Progetto Java Script Advanced per il Master in Front End Development di Luca Radatti

Book Finder

Book Finder è un’applicazione web intuitiva progettata per offrire agli utenti un’esperienza semplice e veloce per cercare libri basati su categorie specifiche.

Grazie a un’interfaccia piacevole e a una barra di ricerca, gli utenti possono esplorare un vasto catalogo di libri, visualizzare i risultati in tempo reale e approfondire i dettagli dei libri di loro interesse.

Caratteristiche Principali

Ricerca per categorie

Visualizzazione dei risultati

Approfondimento del Libro

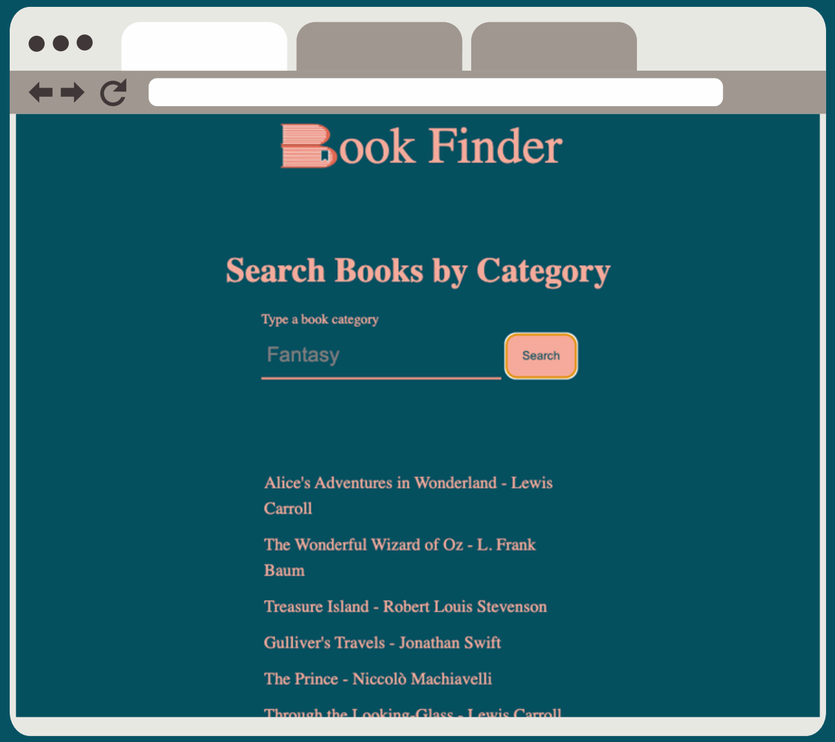
Interfaccia Responsive

Ricerca per categorie

Gli utenti possono digitare una categoria predefinita (es. narrativa, saggistica, fantascienza, poesia) direttamente nella barra di ricerca.

