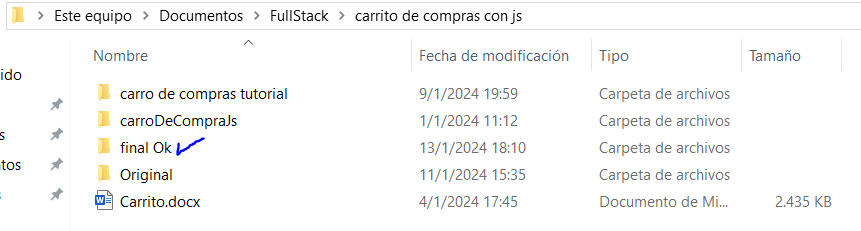
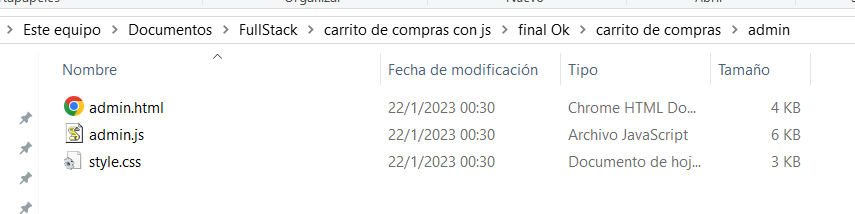
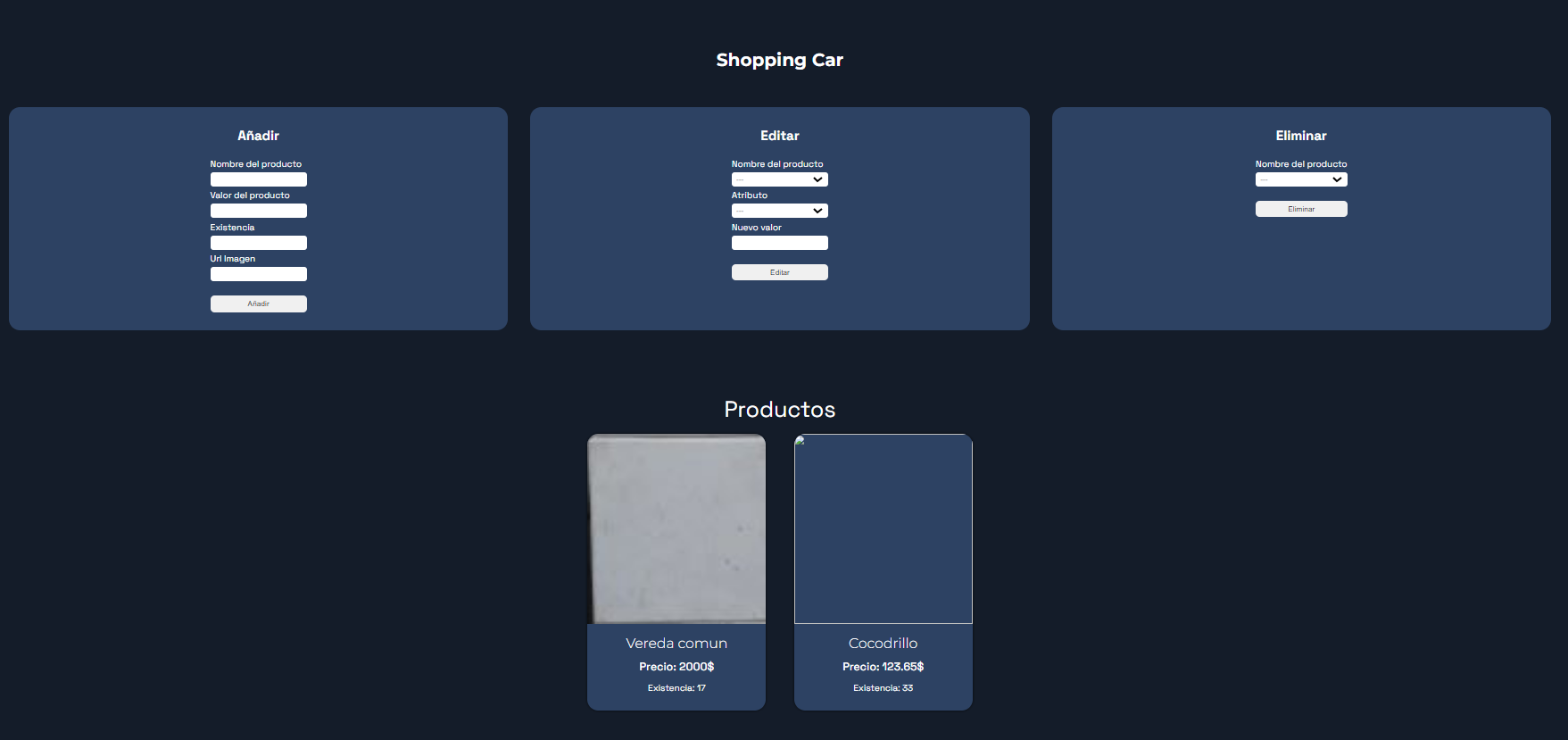
Carrito de compras shopping cart







El archivo admin.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Carrito</title>

