

## **Build a Mini Online Bookshop**

### **Description**

This challenge is to build a mini online Bookshop web application with HTML5 Web Storage/Bootstrap/Javascript/jQuery/EJS Template Engine. This project will be completed individually.

### **User Story:**

#### **As a user:**

I should be able to view all books for sale on the homepage: The book title and year of publishing, book author and price.

I should be able to search for book titles containing certain words

I should be able to click on a book and it should display more details about the book like Short extract, number of pages, publisher, Language, Country, Awards and any other details you want to display.

I should be able to add a new book and remove book.

I should be able to favorite books I like and it should persist even if the browser tab is refreshed.

To keep things simple, you can upload “book images” you intend to use in this project to [cloudinary.com](https://cloudinary.com) and store only paths to the images in your local storage (visit <https://cloudinary.com> for more information).

Hint: You would want to store an array containing objects of all books in local storage under a single key. You should create a constructor function for creating book object.

### **NOTE:**

This challenge should be approached as an individual challenge but you should collaborate with teammates to get insights and help when blocked.

Also, you must use jQuery for all DOM manipulations and EJS template engine for views (HTML rendering).

You are free to use any kind of design, you can however check out sites like, awwwards.com and dribbble for design inspirations. Keep the design simple though.

### **How to submit:**

Create a repository on GitHub to house the source code for your Frontend and Backend Code.

When done, deploy the project to netlify and include a link to the web app on the Github repo.

Include a readme.md file in the repository with steps on how to run the project locally.

=====SUBMISSION=====

Date: Monday, April 27, 2020

Time 9:00 am