# PROJECT FOR CS415 –DB-202108

## A Database-driven Web Application Design and Development Project – City Library

Here is the requirements specification/guide for the City Library website:

*(Note: Upon code completion, you are required to push your finished project to a repo on github and then make a submission to the 'Project Delivery' Assignment item on Sakai, by 6pm Thursday, Aug/12/2021)*

Assume you have been hired by the City of Fairfield to design, develop and deliver a Website for the CityLibrary, which they intend to use to manage the inventory/list of books in their collection. Each Book is published by a Publisher. And the Publishers can publish several Books.

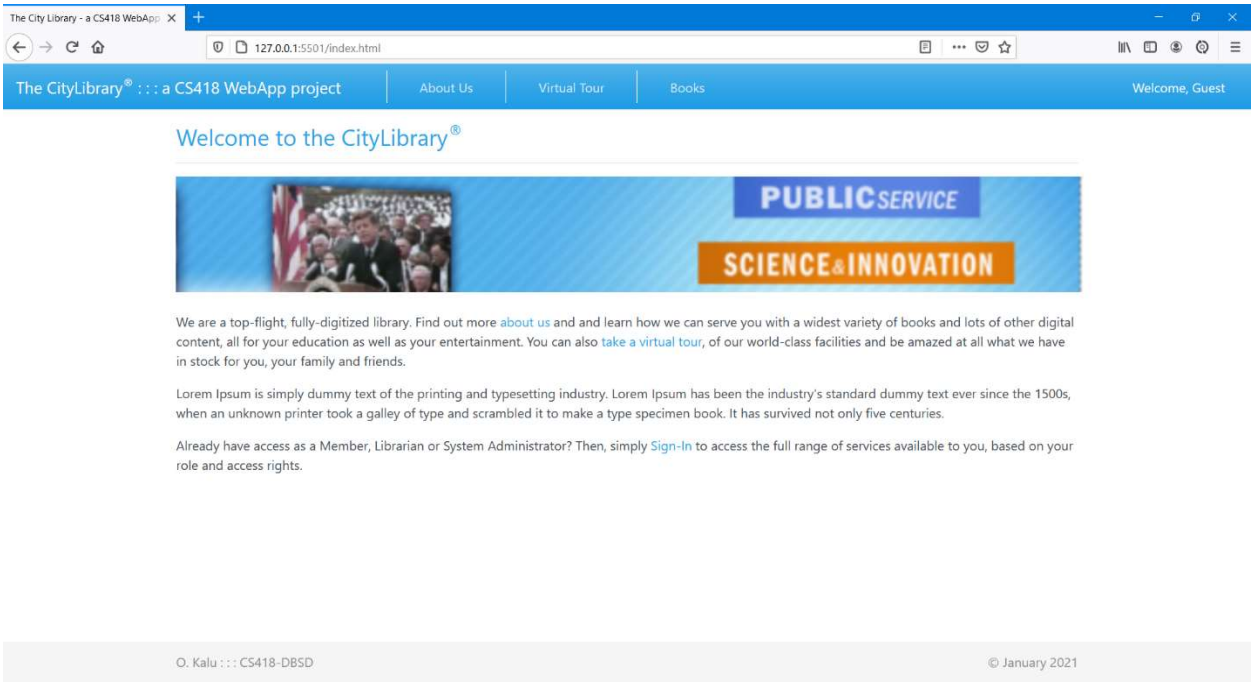
To undertake this Project task, you can form/work in a project team of up to 3 OR if you prefer, you may work by yourself, individually, as a sole Project Owner. You are expected to apply the various Database design and Web Application development methods, tools and technologies, that you have learnt in your MSD course, so far. For the database, you may choose either a Relational (e.g. MySQL) or NoSQL (e.g. MongoDB) option.

