

LumBooks

Progetto del corso di Tecnologie Web
Gruppo LumBreakers
A.A. 2018/2019

Informazioni sul gruppo

Membri	Caccaro Sebastiano - 1142944 Pontecchiani Elena - 1143582 Zanatta Stefano - 1142897 Zhang Linpeng - 1162114
---------------	--

Descrizione

Lorem ipsum

Indice

1 Introduzione

1.1 Abstract

Il progetto *LumBooks*, svolto come finalità del corso di Tecnologie Web nell'anno accademico 2018-2019, si propone di implementare un sito Internet per la compravendita di materiale universitario tra studenti.

Il nome "LumBooks", infatti, nasce dall'insieme delle parole Lum (che fa riferimento all'aula Lum250 del complesso Paolotti, dove si svolgono le lezioni del corso di studio di Informatica) e books.

Per conseguire lo scopo stabilito, la natura del sito è fortemente interattiva. Per i visitatori del sito è possibile infatti registrarsi e creare il proprio account personale, grazie al quale l'utente può cercare e vendere il materiale di studio che desidera, valutando offerte di altri utenti del sito.

Nonostante ciò la registrazione al sito non è obbligatoria, e nel caso un visitatore non volesse iscriversi il sito ha in qualsiasi caso funzionalità informative, in quanto permette comunque al visitatore esterno di cercare un determinato libro visualizzando le offerte che il sito propone.

Il sito è stato sviluppato con l'intenzione di essere poi pubblicato in internet, dunque si è data molta importanza alla sua usabilità, rispettando gli standard W3C, la separazione tra struttura, presentazione, comportamento e le regole di accessibilità richieste.

1.2 About us

2 Analisi

2.1 Studio dell'utenza finale

Il progetto LumBooks si propone come mezzo intermediario all'interno di una compravendita di materiale universitario tra gli studenti; pertanto quest'ultimi saranno l'utenza finale primaria del sito, o comunque soggetti legati all'ambito accademico.

Nel particolare inizialmente la piattaforma viene pensata per essere destinata a studenti frequentanti il corso di studi di Informatica presso l'università di Padova (tant'è che il catalogo contiene prettamente libri appartenenti a questo corso), ma in futuro il sito potrebbe espandersi e rivolgersi a qualsivoglia studente frequentante una qualsiasi università.

In prospettiva di una futura crescita, quindi, l'utente finale potrebbe non essere specializzato o pratico di strumenti informatici, dal momento che potrebbe provenire da diverse facoltà. Per questo è necessario adottare un linguaggio informale, semplice e di facile intuizione, che possa essere compreso dalla maggior parte delle persone. Lo stesso discorso vale per il layout e la struttura del sito, che si prefiggono di essere intuitivi e veloci, rispettando le principali convenzioni del web.

2.2 Casi d'uso

2.2.1 Utente generico

2.2.2 Utente loggato

Definiamo *loggato* un utente che:

- Ha eseguito con successo il login;
- Si è registrato con successo, ed è quindi stato rimandato alla home. In questo caso, il login è avvenuto automaticamente.

L'utente loggato eredita tutti i casi d'uso dell'utente generico, eccetto:

- Registrazione;
- Login;

L'utente loggato dispone, inoltre, dei seguenti casi d'uso:

- Inserimento di un nuovo libro (2.2.2.1);
- Visualizzazione dei propri libri in vendita e venduti (2.2.2.2);
- Modifica di un proprio libro in vendita (2.2.2.3);
- Marcatura di un proprio libro in vendita come venduto (2.2.2.4);
- Rimozione di un proprio libro dal sito (2.2.2.5);
- Visualizzazione dei propri dati di iscrizione al sito (2.2.2.6).

2.2.2.1 Inserimento di un nuovo libro

L'utente loggato entra nella pagina di inserimento libri clickando su *Inserisci* nella navbar. Nella pagina di inserimento l'utente può scegliere se scegliere di inserire:

- Una propria istanza di un libro già in catalogo: in questo modo, molto dei campi saranno già precompilati. Clicando sul menù a tendina, l'utente avrà a disposizione i titoli dei volumi selezionabili.
- Un libro non presente in catalogo: l'utente dovrà inserire più campi rispetto alla prima opzione, ma potrà inserire un libro a libera scelta.

