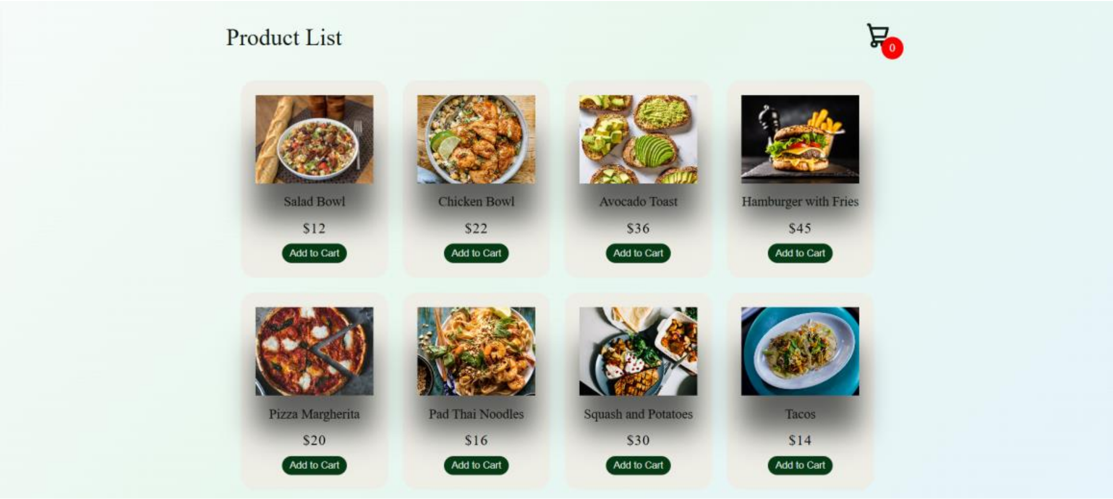
Documentation: Product List Application

1. Overview of Application Screens

The application consists of three primary user interfaces implemented using custom HTML, CSS, and JavaScript.

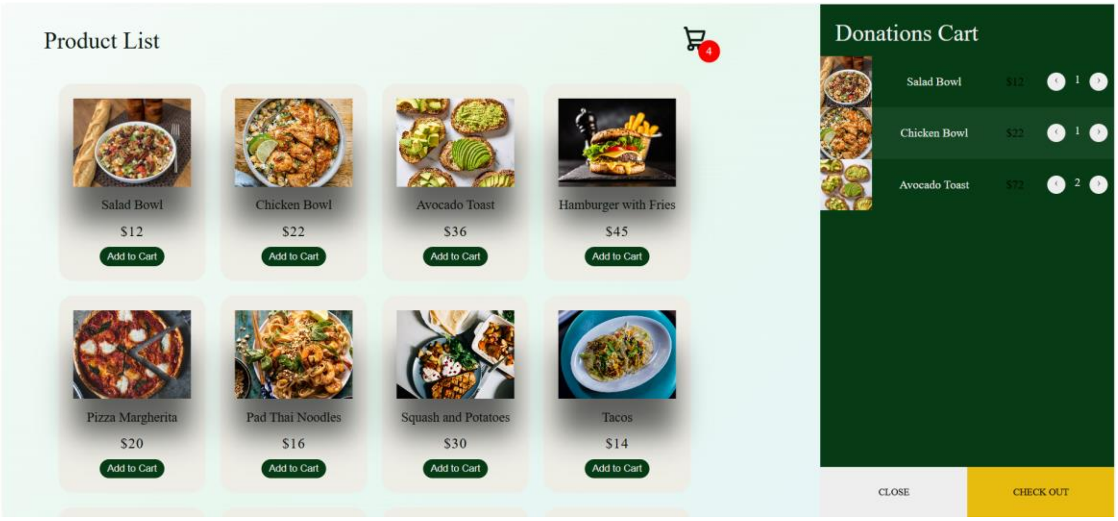
1.1 Product List View (Initial State)

The initial view displays a list of products loaded dynamically from a local JSON file (products.json). Each product includes an image, name, price, and an “Add to Cart” button. The layout is responsive and adapts to different screen sizes using CSS Grid.



