

Présentation du projet

Application web de gestion d'une bibliothèque numérique avec images de couverture intégrées. Elle permet de consulter le catalogue, d'emprunter et de rendre des livres, d'ajouter de nouveaux titres et de suivre l'historique des transactions. Entièrement en JavaScript, sans back-end.

Fonctionnalités

- Catalogue avec couvertures personnalisées (5 livres préconfigurés)
- Emprunt : saisie utilisateur, limite à 3 livres par personne
- Retour : remise en disponibilité du livre sélectionné
- Historique : 20 dernières transactions (emprunts + retours)
- Ajout de livre : titre, auteur, année, URL de couverture (facultatif)
- Fond de page : photo de bibliothèque (biblio.jpg)

Architecture

Fichier	Rôle
index.html	Structure HTML5 — nav, sections (liste/emprunter/rendre/ajouter/historique)
style.css	Design violet/dégradé, fond biblio.jpg, cartes livres, formulaires
script.js	Catalogue, emprunts, retours, historique, ajout dynamique, gestion DOM

FICHER 1 — index.html

■ index.html — Structure HTML5

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Gestion de bibliothèque</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<h1>Gestion de bibliothèque</h1>

<nav>
<button onclick="showSection('liste')">Voir les livres</button>
<button onclick="showSection('emprunter')">Emprunter</button>
<button onclick="showSection('rendre')">Rendre</button>
<button onclick="showSection('ajouter')">Ajouter un livre</button>
<button onclick="showSection('historique')">Historique</button>
</nav>

<section id="liste" class="section active">
<h2>Livres disponibles</h2>
<div id="booksList"></div>
</section>

<section id="emprunter" class="section">
<h2>Emprunter un livre</h2>
```

```

<div class="form-group">
  <label for="borrowUser"><strong>Nom de l'utilisateur :</strong></label>
  <input type="text" id="borrowUser">
  <label for="borrowSelect"><strong>Livre :</strong></label>
  <select id="borrowSelect"></select>
  <button onclick="borrowBook()"><strong>Emprunter</strong></button>
  <p id="borrowMessage" class="message"></p>
</div>
</section>

<section id="rendre" class="section">
  <h2>Rendre un livre</h2>
  <div class="form-group">
    <label for="returnSelect"><strong>livre emprunté :</strong></label>
    <select id="returnSelect"></select>
    <button onclick="returnBook()">Rendre</button>
    <p id="returnMessage" class="message"></p>
  </div>
</section>

<section id="ajouter" class="section">
  <h2>Ajouter un livre</h2>
  <div class="form-group">
    <input type="text" id="newTitle" placeholder="Titre">
    <input type="text" id="newAuthor" placeholder="Auteur">
    <input type="number" id="newYear" placeholder="Année">
    <input type="text" id="newCover" placeholder="URL de l'image (facultatif)">
    <button onclick="addBook()"><strong>Ajouter</strong></button>
    <p id="addMessage" class="message"></p>
  </div>
</section>

<section id="historique" class="section">
  <h2>Historique des emprunts</h2>
  <div id="historyList"></div>
</section>

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

```

FICHER 2 — style.css

■ style.css — Feuille de style

```

/* Base & Background */
body {
  font-family: Arial, sans-serif;
  margin: 20px;
  background: url("img/biblio.jpg") center/cover no-repeat fixed;
}

/* Styles Communs (Titres et Sections) */
h1, h2, .section, nav {
  border-radius: 15px;
  padding: 20px;
}

h1, h2 {
  display: table; /* Remplace inline-block pour centrage auto avec margin */
  margin: 0 auto 20px auto;
  text-align: center;
  background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
}

```

```

h1 { border: 2px solid black; }
h2 { border: 2px solid white; font-size: 1.5rem; }

/* Navigation */
nav {
  background: transparent;
  text-align: center;
  margin-bottom: 20px;
}

nav button {
  background: linear-gradient(45deg, #667eea, #764ba2);
  color: white;
  border: none;
  padding: 12px 20px;
  margin: 5px;
  border-radius: 25px;
  cursor: pointer;
  font-weight: bold;
  transition: transform 0.2s;
}

nav button:hover { transform: scale(1.05); }

/* Sections */
.section { display: none; background-color: white; margin-top: 20px; }
.section.active { display: block; }

#liste { background: transparent; }
#emprunter, #rendre, #ajouter, #historique {
  background: rgba(150, 106, 167, 0.7);
}

/* Objets (Livres & Formulaires) */
#booksList { display: flex; flex-wrap: wrap; gap: 15px; justify-content: center; }