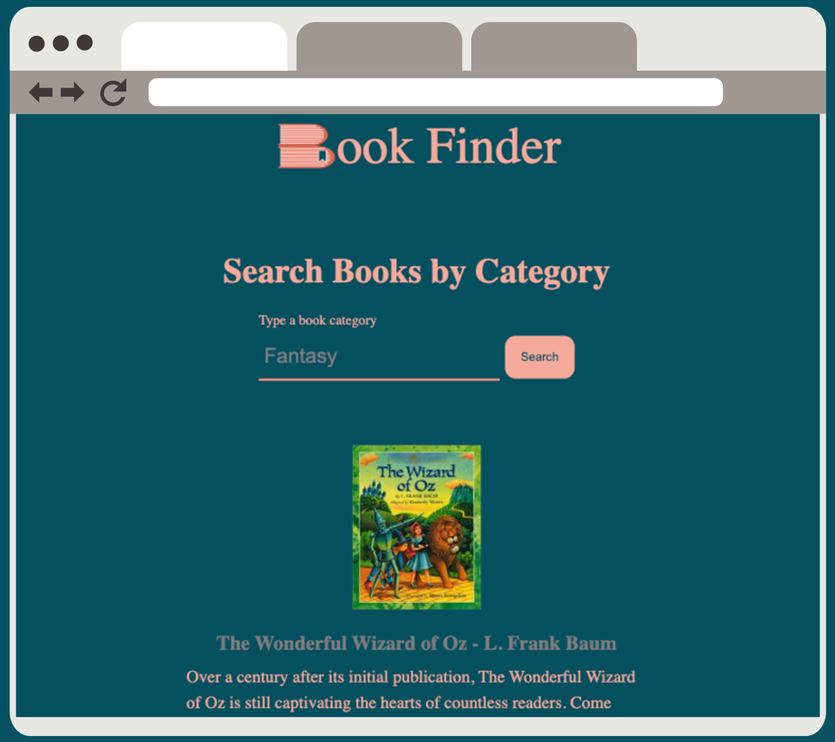
Visualizzazione dei risultati

I risultati della ricerca vengono presentati in un layout semplice ed ordinato



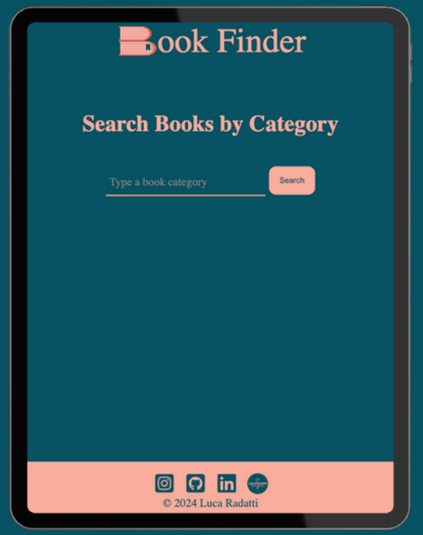
Approfondimento del Libro

Cliccando su un libro specifico, l’utente può accedere alla descrizione e alla copertina.



Interfaccia Responsive

Il sito è ottimizzato per funzionare su dispositivi desktop, tablet e mobile, garantendo un’esperienza fluida su tutte le piattaforme.



Requisiti Progetto

L’applicazione dovrà essere composta da un semplice **textbox** (**Google** style) per permettere all’utente di cercare tutti i libri di una specifica categoria.

Una volta che l’utente cliccherà su un apposito button, l’applicazione dovrà contattare le API del servizio esterno **Open Library**: https://openlibrary.org/subjects/fantasy.json dove **fantasy** è la categoria inserita dall’utente. L’applicazione una volta recuperata la lista dei **libri** dovrà visualizzare solamente il **titolo** e l’elenco degli **autori**.

Al **click** dell’utente su un **libro** o su un apposito **button**, l’applicazione deve poter visualizzare la descrizione del libro. Per rendere disponibile questa funzionalità l’applicazione deve contattare un’altra **API** del servizio **Open Library** passando la key del libro presente nella risposta al servizio contattato precedentemente.

Tecnologie Utilizzate

Bundler di moduli: Webpack

Linguaggi: Html, Scss, JavaScript

Librerie: Lodash, Axios

Approfondimento

Search Bar

L’applicazione dispone di una **barra di ricerca** con un pulsante “**Search**”.
Quando l’utente digita la **categoria preferita**, può avviare la ricerca cliccando sul **pulsante** oppure premendo il tasto **Invio**.
Il valore inserito dall’utente viene acquisito tramite il metodo .value del form.

Successivamente, il valore viene elaborato per:

- **Rimuovere** eventuali **spazi** all’inizio e alla fine della stringa.
- **Convertire** l’intera stringa in **minuscolo** utilizzando il metodo .toLowerCase().
- **Rimuovere** ulteriori spazi tra le stringhe, sostituendoli con un **underscore**.

Se l’utente non inserisce alcuna **categoria**, verrà mostrato un **messaggio di errore** che lo inviterà a specificare una **categoria** prima di proseguire.

Approfondimento

Creazione dell’Url

Una volta **elaborato** il valore inserito dall’utente, questo viene integrato **nell’URL** tramite la creazione di un oggetto new URL.

Il **valore** dell’input sostituisce la parte relativa alla **categoria**, mentre la base **dell’URL** rimane invariata, puntando al sito https://openlibrary.org/subjects/.

