Ecommerce Application

Overview:

An Online shopping application is a digital platform where users can browse, select, and purchase products or services over the internet. It provides a user-friendly features like product listings, search functionality, user accounts, shopping carts, secure payment
processing, order tracking, and customer support. These applications aim to provide a convenient and seamless shopping experience for users, allowing them to shop from anywhere at any time.

Online shopping is becoming increasingly popular for variety of reasons. There are certainly outside factors such as increasing gas prices, difficulty in getting to traditional stores and hassles often associated with shopping malls and other traditional stores to contribute to the increased interest in online shopping.

Consumers can get full information about the product with its reviews being passed by the existing users. If one wants to buy a product he/she is no longer limited to asking the friends and families because there are many products reviews on the web which gives opinions of the existing users of the product.

Online Shopping sites contain wide variety of goods both high quality and mild quality keeping in mind level of people.

- There are no national and International Barriers.
- In online shopping the consumers will be in a demanding position and suppliers will not be in a commanding position.
- There is enormous employment opportunities in online shopping.

Functional Requirement:

1. Product Catalog Management:

- Product Upload: Sellers can add new products to the catalog with details such as name, description, price, and images.
- Product Categories: Organize products into categories for easy browsing.
- Product Search: Users can search for products using filters like category, price

range, and brand.

2. Shopping Cart and Checkout:

- Add to Cart: Allow users to add products to their shopping cart for future purchase.
- Cart Management: Users can view, edit, and remove items from their shopping cart.
- Secure Checkout: Implement a secure payment gateway for processing transactions.
- Order Tracking: Provide order tracking functionality for users to monitor the status of their purchases.

3. User Reviews and Ratings:

- Review System: Enable users to leave reviews and ratings for products they have purchased.
- Rating Aggregation: Display average ratings for products based on user feedback.
- Review Moderation: Allow admins to moderate reviews to ensure authenticity and appropriateness.

4. Wishlist and Favorites:

- Wishlist Management: Users can create and manage Wishlist of products they are interested in purchasing.
- Favorite Products: Allow users to mark products as favorites for easy access later.

5. Order History and Account Management:

- Order History: Users can view their past orders and order details.
- Account Settings: Provide options for users to manage their account information, including password changes and email preferences.

6. Promotions and Discounts:

- Discount Codes: Support the use of discount codes during checkout for promotional offers.
- Promotional Campaigns: Allow admins to create and manage promotional campaigns such as discounts, flash sales, and loyalty programs.

7. Customer Support:

- Help Center: Offer a comprehensive help center with FAQs, guides, and troubleshooting tips.
- Contact Us: Provide users with a means to contact customer support for assistance with orders or inquiries.

8. Multi-platform Accessibility / Scalability:

- Responsive Design: Ensure the app is accessible and user-friendly across various devices and screen sizes.
- Mobile App: Provide a mobile app version of the platform for seamless shopping on smartphones and tablets.

Non-Functional Requirements for Online Shopping Application:

1.Performance Requirements

Response Time: The system should respond to user actions (e.g., adding items to cart, searching for products) within 2 seconds under normal load conditions.

Throughput: The application should support at least 10,000 concurrent users without performance degradation.

Scalability: The system should handle peak loads during high-traffic events (e.g., Black Friday, Cyber Monday) by scaling up seamlessly.

2.Usability Requirements

Intuitive Interface: The user interface should be easy to navigate, with a clear and consistent design that adheres to usability standards.

Accessibility: The application must comply with WCAG 2.1 Level AA standards to ensure accessibility for users with disabilities.

Localization: The application should support multiple languages and regional formats, including currency and date formats.

3.Reliability Requirements

Uptime: The system should have an uptime of 99.9% to ensure high availability.

Error Handling: The system should provide clear error messages and guidance for users when errors occur.

Data Integrity: The system must ensure the integrity and consistency of user data and transactions, with mechanisms for backup and recovery.

4.Security Requirements:

Data Encryption: All sensitive data, including user information and payment details, should be encrypted in transit (using HTTPS) and at rest.

Authentication and Authorization: The system should implement strong authentication mechanisms, including multi-factor authentication, and ensure proper authorization controls to protect user data.

Compliance: The application must comply with relevant regulations and standards, such as GDPR for data protection and PCI DSS for payment processing.

5.Maintainability Requirements:

Code Quality: The application code should be well-documented, modular, and follow industry best practices to facilitate easy maintenance and updates.

Automated Testing: The system should have a suite of automated tests (unit, integration, and end-to-end) to ensure robustness and facilitate continuous integration and delivery (CI/CD).

6.Scalability Requirements

Horizontal Scaling: The application should support horizontal scaling to handle increased load by adding more servers.

Database Scaling: The database architecture should support sharding and replication to handle large volumes of data and high transaction rates.

7.Portability Requirements:

Platform Independence: The application should be platform-independent, running on various operating systems and browsers without compatibility issues.

Deployment Flexibility: The system should support deployment in different environments, including on-premises, cloud (AWS, Azure, GCP), and hybrid setups.

8.Compatibility Requirements:

Browser Compatibility: The application should be compatible with all major browsers (e.g., Chrome, Firefox, Safari, Edge) and their latest versions.

Device Compatibility: The application should be fully responsive and functional on various devices, including desktops, tablets, and smartphones.

9.Legal and Compliance Requirements:

Data Protection: The application must ensure data protection and privacy in accordance with laws and regulations such as GDPR, CCPA, and others relevant to the jurisdictions where it operates.

Transaction Compliance: The system must comply with financial transaction regulations, including PCI DSS for payment processing.

10.Support and Maintenance Requirements:

Customer Support: The system should provide robust customer support options, including live chat, email, and phone support.

Monitoring and Alerts: The application should include monitoring tools to track performance, detect issues, and send alerts for critical failures or security breaches.

Conclusion

These non-functional requirements are crucial for ensuring that the online shopping application is reliable, efficient, and user-friendly. Meeting these requirements will help provide a seamless shopping experience for users, ensure data security, and support the application's long-term sustainability and scalability.

System Architecture for Online Shopping App

Frontend: React.js

Backend: Java with Spring Boot

Database: MySQL or MongoDB

Frontend: Hosted on Vercel, Netlify, or web servers.

Backend: Hosted on AWS, Heroku, or dedicated servers.

Database: Managed services like AWS RDS (MySQL) or MongoDB Atlas.

CI/CD: Tools like Jenkins, GitHub Actions for continuous integration and deployment.

Feature- Login

As a user a I want a successful login through this online shopping application software

Scenario: Successful Login

Given: I have username as person

And: I have password as pass@123

When: I go to login page

And: I enter my username and password

And: I click on login button

Then: It should display login successful

And: I should be taken to dashboard

Feature: Filtering Data

As a user, I want to filter the data according to my needs. So that I can easily find what I'm looking for.

Scenario: Successful Filtering

Given I'm on the home page of application it should display available filter categories.

When I enter filters in filter panel.

And I click the apply filter button.

Then I should be taken to the filtered data page.

Feature: Adding object into cart

As a user I want to select object from ecommerce and that object should be added to cart.

Scenario: Successfully adding object in cart.

Given I am in home page

When I select any object

then I should be taken to detail of this object

And I click the "Add to cart" button

Then I should be taken to into my cart page