<link rel="stylesheet" href="style.css">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Space+Grotesk:wght@400;500;600;700&display=swap"

rel="stylesheet">

<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@300;400;500;600;700&display=swap"

rel="stylesheet">

</head>

<body>

<header id="header">

<h1>Shopping Car</h1>

</header>

<main>

<div class="contenedor">

<!-- Añadir -->

<div class="añadir">

<h2>Añadir</h2>

<form>

<label>Nombre del producto</label>

<input type="text" id="productoAñadir" name="nombreDelProducto">

<label>Valor del producto</label>

<input type="number" id="valorAñadir">

<label>Existencia</label>

<input type="number" id="existenciaAñadir">

<label>Url Imagen</label>

<input type="text" id="ImagenAñadir">

<input class="button" type="button" id="botonAñadir" value="Añadir">

</form>

</div>

<!-- Editar -->

<div class="editar">

<h2>Editar</h2>

<form>

<label>Nombre del producto</label>

<select id="productoEditar">

<option value="">---</option>

</select>

<label>Atributo</label>

<select id="atributoEditar">

<option value="">---</option>

</select>

<label>Nuevo valor</label>

<input type="text" id="nuevoAtributo">

<input class="button" type="button" id="botonEditar" value="Editar">

</form>

</div>

<!-- Eliminar -->

<div class="eliminar">

<h2>Eliminar</h2>

<form>

<label>Nombre del producto</label>

<select id="productoEliminar">

<option value="">---</option>

</select>

<input class="button" type="button" id="botonEliminar" value="Eliminar">

</form>

</div>

</div>

<!-- Mostrar el mensaje -->

<div class="contenedorMensaje">

<div id="mensaje"></div>

</div>

<!-- Productos -->

<div class="contenedorProductos">

<h2>Productos</h2>

<div class="mostrarProductos" id="mostrarProductos">

</div>

</div>

</div>

</main>

<script src="admin.js"></script>

</body>

</html>

El archivo admin.js

function guardarAlmacenamientoLocal(llave, valor\_a\_guardar) {

localStorage.setItem(llave, JSON.stringify(valor\_a\_guardar))

}

function obtenerAlmacenamientoLocal(llave) {

const datos = JSON.parse(localStorage.getItem(llave))

return datos

}

let productos = obtenerAlmacenamientoLocal('productos') || [];

let mensaje = document.getElementById('mensaje')

//Añadir un producto

const añadirProducto = document.getElementById('productoAñadir')

const añadirValor = document.getElementById('valorAñadir')

const añadirExistencia = document.getElementById('existenciaAñadir')

const añadirImagen = document.getElementById('ImagenAñadir')

document.getElementById("botonAñadir").addEventListener("click", function (event) {

event.preventDefault()

let productoAñadir = añadirProducto.value

let valorAñadir = añadirValor.value

let existenciaAñadir = añadirExistencia.value

let imagenAñadir = añadirImagen.value

let van = true

if (productoAñadir == '' || valorAñadir == '' || existenciaAñadir == '' || imagenAñadir == '') {

mensaje.classList.add('llenarCampos')

setTimeout(() => { mensaje.classList.remove('llenarCampos') }, 2500)

van = false

}

else {

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

if (productos[i].nombre == productoAñadir) {

mensaje.classList.add('repetidoError')

setTimeout(() => { mensaje.classList.remove('repetidoError') }, 2500)

van = false

}

}

}

if (van == true) {

productos.push({

nombre: productoAñadir,

valor: valorAñadir,

existencia: existenciaAñadir,

urlImagen: imagenAñadir

})

mensaje.classList.add('realizado')

setTimeout(() => {

mensaje.classList.remove('repetidoError')

window.location.reload()

}, 1500)

}

guardarAlmacenamientoLocal('productos', productos);

})

// Editar

const productoEd = document.getElementById('productoEditar')

const atributoEd = document.getElementById('atributoEditar')

const nuevoAtributoEd = document.getElementById('nuevoAtributo')

document.getElementById("botonEditar").addEventListener("click", function (event) {

event.preventDefault()

let productoEditar = productoEd.value

let atributoEditar = atributoEd.value

let nuevoAtributo = nuevoAtributoEd.value

let van = false

if (productoEditar == '' || atributoEditar == '' || nuevoAtributo == '') {

mensaje.classList.add('llenarCampos')

setTimeout(() => { mensaje.classList.remove('llenarCampos') }, 2500)

}

else {

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

if (productos[i].nombre == productoEditar) {

productos[i][atributoEditar] = nuevoAtributo

van = true

}

}

if (van == true) {

mensaje.classList.add('realizado')

setTimeout(() => {

mensaje.classList.remove('realizado')

window.location.reload()

}, 1500);

}

else {

mensaje.classList.add('noExisteError')

setTimeout(() => { mensaje.classList.remove('noExsiteError') }, 2500);

}

guardarAlmacenamientoLocal('productos', productos);

}

})

// Eliminar

const productoE = document.getElementById('productoEliminar')

document.getElementById("botonEliminar").addEventListener("click", function (event) {

event.preventDefault()

let productoEliminar = productoE.value

let van = false

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

if (productos[i].nombre == productoEliminar) {

productos.splice(i, 1)

van = true

}

}

if (van == false) {

mensaje.classList.add('noExsiteError')

setTimeout(() => { mensaje.classList.remove('noExsiteError') }, 2500);

}

else {

mensaje.classList.add('realizado')

setTimeout(() => {

mensaje.classList.remove('realizado')

window.location.reload()

}, 1500);

