**HTML**

1. Encabezado del documento (<head>)

<head>

...

<title>Finalizar Compra - Kick Store</title>

...

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

</head>

Define el título de la pestaña del navegador.

Carga fuentes desde Google Fonts: Montserrat, Open Sans y Bebas Neue.

Carga el archivo de estilos externo stylepago.css.

2. Encabezado del sitio (<header>)

<header>

<div class="logo"><a href="pagina.html">Kick Store</a></div>

<nav>...</nav>

</header>

Contiene el logo (que también es un enlace a la página principal) y un menú de navegación.

Muestra el ícono del carrito con el número de artículos (cart-count), aunque no es interactivo en esta página (no hay onclick).

3. Contenido principal (<main>)

Dividido en dos secciones importantes:

📝 Formulario de Pago

<div class="checkout-form">

<h2>Datos de envío y pago</h2>

<form id="paymentForm">

...

<button type="submit">Confirmar Pago</button>

</form>

</div>

Recoge:

Nombre completo

Dirección

Número de tarjeta (valida 16 dígitos)

Fecha de vencimiento (formato MM/AA)

CVV (3 o 4 dígitos)

Estos campos tienen validaciones básicas HTML5 (required, pattern, title).

🛍️ Resumen del Pedido

<div class="summary">

<h2>Resumen del Pedido</h2>

<ul id="cartSummaryList"></ul>

<p class="total-line">Total: $<span id="cartSummaryTotal">0</span> MXN</p>

</div>

Se llena dinámicamente con los productos del carrito almacenado en localStorage.

Muestra el total en pesos mexicanos.

4. Pie de página (<footer>)

<footer>

&copy; 2025 Kick Store - Todos los derechos reservados

</footer>

Pie simple con derechos de autor.

5. Script al final (<script src="scriptpago.js">)

Este archivo es el encargado de:

Mostrar los productos guardados del carrito.

Calcular el total.

Validar y procesar el formulario de pago

**CSS**

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

font-family: 'Open Sans', sans-serif;

}

Aplica a todos los elementos.

Elimina márgenes y rellenos por defecto.

Usa Open Sans como fuente base para todo el documento.

Usa box-sizing: border-box para que el tamaño de los elementos sea más controlado.

body {

background: linear-gradient(to bottom, #020001, #340016);

color: #FE6C2B;

min-height: 100vh;

display: flex;

flex-direction: column;

}

Fondo con degradado vertical de negro oscuro (#020001) a rojo vino (#340016).

Color del texto principal: naranja fuerte (#FE6C2B).

Usa flex-direction: column para que el footer quede al final de la página.

header {

background-color: #020001;

padding: 20px 50px;

color: #FE6C2B;

display: flex;

justify-content: space-between;

align-items: center;

box-shadow: 0 4px 10px rgba(0,0,0,0.8);

}

Fondo negro oscuro.

Espaciado interno (padding) y diseño flexible para que logo y menú se alineen.

box-shadow: agrega una sombra para dar un efecto elevado.

.logo a {

font-family: 'Montserrat';

font-weight: 700;

font-size: 1.8rem;

color: #FE6C2B;

text-decoration: none;

user-select: none;

}

El logo "Kick Store" usa letra Montserrat en negrita y grande.

Es un enlace, pero sin subrayado.

nav a {

margin-left: 30px;

font-family: 'Montserrat';

font-weight: 700;

color: #FE6C2B;

text-decoration: none;

transition: color 0.3s;

user-select: none;

}

nav a:hover {

color: #D43B2D;

}

nav a.active {

color: #D43B2D;

text-decoration: underline;

}

Enlaces de navegación usan la fuente Montserrat.

Cambian de color al pasar el mouse (de naranja a rojo más intenso).

La clase .active destaca la sección actual.

.cart-icon {

position: relative;

cursor: pointer;

margin-left: 20px;

font-size: 1.6rem;

color: #FE6C2B;

user-select: none;

transition: color 0.3s;

}

.cart-count {

position: absolute;

top: -10px;

right: -10px;

background: #D43B2D;

color: white;

border-radius: 50%;

padding: 2px 6px;

font-size: 0.75rem;

font-weight: 700;

font-family: 'Montserrat';

}

Ícono tipo emoji.

La burbuja (cart-count) aparece encima con el número de productos, en rojo.

main {

max-width: 900px;

margin: 40px auto;

display: flex;

gap: 40px;

background: #340016;

padding: 30px;

border-radius: 15px;

box-shadow: 0 8px 20px #020001;

color: #FE6C2B;

user-select: none;

}

Sección centrada y con sombra.

