**Online Shopping System**

**Author: Lokesh kumar S**

**1. Abstraction 3**

**2. Introduction 3**

2.1 Obective …. 4

2.2 Scope ……. 4

**3. Functional Requirements 3**

3.1 User Management 3

3.2 Product Management 4

3.3 Shopping Cart 4

3.4 Checkout Process 4

3.5 Order Management 4

**4. Non-Functional Requirement 4**

4.1 Performance Management 4

4.2 Security Management 4

4.3 Usability Management 4

**5. Design 5**

5.1 High Level Design 5

5.1.1 Presentation Layer 5

5.1.2 Business Logic Layer 5

5.1.3 Data Access Layer 5

5.2 Low Level Design 5

5.1.1 Class Structure 6

5.2.2 Database Schema 6

5.2.3 Algorithm 6

**6. Flow Chart 6**

6.1 Use Case Diagram 6

6.2 Class Diagram 7

6.3 Sequence Diagram 7

**7. Testcase 8**

7.1 User Registration and Login Processes 8

7.2 Shopping Cart Management 8

7.3 Checkout Procedures 8

7.4 Order Management Functionalities 8

* **Abstraction**

The Online Shopping System (OSS) Software Requirements Specification (SRS) report outlines the essential specifications and requirements needed to develop a dependable and user friendly online shopping platform. In the current digital era, e-commerce has become an essential part of the retail environment.

This SRS document offers the framework for creating an efficient and user-friendly online purchasing experience for both administrators and customers. This project's main goal is to design, develop, and implement an online shopping system that satisfies the diverse demands of modern consumers.

This system will offer a wide range of products, user-friendly interfaces, secure payment options, and efficient order management for clients. It will also give administrators access to comprehensive inventory and sales administration tools.

* **Introduction**
* **Objective:**

The objective of the project is to build an Online Shopping System (OSS) that aims to provide a convenient and user-friendly platform for customers to browse, select and purchase products through online.

* **Scope:**

The Online Shopping System will include features for both customers and administrators. Customers will be able to browse products, add items to their cart, make purchase, and track their orders. Administrators will have access to features such as managing product listing, processing orders, and monitoring inventory.

* **Functional Requirements**
* **User Management:**

The system should allow users to register by providing necessary details such as full name, email address, and password. Upon successful registration, users should receive a confirmation email. Registered users should be able to securely log in using their credentials and log out when necessary. Administrators should have the capability to manage user accounts.

* **Product Management:**

Administrators should be able to add new products to the system, specifying details like name, description, price, category, and quantity. They should also have an ability to edit existing product information and remove products that are no longer available. The system should support image uploads for product association.

* **Shopping Cart:**

Users should be able to add products to their shopping carts from product listings or product detail pages. They should also have the option to remove products from their carts or update the quantity of items. The cart summary should reflect any changes made by the user.

* **Checkout Process:**

Users should be guided through a seamless checkout process, allowing them to review their cart contents, select a payment method, and enter shipping information. They should have the opportunity to confirm their order before finalizing the purchase. The system should support multiple payment methods for user convenience.

* **Order Management:**

Administrators should be able to view a list of all orders placed on the platform, including order details and status. They should have the ability to process orders by updating their status (e.g., pending, processing, shipped, delivered) based on order fulfillment. Additionally, administrators should be able to manage inventory levels to ensure product availability and prevent overselling.

* **Non-Functional Requirements**
* **Performance Management:**

The system should respond promptly to user actions, with page loads and form submissions taking no more than 2 seconds under normal load conditions. It should be capable of handling up to 1000 concurrent users without significant degradation in performance.

* **Security Management:**

**U**ser passwords and sensitive information should be securely encrypted using industry-standard hashing algorithms. Payment transactions should be conducted over secure connections using SSL/TLS encryption to protect sensitive data during transmission.

* **Usability Management:**

The user interface should be intuitive and easy to navigate, with clear labelling and visual cues. Error messages should be informative, indicating the nature of the issue and providing guidance on how to resolve it.

