

**ÉCOLE MAROCAINE DES SCIENCES DE L'INGÉNIEUR**

3ème année IIR

Annee universitaire 2025-2026

# **Systeme de Gestion Bibliothèque :**

**Rapport présenté par :**

**ilyasse elkaui - reda harchid - amine rahmouni**

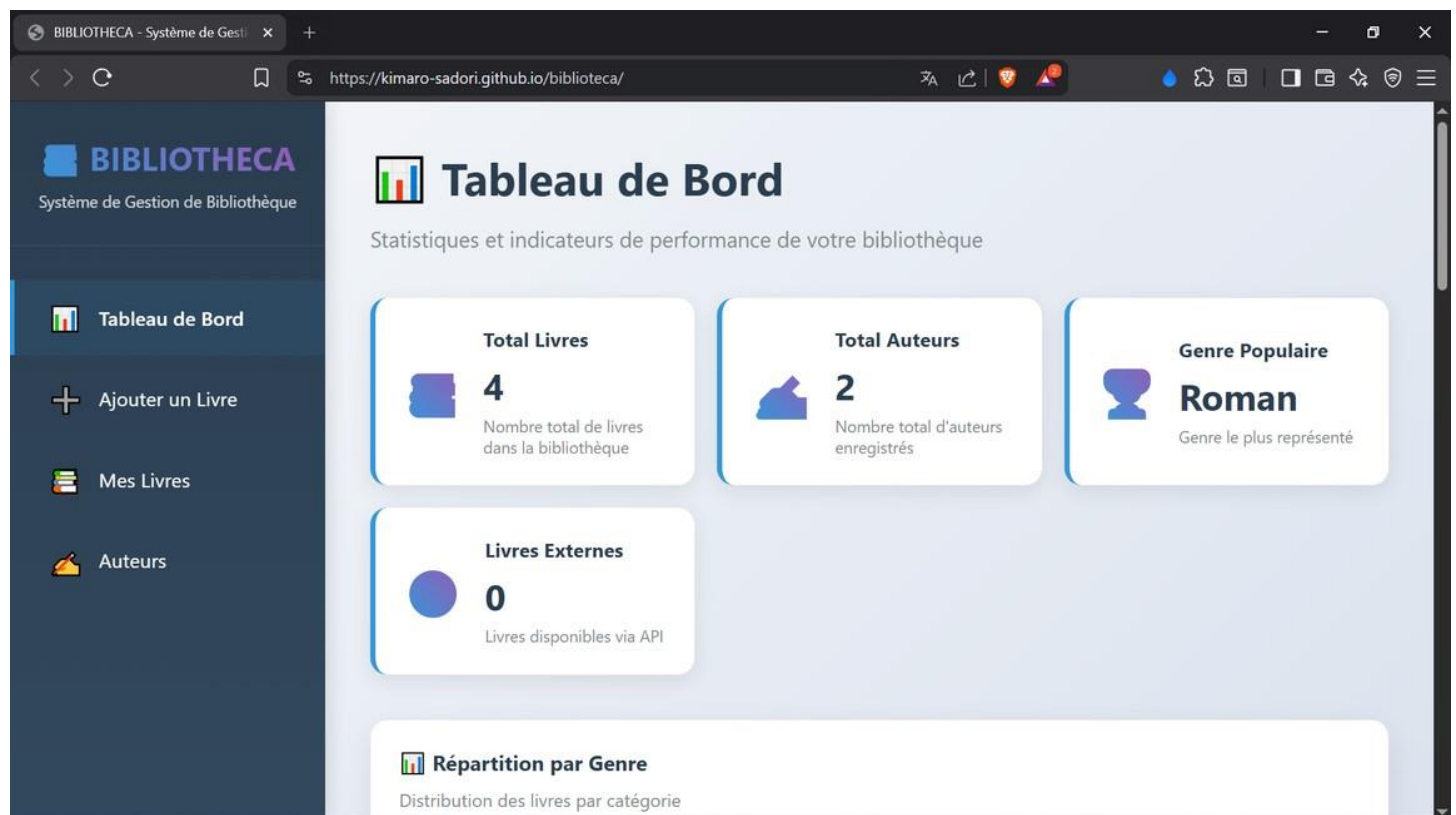
# Sommaire :

