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">

</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>

<!-- 🔔 Messaggio feedback duplicati o errori -->

<div id="messaggio-feedback" style="color: red; font-weight: bold; margin-bottom: 10px;"></div>

<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"

};

if (!firebase.apps.length) {

firebase.initializeApp(firebaseConfig);

}

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

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("filtroTesto");

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

const messaggioFeedback = document.getElementById("messaggio-feedback");

// DISABILITA TEMPORANEAMENTE FORMATI DISPONIBILI

["ebook", "paperback", "hardcover"].forEach(id => {

document.getElementById(id).disabled = true;

document.querySelector(`label[for="${id}"]`).style.display = "none";

});

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);

}

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";

firebase.auth().signOut();

}

});

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

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

firebase.auth().signInWithPopup(provider)

.then(result => {

alert(`Benvenuto ${result.user.displayName}!`);

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

})

.catch(error => alert("Errore di autenticazione: " + error.message));

});

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

firebase.auth().signOut()

.then(() => {

alert("Logout effettuato con successo!");

location.reload();

})

.catch(error => alert("Errore nel logout"));

});

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 => alert("Errore nel caricamento dei libri: " + error.message));

}

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) {

mostraNotifica("Errore nel caricamento dell'immagine. Riprova.", "errore");

return null;

}

const data = await response.json();

return data.secure\_url;

} catch (error) {

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

return null;

}

} // <--- Questa parentesi chiude correttamente la funzione

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 prezzo = parseFloat(document.getElementById("prezzo")?.value.replace(",", ".")) || 0;

const valuta = document.getElementById("valuta")?.value || "$";

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

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

alert("⚠️ Tutti i campi devono essere compilati!");

return;

}

// 🔁 CONTROLLO DOPPIONE PER LINK AMAZON

const snapshot = await firebase.database().ref("libri")

.orderByChild("linkAmazon")

.equalTo(linkAmazon)

.once("value");

if (snapshot.exists()) {

mostraNotifica("🚫 Questo libro è già stato inserito (link Amazon duplicato)", "errore");

return;

}

let urlImmagine = "placeholder.jpg";

if (immagine) {

urlImmagine = await uploadImmagine(immagine);

if (!urlImmagine) {

mostraNotifica("Caricamento immagine fallito. Il libro non è stato salvato.", "errore");

return;

}

const libro = {

titolo,

autore,

descrizione,

linkAmazon,

prezzo: prezzo.toFixed(2),

valuta,

immagine: urlImmagine

};

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

libro.id = libriRef.key;

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

// Controlla se il libro esiste già nella cache

const index = cacheLibri.findIndex(l => l.linkAmazon === linkAmazon);

if (index !== -1) {

cacheLibri[index] = libro; // Se il libro esiste già, aggiorna i dati

} else {

cacheLibri.push(libro); // Altrimenti, aggiungi un nuovo libro

}

// Aggiorna la visualizzazione

mostraLibriInseriti();

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

libroForm.reset();

}).catch(error => { // Qui correggiamo la parentesi

mostraNotifica("❌ Errore durante l'aggiunta del libro: " + error.message, "errore");

});

// ELIMINA LIBRO

function eliminaLibro(libroId) {

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

confermaDiv.classList.add("notifica-conferma");

confermaDiv.innerHTML = `

<p>❗ Sei sicuro di voler eliminare questo libro?</p>

<button id="conferma-elimina">✅ Conferma</button>

<button id="annulla-elimina">❌ Annulla</button>

`;

document.body.appendChild(confermaDiv);

document.getElementById("conferma-elimina").addEventListener("click", () => {

firebase.database().ref(`libri/${libroId}`).remove()

.then(() => {

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

mostraLibriInseriti();

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

})

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

confermaDiv.remove(); // Rimuove la finestra di conferma

});

document.getElementById("annulla-elimina").addEventListener("click", () => {

mostraNotifica("❌ Eliminazione annullata.");

confermaDiv.remove(); // Rimuove la finestra se l'utente annulla

});

}

function mostraLibriInseriti() {

libriInseriti.innerHTML = "";

let libriFiltrati = [...cacheLibri];

const testoFiltro = filtroTesto.value.toLowerCase();

if (testoFiltro) {

libriFiltrati = libriFiltrati.filter(libro =>

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

libro.autore.toLowerCase().includes(testoFiltro)