}

guardarAlmacenamientoLocal('productos', productos);

})

// mostrar productos

window.addEventListener("load", () => {

const productoEd = document.getElementById('productoEditar')

const productoEl = document.getElementById('productoEliminar')

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

productoEd.innerHTML += `<option>${productos[i].nombre}</option>`

productoEl.innerHTML += `<option>${productos[i].nombre}</option>`

}

Object.keys(productos[0]).forEach(element => {

atributoEd.innerHTML += `<option>${element}</option>`

});

let mostraProductos = document.getElementById('mostrarProductos')

mostraProductos.innerHTML = ''

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

mostraProductos.innerHTML += `<div class="contenedorProductos"><img src="${productos[i].urlImagen}"><div class="informacion"><p>${productos[i].nombre}</p><p class="precio"><span>Precio: ${productos[i].valor}$</span></p> Existencia: ${productos[i].existencia}<p></p></div></div>`

}

})

Y el se estilos styles.css

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

color: white;

}

body {

font-family: 'Space Grotesk', sans-serif;

background-color: #141C29;

padding-top: 80px;

}

h1 {

font-family: 'Montserrat', sans-serif;

text-align: center;

margin: 30px 0px;

}

img {

width: 320px;

height: 340px;

object-fit: cover;

border-radius: 20px 20px 0px 0px;

}

header {

display: flex;

align-items: center;

justify-content: center;

gap: 15%;

height: 70px;

width: 100%;

}

.contenedor {

display: flex;

justify-content: space-between;

align-items: flex-start;

margin: 50px auto 45px auto;

width: 90%;

flex-wrap: wrap;

gap: 40px;

}

.contenedor>div {

display: flex;

align-items: center;

justify-content: flex-start;

background-color: #2D4263;

height: 400px;

width: 400px;

border-radius: 20px;

flex-direction: column;

padding-top: 35px;

gap: 25px;

flex: 1 1 300px;

}

.contenedor>div>form {

display: flex;

flex-direction: column;

gap: 5px;

}

.contenedor>div>form>input,

select {

border-radius: 5px;

border: none;

padding: 5px;

color: black;

}

option{

color: black;

}

.contenedor .button {

margin-top: 20px;

height: 30px;

border-radius: 7px;

border: none;

color: black;

}

.contenedorProductos h2 {

text-align: center;

padding-bottom: 20px;

font-weight: 300;

font-size: 40px;

}

.mostrarProductos {

width: 90%;

margin: auto;

display: flex;

justify-content: center;

gap: 50px;

margin-bottom: 100px;

flex-wrap: wrap;

}

.mostrarProductos>div {

box-shadow: 0px 0px 5px 0px rgba(0, 0, 0, 0.75);

background-color: #2D4263;

border-radius: 20px;

}

.informacion {

height: 150px;

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

gap: 15px;

}

.informacion>p:first-child {

font-family: 'Montserrat', sans-serif;

font-size: 25px;

font-weight: 300;

}

.informacion .precio {

font-size: 20px;

font-weight: 500;

}

.contenedorMensaje {

display: flex;

justify-content: center;

width: 100%;

margin-left: auto;

height: 70px;

}

.repetidoError,

.realizado,

.llenarCampos,

.noExsiteError {

margin: 15px 0px;

}

.llenarCampos::after,

.noExsiteError::after,

.repetidoError::after,

.realizado::after {

background-color: red;

font-size: 17px;

padding: 10px;

border-radius: 10px;

}

.realizado::after {

content: 'Realizado';

background-color: green;

}

.repetidoError::after {

content: 'Este elemento ya existe';

}

.llenarCampos::after {

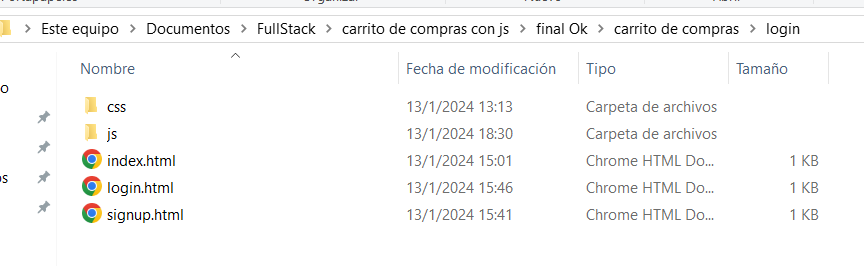
content: 'Por favor compruebe que todos los campos estan llenos';

}

.noExsiteError::after {

content: 'Este elemento no existe';

}



El archivo de estilos dentro de CSS es style.css

:root{

--font-mono: ui-monospace, Menlo, Monaco, 'Cascadia Mono', 'Segoe UI Mono',

'Roboto Mono', 'Oxygen Mono', 'Ubuntu Monospace', 'Source Code Pro',

'Fira Mono', 'Droid Sans Mono', 'Courier New', monospace;

--transition: all 0.1s cubic-bezier(0.645, 0.045, 0.355, 1)

}

\* {

margin:0;

padding:0;

box-sizing:border-box;/\*tamaño de la caja\*/

}

body{

font-family: var(--font-mono);