Contiene dos columnas:

.checkout-form: el formulario de pago.

.summary: el resumen del pedido.

label {

display: block;

margin: 15px 0 5px;

font-weight: 600;

font-family: 'Montserrat';

color: #FE6C2B;

text-shadow: 1px 1px 2px #020001;

user-select: text;

}

Fuente Montserrat, con sombra ligera.

Bien espaciadas para claridad.

input {

width: 100%;

padding: 12px;

border: 1px solid #9F102C;

border-radius: 8px;

font-size: 1rem;

background-color: #020001;

color: #FE6C2B;

transition: border-color 0.3s;

user-select: text;

}

input:focus {

border-color: #FE6C2B;

box-shadow: 0 0 8px #FE6C2B;

}

Fondo oscuro, texto naranja.

Cuando el usuario escribe (focus), se ilumina el borde.

button {

font-family: 'Bebas Neue';

background-color: #FE6C2B;

color: #020001;

padding: 15px;

width: 100%;

border: none;

border-radius: 10px;

font-size: 1.3rem;

cursor: pointer;

box-shadow: 0 4px 10px #9F102C;

transition: background-color 0.3s, color 0.3s;

user-select: none;

}

button:hover {

background-color: #D43B2D;

color: #fff;

}

Usa la fuente Bebas Neue, que es llamativa.

Colores invertidos al pasar el cursor.

Efectos de sombra para destacar.

.summary ul {

list-style: none;

padding: 0;

margin: 0;

max-height: 350px;

overflow-y: auto;

}

.summary li {

display: flex;

justify-content: space-between;

margin-bottom: 12px;

font-weight: 600;

color: #FFB86C;

text-shadow: 0 0 2px #020001;

}

.total-line {

border-top: 1px solid #9F102C;

padding-top: 10px;

margin-top: 10px;

font-weight: 700;

font-family: 'Montserrat';

font-size: 1.4rem;

color: #FE6C2B;

text-shadow: 1px 1px 3px #020001;

user-select: text;

}

Lista con scroll si hay muchos productos.

Cada producto se muestra con su nombre y precio.

El total se destaca al final con línea separadora.

footer {

text-align: center;

padding: 20px;

background: #020001;

color: #FE6C2B;

font-family: 'Montserrat';

font-weight: 700;

font-size: 1rem;

user-select: none;

box-shadow: 0 -2px 5px rgba(0,0,0,0.7);

margin-top: 40px;

}

Fondo negro, texto naranja, centrado.

Fuente Montserrat en negrita.

Sombra superior para dar separación del contenido.

**SCRIPT**

**1. Cargar productos del carrito desde localStorage**

const cart = JSON.parse(localStorage.getItem('cart') || '[]');

* Recupera los productos guardados en el navegador como texto (localStorage.getItem('cart')) y los convierte a un arreglo de objetos con JSON.parse.
* Si no hay nada guardado, se crea un arreglo vacío [].

**📌 2. Referencias al DOM (elementos HTML)**

const cartList = document.getElementById('cartSummaryList');

const cartTotal = document.getElementById('cartSummaryTotal');

const cartCount = document.getElementById('cart-count');

* Accede a los elementos del HTML donde se mostrará:
  + La lista de productos del resumen.
  + El total a pagar.
  + El número total de productos (ícono del carrito).

**➕ 3. Mostrar productos en el resumen**

let total = 0;

cartList.innerHTML = '';

cart.forEach(item => {

const li = document.createElement('li');

li.innerHTML = `<span>${item.name}</span><span>$${item.price} MXN</span>`;

cartList.appendChild(li);

total += item.price;

});

* Limpia el contenido de la lista antes de agregar nuevos productos.
* Recorre cada producto del carrito:
  + Crea un nuevo elemento <li>.
  + Le pone el nombre y el precio del producto.
  + Lo agrega al HTML.
* Suma el total de precios.

**💰 4. Mostrar el total y la cantidad de productos**

cartTotal.textContent = total;

cartCount.textContent = cart.length;

* Muestra el total calculado.
* Muestra cuántos productos hay en el carrito.

**🧾 5. Procesar el formulario de pago**

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

paymentForm.addEventListener('submit', function (e) {

e.preventDefault();

alert('¡Gracias por tu compra!');

localStorage.removeItem('cart');

window.location.href = 'pagina.html';

});

* Cuando el usuario envía el formulario:
  + e.preventDefault() evita que se recargue la página.
  + Se muestra un mensaje de agradecimiento.
  + Se borra el carrito del localStorage (ya que la compra se finalizó).
  + Redirige al usuario a la página principal (pagina.html).