**Wishlist Page – Documentation**

**Members:**

1. Shravya Javvadi
2. Polimati Srija Sarojini

**Task:** Wishlist Page

**1. Overview**

The Wishlist page is a personalized interface within the SweetSpot Doorstep Delight Web Solution, enabling customers to save and manage their desired products for future reference. This feature enhances user engagement, allowing customers to bookmark products for later purchase, organize and prioritize their preferred items, and receive updates on price changes, promotions, or availability. The Wishlist page is designed with a clear, interactive, and responsive layout, ensuring seamless navigation across devices, and leverages UI components, state management, and backend APIs to fetch and update wishlist data, providing a cohesive and intuitive user experience.

**2. Purpose**

* Allow customers to save and manage their desired products for future reference.
* Display saved products with details (product name, image, price, and availability).
* Enable customers to prioritize and organize their wishlist by adding/removing products, and marking items as favorites.
* Provide price drop notifications and promotions for wishlist items.
* Offer a seamless checkout experience for wishlist items.

1. **Page Structure & Components**

The Wishlist page is structured into functional sections for modularity and maintainability.

**3.1 Header Section**

* Displays the page title ("My Wishlist") and breadcrumb navigation for returning to the Home/Dashboard.
* Maintains consistent styling with other pages using shared Layout and Header components.

**3.2 Wishlist Table/Card**

* Displays a list/table of saved products with details (product name, image, price, and availability).
* Allows customers to prioritize and organize their wishlist by adding/removing products, and marking items as favorites.

**3.3 Product Details Section**

* Displays detailed information about each product, including description, reviews, and ratings.
* Provides a clear call-to-action (CTA) for adding/removing products from the wishlist.

**3.4 Wishlist Management**

* Enables customers to manage their wishlist by adding/removing products, and marking items as favorites.
* Provides options for moving products to cart or deleting them from the wishlist.

**4. Data Handling**

**4.1 Data Sources**

* Wishlist data is retrieved from:
* API endpoint (in production)
* Mock data file/service (for local testing)

Includes:

* Product ID
* Product details
* Customer ID
* Wishlist creation time

**4.2 State Management**

The page uses React state hooks for:

* + Current wishlist data
  + Product additions/removals
  + Updates trigger UI re-render to reflect the latest information.

**5. Implementation Approach**

The implementation is structured to be modular, maintainable, and responsive.

**5.1 Page Initialization**

When the page loads:

* Trigger an API request to get the customer's wishlist.
* Store the response in component state.

**5.2 Rendering the UI**

* Header and footer are imported as reusable components.
* Wishlist table displays products with details.
* Product additions/removals update the wishlist in real-time.

**5.3 Wishlist Updates**

* + The component updates the wishlist:
  + When products are added/removed
  + When product details change

**6. User Flow**

**1.** User logs in and navigates to Wishlist page.

**2.** Page loads wishlist data via API/mock service.

**3.** User can view and manage their wishlist.

**4.** User navigates away via action buttons or site navigation.

**7. Key Features Summary**

* + Personalized wishlist management
  + Real-time product updates
  + Responsive design for mobile and desktop
  + Error and empty state handling

**8. Dependencies**

* + **React:** Core UI library for building components.
  + **React Router:** For route-based access to the wishlist page.
  + **CSS / TailwindCSS:** For styling and responsiveness.
  + **API services module:** For making HTTP requests**.**