

CMPS 350 – Web Development Fundamentals

Practical Final Exams

- The exam duration is 120 minutes. So, read the exam questions carefully and plan your time accordingly.
- Push your code to GitHub regularly (at least every 30 minutes) to avoid unpleasant surprises, as your computer might hang!
- The Exam is an open book. In case of plagiarism, both parties will receive 0 points. Hence do not share or receive any code from anyone.
- Once you complete the Exam, you should:
 - Add a screenshot for each question to the provided testing sheet.
 - Push your code and testing sheet to your GitHub repo under the **final** subfolder.
 - Demo your work before leaving the Exam.

Library App

In this Exam, you will utilize your front-end and back-end web development skills to create a library web application. The application will utilize Next JS, React, and Prisma to display a list of books. Before beginning the Exam, a demo of the app's functionality will be presented, and the UI design can be referenced in the accompanying figures.

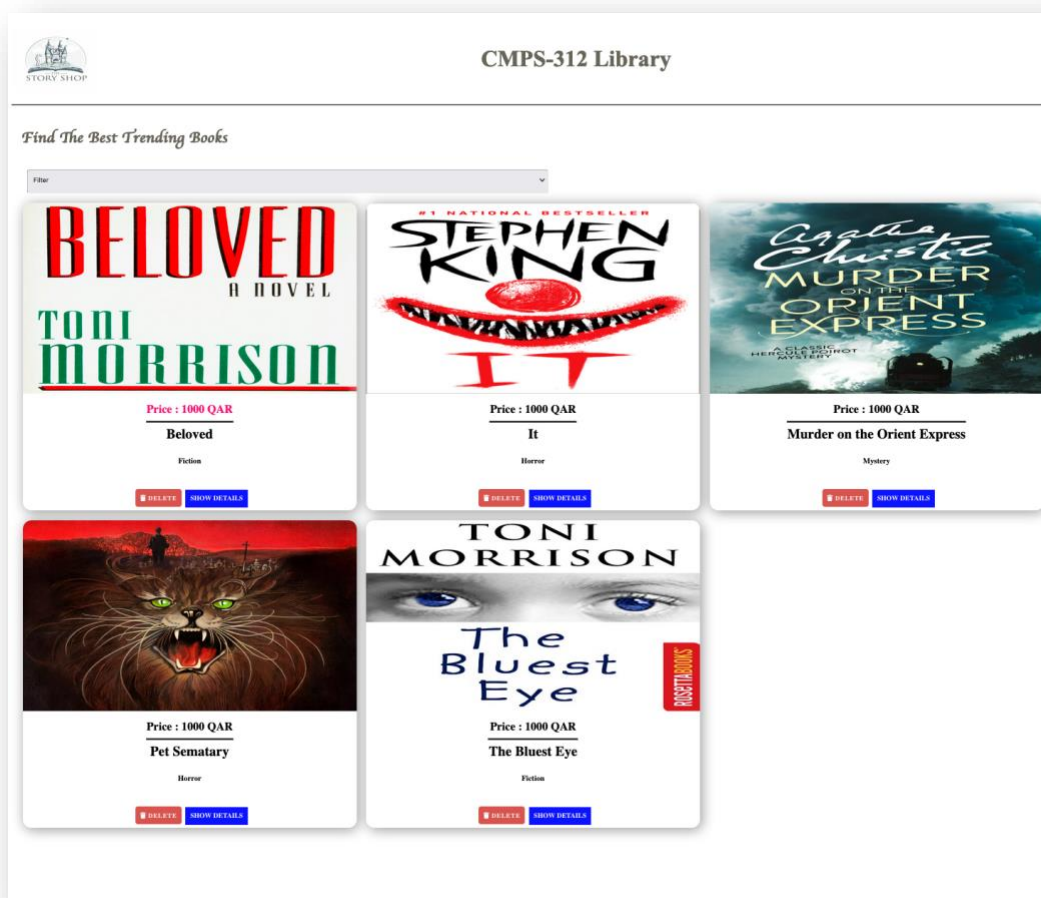
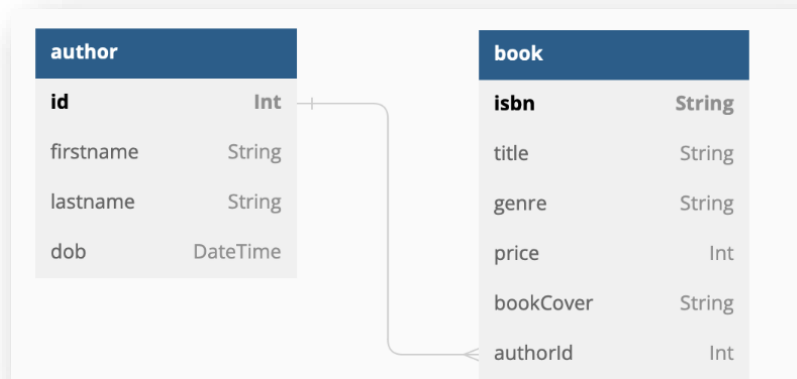


