**문제 보기**

**문제지**

당신은 당신의 FE 기술을 이용해 새로운 쇼핑몰 사이트의 프로토타입을 만드는 업무를 받게 되었습니다. 그런데 아무런 외부 라이브러리를 사용하지 말고 vanilla js를 이용한 SPA로 만들라는 요구사항이 떨어졌습니다.

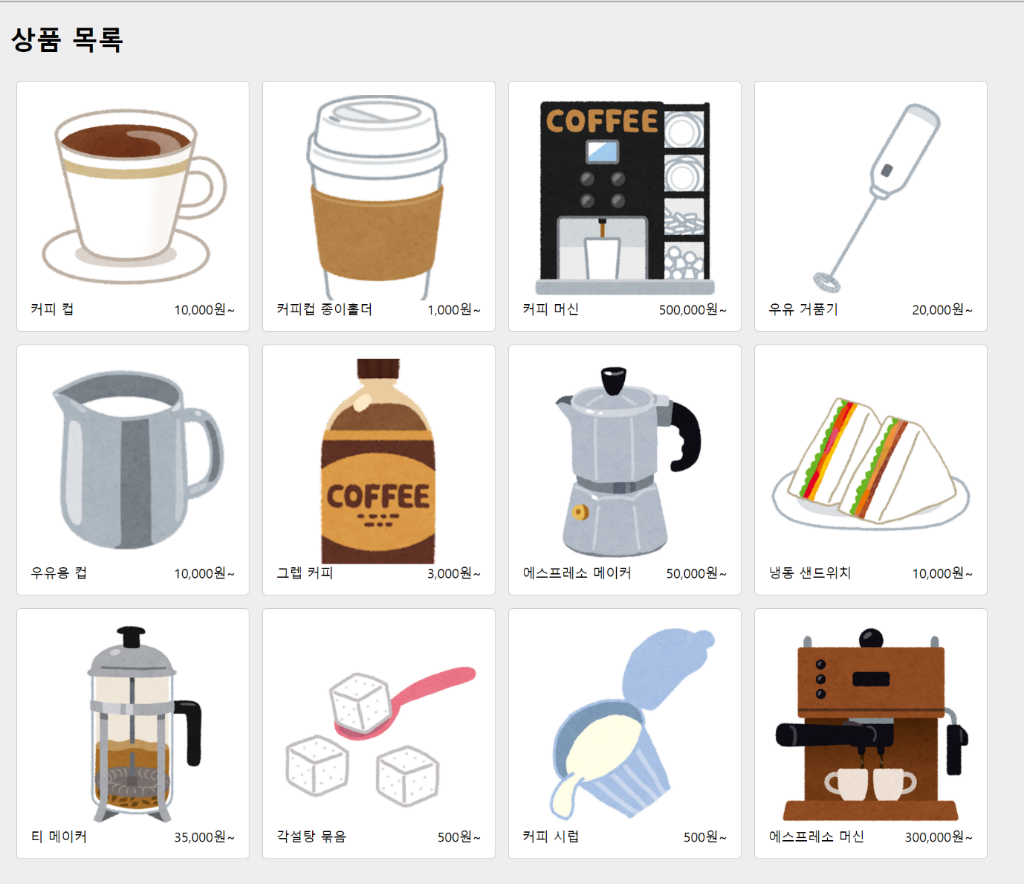
아래 요구사항을 잘 보고, 업무를 완료할 수 있도록 요구사항을 잘 만족하는 SPA 쇼핑몰을 만들어보도록 합시다.

