CLASE 1 - Cod a pegar para tener commits de parte de todos:

1. Mauro : crear archive base - Index.html y Por fuera del index.html Creo la CARPETA SRC, luego pego este código :

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>ProyectoLabAppWeb

</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css"

rel="stylesheet"

integrity="sha384-sRIl4kxILFvY47J16cr9ZwB07vP4J8+LH7qKQnuqkuIAvNWLzeN8tE5YBujZqJLB"

crossorigin="anonymous">

</head>

<body>

<!-- Navbar -->

<nav class="navbar navbar-expand-lg bg-body-tertiary">

<div class="container-fluid">

<a class="navbar-brand" href="#">Navbar</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse"

data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="#">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" role="button"

data-bs-toggle="dropdown" aria-expanded="false">

Dropdown

</a>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

<li class="nav-item">

<a class="nav-link disabled" aria-disabled="true">Disabled</a>

</li>

</ul>

<form class="d-flex" role="search">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search"/>

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<!-- Main -->

<main>

<div class="row row-cols-1 row-cols-md-2 g-4" id="product-list">

<!-- generado desde JS -->

</div>

</main>

<!-- Footer -->

<footer class="mt-4 text-center">

<p class="text-muted">&copy; 2025 Mi Tienda Online</p>

</footer>

<h1 class="text-center mt-4">Hello, world!</h1>

<!-- Scripts -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.bundle.min.js"

integrity="sha384-FKyoEForCGlyvwx9Hj09JcYn3nv7wiPVlz7YYwJrWVcXK/BmnVDxM+D2scQbITxI"

crossorigin="anonymous"></script>

<script type="module" src="./src/index.js"></script>

</body>

</html>

1. Vivi- crear archivo Api.js

DENTRO de carpeta SRC Y PEGAR este cod:

export function getProducts() {

return fetch('https://fakestoreapi.com/products')

.then(response => response.json()) // acá obtenés el JSON

.then(data => {

console.log(data); // si querés ver qué llega

return data; // devolvés los productos al .then() de index.js

});

}

Luego , creamos TAMBIÉN dentro de la carpeta SRC, el archivo Index.js y pegamos este código:

import { getProducts } from "./api.js";

import { RenderCards } from "./components/cards.js";

RenderCards();

1. Por último, para hacer commit todos, DARI crea la carpeta “Components” dentro de SRC

Dentro de Components, se crea el archivo “Cards.js” y se pega el siguiente codigo:

import { getProducts } from "../api.js"

export function RenderCards(){

let productList = document.querySelector("#product-list");

console.log(productList);

getProducts().then((products) => {

let tempalte = "";

products.forEach((p) => {

tempalte += `

<div class="col">

<div class="card">

<img src="${p.image}" class="card-img-top" alt="${p.title}">

<div class="card-body">

<h5 class="card-title">${p.title}</h5>

<p class="card-text">${p.description}</p>

</div>

</div>

</div>

`;

});

productList.innerHTML = tempalte;

});

}

Clase 2:

1. Mauro:

Index. Html ( lo actualizo)

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>ProyectoLabAppWeb

</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css"

rel="stylesheet"

integrity="sha384-sRIl4kxILFvY47J16cr9ZwB07vP4J8+LH7qKQnuqkuIAvNWLzeN8tE5YBujZqJLB"

crossorigin="anonymous">

</head>

<body>

<!-- Navbar -->

<nav class="navbar navbar-expand-lg bg-body-tertiary">

<div class="container-fluid">

<a class="navbar-brand" href="#">Navbar</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse"

data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="#">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" role="button"

data-bs-toggle="dropdown" aria-expanded="false">

Dropdown

</a>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="#">Action</a></li>

<li><a class="dropdown-item" href="#">Another action</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Something else here</a></li>

</ul>

</li>

<li class="nav-item">

<a class="nav-link disabled" aria-disabled="true">Disabled</a>

</li>

</ul>

<form class="d-flex" role="search">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search"/>