1. Introduction générale
2. Objectifs du projet
3. Présentation générale de l'application
4. Technologies utilisées
5. Architecture générale du système
6. Structure de l'interface utilisateur
7. Tableau de bord et indicateurs de performance
8. Gestion des livres
9. Stockage des données
10. Recherche et tri des livres
11. Gestion des auteurs
12. Statistiques et visualisation des données
13. Intégration d'une API externe
14. Ajout de livres depuis l'API
15. Initialisation et fonctionnement global
16. Limites et contraintes du projet
17. Conclusion générale

## 1. Introduction générale

Le projet **BIBLIOTHECA** est une application web conçue pour gérer une bibliothèque de manière simple, moderne et efficace. Elle permet à un utilisateur de gérer une collection de livres, d'enregistrer des auteurs, de visualiser des statistiques, et d'enrichir sa bibliothèque grâce à une API externe.

Ce projet a été réalisé dans le cadre d'un examen, avec pour objectif principal de démontrer la maîtrise des technologies web fondamentales : HTML, CSS, JavaScript, ainsi que l'utilisation d'une API REST.



## 2. Objectifs du projet

Les objectifs principaux de BIBLIOTHECA sont :

- Créer une **application web dynamique** sans rechargement de page
- Implémenter un **CRUD complet** (Créer, Lire, Modifier, Supprimer)
- Gérer des données localement avec **LocalStorage**
- Afficher des **statistiques visuelles** à l'aide de graphiques
- Intégrer une **API externe** pour récupérer des livres

**Offrir une interface claire et responsive**

## 3. Technologies utilisées

### 3.1 HTML5

Utilisé pour structurer le contenu de l'application.

### 3.2 CSS3

Utilisé pour la mise en forme, la disposition (layout), et l'ergonomie.

### 3.3 JavaScript

Cœur logique du projet :

- gestion des événements
- manipulation du DOM
- stockage local
- communication avec une API

### 3.4 Chart.js

Bibliothèque JavaScript utilisée pour créer des graphiques interactifs.

