Software Requirements Specification (SRS) for Blisskart - Gifting Website

1. Introduction
   * Purpose
     + The purpose of this document is to provide a comprehensive outline of the requirements for the development of Blisskart, a gifting website.
   * Scope
     + Blisskart aims to provide users with a platform to browse, search, and purchase a wide range of gift products. Users can create accounts, add items to their cart, make payments, and track their orders.
   * Definitions, acronyms, and abbreviations
     + None.
2. Overall Description
   * Product Perspective
     + Blisskart will be a standalone gifting website built using React for the frontend, Node.js for the backend, and FirebaseDB for the database. It will interact with users through a web interface.
   * Product Functions
     + User registration and authentication
     + Product browsing and searching
     + Product categorization and filtering
     + Product details and descriptions
     + Shopping cart management
     + Payment processing
     + Order tracking and status updates
     + Wishlist and favorites management
     + User reviews and ratings
     + Gift wrapping and personalized messages
     + User account management
     + Admin features and tools
   * User Characteristics
     + The website will cater to both registered and non-registered users. Registered users will have additional features such as saved addresses and order history.
   * Constraints
     + The website should be compatible with modern web browsers and responsive across different devices.
   * Assumptions and Dependencies
     + It is assumed that users will have access to an internet connection and modern web browsers to access the website.
3. Specific Requirements
   * External Interface Requirements
     + User Interfaces
       - The website will have an intuitive and user-friendly interface allowing users to browse and interact with products, manage their accounts, and perform various actions.
     + Hardware Interfaces
       - The website will be accessed through standard web browsers on devices such as desktops, laptops, tablets, and smartphones.
     + Software Interfaces
       - The website will utilize React for the frontend and Node.js for the backend. FirebaseDB will be used as the database.
   * Functional Requirements
     + User Registration and Authentication
       - Users should be able to create accounts, log in, and log out.
       - User authentication should be secure and include password recovery options.
     + Product Browsing and Searching
       - Users should be able to browse products by categories, search for specific products, and view product details.
     + Product Categorization and Filtering
       - Products should be categorized into different categories and subcategories for easy navigation.
       - Users should be able to filter products based on various attributes such as price, popularity, and ratings.
     + Product Details and Descriptions
       - Each product should have a detailed description, images, pricing information, and customer reviews.
     + Shopping Cart Management
       - Users should be able to add products to their cart, update quantities, remove items, and proceed to checkout.
     + Payment Processing
       - The website should support secure payment processing using popular payment gateways.
     + Order Tracking and Status Updates
       - Users should be able to track the status of their orders and receive notifications regarding order updates.
     + Wishlist and Favorites Management
       - Users should be able to create and manage wishlists and mark products as favorites.
     + User Reviews and Ratings
       - Users should be able to leave reviews and ratings for products.
     + Gift Wrapping and Personalized Messages
     + Users should have the option to add gift wrapping and personalized messages to their orders.
     + User Account Management
     + Users should be able to manage their profile, update personal information, and view their order history.
     + Admin Features and Tools
     + An admin panel should be available for managing products, orders, user accounts, and website content.
   * Performance Requirements
     + The website should provide fast response times for browsing, searching, and processing user actions.
     + The system should be scalable to handle a large number of concurrent users.
     + The website should have high availability with minimal downtime.
   * Security Requirements
     + User data should be securely stored and transmitted.
     + Secure payment gateways should be used for handling financial transactions.
     + User authentication and authorization should be implemented to protect user accounts and data.
   * Reliability Requirements
     + The system should be reliable, with minimal errors and downtime.
     + Error handling and recovery mechanisms should be in place to handle unexpected situations.
   * Usability Requirements
     + The website should have an intuitive and user-friendly interface.
     + The design should be responsive, ensuring a consistent experience across different devices.
     + Accessibility features should be implemented to ensure usability for users with disabilities.
   * Compatibility Requirements
     + The website should be compatible with major web browsers such as Chrome, Firefox, Safari, and Edge.
     + The website should be responsive and optimized for various devices, including desktops, laptops, tablets, and smartphones.
   * Legal and Compliance Requirements
     + The website should comply with relevant data protection regulations and privacy laws.
     + Terms of service and a privacy policy should be provided and accessible to users.
   * Documentation Requirements
     + User manuals and guides should be provided to help users navigate and utilize the website effectively.
     + Technical documentation should be available for developers and system administrators.
4. System Models
   * Use Case Diagrams
   * Sequence Diagrams
   * Class Diagrams
5. Appendices
   * Glossary
   * Use Case Descriptions
   * Data Dictionary