

```
export default function cartReducer(state, action) {
  switch (action.type) {
    case 'ADD_TO_CART':
      let item = { ...action.payload };
      item.qty = 1;
      item.amount = item.productPrice;

      let items = [...state.cartItems, item];
      return {
        cartItems: items,
        total: items.map(item => item.amount).reduce((v1, v2) => v1 + v2, 0)
      };
    case 'INCREMENT': return state;
    case 'DECREMENT': return state;
    case 'CLEAR_CART': return {
      cartItems: [],
      total: 0
    };
    default: return state;
  }
}
```

```
const initialState = {
  cartItems: [],
  total: 0
}
```

```
let [state, dispatch] = useReducer(cartReducer, initialState);
```

```
cartItems: [],
total: 0
```

```
return <CartContext.Provider value={{
  products: state.cartItems,
  total: state.total,
  addToCart,
  increment,
  checkout}}>
  {props.children}
</CartContext.Provider>
```

```
const CartContext = createContext();
```

products  
total

```
export default function cartReducer(state, action) {
  switch (action.type) {
    case 'ADD_TO_CART':
      let item = { ...action.payload };
      item.qty = 1;
      item.amount = item.productPrice;

      let items = [...state.cartItems, item];
      return {
        cartItems: items,
        total: items.map(item => item.amount).reduce((v1, v2) => v1 + v2, 0)
      };
    case 'INCREMENT': return state;
    case 'DECREMENT': return state;
    case 'CLEAR_CART': return {
      cartItems: [],
      total: 0
    };
    default: return state;
  }
}
```

```
const initialState = {
  cartItems: [],
  total: 0
}
```

```
let [state, dispatch] = useReducer(cartReducer, initialState);
```

```
cartItems: [],
total: 0
```

```
const CartContext = createContext();
```

products  
total

```
<FontAwesomeIcon
  onClick={() => addToCart(product)}
  color="blue"
  icon={faShoppingCart} />
```

```
function addToCart(product) {
  dispatch({type: 'ADD_TO_CART', payload: product});
}
```