}

section{

display:flex;

flex-wrap: wrap;/\* especifica si los elementos "hijos" son obligados a permanecer en una misma línea o pueden fluir en varias líneas\*/

height: 100vh;

align-items: center;

justify-content: center;/\*define cómo el navegador distribuye el espacio entre y alrededor de los elementos flex, a lo largo del eje principal de su contenedor.\*/

flex-direction: column;/\* determina la dirección en que se ordenarán los elementos en el espacio. Esto puede ser de izquierda a derecha (row), de derecha a izquierda (row-reverse), de arriba a abajo (column) o de abajo a arriba (column-reverse) ( flex-flow flex-direction ).\*/

padding: 1em;/\*establece el área de relleno arriba | derecha | abajo | izquierda\*/

}

section h1{

font-size: 3em;

}

section h2{

text-align:center;

width: 100%;

margin-bottom: 1em;/\*( margen-inferior ) es la propiedad de un elemento que establece el espacio requerido en la parte inferior de un elemento\*/

font-size: 2.5em

}

section form{

padding: 1.5em;/\*establece el área de relleno arriba | derecha | abajo | izquierda\*/

display:flex;

flex-wrap: wrap;/\*especifica si los elementos "hijos" están obligados a permanecer en una misma línea o pueden fluir en varias líneas. Si la cobertura (wrap) está permitida, esta propiedad también te permite controlar la dirección en la cual serán apilados los elementos.\*/

max-width: 500px;/\* is used to define the maximum width of an element\*/

border-radius: 20px;

border: 2px solid rgba(0,0,0,.1);

}

section form label{

width: 100%;

padding: 0.2em;

margin-bottom: 0.5em;/\*es la propiedad de un elemento que establece el espacio requerido en la parte inferior de un elemento\*/

}

section form input{

width: 100%;

margin-bottom: 1em;/\*es la propiedad de un elemento que establece el espacio requerido en la parte inferior de un elemento\*/

padding: 0.8em;/\*arriba y abajo | izquierda y derecha\*/

border-radius: 9px;

border: none;

outline-color: #003DFF;/\*es una forma reducida para establecer una o más de las propiedades individuales de contorno outline-style , outline-width y outline-color en una sola declaración\*/

background-color: #eee;

}

section form input[type='submit']{

margin-top: 1.5em;/\*establece el margen superior de un elemento especificando una longitud o un porcentaje. Los valores en porcentaje se refieren al ancho del elemento padre.\*/

background-color: #003DFF;

color: #fff;

font-family: var(--font-mono);

transition: var(--transition);

}

section form input[type='submit']:active{

transform: scale(0.97);

}

section p{

margin-top: 1.5em;/\*establece el margen superior de un elemento especificando una longitud o un porcentaje. Los valores en porcentaje se refieren al ancho del elemento padre.\*/

}

section p a{

text-decoration: none;

color: #003DFF;

}

/\* Home \*/

section button{

background-color: #003DFF;

color: #fff;

font-family: var(--font-mono);

transition: var(--transition);

border: none;

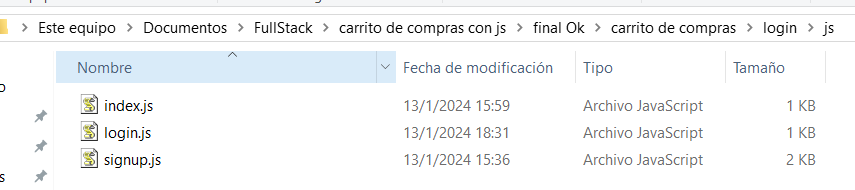
padding: 0.5em 1em;/\* arriba y abajo | izquierda y derecha \*/

border-radius: 9px;

margin-top: 1.5em;

font-size: 1em;

}



Index.js

//busca la variable creada en el login.js

const user = JSON.parse(localStorage.getItem('login\_success')) || false

if(!user){

window.location.href = 'login.html'//si el usuario no ingreso correct// los datos

}

const logout = document.querySelector('#logout')

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

alert('Hasta pronto!')

localStorage.removeItem('login\_success')//vaciamos el storage

window.location.href = 'login.html'

})

const loginForm = document.querySelector('#loginForm')

loginForm.addEventListener('submit', (e)=>{

e.preventDefault()/\*evitar que se recargue el html\*/

const email = document.querySelector('#email').value

const password = document.querySelector('#password').value

//traemos la bsase del storage, users es la clave del storage

const Users = JSON.parse(localStorage.getItem('users')) || []

const validUser = Users.find(user => user.email === email && user.password === password)

if(!validUser){//buscar si el mail y clave coinciden, se logueo correctamente

return alert('Usuario y/o contraseña incorrectos!')

}

alert(`Bienvenido ${validUser.name}`)

localStorage.setItem('login\_success', JSON.stringify(validUser))//capturo el usuario logueado desde el storage

window.location.href = '../admin/admin.html'

})

Signup.js

//creamos una mini base de datos

const signupForm = document.querySelector('#signupForm')

signupForm.addEventListener('submit', (e)=>{//le agregamos el evento submit

e.preventDefault()//para que no se recargue el formulario

const name = document.querySelector('#name').value//input del form

const email = document.querySelector('#email').value//idem

const password = document.querySelector('#password').value//idem

//mi base, si tiene un usuario lo guarda en el storage y si no que asigne un array vacio

const Users = JSON.parse(localStorage.getItem('users')) || []

const isUserRegistered = Users.find(user => user.email === email)//comprara el mail que se acaba de ingresar con el del localstorage

if(isUserRegistered){//por verdadero, quiere decir que existe

return alert('El usuario ya esta registado!')

