

BookVue: una breve descrizione

BookVue è una grande libreria di Torino che ha deciso di allargare la propria rete di vendita a clienti privati su tutta Italia. BookVue offre una vasta selezione di libri contemporanei di narrativa e saggistica italiana e ha deciso di aprire un e-commerce così da poter vendere i libri del suo catalogo anche online.

La **navbar** dell'e-commerce è costituita da:

- Home
- Libri
- Chi siamo
- Promozioni
- Area "ricerca"
- Area "carrello"

All'interno della **homepage** troviamo una struttura su due colonne (così come per l'intera single web application): in quella principale abbiamo un pulsante che ricollega alla vista '**Chi siamo**', in cui spieghiamo i nostri obiettivi e il funzionamento del servizio, un carosello con i "*Libri in evidenza*" dove cliccando sul bottone "Dettagli" si accede alla vista di dettaglio del libro ed infine un "hero" promozionale che rimanda alla possibilità di registrarsi sul sito. Invece, nella *sidebar* abbiamo le card con le varie promozioni in corso: coupon di benvenuto per gli studenti delle scuole superiori, possibilità di vincere un libro rispondendo correttamente alle domande di un quiz e di risparmiare vendendo un libro usato.

La vista **libri** permette di visualizzare l'elenco dei libri presenti nello store con informazioni chiave quali il titolo, una breve descrizione, l'autore, il prezzo e disponibilità in magazzino. Ogni libro ha un bottone "Dettagli" che rimanda alla **vista di dettaglio** in cui troviamo ulteriori dettagli come:

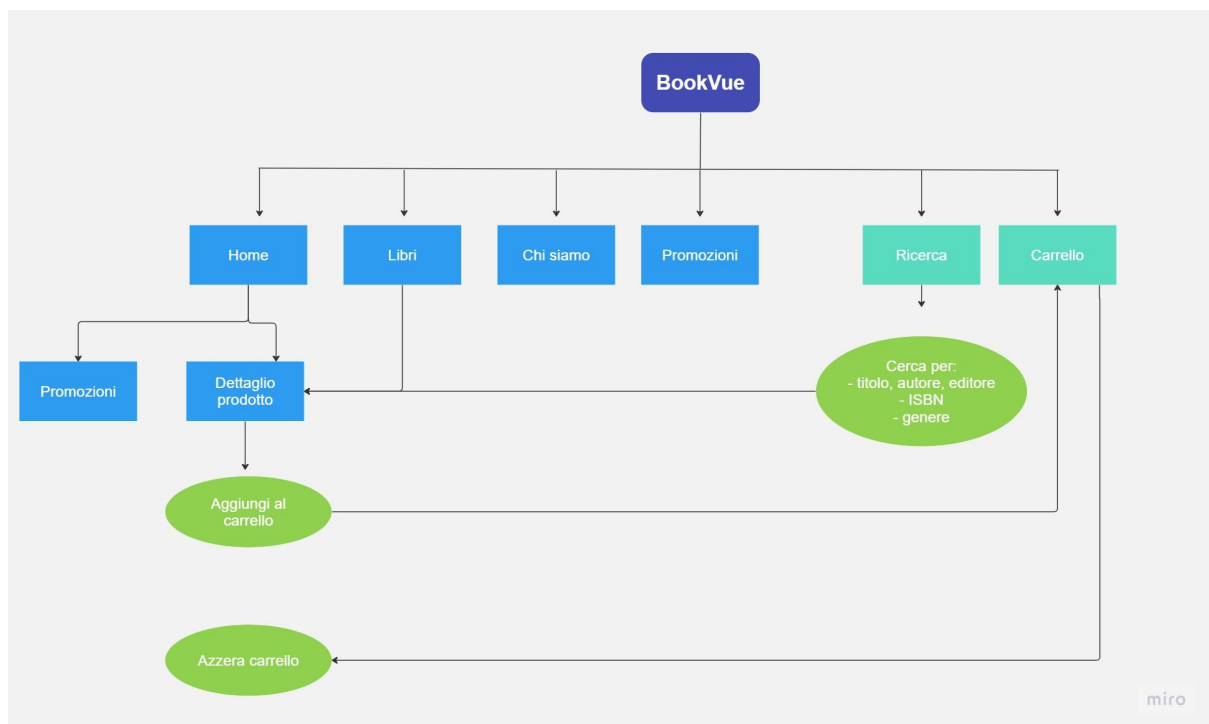
- Immagine della copertina
- Nome dell'autore/autrice
- Titolo del libro
- Editore
- Genere
- ISBN
- Anno di pubblicazione
- Numero pagine
- Lingua
- Disponibilità a magazzino
- Prezzo
- Trama