In caso l'utente inserisca un campo dati non conforme al formato atteso (ex. ISBN con lunghezza diversa da 13, prezzo con tre cifre decimali), sarà di ciò dal sito, e verrà invitato a inserire i dati nel modo corretto. Se, invece, l'inserimento è avvenuto con successo, il sito informerà l'utente e lo reindirizzerà alla home.

2.2.2.2 Visualizzazione dei propri libri in vendita e venduti

L'utente loggato entra nella pagina di visualizzazione libri clickando su *I miei Libri* nella navbar. Da qui, l'utente può visualizzare i propri libri divisi fra *In vendita* e *Venduti*, e rappresentati tramite delle card simili a quelle presenti nelle schermate di ricerca libro [LINK](#)AGGIUNGERE [LINK](#), ma presentanti tre bottoni:

- Modifica (2.2.2.3);
- Venduto (2.2.2.4);
- Rimuovi (2.2.2.5).

L'utente inoltre, oltre alle informazioni che vedrebbe nella pagina di ricerca del libro, ha a disposizione la data di aggiunta del libro stesso.

2.2.2.3 Modifica di un proprio libro in vendita

Nella schermata di visualizzazione dei propri libri, l'utente preme il tasto *Modifica* sulla card di un libro in vendita. Viene quindi caricata la pagina di modifica del libro, molto simile a quella di inserimento. L'utente, quindi, modifica i campi a sua scelta, e poi conferma i cambiamenti apportati. Se tutto è andato bene, il sito reindirizza l'utente ai propri libri; in caso contrario, l'utente viene informato dell'errore e invitato a riprovare.

2.2.2.4 Marcatura di un proprio libro in vendita come venduto

Nella schermata di visualizzazione dei propri libri, l'utente preme il tasto *Venduto* sulla card di un libro in vendita. Ciò comporta lo spostamento del libro nella sezione *Libri Venduti* tramite un refresh della pagina. Un libro che viene segnato come venduto non può essere rimesso in vendita, e rimangono disponibili solo le seguenti operazioni:

- Visualizzazione del libro [LINK](#)INSERIRE [LINK](#);
- Rimozione del libro (2.2.2.5).

2.2.2.5 Rimozione di un proprio libro dal sito

Nella schermata di visualizzazione dei propri libri, l'utente preme il tasto *Rimuovi* sulla card di un libro. Il libro viene rimosso dal sito tramite un refresh della pagina, e non è più possibile reperirlo nella piattaforma.

2.2.2.6 Visualizzazione dei propri dati di iscrizione al sito

L'utente, una volta loggato, visualizza la propria immagine profilo nella navbar. Se invece, al momento dell'iscrizione, l'utente avesse scelto di non caricarne una, visualizzerà un'immagine generica. Per visualizzare il resto dei suoi dati, l'utente clicca il link *I miei dati* sulla navbar, e viene portato a una pagina riassuntiva con le sue informazioni.

2.2.3 Amministratore

L'amministratore è l'utente caratterizzato dall'indirizzo email fittizio `admin@admin.com`. Eredita tutti gli use case di *utente loggato*, e dispone di alcune funzionalità extra:

- Accesso al pannello amministratore (2.2.3.1);
- Inserimento libri in catalogo (2.2.3.2);
- Rimozione libri da catalogo (2.2.3.3);

2.2.3.1 Accesso al pannello amministratore

L'amministratore può accedere al pannello amministratore cliccando il link *Pannello Amministratore* sulla navbar. Mentre l'utente normale viene bloccato da un messaggio di errore, l'amministratore visualizza il pannello che consiste in:

- Un form per l'inserimento nei libri in catalogo;
- La lista dei libri in catalogo, con annessi id interno del database e bottone *Elimina*.