**요구사항**

**공통 요구사항**

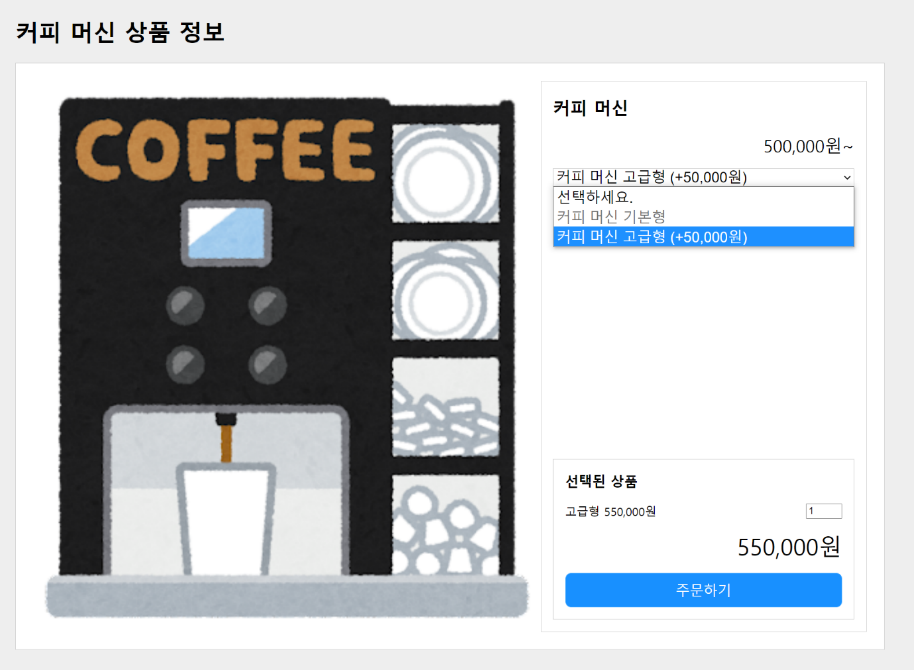
* **(중요) index.html에 들어있는 마크업 가이드를 참고하여, 각 마크업의 구조 및 class 이름 등을 반드시 유지해주세요. 만약 마크업이 가이드와 맞지 않거나 class 이름 등이 누락될 경우 0점 처리될 수 있습니다.**
* **(중요) 아래 요구사항에 적혀진 페이지별 url, localStorage key 이름 등은 반드시 지켜주세요.**
* **(필수) 총 3개의 페이지를 만들어야 합니다.**
* **(필수) 기본으로 제공되는 index.html의 마크업대로 각 컴포넌트의 UI를 작성하도록 합니다.**
* 상품의 가격을 표시할 때에는 3자리마다 콤마(,)를 찍도록 합니다.

**상품 목록 페이지**



* url: /web/
* 위의 디자인에 따라 상품 목록을 렌더링 합니다.
* 목록에서 상품 카드를 클릭하면 상품 상세 페이지로 이동합니다.

**상품 상세 페이지**



* url: /web/products/:productId
* **(필수) productId에 해당하는 상품을 불러오고, 상품 정보를 렌더링 합니다.**
  + 반드시 아래 형식에 맞게 옵션 텍스트를 렌더링 해주세요.
    - **(필수) 상품 옵션가가 0인 경우, 아래의 형식으로 전체 옵션이름을 렌더링 합니다.**
      * ${상품이름} ${옵션이름}
    - **(필수) 상품 옵션가가 0보다 큰 경우, 아래의 형식으로 전체 옵션이름을 렌더링 합니다.**
      * ${상품이름} ${옵션이름} (+${옵션가격}원)
        + 옵션 가격을 렌더링하는 경우 가격 부분에 3자리마다 콤마(,)를 찍도록 합니다.
    - **(필수) 재고가 0인 상품의 경우, 아래의 형식으로 전체 옵션이름을 렌더링 합니다.**
      * (품절) ${상품이름} ${옵션이름}
        + option에 disabled attribute를 지정하여 선택되지 않게 합니다.
* **(필수) 상품의 옵션을 선택하면, 선택된 상품을 보여주는 영역에 추가합니다.**
  + **(필수) 이미 선택된 상품은 다시 선택해도 선택된 상품에 추가되지 않아야 합니다.**
  + **(필수) 선택된 옵션들의 총 가격을 계산해서 위의 이미지처럼 렌더링합니다.**
    - 옵션별 가격은 상품 가격 + 옵션 가격을 합친 값입니다.
    - 각 옵션별 가격은 수량만큼 곱해야 합니다.
* 선택된 상품의 input에서 수량을 변경하면 수량이 변경되어야 합니다.
  + 선택한 옵션의 개수를 변경 시, option의 stock을 넘을 수 없게 해야합니다.
  + 해당 input에 숫자가 아닌 값을 넣은 경우 무시하도록 합니다.
* **(필수) 주문하기 버튼을 누르면 localStorage에 아래 형태의 데이터로 상품들을 추가하고, /cart 페이지로 이동합니다. 이때 localStorage에 담는 키 이름은 반드시 products\_cart 라는 이름으로 하도록 합니다.**

