SRS For Ecommerce WebApp

1. Functional Vs Non-Functional Requirements

Functional Requirements:-

* Function requirements describe the specific functions or actions that the system must perform. These are typically the "features" or capabilities that the system should have.

Non-Functional Requirements:-

Non-function requirements, on the other hand, define the overall qualities or attributes that the system must have. These requirements focus on aspects like performance, usability, reliability, security, and scalability.

1. Functional Requirements in the project:-

1.User Registration and Authentication:

Users should be able to register for an account.

Users should be able to log in securely with their credentials.

Password reset functionality should be available.

2.Product Browsing and Searching:

Users should be able to browse products by category, brand, or other filters.

A search functionality should allow users to find products based on keywords.

Product pages should display detailed information, including images, descriptions, prices, and availability.

3.Shopping Cart Management:

Users should be able to add products to their shopping cart.

Users should be able to view and edit the contents of their shopping cart.

The shopping cart should calculate and display the total price, including taxes and shipping fees.

4.Checkout Process:

Users should be guided through a multi-step checkout process.

Users should be able to enter shipping and billing information.

Payment options should be provided, such as credit/debit , PayPal.

Users should receive confirmation of their orders via email.

5.Order Management:

Users should be able to view their order history.

Users should be able to track the status of their orders.

Admins should have access to manage orders, update order statuses, and handle.

3.Non-Functional Requirements in the Project:-

1.Performance:

Response Time: Specify the maximum acceptable time for the system to respond to user actions (e.g., page loading time).

2.Scalability:

Define how the system should handle increasing numbers of users,

transactions, and data volume without a significant degradation in performance.

Throughput: Determine the maximum number of transactions the system should be able to handle within a given time frame.

3.Reliability:

Availability: Specify the percentage of time the system should be operational (e.g., 99.9% uptime).

Fault Tolerance: Define how the system should handle failures gracefully, ensuring minimal disruption to users.

Data Integrity: Ensure that data remains accurate and consistent throughout transactions and interactions with the system.

4.Security:

Authentication and Authorization: Define requirements for user authentication mechanisms and access control to ensure only authorized users can access certain features or data.

Data Encryption: Specify encryption standards for sensitive data such as user credentials, payment information, and personal details.

Protection against Attacks: Define measures to protect against common web security threats like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

5.Usability:

User Interface Design: Specify usability guidelines for the user interface, ensuring intuitive navigation, clear layouts, and consistent design elements.

Accessibility: Ensure that the application is accessible to users with disabilities, following accessibility standards such as WCAG (Web Content Accessibility Guidelines).

6.Scalability:

Horizontal Scalability: Define how the application should be able to scale horizontally by adding more servers or instances to handle increased load.

Vertical Scalability: Specify requirements for vertical scaling, such as the ability to upgrade hardware resources (e.g., CPU, RAM) to accommodate growth.

7.Maintainability:

Modularity: Specify requirements for a modular architecture that allows easy maintenance and updates without affecting the entire system.

Documentation: Define standards for code documentation, system architecture documentation, and user manuals to facilitate ongoing maintenance and development.

4.Diagrams

1) Use Case Diagram :-

A diagram of a diagram

Description automatically generated

2) Class Diagram:-

A diagram of a software application

Description automatically generated with medium confidence

3) Sequence Diagram:-

A diagram of a diagram

Description automatically generated

A diagram of a diagram

Description automatically generated

4)Activity Diagram:-

A diagram of a flowchart

Description automatically generated