

CMPS 350 Web Development - Assignment 3

Deadline Sunday March 24 @ 11:59 PM

In this assignment, you will create **LibraryLocus** app that allows users to manage their favorite books. You will use the **Client JS**, **Local Storage**, and **an external API** to build this app. The app should have the following features. To understand more about the functionality of the app, watch this **Video**.

1. **List Books** : When the **app first loads**, it should make a request the following **external API** to fetch an **initial list of books**, displaying these books in a **grid format**. Each book's presentation should include its **cover image, title, authors, and a brief description**, as illustrated in Figure 1. **Additionally, this list of books should be stored in Local Storage** to facilitate offline access for the user. On subsequent app launches, the **app should first check Local Storage** for the presence of the books list. If the **list is already stored**, the app should **use this local copy** instead of making another API call, thereby optimizing performance and enhancing user experience by leveraging locally available data.

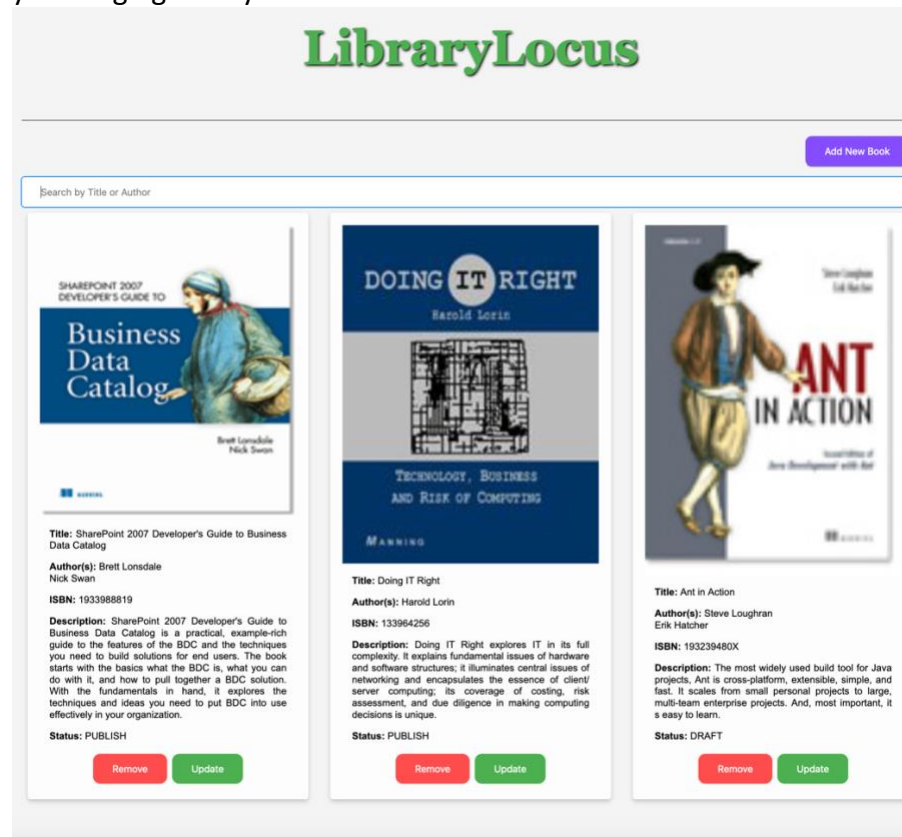


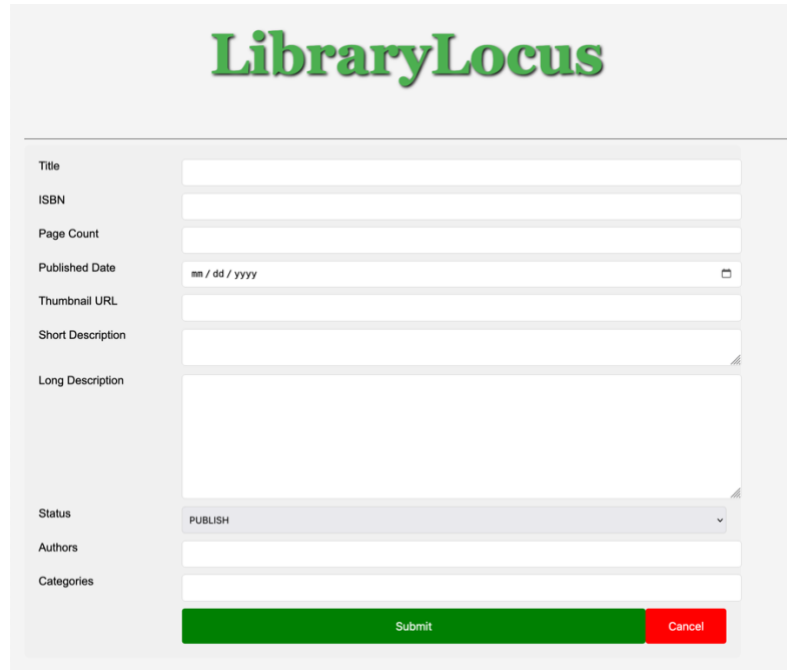
Figure 1 List Books

2. **Add Book**: When the user clicks on the **add new book button**, the app should **show** the **Add Book Form** that allows the user to add a new book as shown in figure 2. Once they **click** on the **Add button**, the book should be saved to the **Local Storage**, and the app should **hide** the form from the page and **show** the user the newly added book.

add the book to the first index in local storage.

SHOULD BE VALIDATED. ✓