**PHASE 1:**

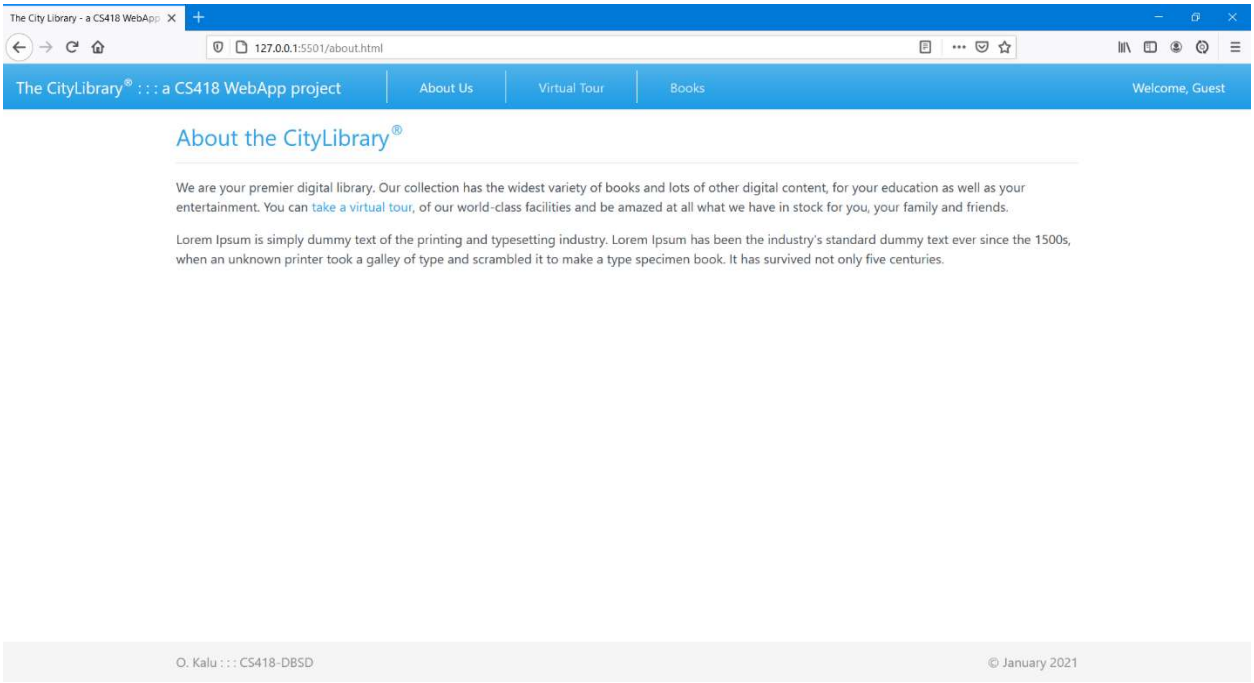
Here are the use-cases, with sample screenshots. Note: Your UI design does NOT necessarily have to look like the sample shown here. However, the functionality described is what is required.

1. You may download and use the image files provided for you in the project's resources folder (found on both Sakai and MS Teams) or you are free to create/obtain you own images.
2. Implement code for the CityLibrary Homepage, About page and Virtual Tour page:
    a. The figures below are example screenshots of what the pages may look like. **Note**: You may use a CSS framework (such as Bootstrap) to apply necessary styling, to achieve this sample or similar look and page layout:

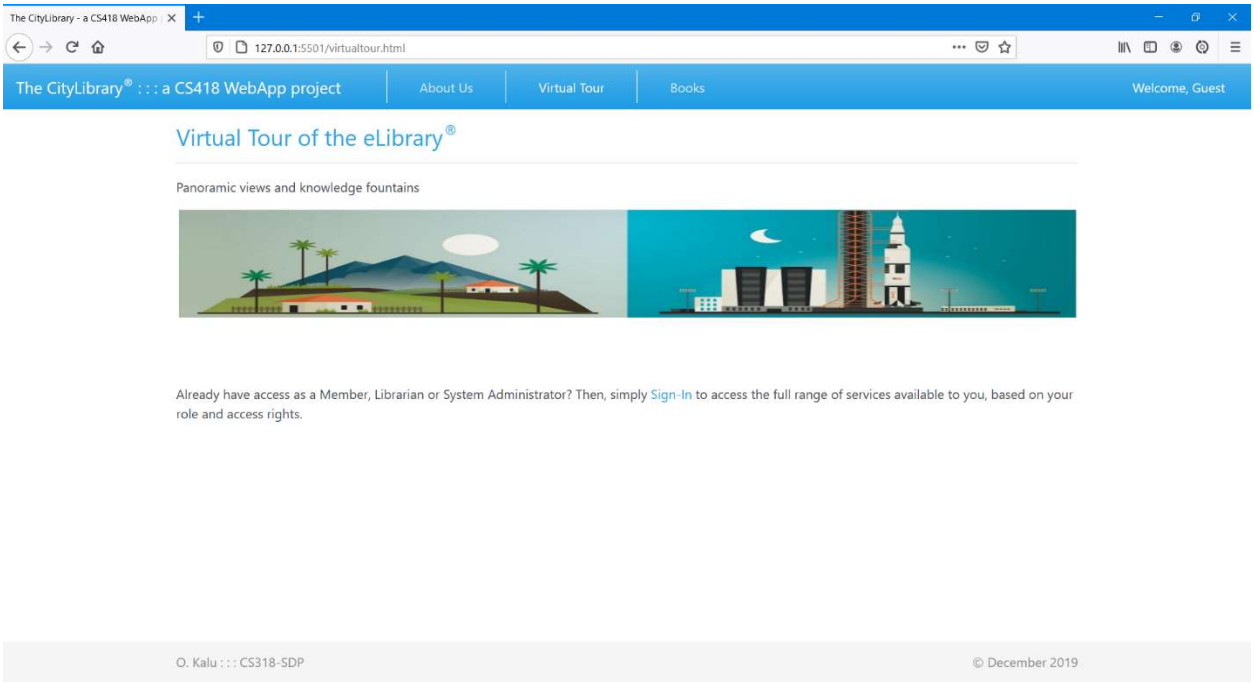
Home page:




About page:




# Virtual Tour page:

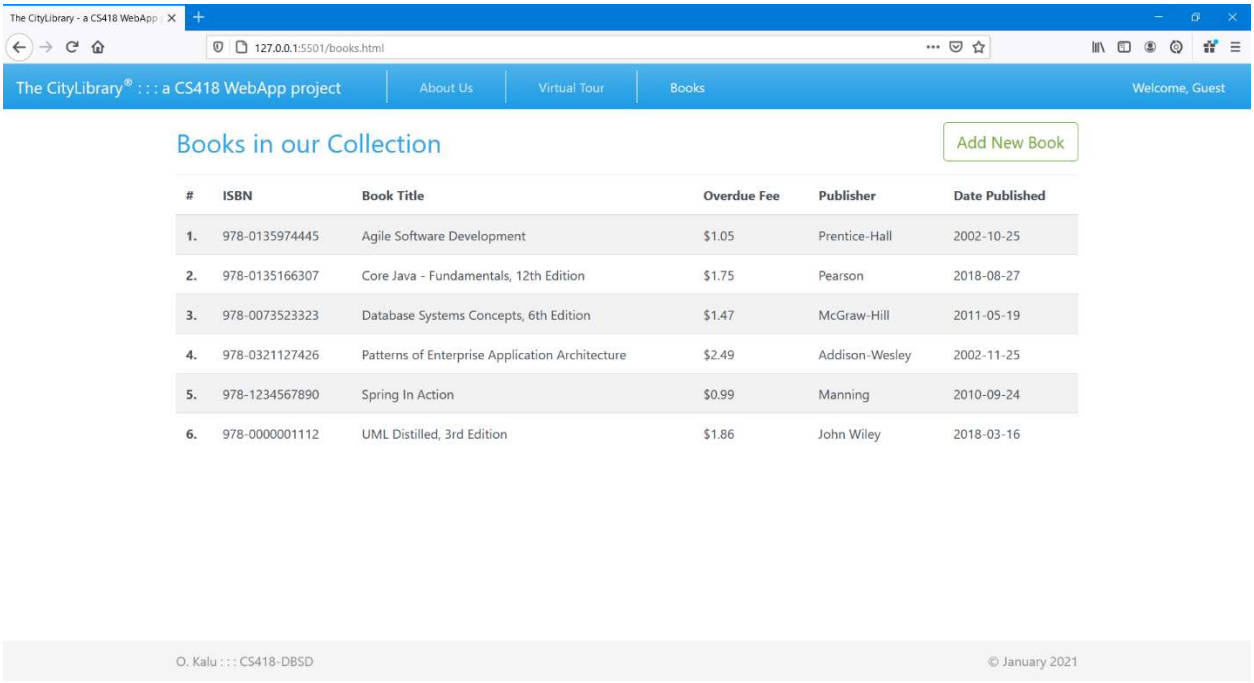


3. Implement code for the CityLibrary Books page:
   Note: The Books data should be stored and loaded from a database. You may design and build/consume a RESTful service API (i.e. Web API) for this. Example url - http://localhost:3000/citylibrary/api/book/list.

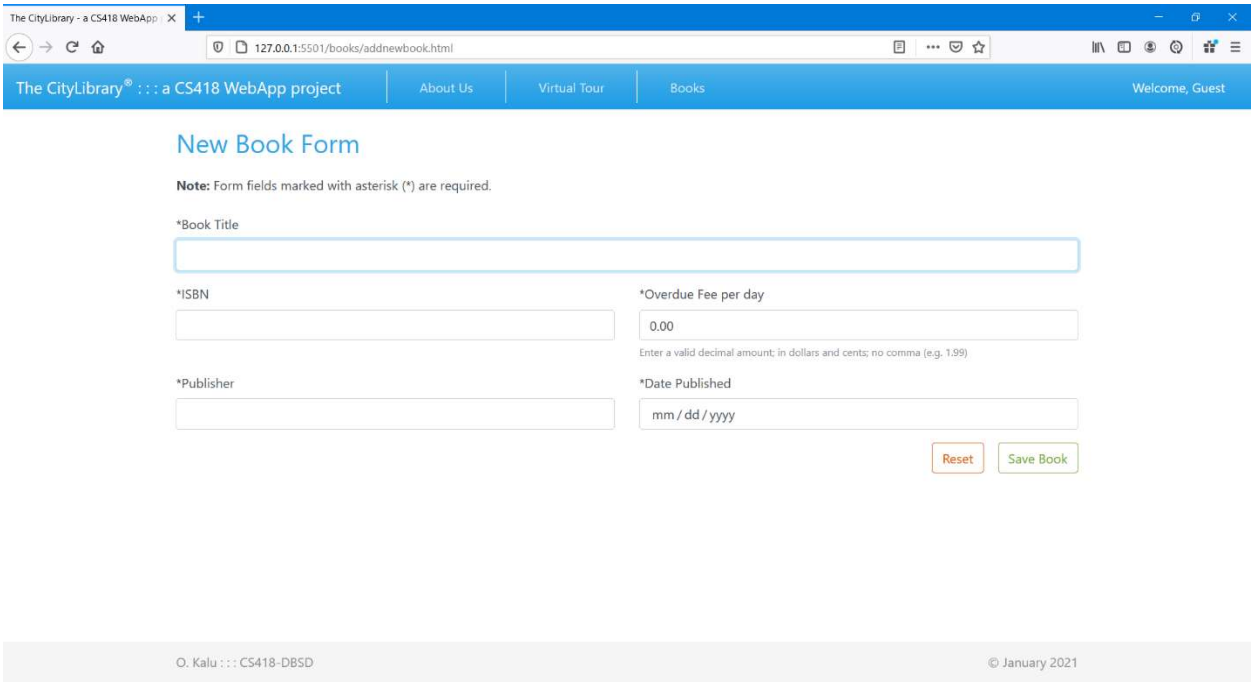
   Have your front-end webapp fetch and present the data as shown below:

   a. When the Books page is loaded in the Browser window, the books data is fetched from the RESTful service API and presented as shown below:

Books page:



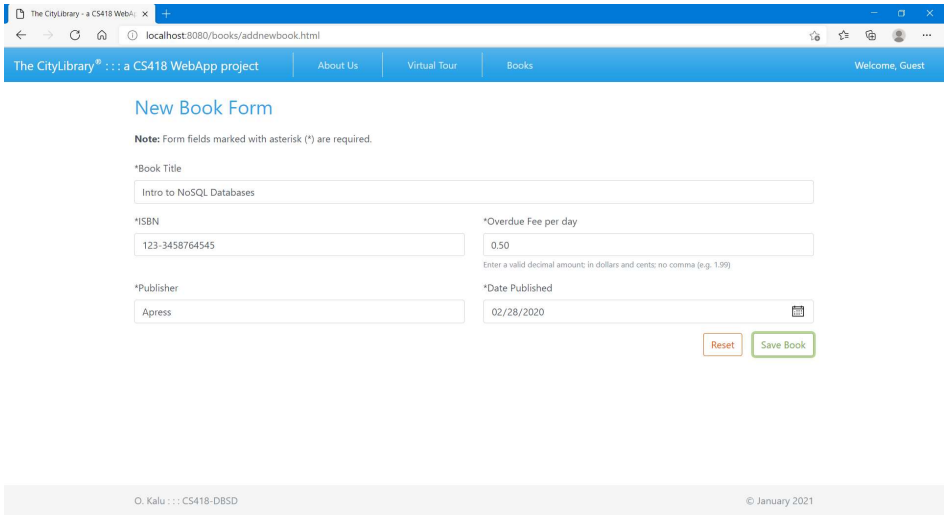

4. Adding a new Book

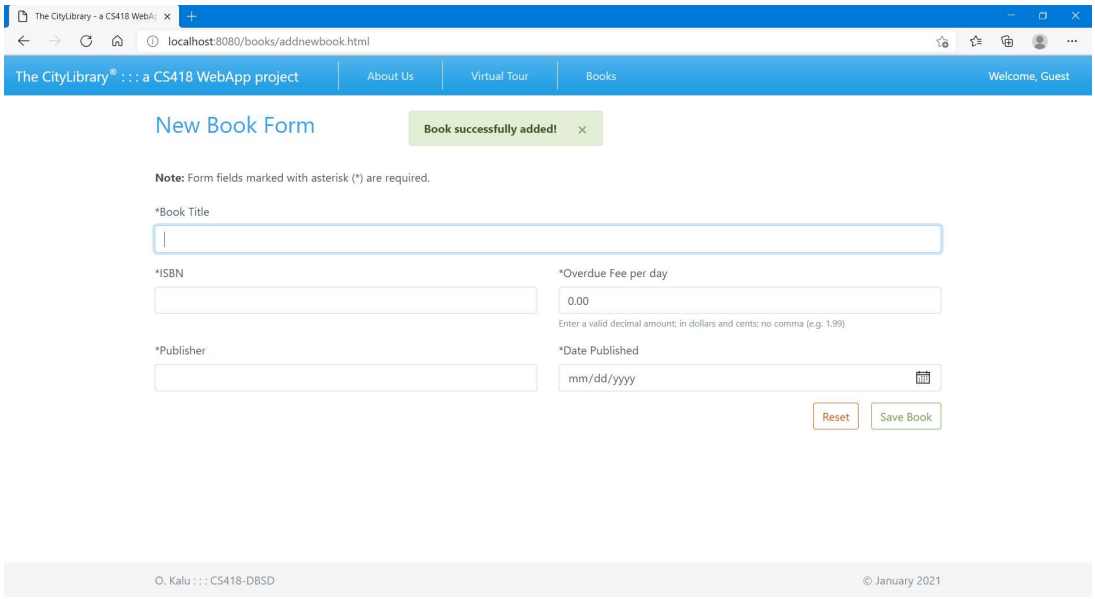



5. Implement code for the CityLibrary “Add new Book” page:
   Note: The endpoint url of the RESTful Web API for adding a new book to the system, can be - https://localhost/citylibrary/api/book/add

