cartella progetto: F:landing-page-libri

progetto Vercel: catalogo-libri-nuovo

immagine sito attuale: Immagine che contiene testo, schermata, design

Il contenuto generato dall'IA potrebbe non essere corretto.

database Firebase: aggiungilibri

CARTELLA PUBLIC

admin.css

/\* Stile generale \*/

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 20px;

text-align: center;

}

/\* Titoli \*/

h1 {

color: #333;

font-size: 24px;

}

h2 {

color: #555;

margin-top: 30px;

font-size: 20px;

}

/\* Contenitore del form \*/

form {

width: 100%;

max-width: 400px;

margin: 20px auto;

padding: 20px;

background: white;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

/\* Etichette e input \*/

label {

display: block;

margin-top: 10px;

font-weight: bold;

font-size: 14px;

text-align: left;

}

input[type="text"],

input[type="url"],

textarea,

input[type="number"],

input[type="file"],

select {

width: 100%;

padding: 6px; /\* Ridotto da 10px per evitare sovrapposizioni \*/

margin-top: 5px;

border: 1px solid #ddd;

border-radius: 5px;

font-size: 13px; /\* Ridotto da 14px \*/

}

/\* Correzione allineamento caselle di spunta \*/

.formati-container {

display: flex;

gap: 10px;

flex-wrap: wrap;

justify-content: center;

margin-top: 10px;

}

.formati-container label {

display: flex;

align-items: center;

gap: 5px;

font-size: 14px;

}

/\* Stile pulsanti \*/

button {

width: 100%;

background-color: #28a745;

color: white;

padding: 12px;

margin-top: 15px;

border: none;

border-radius: 5px;

cursor: pointer;

font-size: 16px;

transition: 0.3s;

}

button:hover {

background-color: #218838;

}

/\* Contenitore libri inseriti \*/

#libri-inseriti {

margin-top: 30px;

display: grid;

grid-template-columns: repeat(auto-fill, minmax(150px, 1fr)); /\* Adatta il numero di colonne \*/

gap: 15px;

justify-content: center;

}

/\* Box singolo per ogni libro \*/

.admin-libro {

width: 180px;

padding: 15px;

background: white;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

text-align: center;

transition: transform 0.3s;

}

.admin-libro h3 {

font-size: 14px; /\* Ridotto da 16px \*/

}

.admin-libro p {

font-size: 12px; /\* Ridotto da 14px \*/

}

.admin-libro:hover {

transform: scale(1.05);

}

.admin-libro img {

width: 100px;

height: 150px;

object-fit: cover;

border-radius: 5px;

}

/\* Pulsante elimina libro \*/

.admin-libro button {

width: auto;

padding: 8px 12px;

background-color: #dc3545;

border: none;

color: white;

font-size: 12px;

cursor: pointer;

border-radius: 4px;

margin-top: 10px;

}

.admin-libro button:hover {

background-color: #c82333;

}

/\* Etichetta per il campo di upload della copertina \*/

label[for="immagine"] {

display: block;

font-weight: bold;

margin-top: 10px;

}

#notifica {

position: fixed;

bottom: 20px;

left: 50%;

transform: translateX(-50%);

background-color: rgba(0, 0, 0, 0.8);

color: white;

padding: 10px 20px;

border-radius: 5px;

font-size: 16px;

display: none; /\* Assicurati che parta nascosta \*/

transition: opacity 0.5s ease-in-out;

z-index: 9999;

}

.notifica-errore {

background-color: #ff4d4d;

}

.notifica-successo {

background-color: #4CAF50;

