeCard.js

```
import React, { useContext } from 'react';
import ThemeContext from '../ThemeContext';

function EmployeeCard() {
  const theme = useContext(ThemeContext); // get theme value from context

  return (
    <div>
```

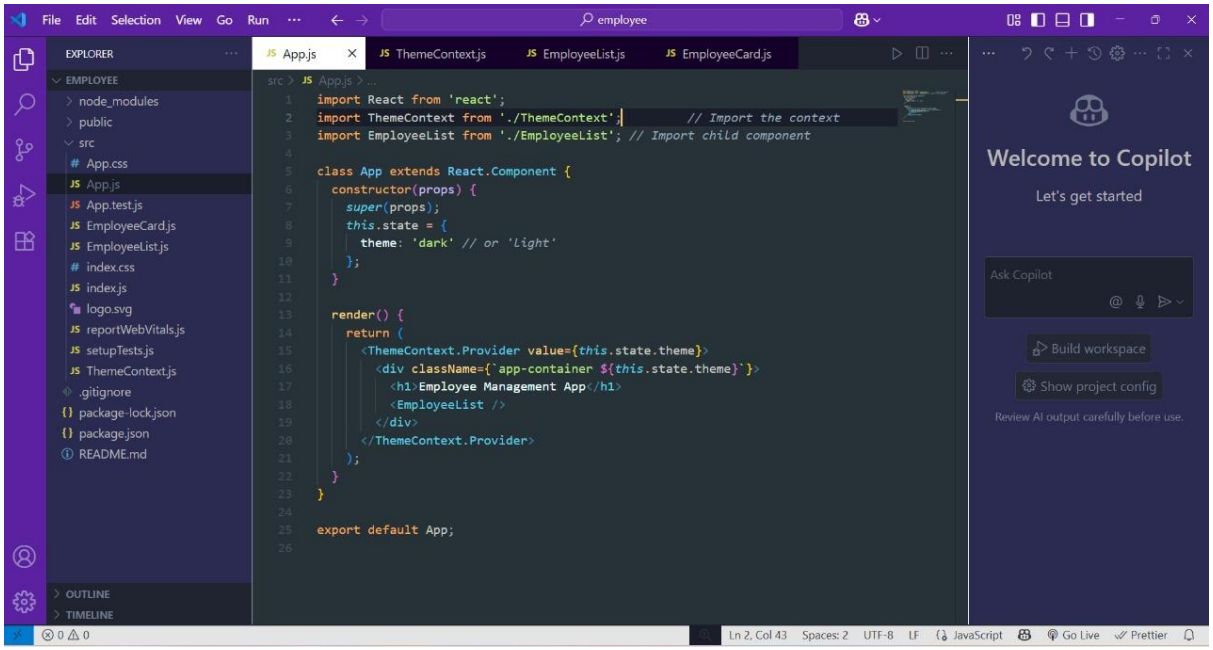
```
      <h3>John Doe</h3>
      <button className={theme}>Contact</button>
    </div>
  );
}

export default EmployeeCard;
```

