

**Project Overview**

**Course: SCD**

**Submitted to: Maam Fatima Gillani**

**Submitted by: Adil Bilal (FL21499)**

**Sana Bushra (FL21526)**

**Section: BSSE-5B**

**Software Requirements Specification (SRS) Document for "EZCart" E-Commerce Platform**

**1. Project Overview**

**1.1 Main Goal of the Project**

The main goal of the **EZCart** project is to develop a user-friendly, scalable, and secure e-commerce platform that enables businesses to create and manage online stores efficiently. The platform will provide customers with a seamless shopping experience, including product browsing, purchasing, and order tracking.

**1.2 Problem the Software Aims to Solve**

EZCart aims to address the following challenges:

* **For Businesses**: Difficulty in setting up and managing an online store due to complex tools and high costs.
* **For Customers**: Lack of intuitive and reliable platforms for online shopping, leading to poor user experiences.
* **For Both**: Inefficient order management, payment processing, and inventory tracking.

EZCart will simplify the process of creating and managing an online store while providing customers with a fast, secure, and enjoyable shopping experience.

**2. Scope and Objectives**

**2.1 Scope**

The scope of the EZCart project includes:

* Development of a web-based e-commerce platform.
* Features for businesses to manage products, inventory, orders, and customer data.
* Features for customers to browse products, add items to a cart, make purchases, and track orders.
* Integration with payment gateways for secure transactions.
* Responsive design for accessibility on desktop and mobile devices.
* Admin dashboard for store management and analytics.

**2.2 Objectives**

* Provide an intuitive interface for businesses to set up and manage their online stores.
* Deliver a seamless and secure shopping experience for customers.
* Ensure scalability to handle growing numbers of users and transactions.
* Implement robust security measures to protect user data and transactions.
* Optimize performance for fast loading times and smooth navigation.

**3. Technical Requirements**

**3.1 Technologies, Programming Languages, and Databases**

* **Frontend**:
  + Java GUI for building the desktop-based user interface.
* **Backend:**
  + Java for server-side logic and business operations.
* **Version Control**:
  + Git and GitHub for version control and collaboration.

**3.2 Justification for Technology Choices**

* **Java GUI**: Java provides a robust framework for building cross-platform desktop applications with a rich user interface.
* **Java**: Java is a mature, scalable, and secure programming language, making it ideal for building enterprise-level backend systems.
* **Git and GitHub**: Industry-standard tools for version control and collaborative development.

**4. Functional Requirements**

**4.1 User Functionalities**

**For Customers:**

1. **User Registration and Login**:
   * Create an account, log in, and manage profile information.
2. **Product Browsing**:
   * Search and filter products by category, price, ratings, and availability.
3. **Shopping Cart**:
   * Add/remove products, update quantities, and view the total cost.
4. **Checkout and Payment**:
   * Select shipping options, enter payment details, and complete purchases.
   * Enable registered users to complete purchases with a single click using saved payment and shipping details.
5. **Order Tracking**:
   * View order status and history.
   * Integrate with logistics APIs to provide real-time updates on order location and delivery status.
6. **Product Reviews and Ratings**:
   * Leave reviews and ratings for purchased products.

**For Businesses (Admin):**

1. **Store Management**:
   * Add, update, and delete products.
   * Manage inventory and product categories.
   * Predict inventory needs and suggest restocking based on sales trends and seasonality using AI-Powered Inventory Management.
2. **Order Management**:
   * View and process customer orders.
   * Update order status (e.g., shipped, delivered).
   * Automate order status updates and notifications using predefined rules.
3. **Customer Management**:
   * View customer details and order history.
4. **Analytics and Reporting**:
   * Generate sales reports and view performance metrics.

**For System:**

1. **Payment Gateway Integration**:
   * Process payments securely and provide transaction records.
   * Allow customers to pay in their local currency with automatic conversion.
2. **Email Notifications**:
   * Send order confirmation, shipping updates, and promotional emails.
   * Use customer data to send personalized product recommendations and offers.
3. **Security**:
   * Implement user authentication, data encryption, and secure payment processing.
   * Allow users to log in using fingerprint or facial recognition for enhanced security.

**5. Non-Functional Requirements**

**5.1 Performance**

* The platform should handle up to **10,000 concurrent users** without performance degradation.
* Page load times should not exceed **3 seconds** under normal conditions.
* Ensure smooth and responsive performance.

**5.2 Scalability**

* The system should be designed to scale horizontally to accommodate increasing numbers of users and transactions.

**5.3 Security**

* Implement **SSL/TLS encryption** for secure data transmission between the frontend, backend, and database.
* Use **strong password policies** and **encryption** for storing user credentials.
* Regularly update the system to patch vulnerabilities and address security threats.

**5.4 Usability**

* The system should have an intuitive and user-friendly interface.
* Provide clear navigation and accessibility features for all users.
* Ensure the platform is easy to learn and use for both customers and business administrators.

**5.5 Reliability**

* Ensure **99.9% uptime** for the platform to minimize downtime and ensure availability.
* Implement **backup and recovery mechanisms** to prevent data loss in case of system failures.
* Conduct regular testing to identify and resolve potential issues before they impact users.

**5.6 Compatibility**

* The system should be compatible with major operating systems (Windows, macOS, Linux).
* Test the platform on different hardware configurations to ensure broad compatibility.

**5.7 Maintainability**

* Follow **modular coding practices** in Java to make the system easy to update and maintain.
* Provide comprehensive **documentation** for developers, including code comments, user manuals, and technical guides.
* Use **Git and GitHub** for version control to track changes and collaborate effectively.

**Conclusion**

The EZCart e-commerce platform aims to revolutionize online shopping by providing businesses with an easy-to-use tool for managing their stores and offering customers a secure and enjoyable shopping experience. By adhering to the outlined functional and non-functional requirements, EZCart will deliver a scalable, reliable, and user-friendly solution for the e-commerce industry.