}

admin.html

<!DOCTYPE html>

<html lang="it">

<head>

<meta charset="UTF-8">

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

<title>Pannello di amministrazione</title>

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

<link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

<body>

<h1>Pannello di amministrazione</h1>

<div id="login-container">

<button id="login-button">Accedi con Google</button>

</div>

<div id="admin-panel" style="display: none;">

<button id="logout-button">Esci</button>

<h2>Inserisci nuovo libro</h2>

<form id="libro-form">

<label for="titolo">Titolo:</label>

<input type="text" id="titolo" name="titolo" required>

<label for="autore">Autore:</label>

<input type="text" id="autore" name="autore" required>

<label for="descrizione">Descrizione:</label>

<textarea id="descrizione" name="descrizione" required></textarea>

<label for="linkAmazon">Link Amazon:</label>

<input type="url" id="linkAmazon" name="linkAmazon" required>

<label for="immagine">Copertina:</label>

<input type="file" id="immagine" name="immagine" accept="image/\*">

<label for="prezzo">Prezzo:</label>

<select id="valuta" name="valuta">

<option value="$">$</option>

<option value="€">€</option>

<option value="£">£</option>

</select>

<input type="number" id="prezzo" name="prezzo" step="0.01" required>

<label for="formati">Formati disponibili:</label><br>

<input type="checkbox" id="ebook" name="formati" value="eBook">

<label for="ebook">eBook</label>

<input type="checkbox" id="paperback" name="formati" value="Paperback">

<label for="paperback">Paperback</label>

<input type="checkbox" id="hardcover" name="formati" value="Hardcover">

<label for="hardcover">Hardcover</label>

<button type="submit">Aggiungi libro</button>

</form>

<h2>Gestione libri</h2>

<div id="gestione-libri">

<div id="filtro-container">

<input type="text" id="filtroTesto" placeholder="Cerca per titolo o autore...">

<select id="ordinamento">

<option value="titolo">Ordina per Titolo</option>

<option value="autore">Ordina per Autore</option>

<option value="prezzo">Ordina per Prezzo</option>

</select>

</div>

<h3>Libri Inseriti (solo lato admin)</h3>

<div id="libri-inseriti"></div>

</div>

<h2>Backup e Ripristino</h2>

<button id="esporta-dati">Esporta JSON</button>

<input type="file" id="importa-dati" accept=".json">

<button id="carica-dati">Importa JSON</button>

</div>

<script src="https://www.gstatic.com/firebasejs/9.6.1/firebase-app-compat.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.6.1/firebase-auth-compat.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.6.1/firebase-database-compat.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.6.1/firebase-storage-compat.js"></script>

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

</body>

</html>

admin.js

// CONFIGURAZIONE FIREBASE

const firebaseConfig = {

apiKey: "AIzaSyAOIp2reVVoeikYjZUk73yQpZNPaDVvCkw",

authDomain: "aggiungilibri.firebaseapp.com",

databaseURL: "https://aggiungilibri-default-rtdb.firebaseio.com",

projectId: "aggiungilibri",

storageBucket: "aggiungilibri.appspot.com",

messagingSenderId: "215130413037",

appId: "1:215130413037:web:058d3395ddef3b7441f9e4"

};

// INIZIALIZZAZIONE FIREBASE

if (!firebase.apps.length) {

firebase.initializeApp(firebaseConfig);

}

// LISTA AMMINISTRATORI AUTORIZZATI

const amministratoriAutorizzati = ["robpacpublishing@gmail.com"];

// ELEMENTI HTML

const loginButton = document.getElementById("login-button");

const logoutButton = document.getElementById("logout-button");

const loginContainer = document.getElementById("login-container");

const adminPanel = document.getElementById("admin-panel");

const libroForm = document.getElementById("libro-form");

const libriInseriti = document.getElementById("libri-inseriti");

const filtroTesto = document.getElementById("filtro-testo");

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

// CACHE LOCALE

let cacheLibri = [];

// NOTIFICHE

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

notifica.id = "notifica";

document.body.appendChild(notifica);

function mostraNotifica(testo, tipo = "successo") {

notifica.textContent = testo;

notifica.className = tipo === "errore" ? "notifica-errore" : "notifica-successo";

notifica.style.display = "block";

notifica.style.opacity = "1";

setTimeout(() => {

notifica.style.opacity = "0";

setTimeout(() => { notifica.style.display = "none"; }, 500);

