React Managing Object State with useState Assignment

React Assignment: Managing Object State with `useState`

Objective:

Create a React functional component named `OrderStatus` that utilizes the `useState` hook to manage and update an object state. This component will simulate a basic order tracking system, allowing users to update the status of an order through a user interface.

Skills Practiced:

- Using the `useState` hook to manage object states
- Updating state objects in React
- Handling form inputs and events

Requirements:

- A basic understanding of React and functional components
- Familiarity with the `useState` hook and event handling in React

Assignment Instructions:

Task 1: Setting Up the `OrderStatus` Component

1. In the `src` directory of your existing React app, create a new file named `OrderStatus.js`.
2. Start by importing React and the `useState` hook:

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

React Managing Object State with useState Assignment

3. Define the `OrderStatus` functional component. Initially, use `useState` to set up an order state with the following structure:

```jsx
const [order, setOrder] = useState({
  id: 'ORD123',
  status: 'Processing',
  customer: 'John Doe',
  items: 3
});
```

4. Render the current order details in your component.

Task 2: Creating the Update Interface

1. Add a dropdown (`<select>`) element that allows the user to update the order status. Include options such as "Processing", "Shipped", and "Delivered".

2. Add a button labeled "Update Status" which, when clicked, updates the order's status based on the selected option.

Task 3: Implementing the Update Logic

1. Write a function `handleStatusChange` that updates the `order` state's `status` property based on the selected dropdown value.

2. Ensure that the `handleStatusChange` function only updates the `status` property of the order, keeping the other properties unchanged. This can be achieved using the spread operator to maintain the rest of the order object intact.

React Managing Object State with useState Assignment

Task 4: Displaying Updated Order Details

1. After the status update, the component should re-render to display the updated order details.

2. Test your component by selecting different statuses and clicking the "Update Status" button. Verify that the order's status updates accordingly without altering other order details.

Deliverables:

- The `OrderStatus.js` file containing your component code.

- Comments within your code explaining your logic, particularly how state is managed and updated.

Evaluation Criteria:

- Correct implementation of the `useState` hook to manage an object state.

- Successful updating of the state's property without affecting other properties.

- Proper handling of user inputs and events to update the state.

- Code quality, including readability, proper use of React conventions, and comments.

This assignment reinforces the concept of managing complex state using the `useState` hook in functional components, focusing on object states. It provides a practical scenario that simulates real-world application features, enhancing your skills in state management and React application development.