Case Sudy1:

```
import { useEffect, useState } from "react";


const ProductDetails = ({ productId, onBack }) => {
  const [product, setProduct] = useState(null);
  const [loading, setLoading] = useState(false);


  useEffect(() => {
    const fetchProduct = async () => {
      setLoading(true);
      const res = await fetch(`https://fakestoreapi.com/products/${productId}`);
      const data = await res.json();
      setProduct(data);
      setLoading(false);
    };
    fetchProduct();
  }, [productId]);


  if (loading) return <p>Loading...</p>;


  return (
    product && (
      <div>
        <button onClick={onBack}>⬅ Back to Products</button>
        <h2>{product.title}</h2>
        <img src={product.image} alt={product.title} width="120" />
        <p>{product.description}</p>
        <p>Price: ${product.price}</p>
        <p>
          Rating: {product.rating?.rate} ({product.rating?.count} reviews)
        </p>
      </div>
    )
  );
};

const ProductListing = () => {
  const [products, setProducts] = useState([]);
  const [category, setCategory] = useState("all");
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState("");
  const [selectedProductId, setSelectedProductId] = useState(null);

  useEffect(() => {
    const fetchProducts = async () => {
      setLoading(true);
      setError("");
      try {
        const url =
          category === "all"
            ? "https://fakestoreapi.com/products"
            : `https://fakestoreapi.com/products/category/${category}`;
        const res = await fetch(url);
        if (!res.ok) throw new Error("API Failed");
        const data = await res.json();
        setProducts(data);
      } catch (err) {
        setError(err.message);
      } finally {
        setLoading(false);
      }
    };
    fetchProducts();
  }, [category]);

  if (selectedProductId) {
    return (
      <ProductDetails
        productId={selectedProductId}
        onBack={() => setSelectedProductId(null)}
      />
    );
  }

  return (
    <div>
      <h1>E-Commerce Store</h1>

      <select value={category} onChange={(e) => setCategory(e.target.value)}>
        <option value="all">All</option>
        <option value="electronics">Electronics</option>
        <option value="jewelery">Jewelery</option>
        <option value="men's clothing">Men's Clothing</option>
        <option value="women's clothing">Women's Clothing</option>
      </select>

      {loading && <p>Loading...</p>}
      {error && <p>{error}</p>}

      <ul>
        {products.map((p) => (
          <li key={p.id} onClick={() => setSelectedProductId(p.id)}>
            <img src={p.image} alt={p.title} width="50" />
            <p>{p.title}</p>
            <p>${p.price}</p>
          </li>
        ))}
      </ul>
    </div>
  );
};

export default ProductListing;
```

Casestudty2:Shopping Cart

```
import { useEffect, useState } from "react";

const ProductDetails = ({ productId }) => {
  const [product, setProduct] = useState(null);
  const [loading, setLoading] = useState(false);

  useEffect(() => {
    const fetchProduct = async () => {
      setLoading(true);
      const res = await fetch(`https://fakestoreapi.com/products/${productId}`);
      const data = await res.json();
      setProduct(data);
      setLoading(false);
    };
    fetchProduct();
  }, [productId]);

  if (loading) return <p>Loading...</p>;

  return (
    product && (
      <div>
        <img src={product.image} alt={product.title} width="120" />
        <h2>{product.title}</h2>
        <p>{product.description}</p>
        <p>Price: ${product.price}</p>
        <p>
          Rating: {product.rating?.rate} ({product.rating?.count} reviews)
        </p>
      </div>
    )
  );
};

export default ProductDetails;
```

Casestudy3:User Management Dashboard:

```
import { useEffect, useState } from "react";

const UserDetails = ({ userId, onBack }) => {
  const [details, setDetails] = useState(null);

  useEffect(() => {
    fetch(`https://jsonplaceholder.typicode.com/users/${userId}`)
      .then((res) => res.json())
      .then(setDetails);
  }, [userId]);

  if (!details) return <p>Loading user details...</p>;

  return (
    <div>
      <button onClick={onBack}>⬅ Back to Users</button>
      <h2>{details.name}</h2>
      <p>Email: {details.email}</p>
      <p>Phone: {details.phone}</p>
      <p>Website: {details.website}</p>
    </div>
  );
};

const Users = () => {
  const [users, setUsers] = useState([]);
  const [selectedUserId, setSelectedUserId] = useState(null);

  useEffect(() => {
    fetch("https://jsonplaceholder.typicode.com/users")
      .then((res) => res.json())
      .then(setUsers);
  }, []);

  if (selectedUserId) {
    return <UserDetails userId={selectedUserId} onBack={() => setSelectedUserId(null)} />;
  }

  return (
    <div>
      <h1>User Management</h1>
      <ul>
        {users.map((u) => (
          <li key={u.id} onClick={() => setSelectedUserId(u.id)}>
            {u.name} ({u.email})
          </li>
        ))}
      </ul>
    </div>
  );
};

export default Users;
```

