**Software Requirements Specification (SRS) Document**
**Project: Online Shopping System**

---

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to define the functional, non-functional, and technical requirements for the Online Shopping System. The system will allow users to browse products, add them to a shopping cart, purchase items securely, and track orders.

## 1.2 Document Conventions

- Functional requirements are prefixed with **FR**.
- Non-functional requirements are prefixed with **NFR**.
- **[Prototype]** indicates features to be implemented in the initial submission.
- **[Future Version]** indicates features for future enhancements.

## 1.3 Intended Audience and Reading Suggestions

- **Developers:** Implementing the system.
- **Project Managers:** Tracking progress.
- **End Users:** Understanding system capabilities.
- **Testers:** Validating functionality.

## 1.4 Product Scope

This Online Shopping System provides an intuitive and secure shopping experience, allowing users to:

- Browse a catalog of products.
- Add/remove items from a cart.
- Make secure online payments. **[Prototype]**
- Track order history and shipments. **[Prototype]**
- AI-based product recommendations. **[Future Version]**

## 1.5 References

- IEEE 830-1998 Software Requirements Specification Standard.
- Java EE documentation for web applications.

---

# 2. Overall Description

## 2.1 Product Perspective

This system is a standalone web-based platform that integrates with third-party payment gateways and shipping services.

### 2.2 Product Functions

- User Registration & Authentication. **[Prototype]**
- Product Catalog Browsing. **[Prototype]**
- Shopping Cart Management. **[Prototype]**
- Secure Payment Processing. **[Prototype]**
- Order Tracking & Management. **[Prototype]**
- AI-based Chatbot for Customer Support. **[Future Version]**

### 2.3 User Characteristics

- **Customers:** Purchase products and track orders.
- **Admins:** Manage product listings, orders, and users.
- **Guest Users:** Browse the product catalog but require login for purchases.

### 2.4 Constraints

- System must be accessible via desktop and mobile browsers.
- Transactions must comply with PCI-DSS security standards.

### 2.5 Assumptions and Dependencies

- Users have stable internet access.
- Payment processing relies on third-party services (e.g., PayPal, Stripe).
- The system must support at least 500 concurrent users.

---

## 3. Specific Requirements

### 3.1 Functional Requirements

- **FR1**: Users must be able to create and manage accounts. **[Prototype]**
- **FR2**: Users should browse, search, and filter products. **[Prototype]**
- **FR3**: Users must be able to add/remove items from the shopping cart. **[Prototype]**
- **FR4**: The system must securely process payments through an integrated gateway. **[Prototype]**
- **FR5**: Users must receive order confirmation and tracking details. **[Prototype]**
- **FR6**: Admins must be able to add, edit, and remove products. **[Prototype]**
- **FR7**: The system must send email notifications for orders and payments. **[Prototype]**
- **FR8**: Users should be able to cancel orders within 24 hours. **[Future Version]**
- **FR9**: Integration with third-party logistics APIs for real-time tracking. **[Future Version]**
- **FR10**: AI-powered product recommendations. **[Future Version]**

### 3.2 Non-Functional Requirements

- **NFR1**: The system should have 99.9% uptime. **[Prototype]**
- **NFR2**: The response time for user actions should be less than 2 seconds. **[Prototype]**
- **NFR3**: All transactions must be secured with TLS 1.2+ encryption. **[Prototype]**
- **NFR4**: The system must be scalable to accommodate 10,000 daily transactions. **[Future Version]**
- **NFR5**: Compliance with GDPR and data protection laws. **[Future Version]**

---

## 4. External Interface Requirements

### 4.1 User Interfaces

- Responsive web interface with modern UI. **[Prototype]**
- Mobile-friendly design for smartphones and tablets. **[Future Version]**

### 4.2 Hardware Interfaces

- The system must support cloud hosting with at least 4-core processors and 16GB RAM. **[Prototype]**

### 4.3 Software Interfaces

- **Frontend:** ReactJS, HTML, CSS. **[Prototype]**
- **Backend:** Spring Boot (Java EE), MySQL database. **[Prototype]**
- **External APIs:** PayPal, Stripe, UPS tracking. **[Future Version]**

### 4.4 Communication Interfaces

- RESTful APIs for all external integrations. **[Prototype]**
- Email notifications through SMTP services. **[Prototype]**

---

## 5. Other Requirements

### 5.1 Security Requirements

- Passwords must be encrypted using bcrypt. **[Prototype]**
- Users must be logged out after 15 minutes of inactivity. **[Prototype]**

### 5.2 Performance Requirements

- The homepage should load within 3 seconds. **[Prototype]**
- Payment transactions should complete within 5 seconds. **[Prototype]**

### 5.3 Availability & Reliability

- Daily backups must be taken to prevent data loss. **[Prototype]**
- System should support disaster recovery within 12 hours. **[Future Version]**

### 5.4 Maintainability & Scalability

- System should be modular to allow easy updates. **[Prototype]**
- It should support deployment on AWS and Azure. **[Future Version]**

---

## 6. Appendix

- Use Case Diagrams
- Sequence Diagrams
- Class Diagrams
- Data Flow Diagrams
- ER Model of the Database

**End of Document**