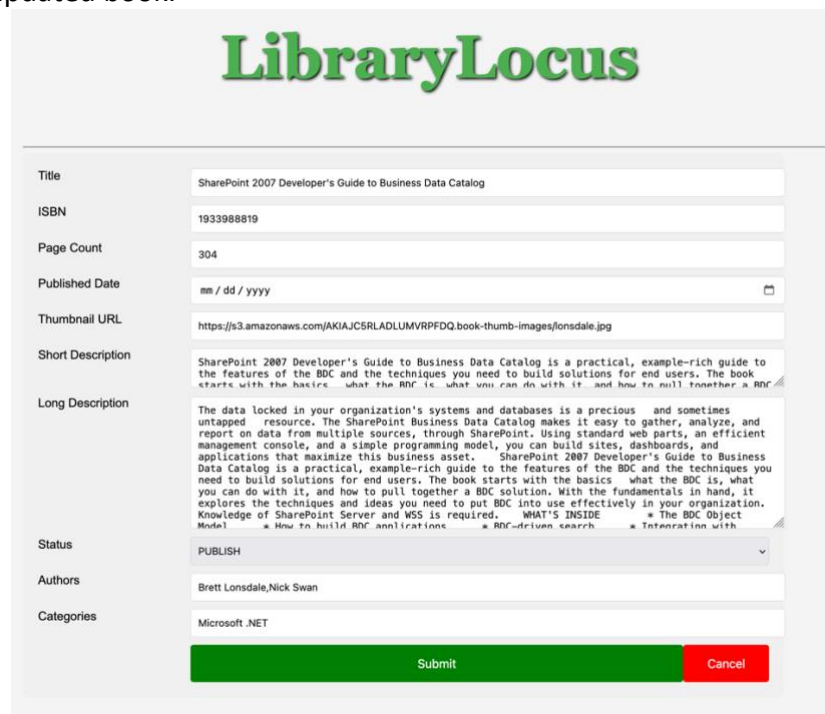
CHECK EVERYTHING AGAINST THE VIDEO LATER



The image shows the 'Add Book' form in the LibraryLocus application. The form is titled 'LibraryLocus' in a large green font at the top. Below the title, there is a form with several input fields: Title, ISBN, Page Count, Published Date (with a date picker icon), Thumbnail URL, Short Description, and Long Description. There is also a Status dropdown menu set to 'PUBLISH', and fields for Authors and Categories. At the bottom of the form are two buttons: a green 'Submit' button and a red 'Cancel' button. A large blue checkmark is drawn to the right of the form.

Figure 2 Add Book

- 3. Update Book:** If the user clicks on the update button, the app should open the *Update Book Form* and load the details of the selected book into the form to allow the user to edit the book details (see Figure 3). Once they click on the submit button, the book should be updated in the Local Storage, and the app should show the user the book page where they can see the updated book.



The image shows the 'Update Book' form in the LibraryLocus application. The form is titled 'LibraryLocus' in a large green font at the top. Below the title, there is a form with several input fields: Title, ISBN, Page Count, Published Date (with a date picker icon), Thumbnail URL, Short Description, and Long Description. There is also a Status dropdown menu set to 'PUBLISH', and fields for Authors and Categories. At the bottom of the form are two buttons: a green 'Submit' button and a red 'Cancel' button. The form is pre-filled with data for the book 'SharePoint 2007 Developer's Guide to Business Data Catalog'.

