HTML, CSS & Javascript

Building CRUDL with MVC

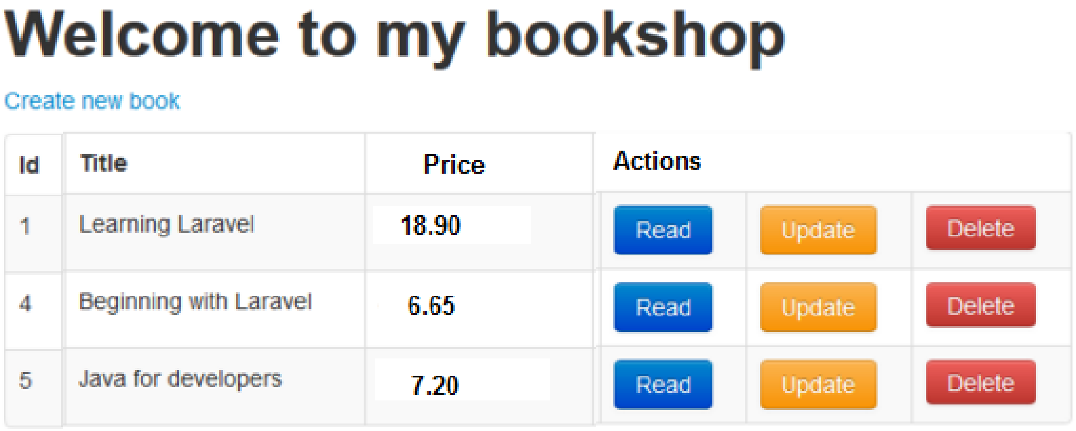
Books Shop (code name: book-shop)

Code an app that shows a list of books: *id*, *name*, *price, imgUrl*.

We will allow the user – a shop owner – to manage the books.

* Follow the conventions and coding style of the project we coded in class.

1. Setup a basic HTML with <table> <thead> and <tbody>
2. Code the function renderBooks() that will render the books with table rows <tr> and <th> and <td>
3. Use a book-service to manage the books (gBooks)
    a. Use the localStorage to load and store books.
4. Now, let's handle CRUDL (Create, Read, Update, Delete and List)
    a. Add an Actions column to the table



Welcome to my bookshop

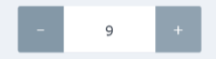
Create new book

Id	Title	Price	Actions		
1	Learning Laravel	18.90	Read	Update	Delete
4	Beginning with Laravel	6.65	Read	Update	Delete
5	Java for developers	7.20	Read	Update	Delete

5. Delete - when the button Delete is clicked call onRemoveBook that will use the service's function removeBook(bookId))
6. Support adding a new book:
    a. When clicked, call the function onAddBook() that will read (prompt) the details from the user: name and price, then will call a function addBook(name, price) that pushes a new book into the gBooks array.
    b. Call the renderBooks() to redraw the table
7. Support updating a book:
    a. When clicked, call the function: onUpdateBook (bookId) that will prompt for the book new price and call the service's function

updateBook(bookId, bookPrice).
Then Call the renderBooks() to redraw the table

8. Use a modal to show all the details of a selected book
    a. When read is clicked, show the modal with details of the selected book.
    b. Add a rate property for the book, set 0 as default, the rate should be a number between 0 and 10.
    c. In the Book-Details modal, allow the user to change the rate of the book by clicking the + / - buttons:



Filtering

9. Support filtering the books by maxPrice, minRate and also by using a search box

Bookmark-ability

10. Filtering should be reflected in query-params
11. Bonus: Add bookmark-ability also to the opened modal state (so when refreshing when the book-details modal is open, it re-opens when page reloaded)

Paging

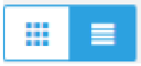
12. Support paging through the books with Next and Prev buttons
13. Update the disabled property of the button appropriately

Grouping

14. Show how many books there are from each category: expensive, normal, cheap

More Bonuses

15. Allow the user to switch between the two presentation modes (cards/table)



    a. Remember the user preference in localStorage key: favLayout: table/cards

16. Read the information from the user using an <input> instead of prompt
17. Make the header of the table clickable to support sorting by name or price
18. Add advanced paging: