CSCI-UA.0060 – Fall 2025

Assignment 6 – Bookstore, Part 1

Deliverables

Upload your project folder to GitHub. Your repository should include:
- app.py
- The templates/ folder (with your HTML pages)
- The static/ folder (with your CSS and image files)

Overview

In this assignment, you will update the provided website skeleton to create your own bookstore. The content on the Category page will come from Python data structures you define in app.py.

You will use Flask to serve your HTML pages dynamically, displaying book categories and book listings using loops and conditionals.

1. Choose Your Bookstore's Theme

Decide on your bookstore's specialty, pick a name, and create or find a logo (100–200 pixels on each side). Add your logo image to static/images/misc/.

2. Choose Your Categories and Books

1. Create four categories of books.
2. Add four books per category.
3. Find cover images for each book (150–200 pixels on the long edge, consistent across all images). You may use sources such as alibris.com or other bookstore sites.
4. Add one image per category (up to 175px per side).

All images go into:

```
static/images/
   books/
   categories/
   misc/
```

It's best to name each category image file to match the category name (for example, fiction.jpg → "Fiction"), so you can easily generate file names dynamically in your HTML templates.

3. Provided Starter Files

Templates folder:
- index.html – the homepage
- category.html – displays books for a selected category

- base.html – defines the shared header/footer layout
- error.html – displays any error messages

Static folder:
- css/main.css – contains styles for both pages
- images/ – includes example subfolders for books, categories, and misc assets

Python file:
- app.py – the Flask backend where you will define lists and routes

4. Update app.py

Inside app.py:
1. Create a categories list. Each entry should be a dictionary with an id and a name.
2. Create a books list. Each entry should be a dictionary containing: title, author, ISBN, price, categoryId, and image file name.
3. Update your Flask route functions:
   - The home() function should return index.html with the categories list.
   - The category() function should:
     - Retrieve the categoryId from the URL.
     - Filter the books list to only include books from that category.
     - Return category.html with the categories list, selected books, and the selected categoryId.

5. Update base.html

Add your store title and logo to the header.
Add your copyright information in the footer.
Implement the category dropdown menu in the header using a loop. Each item should be a <p> element with a link (<a>) that navigates to /category/<id>.

6. Update index.html

Add descriptive text about your bookstore on the left side of the page.
On the right side, show the category images and names dynamically. Use a loop to iterate through categories, and for each one, insert an image linked to the category page and the category name as a clickable link.

Hint: You may need to add the file extension to the image name (for example, biographies → biographies.jpg).

7. Update category.html

Left side:
- Display the category buttons dynamically.
- Highlight the selected category using an if statement and the CSS classes selectedCategory and notSelectedCategory.

Right side:
- Display all books in the selected category using a loop.
- Use the book data (title, author, ISBN, price, and image) to fill in the provided template.

8. Customization

You are encouraged to personalize your site:
- Modify colors, fonts, and layout.
- Add creative text and images.
- Improve navigation or page structure.

Minimum requirement: Your site must display four categories, and each category must show four books retrieved dynamically from your Python lists.

Submission

When complete:

1. Test your Flask app in your browser.
   - Run python3 app.py in EdStem.
   - Open the Web Preview (Port 8080).
2. Commit and push your entire project folder to GitHub.

Grading

See Brightspace for the detailed grading rubric.