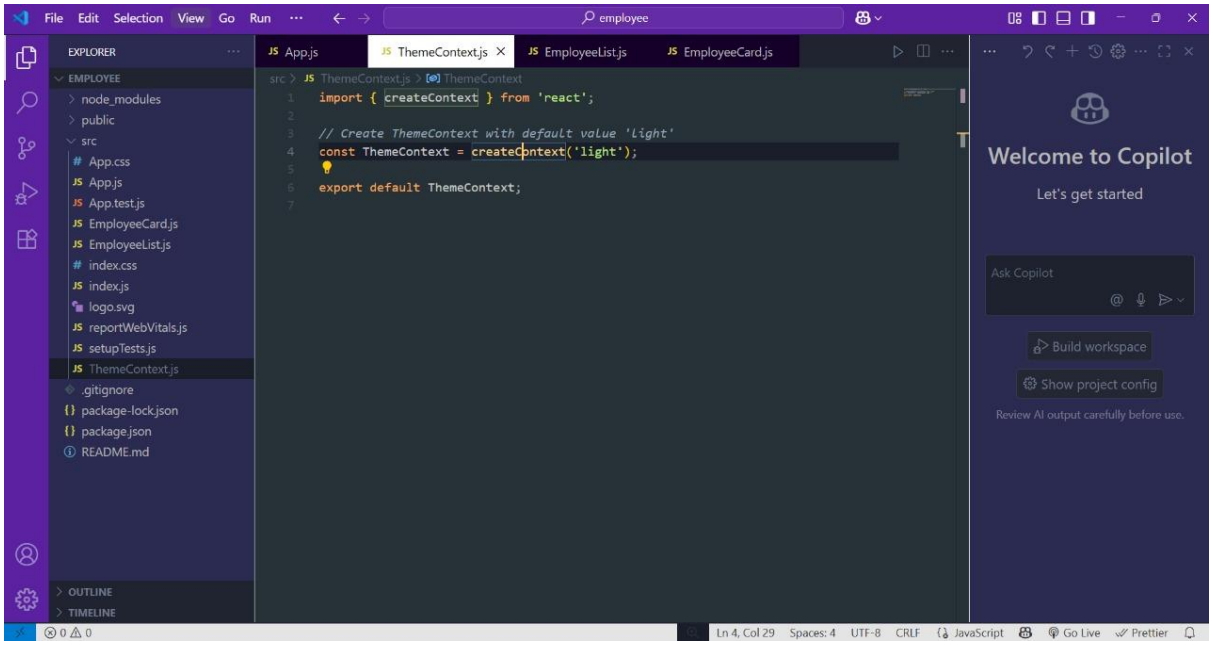
EmployeeList.js

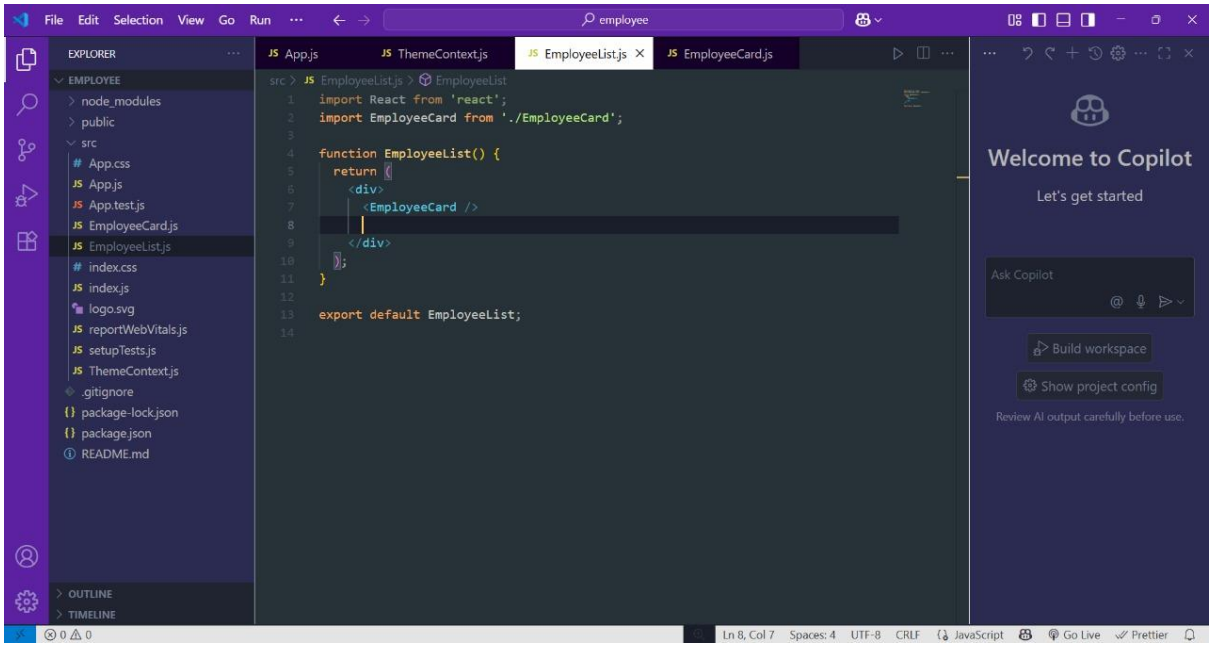
```
import React from 'react';
import EmployeeCard from './EmployeeCard';

function EmployeeList() {
  return (
    <div>
      <EmployeeCard />
      <EmployeeCard />
    </div>
  );
}

export default EmployeeList;
```