2.2.3.2 Inserimento libri in catalogo

Una volta nel pannello, l'amministratore può usare il form presente per inserire libri nel catalogo. Se inserisce dati validi, la pagina verrà aggiornata, e sarà possibile visualizzare a schermo la nuova entry; altrimenti verrà mostrato un messaggio d'errore.

2.2.3.3 Rimozione libri da catalogo

Una volta nel pannello, l'amministratore può decidere di rimuovere uno qualsiasi dei libri dal catalogo premendo il tasto *Elimina*. Dopo un aggiornamento automatico della pagina, tale libro non sarà più visibile. Da notare che tale operazione non elimina le istanze dei libri presenti nel database, che verranno invece disassociate dal libro in catalogo durante la procedura di eliminazione di quest'ultimo.

3 Progettazione

3.1 Obiettivi

Nello sviluppo del sito sono stati perseguiti alcuni obiettivi, elencati in seguito.

- **Separazione struttura-presentazione:** Questo obiettivo è senza ombra di dubbio da considerarsi il più importante, in quanto il suo raggiungimento comporta una maggior facilità nel soddisfare anche gli altri obiettivi. I contenuti del sito e la loro struttura devono essere separati dalla parte di presentazione grafica: si sono quindi definite le componenti di stile nei fogli CSS, mentre il contenuto è perlopiù gestito da PHP e HTML.
- **Accessibilità:** Il sito deve poter essere fruibile agevolmente dal maggior numero utenti possibile, compresi quelli con gravi disabilità visive e/o motorie. Per raggiungere tali obiettivi, sono state adottate varie misure, fra i quali citiamo le più significative:
 - TabIndex sui form;
 - Testo alternativo per le immagini;
 - Testi e link con buoni livelli di contrasto;
- **Flessibilità:** Il sito deve essere consultabile da varie tipologie di dispositivi, smartphone compresi. Deve, inoltre, essere adattabile a differenti dimensioni di schermo con il minor sforzo possibile.
- **Facilità d'uso:** Per rendere il sito maggiormente intuitivo per l'utente finale si è scelto di rispettare le convenzioni del web, antepoendo, quando possibile, queste ultime all'estetica del sito. Un esempio di tale scelta si può notare nei link: essi sono di colore blu sottolineato come l'utente è solito aspettarsi, e diventato viola se già visitati. Alcuni link, posti su sfondo azzurro, sono invece bianchi per avere un livello di contrasto sufficiente.

3.2 Layout

Per il sito è stato scelto un *layout a tre pannelli*, come illustrato in ???. Tale modello garantisce una superiore flessibilità e facilità di modifica in caso di aggiunta o rimozione di contenuti, e una maggiore adattabilità a taglie di schermo differenti. Ogni pagina del sito segue questo schema.

3.2.1 Resize e Mobile

Essendo tutte le unità di misura del sito definite in *em*, è possibile cambiare la dimensione del font alla base del sito per far scalare di conseguenza l'interfaccia, con buoni risultati.

Tuttavia, sono stati necessari alcuni accorgimenti per avere un'interfaccia mobile utilizzabile, il più radicale dei quali è stato inserire una navbar a scomparsa. Il risultato è visibile in ???. Ciò è stato ottenuto mediante l'inserimento di una