   To add a new Book data into the system, do the following:
   a. add a new web page source file named, "addnewbook.html". And make the Add New Book Form UI like the sample shown in step 4, above.
   b. when the “Save Book” button is clicked, the new book data is read from the form input fields, constructed into a JSON formatted object and sent to RESTful service API, using Fetch API, and as an HTTP POST request.
   c. Note: Make all the data fields to be required fields. i.e. prevent submission of the form, unless all the Book data fields are entered.

d. After the form data is successfully submitted, clear the input data fields and you may also show a “Success” info message, as shown below:

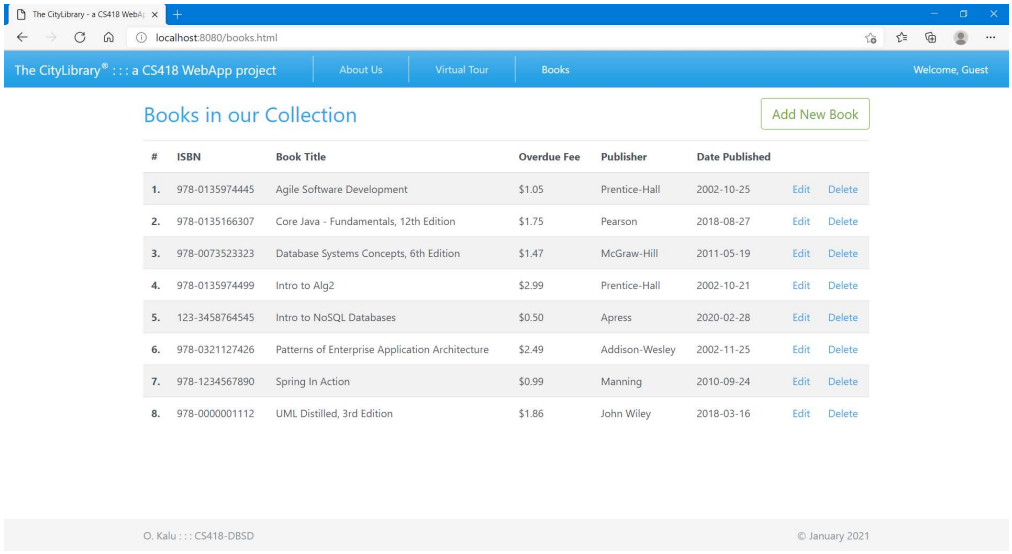


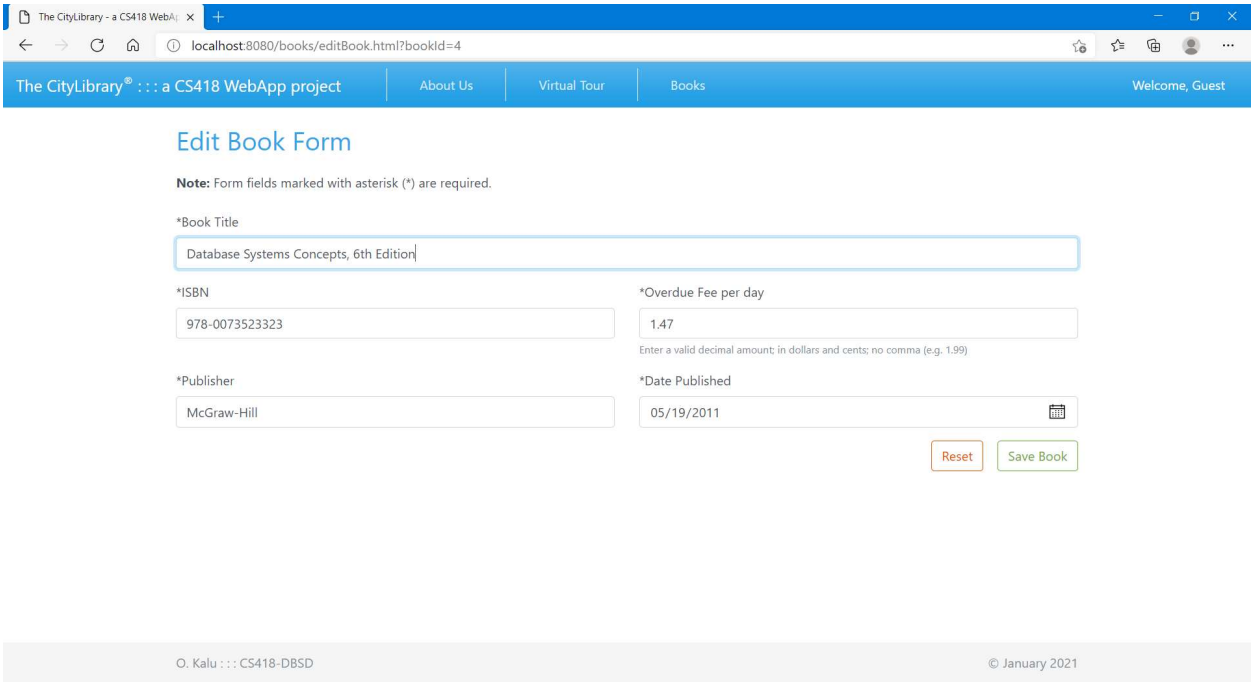

**PHASE 2 Requirements specification:**

6. Editing/Updating and Deleting an existing Book data:

a. On the Books list page, add two new columns to the table/grid. One for a hyperlink named, Edit and the other for a hyperlink named, Delete – as shown below.



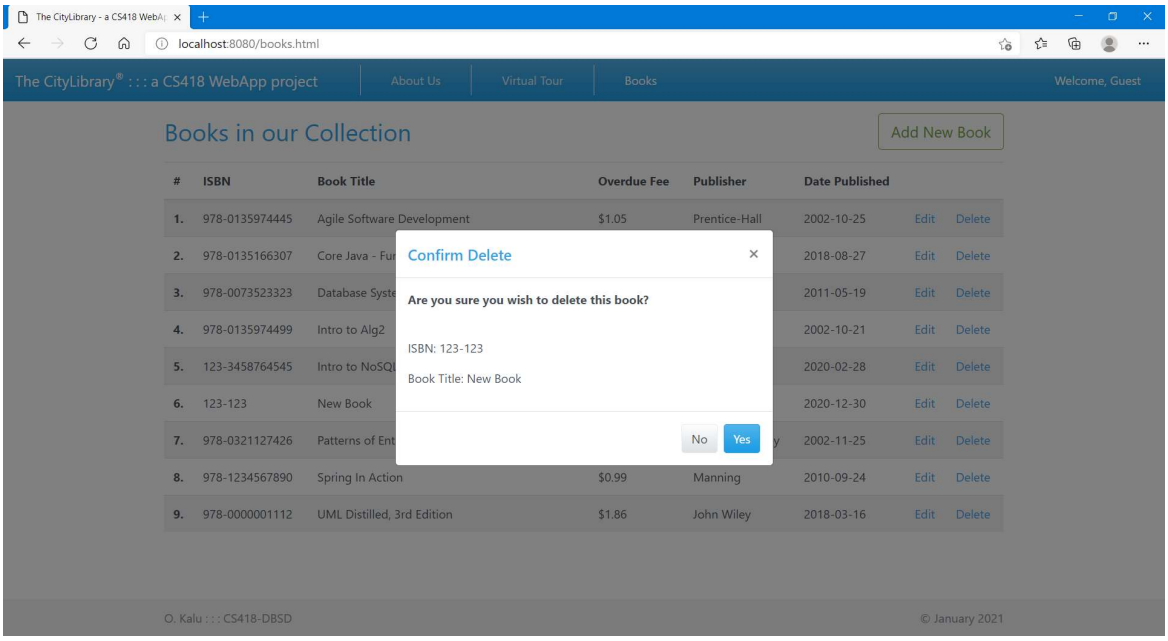

b. For Editing/Updating a Book:
    i. When the Edit link is clicked, make your webapp open the Book's data in an "Edit Book" form