}, 5000);

}

// CONTROLLO ACCESSO AMMINISTRATORI

firebase.auth().onAuthStateChanged(user => {

if (user && amministratoriAutorizzati.includes(user.email)) {

loginContainer.style.display = "none";

adminPanel.style.display = "block";

caricaLibri();

} else {

loginContainer.style.display = "block";

adminPanel.style.display = "none";

mostraNotifica("⚠️ Accesso negato o utente non autenticato!", "errore");

firebase.auth().signOut();

}

});

// LOGIN CON GOOGLE

loginButton.addEventListener("click", () => {

const provider = new firebase.auth.GoogleAuthProvider();

firebase.auth().signInWithPopup(provider)

.then(result => {

mostraNotifica(`Benvenuto ${result.user.displayName}!`, "successo");

setTimeout(() => { location.reload(); }, 1000);

})

.catch(error => mostraNotifica("Errore di autenticazione: " + error.message, "errore"));

});

// LOGOUT

logoutButton.addEventListener("click", () => {

firebase.auth().signOut().then(() => {

mostraNotifica("Logout effettuato con successo!", "successo");

location.reload();

}).catch(error => mostraNotifica("Errore nel logout", "errore"));

});

// CARICAMENTO LIBRI

function caricaLibri() {

firebase.database().ref('libri').once('value').then(snapshot => {

cacheLibri = snapshot.val() ? Object.entries(snapshot.val()).map(([id, libro]) => ({ id, ...libro })) : [];

mostraLibriInseriti();

}).catch(error => mostraNotifica("Errore nel caricamento dei libri: " + error.message, "errore"));

}

// FUNZIONE UPLOAD CLOUDINARY

async function uploadImmagine(file) {

const formData = new FormData();

formData.append("file", file);

formData.append("upload\_preset", "robpac\_upload");

formData.append("folder", "copertine");

try {

const response = await fetch("https://api.cloudinary.com/v1\_1/robpac/image/upload", { method: "POST", body: formData });

if (!response.ok) throw new Error(`Errore nell'upload: ${response.statusText}`);

const data = await response.json();

return data.secure\_url;

} catch (error) {

mostraNotifica("Errore nel caricamento dell'immagine: " + error.message, "errore");

return null;

}

}

// INSERIMENTO NUOVO LIBRO

libroForm.addEventListener("submit", async function(event) {

event.preventDefault();

const titolo = document.getElementById("titolo").value.trim();

const autore = document.getElementById("autore").value.trim();

const descrizione = document.getElementById("descrizione").value.trim();

const linkAmazon = document.getElementById("linkAmazon").value.trim();

const immagine = document.getElementById("immagine").files[0];

const prezzo = parseFloat(document.getElementById("prezzo").value.replace(",", "."));

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

const formati = Array.from(document.querySelectorAll('input[name="formati"]:checked')).map(checkbox => checkbox.value);

if (!titolo || !autore || !descrizione || !linkAmazon || isNaN(prezzo)) {

return mostraNotifica("⚠️ Tutti i campi devono essere compilati!", "errore");

}

let urlImmagine = "placeholder.jpg";

if (immagine) {

urlImmagine = await uploadImmagine(immagine);

if (!urlImmagine) return;

}

const libro = { titolo, autore, descrizione, linkAmazon, prezzo: prezzo.toFixed(2), valuta, immagine: urlImmagine, formati };

const libriRef = firebase.database().ref('libri').push();

libro.id = libriRef.key;

libriRef.set(libro).then(() => {

cacheLibri.push(libro);

mostraLibriInseriti();

mostraNotifica("📚 Libro aggiunto con successo!", "successo");

libroForm.reset();

}).catch(error => mostraNotifica("Errore durante l'aggiunta del libro: " + error.message, "errore"));

});

// MOSTRA LIBRI INSERITI SOLO NEL LATO ADMIN

function mostraLibriInseriti() {

libriInseriti.innerHTML = "";

