

# Cosmetic Inventory Management System (CIMS)

## Admin Panel

---

### 1. Introduction

The **Cosmetic Inventory Management System (CIMS)** is a web-based application designed to manage cosmetic products, inventory, sales, suppliers, users, and billing operations efficiently. The system consists of a **React (Vite) frontend** and a **PHP-based backend**, ensuring a responsive user interface and robust server-side processing.

---

### 2. Purpose

The purpose of this document is to define the **business and system requirements** for the Cosmetic Inventory Management System. This BRD provides a clear understanding of the system scope, functional modules, validation rules, and overall workflow for stakeholders, developers, and reviewers.

---

### 3. Business Objectives

- Centralize cosmetic inventory and sales management
  - Ensure accurate tracking of stock levels and expiry dates
  - Improve efficiency of POS billing and customer handling
  - Reduce losses caused by expired or low-stock products
  - Provide role-based system access for admins and users
  - Enable data-driven decisions through dashboards and reports
- 

### 4. Technology Stack

- **Frontend:** React.js (Vite)
  - **Backend:** PHP
  - **Database:** MySQL
  - **Validation:** JavaScript Parsley.js
  - **UI Framework:** HTML, CSS, JS, Bootstrap 5.3
- 

## 5. User Roles

Role	Description
Admin	Full access to all system modules
User	Limited access based on assigned permissions

---

## 6. Scope

### 6.1 In-Scope

- Admin and user authentication
  - Cosmetic product and category management
  - Inventory, expiry, and low-stock monitoring
  - POS billing and customer management
  - Supplier and user management
  - Profile management
- 

## 7. System Overview

### 7.1 Authentication Module

- Single **Sign In** form for both admin and users
  - Single **Sign Up** form for both admin and users
  - Fields include name, email, contact number, and password
  - Client-side validation using Parsley.js
  - Credentials stored securely in the database
-

# 8. Functional Requirements

## 8.1 Dashboard Module

- Display inventory summary
  - Show low-stock and expired product alerts
  - Provide quick navigation to modules
- 

## 8.2 Cosmetic Product Module

### 8.2.1 Cosmetic List

- Display product list with image, name, category, brand, price, quantity, and status
- Provide update and delete actions

### 8.2.2 Cosmetic Form

- Bootstrap-based form with Parsley.js validation
  - Fields:
    - Product Name
    - Category Selector
    - Brand Selector (disabled until category selection; filtered by category)
    - Minimum Quantity (no negative values)
    - Maximum Quantity (must be greater than minimum quantity)
    - Expiry Date (Bootstrap Datepicker)
    - Product Description (max 1000 characters)
    - Discount (numeric only, max 2 digits, not exceeding 100, no negative values)
    - Price per Unit (numeric, no negative values)
    - Status (Active / Inactive)
    - Product Image (image only, max 5MB, preview enabled, no PDF/Excel)
    - Gallery Images (max 5 images, each under 5MB, preview enabled)
- 

## 8.3 Category Module

- **Category Form Card:**
  - Category Name
  - Brand Name
  - Status
  - Add button with Parsley.js validation
- **Category List Card:**
  - Display category name, brand name, status

- Update and delete actions
  - Toaster message on successful operations
- 

## 8.4 Expired Management Module

- Display datatable with product name and expiry date
  - Compare product expiry date with current date
  - Automatically list expired products
  - Delete action with toaster notification
- 

## 8.5 Low Stock Module

- Display products where quantity is less than minimum quantity
  - Datatable fields: product name, category, brand, quantity, minimum quantity
  - Delete action with toaster notification
- 

## 8.6 POS Billing Module

- Add Customer form with mandatory fields: name, contact number, email
  - Optional fields: city, country
  - Validation using Parsley.js
  - Category slider for filtering products
  - Product cards with Add to Cart functionality
  - Cart displays product name, price, quantity controls
  - Auto-generate order ID for new customers
  - Walk-in customer support
  - Checkout process:
    - Deduct product quantity from inventory
    - Display success toaster
    - Clear cart functionality
- 

## 8.7 Supplier Module

### Supplier List

- Display supplier name, email, contact, city, country

## Supplier Form

- Fields: name, email, contact, country, city, address, description
  - Contact validation: Indian numbers starting with 6, 7, 8, or 9
  - Email validation: must start with alphabet characters
- 

## 8.8 Users Module

### User List

- Display name, contact, email, role, created date, status
- Update and delete actions

### User Form

- Fields: name, contact, email, role, status
  - Validation using Parsley.js
- 

## 8.9 Profile Module

- Display logged-in user profile details
  - Fields: name, email, contact
  - Option to update profile information
- 

# 9. Non-Functional Requirements

## 9.1 Performance

- Support multiple concurrent users
- Page load time under 3 seconds

## 9.2 Security

- Encrypted password storage
- Role-based access control
- Secure session management

## 9.3 Usability

- Responsive UI
  - Consistent validation messages
  - Easy navigation
- 

## 10. Assumptions

- Users have stable internet access
  - System is used by a single organization
- 

## 11. Constraints

- Web-based application only
  - Dependent on browser compatibility
- 

## 12. Success Criteria

- Accurate cosmetic inventory tracking
  - Reduced expired and low-stock losses
  - Smooth POS billing operations
  - High user satisfaction
- 

# User facing - Frontend

## 13.1 Overview

The User-Facing Frontend of the Cosmetic Inventory Management System (CIMS) provides customers with an interactive, responsive, and visually engaging interface to browse cosmetic products, manage wishlists and carts, place orders, and interact with the brand. The frontend is built using **React.js (Vite)** with **Bootstrap 5.3**, ensuring optimal performance and usability across devices.

