[한별이에게]

1. 코드 분석

1) add.js (관리자 페이지)

- const table : table, thead, tbody 각각 요소 만들어서 선언 -> const tr1 : tr 만듬 (createElement)
- ["사진", "카테고리" ~].forEach : th명들을 배열로 만들어서 forEach문으로 th만듬 -> appendChild로 테이블과 HTML에 넣어줌
- window.onload : 새로고침 시 데이터 유지
- addEventListener("input") : 각 인풋창 실시간 바뀌는 내용 검사
  -> 나는 oninput을 사용했는데 다른 방법이라 새로움
- function checkedFaild(input) : 인풋값 가져와서 const에 넣어줌
- if 문으로 각 input 내용 검사 > isVaild를 true로 선언하여두고 조건이 다 맞으면 false로 바꾸어 버튼 활성화
- const categoryList : 두 가지 카테고리 담긴 div 박스 선언
- categoryList.addEventListener("click", function(event) : event.target을 사용하여 기본 값은 안경으로 해두고, 클릭 시에 선택 된 카테고리로 변경
  -> radio 버튼을 사용하지 않고 JS로 구현
- function saveData() : 저장버튼을 누르면 input값을 가져와 선언 후 로컬스토리지에 등록
- function addRowToTable() : 로컬스토리지 값으로 테이블 생성, innerHTML이 아닌 tr, td, img, button을 각각 따로 생성하여 appendChild로 추가
- function editRow(cartInfo, row) : 테이블 수정 함수. row.cells[i]로 각 행의 값 가져와서 수정
  -> 카테고리는 셀렉트로 바뀌어서 수정 가능
  -> input 각각 만들어서 값, 타입 따로 지정 (나는 innerHTML로 한번에 만듬)
- editBtn.onclick() : editRow 함수 안에서 수정 버튼 클릭 시 값 검사 및 저장.
- remove_tr(This) : 행 삭제 함수. This.closest("tr") 로 가까운 tr 찾아서 row.remove() 실행

2) cart.js (장바구니 페이지)

- document.addEventListener("DOMContentLoaded", function () : onload와 같은 기능, 먼저 함수들 실행
- function addClearCartButton() : 조건문으로 버튼 없으면 장바구니 버튼 생성
- function addCheckoutButton() : 결제하기 버튼 js로 생성
- function displayCartItems() : 장바구니 항목 표시 함수, if (cartItems.length === 0) 조건으로 장바구니 비어있는 상태 표시
- function addEventListeners() : 장바구니 수량 변경 → change로 수량 변경 되면 data-index 가져와서 수량 변경 후 로컬스토리지에 저장

```
document.querySelectorAll(".quantity-input").forEach((input) => {
  input.addEventListener("change", function ()
```

- cartItems.splice(index, 1); : 장바구니에서 하나만 삭제 시에 this.dataset으로 index 가져온 후 splice로 삭제, 로컬스토리지에 저장

```
const index = this.dataset.index;
cartItems.splice(index, 1);
```

3) detail.js

- const isWished = wishlist.includes(productId); : boolean 값 if문에 사용
- function addShoppingCart(product) : 장바구니 담기 함수. cartItems 로컬스토리지에서 불러와서 find로 중복 여부 확인 후 if 문으로 개수 추가하거나 1로 넣어줌

4) footer.js , header.js

- → js로 구현

5) main.js

- 상품 클릭 및 좋아요 버튼은 event.target으로 값 찾음

```
addEventListener("click", function (event) {
  const clickedElement = event.target;
```

2. 개선점

- 헤더, 푸터 > JS로 만듬

→ 코드가 복잡하고 유지보수나 CSS 입히는 것이 조금 어려워 보인다.

→ JS로만 대부분의 파일을 만들어서 코드가 복잡함 > 오히려 일부분은 HTML로 만드는 것이 좋아 보인다.
→ 헤더와 푸터가 여러 번 중복되는 페이지의 경우 fetch() 라는 것을 사용하여 html과 js를 같이 쓰는 방법이 좋음.

```
document.addEventListener("DOMContentLoaded", function () {
fetch("/header.html")
.then(response ⇒ response.text())
.then(data ⇒ document.querySelector("header").innerHTML = data);

fetch("/footer.html")
.then(response ⇒ response.text())
.then(data ⇒ document.querySelector("footer").innerHTML = data);
});
```

→ createElement , class명, appendchild 가 반복적으로 많아서 더 많은 내용을 만들려면 힘들 것 같음.

- 오류

→ [관리자 페이지] 수정버튼 눌렀을 때 가격 값이 안 끌어와짐

→ [헤더 하트 및 메인 페이지 하트] 좋아요 페이지 눌렀다가 메인으로 돌아오면 하트 사라짐 오류

- CSS

→ [장바구니 페이지] 장바구니에 안 담겨있을 때 장바구니가 비어있습니다. 글씨가 잘려있음.

→ alert 창이 div여서 sweetalert로 변경하면 좋을 것 같다.

→ 하단 푸터의 메뉴들에 마우스 커서 올렸을 때 마우스 모양 바뀌도록 하는 것이 좋을 것 같음 (css : cursor) + alert

3. 배운점

1. 대부분의 코드를 HTML이나 CSS가 아닌 JS로 만들어서 JS로 이런 부분까지 가능하구나를 많이 느낌

→ 특히 헤더와 푸터 : createElement 생성 > class이름 생성 > css도 js로 .style로 줌

2. 장바구니 수량 변경 버튼 구현

→ 수량 변경 어려워서 중복 값은 저장하지 않는 것으로 구현했었는데 이 코드처럼 다시 짜보고 싶다.

3. closest 가까운 클래스 찾아주는 코드

→ 나는 다 선언해놨는데 편리해보여서 사용하고 싶음.

4. 함수 선언을 잘해놔서 코드가 깔끔하고 보기가 편하다. -> 여러번 사용하는 것은 함수에 미리 선언해두고 사용하기.

5. 헤더, 푸터 : 스크립트 파일을 같은 html 파일에서 선언하면 선언된 함수 사용 가능한 점
   → 대신 헤더에 선언, html 안에서도 함수를 만든 파일이 위에 있어야 함.

6. 장바구니 비어있지 않으면 버튼 숨기기 코드

```
clearCartButton.style.display = localStorage.getItem("cartItems")
    ? "inline-block"
    : "none";
```

→ 삼항연산자로 코드가 깔끔하고 간단해서 다른 프로젝트에서 사용해보고 싶은 코드

→ 나는 if 문이어서 코드가 너무 길고 복잡하다.

7. data-* 속성 : data-category, data-set 등등 + const category = event.target.getAttribute("data-category");

→ class 대신에 data-* 속성 를 사용해서 map의 인덱스나 id 값을 설정해주어서 값을 가져오기에 용이함