// Applica il filtro testo se presente

let libriFiltrati = cacheLibri;

if (filtroTesto.value.trim() !== "") {

const filtro = filtroTesto.value.trim().toLowerCase();

libriFiltrati = libriFiltrati.filter(libro =>

libro.titolo.toLowerCase().includes(filtro) ||

libro.autore.toLowerCase().includes(filtro) ||

libro.descrizione.toLowerCase().includes(filtro)

);

}

// Applica l'ordinamento se selezionato

if (ordinamento.value === "titolo") {

libriFiltrati.sort((a, b) => a.titolo.localeCompare(b.titolo));

} else if (ordinamento.value === "autore") {

libriFiltrati.sort((a, b) => a.autore.localeCompare(b.autore));

} else if (ordinamento.value === "prezzo") {

libriFiltrati.sort((a, b) => parseFloat(a.prezzo) - parseFloat(b.prezzo));

}

// Visualizza i libri filtrati e ordinati

libriFiltrati.forEach(libro => {

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

div.classList.add("admin-libro");

div.dataset.id = libro.id;

div.innerHTML = `

<h3>${libro.titolo}</h3>

<p>Autore: ${libro.autore}</p>

<p>Prezzo: ${libro.valuta} ${libro.prezzo}</p>

<img src="${libro.immagine}" alt="${libro.titolo}" style="width:100px; height:150px;">

<p>Descrizione: ${libro.descrizione}</p>

<p>Formati: ${libro.formati && libro.formati.length > 0 ? libro.formati.join(", ") : "Nessun formato specificato"}</p>

<button onclick="eliminaLibro('${libro.id}')">❌ Elimina</button>

`;

libriInseriti.appendChild(div);

});

}

// ELIMINA LIBRO

function eliminaLibro(libroId) {

firebase.database().ref('libri/' + libroId).remove()

.then(() => {

cacheLibri = cacheLibri.filter(libro => libro.id !== libroId);

mostraLibriInseriti();

mostraNotifica("Libro eliminato con successo!", "successo");

})

.catch(error => mostraNotifica("Errore nell'eliminazione del libro: " + error.message, "errore"));

}

// EVENTI FILTRO E ORDINAMENTO

if (filtroTesto) filtroTesto.addEventListener("input", mostraLibriInseriti);

if (ordinamento) ordinamento.addEventListener("change", mostraLibriInseriti);

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Book Catalog</title>

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

</head>

<body>

<header>

<div id="titolo-container">

<h1>RobPac Publishing</h1>

<h2>A World of Books at Your Fingertips</h2>

</div>

<div id="logo-container">

<img id="logo-catalogo" src="/logo3senzasfondo.png" alt="Catalog Logo">

</div>

<img id="banner-catalogo" src="/banner-landpage2.png" alt="Catalog Banner">

</header>

<main>

<h2>Book Catalog</h2>

<!-- 🔹 Search bar for filtering books -->

<input type="text" id="filtro-testo" placeholder="Search books...">

<div id="libri"></div>

</main>

<footer>

<p>&copy; 2025 Rob Pac - All Rights Reserved</p>

</footer>

<!-- 🔹 Popup Modal for Book Details -->

<div id="book-popup" class="popup-container">

<div class="popup-content">

<span class="close-popup">&times;</span>

<img id="popup-image" src="" alt="Book Cover">

<h3 id="popup-title"></h3>

<p id="popup-author"></p>

<p id="popup-description"></p>

<p id="popup-formats"></p>

<p id="popup-price"></p>

<button id="popup-buy-now" class="buy-now-button">Buy Now</button>

</div>

</div>

<script src="https://www.gstatic.com/firebasejs/9.6.1/firebase-app-compat.js"></script>

<script src="https://www.gstatic.com/firebasejs/9.6.1/firebase-database-compat.js"></script>