### 3.5 OpenLibrary API

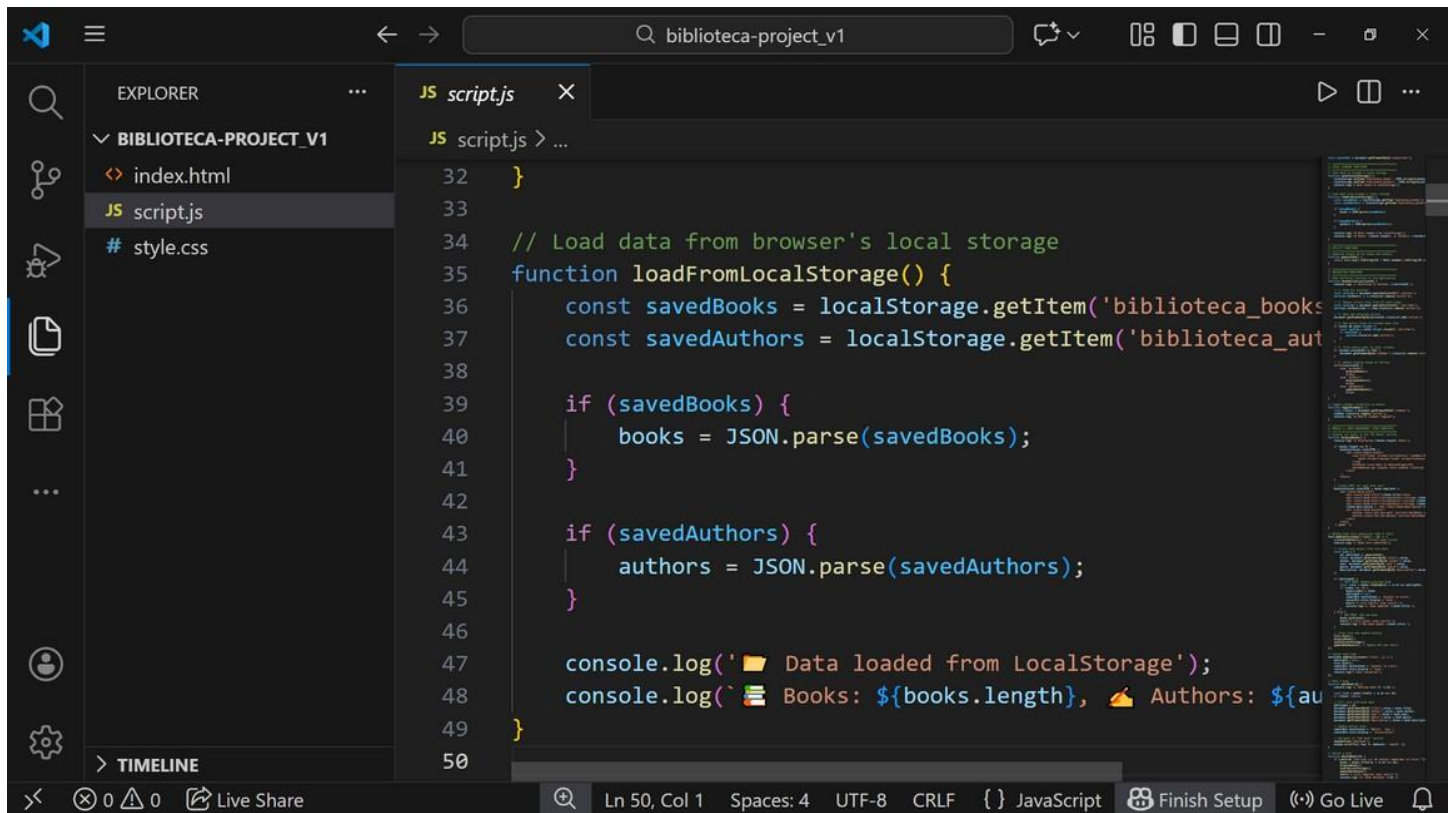
API publique permettant de rechercher des livres en ligne.

## 4. Architecture générale du projet

Le projet est composé de **trois fichiers principaux** :

- index.html → structure de l'interface
- style.css → apparence visuelle script.js →
- logique et fonctionnalités

Cette séparation respecte le principe de **séparation des responsabilités**.



The screenshot shows the Visual Studio Code interface with the 'BIBLIOTECA-PROJECT\_V1' project open. The Explorer sidebar on the left lists the files: index.html, script.js, and style.css. The main editor window displays the content of 'script.js', which includes a function 'loadFromLocalStorage()' that retrieves data from browser's local storage and logs it to the console.

```
32 }
33
34 // Load data from browser's local storage
35 function loadFromLocalStorage() {
36     const savedBooks = localStorage.getItem('biblioteca_books');
37     const savedAuthors = localStorage.getItem('biblioteca_authors');
38
39     if (savedBooks) {
40         books = JSON.parse(savedBooks);
41     }
42
43     if (savedAuthors) {
44         authors = JSON.parse(savedAuthors);
45     }
46
47     console.log('📁 Data loaded from LocalStorage');
48     console.log('📖 Books: ${books.length}, 📑 Authors: ${authors.length}');
49 }
50
```

## 5. Structure de l'interface (HTML)

Le fichier index.html définit une **Single Page Application (SPA)**.

### 5.1 Barre latérale (Sidebar)

La barre latérale contient :

- Tableau de bord
- Ajouter un livre
- Mes livres
- Auteurs

Elle permet la navigation **sans recharger la page**.

## 5.2 Sections principales

Chaque partie de l'application est une <section> :

- dashboard add-
- book my-books
- authors
- 

Une seule section est visible à la fois grâce à JavaScript.

