Assessment 8

Creating the Database for the Book Store App

Deadline [Next week before the lab]

In part B you should implement the entity schemas and the DB repository for the **BookStore App**. The DB repository should implement the same functionality as the file-based repository provided in the base solution. NOTE : **You should test your implementation as you progress and document your testing .**

1. Open the **Book Store App**. Change app.js to connect to "**BooksDB**" MongoDB database.
2. Create four Models [Book, Author, Borrower, and Borrowing]. The schema of these models should be derived based on the json data files provided in the base solution. Hints:
   - Book.authors property should be an array of refences to the Author model.
   - Author.books property should be an array of references to the Book Model.
   - Borrowing.bookId property should be a reference to the Book Model.
   - Borrowing.borrowerId should be a reference to the Borrower Model.

Make sure the Book, Borrower, Borrowing and Author Model schema properties are validated with custom validation. Example, the book title, author, …. are required properties.

3. In the **books-repository.js** import both models **Author, Book, Borrowing, Borrower** models and implement all the methods in books-repository.js using those models.
   Make sure that you implement a method to load the provided json data to MongoDB.

4. Add the needed repository and service methods to implement the following reports:
   a. Books Summary report: return the books counts, average page count per book category.
   b. Top 3 borrowers with the total number of books they have borrowed.
   c. Top 3 borrowed books and the number of times they have been borrowed.
   d. Borrowing summary: Summary of borrowings by book category. The report should return the total number of borrowings per book category.

   Test these reports using Postman. No need to provide a UI for them.

NOTE: All the query should be done on the Database. You should NOT do any filtering or aggregation on the using JavaScript array methods. You should use the database queries capabilities to implement all the aggregation, filtering needed for your solution.

Further details about MongoDB query operators is available at https://docs.mongodb.org/manual/reference/operator/query/ . You may use **Compass or Robo 3T** to interact with MongoDB database.

You need **to** test your implementation as you progress and document your testing. After you complete the lab, fill in the ***TestingDoc-Grading-Sheet.docx*** and save it inside **Assessment** folder. Sync your repository to push your work to GitHub.