ThemeContext.js

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

const ThemeContext = createContext('light'); // default value

export default ThemeContext;
```

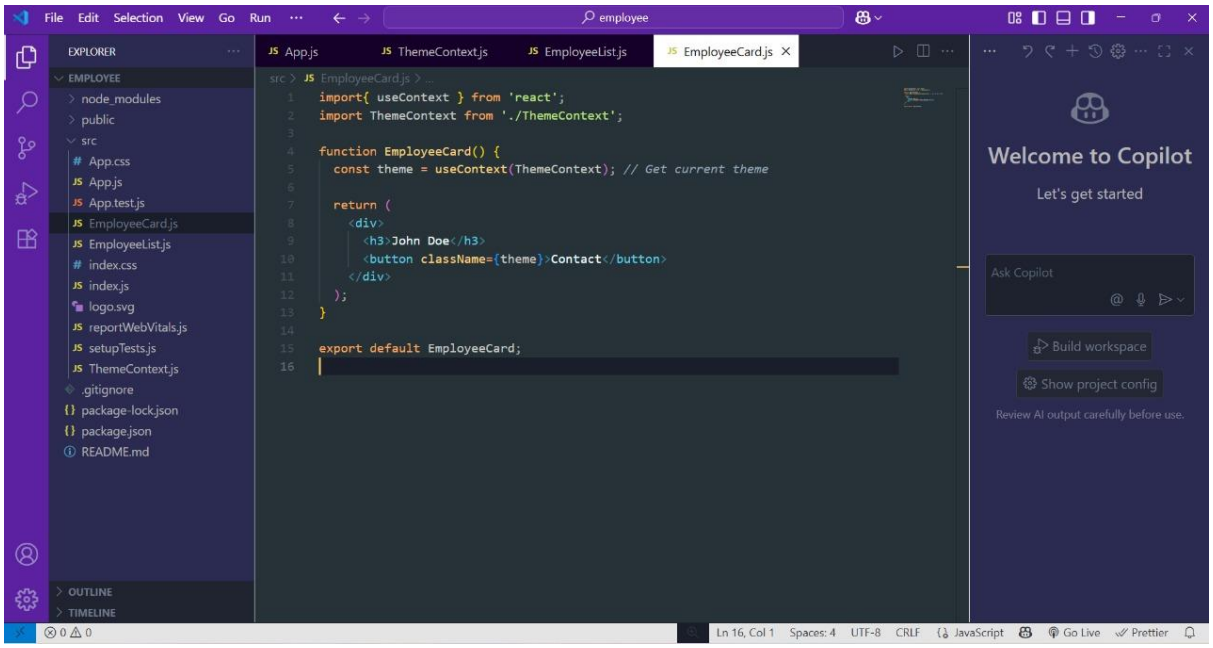





File Edit Selection View Go Run
employee
EXPLORER
EMPLOYEE
node_modules
public
src
App.css
App.js
App.test.js
EmployeeCard.js
EmployeeList.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
ThemeContext.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
App.js
ThemeContext.js
EmployeeList.js
EmployeeCard.js
src > EmployeeList.js > EmployeeList
import React from 'react';
import EmployeeCard from './EmployeeCard';
function EmployeeList() {
return (

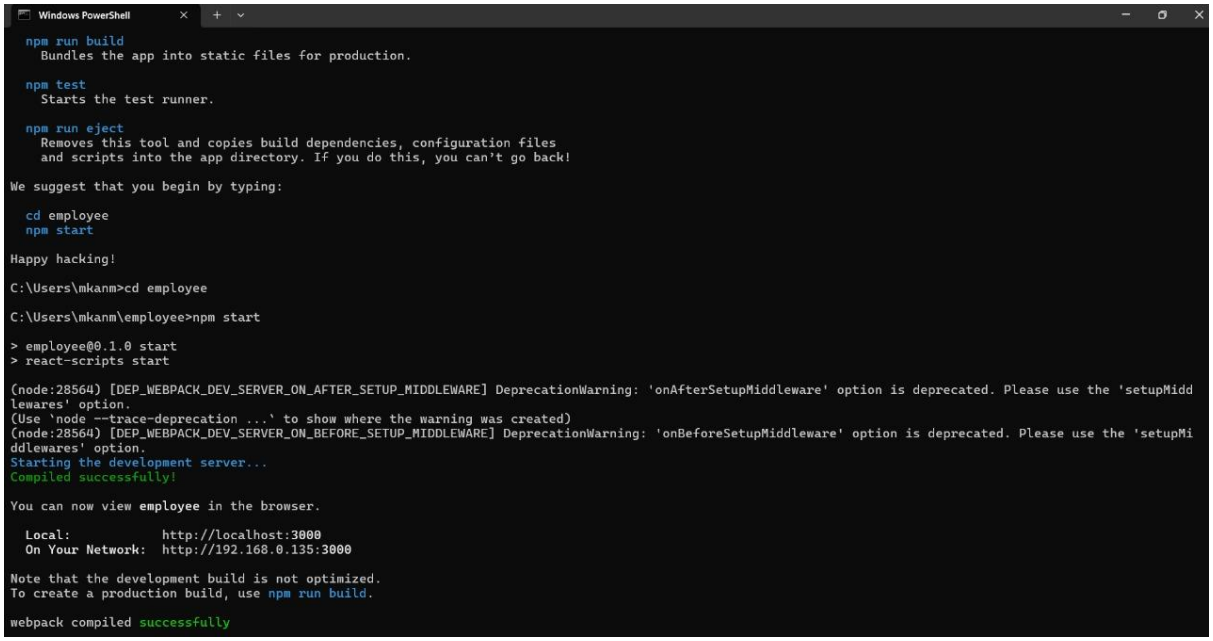
<EmployeeCard />

);
}
export default EmployeeList;
Welcome to Copilot
Let's get started
Ask Copilot
Build workspace
Show project config
Review AI output carefully before use.
Ln 8, Col 7
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Prettier