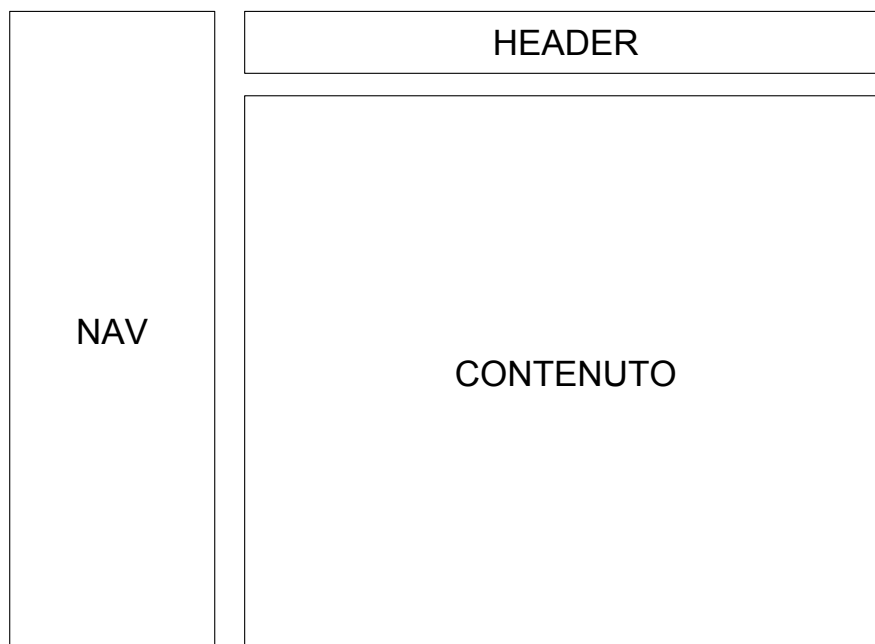


Figura 1: Layout di una generica pagina del sito

burger icon nell'header: un tap su tale icona apre la navbar, che si sovrappone al resto della pagina. Il menù si può quindi chiudere attraverso la x posta in alto a destra.

Altro dettaglio che cambia a seconda della dimensione della finestra è la visualizzazione dei risultati di ricerca di un libro, come rappresentato in ???. I risultati della ricerca vengono mostrati cercando di avere un buon compromesso fra leggibilità e necessità di scrolling.

3.3 Accessibilità

3.3.1 Trasformazione elegante

Il primo punto per progettare un sito web accessibile è garantire una trasformazione elegante delle pagine web. Per assicurarsi ciò si è innanzitutto separato la gli aspetti riguardanti contenuto, presentazione e struttura, come si è già discusso precedentemente. Successivamente sono state fornite equivalenti testuali per ogni media, permettendo anche agli utenti non vedenti di accedere alle informazioni attraverso l'udito. Inoltre le pagine sono responsive, poiché si adattano a seconda delle dimensioni dello schermo dell'utilizzatore, grazie al vasto utilizzo di dimensioni relative nei fogli di stile e grazie all'utilizzo di punti di rottura che ne facilitano la progettazione multi-piattaforma.

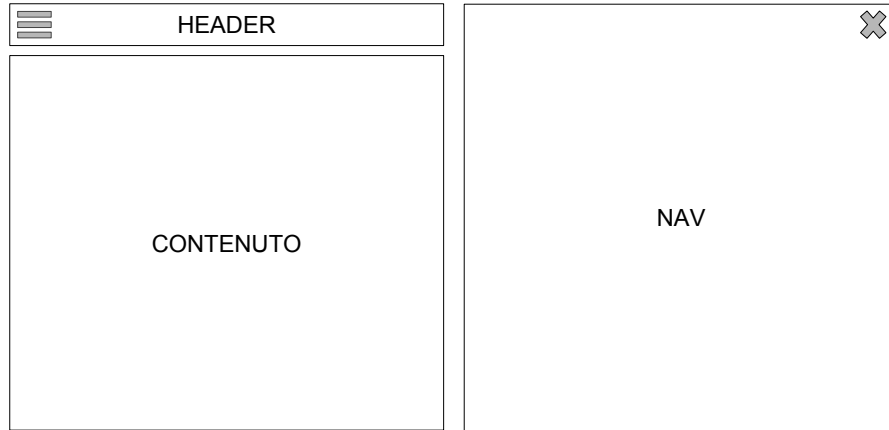


Figura 2: Layout di una generica pagina del sito in versione mobile, a sinistra con navbar nascosta, a destra con navbar aperta

