Final Project – Book Exchange Web Application

due date on Owl Space

Instructions

Your task is to implement a working version of the Duncan College Book Exchange. Your application should have the same basic functionality as the Duncan College Book Exchange (found on duncancollege.net). You should use the existing implementation as a reference when creating yours. If you wish to add additional components or functionality, you may choose to do so; however you will only be graded on the criteria below.

You have been provided with significant help: most of the files that you need have been created for you with comments that will guide you through the process. The Routes, Controllers, and libraries that handle user sessions, accounts, and authentication have been implemented for you.

The project starter files also contain a simple, clean layout. You may choose to stick with the provided layout, or you can utilize the views and layouts you created as part of Assignments 3 and 5.

Before you start programming, you should:

- Read through the source code of all of the provided files
- Start up the development server and verify that you can successfully log into the application using your network ID

You can find the instructions you need to get started here:
https://github.com/mschurr/coll144-projectstarter/blob/master/README.md

Rubric

This project is worth 30% of your final grade. The project grade itself will be out of 100 points. You will be graded on the criteria below:

5 pt – Home Page

- Application includes a home page (located at "/") that explains how someone might use your application and what problem(s) it solves. Page is viewable regardless of whether or not client is logged in.

5 pt – Navigation System

- Application includes a navigation bar that is easily locatable and present in the same place on every page. If the client logged in, bar contains links to home, logout, and all protected book exchange pages. If client is not logged in, bar contains only a link to home and login.

10 pt – Log In / Log Out System

- Users can successfully log in and out of your application using Rice's Central Authentication Service by following the links in your application. The code has been provided for you; you must simply avoid breaking the links to receive credit here.

5 pt – View Book

- Displays all of the information in the database for a particular book.
- Displays all of the information in the database about the seller associated with the book.
- Information must be displayed in human-readable format.

20 pt – Browse Books

- Retrieves and lists books from the database. The listing is paginated and sorted by the order in which books were listed. Clicking on a book should bring up a page containing further information about the book and its seller (view book). Hint: review the notes on pagination.
- Includes a search form that will filter the provided list of books based on keywords. The searching should occur within the database system. Hint: you may have a very long WHERE clause
- Users can sort the order in which books appear by clicking on the title of a column. Sorting should work even when a search filter is active. Hint: you will find SQL's ORDER BY clause helpful

10 pt – My Books

- Displays all of the books connected to the user's account.
- Books are separated into two categories on the display: listed and sold.
- Users can select books and move them between listed and sold categories.
- Clicking on a book title opens the page to edit that book.
- Contains a link to add a new book.

10 pt – Add Books

- Displays a form containing at least all of the fields present in the Duncan Book Exchange.
- Database entry is properly updated when the form is submitted.
- Input is validated before being used. If validation fails, user is re-shown the form with errors and input preserved.
- Verifies that the user has filled out their contact information before allowing them to add a book.

10 pt – Edit Book

- Handles the case in which the requested book does not exist and displays an error.
- Verifies that the book belongs to the account of the user attempting to edit it.
- Default values are pulled from the database based on the provided book id.
- Displays a form containing at least all of the fields present in the Duncan Book Exchange.

- Database entry is properly updated when the form is submitted.
- Input is validated before being used. If validation fails, user is re-shown the form with errors and input preserved.

10 pt – My Information

- Displays a form for users to enter their contact information; must contain at least all of the fields present in Duncan Book Exchange.
- Information properly stored in database after form is submitted (regardless of whether or not a record is already present).
- Input is validated before being used. If validation fails, user is re-shown the form with errors and input preserved.
- If user has previously filled out the form, default values should be pulled from the database.

15 pt – Usability and Security

- Application is functional and serves its intended purpose. Requires that users are successfully able to: log in, view books on their account, add books, edit existing books, mark books as sold, browse through books in the database, and get information about individual books.
- Application is usable. Requires that all links are present and user never has to navigate to a page by typing a URL in directly (other than the home page) and that the user interface is intuitive and not cumbersome or distracting.
- Application is secure. Requires that only authenticated clients can access protected pages (any page other than the home, log-in, and log-out pages).

As you are new to web application development, you will not lose points if your application is vulnerable to any common web security exploits (SQL Injection, CSRF or XSS). However, you should still take precautions against these attacks.

- SQL Injection: utilize only prepared statements; user input should only be passed as parameters to execute() and not concatenated as part of the query itself
- CSRF: include a randomized token in forms (see notes on CSRF API)
- XSS: sanitize information originating from user input before displaying it (see notes on escape_html)