

CMPS 350 Web Development Fundamentals - Assignment # 5

Due Date Saturday, May 27, 2024, @11: 59 PM

Overview:

In Assignment 4, you developed the LibraryLocus app using JSON files for backend data management and a Web API for client-server interaction. This assignment progresses to enhance data management and user interface techniques by introducing a SQLite database with Prisma and a Next.js based front end.

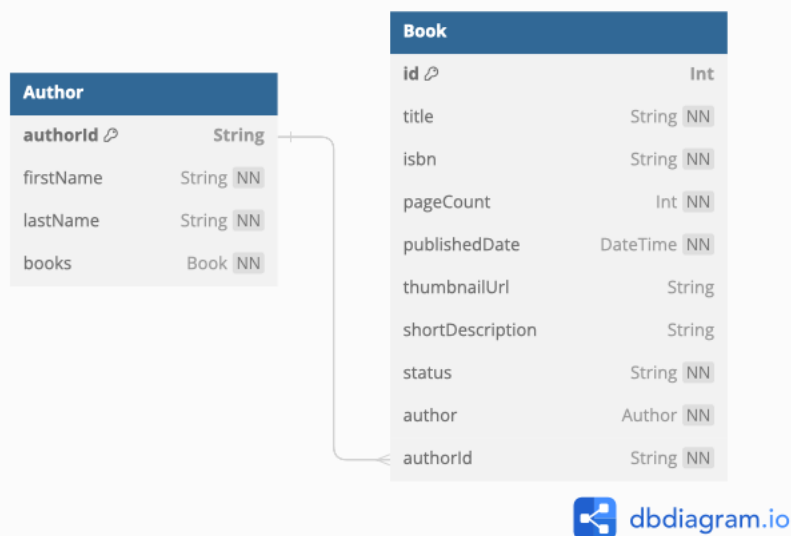
Objective: Transition the LibraryLocus app backend from JSON file handling to a SQLite database managed with Prisma. Replace the basic HTML/CSS frontend in the public folder with a more dynamic and user-friendly interface using Next.js and React. The interaction with the server-side could use **fetch** or **Server Actions**.

Setup Instructions:

1. Sync the **cmpps350-lab** repository to download the Assignment 5 files.
2. Navigate to the **assignment5/librarylocus** app project folder, containing a base implementation modified from Assignment 4.
3. Execute **npm install** to install required packages.
4. Launch the application with **npm run dev** and open **http://localhost:[PORT]/index.html** to view the demo.

Part A - Implementing the Database (40%)

- **Task:** Utilize Prisma to define the data model and handle database operations within **books-repo.js**. The entity relationship diagram of the database is given below.



- **Relationships:**
 - An author can have one or more books (**one-to-many**).
 - Each book is authored by exactly one author (**many-to-one**) [for simplicity].

- **Requirements:**
 - Rewrite the functionalities of the repository, which previously relied on a file-based system, to now utilize Prisma methods.
 - **Key Functions:**
 - **addBook():** Add a new book entry.
 - **updateBook(isbn):** Update an existing book by ISBN.
 - **deleteBook(identifier):** Remove a book using its ID or ISBN.
 - **getBook(name):** Fetch a book by name, handling non-existence.
 - **getBookByISBN(isbn):** Get a book by ISBN.
 - **getBooksByPageCount(minPages):** Retrieve books with at least **minPages**.
 - **getBooksByAuthor(author):** List books (co-authored included) by an author.
 - **getBooksByCategory(category):** Get books within a specified category.
 - **getAuthorsBookCount():** Generate a summary of books per author.
 - Create seed.js file that will allow you to populate the database with the given two json files of book and authors.

Part B - Implementing the UI using Next.js and React (60%)

- **Task:** Develop a dynamic web interface to replace the existing static HTML/CSS.
- **Functionalities:**
 - Implement and enhance the user interface to support **listing, adding, deleting, and updating** book entries.
 - Provide a responsive layout for displaying book **details, summaries, and interactive filtering** options.
- **UI Components:**
 - **List View:** Display books in a card format with options to filter by various attributes. [See figure 1]
 - **Detail View:** Offer a detailed view of book attributes. [see Figure 2]
 - **Add/Update Forms:** Enable forms to add new books and update existing ones. [see Figure 3 and 4]
 - **Summary View:** Show a summary view of book data as per requirements.[See Figure 5]
- **Testing:** Ensure all functionalities are operational as per the demo views shown below.

