Software Requirements Specification (SRS):-

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the development of an eCommerce application. It outlines the functionalities, interfaces, and constraints of the system to be developed.

1.2 Scope

The eCommerce application will allow users to browse products, add them to a shopping cart, and complete purchases securely. It will include user registration, product management, order processing, and payment integration features.

1.3 Definitions, Acronyms, and Abbreviations

eCommerce: Electronic Commerce

SRS: Software Requirements Specification

UI: User Interface

API: Application Programming Interface

SSL: Secure Sockets Layer

1.4 References

List any references or documents consulted during the creation of this SRS.

1.5 Overview

The remainder of this document will detail the specific requirements, including functional and non-functional requirements, use cases, external interfaces, and system architecture.

2. Overall Description

2.1 Product Perspective

The eCommerce application will serve as a standalone system interacting with users through a web-based UI. It may integrate with external services for payment processing and shipping, but the core functionalities will be self-contained.

2.2 Product Features

User registration and authentication

Product browsing and searching

Shopping cart management

Checkout and payment processing

Order management for users and administrators

Product management for sellers/vendors

2.3 User Classes and Characteristics

Buyers: Users who browse products and make purchases.

Sellers: Users who list and manage products for sale.

Administrators: Users who manage the overall system, including user accounts, orders, and products.

2.4 Operating Environment

The eCommerce application will be accessible via standard web browsers on desktop and mobile devices. It will run on web servers with support for server-side scripting languages such as Node.js, Python, or Java.

2.5 Design and Implementation Constraints

The application must comply with relevant regulations and standards for eCommerce security and privacy.

The UI must be responsive and accessible across different devices and screen sizes.

Performance and scalability considerations must be taken into account for handling concurrent user requests and large volumes of data.

2.6 Assumptions and Dependencies

The development team will have access to necessary resources, including hardware, software, and development tools.

External services for payment processing and shipping integration will be available and properly configured.

3. Specific Requirements

3.1 External Interface Requirements

User Interfaces: The UI must provide intuitive navigation, clear product displays, and secure payment processing.

Hardware Interfaces: The application will run on standard web servers and utilize databases for data storage.

Software Interfaces: Integration with external services such as payment gateways and shipping APIs will be required.

Communication Interfaces: Secure communication protocols (e.g., HTTPS) will be used for data exchange between the client and server.

3.2 Functional Requirements

User Registration: Users must be able to register accounts with valid email addresses and passwords.

Product Browsing: Users should be able to browse products by category, search for specific items, and view product details.

Shopping Cart Management: Users must be able to add items to a shopping cart, update quantities, and remove items.

Checkout: Users should be guided through a secure checkout process, including entering shipping and payment information.

Order Processing: Orders must be recorded and processed, with email notifications sent to users upon completion.

Product Management: Sellers should be able to add, edit, and remove products from their inventory.

3.3 Non-Functional Requirements

Performance: The application must be responsive, with fast page load times and minimal latency during peak traffic.

Security: User data, including passwords and payment information, must be encrypted and stored securely. Secure authentication mechanisms must be implemented to prevent unauthorized access.

Scalability: The system should be designed to handle increasing numbers of users and products without significant degradation in performance.

Reliability: The application should be robust and fault-tolerant, with mechanisms in place for error handling and recovery.

4. Use Cases

4.1 Use Case Diagram:

• The use case diagram illustrates the interaction between users (Buyer and Seller) and the system.

4.2 Use Case Descriptions:

4.2.1 Buyer Use Cases:

• UC1: Register Account

• Description: Buyer registers an account to access the eCommerce platform.

• Actors: Buyer

• Preconditions: None

• Main Flow:

1. Buyer navigates to the registration page.

2. Buyer fills in the required information.

3. Buyer submits the registration form.

4. System validates the information and creates a new account.

5. System sends a confirmation email.

• UC2: Browse Products

• Description: Buyer searches and views products available on the platform.