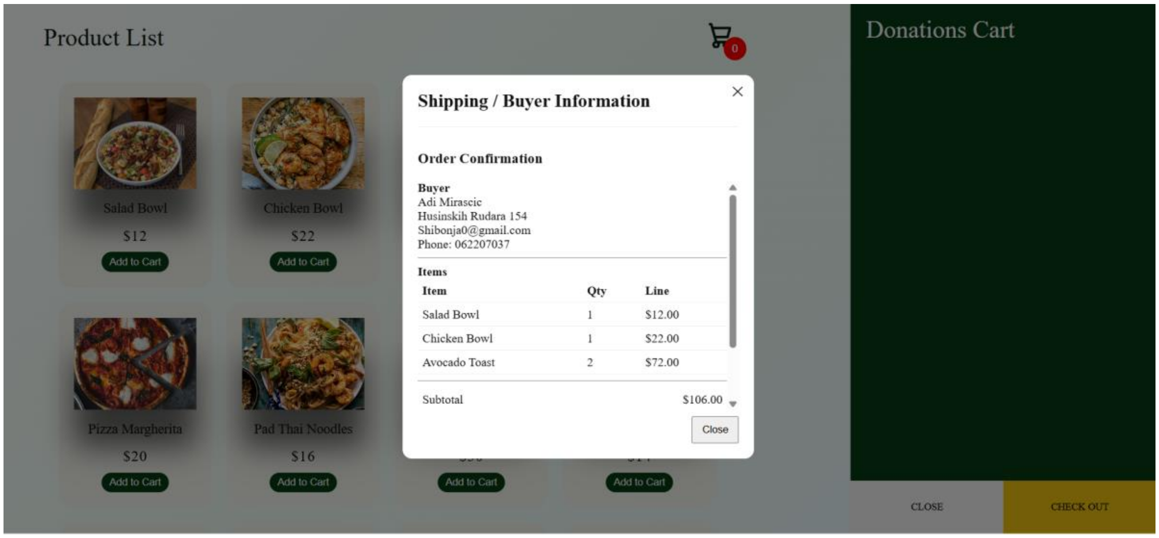
1.2 Shopping Cart Panel

The shopping cart is implemented as a custom slide-in panel that appears from the right side of the screen. It displays selected products with quantity controls and dynamically updates the total item count. Cart data persists using the browser’s localStorage.



1.3 Checkout and Order Confirmation

The checkout process is handled through a custom modal overlay. Users enter shipping and contact information, after which an order confirmation summary is displayed.