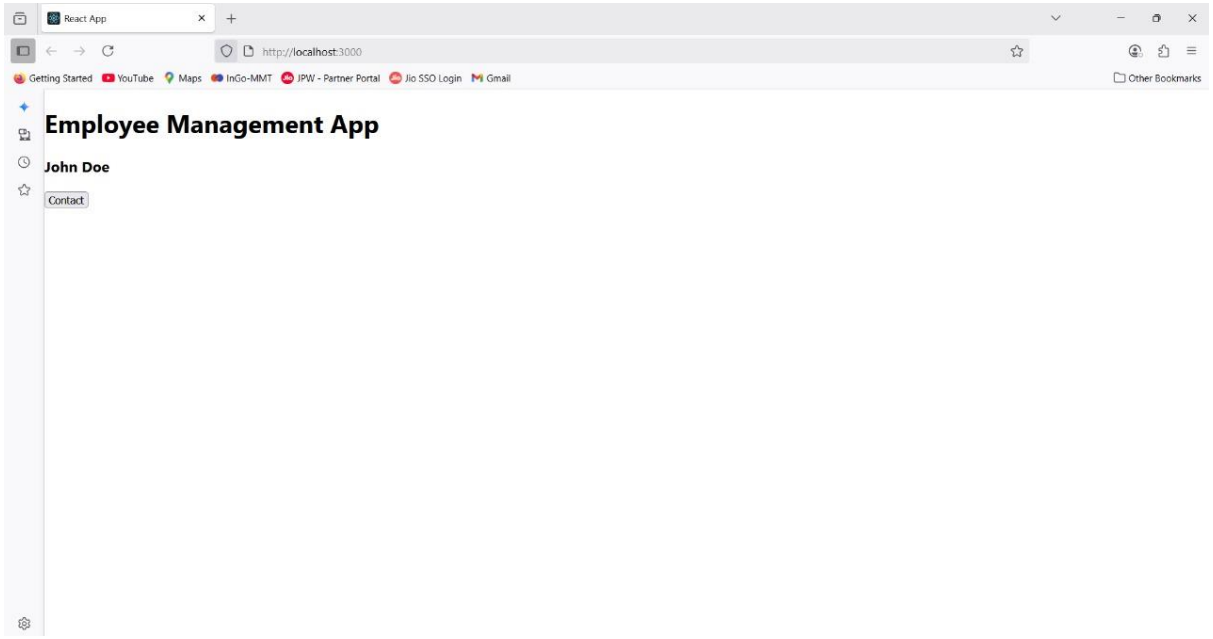

File Edit Selection View Go Run
employee
EXPLORER
EMPLOYEE
node_modules
public
src
App.css
App.js
App.test.js
EmployeeCard.js
EmployeeList.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
ThemeContext.js
.gitignore
package-lock.json
package.json
README.md
OUTLINE
TIMELINE
App.js
ThemeContext.js
EmployeeList.js
EmployeeCard.js
src > EmployeeCard.js > ...
import{ useContext } from 'react';
import ThemeContext from './ThemeContext';
function EmployeeCard() {
const theme = useContext(ThemeContext); // Get current theme
return (

<h3>John Doe</h3>
<button className={theme}>Contact</button>

);
}
export default EmployeeCard;
Welcome to Copilot
Let's get started
Ask Copilot
Build workspace
Show project config
Review AI output carefully before use.
Ln 16, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Prettier



OUTPUT:



KANMANI MURUGHAIYAN

SUPERSET ID: 6407636