Al termine della **manipolazione**, **l’URL** completo sarà, ad esempio: https://openlibrary.org/subjects/love.json

Approfondimento

Invio richiesta HTTP

Grazie **all'URL generato**, possiamo inviare una **richiesta HTTP** al sito **openlibrary.org** per ottenere i **dati** relativi ai **libri** della **categoria selezionata**.
Per effettuare questa operazione, utilizziamo la libreria **JavaScript Axios**, che semplifica l'invio della **richiesta** e ci restituisce i **dati** già in formato **JSON**.

Successivamente, **manipoliamo** la **risposta** per estrarre informazioni utili, come:

- I **titoli** dei **libri**.
- Gli **autori**.
- Le **chiavi di identificazione** dei singoli libri.
- Le **chiavi** identificative delle **copertine**.

Per gestire in modo efficiente **l'estrazione** di questi **dati**, utilizziamo la libreria **Lodash** e il metodo _.get, che consente di cercare e accedere facilmente ai **valori** all'interno **dell'oggetto risposta**.

Alla fine del processo, mostreremo una **lista** contenente i **titoli** dei **libri** e i rispettivi **autori**.

Approfondimento

Scelta del libro

Una volta visualizzata la **lista** di **titoli** e **autori**, l'utente potrà selezionare un **titolo** per accedere alla **descrizione dettagliata** del **libro**.
Cliccando sul **titolo**, verrà preso il **valore** contenuto nel tag <li> che rappresenta il **titolo selezionato**.

La **stringa** estratta viene poi **manipolata** per ottenere solo il **titolo** del **libro**, escludendo **l'autore**.
Questo viene realizzato tramite una **funzione personalizzata** che identifica il carattere "-" e restituisce soltanto la parte che lo precede, ovvero il **titolo**.

Il **titolo** così ottenuto viene **normalizzato** e passato a un'altra **funzione**, che lo confronta con un **array** di **oggetti** contenente i **titoli** dei **libri** della **categoria**, insieme ad altre informazioni come **autore**, **ID** della **copertina** e **ID** del **libro**.

Una volta trovata la **corrispondenza** con il **titolo** scelto dall'utente, la **funzione** restituisce un **oggetto** contenente tutti i **dettagli** del **libro selezionato**.

Approfondimento

Nuova richiesta HTTP

Una volta ottenuto **l'oggetto** con tutti i **dati necessari**, estraiamo **l'ID** del **libro**,
ad esempio works/OL21177W, e lo inseriamo **nell'URL** nello stesso modo utilizzato per la richiesta dei **libri** per **categoria**.

A questo punto, inviamo la **richiesta** per ottenere i **dati specifici** del **libro selezionato**. I **dati** ricevuti vengono **manipolati** per ottenere due **elementi principali**:

- La **descrizione** del **libro**.
- **L'immagine di copertina**.

La **descrizione** viene sottoposta a **un’istruzione di controllo if-else**, che verifica la presenza del **testo** per poi inserirlo in un **contenitore** insieme **all'immagine di copertina**.

Infine, è stato introdotto un **ritardo di 2 secondi** per mostrare la **descrizione**, garantendo che **l'immagine di copertina** venga **caricata completamente** prima della **visualizzazione**.

Approfondimento

Creazione dinamica del contenitore

È stato scelto di **creare** un **contenitore** in modo **dinamico** per **visualizzare** sia i **risultati** della chiamata HTTP relativa alla ricerca per **categoria**, sia i **dettagli** del **libro** selezionato.

Lo stesso **contenitore** viene riutilizzato in entrambi i casi, venendo **svuotato** prima di inserire i nuovi **dati**, garantendo così una **gestione ordinata** delle **informazioni** visualizzate.

Un **contenitore** separato è stato **creato appositamente** per ospitare **l'immagine** di copertina del **libro**, concentrandosi esclusivamente su questo **elemento visivo**.

Grazie per l’attenzione

Lascio qui sotto il link per il repository GitHub e il link per provare l’applicazione

https://github.com/radattiluca/project-book-finder

https://bookfinderbycategory.netlify.app

Progetto Java Script Advanced