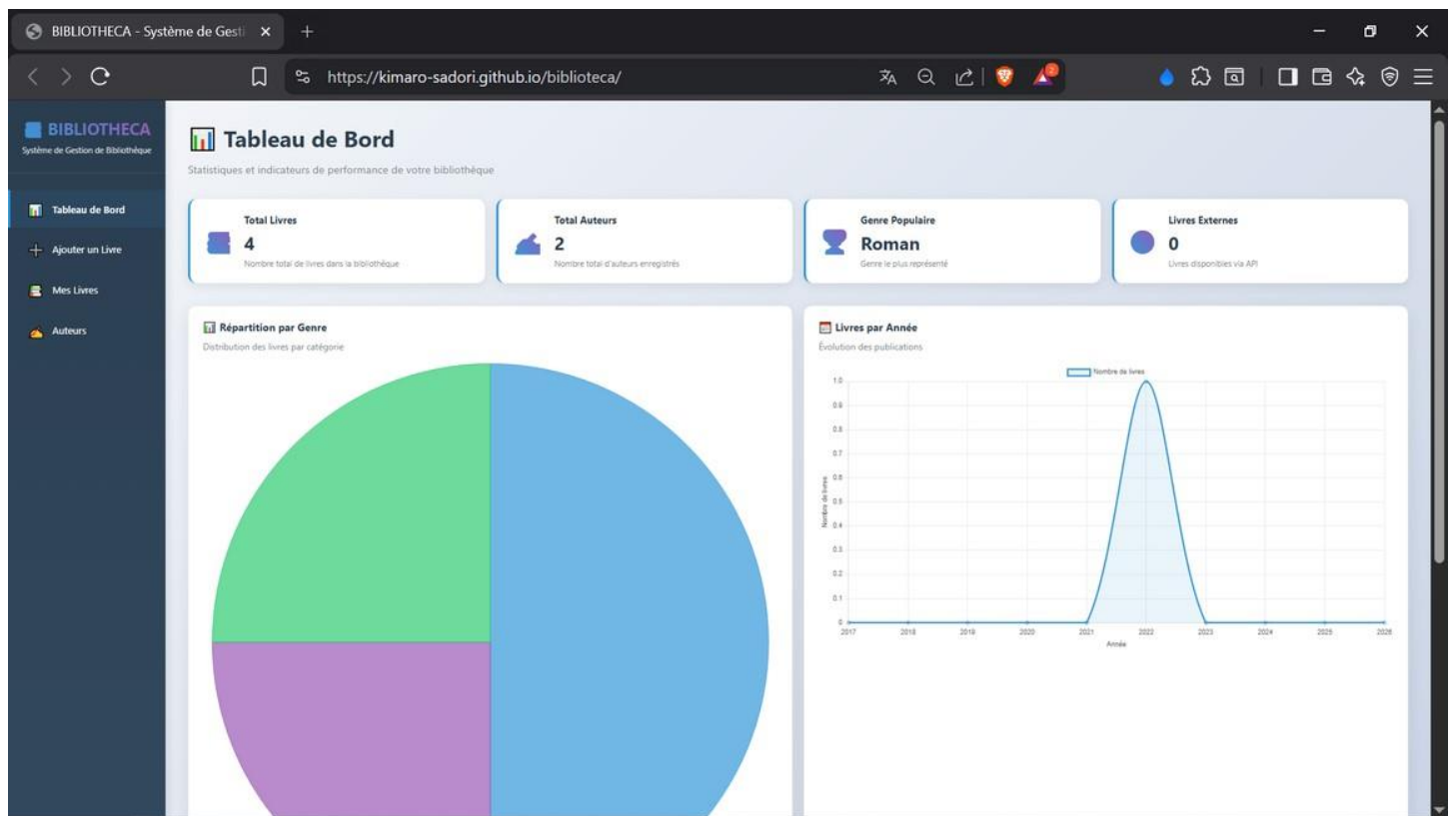
## 6. *Tableau de bord (Dashboard)*

Le tableau de bord affiche des **indicateurs clés (KPI)** :

- Nombre total de livres
- Nombre total d'auteurs Genre le
- plus populaire Nombre de livres
- trouvés via l'API

Il contient également deux graphiques :

- Répartition des livres par genre
- Évolution des livres par année



## 7. Gestion des livres (Module 1)

### 7.1 Ajout d'un livre

Un formulaire permet de saisir :

- Titre
- Auteur
- Année
- Genre
- Description

Lors de la soumission :

- les données sont validées un objet
- JavaScript est créé le livre est
- ajouté au tableau books

## 7.2 Modification d'un livre

Lorsqu'on clique sur **Modifier** : les données du livre sont

- chargées dans le formulaire l'utilisateur peut les
- mettre à jour

## 7.3 Suppression d'un livre

Un message de confirmation empêche toute suppression accidentelle.