Per ogni libro c'è la possibilità di **aggiungere il prodotto** al carrello tramite il pulsante "Aggiungi". Nel momento in cui un libro viene aggiunto al carrello viene mostrata sotto al pulsante la quantità di copie aggiunte e il n. di copie disponibili a magazzino si aggiorna in automatico diminuendo.

L'area di **ricerca** propone tre modalità: la ricerca testuale nel catalogo (che cerca negli attributi titolo, nome dell'autore/autrice, editore), la ricerca per codice ISBN, la ricerca per categoria.

Infine, l'area del **carrello** offre un riepilogo dei prodotti aggiunti indicandone titolo, quantità e prezzo, e mostrando la copertina del libro. E' inoltre presente il prezzo totale dei prodotti presenti nel carrello. Con il pulsante "azzerà carrello" è possibile rimuovere tutti i prodotti dal carrello e iniziare da capo con gli acquisti.

Flowchart



Costruzione dell'ambiente e GitHub

Per iniziare a lavorare sul nostro progetto, siamo partiti dall'installazione e configurazione di git e del framework Vue.js. Dopo aver generato le chiavi crittografiche ed aver effettuato la registrazione su **GitHub**, abbiamo proceduto con la creazione di un new repository che abbiamo condiviso tra di noi così da poter lavorare simultaneamente al progetto.

A questo punto, siamo passati alla creazione del progetto vue. Uno dei primi passaggi è stato quello di installare il framework Bootstrap ed importarlo nel file **main.js**, successivamente abbiamo creato in **store/index.js** il nostro *database* con tutti i dati indispensabili per l'importazione nelle pagine del sito come, ad esempio, le "cards" (promozioni), i "products" (libri) ecc. Inoltre, in corso d'opera, sono state aggiunte anche delle mutazioni (esempio: addToCart) e dei *getters* che hanno permesso di ottenere dati dallo stato, come ad esempio la lista degli articoli nel carrello.

Abbiamo poi organizzato le immagini come segue: quelle destinate ad essere utilizzate direttamente sulle pagine, come ad esempio quelle per la scheda prodotto, sono state

inserite nella cartella “**public**”, invece gli elementi grafici, come ad esempio il logo, nella cartella “**src/assets/images**”.

Successivamente, abbiamo definito la **struttura del layout**, creando i componenti corrispondenti per la NavBar, NavIcon, SideBar FooterBar, vista principale e la SideBar attraverso l'utilizzo di codice HTML, JavaScript e CSS.

Abbiamo creato la vista **MainView** dove sono presenti jumbotron con presentazione del servizio, carousel, hero con sconto promozionale. Il carousel mostra i libri in evidenza, cioè quelli che nello store hanno l'attributo *'highlight: true'*. Attraverso il link su dettagli si viene rimandati alla vista della scheda prodotto (con la logica spiegata di seguito).

A questo punto siamo passati alla creazione della vista **ProdottiView** che contiene l'elenco dei prodotti, il quale viene visualizzato attraverso la direttiva *v-for*. Abbiamo poi implementato un *router-link* che rimanda alla scheda prodotto (**DettaglioView.vue**). La selezione del prodotto avviene tramite la variabile *productSelected* presente nello state dello store; questa variabile si aggiorna attraverso una *mutation* quando il prodotto viene selezionato. Nella DettaglioView, una *computed property* utilizza questa variabile per consentire la visualizzazione degli attributi del prodotto selezionato nella vista.

Per aggiungere il prodotto al carrello, abbiamo creato la vista di riferimento del carrello, ovvero la vista **CartPage.view**, che contiene una lista di prodotti che si attiva nel momento in cui l'utente, all'interno della scheda prodotto (**DettaglioView.vue**), decide di cliccare sul pulsante 'aggiungi'. Per garantire un'esperienza chiara e comprensibile per l'utente e per assicurare che ciò avvenga correttamente, abbiamo impostato all'interno dello store due variabili: *'cart'*, inizializzata a 0 così che il carrello inizialmente sia sempre vuoto, e *'inventory'*, che stabilisce la quantità dei libri presenti nello store.

