

Application Full Stack Data

Réalisation d'une bibliothèque digitale

ROUAI Souhail , EL GUEZDI Mohamed Zinelabidine , BOUZMOUL Mohamed amine

26/11/2022



Table de matiere :

Partie 1 : Contexte générale

L'idée de projet

Partie 2 : Réalisation du projet

Technologies utilisées

Démonstration du projet

Guide pour lancer le projet

Partie 1 : Contexte générale

- l'idée de notre projet est de réaliser une application web qui permet à l'utilisateur de chercher un livre et de lui donner la possibilité de l'ajouter sur sa bibliothèque numérique.
Ainsi de faire la classification des livres selon l'état (livre fini / en cours de lecture , non fini) .
Et finalement l'utilisateur peut évaluer son livre, ça dépend de sa satisfaction .

Partie 2 : Réalisation du projet

Technologies utilisées :

Back end :

Pour la partie backend , nous avons travaillé avec le langage python et précisement avec un le framework FastAPI

Nous avons commencé par la définition d'ensemble des fonctions nécessaires pour l'application .



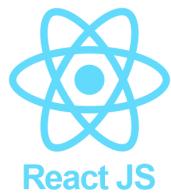
Pour la base de données , nous avons utilisé PostgreSQL qui est un système de gestion de base de données relationnelle orienté objet puissant qui est capable de prendre en charge en toute sécurité les charges de travail de nos données.

Création d'une table : **book_library**

avec 7 colonne (id , volume_id , title , authors , thumbnail , state , rating)

	id [PK] bigint	volume_id character varying(50)	title character varying(50)	authors character varying(50)	thumbnail character varying(500)	state integer	rating integer
1	43	2oVeEAAQBAJ	Soccer	Yvan DeMuy	http://books.google.co...	0	3
2	44	uwOjDwAAQBAJ	L'économie de partag...	Soraya Sedkaoui, Mo...	http://books.google.co...	0	3
3	45	rFOQBAAAQBAJ	La Révolution Big data	Yves Eychenne, Jean-...	http://books.google.co...	2	null
4	46	kEBmDwAAQBAJ	Intelligence artificiel...	Fernando Lafrate	http://books.google.co...	0	5
5	47	apjBAgAAQBAJ	Big Data at Work	Thomas Davenport	http://books.google.co...	0	3
6	48	2oVeEAAQBAJ	Soccer	Yvan DeMuy	http://books.google.co...	2	3

Front end :



Le framework React est le choix idéal surtout sur les applications à page unique (SPA), car il peut réécrire et mettre à jour le contenu d'une page Web sans qu'il soit nécessaire de recharger ou de rafraîchir la page .

Pour qu'on puisse chercher un livre sur internet , nous avons utilisé un API (Google Books APIs) qui permet de faire cela .

La première chose est la génération du clé API sur google cloud .

The screenshot shows the Google Cloud Platform API library interface. On the left, there's a sidebar with options like 'API et services activés', 'Bibliothèque', 'Identifiants' (which is selected), and 'Écran de consentement OAuth'. The main area is titled 'Identifiants' and has a sub-section 'Clés API'. A modal window is open over the list, titled 'Clé API 1'. It contains instructions: 'Utilisez cette clé dans votre application en la transmettant avec le paramètre key=API_KEY .' Below this is a text input field labeled 'Votre clé API' containing the value 'AIzaSyBEzkPTX6ugmnTzbdb4mv3-RGF4rAE3jVw'. A warning message follows: '⚠ Cette clé n'est pas restreinte. Pour éviter toute utilisation abusive, nous recommandons d'ajouter des restrictions pour limiter les emplacements et les API pour lesquels elle peut être utilisée. [Modifiez la clé API](#) pour ajouter des restrictions.' At the bottom of the modal are 'FERMER' and 'AFFICHER LA CLÉ' buttons.

Ensuite nous avons utilisé Books API sur notre application .