}

//si es false , no esta registrado, por lo cual asigna los valores

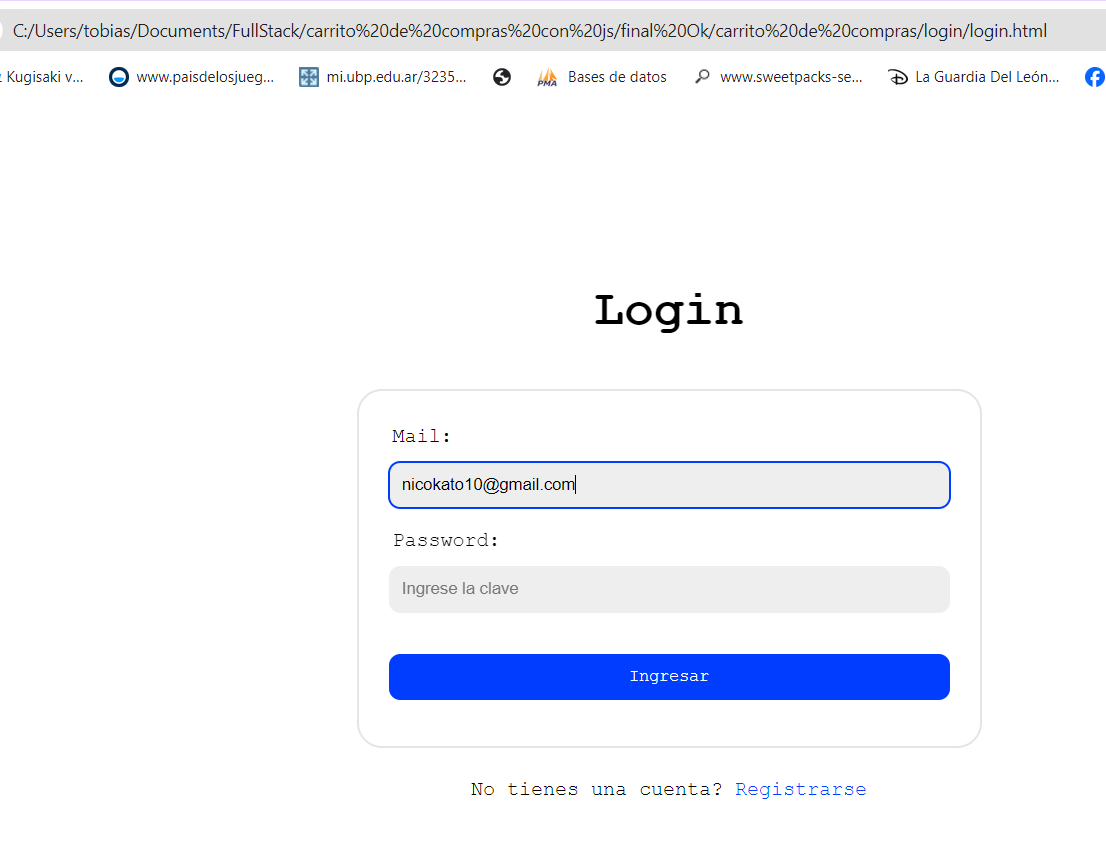
Users.push({name: name, email: email, password: password})

localStorage.setItem('users', JSON.stringify(Users))

alert('Registro Exitoso!')

window.location.href = 'login.html'

})



Login.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="css/styles.css">

<title>Ingreso al sistema</title>

</head>

<body>

<section>

<h2>Login</h2>

<form id="loginForm">

<label>Mail:</label>

<input type="email" name="" id="email" placeholder="Ingrese el mail" required autofocus>

<label>Password:</label>

<input type="password" name="" id="password" placeholder="Ingrese la clave" required>

<input type="submit" value="Ingresar">

</form>

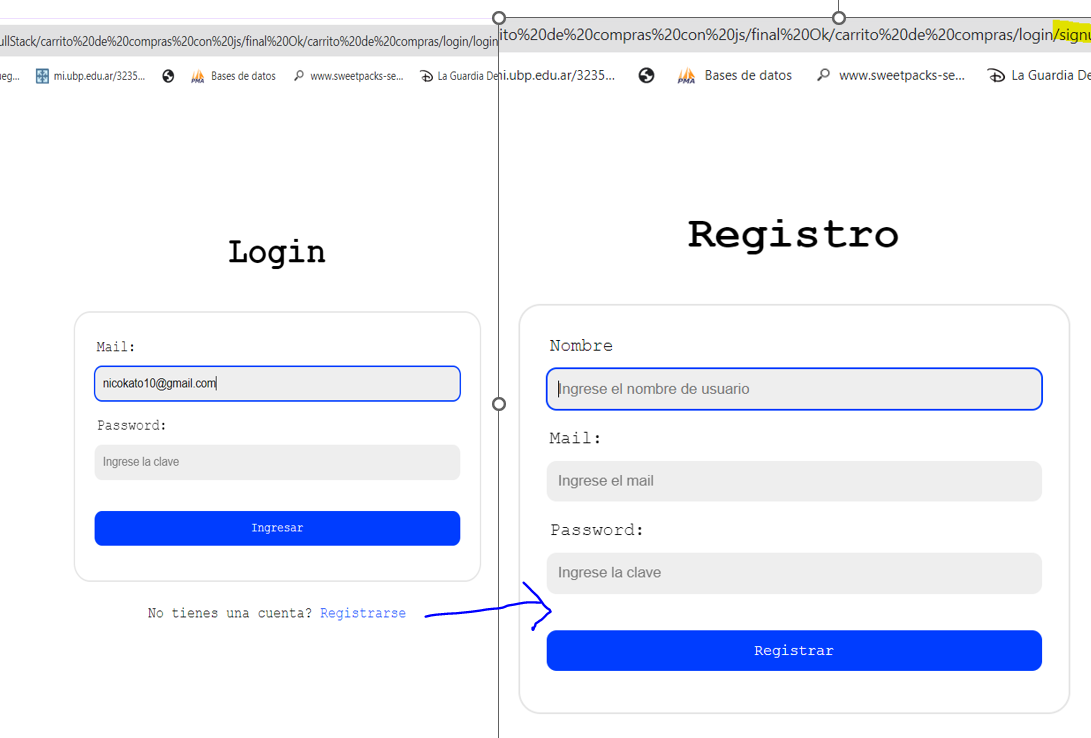
<p>No tienes una cuenta? <a href="signup.html">Registrarse</a></p>