---

## 13.2 Global Navigation & Header

### 13.2.1 Top Navigation Bar

The top navigation bar shall be consistently displayed across all frontend pages and includes the following components:

#### 1. Profile Icon (User Account Menu)

- Displays a profile icon on the right side of the navbar
- On click, a dropdown menu appears with:
  - **Settings**
  - **Logout**
- On clicking **Logout**:
  - User session is destroyed
  - User is redirected to the **Sign In** page

#### 2. Wishlist Icon (Heart Icon)

- Displays wishlist items count (if applicable)
- Clicking the icon redirects the user to the **Wishlist Page**
- If the user is not logged in:
  - Redirect to **Sign In** page

#### 3. Cart Icon (Basket Icon)

- Displays the number of products added to cart
  - Clicking the icon opens the **Cart Page**
  - Cart data is maintained per session/user
-

## 13.3 Home Page Module

### 13.3.1 Hero Section

- Dynamic image slider
- Content managed from backend
- Used for promotions, offers, or featured products

### 13.3.2 Top Sale Section

- Displays products currently on sale
- Dynamically fetched from the database
- Includes product image, name, price, and discount

### 13.3.3 Latest Products Section

- Displays **6 latest products**
- Dynamically fetched
- Products sorted by creation date

### 13.3.4 Promotional Banner Section

- **3 static promotional banners**
- Used for marketing or seasonal offers

### 13.3.5 Additional Products Section

- Displays **3 additional highlighted products**

- Static or admin-configurable

### 13.3.6 View All Navigation

- “View All” button redirects users to the **Products Page**
- 

## 13.4 Blog Section

- Displays **3 blog banners**
  - Static content
  - Used for beauty tips, product education, and brand storytelling
- 

## 13.5 Footer Section

The footer shall be displayed across all pages and includes:

- **Brand Logo**
- **Quick Links** (Home, Products, About, Contact)
- **Customer Policy Links**
  - Privacy Policy
  - Terms & Conditions
  - Return Policy
- **Social Media Links**
  - Facebook
  - Instagram

- Twitter
  - LinkedIn
- 

## 13.6 About Page Module

- Static content page
  - Displays company information, mission, vision, and brand story
  - No backend dependency
- 

## 13.7 Products Page Module

### 13.7.1 Category Slider

- Dynamic category cards displayed in a horizontal slider
- Data fetched from backend
- Used for category-based filtering

### 13.7.2 Product Listing Section

- Displays **9 product cards** per view
- Dynamically loaded
- Supports pagination or lazy loading

### 13.7.3 Recommended Products Section

- Displays **3 recommended products**

- Based on category or popularity
- 

## 13.8 Product Card Behavior

Each product card shall display:

- Product image (dynamic)
- Product name
- Discount percentage (auto-calculated)
- Final price after discount
- Wishlist icon (top-left, visible on hover)
- Stock badge (top-right):
  - Displayed when product quantity is **less than 50**
  - Shows remaining quantity

### 13.8.1 Wishlist Interaction

- If user is **not logged in** and clicks wishlist icon:
  - Redirect to **Sign In** page
- If user is logged in:
  - Product is added to wishlist
  - Success toaster message displayed

### 13.8.2 Product Navigation

- Clicking on the product image:

- Redirects to **Product Details Page**
- 

## 13.9 Product Details Page Module

Displays detailed information for the selected product:

- Product image
- Product name
- Price and discount
- Quantity selector:
  - Increase (+) and Decrease (–) buttons
- **Add to Cart** button

### **Add to Cart Behavior:**

- On click:
    - Product quantity is deducted from inventory
    - Product added to cart
    - Success toaster message displayed
- 

## 13.10 Wishlist Page Module

- Displays all wishlisted products
- Users can remove products from wishlist

- Products can be added to cart from wishlist
- 

## 13.11 Cart Page Module

- Displays selected products
  - Quantity controls available
  - Checkout navigation
  - Cart cleared after successful order
- 

## 13.12 Contact Page Module

### Contact Form Fields:

- First Name (Required)
- Last Name (Required)
- Email (Required, valid format)
- Message (Required)

### Validation:

- Client-side validation using **Parsley.js**
- Mandatory field enforcement

### Data Handling:

- Submitted data stored securely in the database

- Success toaster message displayed after submission
- 

### **13.13 Frontend Non-Functional Requirements**

- Fully responsive across desktop, tablet, and mobile
- Page load time under 3 seconds
- Consistent UI/UX across modules
- Clear toaster notifications for all user actions