Per fare in modo che, una volta aggiunti i libri nel carrello, questi due valori vengono modificati, abbiamo definito una funzione *'addToCart'* all'interno della vista di dettaglio. Questa funzione consente, una volta aggiunto il prodotto nel carrello, di modificare contemporaneamente sia il valore iniziale del carrello che la quantità disponibile di quel determinato prodotto.

Dopo di che è stato necessario implementare una funzione che permettesse di aggiungere effettivamente i prodotti nel carrello andando a mostrarli sotto forma di lista. Per questo abbiamo impostato un *'getter'* che modificasse lo stato iniziale del carrello e una *mutations* chiamata *'addToCart'* nello store che lavora sullo stato del carrello e del prodotto riferendosi ai loro id. Dopo di che abbiamo utilizzato *'MapGetters'* così da poter collegare il getter alle operazioni dei componenti.

Infine, è stato necessario richiamare la visualizzazione dei prodotti nel carrello. Per questo motivo, all'interno di **CartPage.view** è stata creata una lista che recuperasse dallo store i dettagli dei prodotti, tra cui l'immagine, il titolo, l'autore, la quantità del prodotto selezionata e il prezzo. Per quanto riguarda il prezzo abbiamo utilizzato il metodo *'totalPrice'* per calcolare il costo totale dei libri presenti nel carrello, tenendo conto delle rispettive quantità. Inoltre, abbiamo garantito all'utente la possibilità di poter rimuovere gli elementi scelti; a tal fine, è stata creata la funzione *'azzeraCarrello'*, attivata attraverso l'evento di click sul relativo pulsante nella pagina. Questa funzione richiama un *methods* inserito all'interno dello store, che va a modificare lo stato del carrello riportandolo al valore iniziale e la quantità di libri presenti al valore precedente alla selezione.

Successivamente, ci siamo occupati della funzionalità di ricerca all'interno della vista **SearchView.vue**. In questa vista, sono presenti le tre modalità di ricerca precedentemente menzionate. Nel caso di "cerca nel catalogo", quanto l'utente scrive nel campo di input viene assegnato alla variabile '*SearchTerm*' attraverso la funzione *search*. All'interno di questa funzione, il valore immesso viene confrontato con gli attributi "titolo", "autore" e "editore" dei libri attraverso '*filter*'. Se uno di questi attributi contiene il termine di ricerca specificato (utilizzando il metodo "*includes*"), il prodotto viene incluso nella lista dei risultati. Le altre possibilità di ricerca (ISBN e genere) sono visibili aprendo i relativi *accordion*.

Infine, abbiamo lavorato sul **routing** delle pagine: abbiamo definito le diverse viste all'interno del file "**router/index.js**", importando i rispettivi componenti e specificando i percorsi corrispondenti a ciascuna vista.

Istruzioni per testare il progetto

1. Aprire il prompt dei comandi e fare **git clone** + indirizzo SSH (recuperato da GitHub)
2. Entrare nella cartella in cui si è clonato il progetto ed installare il pacchetto node modules con il comando **npm install**
3. Eseguire il progetto attraverso il comando **npm run serve**
4. Esempio di navigazione sul sito:
 - a. Aprire la homepage tramite il localhost con il numero della porta suggerita dal terminale, cliccare sulla navbar su 'Libri', arrivare sulla vista 'Prodotti', cliccare sul pulsante 'Dettagli' all'interno delle card che contengono il prodotto, atterrare sulla vista di dettaglio, aggiungere il prodotto al carrello, visualizzare la vista carrello per decidere se tenere i prodotti selezionati o azzerare il carrello.
 - b. Una volta aperta la homepage, cliccare sull'icona 'Ricerca' a destra della navbar, atterrare sulla vista 'Search' dove si possono effettuare ricerche nel catalogo utilizzando diverse opzioni. I risultati della ricerca verranno mostrati sotto e cliccando sul titolo del libro si viene reindirizzati alla vista di dettaglio relativa a quel libro.
 - c. Partendo dalla homepage cliccando sulle card all'interno della sidebar si arriva sulle pagine di dettaglio delle promozioni.