**HTML**

<!DOCTYPE html>

<html lang="es">

Indica que es un documento HTML5 y el idioma es español.

<head>

<meta charset="UTF-8" />

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

<title>Kick Store - Venta de Tenis</title>

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;600&display=swap" rel="stylesheet">

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

</head>

Dentro del <head> se define:

El juego de caracteres (UTF-8) para que se muestren bien los caracteres especiales.

La configuración del viewport para que la página sea responsive (se adapte a móviles).

El título que aparece en la pestaña del navegador.

Se carga la fuente Poppins desde Google Fonts (con pesos 300 y 600).

Se enlaza el archivo CSS externo llamado styles.css, que contiene los estilos.

<body>

<header>

<div class="logo"><img src="logo.png" ></div>

<nav>

<a href="#">Inicio</a>

<a href="#" id="loginMenuLink">Iniciar Sesión</a>

<a href="productos.html">Productos</a>

<a href="info.html">Información</a>

<a href="contactos.html">Contacto</a>

</nav>

</header>

En el <body> empieza el contenido visible.

El <header> contiene la parte superior:

Un div con clase "logo" que muestra la imagen del logo (archivo logo.png).

Un menú de navegación (nav) con enlaces:

Inicio (enlace a la misma página por ahora con href="#").

Iniciar Sesión con un ID para manipularlo con JavaScript.

Otros enlaces a las páginas Productos, Información y Contacto.

<section class="login-section" id="loginSection" style="display: none;">

<h2>Iniciar Sesión</h2>

<form id="loginForm">

<input type="text" id="username" placeholder="Usuario" required />

<input type="password" id="password" placeholder="Contraseña" required />

<button type="submit">Entrar</button>

</form>

</section>

Esta sección contiene el formulario para iniciar sesión, pero por defecto está oculta (style="display: none;").

Incluye:

Un título.

Un formulario con dos campos:

Usuario (texto).

Contraseña (campo oculto).

Un botón para enviar el formulario.

<section class="user-info" id="userInfo" style="display: none;">

<img src="saludo.gif" alt="Usuario" class="user-avatar">

<p id="welcomeMessage"></p>

<button id="logoutButton">Cerrar Sesión</button>

</section>

Otra sección oculta inicialmente.

Aquí se muestra la información del usuario una vez que ha iniciado sesión:

Imagen animada o avatar.

Un párrafo donde se mostrará un mensaje de bienvenida dinámico.

Un botón para cerrar sesión.

<section class="hero">

<div class="hero-text">

<h1>Kick Store</h1>

<p>En cada paso hay una historia. En cada suela, un destino. Nuestra tienda nació para acompañar tu camino, ya sea que corras detrás de tus sueños o simplemente quieras llegar más lejos. Distribuimos tennis que no solo se ven bien, sino que se mueven contigo.</p>

</div>

<img src="hola.jpg" alt="Tenis destacados">

</section>

La sección "hero" es la parte principal de bienvenida o presentación:

Texto grande y destacado (h1) con un párrafo explicativo.

Una imagen que representa los tenis destacados.

<section class="products" id="productos">

<h2>Nuestros Productos</h2>

<div class="product-list">

<div class="product">

<img src="hola1.jpg" alt="Tenis 1">

<h3>Tenis Urbanos</h3>

<p>Diseño moderno y comodidad asegurada para tu día a día.</p>

</div>

<div class="product">

<img src="hola2.jpg" alt="Tenis 2">

<h3>Tenis Deportivos</h3>

<p>Perfectos para entrenamientos y actividades físicas.</p>

</div>

<div class="product">

<img src="hola3.jpg" alt="Tenis 3">

<h3>Tenis Edición Limitada</h3>

<p>Modelos únicos para verdaderos coleccionistas.</p>

</div>

</div>

</section>

Sección donde se muestran productos destacados:

Título con h2.

Dentro de product-list hay 3 productos, cada uno con:

Imagen.

Título (h3).

Descripción corta.

<footer class="footer" id="contacto">

<p>&copy; 2025 Kick Store - Todos los derechos reservados</p>

</footer>

El pie de página con derechos reservados.

Se puede usar el ID contacto para enlazar desde el menú o para aplicar estilos específicos.

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

</body>

</html>

Al final se incluye un archivo JavaScript (script.js) para manejar interactividad, como mostrar y ocultar la sección de login, guardar el usuario, etc.

**CSS**

@import url('https://fonts.googleapis.com/css2?family=Montserrat:wght@700&family=Open+Sans:wght@400&family=Bebas+Neue&display=swap');

Trae las fuentes desde Google Fonts:

Montserrat (con peso 700 para títulos).

Open Sans (peso 400 para texto general).

Bebas Neue (para frases especiales o citas).

2. Reset básico y configuración global

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

Quita márgenes y rellenos por defecto en todos los elementos.

Usa box-sizing: border-box para que el padding y borde se incluyan dentro del ancho/alto que asignemos.