Shipping / Buyer Information

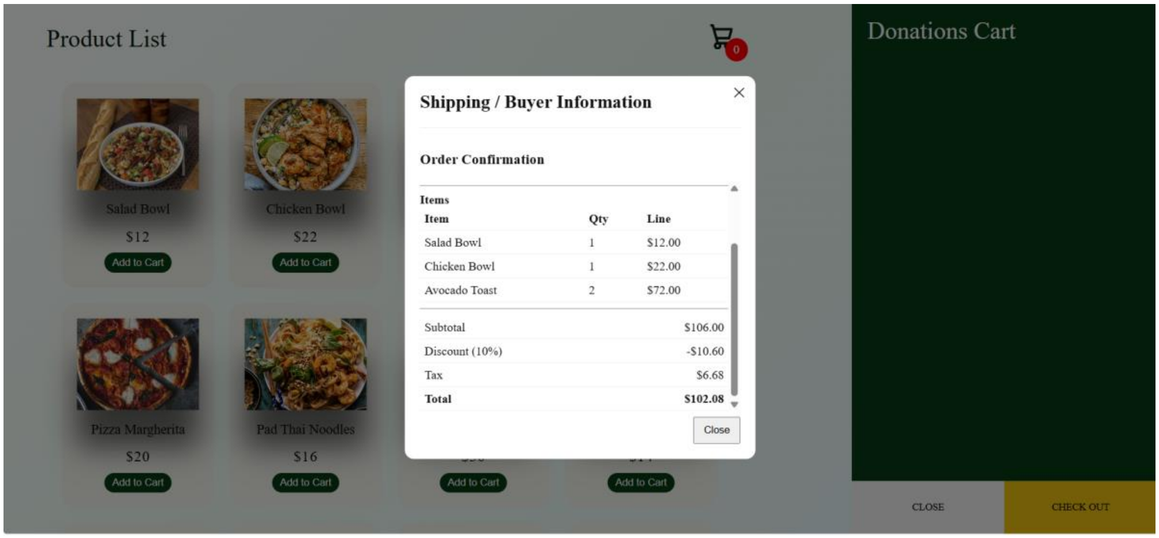
Order Confirmation

Buyer
Adi Mirascic
Husinskih Rudara 154
Shibonja0@gmail.com
Phone: 062207037

Items

Item	Qty	Line
Salad Bowl	1	$12.00
Chicken Bowl	1	$22.00
Avocado Toast	2	$72.00

Subtotal	$106.00



Shipping / Buyer Information

Order Confirmation

Items

Item	Qty	Line
Salad Bowl	1	$12.00
Chicken Bowl	1	$22.00
Avocado Toast	2	$72.00

Subtotal	$106.00
Discount (10%)	-$10.60
Tax	$6.68
Total	**$102.08**

2. Front-End Structure and Styling

This application does not use external UI frameworks such as Bootstrap. All layout, styling, and UI behavior are implemented using custom CSS.

The product list is displayed using CSS Grid with four columns on large screens, three columns on tablets, and two columns on mobile devices. Media queries are used to ensure responsive behavior.

Each product card is a custom-styled container that includes an image, product name, price, and an “Add to Cart” button. Product cards are generated dynamically via JavaScript.

The shopping cart panel is implemented as a fixed-position sidebar. Its visibility is controlled by toggling a CSS class on the body element, enabling a smooth slide-in animation.

The checkout interface is implemented as a custom modal overlay. Visibility is controlled using a reusable hidden CSS class.

3. JavaScript Functionality

Product data is fetched asynchronously from products.json and rendered dynamically on page load.

Cart items are stored as objects containing a product ID and quantity. When a product is added multiple times, its quantity is increased. Quantity controls allow users to increment or decrement item counts, and items are removed when the quantity reaches zero. Cart data is stored in localStorage to persist across page reloads.

Checkout is blocked if the cart is empty. Client-side validation ensures correct formatting for name, email, phone number, address, and ZIP code. Validation errors are displayed inline.

4. Price Calculation Logic

The subtotal is calculated as the sum of each product’s price multiplied by its quantity.

A 10 percent discount is applied only when the total number of items in the cart is three or more.

A 7 percent tax rate is applied to the discounted subtotal.

The final total is calculated as the subtotal minus any discount plus tax.

5. Order Confirmation

The order confirmation screen displays buyer information, an itemized list of purchased products, and a pricing breakdown including subtotal, discount, tax, and total. After confirmation, the shopping cart is cleared and stored cart data is reset.

6. Technical Summary

The Product List Application is a fully client-side web application built without external frameworks. It features dynamic content loading from JSON, persistent cart storage using localStorage, responsive design using CSS Grid and media queries, and custom JavaS3cript for cart management and checkout functionality.

Component	**Purpose in Application**	**Inferred Bootstrap Classes**
Grid System	Used to arrange the product items in a responsive 4-column layout.	container, row, col-lg-3, col-md-4, col-sm-6
Cards	Each product item is contained within a card component, which neatly groups the image, title, price, and action button.	card, card-img-top, card-body
Buttons	Used for the 'Add to Cart' action and the 'Check Out' action. The 'Add to Cart' button appears to be a custom-styled primary button.	btn, btn-success (custom color), btn-lg
Offcanvas / Sidebar	The 'Donations Cart' appears to slide in from the right, which is a perfect use case for Bootstrap's Offcanvas component.	offcanvas, offcanvas-end
Badges	The red circle on the shopping cart icon, indicating the number of items in the cart, is a classic use of the Badge component.	badge, bg-danger, rounded-pill
Modal	The 'Shipping / Buyer Information' and 'Order Confirmation' pop-up is implemented using the Modal component.	modal, modal-dialog, modal-content
Tables	Used within the modal to clearly present the itemized list and the final calculation breakdown.	table
Calculation Step	**Description**	**Inferred JavaScript Logic**
Subtotal	Sum of (Price * Quantity) for all items in the cart.	subtotal = cartItems.reduce((acc, item) => acc + (item.price * item.qty), 0)
Discount (10%)	A fixed 10% discount applied to the Subtotal.	discount = subtotal * 0.10
Tax	A tax rate (inferred to be 7.0% based on the screenshot) applied to the discounted subtotal.	tax = (subtotal - discount) * 0.07
Total	The final amount due.	total = subtotal - discount + tax