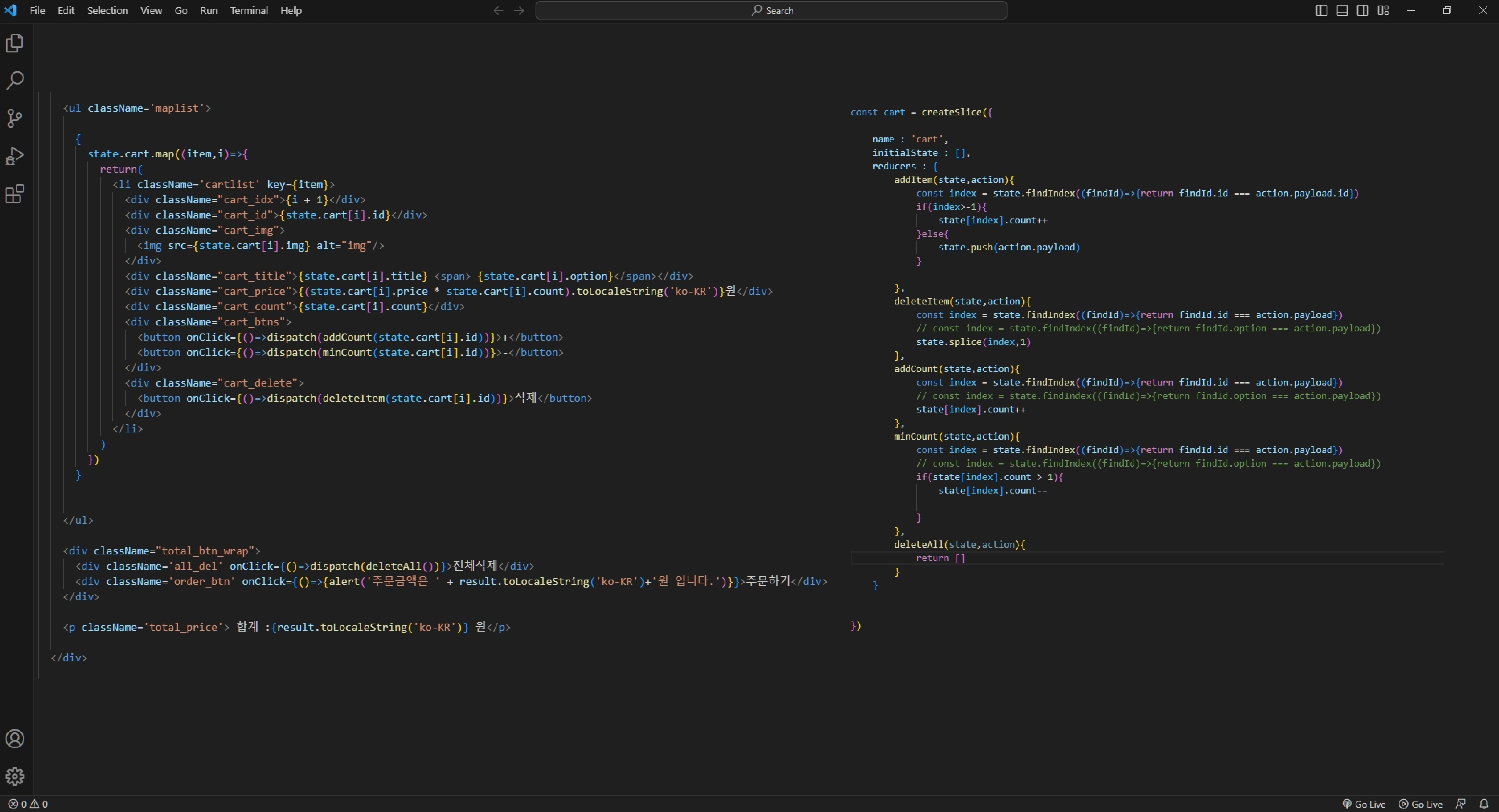

```
  <ul className='maplist'>

    {
      state.cart.map((item,i)=>{
        return(
          <li className='cartlist' key={item}>
            <div className="cart_idx">{i + 1}</div>
            <div className="cart_id">{state.cart[i].id}</div>
            <div className="cart_img">
              <img src={state.cart[i].img} alt="img"/>
            </div>
            <div className="cart_title">{state.cart[i].title} <span> {state.cart[i].option}</span></div>
            <div className="cart_price">{(state.cart[i].price * state.cart[i].count).toLocaleString('ko-KR')}원</div>
            <div className="cart_count">{state.cart[i].count}</div>
            <div className="cart_btns">
              <button onClick={()=>dispatch(addCount(state.cart[i].id))}>+</button>
              <button onClick={()=>dispatch(minCount(state.cart[i].id))}>-</button>
            </div>
            <div className="cart_delete">
              <button onClick={()=>dispatch(deleteItem(state.cart[i].id))}>삭제</button>
            </div>
          </li>
        )
      })
    }


  </ul>

  <div className="total_btn_wrap">
    <div className='all_del' onClick={()=>dispatch(deleteAll())}>전체삭제</div>
    <div className='order_btn' onClick={()=>{alert('주문금액은 ' + result.toLocaleString('ko-KR')+'원 입니다.')}}>주문하기</div>
  </div>

  <p className='total_price'> 합계 :{result.toLocaleString('ko-KR')} 원</p>

</div>
```

```
const cart = createSlice({

    name : 'cart',
    initialState : [],
    reducers : {
        addItem(state,action){
            const index = state.findIndex((findId)=>{return findId.id === action.payload.id})
            if(index>-1){
                state[index].count++
            }else{
                state.push(action.payload)
            }

        },
        deleteItem(state,action){
            const index = state.findIndex((findId)=>{return findId.id === action.payload})
            // const index = state.findIndex((findId)=>{return findId.option === action.payload})
            state.splice(index,1)
        },
        addCount(state,action){
            const index = state.findIndex((findId)=>{return findId.id === action.payload})
            // const index = state.findIndex((findId)=>{return findId.option === action.payload})
            state[index].count++
        },
        minCount(state,action){
            const index = state.findIndex((findId)=>{return findId.id === action.payload})
            // const index = state.findIndex((findId)=>{return findId.option === action.payload})
            if(state[index].count > 1){
                state[index].count--

            }
        },
        deleteAll(state,action){
            return []
        }
    }


})
```