# Detailed Functional Requirements for Best Brightness System

## 1. Customer Interface

### 1.1 Homepage

- The system shall display a responsive homepage built with Ionic and Angular.

- The homepage shall showcase featured products and current promotions.

- The system shall provide a search functionality for products.

- The system shall display a navigation menu with links to product categories, shopping cart, and user account.

### 1.2 User Account Management

- The system shall allow customers to create new accounts.

- The system shall provide secure login functionality for existing users.

- The system shall allow users to reset their passwords via email.

- The system shall provide an option for users to update their profile information.

### 1.3 Product Browsing and Search

- The system shall display products in a grid or list view, with options to sort and filter.

- Each product shall have a dedicated page showing detailed information, including images, description, price, and availability.

- The system shall provide a search function that allows users to find products by name, category, or keywords.

### 1.4 Shopping Cart

- The system shall allow users to add products to their shopping cart.

- Users shall be able to view, update quantities, and remove items from their cart.

- The system shall calculate and display the total cost, including any applicable taxes or shipping fees.

- The shopping cart shall persist across sessions for logged-in users.

### 1.5 Checkout Process

- The system shall provide a multi-step checkout process.

- Users shall be able to enter shipping information or select from saved addresses.

- The system shall integrate with secure payment gateways for processing transactions.

- After successful payment, the system shall generate and display an order confirmation.

### 1.6 Order Tracking

- The system shall provide an order history page for users to view their past and current orders.

- Each order shall have a detailed view showing order status, items purchased, and shipping information.

- The system shall send email notifications to users about order status updates.

## 2. Cashier Interface

### 2.1 Cashier Authentication

- The system shall provide a secure login page for cashiers.

- Cashier accounts shall have limited permissions compared to admin accounts.

### 2.2 Cashier Dashboard

- Upon login, the system shall display a dashboard showing daily sales overview and quick access to sales functions.

### 2.3 Sales Management

- The system shall provide a point-of-sale (POS) interface for cashiers to process in-store purchases.

- Cashiers shall be able to scan product barcodes or manually search and add products to the cart.

- The system shall allow cashiers to apply discounts to individual items or the entire order.

- The system shall support multiple payment methods (cash, card) and calculate change for cash transactions.

- After completing a sale, the system shall generate a receipt/invoice that can be printed or emailed to the customer.

### 2.4 Sales Reports

- Cashiers shall be able to view daily sales reports.

- The system shall provide options to view weekly and monthly sales summaries.

## 3. Admin Interface

### 3.1 Admin Authentication

- The system shall provide a secure login page for admin users with enhanced security measures (e.g., two-factor authentication).

### 3.2 Admin Dashboard

- The admin dashboard shall display key metrics including total sales, inventory status, and recent customer orders.

- The dashboard shall provide quick links to all admin functions.

### 3.3 Inventory Management

- Admins shall be able to add new products to the inventory, including details such as name, description, price, and stock quantity.

- The system shall allow updating existing product information and stock levels.

- The system shall generate alerts for low stock items based on predefined thresholds.

- Admins shall be able to view reports on slow-moving and fast-moving items.

- The system shall support generating purchase orders for restocking.

### 3.4 Sales Reporting

- The system shall provide detailed sales reports with options to filter by date range, product category, or individual products.

- Reports shall include breakdowns of sales by payment method (cash vs. card).

- The system shall allow exporting of reports in various formats (e.g., CSV, PDF).

### 3.5 Customer Management

- Admins shall be able to view and manage customer information.

- The system shall provide functionality to send promotional emails to customers or specific customer segments.

- Admins shall be able to view individual customer purchase history and behavior.

### 3.6 Order Management

- The system shall display a list of all orders with filtering and sorting options.

- Admins shall be able to view detailed information for each order and update order statuses (e.g., processing, shipped, delivered).

- The system shall support generating shipping labels and packing slips for orders.

### 3.7 User Management

- Admins shall be able to create, edit, and deactivate user accounts for staff members (including cashiers).

- The system shall allow assigning specific roles and permissions to user accounts.

## 4. Database Requirements (using phpMyAdmin)

### 4.1 Data Storage

- The system shall use MySQL database managed through phpMyAdmin for storing all persistent data.

- The database shall have tables for users, products, orders, inventory, and sales transactions.

### 4.2 Data Integrity

- The database shall enforce referential integrity through foreign key relationships.

- The system shall implement transactions to ensure data consistency, especially for critical operations like order processing and inventory updates.

### 4.3 Data Backup and Recovery

- The system shall support regular automated backups of the database.

- Admins shall be able to initiate manual backups through the phpMyAdmin interface.

### 4.4 Performance Optimization

- The database shall be optimized with appropriate indexes for frequently accessed data.

- The system shall implement caching mechanisms to reduce database load for frequently accessed, relatively static data (e.g., product information).

## 5. Integration and API Requirements

### 5.1 Payment Gateway Integration

- The system shall integrate with secure payment gateways to process online transactions.

### 5.2 Email Service Integration

- The system shall integrate with an email service provider to send transactional and promotional emails.

### 5.3 API for Mobile App

- The system shall provide a RESTful API to support the Ionic/Angular-based mobile application, allowing access to product information, user data, and order processing.

## 6. Security Requirements

### 6.1 Authentication and Authorization

- The system shall implement secure user authentication for all interfaces (customer, cashier, and admin).

- User passwords shall be securely hashed and salted before storage.

- The system shall implement role-based access control to restrict access to sensitive functions and data.

### 6.2 Data Protection

- All communication between the client and server shall be encrypted using HTTPS.

- Sensitive data in the database (e.g., payment information) shall be encrypted at rest.

### 6.3 Input Validation

- The system shall implement thorough input validation on both client and server sides to prevent injection attacks and other security vulnerabilities.

## 7. Performance Requirements

### 7.1 Response Time

- The system shall aim for a maximum page load time of 3 seconds under normal conditions.

- API responses for the mobile app shall have a maximum latency of 500ms for non-complex queries.

### 7.2 Scalability

- The system architecture shall support horizontal scaling to handle increased load during peak times (e.g., sales events).

### 7.3 Concurrent Users

- The system shall support at least 1000 concurrent users without significant degradation in performance.

These functional requirements provide a comprehensive overview of the system's capabilities across different interfaces and components, tailored to the use of Ionic, Angular, and phpMyAdmin for database management. They serve as a foundation for development and can be further refined based on specific business needs and technical constraints.