);

}

const criterio = ordinamento.value;

libriFiltrati.sort((a, b) => {

if (criterio === "prezzo") {

const prezzoA = isNaN(parseFloat(a.prezzo)) ? 0 : parseFloat(a.prezzo);

const prezzoB = isNaN(parseFloat(b.prezzo)) ? 0 : parseFloat(b.prezzo);

return prezzoA - prezzoB; // Ordina dal più basso al più alto

}

return a[criterio].localeCompare(b[criterio]);

});

libriFiltrati.forEach(libro => {

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

div.className = "admin-libro";

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;">

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

`;

libriInseriti.appendChild(div);

});

}

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="banner-container">

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

<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>

</div>

</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">

<div class="popup-text">

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

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

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

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

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

</div>

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

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

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

// Traduzione in inglese per il catalogo libri

const TEXTS = {

buyNow: "Buy Now",

author: "by",

priceFrom: "from "

};

// 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 = "";

if (libri) {

Object.entries(libri).forEach(([id, 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 || "Title not available";

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

autore.textContent = libro.autore ? `${TEXTS.author} ${libro.autore}` : "Unknown author";

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

prezzo.innerHTML = libro.valuta && libro.prezzo ? `<b>${TEXTS.priceFrom}${libro.valuta}${libro.prezzo}</b>` : "Price not available";

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

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

buyButton.textContent = TEXTS.buyNow;

if (libro.linkAmazon) {

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

event.stopPropagation();

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

});

} else {

buyButton.disabled = true;

}

libroDiv.appendChild(immagine);

libroDiv.appendChild(titolo);

libroDiv.appendChild(autore);

libroDiv.appendChild(prezzo);

libroDiv.appendChild(buyButton);

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

libriDiv.appendChild(libroDiv);

});

} else {

libriDiv.innerHTML = "<p>No books available.</p>";

}

}).catch(error => {

console.error("Error retrieving books:", 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;

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

popupImage.alt = libro.titolo;

popupTitle.textContent = libro.titolo || "Title not available";

popupAuthor.textContent = libro.autore ? `${TEXTS.author} ${libro.autore}` : "Unknown author";

popupDescription.textContent = libro.descrizione && libro.descrizione.trim() !== "" ? libro.descrizione : "No description available.";

if (libro.valuta && libro.prezzo) {

popupPrice.innerHTML = `<b>${TEXTS.priceFrom}${libro.valuta}${libro.prezzo}</b>`;

} else {

popupPrice.innerHTML = "";

}

if (libro.linkAmazon) {

popupBuyNow.style.display = "block";

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

} else {

popupBuyNow.style.display = "none";

}

popupContainer.style.display = "flex";

}).catch(error => {

console.error("Error retrieving book details:", error);

});

}

// Chiude il popup

function closePopup() {

popupContainer.style.display = "none";

}

if (closePopupButton) {

closePopupButton.addEventListener("click", closePopup);

}

// Funzione per mostrare il logo

function mostraLogo() {

const logoContainer = document.getElementById("logo-container");

if (logoContainer) {

logoContainer.style.display = "block";

} else {

console.error("Errore: elemento 'logo-container' non trovato.");

}

}

// Funzione per mostrare il banner

function mostraBanner() {

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

if (banner) {

banner.style.display = "block";

} else {

console.error("Errore: elemento 'banner-catalogo' non trovato.");

}

}

// Funzione di controllo duplicati (basato su link Amazon)

async function libroEsiste(linkAmazon) {

const snapshot = await firebase.database().ref("libri").orderByChild("linkAmazon").equalTo(linkAmazon).once("value");

return snapshot.exists();

}

// Sovrascrittura evento form per bloccare i duplicati

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

if (form) {

form.addEventListener("submit", async (e) => {

e.preventDefault();

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

if (!linkAmazon) {

alert("⚠️ Inserisci il link Amazon.");

return;

}

const esiste = await libroEsiste(linkAmazon);

if (esiste) {

alert("❌ Libro già presente nel catalogo (link Amazon duplicato).");

return;

}

// Procedi con la logica originale del submit (eventuale altro script)

form.submit();

});

}

// Avvia il caricamento dei libri e logo/banner

mostraLibri();

mostraLogo();

mostraBanner();

style.css

/\* General Styles \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f5f5f5;

