Capstone 1 for Web Application Development by JAVA (Individual)

Congratulations on reaching the first capstone of the course! Over the next three days, you will be working on your first capstone that will allow you to further apply your knowledge and skills to build another real-world application. This project will serve as a demonstration of your proficiency in building a Spring boot server, focusing on Validation, Controller and Service Layer.

Project Description

Create E-commerce website where users can add, get, update, delete and buy products (amazon clone)

Each student will work solo and submit their own project. Below you will find expectations and minimum requirements.

Be creative, and have fun!

Rules & Guidelines

- **Dependencies:** Each student should use all required dependencies: Spring web, Lombok, Validation.
- **Layers:** Each student should use all required layer: Controller, Service Layer.
- **Individual Work**: Each student is expected to work individually on their project. Collaboration is not allowed, and each student should submit their unique work.
- **Version Control (Git)**: Create a GitHub repository to host your capstone project. Use non-main branches for development and create Pull Requests to merge major features. Commit frequently with descriptive messages to show your progress.
- **Academic Integrity**: Do not copy or reuse code from previous modules or external sources. The project is a showcase of your learning progress.
- **Learning References**: You may refer to course materials and code examples as learning references but implement your solutions independently.
- **Instructors Support**: The instructors will be available for general questions and guidance throughout the project.

Schedule & Deadlines

- **Project Start Date**: 04/08/2024
- **Project Submission Deadline**: 06/08/2024

Minimum Requirements

1- Create Product Class:

- id (must not be empty).
- name (must not be empty, have to be more than 3 length long).
- price (must not be empty, must be positive number).
- categoryID (must not be empty).

2- Create Category Class: • id (must not be empty).

• name (must not be empty, have to be more than 3 length long).

3- Create Merchant Class:

- id (must not be empty).
- name (must not be empty, have to be more than 3 length long).

4- Create MerchantStock Class:

- id (must not be empty).
- productid (must not be empty).
- merchantid (must not be empty).
- stock (must not be empty, have to be more than 10 at start).

5- Create User Class:

- id (must not be empty).
- username (must not be empty, have to be more than 5 length long).
- password (must not be empty, have to be more than 6 length long, must have characters and digits).
- email (must not be empty, must be valid email).
- role (must not be empty, have to be in (“Admin”,”Customer”)).
- balance (must not be empty, have to be positive).

6- Create endpoint for getting and adding and deleting updating a Product.

7- Create endpoint for getting and adding and deleting updating a Category.

8- Create endpoint for getting and adding and deleting updating a Merchant.

9- Create endpoint for getting and adding and deleting updating a MerchantStock.

10- Create endpoint for getting and adding and deleting updating a User.

11- Create endpoint where merchant can add more stocks of product to a merchant
Stock

- this endpoint should accept a product id and merchant id and the amount of additional stock.

12- Create endpoint where user can buy a product directly

- this endpoint should accept user id, product id, merchant id.
- check if all the given ids are valid or not
- check if the merchant has the product in stock or return bad request.
- reduce the stock from the MerchantStock.
- deducted the price of the product from the user balance.
- if balance is less than the product price returns bad request.
- Add 3 extra endpoints to enhance your e-commerce website.

Ideas for extra credit

Beyond the minimum requirements, you are encouraged to use your imagination to make this application as nice and complete as you wish.

Add 1-3 extra endpoint.

Below are some ideas you might find interesting, but you are not restricted to them.

Resources

These are some resources that may help you in implements the project:

https://reflectoring.io/bean-validation-with-spring-boot/