**THE SOULED SOUL CLONE**

**SOFTWARE REQUIREMENT SPECIFICATION**

DONE BY

THILAGAVATHY

ANUJ

ADITYA

PRITHAK

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **S.N.** | **CONTENTS** |
| 1. | INTRODUCTION |
| 2. | INTERFACE REQUIREMENTS |
| 3. | PERFORMANCE REQUIREMNETS |
| 4. | DESIGN CONSTRAINTS |
| 5. | FUNCTIONAL REQUIREMENTS |
| 6. | NON-FUNCTIONAL REQUIREMENTS |
| 7. | PRELIMINARY SCHEDULE AND BUDGET |
| 8. | USES OF SRS |
| 9. | CONCLUSION |

**Introduction:**

* The Soul Store is an eCommerce platform designed to provide a seamless shopping experience for clothing enthusiasts.
* The Soul Store aims to offer a wide range of clothing items to customers, ensuring quality, variety, and convenience in their shopping experience. Users can browse, select, purchase, and track orders effortlessly through the platform.

**Interface Requirements**

* Intuitive and responsive UI design.
* Consistent layout and navigation.
* Mobile-friendly design for seamless browsing on smartphones and tablets.

**Performance Requirements**

* Fast loading times for product pages and checkout process.
* Scalability to handle a large number of concurrent users.
* Efficient database queries for retrieving product information and order details.

**Design Constraints**

* Use of React for frontend development.
* Integration with Tailwind CSS for UI styling.
* Compatibility with modern web browsers (Chrome, Firefox, Safari).

**Functional Requirements:**

1. **User Authentication**:
   * Users should be able to register with the platform by providing necessary information such as name, email address, and password.
   * Registered users should be able to log in securely using their credentials.
   * Users should have the option to reset their passwords in case they forget them.
   * Registered users should have the ability to manage their accounts, including updating personal information and changing passwords.
2. **Product Catalog**:
   * The platform should categorize clothing items into various sections such as tops, bottoms, dresses, and accessories for easy navigation.
   * Each product should have a detailed description, including price, size options, color variants, and available quantities.
   * Users should be able to view high-quality images of products from different angles.
   * Products should be organized in a visually appealing manner to enhance the browsing experience.
3. **Search Functionality**:
   * Users should be able to search for specific products using keywords entered into a search bar.
   * Advanced search options should allow users to filter products by categories, sizes, colors, brands, and price ranges.
   * Search results should be displayed accurately and efficiently, with relevant products appearing at the top of the list.
4. **Recommendation System**:
   * The platform should analyze user preferences, purchase history, and browsing behavior to generate personalized product recommendations.
   * Recommended products should be displayed prominently on the homepage, product pages, and during the checkout process.
   * Users should have the option to provide feedback on recommended products to further refine future recommendations.
5. **Shopping Cart**:
   * Users should be able to add products to their shopping carts with a single click.
   * The shopping cart should display a summary of all selected items, including quantities and total prices.
   * Users should be able to remove items from their carts or update quantities as needed.
   * The shopping cart should be persistent across sessions, allowing users to resume shopping from where they left off.
6. **Checkout Process**:
   * The checkout process should be divided into multiple steps to guide users through the process smoothly.
   * Users should be prompted to provide shipping and billing information, select shipping methods, and choose payment options.
   * The platform should support various payment methods such as credit/debit cards, PayPal, and other online payment gateways.
   * Users should receive confirmation emails and order summaries upon successful completion of the checkout process.
7. **Order Management**:
   * Registered users should have access to their order history, including past purchases and current order statuses.
   * Users should be able to track the status of their shipments in real-time, with tracking information provided by the shipping carrier.
   * The platform should allow users to initiate returns or exchanges for products within a specified timeframe, with clear instructions and guidelines provided.
8. **Admin Panel**:
   * Administrators should have access to an administrative interface where they can manage products, orders, and user accounts.
   * The admin panel should allow administrators to add new products, update product details, and remove discontinued items.
   * Administrators should be able to view and process orders, including updating order statuses and managing inventory levels.
   * The admin panel should provide tools for monitoring user activity, resolving disputes, and addressing any technical issues that may arise.

**Non-Functional Requirements**

1. **Reliability**:
   * The platform should have a robust infrastructure and reliable hosting services to ensure 24/7 availability.
   * Measures should be in place to minimize downtime, such as regular maintenance schedules and proactive monitoring for potential issues.
   * Backup and recovery mechanisms should be implemented to safeguard against data loss in the event of system failures.
   * The platform should have a failover system in place to automatically switch to backup servers in case of hardware or network failures.
   * Performance metrics should be continuously monitored to identify and address any performance bottlenecks or reliability concerns promptly.
2. **Usability**:
   * The user interface should be designed with simplicity and clarity in mind, allowing users to navigate the platform effortlessly.
   * Intuitive design principles should be applied to ensure consistency across all pages and functionalities.
   * Clear and descriptive labels should be used for buttons, links, and form fields to minimize confusion and facilitate task completion.
   * The platform should provide helpful prompts and tooltips to guide users through complex processes such as the checkout process.
   * Accessibility features should be incorporated to accommodate users with disabilities, such as screen readers and keyboard navigation.
3. **Scalability**:
   * The system architecture should be designed to handle a large number of concurrent users and scale seamlessly as user traffic grows.
   * Load balancing techniques should be employed to distribute incoming traffic evenly across multiple servers and prevent overloading.
   * Database optimization strategies should be implemented to ensure efficient data storage and retrieval, even under high load conditions.
   * Cloud-based infrastructure solutions such as auto-scaling and elastic computing should be utilized to dynamically allocate resources based on demand.
   * Stress testing and performance tuning should be conducted regularly to assess the platform's scalability and identify potential scalability limitations.
4. **Security**:
   * Data encryption protocols such as SSL/TLS should be implemented to secure data transmission between users and the platform.
   * Strong authentication mechanisms, including multi-factor authentication, should be enforced to prevent unauthorized access to user accounts.
   * Role-based access control should be employed to restrict access to sensitive features and data based on user roles and permissions.
   * Regular security audits and vulnerability assessments should be conducted to identify and address potential security threats.
   * Compliance with industry standards and regulations such as GDPR and PCI-DSS should be ensured to protect user privacy and payment information.
   * Incident response procedures should be established to handle security breaches or data breaches effectively, including notification of affected users and authorities.

**Preliminary Schedule and Budget**

* Development Timeline: 3 months
* Budget Allocation: $50,000

**Uses of SRS Document**

* Serve as a reference for developers during the implementation phase.
* Provide a basis for testing and validation of the final product.
* Guide project management decisions related to scope, timeline, and budget.

**Conclusion**

* The Souled Store clone aims to replicate the seamless shopping experience provided by its inspiration. With a focus on functionality, usability, and performance, this web application will strive to meet the needs of clothing enthusiasts while providing a platform for efficient management and administration.
* Top of Form
* Bottom of Form