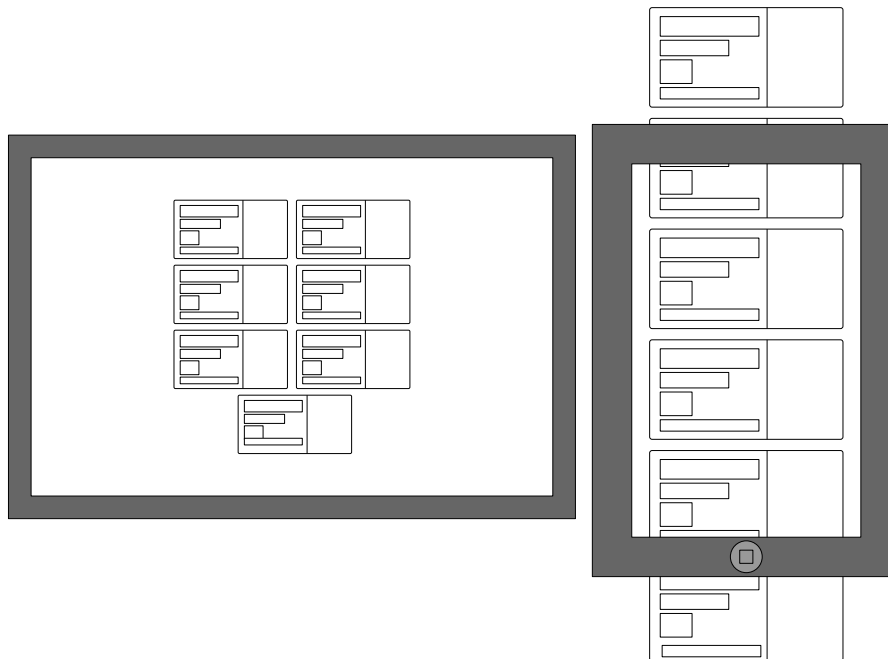


Figura 3: Layout dei risultati di ricerca desktop (sinistra) e mobile (destra)

3.3.2 Tastiera e scorciatoie

Per permettere agli utenti che utilizzano dispositivi differenti dal mouse e per garantire scorciatoie, in particolare modo alle categorie di utenti svantaggiate, sono stati utilizzati diversi strumenti:

- **Fieldset:** Tutti gli input in un form sono stati raggruppati in fieldset, ovviamente corredato da una leggenda opportuna;
- **Tabindex:** Per ogni form sono stati ridefiniti i tabindex, permettendo all'utente di spostarsi da un input al successivo, e stabilendo un opportuno ordine di tabulazione;
- **Accesskey:** Vi è una scorciatoia da tastiera che consente di accedere alla pagina "Chi siamo"; poiché non è possibile far conoscere le scorciatoie in anticipo, è stato inserito un indizio nel nav;

3.3.3 Schema colori

3.4 Schema organizzativo

4 Implementazione

4.1 Linguaggi e strumenti

4.1.1 HTML 5

Il gruppo ha utilizzato il linguaggio HTML5¹ per la struttura del sito, mantenendo comunque la compatibilità con XHTML². Questo permette al sito di funzionare sui browser più obsoleti e migliora la comprensibilità del codice. Per assicurare un codice corretto, sono state seguite le linee guida del corso di TecWeb e W3C³. Il codice è stato validato utilizzando il tool di validazione W3C⁴.

Le regole più importanti sono:

- **Chiusura tag:** ogni tag deve essere chiuso(<tag></tag oppure <tag/>);
- **Tag intestazione:** per ogni sezione (body, div, section) ogni tag di intestazione (h1,h2,...) deve partire sempre da h1;
- **Metatag:** nella sezione header, devono essere inseriti i metatag necessari per migliorare l'accessibilità verso i motori di ricerca. Questo permette al sito di avere una migliore visibilità in internet;
- **Separazione struttura presentazione comportamento:** il codice HTML non deve contenere CSS o script. Questi devono essere scritti in file separati e importati nell'header;
- **Struttura:** il codice HTML non deve sostituire il CSS, quindi non può contenere elementi per la presentazione;
- **Tabelle:** Le tabelle devono essere evitate quando possibile.