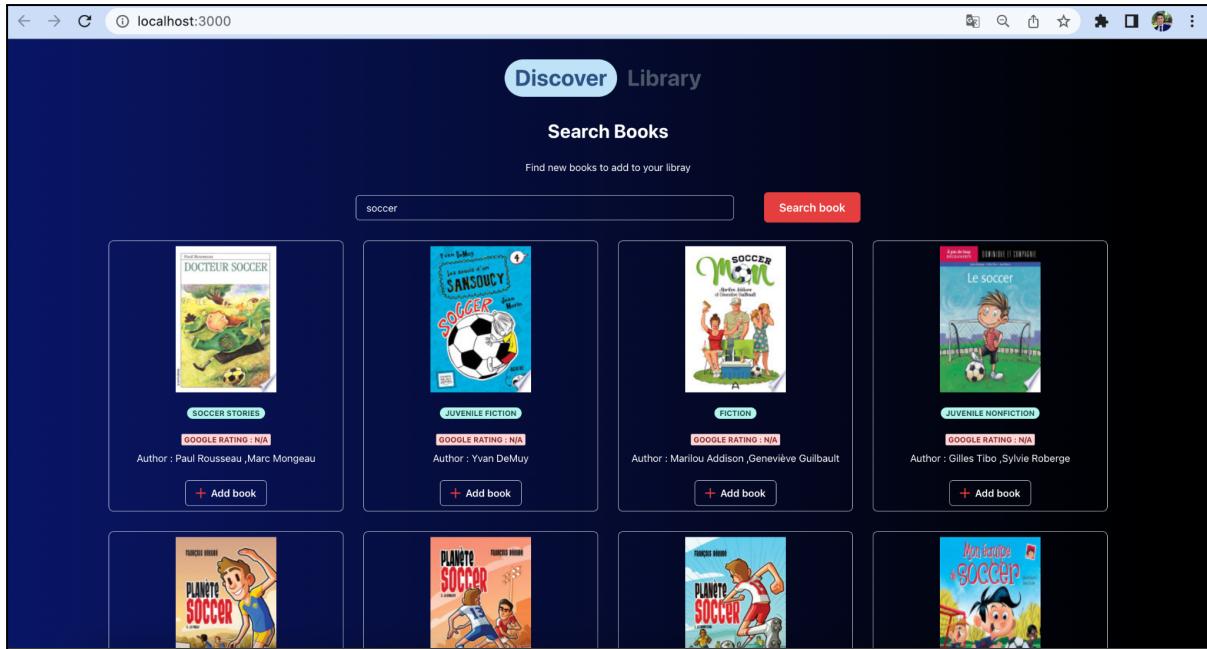
The screenshot shows the Google Cloud API Library interface. At the top, there's a blue header bar with the text "Google Cloud" and "Library". Below the header, the main content area has a title "Books API" with a circular icon containing a diamond shape. Underneath the title, it says "The Google Books API allows clients to access the Google Books repository." There are two buttons: "GÉRER" (Manage) and "ESSAYER CETTE API" (Try this API). A green checkmark indicates "API activée" (API activated). Below these buttons, there are two tabs: "APERÇU" (Preview) which is selected, and "DOCUMENTATION". The "APERÇU" section contains a "Présentation" paragraph and an "À propos de Google" section. The "Présentation" paragraph reiterates that the API allows clients to access the Google Books repository. The "À propos de Google" section provides information about Google's mission to organize the world's information and make it universally accessible and useful. To the right of this preview section, there's a "Plus d'infos" (More info) sidebar with details like Type: SaaS & APIs, Last updated: 21/07/2022, Category: Other, and Service URL: books.googleapis.com.

Démonstration du projet :

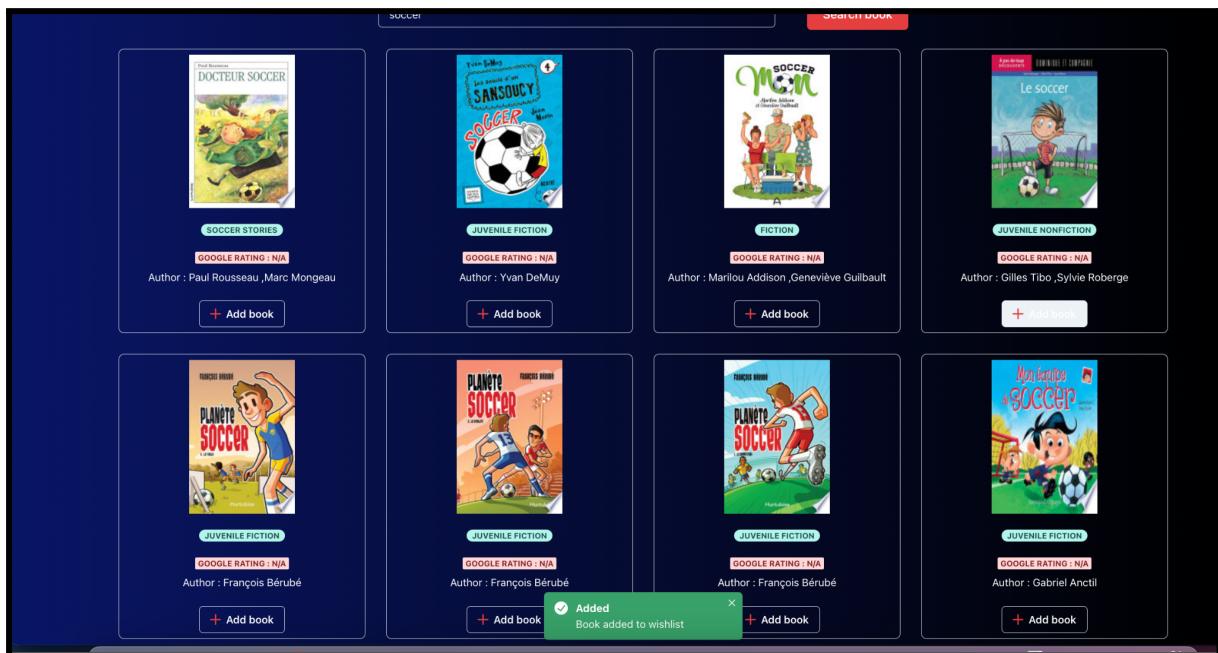
Cette page donne la main à l'utilisateur de chercher un livre

