

1. The React Context API is a feature that provides a way to pass data through the component tree without having to manually pass props down at every level. It is needed to solve a problem known as "prop drilling," where data is passed through multiple layers of components that do not need the data themselves, leading to verbose and difficult-to-maintain code.

The benefits of using the React Context API include:

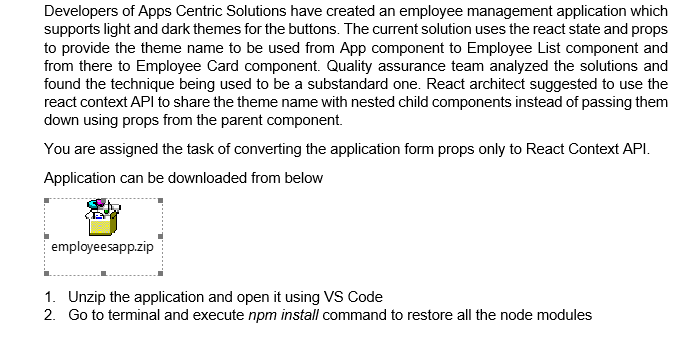
* State Management: It provides a simple and efficient way to manage global state in a React application, such as user authentication, themes, or language preferences.
* Reduced Prop Drilling: It eliminates the need to pass props through intermediate components, making the code cleaner and easier to read.
* Improved Reusability: By making data accessible to any component within the context's scope, it promotes the creation of reusable components that are not tightly coupled to a specific data source.
* Decoupling Components: It helps decouple components from their parent components, as they no longer need to rely on props for data, making them more independent and easier to test.

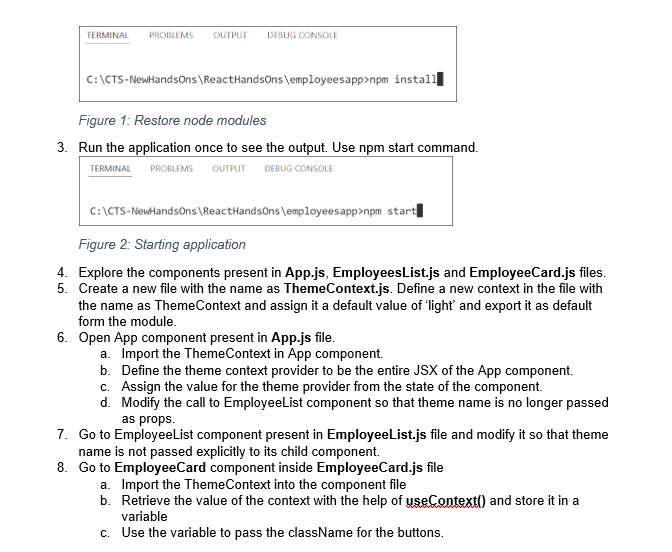
1. The createContext() function is a core part of the React Context API. It is used to create a new context object. When called, it returns an object with two properties: a Provider and a Consumer.

* Provider: The Provider component is used to make the context data available to its child components. It accepts a value prop, which is the data that you want to share. All components wrapped within the Provider will have access to this value.
* Consumer: The Consumer component allows child components to subscribe to changes in the context. However, with the introduction of the useContext Hook, the Consumer is now less commonly used. The useContext Hook provides a cleaner and more direct way to access the context value within a functional component.

1. When working with a library like React Router, there are different types of router components that are used to enable navigation within a React application. The two primary types are:

* BrowserRouter: This type of router uses the HTML5 history API (pushState, replaceState, etc.) to keep our UI in sync with the URL. It is the most common router for web applications and is suitable for all modern browsers. It creates clean URLs (e.g., www.example.com/about).
* HashRouter: This router uses the hash portion of the URL (e.g., www.example.com/#/about) to keep our UI in sync with the URL. It is primarily used for static websites or when the server configuration cannot be changed to handle clean URLs.





**Solution**

**App.js**

import logo from './logo.svg';

import './App.css';

import { EmployeesData } from './Employee';

import EmployeesList from './EmployeesList';

import { useState } from 'react';

import ThemeContext from './ThemeContext';

function App() {

const Employees = EmployeesData;

const [theme, setTheme] = useState('light');

return (

<>

<div>

<label>SELECT A THEME </label>

<select onChange={(e) => setTheme(e.target.value)}>

<option value='light'>Light</option>

<option value='dark'>Dark</option>

</select>

</div>

<ThemeContext.Provider value={theme}>

<EmployeesList employees={Employees} />

</ThemeContext.Provider>

</>

);

}

export default App;

**EmployeeList.js**

import React, { useContext } from 'react';

import EmployeeCard from './EmployeeCard';

import ThemeContext from './ThemeContext';

function EmployeesList({ employees }) {

const theme = useContext(ThemeContext);

return (

<div>

<h1>Employees List</h1>

{

employees.map(employee => (

<EmployeeCard employee={employee} key={employee.id} />

))

}

</div>

);

}

export default EmployeesList;

**EmployeeCard.js**

import { useContext } from 'react';

import Styles from './EmployeeCard.module.css';

import ThemeContext from './ThemeContext';

function EmployeeCard(props) {

const theme = useContext(ThemeContext);

return (

<div className={Styles.Card}>

<h3>{props.employee.name}</h3>

<p>{props.employee.email}</p>

<p>{props.employee.phone}</p>

<p>

<a href="#" className={theme}>Edit</a>

<a href="#" className={theme}>Delete</a>

</p>

</div>

);

}

export default EmployeeCard;

**ThemeContext.js**

import { createContext } from 'react';

const ThemeContext = createContext('light');

export default ThemeContext;

**Output**