.book {
  border-radius: 10px;
  padding: 10px;
  width: 180px;
  background: white;
  font-size: 14px;
}

.book img { width: 100%; height: 200px; object-fit: cover; border-radius: 5px; }

.form-group { display: flex; flex-direction: column; max-width: 400px; margin: 0 auto; gap: 10px; }

.message, .emprunte {
  color: darkred;
  background: white;
  padding: 10px;
  display: inline-block;
  border-radius: 5px;
}

```

FICHER 3 — script.js

■■■ script.js — Logique JavaScript

```

let books = [
  { id: 1, title: "Le Petit Prince", author: "A. de Saint-Exupéry", year: 1943, available: true, cover:
    "img/prince.png"},
  { id: 2, title: "1984", author: "George Orwell", year: 1949, available: true, cover: "img/1984.jpg"},
  { id: 3, title: "Le Seigneur des Anneaux", author: "J.R.R. Tolkien", year: 1954, available: true, cover:
    "img/ring.jpg"},

```

```

{ id: 4, title: "Le Monde de Narnia", author: "C. S. Lewis", year: 1949, available: true, cover:
"img/lion.jpg"},
{ id: 5, title: "La Tour Sombre", author: "Stephen King", year: 1970, available: true, cover:
"img/tour.jpg"}
];

let history = [];
const MAX_BORROWS = 3;

function showSection(id) {
document.querySelectorAll(".section").forEach(s => s.classList.remove("active"));
document.getElementById(id).classList.add("active");

const actions = {
'liste': displayBooks,
'emprunter': updateBorrowSelect,
'rendre': updateReturnSelect,
'historique': displayHistory
};
if (actions[id]) actions[id]();
}

function displayBooks() {
const container = document.getElementById("booksList");
container.innerHTML = books.map(book => `
<div class="book">

<div class="titre"><strong>${book.title}</strong></div>
<div>${book.author} (${book.year})</div>
<div>${book.available ? "Disponible" : "<span class='emprunte'>Emprunte</span>"}</div>
</div>
`).join('');
}

function updateSelect(selectId, filterFn) {
const select = document.getElementById(selectId);
select.innerHTML = books.filter(filterFn).map(b =>
`<option value="${b.id}">${b.title} - ${b.author}</option>`
).join('');
}

const updateBorrowSelect = () => updateSelect("borrowSelect", b => b.available);
const updateReturnSelect = () => updateSelect("returnSelect", b => !b.available);

function countUserBorrows(user) {
const userEvents = history.filter(h => h.user === user);
return userEvents.filter(h => h.action === "emprunt").length -
userEvents.filter(h => h.action === "retour").length;
}

function borrowBook() {
const user = document.getElementById("borrowUser").value.trim();
const id = parseInt(document.getElementById("borrowSelect").value);
const msg = document.getElementById("borrowMessage");

if (!user || !id) return msg.textContent = "Veuillez remplir tous les champs.";
if (countUserBorrows(user) >= MAX_BORROWS) return msg.textContent = `Limite de ${MAX_BORROWS} livres
atteinte.`;

const book = books.find(b => b.id === id);
book.available = false;
history.push({ user, bookId: id, title: book.title, action: "emprunt", date: new Date().toISOString() });

msg.textContent = "Livre emprunte avec succes.";
updateBorrowSelect();
}

function returnBook() {

```

```

const id = parseInt(document.getElementById("returnSelect").value);
if (!id) return;

const book = books.find(b => b.id === id);
book.available = true;
history.push({ user: "Systeme", bookId: id, title: book.title, action: "retour", date: new
Date().toISOString() });

document.getElementById("returnMessage").textContent = "Livre rendu.";
updateReturnSelect();
}

function addBook() {
const title = document.getElementById("newTitle").value.trim();
const author = document.getElementById("newAuthor").value.trim();
if (!title || !author) return alert("Titre et auteur requis");

books.push({
id: Date.now(),
title, author,
year: document.getElementById("newYear").value,
cover: document.getElementById("newCover").value,
available: true
});

["newTitle", "newAuthor", "newYear", "newCover"].forEach(id => document.getElementById(id).value = "");
displayBooks();
}

function displayHistory() {
const container = document.getElementById("historyList");
container.innerHTML = history.length ? history.slice(-20).reverse().map(record => `
<div style="margin-bottom:10px; padding:8px; border-left:4px solid white;
background:rgba(255,255,255,0.1)">
<strong>${record.title}</strong><br>
<small>${record.user} - ${new Date(record.date).toLocaleString()} - Action: ${record.action}</small>
</div>
`).join('') : "<p><strong>Aucun historique.</strong></p>";
}

window.onload = displayBooks;

```