

index.js:

```
import React from "react";  
import ReactDOM from "react-dom";  
import "./styles.css";  
import App from "./App";
```

```
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById("root")  
);
```

App.js:

```
import React, { useEffect, useState } from "react"; // Importing React and hooks
```

```
function App() {  
  // State variables  
  
  const [products, setProducts] = useState([]); // To store fetched products  
  const [loading, setLoading] = useState(true); // To track loading state  
  const [error, setError] = useState(null); // To store any error  
  
  // Effect to fetch products when the component mounts  
  useEffect(() => {  
    const fetchProducts = async () => {  
      try {  
        // Fetching products from the API
```

```
const response = await fetch("https://fakestoreapi.com/products");
if (!response.ok) {
  throw new Error("Network response was not ok"); // Handle error
}

const data = await response.json(); // Parse JSON response
setProducts(data); // Update state with fetched products
} catch (error) {
  setError(error.message); // Update state with error message
} finally {
  setLoading(false); // Set loading to false
}
};

fetchProducts(); // Call the fetch function
}, []); // Empty dependency array means it runs once when the component mounts

// If loading, show loading message
if (loading) {
  return <div>Loading...</div>;
}

// If there's an error, show error message
if (error) {
  return <div>Error: {error}</div>;
}
```

```

return (
  <div className="product-listing">
    <h1>Product Listing</h1>
    <div className="products">
      {products.map((product) => (
        <div key={product.id} className="product-card">
          {" "}
          {/* Product card without random color */}
          <img
            src={product.image}
            alt={product.title} // Alt text for accessibility
            style={{ width: "100px", height: "100px" }} // Image dimensions
          />
          <h2>{product.title}</h2> {/* Product title */}
          <p>${product.price}</p> {/* Product price */}
          <p className="category">{product.category}</p>{" "}
          {/* Product category */}
          <button>Add to Cart</button> {/* Button to add to cart */}
        </div>
      ))}
    </div>
  </div>
);
}

```

```

export default App;

```

Styles.css:

```
.product-listing{  
    max-width: 800px; /* Maximum width for the listing */  
    margin: 0 auto; /* Center the listing */  
    padding: 20px; /* Padding around the listing */  
}  
  
.products {  
    display: grid; /* Use grid layout for products */  
    grid-template-columns: repeat(  
        auto-fill,  
        minmax(200px, 1fr)  
    ); /* Responsive grid */  
    gap: 15px; /* Space between product cards */  
}  
  
.product-card {  
    border: 1px solid #ccc; /* Light gray border */  
    border-radius: 5px; /* Rounded corners */  
    padding: 10px; /* Padding inside each card */  
    text-align: center; /* Center text inside cards */  
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1); /* Shadow effect */  
    transition: transform 0.2s; /* Smooth transition on hover */  
    background-color: #ffffff; /* White background color for consistency */  
}
```

```
.product-card:hover {
  transform: scale(1.05); /* Slightly enlarge on hover */
}
```

```
.product-card img {
  border-radius: 5px; /* Rounded corners for images */
}
```

```
.category {
  font-style: italic; /* Italic text for category */
  color: #555; /* Dark gray color for category text */
  margin-top: 5px; /* Space above category text */
}
```

Output:

