

Web Programming(CS4032)**Sessional-2 Exam**

Date: April 9, 2025

Total Time: 1 hour

Course Instructor(s)

Total Marks: 25

Mahhek Tahir, Saad Farooq

Total Questions: 3

- Attempt all questions on the answer sheet
- ONE A4 Size handwritten cheat sheet with Roll# written on top of the sheet is allowed.

CLO 3: Analyze user requirements and web development problems to design solutions**Question 1****[3 marks]**

Identify the correct React Hook. Provide a small code snippet demonstrating its usage

1. You want to fetch data from an API when the component mounts.
2. You need to store a reference to a DOM element so you can focus an input field when the page loads.
3. You need to store a user's input in a form field and update the UI as they type.

CLO 3: Analyze user requirements and web development problems to design solutions**Question 2****[2 marks]**

Review the following JSX code to identify the error.

```
const Example = () => {  
  return (  
    <h1>Hello</h1>  
    <p>Welcome to React!</p>  
  );  
};  
export default Example;
```

CLO 4: Develop dynamic, server-side or client-side applications using modern web frameworks and databases**Question 3****[20 marks]**You need to build a Product Cart System using *React*, *Redux Toolkit*, and *React Router*.

Given Data (product.js)

You are provided with a **product.js** file that contains the following product data:

```
export const products = [  
  { id: 1, name: "Laptop", price: 1200 },  
  { id: 2, name: "Phone", price: 800 },  
  { id: 3, name: "Headphone", price: 200 },];
```

Tasks:

1. **Product Listing Page (Home Page) (5 marks)**
 - Display a list of products (name, price) from `product.js`. (Remember to import the file)
 - Each product should have an "Add to Cart" button.
 - Do not worry about multiple clicks to the same product.
2. **Cart Page (5 marks)**
 - Display a list of products that were added to the cart.
 - Each cart item should have a "Remove from Cart" button to remove it from the cart.
3. **Redux Setup (5 marks)**
 - Implement `cartSlice.js` using **Redux Toolkit** with the following actions:
 - `addToCart(product)`: Adds the selected product to the cart.
 - `removeFromCart(id)`: Removes the product from the cart.
4. **Routing (App.js) (5 marks)**
 - Use **React Router** to create routes for:
 - Home Page ("/") → Displays the product listing.
 - Cart Page ("/cart") → Displays the cart items.
 -
 - Configure the **Redux store** inside `App.js` and wrap the app with `<Provider>`.

```
import { BrowserRouter as Router, Routes, Route, Link } from
"react-router-dom";
import { Provider } from "react-redux";
import { configureStore } from "@reduxjs/toolkit";
import cartReducer from "./cartSlice";
import ProductList from "./ProductList";
import CartPage from "./CartPage";

/* Configure Redux store here */
const App = () => {
  return (
    <>
      { /* Wrap with Provider and Redux store */ }
      { /* Setup Router */ }
      { /* Navigation Links */ }
      { /* Define Routes */ }
    </>
  );
};
```

Requirements:

- Use **React Hooks** (e.g., `useState`) where applicable.
- Use **Redux Toolkit** for state management (`cartSlice.js`).
- Do not create a separate `store.js` – configure the store inside `App.js`.
- Use **React Router** for navigation.
- No **CSS** or **SCSS** files (Styling is not required).