Software Requirements Specification (SRS)

# E-commerce Website Prototype

## 1. Introduction

### 1.1 Purpose

This document provides a prototype of the Software Requirements Specification (SRS) for an E-commerce website that allows customers to browse, purchase, and track products online. It also provides tools for administrators to manage products, orders, and customers.

### 1.2 Scope

The system includes:  
- A customer portal: browse products, cart, checkout, payment, and order tracking.  
- An admin dashboard: product, inventory, and order management.  
- Integration with external services: payment gateways, email, SMS.

### 1.3 Intended Audience

- Developers  
- Testers  
- Project Manager  
- Stakeholders  
- Designers

## 2. Overall Description

### 2.1 Product Perspective

The system is a web-based application built on a client-server architecture with REST APIs and a relational database.

### 2.2 Users

|  |  |
| --- | --- |
| User Type | Description |
| Customer | End-users browsing and purchasing products |
| Admin | Users who manage products, inventory, and orders |
| Super Admin | Full control of all system modules |

### 2.3 Constraints

- GDPR compliance  
- HTTPS for all communication  
- Mobile-friendly design

## 3. Functional Requirements

### 3.1 Customer Features

- User registration & login  
- Browse/search/filter products  
- View product details & reviews  
- Manage shopping cart & wishlist  
- Checkout and make payment  
- Track order status  
- Rate & review purchased products

### 3.2 Admin Features

- Manage users & roles  
- Add/edit/delete products & categories  
- Manage stock levels  
- Process and track orders  
- Generate sales & inventory reports

### 3.3 System Features

- Email/SMS notifications  
- SEO-friendly URLs  
- Discount & coupon support  
- Analytics integration

## 4. Non-Functional Requirements

|  |  |
| --- | --- |
| Attribute | Requirement |
| Performance | Page loads < 2s |
| Security | Mitigate OWASP Top 10 threats |
| Scalability | Support for 10,000 concurrent users |
| Availability | 99.9% uptime |
| Usability | WCAG 2.1 AA accessible |
| Backup/Recovery | Daily backups & disaster recovery |

## 5. External Interfaces

### 5.1 User Interfaces

- Responsive web UI for customers & admin  
- Mobile-friendly layouts

### 5.2 Software Interfaces

- Database: MySQL/PostgreSQL  
- Payment Gateway: Stripe/PayPal/Razorpay  
- Email/SMS: SendGrid/Twilio

### 5.3 Communication Interfaces

- REST APIs  
- Webhooks for notifications

## 6. Recommended Tools & Technologies

|  |  |
| --- | --- |
| Layer | Suggested Tools |
| Frontend | React.js, TailwindCSS |
| Backend | Node.js + Express |
| Database | PostgreSQL |
| Hosting | AWS, Azure, or GCP |
| Authentication | JWT & OAuth2 |
| Payments | Stripe/Razorpay |
| DevOps | Docker, Jenkins |
| Version Control | Git + GitHub |

## 7. Future Enhancements

- Mobile apps (iOS/Android) with Flutter/React Native  
- AI-based product recommendations  
- Multi-vendor marketplace  
- Loyalty programs

## 8. Appendices

- Glossary: SKU, API, GDPR, etc.  
- References: IEEE 830-1998, W3C guidelines  
- Mockups, diagrams, and ERD to be developed in the next phase

NOTE: This is a prototype document meant to illustrate the high-level structure and contents of an SRS. Detailed specifications, diagrams, and test cases will be added during the design and development phases.