Casestudy4:Posts with comments

```
import { useEffect, useState } from "react";

const PostDetails = ({ postId, onBack }) => {
  const [comments, setComments] = useState([]);
  const [post, setPost] = useState(null);

  useEffect(() => {
    fetch(`https://jsonplaceholder.typicode.com/posts/${postId}`)
      .then((res) => res.json())
      .then(setPost);

    fetch(`https://jsonplaceholder.typicode.com/posts/${postId}/comments`)
      .then((res) => res.json())
      .then(setComments);
  }, [postId]);

  if (!post) return <p>Loading post...</p>;

  return (
    <div>
      <button onClick={onBack}>⬅ Back to Posts</button>
      <h2>{post.title}</h2>
      <p>{post.body}</p>
      <h3>Comments</h3>
      <ul>
        {comments.map((c) => (
          <li key={c.id}>
```

```
            <strong>{c.email}</strong>: {c.body}
          </li>
        ))}
      </ul>
    </div>
  );
};

const Posts = () => {
  const [posts, setPosts] = useState([]);
  const [selectedPostId, setSelectedPostId] = useState(null);

  useEffect(() => {
    fetch("https://jsonplaceholder.typicode.com/posts")
      .then((res) => res.json())
      .then(setPosts);
  }, []);

  if (selectedPostId) {
    return <PostDetails postId={selectedPostId} onBack={() => setSelectedPostId(null)} />;
  }

  return (
    <div>
      <h1>Blog Posts</h1>
      <ul>
        {posts.map((p) => (
          <li key={p.id} onClick={() => setSelectedPostId(p.id)}>
```

```
            {p.title}
          </li>
        ))}
      </ul>
    </div>
  );
};

export default Posts;
```

Casestudy5:Todo Tracker:

```
import React, { useState, useEffect } from "react";

const TodoTracker = () => {
  const [todos, setTodos] = useState([]);
  const [filteredTodos, setFilteredTodos] = useState([]);
  const [filter, setFilter] = useState("all"); // all | completed | pending
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState(null);

  useEffect(() => {
    const fetchTodos = async () => {
      setLoading(true);
      setError(null);

      try {
        const res = await fetch("https://jsonplaceholder.typicode.com/todos?_limit=20");
        if (!res.ok) throw new Error("Failed to fetch todos");
        const data = await res.json();
        setTodos(data);
        setFilteredTodos(data); // default view = all
      } catch (err) {
        setError(err.message);
      } finally {
        setLoading(false);
      }
    };

    fetchTodos();
  }, []);

  useEffect(() => {
    if (filter === "completed") {
      setFilteredTodos(todos.filter((t) => t.completed));
    } else if (filter === "pending") {
      setFilteredTodos(todos.filter((t) => !t.completed));
    } else {
      setFilteredTodos(todos);
    }
  }, [filter, todos]);

  const completedCount = todos.filter((t) => t.completed).length;
  const pendingCount = todos.length - completedCount;

  return (
    <div className="p-6">
```

```
<h1 className="text-2xl font-bold mb-4"> Todo Tracker</h1>

{loading && <p className="text-blue-500">Loading todos...</p>}
{error && <p className="text-red-500"> {error}</p>}

{/* Filter Buttons */}
<div className="flex gap-3 mb-4">
  <button
    className={`px-4 py-2 rounded ${filter === "all" ? "bg-blue-600 text-white" : "bg-gray-200"}`}
    onClick={() => setFilter("all")}
  >
    All
  </button>
  <button
    className={`px-4 py-2 rounded ${filter === "completed" ? "bg-green-600 text-white" : "bg-gray-200"}`}
    onClick={() => setFilter("completed")}
  >
    Completed ({completedCount})
  </button>
  <button
    className={`px-4 py-2 rounded ${filter === "pending" ? "bg-yellow-600 text-white" : "bg-gray-200"}`}
    onClick={() => setFilter("pending")}
  >
    Pending ({pendingCount})
  </button>
</div>
```

```
      {/* Todos List */}
      <ul className="space-y-2">
        {filteredTodos.map((todo) => (
          <li
            key={todo.id}
            className={`p-2 border rounded ${todo.completed ? "bg-green-100" : "bg-yellow-100"}`}
          >
            <span className="font-medium">{todo.title}</span>
            <span className="ml-2 text-sm text-gray-500">
              [{todo.completed ? " Done" : “ Pending"}]
            </span>
          </li>
        ))}
      </ul>
    </div>
  );
};

export default TodoTracker;
```