Assignment-Ruby-on-Rails-Book-Collection 1

Objective

The primary objective of this assignment is to help you get familiar with Ruby on Rails, TDD testing and deployment to Heroku. Specifically, we would be building a simple CRUD Application.

Task description

Your task is to design and develop a Book Collection Application that has four functionalities - Create, Read, Update, Delete a Book.

The book collection app maintains information on books, with each book having 4 attributes: title, author, price, and published date.

During the lab, you'll be guided on how to get started. You'll complete the rest of the assignment on your own.

Submission

1. Deadline: See Canvas
2. You will use your own Github for development, however, you will push your source code to the GitHub Classroom repository: Github Classroom
3. Submit the report (described below) to Canvas.
4. Submit a screenshot of an analysis of your code (e.g., resourceful routes used) against the Canvas rubric using the LLM of your choice. This should communicate improvement suggestions to comply with the given rubric.

Assignment Instructions

1. Accept your assignment from Github Classroom during your lab.

   Copy the link to the browser: LINK

2. Start the pre-built docker image (`docker pull paulinewade/csce431:sp26v1`). See more information in the Docker lab.

   When you're building the app, you can develop in the dev branch, after which you will merge to the test branch. Once your app passes tests, you can merge to the main branch.

3. Coding

   - Using the LinkedIn module Ruby on Rails 7 Essential Training as a guide, develop your book collection app that maintains information about books, with your book model having the attributes title (String), author (String), price (Numerical), published-date (Date/Datetime). The "Task_Manager" app you developed from the LinkedIn module will be similar to this one.

     In this assignment, you will only work on the attribute **title** and the other ones will be implemented in Book Collection 2.

     You should have done at least up to MODULE 3 of LinkedIn Ruby on Rails 7 Essential Training

   - Remember to always make the migrations to create a database table
   - Make sure you follow Rails convention and architecture (MVC architectural design pattern).
   - The application should contain 5 views - FOR THIS ASSIGNMENT, you will only create 1 attribute 'title' which is of type String
     - Home Page - serves two purposes: first is to view all books, second is to contain handlers (e.g., links or buttons) to perform the CRUD (i.e., create, read, update, delete) functions. **Take a screenshot (1) of this page, which you will include in your report later.**
     - Add a Book Page - contains a form that allows you to input attributes of a book. **Take a screenshot (2) of this page, which you will include in your report later.**
     - Update a Book Page - pre-populated with the current details and can be updated. **Take a screenshot (3) of this page, which you will include in your report later.**
     - Show Details Page - to view the details of the book. **Take a screenshot (4) of this page, which you will include in your report later.**

        - Delete a Book Page - contains a confirmation message to delete a book (Check Ruby on Rails 7 Essential Training to see an example of how to implement this.) **Take a screenshot (5) of this page, which you will include in your report later.**

    - Implement a flash notice popping up on the Home Page after performing each operation (except 'show'). **Take a screenshot (6) of this page, which you will include in your report later.**
    - All the pages should have a link going back to the Home Page and the routes should use resourceful routes. **This should be included in the screenshots of the pages described above.**

4. Referring to the Docker lab, run your app in the browser: http://127.0.0.1:3000, and **take a screenshot of the running website. You'll need to put the screenshot (7) into your report later.**
5. Create a new branch test

   $ git checkout -b test
6. Save all code

   $ git add .

   $ git rm node_modules (if node_modules is in your .gitignore, this step is not necessary)

   $ git commit -m "your commit message"
7. Push the changes to your Github repository. To check the remote, use $ git status.

   $ git push origin test
9. Add a screenshot of your pull request to merge to main branch - screenshot (8)
10. Add a screenshot of your commit history in your branch - screenshot (9)

Report

Your report should include the following content:

1. Link to your Github repository where you committed the working code.
2. Screenshots. Please add them sequentially from (1) to (n). For each screenshot, explain what it is about.

Save your report as FirstName_LastName_Lab-BookCollection 1.pdf and submit it to your teams individual submissions folder.

References:

Linkedin Trainings free of charge with your TAMU credentials

Ruby on Rails 7 Essential Training

Articles

Ruby On Rails - HTTP, MVC and Routes

Active record: https://guides.rubyonrails.org/active_record_basics.html

Routing: https://guides.rubyonrails.org/routing.html