**Problem Statement:**

Develop a web based, distributed application using Springboot and Microservices architecture.

eCommerce application with two different roles, Seller and Buyer.

Seller can perform below operations

1. add new products to inventory
2. update product stock numbers, price
3. view product details

Buyer can perform below operations

1. search for product based on search string such as product name/make/model, price range, etc…
2. select a Product, specify quantity
3. Buy product
4. After product is bought, update product stock details
5. view purchase History of specific Buyer.
6. Add/delete products to/from shopping cart, check out shopping cart. This requirement is optional.

Consider using the DB Tables sellers, buyers, products, purchase\_history, you can assume the columns/fields based on the scenario.

**Please see that below best coding practices are applied to the Project:**

1. Use Java 8 and other advanced features
2. Layering and folder structure(with Proper Packages/Sub Packages) - Controller, Service, Repository, Entity classes
3. Use interfaces for loose coupling
4. Proper Exception Handling and create Custom Exceptions
5. Exception Wrapping
6. Close Connections
7. Proper Table Schema with required Foreign key relationships,
8. Validations - use regex

**Deliverables:**

1. Compiled Java source code on Github
2. Output Screenshots(atleast 3 to 4)