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

</body>

</html>

script.js

// Configurazione Firebase

const firebaseConfig = {

apiKey: "AIzaSyAOIp2reVVoeikYjZUk73yQpZNPaDVvCkw",

authDomain: "aggiungilibri.firebaseapp.com",

databaseURL: "https://aggiungilibri-default-rtdb.firebaseio.com",

projectId: "aggiungilibri",

storageBucket: "aggiungilibri.appspot.com",

messagingSenderId: "215130413037",

appId: "1:215130413037:web:058d3395ddef3b7441f9e4"

};

// Inizializzazione Firebase

if (!firebase.apps.length) {

firebase.initializeApp(firebaseConfig);

} else {

firebase.app();

}

// Selettori per gli elementi HTML

const libriDiv = document.getElementById("libri");

const popupContainer = document.getElementById("book-popup");

const popupImage = document.getElementById("popup-image");

const popupTitle = document.getElementById("popup-title");

const popupAuthor = document.getElementById("popup-author");

const popupDescription = document.getElementById("popup-description");

const popupFormats = document.getElementById("popup-formats");

const popupPrice = document.getElementById("popup-price");

const popupBuyNow = document.getElementById("popup-buy-now");

const closePopup = document.querySelector(".close-popup");

// Assicurarsi che gli elementi richiesti esistano

if (!libriDiv) {

console.error("Errore: elemento 'libri' non trovato in index.html.");

}

if (!popupContainer) {

console.error("Errore: elemento 'book-popup' non trovato in index.html.");

}

if (!popupImage) {

console.error("Errore: elemento 'popup-image' non trovato in index.html.");

}

if (!popupTitle) {

console.error("Errore: elemento 'popup-title' non trovato in index.html.");

}

if (!popupAuthor) {

console.error("Errore: elemento 'popup-author' non trovato in index.html.");

}

if (!popupDescription) {

console.error("Errore: elemento 'popup-description' non trovato in index.html.");

}

if (!popupFormats) {

console.error("Errore: elemento 'popup-formats' non trovato in index.html.");

}

if (!popupPrice) {

console.error("Errore: elemento 'popup-price' non trovato in index.html.");

}

if (!popupBuyNow) {

console.error("Errore: elemento 'popup-buy-now' non trovato in index.html.");

}

if (!closePopup) {

console.error("Errore: elemento 'close-popup' non trovato in index.html.");

}

// Funzione per mostrare i libri nel catalogo

function mostraLibri() {

const libriRef = firebase.database().ref("libri");

libriRef.once("value").then(snapshot => {

const libri = snapshot.val();

libriDiv.innerHTML = ""; // Pulisce il contenitore prima di aggiornarlo

if (libri) {

Object.entries(libri).forEach(([id, libro]) => {

// Crea la card del libro

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

libroDiv.classList.add("section-libro");

const immagine = document.createElement("img");

immagine.src = libro.immagine || "placeholder.jpg";

immagine.alt = libro.titolo;

immagine.classList.add("img-libro");

const titolo = document.createElement("h3");

titolo.textContent = libro.titolo || "Titolo non disponibile";

const autore = document.createElement("p");

autore.textContent = libro.autore ? "di " + libro.autore : "Autore sconosciuto";

const prezzo = document.createElement("p");

prezzo.innerHTML = libro.valuta && libro.prezzo ? `<b>da ${libro.valuta}${libro.prezzo}</b>` : "Prezzo non disponibile";

const buyButton = document.createElement("button");

buyButton.classList.add("buy-now-button");

buyButton.textContent = "Acquista ora";

if (libro.linkAmazon) {

buyButton.addEventListener("click", () => {

window.open(libro.linkAmazon, "\_blank");

});

} else {

buyButton.disabled = true;

}

// Aggiunge gli elementi alla card del libro

libroDiv.appendChild(immagine);

libroDiv.appendChild(titolo);

libroDiv.appendChild(autore);

libroDiv.appendChild(prezzo);

libroDiv.appendChild(buyButton);

// Aggiunge evento per aprire il popup

libroDiv.addEventListener("click", () => mostraInfoLibro(id));

// Aggiunge il libro al catalogo

libriDiv.appendChild(libroDiv);

});

