



Software Requirements Specification (SRS)

For LK-PC-Solution (Computer Items E-Commerce Website)

Version 1.0

Created By :- W. M. Lakshan Dananjana Weerasinghe

Contents

1. Introduction	3
1.1 Purpose.....	3
1.2 Scope	3
1.3 Definitions	3
2. Overall Description	3
2.1 User Needs	3
2.2 Assumptions.....	3
2.3 Constraints	3
3. System Features	4
3.1 User Registration & Authentication	4
3.2 Product Catalog	4
3.3 Shopping Cart & Checkout.....	4
3.4 Order Management	4
3.5 Admin Dashboard	4
4. External Interface Requirements	5
4.1 User Interface.....	5
4.2 APIs	5
5. Non-Functional Requirements.....	5
5.1 Performance.....	5
5.2 Security	5
5.3 Compatibility.....	5
6. Technologies Used.....	5
7. Glossary	6
8. Next Steps.....	6

1. Introduction

1.1 Purpose

This document outlines the requirements for the LK-PC-Solution web application, a platform for selling computer hardware and accessories. It defines the scope, features, and technical specifications of the system.

1.2 Scope

The system will:

- Allow customers to browse and purchase computer items (e.g., CPUs, GPUs, keyboards, monitors).
- Provide administrators with tools to manage products, orders, and user accounts.
- Include a shopping cart, checkout system, and order tracking.

1.3 Definitions

- **User:** A registered or guest customer.
- **Admin:** A user with privileges to manage products and orders.
- **Product:** A computer item listed for sale.

2. Overall Description

2.1 User Needs

- **Customers:** Browse products, add items to cart, place orders, and track deliveries.
- **Admins:** Add/update products, manage inventory, and view sales reports.

2.2 Assumptions

- Users have access to a modern web browser (Chrome, Firefox, Edge).

2.3 Constraints

- Limited to desktop browsers (no dedicated mobile app).
- Built using HTML, CSS, JavaScript (frontend) and PHP (backend).

3. System Features

3.1 User Registration & Authentication

- **Description:** Users can create accounts, log in, and reset passwords.
- **Requirements:** Secure password hashing (e.g., PHP password_hash()).
- Guest checkout option.

3.2 Product Catalog

- **Description:** Display computer items with details (price, specs, images).
- **Requirements:** Product categorization (e.g., CPUs, GPUs, Peripherals).
- Search and filter functionality (by price, brand, category).

3.3 Shopping Cart & Checkout

Description: Users can add items to a cart and complete purchases.

- **Requirements:** Cart persistence using PHP sessions or cookies.
- Order confirmation email (PHP mail function or SMTP).

3.4 Order Management

- **Description:** Admins can view and update order statuses (e.g., "Processing," "Shipped").
- **Requirements:** Order history page for users.
- Export orders to CSV/PDF (optional).

3.5 Admin Dashboard

- **Description:** Admins manage products, users, and view sales analytics.
- **Requirements:** CRUD (Create, Read, Update, Delete) operations for products.
- Basic sales reports (total revenue, popular products).

4. External Interface Requirements

4.1 User Interface

- **Homepage:** Featured products, categories, and promotions.
- **Product Page:** High-quality images, specifications, and "Add to Cart" button.
- **Admin Dashboard:** Tables for products/orders, forms for adding new items.

4.2 APIs

- **Email Service:** PHP Mailer/SendGrid for notifications.

5. Non-Functional Requirements

5.1 Performance

- Page load time < 3 seconds.
- Support for 200+ concurrent users.

5.2 Security

- SQL injection prevention (PHP prepared statements).
- Cross-Site Scripting (XSS) protection (input sanitization).
- SSL certificate for HTTPS.

5.3 Compatibility

- Responsive design for desktop and tablet screens.

6. Technologies Used

- **Frontend:** HTML5, CSS3, JavaScript (Vanilla JS or jQuery).
- **Backend:** PHP (with MySQL).
- **Hosting:** Apache (e.g., XAMPP).

7. Glossary

- **CRUD:** Create, Read, Update, Delete operations.
- **SSL:** Secure Sockets Layer (encryption protocol).

8. Next Steps

- **Database Design:** Create tables for users, products, orders, and cart.
- **Payment Integration:** Implement PayPal/Stripe API in PHP.
- **Testing:** Validate user flows (registration, checkout, admin tasks).