4.1.2 PHP

Il linguaggio di programmazione PHP è stato usato per il backend e per la composizione di codice HTML.

I file coinvolti nella parte di struttura sono:

- **htmlMaker.php:** contiene una serie di funzioni utili alla costruzione del sito:
 - **generateBookCollection** costruisce una collezione di libri. Prende come parametri *listaLibri* (array contenente una lista di libri) e *lista_bottoni* (opzionale. Rappresenta una lista di bottoni da aggiungere al libro). Questa funzione utilizza `singleItem` e `singleItemWithButtons` che generano un singolo libro;
 - **navbar** costruisce la navbar; questo è l'elemento più mutabile del sito, quindi viene costruito interamente attraverso del codice php;

¹<https://www.w3.org/TR/2017/REC-html52-20171214/>

²<https://www.w3.org/TR/2015/REC-xhtml-rdfa-20150317/>

³<https://www.w3.org/standards/webdesign/htmlcss>

⁴https://validator.w3.org/#validate_by_uri

- **pagina_messaggio** costruisce una pagina molto semplice, usato principalmente per pagine di errore. Prende come parametri titolo, sottotitolo ed extra (opzionale).
- **"nomepagina".php**: (esempio: home.php, login.php) sono i file che costruiscono le pagine del sito. Utilizzano le funzioni di `htmlMaker` e il rispettivo file `html` (es: home.php usa `home.html`) che contiene l'effettivo `html`;

Il gruppo ha deciso di utilizzare codice PHP rivolto alla struttura del sito, perché alcuni contenuti strutturali del sito cambiano se l'utente è sconosciuto oppure ha effettuato l'accesso. Le differenze sono:

- I bottoni per effettuare il login vengono sostituiti dal bottone di logout;
- Alla navbar laterale vengono aggiunti l'immagine di profilo dell'utente e i link per accedere alla configurazione del suo account;
- Alla navbar laterale viene aggiunto il pulsante (inserisci) per la vendita di un libro;

L'utilizzo del PHP per la struttura ha permesso anche le seguenti caratteristiche:

- **Unico header, navbar e footer**: L'header, la navbar e il footer sono gli stessi per ogni pagina. Questo permette una maggiore manutenibilità del sito, in quanto una modifica a un file si riflette su tutte le pagine del sito web;
- **Link circolari**: Un solo codice per tutti i file genera dei link circolari, che vengono rimossi tramite codice PHP (esempio: per rimuovere i link circolari dalla home si utilizza questa funzione: `str_replace('', '<a>', " link alla pagina home ")`);
- **Riempire i campi in caso di errore**: se prova a inserire un libro ma ci sono degli errori (che passano i controlli JavaScript), tramite PHP vengono riempiti i campi corretti, permettendo all'utente di non doverli inserire nuovamente;
- **Costruzione card dei libri**: le pagine che contengono dei libri cambiano a seconda dei libri presenti nel database. Per questo motivo, il file `htmlMaker.php` contiene le funzioni `singleItem` e `singleItemWithButtons` per costruire le card dei libri, a partire da un risultato SQL.

La parte di backend gestisce principalmente le richieste di modifica al database da parte degli utenti del sito:

- **sql_wrapper**: il file `sql_wrapper` contiene delle funzioni per accedere al database; lo scopo di questo file è semplificare l'accesso al database per gli altri;
- **Login**: l'accesso avviene attraverso il file `check_login.php`, che, tramite l'`sql_wrapper`, controlla se l'email e la password esistono. In caso positivo viene aperta la sessione;
- **Registrazione**: la registrazione avviene attraverso il file `new_user.php`. Se la registrazione ha successo, viene fatto il login automatico;

- **Validazione campi:** il file *validator.php* contiene delle funzioni utili alla validazione dei campi in registrazione, login, inserimento libri;
- **Sessione:** il file *back_end.php* contiene funzioni per capire se la sessione è aperta (quindi se il login è stato effettuato).