} else {

libriDiv.innerHTML = "<p>Nessun libro disponibile.</p>";

}

}).catch(error => {

console.error("Errore nel recupero dei libri:", error);

});

}

// Funzione per mostrare le informazioni del libro nel popup

function mostraInfoLibro(libroId) {

const libriRef = firebase.database().ref(`libri/${libroId}`);

libriRef.once("value").then(snapshot => {

const libro = snapshot.val();

if (!libro) return;

// Aggiorna il contenuto del popup

popupImage.src = libro.immagine || "placeholder.jpg";

popupImage.alt = libro.titolo;

popupTitle.textContent = libro.titolo || "Titolo non disponibile";

popupAuthor.textContent = libro.autore ? "di " + libro.autore : "Autore sconosciuto";

popupDescription.textContent = libro.descrizione || "Nessuna descrizione disponibile.";

// Mostra i formati disponibili

if (libro.formati && Array.isArray(libro.formati) && libro.formati.length > 0) {

popupFormats.textContent = "Formati disponibili: " + libro.formati.join(", ");

} else {

popupFormats.textContent = "Nessun formato disponibile";

}

popupPrice.innerHTML = libro.valuta && libro.prezzo ? `<b>da ${libro.valuta}${libro.prezzo}</b>` : "Prezzo non disponibile";

if (libro.linkAmazon) {

popupBuyNow.style.display = "block";

popupBuyNow.onclick = () => window.open(libro.linkAmazon, "\_blank");

} else {

popupBuyNow.style.display = "none";

}

// Mostra il popup

popupContainer.style.display = "flex";

}).catch(error => {

console.error("Errore nel recupero dei dettagli del libro:", error);

});

}

// Chiude il popup

closePopup.addEventListener("click", () => {

popupContainer.style.display = "none";

});

// Funzione per mostrare il logo

function mostraLogo() {

const logo = localStorage.getItem("logo");

if (logo) {

document.getElementById("logo-catalogo").src = logo;

}

}

// Funzione per mostrare il banner

function mostraBanner() {

const banner = localStorage.getItem("banner");

if (banner) {

document.getElementById("banner-catalogo").src = banner;

}

}

// Inizializza il contenuto

mostraLibri();

mostraLogo();

mostraBanner();

style.css

/\* General Styles \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f5f5f5;

}

/\* HEADER \*/

header {

background-color: #fff;

padding: 20px;

text-align: center;

border-bottom: 2px solid #ddd;

}

#titolo-container {

text-align: center;

}

#logo-container {

position: relative;

text-align: center;

margin-bottom: 10px;

}

#logo-catalogo {

max-width: 150px;

display: block;

margin: 0 auto;

}

#banner-catalogo {

width: 100%;

max-height: 250px;

object-fit: cover;

}

/\* MAIN CONTENT \*/

main {

max-width: 1100px;

margin: 20px auto;

padding: 20px;

background-color: white;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

border-radius: 8px;

}

h2 {

text-align: center;

}

/\* SEARCH BAR \*/

#filtro-testo {

width: 100%;

padding: 10px;

font-size: 16px;

margin-bottom: 20px;

border: 1px solid #ccc;

border-radius: 5px;

}

/\* FORM GESTIONE LIBRI \*/

#admin-panel {

max-width: 900px;

margin: 20px auto;

padding: 20px;

background: #f5f5f5;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

/\* FORM IN GRIGLIA \*/

#libro-form {

display: grid;

grid-template-columns: repeat(2, 1fr);

gap: 15px;

align-items: center;

}

#libro-form label {

font-weight: bold;

}

/\* LABEL CHIARA PER UPLOAD IMMAGINE \*/

#libro-form #label-immagine {

grid-column: span 2;

font-weight: bold;

text-align: center;

margin-top: 10px;

}

/\* TESTO DI SELEZIONE IMMAGINE \*/

#libro-form #immagine {

width: 100%;

display: block;

padding: 5px;

font-size: 14px;

cursor: pointer;