# 8. *Stockage local (LocalStorage)*

## 8.1 Pourquoi LocalStorage ?

Le LocalStorage permet de :

- conserver les données même après fermeture du navigateur
- éviter l'utilisation d'une base de données serveur

## 8.2 Fonctionnement

Les données sont sauvegardées sous forme de **JSON** :

- biblioteca\_books
- biblioteca\_authors

Lors du chargement de la page, les données sont relues automatiquement.

# 9. *Recherche et tri des livres*

## 9.1 Recherche

La recherche fonctionne en temps réel :

- par titre
- par auteur

Elle utilise filter() et includes().



## 9.2 Tri

Le tri permet d'organiser les livres par :

- titre
- auteur
- année

Cela améliore l'expérience utilisateur.

## 10. Gestion des auteurs (Module 2)

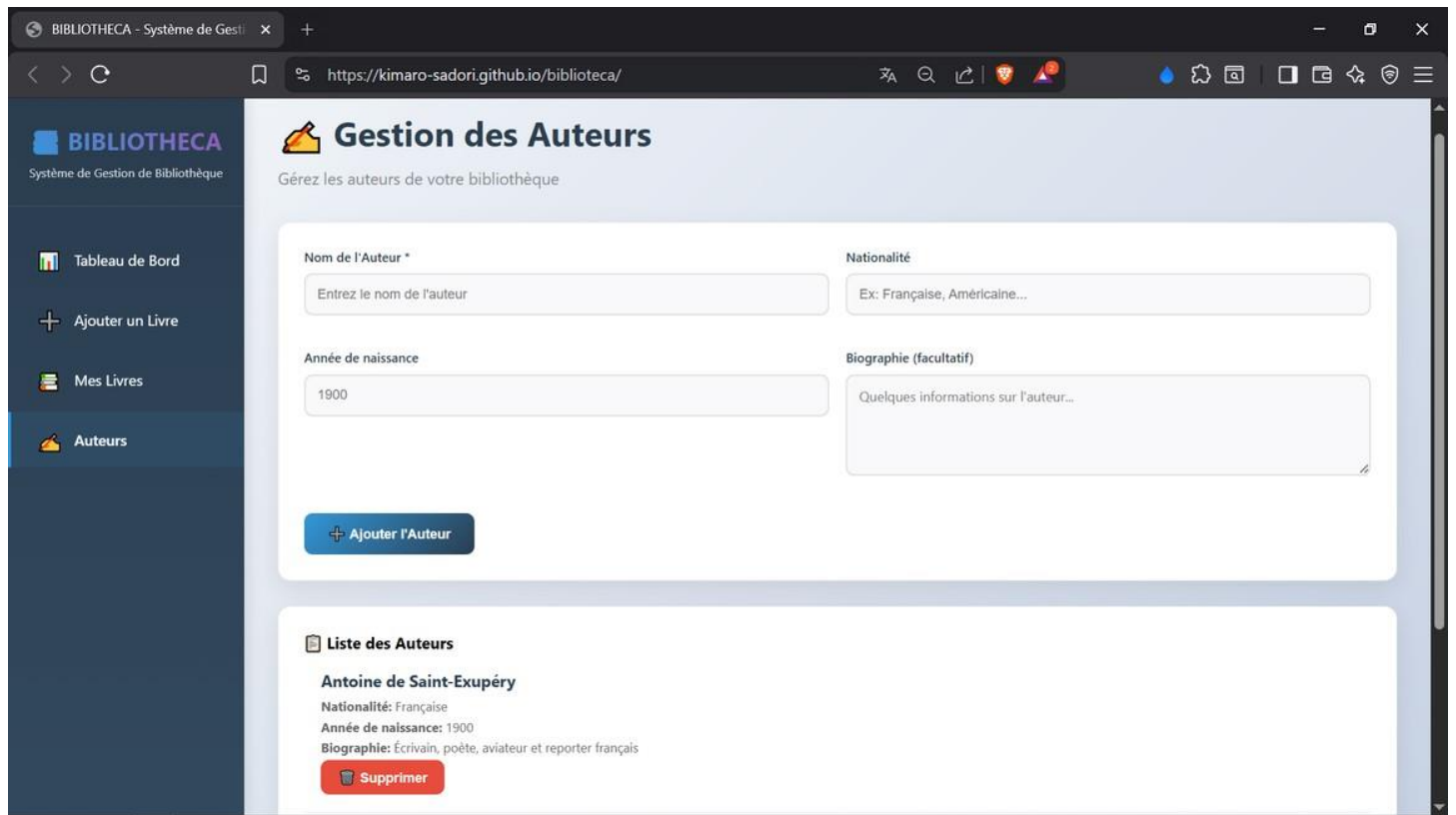
Chaque auteur possède :

