# Topic : useMemo Hook..

## **Assignment: Product Price Calculator**

## Scenario:

You are tasked with building a Product Price Calculator where users can add products to their shopping cart and see the total price.

The application will also show the most expensive product in the cart. To optimize performance, you will use the useMemo hook to ensure that the total price and the most expensive product are only recalculated when necessary

## Requirements:

- ➢ Project Setup:
- - Create a new React project with a TypeScript template and name it ProductCalculator.
- ➢ Component Structure:

Create a ProductList Component:

- - This component should render a list of products available for purchase.
- - Each product should have a "Name" and "Price" displayed.
- - Each product should have an "Add to Cart" button, which, when clicked, adds the product to the cart.

Create a Cart Component:

- - This component should display the products currently in the cart.
- - It should also display the total price of the items in the cart.
- - It should show the name and price of the most expensive product in the cart.

Create an App Component:

- This component should manage the state for the products available for purchase and the products added to the cart.
- It should include the ProductList and Cart components.
- The total price and most expensive product should be calculated using the useMemo hook.

## Tasks:

1. useMemo Hook:
- Use the useMemo hook in the App component to calculate the total price of the items in the cart. This calculation should only run when the cart's contents change.
- Use another useMemo hook to find the most expensive product in the cart, recalculating only when the cart's contents change.
2. State Management:
- Manage the state of the available products and the cart in the App component.
- Pass the necessary state and functions as props to the ProductList and Cart components.
3. User Interaction:
- Users should be able to add products to the cart by clicking the "Add to Cart" button.
- The cart should automatically update the total price and display the most expensive product when items are added.
4. Expected Outcome:
- A Product Price Calculator where users can:
  o View a list of products.
  o Add products to the cart.
  o View the total price of the products in the cart.
  o See the most expensive product in the cart.