3. Estilos generales del cuerpo

body {

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

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

color: #FE6C2B;

}

Fuente base: Open Sans para lectura cómoda.

Fondo con degradado de tonos oscuros púrpura a casi negro.

Color de texto en naranja para contraste.

4. Encabezado (header)

header {

background: #020001;

color: #FE6C2B;

display: flex;

justify-content: space-between;

align-items: center;

padding: 20px 50px;

box-shadow: 0 4px 10px rgba(0,0,0,0.3);

}

Fondo oscuro.

Texto naranja.

Uso de flex para alinear logo y menú en extremos opuestos.

Sombra sutil para dar profundidad.

Relleno amplio para separar contenido de los bordes.

5. Logo

.logo {

width: 120px;

height: 60px;

background-color: #FE6C2B;

border-radius: 8px;

display: flex;

justify-content: center;

align-items: center;

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

font-weight: bold;

color: #020001;

}

Caja rectangular con fondo naranja (color principal).

Texto en color oscuro.

Fuente Montserrat negrita para destacar.

Centrado vertical y horizontal con flexbox.

Bordes redondeados para suavizar la forma.

6. Navegación (nav a)

nav a {

margin-left: 30px;

text-decoration: none;

color: #FE6C2B;

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

font-weight: 700;

transition: color 0.3s;

}

nav a:hover {

color: #D43B2D;

}

Los enlaces tienen separación a la izquierda para espaciar entre ellos.

Sin subrayado.

Texto naranja, fuente Montserrat en negrita.

Efecto suave (transición) al cambiar color cuando el usuario pasa el cursor (hover).

Color de hover cambia a un rojo más oscuro.

7. Sección de inicio de sesión (.login-section)

.login-section {

padding: 40px;

text-align: center;

background-color: #340016;

color: #FE6C2B;

}

.login-section h2 {

margin-bottom: 20px;

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

}

.login-section form input {

padding: 10px;

margin: 10px;

width: 200px;

border-radius: 8px;

border: 1px solid #9F102C;

background-color: #020001;

color: #FE6C2B;

}

.login-section form button {

padding: 10px 20px;

background-color: #9F102C;

color: white;

border: none;

border-radius: 8px;

cursor: pointer;

}

Fondo púrpura oscuro.

Texto naranja para mantener coherencia.

Inputs con padding, borde redondeado y borde color vino.

Botón con fondo vino oscuro, texto blanco y cambio de cursor al pasar por encima para indicar que es clickeable.

8. Sección de información del usuario (.user-info)

.user-info {

padding: 20px;

display: flex;

align-items: center;

gap: 15px;

background-color: #D43B2D;

color: white;

margin: 20px;

border-radius: 12px;

animation: fadeIn 0.5s ease-in-out;

}

.user-avatar {

width: 80px;

height: 80px;

border-radius: 12px;

border: 2px solid #FE6C2B;

object-fit: cover;

}

Fondo rojo vino, texto blanco para alto contraste.

Flexbox para que la imagen y texto queden alineados en una fila.

Separación entre elementos con gap.

Bordes redondeados para suavizar.

Animación de entrada (fadeIn) para que aparezca con suavidad.

Imagen del usuario en tamaño cuadrado, borde naranja y esquinas redondeadas.

9. Animación fadeIn

@keyframes fadeIn {

from { opacity: 0; transform: translateY(-10px); }

to { opacity: 1; transform: translateY(0); }

}

Aparece la sección desvaneciéndose y bajando suavemente 10px para dar dinamismo.

10. Sección Hero (presentación principal)

.hero {

display: flex;

align-items: center;

justify-content: space-between;

padding: 80px 50px;

background: linear-gradient(135deg, #9F102C, #D43B2D);

color: white;

}

.hero-text {

max-width: 600px;

}

.hero-text h1 {

font-size: 3em;

margin-bottom: 20px;

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

}

.hero-text p {

font-size: 1.2em;

}

.hero img {

width: 400px;

border-radius: 20px;

box-shadow: 0 0 20px rgba(0,0,0,0.4);

}

Fondo degradado vino, texto blanco para que destaque.

Flexbox para poner texto y la imagen lado a lado.

Título grande, fuente Montserrat.

Imagen con bordes redondeados y sombra para dar profundidad.

11. Productos

.products {

padding: 50px;

background-color: #340016;

color: #FE6C2B;

}

.products h2 {

text-align: center;

font-size: 2em;

margin-bottom: 40px;

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

}

.product-list {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 30px;

}

.product {

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

border-radius: 20px;

padding: 20px;

box-shadow: 0 8px 20px rgba(0,0,0,0.3);

text-align: center;

}

.product img {

width: 100%;

border-radius: 12px;

}

.product h3 {

margin: 15px 0 10px;

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

}

.product p {

color: #f9f9f9;

}

Fondo púrpura oscuro.

El título está centrado y grande.

Uso de grid para que los productos se adapten al ancho y tengan un mínimo tamaño.