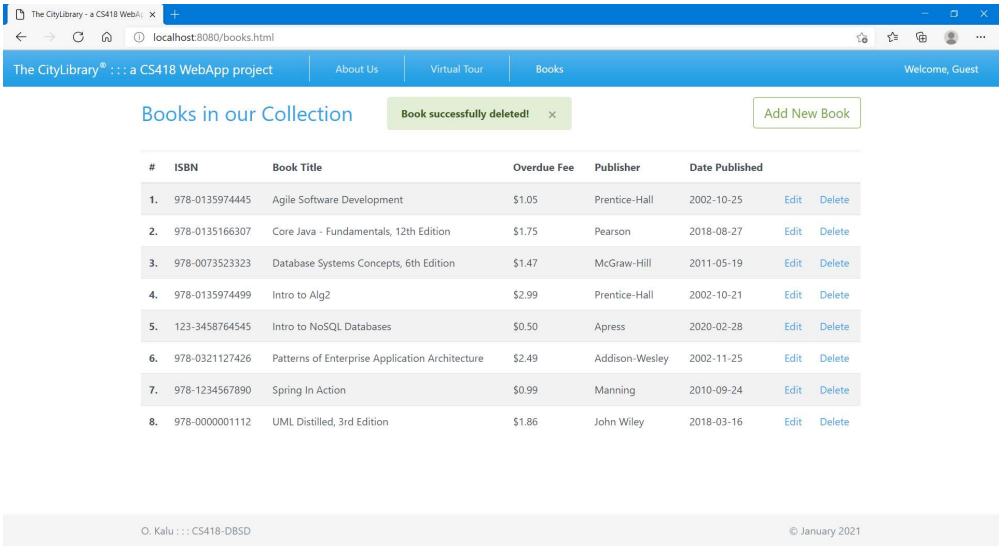


ii. Add code such that when the “Save Book” button is clicked, the edited book data is read, formatted into a JSON Book object and sent to a Web API url such as https://localhost/elibrary/api/book/update/**{bookId}**. **Note: substitute the {bookId} in the path with the actual bookId value.**

c. For Deleting a Book:
   i. Implement code such that when the Delete hyperlink is clicked, the Book data is deleted from the system

   For a good user experience (UX), you may display a Delete warning/confirmation message to prompt the user to confirm the Delete action, before proceeding with the Deletion.




The CityLibrary - a CS418 WebA
localhost:8080/books.html
The CityLibrary® : : : a CS418 WebApp project
About Us
Virtual Tour
Books
Welcome, Guest
Books in our Collection
Book successfully deleted!
Add New Book
# ISBN Book Title Overdue Fee Publisher Date Published
1. 978-0135974445 Agile Software Development $1.05 Prentice-Hall 2002-10-25 Edit Delete
2. 978-0135166307 Core Java - Fundamentals, 12th Edition $1.75 Pearson 2018-08-27 Edit Delete
3. 978-0073523323 Database Systems Concepts, 6th Edition $1.47 McGraw-Hill 2011-05-19 Edit Delete
4. 978-0135974499 Intro to Alg2 $2.99 Prentice-Hall 2002-10-21 Edit Delete
5. 123-3458764545 Intro to NoSQL Databases $0.50 Apress 2020-02-28 Edit Delete
6. 978-0321127426 Patterns of Enterprise Application Architecture $2.49 Addison-Wesley 2002-11-25 Edit Delete
7. 978-1234567890 Spring In Action $0.99 Manning 2010-09-24 Edit Delete
8. 978-0000001112 UML Distilled, 3rd Edition $1.86 John Wiley 2018-03-16 Edit Delete
O. Kalu : : : CS418-DBSD
© January 2021

7. …

//-- The End --//