[

{

productId: '상품 id',

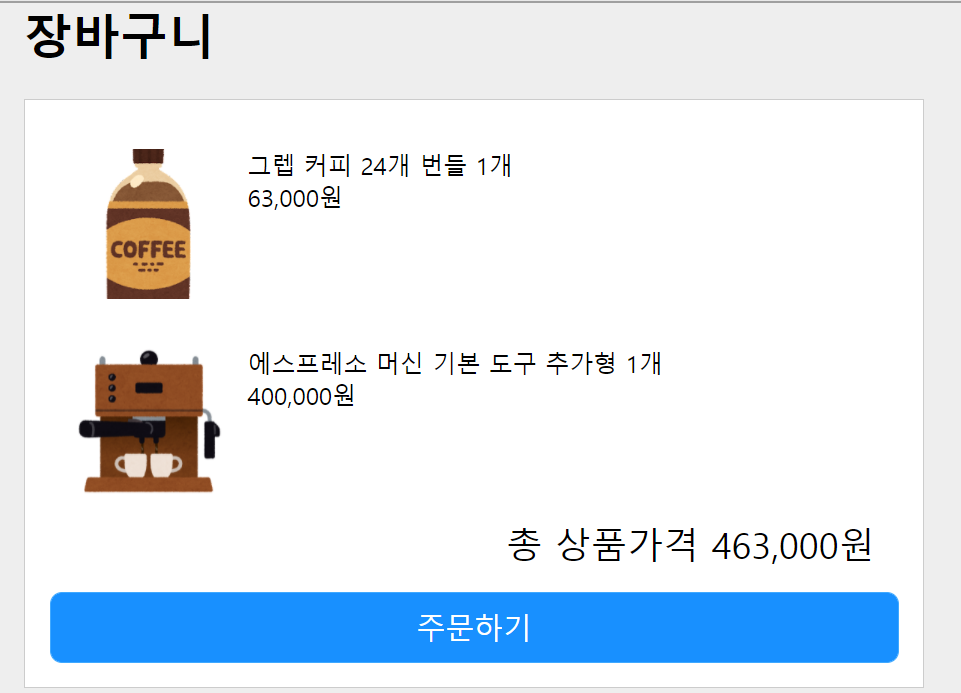
optionId: '선택한 옵션의 id',

quantity: '선택한 수량'

}

]

