React Context API

Introduction to Context API

What is the Context API?

- The Context API allows you to pass data through the component tree without having to pass props manually at every level.
- Useful for themes, user authentication, and other global settings.

When to use the Context API?

- When you need to share data that is considered global for a tree of React components.
- To avoid prop drilling, where you pass data through many intermediate components.

Steps to create and use Context:

1. **Create Context:** Use createContext() to create a context.
2. **Provide Context:** Use Context.Provider in the parent component to supply the context value.
3. **Consume Context:** Use useContext in the child component to access the context value.

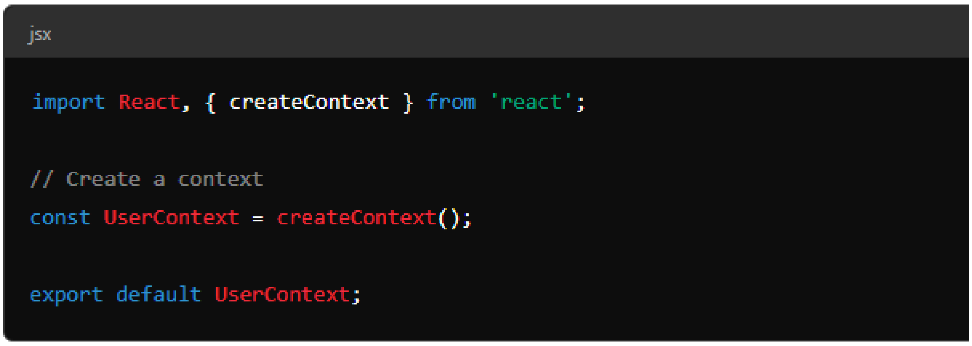
This example shows how to pass user data from a parent component to a child component using the Context API, simplifying the data flow compared to prop drilling.

Example: Passing User Data from Parent to Child

Step 1: Create Context

- Define a context to hold user data.

UserContext.js

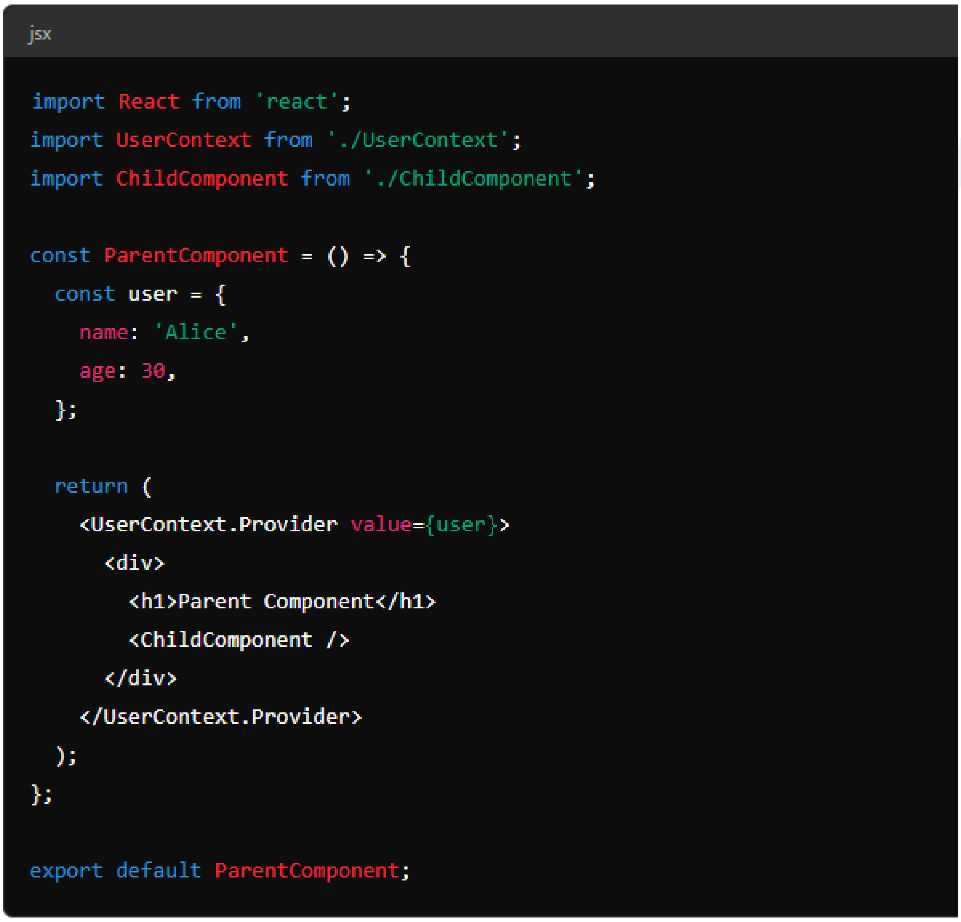


```jsx
import React, { createContext } from 'react';

// Create a context
const UserContext = createContext();

export default UserContext;
```

Step 2: Provide Context

- Use UserContext.Provider in the parent component to supply the context value.