- nom
- nationalité année
- de naissance
- biographie

Les auteurs peuvent être :

- ajoutés
- affichés
- supprimés

Le nombre total d'auteurs est automatiquement mis à jour dans le tableau de bord.



## 11. Graphiques et statistiques (Chart.js)

### 11.1 Graphique par genre

Un graphique circulaire (pie chart) montre :

- la répartition des livres par genre

### 11.2 Graphique par année

Un graphique linéaire montre :

- l'évolution du nombre de livres sur les 10 dernières années

Les graphiques sont **dynamiquement mis à jour** après chaque modification.

## 12. Intégration de l'API OpenLibrary

### 12.1 Qu'est-ce qu'une API ?

Une API (Application Programming Interface) permet à une application de communiquer avec une autre application via Internet.

### 12.2 Pourquoi OpenLibrary ?

OpenLibrary est une base de données publique contenant des millions de livres.

## 13. Fonctionnement de fetch (explication simple)

La fonction fetch permet d'envoyer une requête HTTP.

Étapes :

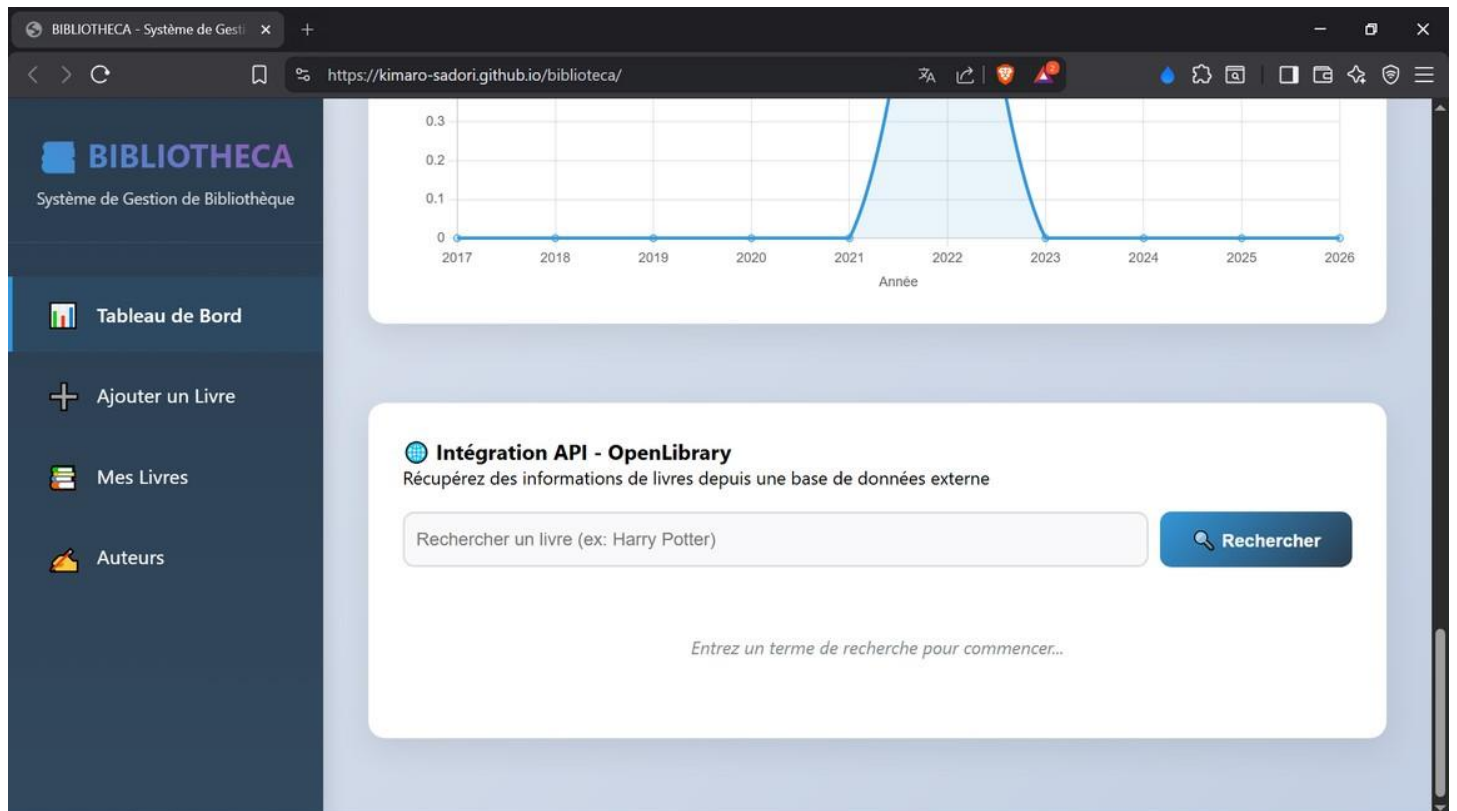
1. L'utilisateur saisit un mot-clé
2. fetch envoie une requête à OpenLibrary
3. Le serveur répond avec des données JSON
4. JavaScript traite ces données
5. Les résultats sont affichés à l'écran Cela se fait **sans recharger la page**.

## 14. Ajout d'un livre depuis l'API

Lorsqu'un utilisateur clique sur **Ajouter à ma bibliothèque** :

- un livre est créé à partir des données
- API il est ajouté au tableau books il est
