COSC 264 Project

Due Dates:

All code must be committed by April 13th, 2020

Overview:

The project is designed to help develop your skills for full stack development. With this project, you will build a bookstore website that is able to display content from the web site. The front-end will be required to be responsive (mobile). The backend will be developed in PHP. **This is an individual project.** You are encouraged to use code that you have developed in the labs to complete this activity (as most of the functionality required in this project, has already been competed in a lab activity and just needs to be repurposed/modified). Start with building the logic of the site to ensure that it is working and then move onto styling. Additionally, this will help you in preparation for your final exam.

Getting Started:

Accept the GitHub Classroom invitation and checkout the repository to your local machine. Remember to commit your work frequently and to push your code upstream by the due date.

Hardware and Software:

You will develop the project using Linux, MySQL, Apache and PHP in addition to CSS, HTML5 and JavaScript/JQuery on the client-side. **The backend must be completed with PHP. You will need to use XAMPP on your local machine for PHP. The database is hosted for you on `COSC499.ok.ubc.ca`.** Your project will be stored in the class provided repository. You do not need to deploy your web site, but your repository will be required to be cloned and run without any changes. All source code must be available for review. You are required to keep your code under version control with GIT. Database access will be provided for you and you will be able read data from the project database. When you checkout the project, please note the locations of files in the repository. You must use relative referencing so that when your site is deployed and marked, it will work.

The majority of the framework of this project can be (should be) built with the code you have put together across various labs. You are free to re-use/repurpose code samples from class and labs. The goal with the project is to practice the skills required to put together a simple dynamic website. If you repurpose code from an external source, please appropriately attribute your sources. For example, if you use a code snippet from stack overflow, correctly attribute in your code.

You will use an existing database that has read only privileges on `cosc499.ok.ubc.ca`. The database is named `db_project` and the connection information is:

Username: **WebUser**

Password: **9UcM0QQcK1BwAXLk**

At the end of the project, you will have a website that you can use as a portfolio piece to demonstrate your skills. This site has a limited number of pages but demonstrated the power of dynamic content generation and code re-use. I think that at the end of the project you will be surprised how little code is required to accomplish the takes for this site.

MyBookStore:

This project is designed to help you practice and demonstrate the skills you have gained over the course of 264. This project can be built with the different components that you have produced over the course of the labs. The project is an online book lookup resource that allows a user to search and lookup information on a given database. You have a lot of freedom of how you choose to implement the different features and what the site looks like, but please ensure that your site have the required features. The project will be marked on if a feature is present and works, then the point will be awarded for that. Along with your site, will be asked to produce a brief summary of the features you have implemented. Focus on the list of features that are required.

Project Statement: To build a web-based book lookup site that that allows users engage in activities for searching for book resources.

The website will display information about the books in the database that has been provided for you. **Book items are required to have**: a title, short description, price, publisher, publication data, edition, number of page, author name, and category. Additionally, an image needs to be show for each book from the `img` folder in the project repository. Images are referenced by ISBN (meaning that you will need to use the ISBN from the database for a specific book to load the image from the img folder. There are two types of images for each book: a small thumbnail (which will be used on the search page) and a medium size image (which will be used on the book details page). Thumbnail images have THUMB in their name (i.e. `007184158X.THUMB.jpg`) and the medium size images have MED in their name (i.e. `007184158X.MED.jpg`).

You are free to experiment with the design of the page as long as it has the key requirements as it is intended to showcase your skills. You are welcome (and should) use materials developed in the labs. The site will consist for the following types of pages:

- main/search page where a user can search for books and see results of the search (`index.php`)
- product page where a user can see the details (`product.php`)
- Contact Us page where a user can enter the results (`contact.php`)
- A confirmation page that displays the information POSTed from `contact.php` and a thank you message (`processContact.php)`
- You will need to create a page to handle and return the asynchronous search results. It is up to you what you call this (this will be the biggest challenge). The page can simply return the HTML for the search content and you can then update in your main search page. **TIP: remember to use the power of loops in your code when returning search results. If you are unable to generate search results asynchronously, then you will need to show the results on a separate search results page. A single page that processes search queries will be able to handle all of the search queries for this site. Think about how you could handle search options (i.e. using a query string for different search fields).**

A user to the site will be able to search and find on items on the site. Updates are required to be done asynchronously (i.e. only the search content updates).

The site will be **required to have a main page** (`index.php`) where you will be able to search for a book based on different fields (author, title, category) in the same field. Here is a mock-up to give you an idea of what you could design for you could build (Figure 1).

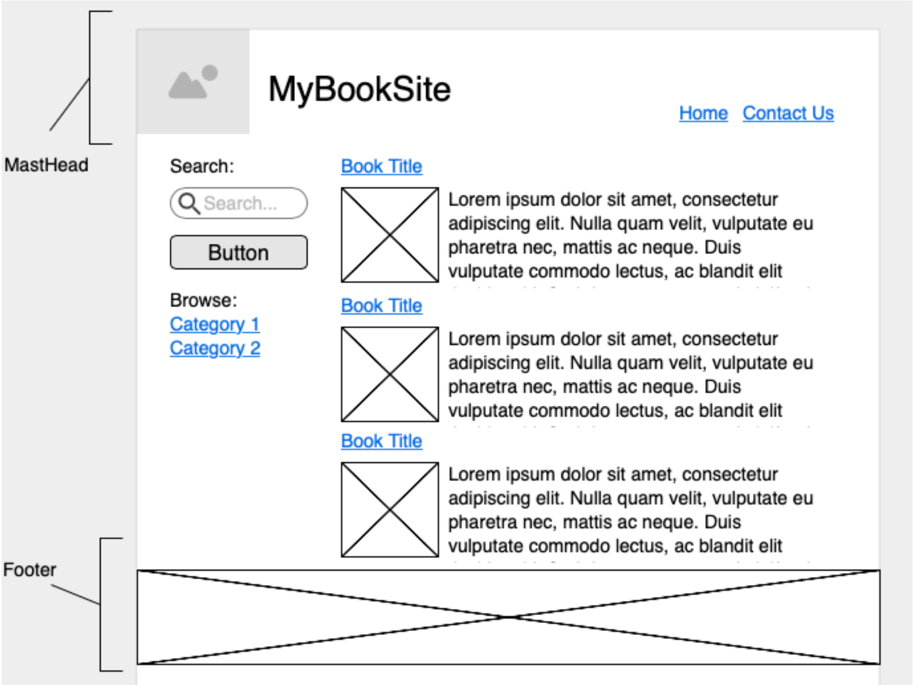
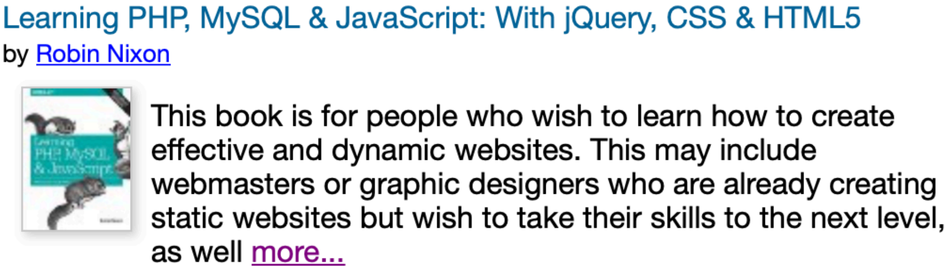

MastHead
MyBookSite
Home
Contact Us
Search:
Search...
Button
Browse:
Category 1
Category 2
Book Title
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit
Book Title
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit
Book Title
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit
Footer

Figure 1 - A sample layout

You will need to support searches author, title, category. The search results are required to be returned asynchronously (see rubric). The search results will have a **hyperlink for the book title**, that when will take you to the product details page. It will also have the **author name which is also hyperlinked**. If you click on the author, the page should regenerate the search results showing the books the author has written. Additionally, it will show the thumbnail for the book as well a small excerpt of the description. The follow image (Figure 2) shows an example what should be returned from the search results (but it doesn't have to look like this as it is up to you to style):



Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5

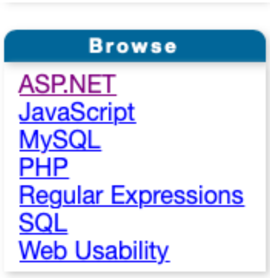
by Robin Nixon

This book is for people who wish to learn how to create effective and dynamic websites. This may include webmasters or graphic designers who are already creating static websites but wish to take their skills to the next level, as well more...

Figure 2 - Detailed search results

When you click on a result from a search, it will redirect to a product page (i.e. `product.php`) that will display all of the book information in a tabular form, including the product image. It will show the price, ISBN, publisher, publication date, number of pages, edition, author, title, category and full description. The author will need to be hyperlinked that will take you to the search results for all books written by that author. The flow of this is similar as to what you would see on amazon.

On all pages, the side-bar will have a general search option as well as links for the different categories of books available that will allow the user to click on a category and get the search results showing all books in that category. The following image (Figure 3) show an example of the browse by category option:



Browse

ASP.NET
JavaScript
MySQL
PHP
Regular Expressions
SQL
Web Usability

Figure 3 - Browse/Search by Category

Your site is required to show the current time using an asynchronous call to `cosc499.ok.ubc.ca/currentTime.php` on your site. The site returns the current time as a text string which will be shown on the sidebar

The site is also required to have a Contact Us form that will allow a user to enter a message, along with their personal information (i.e. name, address (including postal code), phone number, email address and text field for user input. You will be required to perform client-side and server-side validation on all data fields.   This page will POST to `processContact.php` that will confirm the information from the POST along with a thank-you message.

The page requires a masthead as well as a footer. The navigation links need to be available regardless of the where a user is viewing the page (a link to the main search page, link to contact page). You will need to think about the requirements of what pages look like based on the needs of your design. You will be required to make/find a logo for your page (i.e. SVG) along with a brand for your site (name). The footer will need to contain copyright information about your site (make something up) and a general search field and will need to be displayed on all pages.

A first goal is to create the layout for the site. The layout is to be a 2 layout with navigation links along the top**. Styling frameworks are permitted as you are to develop the styling for the page using CSS**. The page requires a masthead as well as a footer. The navigation links need to be available regardless of the where a user is viewing the page.   You will need to think about the required of what pages look like based on the needs of an online store.

Project Objectives: This project is rubric bases.   Each objective has a point value. Implementing only the minimal requirements for the project will permit a passing grade (C). Of the requirements, some are more heavily weighted (you need to implement as part of the core functionality) while others are less weighted. Layout and basic search functionality is key to the design of the site. The other requirements are some of the additional options that can be added to create an improved project.

Functional Requirements:

- Search for items by author, title, category
- Browse items in systems (images will be provided)
- **You can hand-style, use flexgrid or basic bootstrap layout. Bootstrap Themes or existing design layouts are not permitted. The goal is for you to use the tools to design your own response layout.**

- 2 column layout using appropriate design principles (i.e. highlighting nav links when hovered over, etc) responsive design appropriately styled with CSS
- Form validation with JavaScript
- Server-side scripting with PHP
- Data access using MySQL database on cosc499.ok.ubc.ca
- Responsive design philosophy (minimum requirements for different non-mobile display sizes and mobile platforms)
- AJAX (or similar) utilization for asynchronous updates (searching, page updating)
- Search for items (title, author, category)

Bonus:

- Create comments for items (stored in your personal database on cosc499.ok.ubc.ca)
- Collapsible items without page reloading
- Your choice (this is your opportunity to add additional flourish's to your site but will need to be documented in the project summary

Additional allowable technologies:

In addition to the core CSS3, PHP, HTML5 and JavaScript technologies, Bootstrap and JQuery are permitted to be used.

Deliverables:

This project should demonstrate your knowledge in full stack web design and programming. Your final submission will be submitted both electronically in the provided repository. Your site is to use the provided database so it can be run from any web server.

Your final deliverables will include:

- Brief summary of how your site satisfies the requirements. What features did you implement? Include a description of the PHP and JavaScript files of your web site. How does your web site work at a high-level? Identify known limitations of the site?
- Comments on site layout (hand built/Bootstrap)
- Client-side validation
- Ability to search items
- Asynchronous updates for item search
- Main page
- Product page
- Contact Us page
- ProcessContact page
- Page for generating search results (asynchronous) or displaying results

Comments:

This project is not intended to be a complex project (in terms of the content and number of pages) but is intended to provide the opportunity to develop and showcase your full-stack skills. In the development of the project, focus on key functional objectives and work through them in a planned fashion. A simple, functional and well-organized side is acceptable as long as it contains the required functionality utilizing the appropriate technology. **Please review the functional requirements to**

ensure your design satisfies the requirements. I would recommend building the un-styled pages first to test the logic of the site (function over form as the majority of marks comes from functionality). The construction of this site will help you prepare for the final exam as it covers the majority of concepts that will be tested.

Each page in the site needs to be styled using the same layout (look/feel consistency)

Rubric:

/4 Site layout (2 column) with masthead and footer (with fake contact info, address, company name, etc). You can use hand-styling, flexgrid or bootstrap.

/2 responsive design (works and resizes appropriately for mobile devices) (It is up to you how it will appear on a mobile device)

/6 Search functionality for items (title, author, category) (can redirect to new page with details). This will require both frontend and back end pages and includes database connectivity (this includes the front-end search options as well as the generation of the search results on the server side).

/1 User input is sanitized against SQL injection attack

/4 For search results to be updated using AJAX (`index.php` page doesn't reload)

Search results (contains the following):

- book title, author, short description (/2)
- displays thumbnail image for book on search (from img folder) (/1)
- hyperlink on book title to product page for product details (`product.php`) (/1)
- hyperlink on another to update search results showing all other books author has written (`redirects to index.php` with search results – use your existing search functionality) (/1)
- more link at end of desc that will link to product page for product details (`product.php`) (/1)

Product page (Product page displaying information (all details) for book (redirects here when user clicks on search results):

- product details (/2)
- product image (from img folder) (/1)
- author hyperlink to search results for books by author (`redirects to index.php` with search results – use your existing search functionality) (/1)

Side bar:

- general search input field and button (/1)
- search by categories (with links)(this should reuse general search but pass in specific parameters for the categories) (/1)
- Pages show time using asynchronous updates (page doesn't reload but time updated using cosc499.ok.ubc.ca/currentTime.php) (/1)

Contact Us (Contact form that submits to a page on your sever that will display the data (make sure to use relative addressing for time):

- Required fields and posts to processContact.php (/2)
- validation (/1)

ProcessContact:

- confirmation page that displays the information POSTed from `contact.php` and a thank you message (2)

Masthead (on all pages):

- Create logo for masthead (SVG/other for logo + branding) (/1)
- Nav links for main page and contact page (/1)

Footer (on all page):

- Search field and copyright information (/1)

/2 Summary document

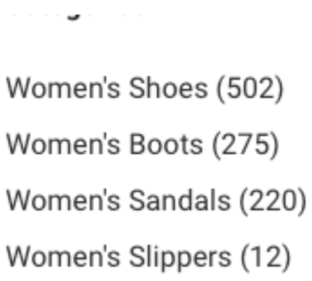
Bonus:

Make sure you include a list of bonus items in your summary document.

/1 Collapsible search results

/2 Contact information is stored in your own database. You will need to create tables to store this information

/1 Add category count to the sidebar Browse (search by category menu) to show the number of books in each category (this needs to be dynamically calculated via query on the database). An example would be (this is from another site) (Figure 4):



Women's Shoes (502)

Women's Boots (275)

Women's Sandals (220)

Women's Slippers (12)

Figure 4 - Count of items in category

/1 If the search results only returns a single item, instead of showing the result in the search result on `index.php`, have it display the product on the `product.php` page.

/1 Make your page more accessible (for an accessibility browser)

2/ Your own features!? – Make sure you clearly document anything else bonusy you add.