}

/\* MENU A TENDINA VALUTE \*/

#valuta {

width: 100%;

}

/\* MENU VALUTE CON OPZIONI RIORDINATE \*/

#valuta option:first-child {

font-weight: bold;

}

/\* BOTTONE DI INVIO \*/

#libro-form button {

grid-column: span 2;

background: #007bff;

color: white;

border: none;

cursor: pointer;

font-weight: bold;

}

#libro-form button:hover {

background: #0056b3;

}

/\* BOOK GRID \*/

#libri {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));

gap: 20px;

}

/\* BOOK CARD \*/

.section-libro {

background-color: white;

padding: 10px;

border-radius: 8px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

text-align: center;

transition: transform 0.2s ease-in-out;

cursor: pointer;

}

.section-libro:hover {

transform: scale(1.05);

}

.img-libro {

width: 100%;

max-height: 250px;

object-fit: cover;

border-radius: 5px;

}

.section-libro h3 {

font-size: 18px;

margin: 10px 0;

}

.section-libro p {

font-size: 14px;

color: #666;

}

/\* BUY NOW BUTTON \*/

.buy-now-button {

background-color: #ff9900;

color: white;

border: none;

padding: 10px;

font-size: 16px;

cursor: pointer;

border-radius: 5px;

width: 100%;

margin-top: 10px;

}

.buy-now-button:hover {

background-color: #e68900;

}

/\* POPUP MODAL \*/

#book-popup {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.7);

justify-content: center;

align-items: center;

}

.popup-content {

background: white;

width: 80%;

max-width: 600px;

padding: 20px;

border-radius: 8px;

text-align: center;

position: relative;

}

.popup-content img {

width: 100%;

max-height: 300px;

object-fit: cover;

border-radius: 5px;

}

.close-popup {

position: absolute;

top: 10px;

right: 15px;

font-size: 24px;

cursor: pointer;

}

/\* ADMIN BOOK LIST \*/

#libri-inseriti {

margin-top: 20px;

display: flex;

flex-wrap: wrap;

gap: 15px;

}

.admin-libro {

background: white;

padding: 10px;

border: 1px solid #ddd;

border-radius: 5px;

text-align: center;

box-shadow: 0px 2px 5px rgba(0, 0, 0, 0.1);

width: 200px;

}

.admin-libro img {

width: 100%;

height: auto;

border-radius: 5px;

object-fit: cover;

}

/\* DELETE BUTTON ADMIN \*/

.admin-libro button {

background: #ff4d4d;

color: white;

padding: 5px 10px;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 12px;

margin-top: 5px;

}

.admin-libro button:hover {

background: #cc0000;

}

/\* RIDUZIONE IMMAGINE GRANDE IN BASSO \*/

#libri img {

max-width: 120px;

height: auto;

object-fit: cover;

}

/\* FOOTER \*/

footer {

text-align: center;

padding: 10px;

background: #222;

color: white;

}

/\* RESPONSIVE DESIGN \*/

@media (max-width: 768px) {

#logo-catalogo {

max-height: 100px;

}

#banner-catalogo {

max-height: 200px;

}

#libri {

grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));

gap: 10px;

}

.libro {

max-width: 180px;

}

.libro img {

max-width: 100%;

}

.libro h3 {

font-size: 0.9rem;

}

.section-libro img {

width: 120px;

height: 180px;

}

.info-libro {

flex-direction: column;

width: 100%;

}

.info-libro img {

width: 100%;

margin-right: 0;

margin-bottom: 20px;

}

#titolo-container h1 {

font-size: 20px;

}

#titolo-container h2 {

font-size: 16px;

}

#admin-panel {

width: 95%;

padding: 10px;

}

#libro-form {

grid-template-columns: 1fr;

}

#libro-form textarea,

#libro-form button {

grid-column: span 1;

}

}

cors.json

[

{

"origin": ["http://localhost:8000"],

"method": ["GET", "POST", "PUT", "DELETE", "OPTIONS"],

"maxAgeSeconds": 3600

}

]