Software Requirements Specification (SRS)

Full Stack .NET E-Commerce Web Application

1. Introduction:
E-Commerce project for Electronics market

1.1 Purpose

This document provides a comprehensive Software Requirements Specification (SRS) for the development of a **.NET Full-Stack E-Commerce Web Application**. The system aims to facilitate online shopping by allowing users to browse, purchase products, and manage orders efficiently.

1.2 Scope

The e-commerce application will be developed using **ASP.NET Core (backend)** and **React/Angular (frontend)** with a **SQL Server database**. The system will provide features such as:

- User registration and authentication
- Product listing and search
- Shopping cart and checkout
- Order management
- Payment integration
- Admin panel for inventory management

1.3 Intended Audience and Usage

This document is intended for **developers, testers, project managers, and stakeholders** involved in the development and deployment of the system.

1.4 Definitions and Acronyms

- **ASP.NET Core**: A cross-platform framework for building web applications.
- **SQL Server**: A relational database management system.
- **JWT (JSON Web Token)**: Used for authentication.
- **API**: Application Programming Interface.

---

2. Overall Description

2.1 Product Perspective

The e-commerce web application is a **standalone system** designed to manage an online store. It will have both **customer-facing and admin-facing interfaces**.

2.2 Product Features

1. **User Management**
    - User registration and login (OAuth, JWT authentication)
    - Profile management
2. **Product Management**
    - Categories and subcategories
    - Product search and filtering
3. **Shopping Cart & Checkout**
    - Add, remove, and update items in the cart
    - Checkout process with payment integration
4. **Order Management**
    - Track order status
    - View order history
5. **Admin Dashboard**
    - Manage users, products, and orders
    - Inventory and stock management
6. **Security & Performance**
    - Secure API endpoints
    - Scalability considerations

2.3 Assumptions and Dependencies

- The system assumes a stable internet connection.
- Third-party payment gateways (PayPal, Stripe) will be used.
- The frontend will consume RESTful APIs from the backend.

---

3. Functional Requirements

3.1 User Management

- Users should be able to register and log in securely.
- Passwords should be stored using encryption.

3.2 Product Listing & Search

- Users can browse products by category, price, and brand.
- The system should support keyword-based search.

3.3 Shopping Cart & Checkout

- Users can add/remove items from the cart.
- The checkout process should integrate a payment gateway.

3.4 Order Processing

- Users should be able to view their order history.
- Admins should be able to update order statuses.

3.5 Admin Panel

- Only authorized admins should have access.
- Admins should be able to add/edit/delete products.

---

4. Non-Functional Requirements

4.1 Performance Requirements

- The system should handle at least **500 concurrent users**.
- API response times should not exceed **2 seconds**.

4.2 Security Requirements

- All sensitive data should be encrypted.
- Users should be authenticated using **JWT tokens**.

4.3 Usability Requirements

- The UI should be **mobile-friendly and responsive**.
- Navigation should be intuitive.

4.4 Scalability Requirements

- The system should be scalable to support increasing product listings and user traffic.

---

5. System Design

5.1 Architecture

- **Frontend**: React/Angular
- **Backend**: ASP.NET Core Web API
- **Database**: SQL Server
- **Authentication**: JWT-based authentication
- **Hosting**: IIS

5.2 API Endpoints

Endpoint	Method	Description
/api/users/register	POST	Register a new user
/api/users/login	POST	Authenticate user
/api/products	GET	Get all products
/api/cart/add	POST	Add an item to the cart
/api/orders	GET	Get order history

---

6. Appendix

- **Technology Stack**: ASP.NET Core, React/Angular, SQL Server
- **Development Tools**: Visual Studio, Git, Docker
- **Third-Party Integrations**: Stripe, PayPal, Authentication

This document serves as a foundation for the development and implementation of the e-commerce system.