- sauvegardé dans le LocalStorage le
- tableau de bord est mis à jour

Cela montre une **interaction complète entre API externe et application locale**.



BIBLIOTHECA - Système de Gestion de Bibliothèque

https://kimaro-sadori.github.io/biblioteca/

**Intégration API - OpenLibrary**  
Récupérez des informations de livres depuis une base de données externe

happy

Rechercher

**The Happy Prince and other tales**  
Auteur: Oscar Wilde  
Année: 1888  
Ajouter à ma bibliothèque

**Where the Wild Things Are**  
Auteur: Maurice Sendak  
Année: 1963  
Ajouter à ma bibliothèque

**A Wrinkle in Time**  
Auteur: Madeleine L'Engle

## 15. Initialisation de l'application

Au chargement de la page :

- les données sont chargées depuis le LocalStorage des
- livres exemples sont ajoutés si la bibliothèque est vide •
- les graphiques et statistiques sont affichés

Cela garantit une application toujours fonctionnelle.

## 16. Sécurité et limitations

- Pas de base de données serveur
- Pas d'authentification utilisateur
- Dépendance à la connexion Internet pour l'API

Ces limites sont normales pour un projet académique.

## 17. Conclusion

Le projet **BIBLIOTHECA** est une application web complète qui démontre :

- la maîtrise du JavaScript
- moderne la gestion dynamique
- du DOM l'utilisation d'API REST la
- visualisation de données une
- bonne organisation du code

Il constitue une base solide pour un futur projet professionnel ou une extension vers une application full-stack.