CSCI UA.0060 Fall 2024

Assignment 8 – Connect Website to Mongo Database

Deliverables

Upload your PyCharm project to GitHub.

Overall Requirements

In this assignment, you will create a series of webpages that will be used to maintain a Mongo Database that will support an online Bookstore.

Specific Requirements

1. The first thing that you need to do is to create a MongoDB bookstore database that contains a book collection, which in turn contains a minimum of 6 book documents and a category collection that contains four category documents.
2. Each book document should contain a category (category name not Id), title, author, ISBN and price. If you want, you can use the same books that you used for the last assignment.
3. Each category document should have a categoryId and a categoryName (the categoryId should be in addition to the MongoDB created objectId).
4. Clone the bookstoreMaintenance repository into PyCharm. The project currently contains:
   1. A templates folder containing six HTML pages including:
      1. base.html - used for the header and footer for the other pages
      2. index.html – which is the site’s home page.
      3. create.html – used to present a form for users to enter new book info
      4. edit.html – used to allow updates to book information
      5. read.html – used to display all the books in the database
      6. Error.html – that displays some error information
   2. A static folder containing:
      1. A css folder containing one file with rules for all the pages
      2. An images folder which contains a dummy image to allow it to be managed by GutHub You do not need to populate this folder (unless you want to).
   3. An app.py file that contains the overall structure for your Python program
   4. A Bookstore Maintenance screenshots Powerpoint file
   5. This requirements document.
5. Update the base.html page to include your site name, some text in the header and footer and URLs for each of the links.
6. Update the read.html page to display all the books in the database with edit and delete links.
7. Update the create.html page to include an HTML form. This page should:
   1. Allow the user to insert data for each book attribute.
   2. Support the dropdown list of categories being driven from the categories collection (the internal value should be the categoryId, while the category name should be displayed in the dropdown list).
   3. Include a cancel button that links to the read page.
   4. Use the HTTP POST method when submitted.
8. Update the edit.html page to include an HTML form similar to the create page, except that it should display all the attributes for the book that is being edited, including the category in the dropdown.
9. Update the app.py program to include code for all the functions needed to support the specified activities on the HTML pages:
   1. Connect to the MongoDB database.
   2. In the read function, retrieve books from the database and pass those books to the read.html page.
   3. In the create function, retrieve all the categories and pass them to the create.html page.
   4. In the create\_post function, set up the json data structure using the information passed in using the POST method and then insert the document into the database.
   5. In the edit function, retrieve the data for the book selected by the user and retrieve all the categories. Then pass all the data to the edit.html page.
   6. In the edit\_post function, set up the json data structure using the information passed in using the POST method and then edit the document into the database.
   7. In the delete function, send the delete command to the database for the selected book.
10. Depending upon your interest, knowledge, skill and time, you are welcome to modify the site to look how you want it to. You can change colors, fonts, text and layout. However, there is no requirement to update the layout provided. Implementing the maintenance functions is the core of this assignment.
11. Screenshots of sample bookstore maintenance screens are in a powerpoint file that is included in the GitHub repository, so you can use them as a frame of reference. You do not have to exactly replicate these pages.
12. When you have completed the assignment, export the two collections as json files and then copy them into the bookstore project folder. Please rename them to category.json and book.json, so there is no confusion as to what they are. Once done, commit the assignment to GitHub (make sure you include the json files!).

Grading Rubric

See Brightspace for Grading Rubric