**Software Requirements Specification (SRS)**

**Online Store System**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to specify the functional and non-functional requirements for an online store system that displays categories, subcategories, and products, allowing users to browse, like products, add items to a cart, authenticate via OTP, and complete purchases.

**1.2 Scope**

The system is a web-based e-commerce application that supports browsing product categories and subcategories, product display, user authentication by One-Time Password (OTP), cart management, order processing, and email notifications.

**1.3 Definitions**

* **SRS:** Software Requirements Specification
* **OTP:** One-Time Password – a temporary code for user authentication
* **User:** Customer accessing the online store
* **Admin:** User with permissions to manage the store’s data

**2. Overall Description**

**2.1 Product Perspective**

The product will be a responsive web application built with Next.js and PostgreSQL on the backend, using Drizzle ORM for database interaction.

**2.2 Product Functions**

Key features:

* Display product categories and subcategories
* Display products under categories or subcategories
* Allow users to like products
* Shopping cart with add, update quantity, remove features
* Authentication via OTP sent through email/SMS
* Order placement with confirmation email
* Admin features for managing categories, products, and orders (optional for now)

**2.3 User Classes and Characteristics**

* **Customer:** Browses and purchases products
* **Admin:** Manages product catalog and orders

**2.4 Operating Environment**

* PostgreSQL database
* Next.js server and client environment
* Web browsers (desktop and mobile) supporting modern Javascript and cookies
* SMTP server for sending emails
* Optional SMS gateway for OTP delivery

**2.5 Design and Implementation Constraints**

* Authentication only via OTP (no password login)
* HTTPS encryption required for all client-server communication
* Support for Hebrew and English language display
* Use Drizzle ORM to interact with database

**3. Specific Requirements**

**3.1 Functional Requirements**

**3.1.1 Browsing and Navigation**

* Display top-level categories
* Selecting a category directs to a subcategory page or directly to product listing if only one subcategory or none exists
* Display subcategories when multiple exist under a category
* Display products in a list with images and prices
* Product detail page with description, image(s), like button, add to cart button

**3.1.2 User Authentication and Authorization**

* User enters email or phone number to request OTP
* System generates OTP, associates it with the user or email/phone, and sends it out
* User submits OTP for verification
* Validate OTP matches, is unexpired, and unused
* Allow login and associate session with authenticated user

**3.1.3 Cart Management**

* User can add products to cart, specify quantity
* User can view cart contents with product details and total price
* Allow updating quantities or removing items
* Proceed to checkout from cart

**3.1.4 Order Processing**

* Collect necessary shipping/payment info on checkout page
* Confirm order and store order details in database
* Send confirmation email to user on successful order placement
* Mark order status appropriately (e.g., pending, shipped)

**3.1.5 Product Liking**

* Allow users to like/favorite products
* Store liked products per user
* Optionally provide interface for user to view liked products

**3.2 Non-functional Requirements**

**3.2.1 Performance**

* Loading times under 2 seconds for product and category pages under normal conditions

**3.2.2 Availability**

* 99% uptime availability of the service

**3.2.3 Security**

* All client-server communications secured over HTTPS
* OTP codes are single-use and expire after a short configurable lifetime
* Protection from common web security threats (CSRF, XSS)

**3.2.4 Scalability**

* System designed to support growth in users, products, and orders with minimal refactoring

**3.2.5 Maintainability**

* Clean, modular codebase using Next.js and ORM best practices for future extension

**4. Database Design and Relations**

**4.1 Tables**

* **users:** stores user information including email/phone and authentication status
* **categories:** product categories
* **subcategories:** subcategories linked to categories (optional)
* **products:** product details linked to either a category or subcategory
* **likes:** relation between users and liked products
* **carts:** active carts linked to users
* **cart\_items:** products with quantity in carts
* **orders:** order records linked to users
* **order\_items:** products and quantities in an order
* **otps:** OTP codes with associated user/email/phone, expiry, usage status

**4.2 Relationships**

* One-to-many: categories → subcategories
* One-to-many: categories/subcategories → products
* Many-to-many: users ↔ products through likes
* One-to-many: users → carts, carts → cart\_items
* One-to-many: users → orders, orders → order\_items
* One-to-many: users → otps

**5. Use Case Summaries**

* **Browsing Products:** User navigates categories and subcategories, views product listings and details
* **Authentication:** User enters email/phone, receives OTP, submits OTP, gains access
* **Managing Cart:** User adds/removes products, updates quantity, views cart summary
* **Checkout:** User completes purchase, receives confirmation email
* **Liking Products:** User favorites products for future reference

**6. Appendix**

**6.1 Technologies**

* Next.js (frontend & backend)
* PostgreSQL (database)
* Drizzle ORM (database queries & schema)
* SMTP for email
* SMS integration for OTP (optional)

**6.2 Glossary**

* OTP: One-Time Password
* ORM: Object Relational Mapper