The screenshot shows a dark-themed search interface. At the top, there are two buttons: "Discover" and "Library". Below them is a search bar with the placeholder text "Search Books" and a sub-instruction "Find new books to add to your library". To the right of the search bar is a red button labeled "Search book". In the center of the screen, the text "You gotta search to see results !" is displayed in white. The bottom portion of the interface is a large, empty white area.

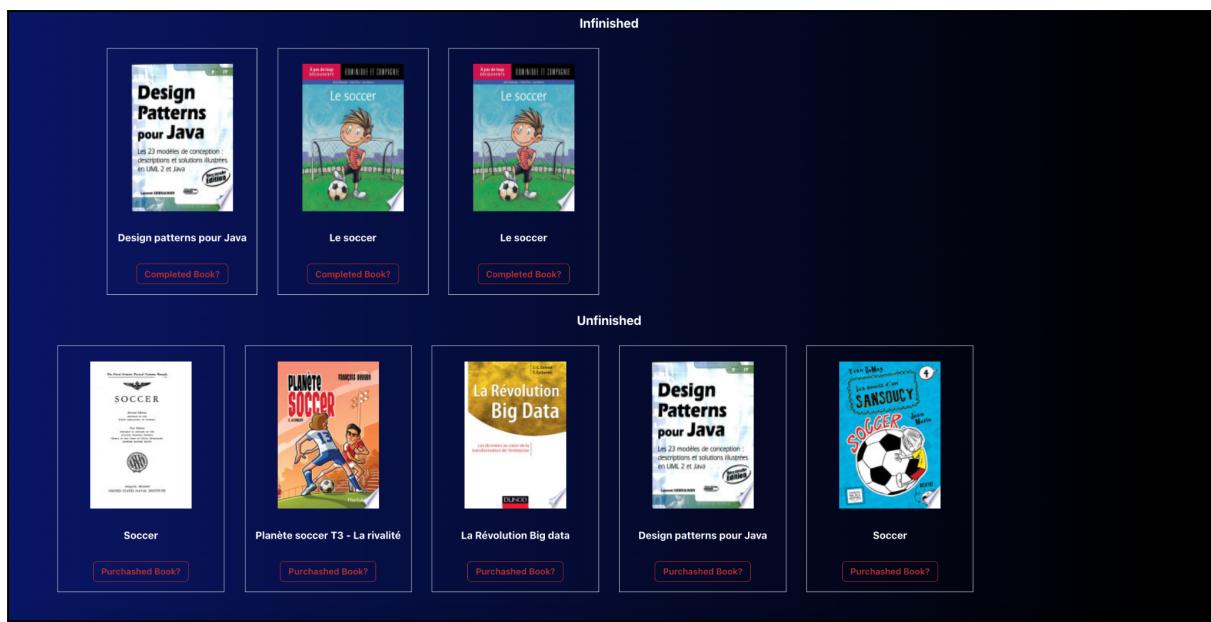
page de résultat qui contient l'ensemble des livres avec les détails (couverture , titre , auteur ..) :