• Actors: Buyer

• Preconditions: Buyer is logged in.

• Main Flow:

1. Buyer navigates to the product browsing section.

2. Buyer searches for products using filters or keywords.

3. Buyer views product listings.

• UC3: Add to Cart

• Description: Buyer adds products to the shopping cart for later purchase.

• Actors: Buyer

• Preconditions: Buyer is logged in.

• Main Flow:

1. Buyer selects a product.

2. Buyer clicks on the "Add to Cart" button.

3. System adds the product to the shopping cart.

• UC4: Checkout

• Description: Buyer completes the purchase process.

• Actors: Buyer, Payment Gateway

• Preconditions: Buyer is logged in and has items in the shopping cart.

• Main Flow:

1. Buyer navigates to the checkout page.

2. Buyer reviews the shopping cart contents.

3. Buyer enters shipping and payment information.

4. Buyer confirms the purchase.

5. System processes the payment via the Payment Gateway.

6. System sends a confirmation email to the buyer.

4.2.2 Seller Use Cases:

• UC1: Register Seller Account

• Description: Seller registers an account to become a vendor on the eCommerce platform.

• Actors: Seller

• Preconditions: None

• Main Flow:

1. Seller navigates to the registration page.

2. Seller fills in the required information.

3. Seller submits the registration form.

4. System validates the information and creates a new seller account.

5. System sends a confirmation email.

• UC2: Add Product

• Description: Seller adds new products to their store.

• Actors: Seller

• Preconditions: Seller is logged in.

• Main Flow:

1. Seller navigates to the product management section.

2. Seller selects the option to add a new product.

3. Seller fills in product details (name, description, price, etc.).

4. Seller uploads product images.

5. Seller saves the product information.

• UC3: Manage Orders

• Description: Seller manages orders placed by buyers.

• Actors: Seller

• Preconditions: Seller is logged in and has received orders.

• Main Flow:

1. Seller navigates to the order management section.

2. Seller views a list of orders.

3. Seller selects an order to view details.

4. Seller updates the order status (e.g., shipped, delivered).

• UC4: Withdraw Funds

• Description: Seller requests withdrawal of funds from their sales.

• Actors: Seller, Payment Gateway

• Preconditions: Seller is logged in and has available funds.

• Main Flow:

1. Seller navigates to the funds management section.

2. Seller selects the option to withdraw funds.

3. Seller enters the withdrawal amount and payment details.

4. Seller submits the withdrawal request.

5. System processes the withdrawal via the Payment Gateway.

6. Seller receives confirmation of the withdrawal.

5. External Interface Specifications

5.1 User Interfaces:

• The user interfaces include web pages for buyers and sellers, as well as administrative interfaces for managing the platform.

• The design should be intuitive, responsive, and accessible across different devices and screen sizes.

5.2 Hardware Interfaces:

• The application will interact with standard hardware components such as servers, databases, and networking equipment.

5.3 Software Interfaces:

• The application will integrate with external systems for payment processing, shipping, and possibly third-party APIs for additional functionality.

• APIs for authentication, product search, and order management may also be required.

5.4 Communication Interfaces:

• The application will communicate with users via email for account registration, order confirmation, and other notifications.

• Secure communication protocols (HTTPS) will be used for data exchange between the client and server.

6. Technical Specifications

6.1 System Architecture:

• The application will be built using a scalable and modular architecture, possibly following a microservices pattern.

• Components may include frontend (HTML/CSS/JavaScript), backend (Node.js, Django, etc.), and database (MySQL, PostgreSQL, etc.).

6.2 Database Design:

• The database will store information about users, products, orders, and transactions.

• Entity-Relationship Diagrams (ERDs) and database normalization principles will guide the design.

6.3 Programming Languages and Technologies:

• Frontend: HTML, CSS, JavaScript (React, Angular, Vue.js)

• Backend: Node.js, Python (Django,