Figure 3 Update Book

4. **Search and Filter:** Develop a feature that enables users to search the book collection by either title or author. As users enter their search query into the search bar, the system should dynamically filter and display books that match the query in the author's name or book title. ✓

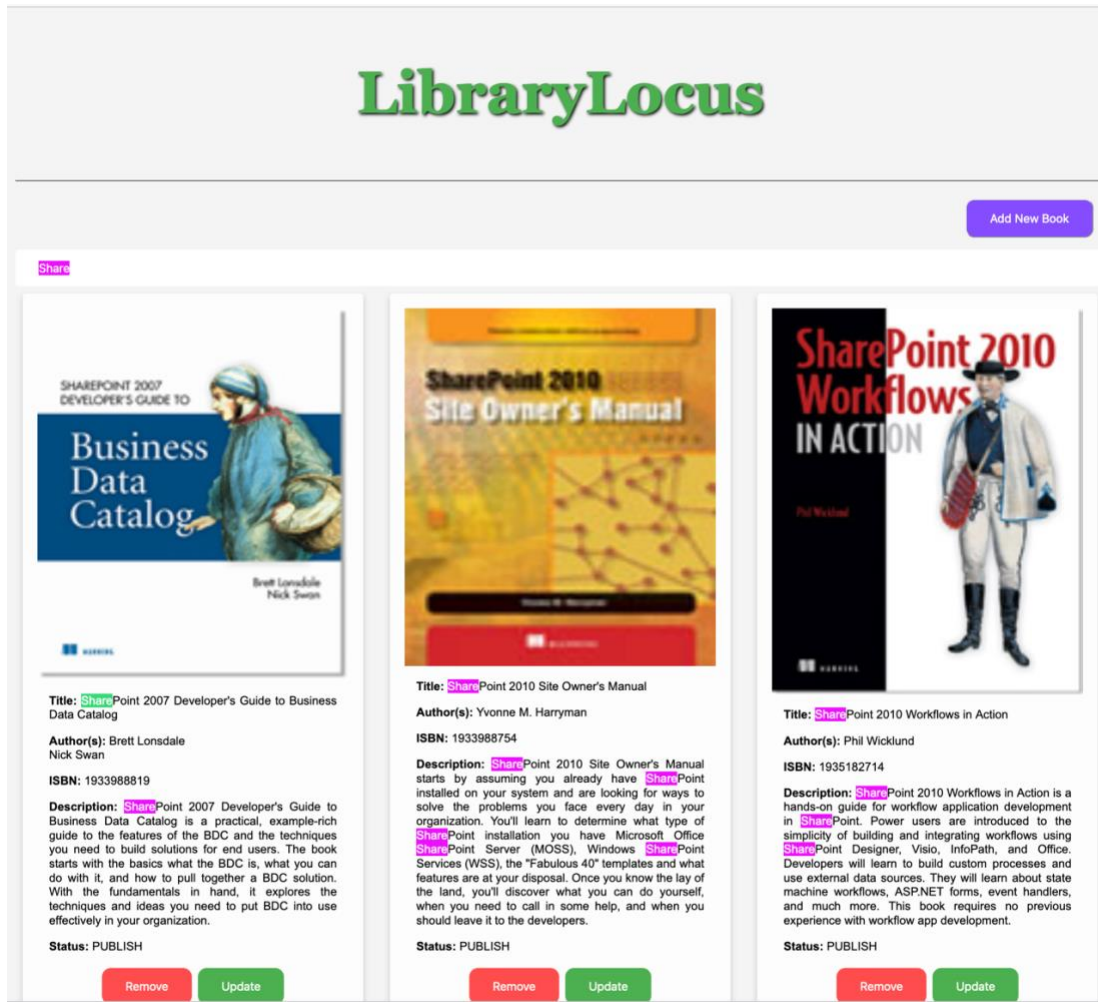


Figure 4 Search and Filter

5. **Delete Book:** If the user clicks on the delete button, you should delete the selected book from the Local Storage and from the books page (i.e., the user should not be able to see the deleted book in the book grid view anymore).

After you complete the Assignment, fill in the **TestingDoc-Grading-Sheet.docx** and save it inside the **assignment3** folder. Push your work to your GitHub repository.