NAME : PRAGHAADEESH R

ROLL NO : CH.EN.U4AIE21035

Case Study 2: E-Commerce Product Listing Page

1. How would you fetch and display a list of products from an API in a React component?

To fetch and display a list of products from an API in a React component, I would use the useEffect hook to make an API call when the component mounts. Inside the useEffect, I'd use the fetch method or a library like axios to request the product data from the API. Upon receiving the data, I would store it in the component's state using the useState hook. The state would then be mapped to dynamically render the list of products on the page.

2. Describe how you would implement search and filter functionalities in the product listing page.

To implement search and filter functionalities, I would create a search input and various filter options (e.g., categories, price ranges) in the UI. The user's input and filter selections would be managed using state variables. I would filter the list of products in real-time by applying the search query and selected filters to the product data. The filtered product list would then be displayed by mapping over the updated state.

3. How can you use React Router to navigate between different pages in the application?

React Router can be used to handle navigation by defining routes in the application. I would wrap the component tree with the BrowserRouter and set up routes using the Route component for each page (e.g., home, product listing, product details). The Link or NavLink components would be used to create navigational links, and the useNavigate hook or Navigate component can programmatically redirect users to different pages based on certain actions.

4. Explain how to use Styled-Components to style the product listing page.

To style the product listing page using Styled-Components, I would create styled versions of React components by using the styled function. For example, I'd define a styled container for the product grid, styled cards for individual products, and other UI elements like buttons or headings. These styled components would then be used in the React component structure, allowing styles to be scoped and applied directly to elements while maintaining the benefits of CSS-in-JS, such as dynamic theming and automatic vendor prefixing.

5. What techniques would you use to optimize the loading time of the product images?

To optimize the loading time of product images, I would employ several techniques:

- **Lazy Loading:** Use the loading="lazy" attribute on image elements or a library like react-lazyload to only load images as they enter the viewport.
- **Image Compression:** Compress images to reduce their file size while maintaining quality.

- **Responsive Images:** Use the srcset attribute to provide different image sizes for different screen resolutions.
- **CDN Delivery:** Serve images from a Content Delivery Network (CDN) to reduce latency.
- **Caching:** Use browser caching and HTTP headers to store images locally, reducing load times for repeat visits.