Para hacer que tu página sea compatible con JSON y aprovecharlo para manejar dinámicamente los datos de los productos, debes configurar un entorno que sirva el contenido de forma más estructurada. Aquí te guiaré paso a paso:

1. Estructura del Proyecto

Asegúrate de que tu proyecto tiene la siguiente estructura:

/mi-proyecto

|-- /img

|-- /node\_modules

|-- app.js

|-- estilo.css

|-- index.html

|-- productos.json

|-- server.js

2. Archivo productos.json

Crea un archivo llamado productos.json en el mismo nivel que los demás. Este archivo contendrá los datos de los productos, como en el ejemplo:

[

{ "nombre": "1021", "tipo": "Ojos", "imagen": "img/ojos1021.jpg", "precio": 24500 },

    { "nombre": "1010", "tipo": "Polvo", "imagen": "img/polvo1010.jpg", "precio": 25000 },

    { "nombre": "1018", "tipo": "Polvo", "imagen": "img/polvo1018.jpg", "precio": 25500 },

    { "nombre": "1019", "tipo": "Polvo", "imagen": "img/polvo1019.jpg", "precio": 26000 },

    { "nombre": "1020", "tipo": "Polvo", "imagen": "img/polvo1020.jpg", "precio": 26500 }

]

3. Servidor Node.js (Archivo server.js)

Configura un servidor con Node.js para servir el archivo JSON y otros recursos estáticos.

Instalación de Express

Ejecuta este comando en la terminal para instalar Express:

npm install express

Código del Servidor

Crea un archivo server.js con el siguiente contenido:

const express = require('express');

const app = express();

const path = require('path');

// Middleware para servir archivos estáticos

app.use(express.static(path.join(\_\_dirname)));

// Ruta para enviar el archivo JSON

app.get('/api/productos', (req, res) => {

    res.sendFile(path.join(\_\_dirname, 'productos.json'));

});

// Iniciar el servidor

const PORT = 3000;

app.listen(PORT, () => {

    console.log(`Servidor corriendo en http://localhost:${PORT}`);

});

4. Frontend (Archivo app.js)

Modifica el archivo app.js para consumir los datos de productos.json a través de una solicitud fetch.

document.addEventListener("DOMContentLoaded", async () => {

const contenedorItems = document.querySelector(".contenedor-items");

try {

// Hacer la solicitud al backend

const response = await fetch('productos.json'); // IMPoRTANTE

if (!response.ok) {

throw new Error("Error al obtener los productos");

}

const productos = await response.json();

// Generar los productos en el DOM

productos.forEach(producto => {

const item = document.createElement("div");

item.classList.add("item");

item.innerHTML = `

<span class="titulo-item">${producto.nombre}</span>

<img src="${producto.imagen}" alt="${producto.nombre}" class="img-item">

<span class="modo-uso-item">${producto.tipo}</span>

<span class="precio-item">$${producto.precio.toLocaleString()}</span>

<button class="boton-item">Agregar al Carrito</button>

`;

contenedorItems.appendChild(item);

});

// Asignar eventos después de crear los productos

const botonesAgregarAlCarrito = document.getElementsByClassName('boton-item');

for (let i = 0; i < botonesAgregarAlCarrito.length; i++) {

botonesAgregarAlCarrito[i].addEventListener('click', agregarAlCarritoClicked);

}

} catch (error) {

console.error("Error al cargar los productos:", error);

}

// Asignar los eventos relacionados con el carrito

ready();

});

5. Ejecutar el Proyecto

En la terminal, inicia el servidor:

node server.js

Abre tu navegador y accede a http://localhost:3000.

6. Verificación

Tu página debería cargar los productos desde el archivo JSON y mostrarlos dinámicamente.

Si necesitas realizar cambios en los datos de los productos, solo actualiza el archivo productos.json.