

Rapport de Projet : Smart Backoffice Dashboard - Bibliotheca

1. Introduction

Le présent projet s'inscrit dans le cadre du module de Développement Web. L'objectif est de concevoir et réaliser une application Web complète de type "Backoffice Dashboard" nommée **Bibliotheca**. Cette application vise à offrir une interface moderne et structurée pour la gestion d'un fonds documentaire.

Conformément aux exigences pédagogiques, le projet est développé exclusivement en **HTML5, CSS3 et JavaScript Vanilla**, sans recours à des frameworks externes, afin de démontrer une maîtrise approfondie de la manipulation du DOM et de l'asynchronisme.

2. Analyse du Sujet

Le sujet choisi, **Bibliotheca**, répond à une problématique de gestion de données liées au monde de l'édition et des bibliothèques.

L'application doit permettre de centraliser les informations concernant deux entités principales : les livres et leurs auteurs.

Les besoins identifiés pour cette application sont les suivants :

- **Visualisation globale** : Un tableau de bord (Dashboard) pour suivre les indicateurs clés (KPI) de la bibliothèque.
- **Gestion métier** : Un module complet pour la manipulation des livres (CRUD) et un module pour la gestion des auteurs.
- **Ouverture sur l'extérieur** : L'intégration de données réelles via l'API OpenLibrary pour enrichir l'expérience utilisateur.

3. Conception des Modules et Données

Pour répondre à ces besoins, l'application est structurée en trois modules distincts, organisés au sein d'une **Single Page Application (SPA)**.

3.1. Structure des Modules

- **Module 1 (Principal - Livres)** : Ce module constitue le cœur du système. Il permet de gérer les ouvrages avec des fonctionnalités de création, lecture, mise à jour et suppression (CRUD complet).
- **Module 2 (Secondaire - Auteurs)** : Ce module gère la liste des auteurs. Il est lié au premier module car chaque livre est rattaché à un auteur spécifique.
- **Module 3 (Analytique - Dashboard)** : Ce module présente une vision synoptique avec des indicateurs de performance (KPI) calculés dynamiquement et des graphiques.

3.2. Modèle de Données

Pour la semaine 1, nous avons défini les champs de données suivants pour le module principal :

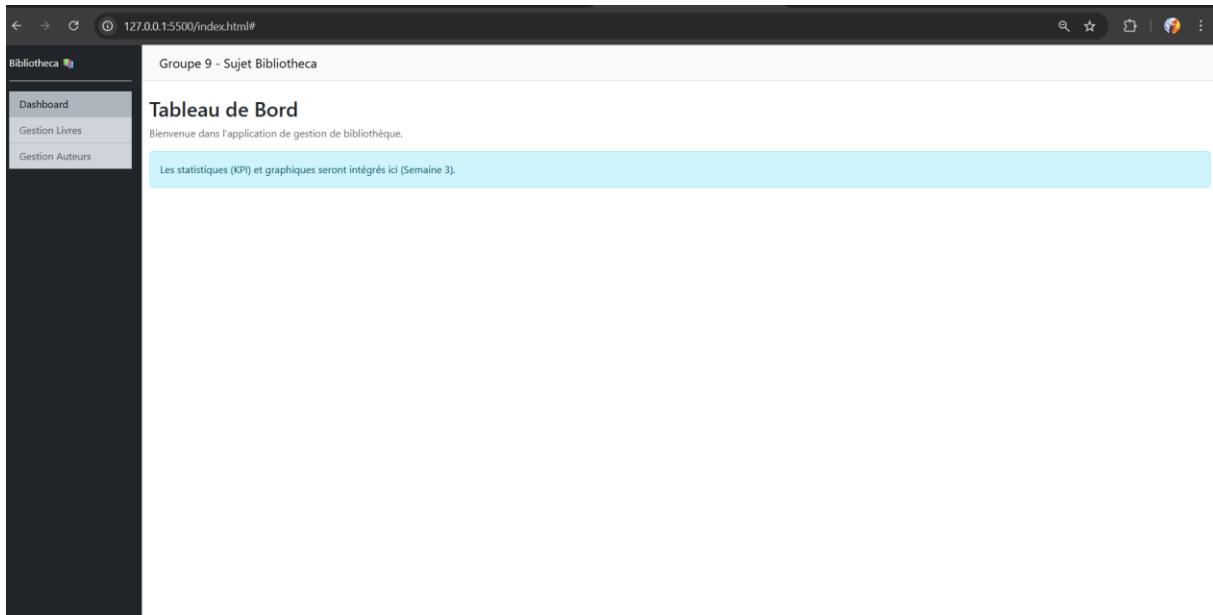
- **Livre** : Titre, Auteur, Genre (Roman, Science-Fiction, Essai, etc.).

3.3. Architecture Technique

L'application repose sur une architecture SPA où toutes les sections sont incluses dans un fichier HTML unique. Le passage d'une section à l'autre (Dashboard, Livres, Auteurs) se fait dynamiquement via JavaScript en manipulant les propriétés d'affichage (display) du DOM, garantissant une navigation fluide sans recharge de page.

Capture 1 : Interface Globale (Dashboard actif)

Cette capture d'écran présente l'interface principale de l'application **Bibliotheca**. Conformément à la structure de type **Single Page Application (SPA)** demandée . La section "Dashboard" visible ici sert de vue analytique où seront intégrés, en semaine 3, les indicateurs clés de performance (KPI) et les graphiques de suivi



Capture 2 : Section Gestion des Livres (Module 1)

Partie dédiée à la gestion du fonds documentaire. Le formulaire a été conçu en **JavaScript Vanilla** pour permettre l'ajout dynamique d'ouvrages. Chaque champ (Titre, Auteur, Genre) est soumis à une validation avant traitement pour garantir l'intégrité des données locales. Cette interface respecte les principes de manipulation avancée du DOM en créant des éléments en temps réel sans rechargement de la page.

