



IT 4203 Individual Project Milestone #3: Books Search AJAX

Fall 2016

Due Date: See D2L

Instruction

We will continue the project and convert everything into the AJAX mode. You will end up with just one page in your project.

Requirements:

- Use Google Books API
- Use jQuery AJAX functions (.ajax, .get, or .getJSON, etc.)
- All of the following functionalities should be implemented in the same page.
- Book search functionality: for users to search books by key terms.
 - A textbox to accept user input of search terms.
 - After the search button is clicked, dynamically construct the service request URL and call the service.
 - Process the service response (JSON format) and display search results.
 - Display the first 50 results (at most) in 5 result pages (10 books for each page; note the page here does not mean the actual web page but result page). Style the page number as links or buttons similar to the following examples. Clicking on a page number will automatically perform a research and display the correct result page. Clearly indicate which result page is being displayed.



- Just display the book title or cover image in search results.
- Display book details: when a book is clicked/selected from the search results, the details about the book should be displayed in a section below the results division. Include a cover image, book title, authors, and price information if available; add more information of your choice.
- Display books from one of your public bookshelves on the same page. Display the books using the same style used for the search results. Clicking on a book will display the detailed book information just like the last functionality.

Other requirements

- Please create a separate folder and style for your milestone 3. Make it independent and look like a web app by its own. You will design your own style but keep them clean and organized. Here is a simple and clean design example: <http://it-ebooks.org/search/?q=modern+web>.
- Images and URLs should be displayed or function correctly.
- Do not use any other library, plugin, or framework other than jQuery itself.
- All web pages should be able to be accessed from your public website lively.
- You can organize script and style files and other partial pages, but there should be only one main UI page (HTML or PHP).

[Bonus – 1 point]

Provide a user sign-in through Google function. After signing-in, a user can choose a bookshelf from his own library and display all books in it.

Submission

Submit the following to D2L

1. Compress all source code files into a .zip file.
2. A report in Microsoft Word (or PDF), including:
 - o The URL to your project and/or the milestone web pages.
 - o Screenshots with explanations. Take a screenshot of each webpage displayed in the browser (only the top part of the page if the page is too long). Make sure your screenshots are clearly shown and without cropping.

Grading guide

Your grade is determined on:

- How much you have satisfied the requirements specified in the instruction, and your creativity
- Web site/page design quality
- Well-formed and clean code. You should not use any wizard to generate the code, nor should you copy and paste codes from other sources without cleaning it.
- Following the submission requirements

Rubric:

Score	Summary	Rating Description
10	Outstanding work; beyond expectation.	Satisfy all requirements. Excellent web site/page design. Well-organized files and code; easy-to-read and clean code.
8-9	Good work; meet expectations	Mostly satisfy requirements but may have some minor issues. Clean web site/page design. Well-organized files and code; easy-to-read and clean code.
6-7	Adequate work;	Missing one or two major requirements. Too simple web page design (missing image or links, or incorrectly displayed). Have some problems of organizing code; difficult to read. Missing some required submission items.
<6	Lack of effort.	Failed to stick to the single page mode. Poor web page design (unorganized HTML). Poor coding practices. Live website does not work. Fail to follow submission requirements.