Figure 1 Library App

1. [35 pts] Creating Database Model and Repository

1.1. Create the data models shown below using Prisma Schema[5 pts]



1.2. Create the library-repo.js under **api/books** file that contains the following methods [30 pts]

Method	Description
getBooks()	Return all books sorted by title in ascending.
searchBook(title)	Return all the books that contain the given title
addBook(book)	Adds new book to the database
deleteBook(isbn)	Deletes a book by isbn
addAuthor(author)	Adds new author to authors table
getBook(isbn)	Return the book that has the same ISBN . This method also should include all the information of the author. <pre> { "isbn": "cjbv8yehd4000000000000000002", "title": "And Then There Were None", "genre": "Mystery", "price": 8.99, "bookCover": "https://m.media-amazon.com/images/I/81B9LhCS2AL._AC_UF1000,1000_QL80_.jpg", "authorId": 1, "author": { "id": 1, "firstname": "Agatha", "lastname": "Christie", "dob": "1890-09-15T00:00:00.000Z" } } </pre>

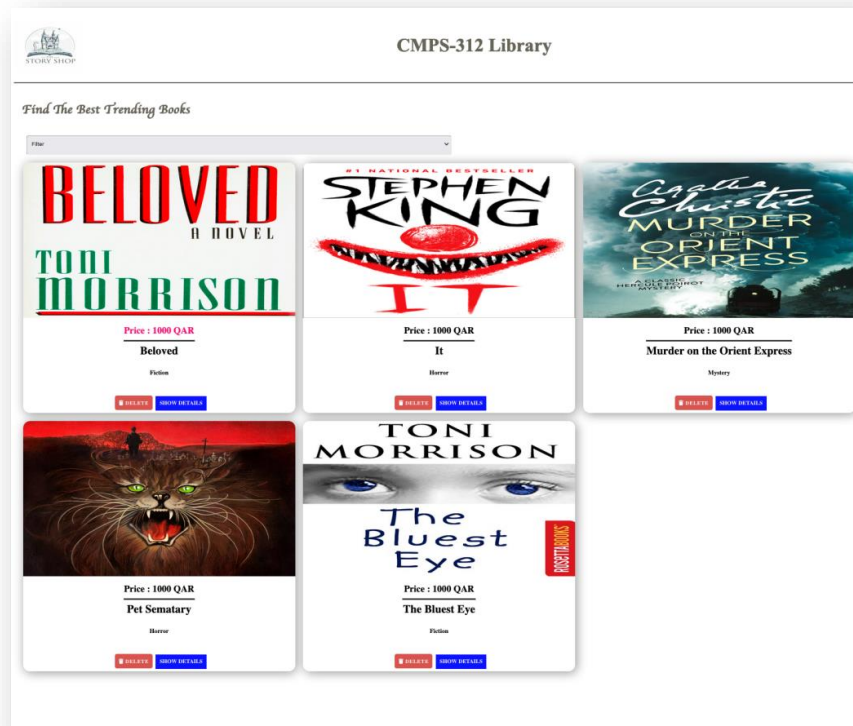
2. Create the Library App API Routes [25 pts]

HTTP Verb	URL	Functionality
GET	/api/books	return all books sorted by title
POST	/api/books	Adds a new book
GET	/api/books/:isbn	Return a book by isbn , that contains the author info as well.
DELETE	/api/books/:isbn	Deletes a book by isbn
POST	/api/authors	Adds a new Author

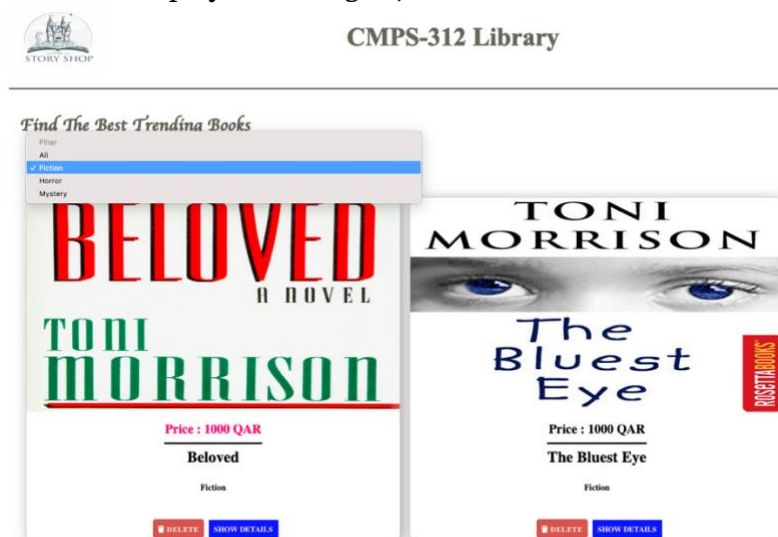
Test your routes/repository using Postman before moving to the next step

3. [45 pts] Creating the Library App User Interface [5 Bonus]

- 3.1. [15 pts] **Create the Books List Page:** When the root page loads, retrieve books from the server and display them in a grid of cards as shown in the image below. The book card should display the Book Cover Image, Book Title, Genre, and Price.



- 3.2. [10 pts] **Delete Book:** within the book card, add a "Delete" button that allows user to delete a given book. As soon as the user deletes the book you remove the book from the database and reload the books list.
- 3.3. [10 pts] **Filter by Genre:** add a filter by drop down that allows the user to filter the books dynamically by the selected genre. The book cards that match the search should be displayed in the grid, while the others are hidden.



- 3.4. **[10 pts] Get Book Details:** within the book card, add a "Show Details" button that allows user to see the complete details of the books. As soon as the user clicks you should load the Book Details page display the book including the information of the author.