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<!-- Main -->

<main class="container">

<input type="text" id="search" name= "Search", id = "", value = "search";>

<div class="row row-cols-1 row-cols-md-4 g-2 mt = -5" id="product-list">

<!-- generado desde JS -->

</div>

<div class="modal fade" id="prodModal" tabindex="-1" aria-labelledby="prodModalLabel" aria-hidden="true"></div>

</main>

<!-- Footer -->

<footer class="mt-4 text-center">

<p class="text-muted">&copy; 2025 Mi Tienda Online</p>

</footer>

<h1 class="text-center mt-4">Hello, world!</h1>

<!-- Scripts -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.bundle.min.js"

integrity="sha384-FKyoEForCGlyvwx9Hj09JcYn3nv7wiPVlz7YYwJrWVcXK/BmnVDxM+D2scQbITxI"

crossorigin="anonymous"></script>

<script type="module" src="./src/index.js"></script>

</body>

</html>

1. Vivi ACTUALIZA el Api.js con el siguiente código:

export function getProducts() {

return fetch('https://fakestoreapi.com/products')

.then(response => response.json()) // acá obtenés el JSON

.then(data => {

console.log(data); // si querés ver qué llega

return data; // devolvés los productos al .then() de index.js

});

}

Luego Vivi de nuevo actualiza el Index.js

import { RenderCards } from "./components/cards.js";

RenderCards();

1. Dario va a Acutalizar el código de Cards.js y además crear el archivo “Modal.js” con los sig. Cod:

Cards.js (Actualizado):

import { getProducts } from "../api.js";

import { Modal } from "./modal.js";

export function RenderCards() {

let productList = document.querySelector("#product-list");

console.log(productList);

getProducts().then((products) => {

let template = "";

products.forEach((p) => {

template += `

<div class="col">

<div class="card justify-content-center align-items-center" style="width: 300px;">

<img src="${p.image}" class="card-img-top" alt="${p.title}" style="height: 300px; width: 259px; object-fit: contain;">

<div class="card-body" style="width: 300px; overflow: hidden;">

<h5 class="card-title text-truncate" style="white-space: nowrap; overflow: hidden; text-overflow: ellipsis;">

${p.title}

</h5>

</div>

<div class="mb-3">

<button class="btn btn-dark" id="btn-${p.id}">Más detalles</button>

</div>

</div>

</div>

`;

});

productList.innerHTML = template;

products.forEach((p) => {

let button = document.querySelector(`#btn-${p.id}`);

if (button) {

button.addEventListener('click', () => {

// console.log(`Botón de producto ${p.id} clickeado`);

Modal(p);

});

}

});

});

}

Modal.js – recordar que al igual que Cards.js , va dentro de la carpeta Components

export function Modal(prod) {

const container = document.querySelector("#prodModal");

container.innerHTML = `

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h1 class="modal-title fs-5" id="prodModalLabel">${prod.title}Modal title</h1>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<div class "row">

<div class"col-md-6">

<img src="${prod.image}" class="img-fluid" alt="${prod.title} " style="height: 300px; width: 259px; object-fit: contain;">

</div>

<div class"col-md-6">

<p>${prod.description}</p>

</div>

<div class"col-12 d-flex justify-content-end aling-items-start">

<p class"justify-content-end aling-items-start">Price: USD${prod.price}</p>

</div>

Aquí va el contenido dinámico que pongamos en el modal...

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>

<button type="button" class="btn btn-primary">Agregar al Carrito</button>

</div>

</div>

</div>

`;

// 3. Ahora sí, instanciamos el modal sobre #prodModal (que sigue teniendo .modal.fade)

const modalElement = document.getElementById("prodModal");

const bootstrapModal = new bootstrap.Modal(modalElement);

// 4. Lo mostramos

bootstrapModal.show();

}

Hasta acá. Ya debería verse asi la web :