* **Design**
* **High Level Design:**

The Online Shopping System's architecture consists of three main components: Presentation Layer, Business Logic Layer, and Data Access Layer.

* **Presentation Layer**

This layer handles the user interface components, including web pages, forms, and client-side interactions. It communicates user inputs and actions to the Business Logic Layer for processing. The Components in this layer include:

* Web pages for product listings, shopping cart, checkout, and user authentication.
* Client-side scripts for form validation and dynamic content rendering.
* UI components for user interaction and feedback.
* **Business Logic Layer:**

This layer contains the core application logic responsible for processing user requests and managing system functionalities. It interacts with both the Presentation Layer and the Data Access Layer to retrieve and manipulate data. The Components in this layer include:

* Controller classes or modules that handle user requests and orchestrate interactions between different components.
* Service classes responsible for business logic implementation, such as order processing, user authentication, and product management.
* Utility functions for tasks like validation, error handling, and encryption.
* **Data Access Layer:**

This layer manages interactions with the database, including querying and updating data. It provides an interface for the Business Logic Layer to access and manipulate persistent data. The Components in this layer include:

* Data access objects (DAOs) or repository classes that encapsulate database operations.
* Database connection management components for establishing and managing connections to the database.
* Entity classes representing database tables and their relationships.
* **Low Level Design:**

The Low-Level Design (LLD) of the Online Shopping System delves into the specifics of each component, detailing class structures, database schema, and algorithms used for implementing functionalities.

* **Class Structure:**

The class structures include essential classes such as User, Product, and Order. The User class manages user-related information and authentication processes, while the Product class handles details regarding the products available in the system, such as name, description, price, and quantity. Additionally, the Order class is responsible for managing order-related data, including order date, status, and total amount.

* **Database Schema:**

The database schema consists of tables that store relevant data for the system's functionalities. The Users table stores user details like userId, fullName, email, password, address, and phoneNumber. The Products table contains information about available products, including productId, name, description, price, and quantityAvailable. Lastly, the Orders table maintains order-related information, such as orderId, userId, orderDate, status, and totalAmount.

* **Algorithm:**

Algorithms are developed to support key functionalities of the system. The Order Processing Algorithm calculates the total amount for each order and updates the product quantity available after an order is placed. Similarly, the Inventory Management Algorithm ensures product availability by checking inventory levels before adding items to the cart or processing orders, updating inventory accordingly.

* **Flow Chart**
* **Use Case Diagram:**



* **Class Diagram:**



* **Sequence Diagram:**



* **Testcase**

Test cases will cover various scenarios to ensure the functionality and performance of the system. These will include testing user registration and login processes, shopping cart management, checkout procedures, order management functionalities, as well as performance and security testing under different load conditions.

* **User Registration and Login Processes**:

Test Case 1: Successful User Registration

Test Case 2: User Registration with Existing Email

Test Case 3: User Registration with Invalid Data

Test Case 4: User Registration Password Strength

Test Case 5: User Registration Successful Redirect

Test Case 6: Successful User Login

Test Case 7: Invalid User Login Credentials

Test Case 8: User Logout Functionality

* **Shopping Cart Management:**

Test Case 9: Adding Product to Cart

Test Case 10: Removing Product from Cart

Test Case 11: Updating Quantity in Cart

Test Case 12: Cart Persistence Across Sessions

* **Checkout Procedures:**

Test Case 13: Proceeding to Checkout from Cart

Test Case 14: Selecting Payment Method

Test Case 15: Entering Shipping Information

Test Case 16: Confirming Order Details

Test Case 17: Placing Order Successfully

Test Case 18: Checkout Process Cancelation

* **Order Management Functionalities:**

Test Case 19: Viewing List of Orders

Test Case 20: Processing Pending Orders

Test Case 21: Updating Order Status

Test Case 22: Managing Product Inventory Levels