RevShop

Application Overview

The RevShop project aims to develop a secure, user-friendly, and versatile e-commerce application for both buyers and sellers. The core functionalities for buyers include browsing products, adding products to a cart, checkout, and payment processing. Sellers can add products, manage inventory, and fulfill orders. The project's completion will be demonstrated through a cloud-hosted working version, technical presentation, and associated diagrams.

Core Functional Scope

Buyer user account:

As a buyer, I should be able to:

1. Register on the platform.
2. Login into the application using email and password.
3. View product details including image, price, description, and user review.
4. Browse products by category or keywords.
5. Add or remove products from the cart and provide quantity.
6. Checkout and enter shipping and billing information.
7. Get email notifications when an order is placed.
8. View order history.
9. Review products.
10. Save the product as a favorite.
11. Make payment using the payment gateway.

Seller account:

As a seller, I should be able to:

1. Register as a seller with email, password, and business details.
2. Login into the application using email and password.
3. Manage inventory of products.
4. Add new products with price and description.
5. See placed orders.
6. Receive email notifications when a user places an order.
7. Provide discounted price along with the maximum retail price.
8. View product review.
9. Get web notifications when the product's quantity is less than the threshold. (Seller sets the threshold value).

Non-Functional Requirements

User Experience:

1. Have an intuitive design for the user to work with the application without any training or guidance
2. Have clean & consistent UI, color theme and easy to use navigations
3. Use bootstrap framework for responsive pages

4. Have proper tab indexing for users to navigate between the fields without usage of mouse.

User Inputs & outputs:

1. Have appropriate HTML fields for the user inputs
2. Wherever possible use the client-side validations for the user input
3. Display the appropriate user info/error message with appropriate colors and icons

Performance:

1. Use compressed images / assets to increase the page performance
2. Use the application validated using the Chrome's Lighthouse tool and improved based on the report

General standards:

1. Ensure the w3 standards are implemented for better accessibility. E.g., Using alt attribute for image tag.
2. Ensure the SEO recommended meta tags are added.

Security:

1. Ensure the CORS restriction is applied, if applicable.
2. Ensure Route Guarding/Authenticated Routing is implemented.
3. Ensure that the secrets are stored as environment variables using secure credential storage.

Validation and Error Handling:

1. Validate the user inputs for its types and format.
2. Display functional related user messages (either for input/error/output) - no system error codes.
3. Handle the exceptions and errors gracefully.

Logging:

1. Ensure the application is using proper logging framework and methods.
2. Ensure the application's log level is configured using configuration files so that it can be changed without changing the code.
3. Also ensure that the application logging is configured to output to the mentioned log file.

Testing:

1. Ensure sufficient test cases are written using appropriate testing frameworks.
2. Ensure the code coverage closed to be 80%

Security:

1. Ensure the SQL/NoSQL injection threat is taken care.

Coding Standard:

1. Use the industry coding standards and conventions.
2. Modular based code development for better reusability.
3. Ensure proper usage of resource objects such as database connectivity objects to avoid resource leakages.
4. Ensure proper usage of design patterns and application layering (such as Business Service, DAO Layer etc.) wherever applicable.