React Project: ShoppyGlobe E-commerce Application (250 marks)

Objective: Create a basic e-commerce application named **ShoppyGlobe**.

Requirements:

Component Structure: (20 marks)

- App: The main component.
- Header: Display the navigation menu and shopping cart icon
- ProductList: Displays a list of products.
- ProductItem: Represents a single product including an "Add to Cart" button.
- Product Detail: Show detailed information about a selected product.
- Cart: Displays the items added to the cart with options to modify quantities or remove items.
- CartItem: Represents a single item in the cart.
- NotFound: Display a 404 page for unknown routes.

Props: (10 marks)

- Utilize props to pass data from parent components to child components.
- Ensure components are functional and reusable with appropriate prop types.

Data Fetching with useEffect: (40 marks)

- ProductList Component: Use useEffect to fetch the list of products from an API endpoint ('https://dummyjson.com/products') when the component mounts.
 Store the fetched data in the component's state.Create custom hook for fetching product list. (20 marks)
- ProductDetail Component: Use useEffect to fetch details of a selected product based on route parameters when the component mounts. Store the fetched data in the component's state. (10 marks)
- Error Handling: Implement error handling to manage failed data fetch requests gracefully. (10 marks)

State Management: (70 marks)

- Redux: Implement Redux for more complex state management.
 Create actions, reducers, and selectors to manage the state of cart items.
 (50 marks)
- Implement a search feature to filter products in the ProductList. (20 marks)

Event Handling: (20 marks)

- Add a button in each ProductItem to add the product to the cart.
- Add a button in each CartItem to remove the product from the cart.
 Ensure that add product and remove product functionality is implemented correctly using redux.

React Routing: (20 marks)

- Implement routing using React Router.
- o Create routes for Home, Product Detail, Cart, and Checkout pages.
- Use route parameters for product details.

React Lists: (10 marks)

- Render the list of products in the ProductList component.
- Render the list of cart items in the Cart component.
- o Ensure that you provide a unique key to each list.

Performance Optimization: (20 marks)

 Implement code splitting and lazy loading for components using React.lazy and Suspense.

Styling: (20 marks)

- o Apply css for styling.
- Ensure the application is responsive and works well on different screen sizes.

Submission Guidelines: (20 marks)

- Ensure your application runs without errors. (10 marks)
- Use proper indentation and comments to explain your code. (5 marks)
- Submit a link to your GitHub repository with the completed project. (5 marks)