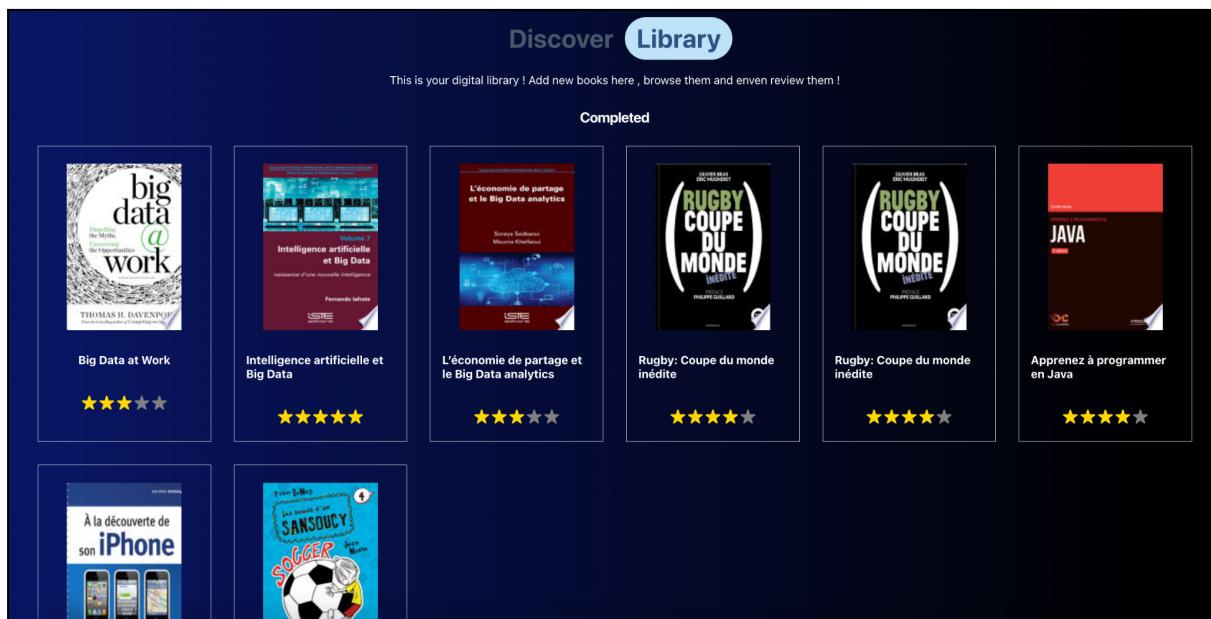
Pour choisir un livre et l'ajouter sur la bibliothèque , il suffit de cliquer sur le bouton Add Book

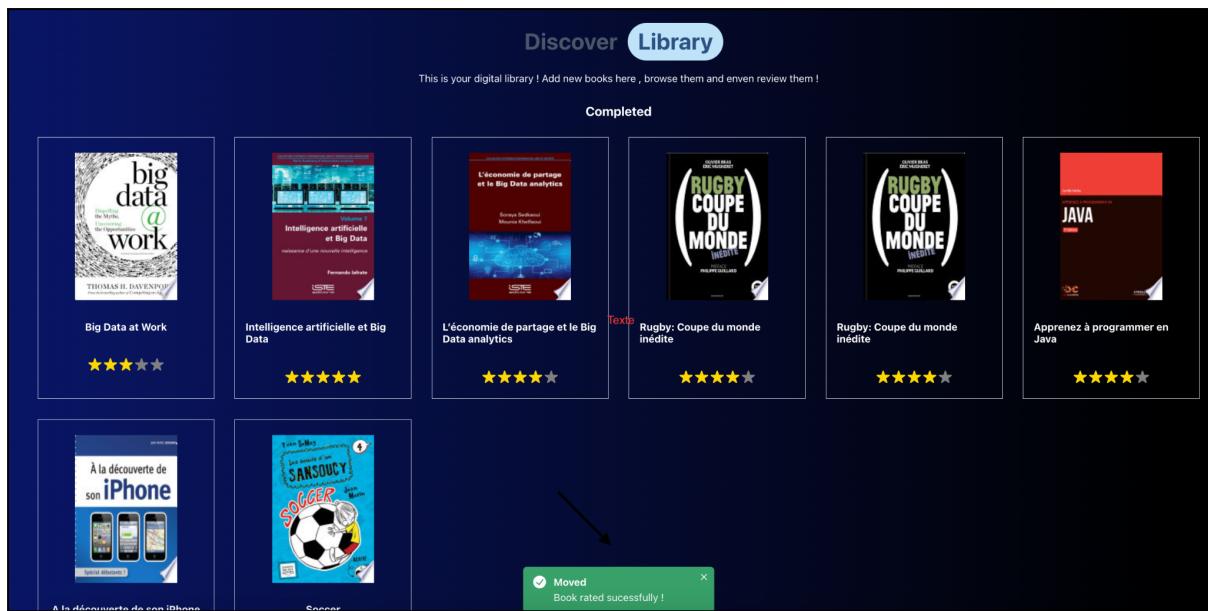


Après le livre s'ajoute automatiquement sur la bibliothèque (Unfinished).
Après , l'utilisateur clique sur Purchased book



Et finalement après la lecture de livre , l'utilisateur peut donner son degré de satisfaction du livre en cliquant sur une étoile correspondant à le livre





Guide pour lancer le projet :

exécution local classique :

Pour lancer le back-end :

uvicorn app:app --reload

Pour lancer front-end :

npm start

pour lancer le projet avec docker :

Sur le répertoire du back : docker run -p 3000:3000 react-image

Sur le répertoire du front : docker run -p 8000:8000 python-fastapi

Sur le fichier docker-compose.yaml : docker-compose up