</section>

<script src="js/login.js"></script>

</body>

</html>



Signup.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="css/styles.css">

<title>Registro</title>

</head>

<body>

<section>

<h2>Registro</h2>

<form id="signupForm">

<label>Nombre</label>

<input type="text" name="nombre" id="name" placeholder="Ingrese el nombre de usuario" required autofocus>

<label>Mail:</label>

<input type="email" name="" id="email" placeholder="Ingrese el mail" required>

<label>Password:</label>

<input type="password" name="" id="password" placeholder="Ingrese la clave" required>

<input type="submit" value="Registrar">

</form>

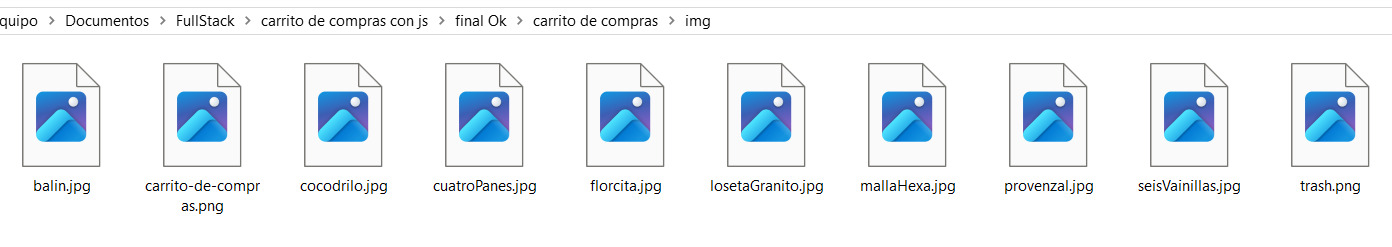
<p>Ya tienes una cuenta? <a href="login.html">Login</a></p>