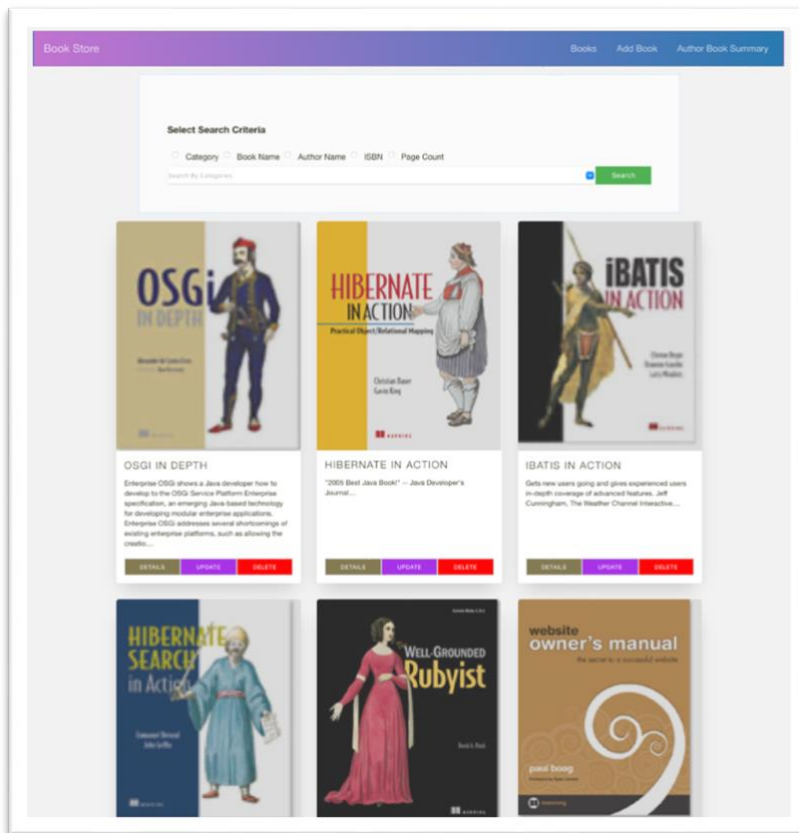


Figure 1 List View

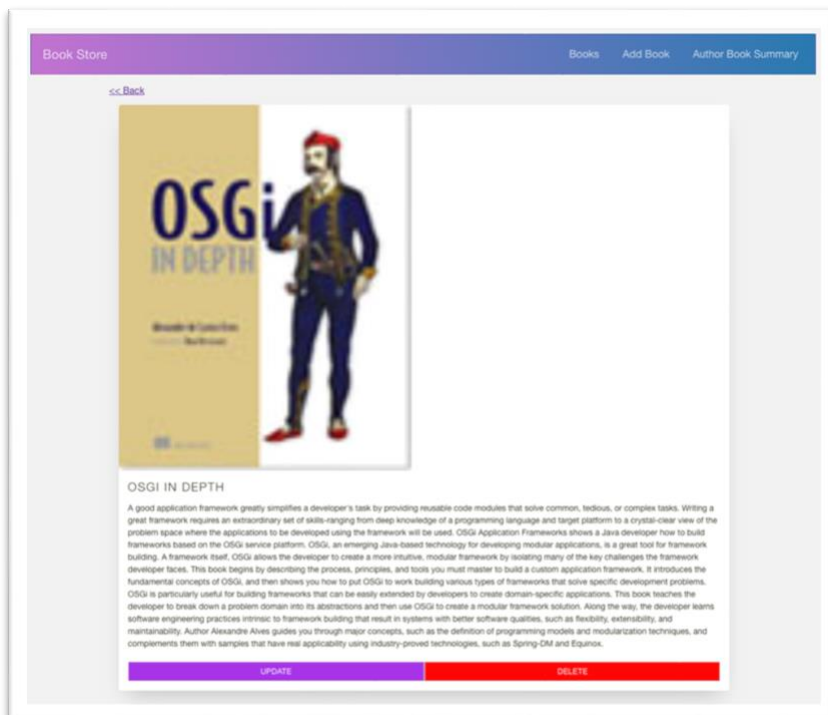


Figure 2 Detail View

Book Store

BooksAdd BookAuthor Book Summary

Add Book

Title

ISBN

isbn

Page Count

page count

Published Date

04/28/2024

thumbnailUrl

Short Description

Long Description

Book Status

REVIEW

Authors

Separate them by Comma

Categories

Separate them by Comma

SUBMIT

Figure 3 Add Book

Book Store

BooksAdd BookAuthor Book Summary

Edit Book

Title

OSGi in Depth

ISBN

isbn

Page Count

325

Published Date

12/12/2011

thumbnailUrl

https://s3.amazonaws.com/AKIAJC5RLADLUMVRFDQ/book-thumb-images/alves.jpg

Short Description

Enterprise OSGi shows a Java developer how to develop to the OSGi Service Platform Enterprise specification, an emerging Java-based technology for developing modular

Long Description

A good application framework greatly simplifies a developer's task by providing reusable code modules that solve common, tedious, or complex tasks. Writing a great framework

Book Status

PUBLISH

Authors

Alexandre de Castro Alves

Categories

Java

SUBMIT

Figure 4 Update Book

Book Store		Books	Add Book	Author Book Summary
Author Name		No Of Books		
Christopher Allen		3		
Shannon Appelcline		3		
Erik Hatcher		3		
Tim Hatton		3		
Daniel Minoli		3		
Jeffrey Palermo		3		
Jimmy Bogard		3		
Daniele Bochicchio		2		
Stefano Mostarda		2		
Marco De Sanctis		2		
Rob Crowther		2		
Steve Loughran		2		
Brandon Trebitowski		2		
Jon Skeet		2		
Bruno Lowagie		2		
Andrew Schmidt		2		
Ted Neward		2		
Kalen Delaney		2		
Greg Low		2		
Adam Machanic		2		
Paul S. Randal		2		
Kimberly L. Tripp		2		
Ben Scheirman		2		

Figure 5 Show Summary