ParentComponent.js

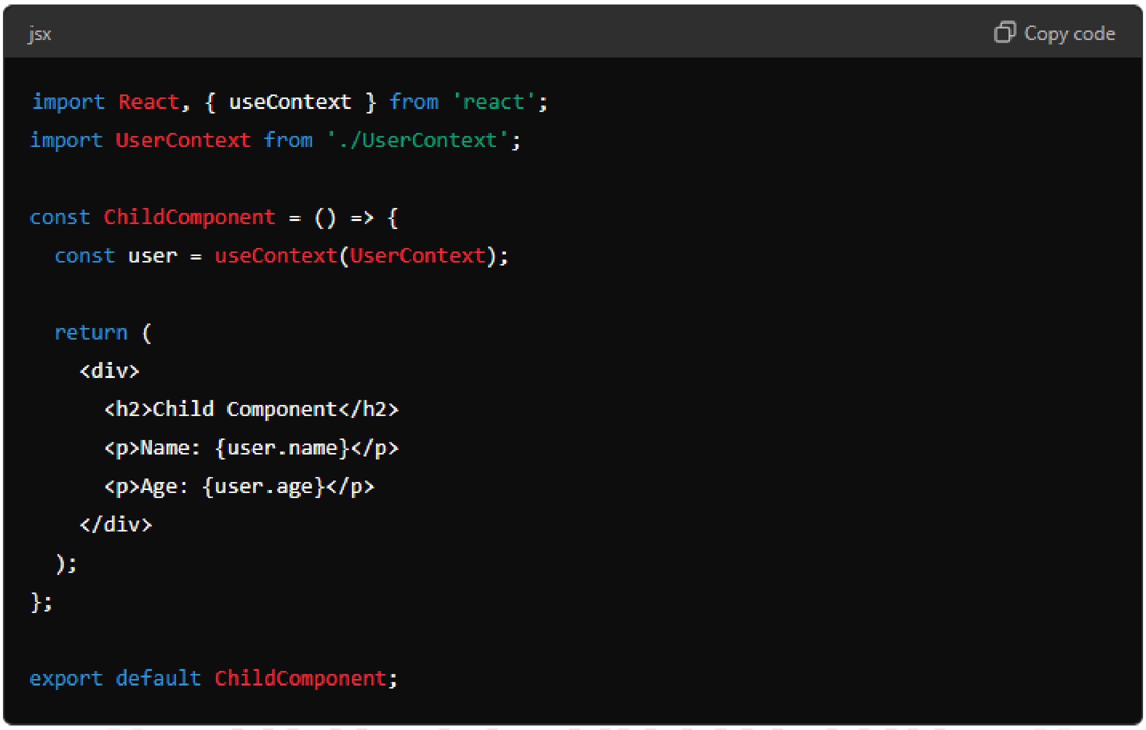


```jsx
import React from 'react';
import UserContext from './UserContext';
import ChildComponent from './ChildComponent';

const ParentComponent = () => {
  const user = {
    name: 'Alice',
    age: 30,
  };

  return (
    <UserContext.Provider value={user}>
      <div>
        <h1>Parent Component</h1>
        <ChildComponent />
      </div>
    </UserContext.Provider>
  );
};

export default ParentComponent;
```

Step 3: Consume Context

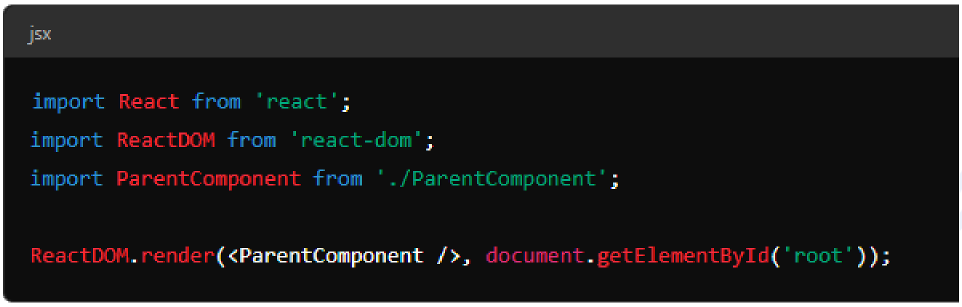
- Access the context value in the child component using useContext.

ChildComponent.js



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

const ChildComponent = () => {
  const user = useContext(UserContext);

  return (
    <div>
      <h2>Child Component</h2>
      <p>Name: {user.name}</p>
      <p>Age: {user.age}</p>
    </div>
  );
};

export default ChildComponent;
```

index.js



```jsx
import React from 'react';
import ReactDOM from 'react-dom';
import ParentComponent from './ParentComponent';

ReactDOM.render(<ParentComponent />, document.getElementById('root'));
```