

## 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. When user removes or saves a book, show success messages
9. 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.

## Filtering

10. Support filtering the books by minRate and by using a search box

## Bookmark-ability

11. Filtering should be reflected in query-params

## Sorting

12. Allow sorting the list: by price, by name

## Paging

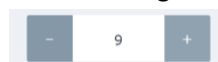
13. Support paging through the books with Next and Prev buttons
14. Update the `disabled` property of the button appropriately

## Grouping

15. At the footer, show in a small table how many books there in the shop are from each category: expensive, normal, cheap

## Book rating

16. Add a rating feature:
  - a. The rate should be a number between 0 (UNKNOWN) and 10 (BEST)
  - b. Add a rate property for the book (default: 0)
  - c. In the Book-Details modal, show the current rate of the book and allow the user to change it by clicking the + / - buttons:



## More Bonuses

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



- a. Remember the user preference in localStorage key: `favLayout`: table/cards
18. Read the information from the user using an `<input>` instead of `prompt`
19. Make the header of the table clickable to support sorting by name or price
20. 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)
21. Add more advanced paging:

