CSCI UA.0060 Spring 2025

Assignment 6 – Bookstore Part 1

Deliverables

Upload your PyCharm project to GitHub.

Overall Requirements

In this assignment, you will update the website skeleton provided in your GitHub repository to reflect your own bookstore. The content on the Category page will be come from a dictionary in your Python code.

Specific Requirements

1. Choose your specialty for the bookstore, then pick a name and find/create a logo. It should be between 100-200 pixels on each side (it does not have to be square).
2. Choose four categories of books and four books for each category. Find images of the book covers (I recommend alibris.com for this). The long edge of the book images must be between 150 and 200 pixels (and preferably the same for each book). You will also need images for the four categories. Each image should be the same size – no greater than 175 pixels on each side (they do not have to be square). Image files can be jpg, png or gif. It is better if they are all the same image type, because then you don’t need to store the type in the database. However, this is not mandatory. The images will all be stored in the images folder of the project folder (See next step for where this is). It is best to make the image names of the categories the same as the category names, so you can use one variable for them both. The rest of these instructions assume you have done that. As you will see, this is not necessary (often it’s not possible) for the books.
3. Clone the bookstore repository into PyCharm. The project currently contains:
   1. A templates folder containing four HTML pages:
      1. Index.html – the welcome page
      2. Category.html – the page that displays books for a category
      3. Base.html - used for the header and footer for the other pages
      4. Error.html – that displays some error information
   2. A static folder containing:
      1. A css folder containing one file with rules for both pages
      2. An images folder with subfolders for books, categories and misc. The books and categories folders each have one image as an example. The misc folder has various icons. You needn’t actually use them.
   3. An app.py file that contains the overall structure for your Python program
   4. Screenshots of sample index and category pages
   5. This requirements document.
4. You must update the Python program and HTML files to make the site functional, as follows:
   1. In the Python program (app.py), create a categories list that contains one list for each category containing id and name and then a books list that contains one list for each book containing the book details. An example of each is given in the code.
   2. Update the home function to return the index.html page and the categories list.
   3. Update the category function to retrieve the categoryId that was supplied at the end of the category page URL and create a new list that only contains the books with the selectedCategory. Then, return the category.html page, with the categories list, selected books list and the selected categoryId.
   4. Update base.html with a title for your bookstore, the image file for your logo, your site name for the copyright in the footer and then implement the category dropdown list in the header. This needs a for loop and one line of HTML code, a <p> element containing an <a> element.
   5. Update index.html with some creative text on the lefthand side of the page that relates to your bookstore and update the righthand side to show the category images and names. This should be done with a for loop, iterating through the categories list and inserting one <p> element containing an <a> element for the text and an <a> element for the image. In your code, you will probably have to add the image file extension to the category name, for example, biographies becomes biographies.jpg.
   6. Update the lefthand side of the category.html page to create a set of buttons. The currently displayed category should be styled differently. This can be done with a for loop iterating through the categories and then an if statement to see which category was selected. Two different CSS classes have already been set up for you, one called “selectedCategory” and the other called “notSelectedCategory”. The class attribute should be included in the opening tag of the HTML <p> element.
   7. Update the righthand side of the category.html page to display the books in the currently selected category. This can be done with a for loop, iterating through the books list and then inserting the appropriate element in each book list into the HTML. A half completed template for this has been provided.
5. Depending upon your interest, knowledge, skill and time, modify the site to look how you want it to. You can change colors, fonts, text and layout. The basic requirements for this assignment are that four different books are displayed on the category page depending upon the user’s selection of at least four different categories and that those books are retrieved from the database that you have created.
6. When you have completed the assignment, commit and push the PyCharm project folder to GitHub.

Grading Rubric

See Brightspace for Grading Rubric