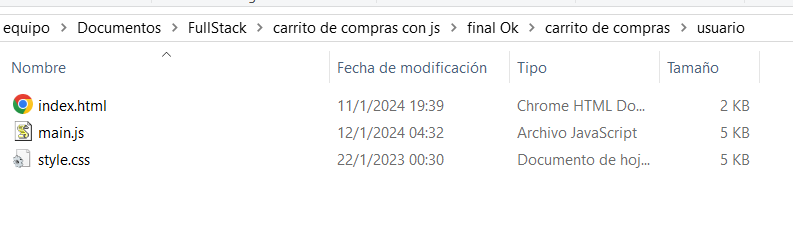
</section>

<script src="js/signup.js"></script>

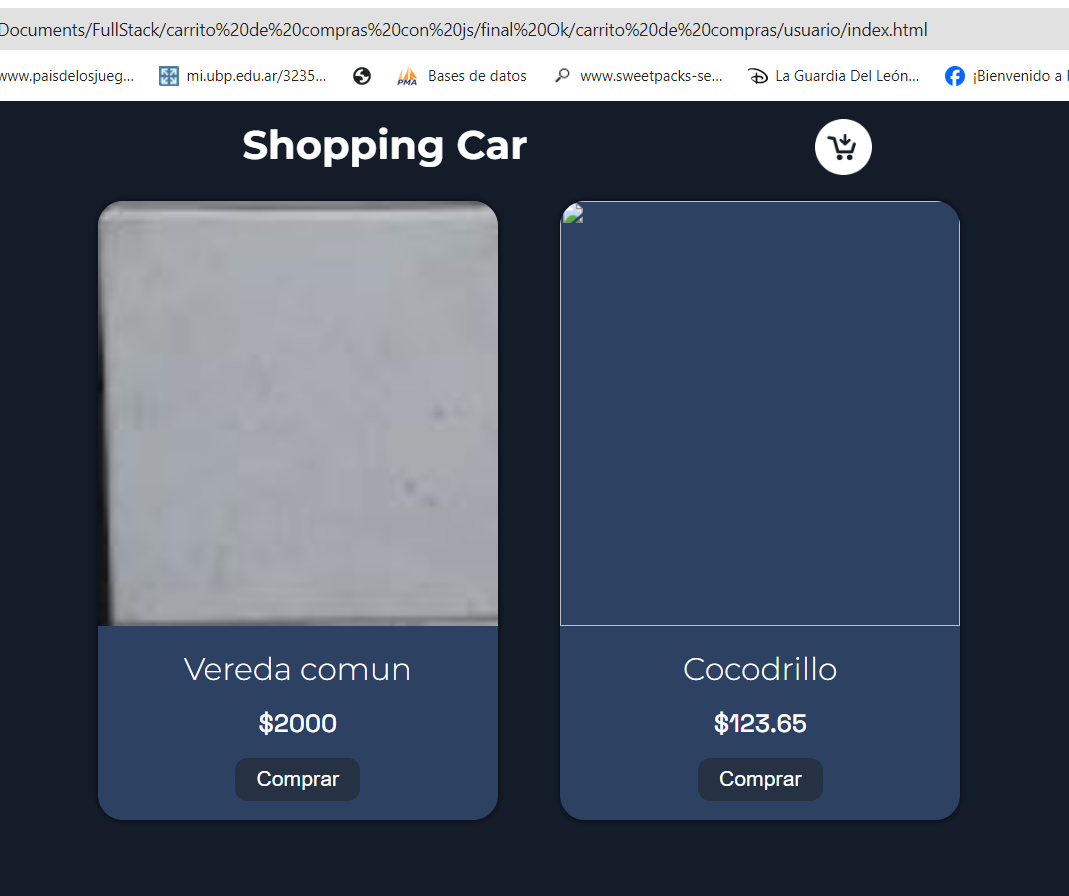
</body>

</html>





Index.html



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="style.css" rel="stylesheet" type="text/css">

<title>Carrito de compras</title>

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Space+Grotesk:wght@400;500;600;700&display=swap"

rel="stylesheet">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@300;400;500;600;700&display=swap"

rel="stylesheet">

</head>

<body>

<header id="header">

<h1>Shopping Car</h1>

<div>

<img id="carrito" class="carrito" src="../img/carrito-de-compras.png" alt="">

<div id="numero"></div>

</div>

</header>

<main>

<div id="contenedor" class="contenedor"> </div>

</main>

<div id="contenedorCompra">

<div class="informacionCompra" id="informacionCompra">

<h2>Cesta</h2>

<p class="x" id="x">x</p>

</div>

<div class="productosCompra" id="productosCompra"></div>

<div class="total" id="total"></div>

<button>Finalizar Compra</button>

</div>

<script src="main.js"></script>

</body>

</html>

Main.js

// Funciones para almacenar y traer los datos que se almacenan

function guardarAlmacenamientoLocal(llave, valor\_a\_guardar) {

localStorage.setItem(llave, JSON.stringify(valor\_a\_guardar))

}

function obtenerAlmacenamientoLocal(llave) {

const datos = JSON.parse(localStorage.getItem(llave))

return datos

}

let productos = obtenerAlmacenamientoLocal('productos') || [];

// Variables que traemos de nuestro html

const informacionCompra = document.getElementById('informacionCompra');

const contenedorCompra = document.getElementById('contenedorCompra');

const productosCompra = document.getElementById('productosCompra');

const contenedor = document.getElementById('contenedor');

const carrito = document.getElementById('carrito');

const numero = document.getElementById("numero");

const header = document.querySelector("#header");

const total = document.getElementById('total');

const body = document.querySelector("body");

const x = document.getElementById('x')

// Variables que vamos a usar en nuestoro proyecto

let lista = []

let valortotal = 0

// Scroll de nuestra pagina

window.addEventListener("scroll", function () {

if (contenedor.getBoundingClientRect().top < 10) {

header.classList.add("scroll")

}

else {

header.classList.remove("scroll")

}

})

window.addEventListener('load', () => {

visualizarProductos();

contenedorCompra.classList.add("none")

})

function visualizarProductos() {

contenedor.innerHTML = ""

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

if (productos[i].existencia > 0) {

contenedor.innerHTML += `<div><img src="${productos[i].urlImagen}"><div class="informacion"><p>${productos[i].nombre}</p><p class="precio">$${productos[i].valor}</p><button onclick=comprar(${i})>Comprar</button></div></div>`

}

else {

contenedor.innerHTML += `<div><img src="${productos[i].urlImagen}"><div class="informacion"><p>${productos[i].nombre}</p><p class="precio">$${productos[i].valor}</p><p class="soldOut">Sold Out</p></div></div>`

}

}

}

function comprar(indice) {

lista.push({ nombre: productos[indice].nombre, precio: productos[indice].valor })

let van = true

let i = 0

while (van == true) {

if (productos[i].nombre == productos[indice].nombre) {

productos[i].existencia -= 1

if (productos[i].existencia == 0) {

visualizarProductos()

}

van = false

}

guardarAlmacenamientoLocal("productos", productos)

i += 1

}

numero.innerHTML = lista.length

numero.classList.add("diseñoNumero")

return lista

}

carrito.addEventListener("click", function(){

body.style.overflow = "hidden"

contenedorCompra.classList.remove('none')

contenedorCompra.classList.add('contenedorCompra')

informacionCompra.classList.add('informacionCompra')

mostrarElemtrosLista()

})

function mostrarElemtrosLista() {

productosCompra.innerHTML = ""

valortotal = 0

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

productosCompra.innerHTML += `<div><div class="img"><button onclick=eliminar(${i}) class="botonTrash"><img src="../img/trash.png"></button><p>${lista[i].nombre}</p></div><p> $${lista[i].precio}</p></div>`

valortotal += parseInt(lista[i].precio)

}

total.innerHTML = `<p>Valor Total</p> <p><span>$${valortotal}</span></p>`

}

function eliminar(indice){

let van = true

let i = 0

while (van == true) {

if (productos[i].nombre == lista[indice].nombre) {

productos[i].existencia += 1

lista.splice(indice, 1)

van = false

}

i += 1

}

guardarAlmacenamientoLocal("productos", productos)

numero.innerHTML = lista.length

if (lista.length == 0){

numero.classList.remove("diseñoNumero")

}

visualizarProductos()

mostrarElemtrosLista()

}

x.addEventListener("click", function(){

body.style.overflow = "auto"

contenedorCompra.classList.add('none')

contenedorCompra.classList.remove('contenedorCompra')

informacionCompra.classList.remove('informacionCompra')

})

Style.css

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

color: white;

}

body {

font-family: 'Space Grotesk', sans-serif;

background-color: #141C29;

padding-top: 80px;

}

h1{

font-family: 'Montserrat', sans-serif;

text-align: center;

margin: 30px 0px;

}

img {

width: 320px;

height: 340px;

object-fit: cover;

border-radius: 20px 20px 0px 0px;

}

header{

display: flex;

align-items: center;

justify-content: center;

gap:15%;

height: 70px;

top: 0;

position: fixed;

width: 100%;

}

.scroll{

background-color: aliceblue;

box-shadow: .1rem .2rem .3rem #333;

}

.scroll img{

filter: invert(15%);

}

.scroll h1{

color: #141C29;

}

.carrito{

filter: invert(100%);

width: 45px;

height: auto;

position: absolute;

top: 14px;

cursor: pointer;

}

.contenedor{

display: flex;

justify-content: center;

flex-wrap: wrap;

gap: 50px;

margin-bottom: 20px;

}

.contenedor>div{

box-shadow: 0px 0px 5px 0px rgba(0, 0, 0, 0.75);

background-color: #2D4263;

border-radius: 20px;

}

.contenedor>div:hover{

background-color: #11294d;

}

.informacion{

height: 150px;

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

gap:15px;

}

.informacion>p:first-child{

font-family: 'Montserrat', sans-serif;

font-size: 25px;

font-weight: 300;

}

.informacion .precio {

font-size: 20px;

font-weight: 500;

}

.informacion .precio span{

font-size: 17px;

font-weight: 300;

}

.informacion button{

background-color: #263143;

width: 100px;

height: 35px;

border-radius: 10px;

border: none;

font-size: 17px;

font-weight: 300;

cursor: pointer;

}

.informacion button:hover{

background-color: #2D4263;

}

.infromacio button:active{

background-color: white;

color: black;

}

.diseñoNumero {

padding: 10px;

background-color: red;

border-radius: 100%;

width: 20px;

height: 20px;

font-size: 10px;

position: relative;

top: -12px;

left: 30px;

z-index: 1;

display: flex;

justify-content: center;

align-items: center;

}

.contenedorCompra{

display: flex;

align-items: center;

justify-content: center;

width: 100%;

height: 100%;

position: fixed;

top:0px;

background-color: rgba(51, 51, 51, 0.6);

flex-direction:column;

gap:35px;

}

.informacionCompra{

width: 500px;

height: 500px;

background-color: #2D4263;

border-radius: 25px;

display: flex;

align-items: flex-start;

justify-content: flex-start;

position: fixed;

}

.informacionCompra h2{

font-weight: 300;

margin: 50px 0px 0px 50px;

font-size: 32px;

width: 100%;

}

.x{

padding-top: 30px;

padding-right: 30px;

cursor: pointer;

font-size: 20px;

color: white;

}

.productosCompra{

font-size: 17px;

position: relative;

z-index: 1;

display: flex;

justify-content: flex-start;

flex-direction: column;

width: 450px;

height: 30%;

overflow-y: auto;

margin-top: 50px;

}

.productosCompra > div {

margin: 0px 45px;

padding: 20px 0px;

width: auto;

border-bottom: 1px dotted black;

display: flex;

justify-content: space-between;

}

.productosCompra::-webkit-scrollbar{

width: 7px;

background-color: #141C29;

border-radius: 5px;

}

.productosCompra::-webkit-scrollbar-thumb{

background: white ;

border-radius: 3px;

}

.img{

display: flex;

flex-direction: row;

align-items: center;

}

.productosCompra img{

width: 25px;

height: auto;

margin-right:10px;

border-bottom: none ;

cursor: pointer;

}

.total{

z-index: 1;

display: flex;

align-items: flex-start;

justify-content: space-between;

gap: 150px;

font-size: 27px;

font-weight: 700;

}

.total span{

font-weight: 300;

}

.contenedorCompra > button{

z-index: 1;

background-color: #263143;

width: 300px;

height: 45px;

border-radius: 10px ;

border: none;

font-size: 17px;

font-weight: 300;

}

.contenedorCompra > button:hover{

background-color: white;

color: black;

}

.none>div,

.none button,

.none p{

width: 0px;

height: 0px;

font-size: 0px;

border: 0px;

}

.soldOut{

background-color: red;

padding: 7px 10px;

border-radius: 10px;

}

.botonTrash{

background-color: rgba(0, 0, 0, 0);

border:none;

display: flex;

justify-content: center;

align-items: center;

}