

E-commerce Application

Link: [Naveena Lingam / E-Commerce Java ACE · GitLab](#)

Basic Overview of Repo

This repository contains models for `Product`, `Vendor`, `Cart`, `User`, and other entities. It serves as the backend for an e-commerce platform with APIs for creating products and vendors.

Assignment 1: Product Management & Authentication

Objective:

- Develop APIs for managing products under vendors.
- Ensure only admins can create products and associate them with specific vendors.
- Create a paginated API to fetch all products, accessible by all authenticated users.
- Implement JWT token-based authentication for different user roles.

Details:

1. APIs for Adding Products Under a Vendor:

- Create an API that accepts product and vendor information and associates them in the database.
- Only admins should have access to this API.

2. APIs to Get All Products:

- Develop an endpoint to retrieve all products in a paginated manner.
- Ensure this API is accessible by all authenticated users (USER, ADMIN, SUPER_ADMIN).

3. JWT Token Mechanism for User Roles:

- Implement JSON Web Tokens (JWT) for user authentication.
- Define three user roles:
 - USER: Can view products and add them to the cart.
 - ADMIN: Can perform all actions allowed by USER plus create vendors and products.
 - SUPER_ADMIN: Can all actions allowed by ADMIN plus adjust product prices by up to 5%.

4. Additional Product Views:

- Create an API to display the total number of products (accessible only by SUPER_ADMIN).
- Create an API to filter products based on specified price ranges

Evaluation Criteria:

- Successful implementation of APIs for product management.
- API Documentation
- Proper handling of user roles and permissions.
- Pagination functionality for fetching products.
- Code quality and adherence to best practices.

Assignment 2: Generate Unit Test cases

Objective

Application should be well covered and tested using unit test cases. Make sure there will be at least 80% code coverage.

Detailed Assignment

1. Write unit test cases by mocking objects as needed

-
2. Make sure API testing with varied inputs and all edge cases are covered.

End Goal

1. Test coverage reports covering all code
2. Test case execution without failures