

Problem Set 6

The following problem set will be worth 100 points. The code will be submitted electronically via Canvas using the “Problem Set 6” dropbox. The assignment is **due at the start of the night of the final exam**.

Your code will be graded on both elegance and *user-friendliness*.

Exercise #1 –Bookstore Problem Extended (100pts)

Overview

Recall with Problem Set 5 you built a web application for a bookstore. Problem set 6 will extend the assignment slightly.

Database Requirements

With the first application you made, all data the application uses was hard-coded inside the application (and used when the application first started up.) With this assignment you will be persisting this data to a database.

Consider the following relationship. An author can have many books. Assuming each book is written by only one author, this relationship is considered to be a *one-to-many relationship* between author and book.

For this problem set the following is required to obtain full points for the assignment:

- Create a database using H2 with two database tables: author and book
- Create Spring Data JPA entity classes for author and book
- Change the repositories in the application you wrote for Problem Set 5 to query the entities using Spring Data JPA
- The data should be persisted to the database when ever a book and or author is updated/added in the system you wrote

Extra Credit Requirements

The following requirements are optional extensions to further enhance your bookstore web site even more:

Alternative Database (+5 pts)

H2 is an in-memory database, which means as soon as your application server is stopped all data is lost. With this requirement, feel free to change your database persistence layer from using H2 to using either a SQL database server (PostgreSQL, MariaDB, MySQL, SQL Server) or a NOSQL/document database (MongoDB for instance.) If doing this option, please submit any SQL you used for constructing your database schema if needed.

Alternative User Management (+5 pts)

With your new database in place (see first extra credit assignment), one area of refactoring for your application is changing the administrative user over from something hard coded in the application to an additional table in your database. Ensure your user table has at minimum user name and passwords as fields.

Alternative User Security Requirements (+5 pts)

Once your user schema is in place, you'll want to add a layer of encryption for the passwords. As you recall, the password for Problem Set 5 was simply 123456. For this exercise, you are to encrypt the passwords in the database (recommended to use the Bcrypt algorithm) and integrate Spring Security to use this in your user details service.

Alternative Front End Interface Requirements (+10 pts)

The customer lastly would like the interfaces for managing books to be handled by a modern JavaScript framework for the UI. For this requirement, integrate one of the popular JavaScript frameworks into some of the screens of your application. Feel free to use one of AngularJS, ReactJS or EmberJS.

Submit all files (including the Gradle script I can use to run your application) in a zip file.

Submission Requirements: Submit the aforementioned files in a zip file with the naming strategy:

First initial + last name + PS + problem set number.zip

As an example, I would submit the code in a zip file named **mmeluskyPS6.zip**. Submit your zip file via the "Problem Set 6" Canvas dropbox before the date of the close of the assignment.