Cada producto tiene fondo con degradado, bordes redondeados y sombra.

Imagen ocupa todo el ancho del producto con bordes redondeados.

Texto del producto en color claro para buen contraste.

12. Footer

.footer {

background-color: #020001;

color: #FE6C2B;

text-align: center;

padding: 30px;

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

}

Fondo negro oscuro.

Texto naranja.

Centrado y con espacio para que se vea limpio.

13. Estilos específicos para mensajes y botón logout

#welcomeMessage {

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

text-align: center;

color: #000000;

font-size: 1.3rem;

line-height: 1.6;

}

.mensaje-extra {

display: block;

margin-top: 10px;

color: #f0f0f0;

font-size: 1rem;

}

#logoutButton {

padding: 10px 25px;

background: linear-gradient(135deg, #9F102C, #D43B2D);

color: #fff;

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

font-weight: bold;

border: none;

border-radius: 10px;

cursor: pointer;

transition: all 0.3s ease;

box-shadow: 0 4px 15px rgba(0, 0, 0, 0.3);

}

#logoutButton:hover {

background: linear-gradient(135deg, #D43B2D, #FE6C2B);

transform: scale(1.05);

}

Mensaje de bienvenida con fuente Montserrat, centrado y negro (para que destaque en el fondo rojo).

Botón de cerrar sesión con degradado, sin borde, bordes redondeados, sombra y animación al pasar el cursor para hacerlo más visible y amigable.

**SCRIPT**

document.addEventListener("DOMContentLoaded", function () {

// Aquí va todo el código para que se ejecute cuando la página termine de cargar

});

Esto asegura que el código dentro solo se ejecute cuando todo el contenido HTML esté cargado y listo.

Así evitamos errores por intentar acceder a elementos que aún no existen.

2. Variables que almacenan elementos HTML

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

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

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

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

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

Usamos getElementById para guardar referencias a elementos importantes de la página:

El enlace del menú que abre el login.

La sección del formulario de login.

El formulario en sí.

La sección donde mostramos la información del usuario.

El mensaje de bienvenida.

3. Mostrar formulario al hacer clic en el menú

loginMenuLink.addEventListener("click", function (e) {

e.preventDefault(); // Evita que el enlace recargue la página

loginSection.style.display = "block"; // Muestra el formulario de login

});

Cuando el usuario haga clic en el enlace para iniciar sesión:

Se evita el comportamiento predeterminado del enlace (no recarga).

Se muestra el formulario cambiando su estilo CSS para que sea visible (display: block).

4. Revisar si el usuario ya está guardado (sesión previa)

const savedUser = localStorage.getItem("username");

if (savedUser) {

mostrarUsuario(savedUser);

}

localStorage es un almacenamiento en el navegador que guarda datos aunque la página se cierre.

Aquí verificamos si hay un usuario guardado.

Si existe, llamamos a la función mostrarUsuario para mostrar el mensaje de bienvenida sin pedir que inicie sesión de nuevo.

5. Manejo del envío del formulario

loginForm.addEventListener("submit", function (e) {

e.preventDefault(); // Evita que el formulario recargue la página

const username = document.getElementById("username").value;

if (username.trim() !== "") {

localStorage.setItem("username", username); // Guarda el nombre en localStorage

mostrarUsuario(username); // Muestra la sección de usuario logueado

}

});

Cuando el usuario envía el formulario:

Se evita la recarga de la página.

Se obtiene el valor escrito en el campo username.

Si el campo no está vacío (el .trim() elimina espacios), guardamos el usuario en el navegador.

Llamamos a la función para mostrar la bienvenida.

6. Función para mostrar el usuario logueado

function mostrarUsuario(nombre) {

welcomeMessage.innerHTML = `

<strong>¡Bienvenido a Kick Store, ${nombre}!</strong><br>

<span class="mensaje-extra">Nos alegra tenerte de vuelta. Descubre los últimos lanzamientos y encuentra tu próximo par favorito.</span>

`;

userInfo.style.display = "block"; // Mostrar la info del usuario

loginSection.style.display = "none"; // Ocultar formulario login

if (loginMenuLink) loginMenuLink.style.display = "none"; // Ocultar enlace de login en menú

}

Cambia el contenido del mensaje de bienvenida, incluyendo el nombre del usuario.

Muestra la sección con la info del usuario.

Oculta el formulario de login.

Oculta el enlace del menú para login porque el usuario ya está dentro.

7. Cerrar sesión

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

if (logoutButton) {

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

localStorage.removeItem("username"); // Borra el usuario guardado

userInfo.style.display = "none"; // Oculta la info del usuario

if (loginMenuLink) loginMenuLink.style.display = "inline-block"; // Vuelve a mostrar el enlace de login

});

}

Cuando el usuario hace clic en el botón de cerrar sesión:

Se elimina el usuario guardado en localStorage.

Se oculta la sección que muestra la info del usuario.

Se vuelve a mostrar el enlace para iniciar sesión en el menú.