4.1.3 SQL

Sql è stato utilizzato per la codifica del database. Il database è composto dalle seguenti tabelle:

- **Utente:** contiene le credenziali appartenenti a un utente registrato al sito. Per proteggere la password, viene codificata in *Pw_Hash* tramite la funzione *password_hash* di php. Il *codice_identificativo* del libro è un numero che incrementa ad ogni libro inserito;
- **Libri_Listati:** contiene i libri listati (catalogo);
- **Libri_In_Vendita:** contiene i libri che gli utenti mettono in vendita. Viene controllato che il *prezzo* sia positivo, che lo *stato* sia "Venduto", "In Vendita" o "Prenotato" e che il *tipo* sia "Libro", "Slide", "Appunti", "Altro" o "Dispense";

4.1.4 JavaScript

Il linguaggio di programmazione Javascript è stato utilizzato per la parte di front end del sito.

Per ridurre il traffico di file, tutte le funzioni sono inserite in *action.js*. JS è stato utilizzato per:

- convalidare i dati nella registrazione, durante l'inserimento di libri in vendita e mentre l'utente effettua il login;
- configurare bottone per chiudere e aprire la navbar laterale, assieme al CSS.

4.2 Funzionamento generale

Ogni pagina è formata da tre elementi principali:

- **header:** contiene il logo, il titolo e i bottoni *accedi* e *registrati* per un utente non autenticato, *Logout* per un utente autenticato;
- **nav:** contiene i link per la navigazione e il footer. L'utente autenticato ha anche l'immagine di profilo e i link per il pannello del controllo. Viene generata da *htmlMaker* (vedere ??);
- **section:** contiene il contenuto principale del sito. Cambia per ogni pagina. Per la maggior parte dei casi, il contenuto di section corrisponde al contenuto della corrispondente pagina ".html" (es. section della Home è il contenuto di *home.html*). Il contenuto può essere diverso, se dipende dai

dati inseriti nel database (card dei libri o i dati dell'utente nel pannello utente).

Il catalogo e i libri in vendita sono una unordered list (`<ul class='books_collection'>`), creata da `generateBookCollection` come definito in `??`. Ogni libro è un elemento (`<li class="search_item">`) con l'id uguale all'id del libro nel database (`md5_Hash`). Gli altri elementi del libro sono un div contenete le informazioni dello stesso e l'immagine (se presente).

L'inserimento di un libro in vendita è permesso solamente a un utente registrato. L'utente può scegliere se il libro è nel catalogo (quindi i campi Titolo, Autore, Casa editrice, Corso non sono richiesti) oppure se non è presente. L'immagine del libro non è obbligatoria, se non presente viene mostrato un placeholder.

4.3 Separazione struttura-contenuto

Il gruppo ha considerato come fondamentale la separazione tra la struttura e il contenuto. Per questo fine, si è posto le seguenti regole:

- **JavaScript non intrusivo:** tutto il codice JS è stato inserito in un file separato dai file Html e Php (*action.js*), evitando quindi l'utilizzo di elementi come `<div "onClick="funzione()"/>`;
- **PHP non intrusivo:** anche se una parte del codice PHP genera codice HTML, la pagina finale non contiene elementi PHP. Inoltre, tutto il codice HTML che non proviene da un file viene scritto attraverso il comando `"echo"`;
- **CSS non intrusivo:** come per JavaScript, anche tutto il codice CSS è stato inserito in un file dedicato (*style.css*).
- **HTML solo per la struttura:** come accennato in `??`, l'HTML non può contenere elementi di stile. L'HTML, quindi, **non** deve essere utilizzato in questi casi:
 - usare una tabella per il layout (usare invece un layout responsive attraverso CSS);
 - definire la dimensione, colore e stile generale del testo (usare invece classi e tagname corretti, per poi essere modificati con il CSS);
 - usare le intestazioni (`h1,h2,h3...`) solo per definire la dimensione del testo;
 - usare il `
` per andare a capo;

style.css e *action.js* sono state importate nell'*head*.

5 Validazione

5.1 Strumenti usati (da espandere)

6 Organizzazione del lavoro