**장바구니**



* url: /web/cart
* **(필수) local storage에 담아둔 장바구니 관련 데이터를 이용해 상품 및 상품 옵션 데이터를 불러오고, 화면 가이드에 맞게 장바구니에 담긴 상품들을 렌더링 합니다.**
  + 출력 텍스트 형식은 마크업, 그리고 예시 화면과 같게 해주세요.
  + ${상품이름} ${옵션이름} ${수량}개의 형식입니다.
  + 상품은 여러 종류를 담을 수 있어야 합니다.
    - 1번 상품 상세 페이지에서 상품을 고른 뒤 주문하기를 누르고, 다시 상품 목록 페이지로 돌아가 2번 상품 상세 페이지에서 상품을 고른 뒤 주문하기를 누르면 장바구니 페이지에서 선택된 1번 상품과 2번 상품 모두 렌더링 되어야 합니다.
    - 위의 경우, 다시 상품 목록 페이지로 돌아가는 것은 뒤로 가기를 통해 이동하면 되며, 별도의 기능 구현은 필요하지 않습니다.
* **(필수) 장바구니에 담긴 상품은 예시처럼 상품의 기본가 + 옵션가와 각 옵션의 수량을 더한 총액을 맨 아래 총 가격 부분에 출력해야 합니다.**
  + 출력 텍스트 형식은 마크업, 그리고 예시 화면과 같게 해주세요.
  + 총 상품가격 ${계산된 총 상품 가격}원의 형식입니다.
* **(필수) 주문하기 클릭 시, alert을 이용해 주문되었습니다 메시지를 띄우고 상품 목록 페이지로 이동시킵니다. 이때 local storage에 담아둔 장바구니 값들은 모두 비워야합니다.**
* 만약 장바구니에 담은 상품이 없는 채로 장바구니 페이지에 들어오면 alert을 이용해 장바구니가 비어 있습니다 메세지를 띄운 후 상품 목록 페이지로 이동시킵니다.

**API 명세**

**상품 목록 조회**

* GET - https://h6uc5l8b1g.execute-api.ap-northeast-2.amazonaws.com/dev/products
  + 상품 목록을 조회합니다.
  + 응답은 아래와 같습니다.

[

{

"id": 1,

"name": "커피 컵",

"imageUrl": "https://grepp-cloudfront.s3.ap-northeast-2.amazonaws.com/programmers\_imgs/assignment\_image/cafe\_coffee\_cup.png",

"price": 10000

},

{

"id": 2,

"name": "커피컵 종이홀더",

"imageUrl": "https://grepp-cloudfront.s3.ap-northeast-2.amazonaws.com/programmers\_imgs/assignment\_image/coffee\_cup\_paper\_sleeve.png",

"price": 1000

},

...

]

* 각 항목이 나타내는 값은 아래와 같습니다.
  + id: 상품 id
  + name: 상품 이름
  + imageUrl: 상품 이미지 주소
  + price: 상품 가격

**상품 옵션 및 수량 조회**

* GET - https://h6uc5l8b1g.execute-api.ap-northeast-2.amazonaws.com/dev/products/{productId}
  + productId에 해당하는 상품을 조회합니다.
  + 응답은 아래와 같습니다.

{

"id": 1,

"name": "커피 컵",

"price": 10000,

"imageUrl": "https://grepp-cloudfront.s3.ap-northeast-2.amazonaws.com/programmers\_imgs/assignment\_image/cafe\_coffee\_cup.png",

"productOptions": [

{

"id": 1,

"name": "100개 묶음",

"price": 0,

"stock": 5,

"created\_at": "2021-08-23T22:52:17.634Z",

"updated\_at": "2021-08-23T22:52:17.638Z"

},

{

"id": 2,

"name": "200개 묶음",

"price": 8000,

"stock": 5,

"created\_at": "2021-08-23T22:52:34.248Z",

"updated\_at": "2021-08-23T22:52:34.252Z"

},

{

"id": 24,

"name": "10개 묶음",

"price": 0,

"stock": 555,

"created\_at": "2021-08-23T23:03:04.873Z",

"updated\_at": "2021-08-23T23:03:04.879Z"

}

]

}

* 각 항목이 나타내는 값은 아래와 같습니다.
  + id: 상품 id
  + name: 상품 이름
  + price: 상품 가격
  + imageUrl: 상품 이미지 주소
  + productsOptions
    - id: 옵션 id
    - name: 옵션 이름
    - price: 옵션 가격
    - stock: 재고