}

/\* HEADER \*/

header {

position: relative;

background-color: #fff;

padding: 0;

text-align: center;

border-bottom: 2px solid #ddd;

}

#logo-container {

position: absolute;

top: 50%;

left: 5%;

transform: translateY(-50%);

z-index: 10;

}

#logo-catalogo {

max-width: 140px;

display: block;

}

@media (max-width: 768px) {

#logo-catalogo {

max-width: 100px;

}

}

#banner-catalogo {

width: 100%;

max-height: 250px;

object-fit: cover;

position: relative;

}

#titolo-container {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

color: white;

text-align: center;

z-index: 10;

font-size: 24px;

font-weight: bold;

text-shadow: 2px 2px 8px rgba(0, 0, 0, 0.7);

}

/\* 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;

}

/\* BOOK GRID \*/

#libri {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(180px, 1fr));

gap: 15px;

text-align: center;

}

/\* BOOK CARD \*/

.section-libro {

background-color: white;

padding: 8px;

border-radius: 8px;

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

transition: transform 0.2s ease-in-out;

cursor: pointer;

display: flex;

flex-direction: column;

justify-content: space-between;

height: 270px;

}

.section-libro:hover {

transform: scale(1.05);

}

.img-libro {

width: 100%;

max-height: 140px;

object-fit: cover;

border-radius: 5px;

}

.section-libro .libro-top {

display: flex;

flex-direction: column;

align-items: center;

}

.section-libro .libro-bottom {

display: flex;

flex-direction: column;

align-items: center;

justify-content: flex-end;

}

.section-libro h3 {

font-size: 14px;

margin-top: 5px;

}

.section-libro p {

font-size: 12px;

color: #666;

margin: 2px 0;

}

/\* BUY NOW BUTTON \*/

.buy-now-button {

background-color: #ff9900;

color: white;

border: none;

padding: 8px;

font-size: 12px;

cursor: pointer;

border-radius: 5px;

width: 90%;

margin-top: 5px;

}

.buy-now-button:hover {

background-color: #e68900;

}

/\* POPUP MODAL - Allargato e spostato più in basso \*/

#book-popup {

display: none;

position: fixed;

top: 55vh;

left: 50%;

transform: translate(-50%, -50%);

width: 90%;

max-width: 900px;

background: white;

padding: 20px;

border-radius: 8px;

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

z-index: 999;

}

.popup-content {

display: flex;

flex-direction: row;

align-items: flex-start;

gap: 20px;

max-height: 85vh;

overflow-y: auto;

}

.popup-content img {

width: 35%;

max-height: 320px;

object-fit: cover;

border-radius: 5px;

}

.popup-text {

width: 60%;

padding: 10px;

text-align: left;

overflow-wrap: break-word;

}

/\* CHIUSURA POPUP \*/

.close-popup {

position: absolute;

top: 10px;

right: 15px;

font-size: 24px;

cursor: pointer;

color: white;

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

padding: 5px 10px;

border-radius: 5px;

z-index: 1000;

}

.close-popup:hover {

background-color: rgba(255, 255, 255, 0.7);

color: black;

}

/\* Responsive per mobile \*/

@media (max-width: 768px) {

#book-popup {

width: 95%;

top: 5vh;

}

.popup-content {

flex-direction: column;

align-items: center;

}

.popup-content img {

width: 100%;

max-height: 200px;

}

.popup-text {

width: 100%;

text-align: center;

}

}

/\* 🔹 Stile per la finestra di conferma eliminazione \*/

.notifica-conferma {

position: fixed;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

background: white;

padding: 20px;

border-radius: 8px;

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

z-index: 1000;

text-align: center;

width: 300px;

}

.notifica-conferma p {

font-size: 16px;

margin-bottom: 15px;

}

.notifica-conferma button {

margin: 5px;

padding: 8px 12px;

border: none;

cursor: pointer;

font-size: 14px;

border-radius: 5px;

}

#conferma-elimina {

background: #ff4d4d;

color: white;

}

#annulla-elimina {

background: #ccc;

color: black;

}

#conferma-elimina:hover {

background: #cc0000;

}

#annulla-elimina:hover {

background: #999;

}

cors.json

[

{

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

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

"maxAgeSeconds": 3600

}

]