# **Single Product Page: WordPress Integration Guide**

This document provides a comprehensive guide for converting the static single-product.html page into a dynamic, manageable WordPress template. It covers the project's design system, structure, assets, and specific recommendations for WordPress implementation.

## **1. Project Overview & Dependencies**

The page is a modern, responsive product/donation page built with Tailwind CSS and enhanced with jQuery for interactivity.

**Key Dependencies:**

* **Tailwind CSS:** The primary framework for styling. The final compiled CSS is in output.css.
* **Google Fonts:** The "Inter" font family is used. It should be enqueued from Google Fonts.
* **jQuery:** Used for interactive elements like tabs, sliders, and accordions. WordPress includes jQuery by default, so we just need to ensure it's loaded as a dependency for our custom script.
* **Custom CSS:** A stylesheet css/single-product.css is included for styles that are difficult or verbose to handle with Tailwind utility classes alone.
* **Custom JavaScript:** A file js/single-product.js contains the jQuery code for the page's dynamic features.
* **Image Assets:** All images are located in an /images directory. These paths will need to be made dynamic in WordPress.

## **2. Design System (from tailwind.config.js)**

The tailwind.config.js file defines the project's core design tokens. When building out the WordPress theme, these values should be referenced to maintain design consistency.

### **Custom Font**

* **Primary Font:** Inter
  + **Tailwind class:** font-sans

### **Custom Color Palette**

The theme extends Tailwind's default colors with a specific brand palette.

| **Color Name** | **Hex Code** | **Tailwind Class** | **Usage Notes** |
| --- | --- | --- | --- |
| brand-teal | #14B8A6 | bg-brand-teal etc. | Primary call-to-action, accents, highlights. |
| brand-teal-dark | #0D9488 | bg-brand-teal-dark | Hover state for primary CTAs. |
| brand-dark | #1F2937 | text-brand-dark etc. | Main headings, dark backgrounds, some text. |
| brand-text-medium | #4B5563 | text-brand-text-medium | Body copy, paragraphs, descriptive text. |
| brand-text-light | #6B7281 | text-brand-text-light | Secondary text, meta information (e.g., location). |
| brand-bg-light | #F9FAFB | bg-brand-bg-light | Main page background color. |
| brand-bg-white | #FFFFFF | bg-brand-bg-white | Card backgrounds, content areas. |
| brand-border | #D1D5DB | border-brand-border | Default border color for inputs and elements. |

## **3. Page Structure & Component Breakdown**

The single-product.html page is composed of several distinct, reusable sections. For WordPress development, each of these sections should be converted into a separate template part file (e.g., in a /template-parts/ directory) and called into the main page template.

### **Section 1: Donation Form (#donation-form-section)**

* **Description:** The main hero section. It features a donation form on the left and a static image on the right (hidden on smaller screens).
* **WordPress Implementation:**
  + **Content:** The heading, paragraph text, and dropdown options should be editable, possibly via the page editor or a custom fields plugin like ACF (Advanced Custom Fields).
  + **Interactivity (single-product.js):**
    1. **Single/Monthly Toggle:** The buttons (#singleBtn, #monthlyBtn) switch an active state. The JS logic for this needs to be included.
    2. **Amount Buttons:** Buttons with data-amount attributes update the value of the custom amount input (#customAmountInput).
    3. **Form Submission:** The "DONATE NOW" button will need to link to the payment processing logic (e.g., WooCommerce checkout, Stripe/PayPal handler).

### **Section 2: Image Gallery (#image-gallery-section)**

* **Description:** Displays a large featured image with a grid of thumbnails below it.
* **WordPress Implementation:**
  + **Content:** This is a perfect candidate for an ACF Gallery field. This would allow the client to easily upload, remove, and reorder images from the WordPress admin panel.
  + **Interactivity (single-product.js):**
    - Clicking a thumbnail (.image-gallery-thumbnail-img) updates the src of the main featured image (#featuredImage) and handles the active state (border color) on the thumbnail.

### **Section 3: Testimonial Carousel (#testimonial-carousel-section)**

* **Description:** A slider that displays user testimonials. It shows 3 cards on desktop and likely 1 on mobile.
* **WordPress Implementation:**
  + **Content:** Create a "Testimonials" Custom Post Type (CPT). Each testimonial would be a post with fields for the quote, author name, and location. The page template would then query these posts and loop through them to generate the carousel items.
  + **Interactivity (single-product.js):**
    - The script handles the sliding logic using CSS transforms on the #testimonial-track.
    - It controls the previous/next buttons (#testimonial-prev, #testimonial-next) and the navigation dots (#testimonial-dots).

### **Section 4: Select Your Blessing (Product Tabs) (#select-blessing-tab-section)**

* **Description:** A tabbed section displaying different categories of products or donation packages.
* **WordPress Implementation:**
  + **Content:** This could be implemented in a few ways:
    1. **WooCommerce:** If this is an e-commerce site, these could be WooCommerce products categorized under "Orphan Daily Meals," "Orphan Baby Milk," etc. You would query products by category for each tab.
    2. **Custom Post Type:** Create a "Blessings" or "Products" CPT with a custom taxonomy for the categories.
    3. **ACF Repeater:** For simpler use cases, use an ACF Repeater field to allow the user to create multiple tabs, and a nested repeater for the cards within each tab.
  + **Interactivity (single-product.js):**
    - The tab buttons (.tab-btn) control the visibility of the corresponding content panes (.tab-pane).

### **Section 5: Why Trust Us (#why-trust-section)**

* **Description:** A simple 4-column feature grid.
* **WordPress Implementation:**
  + **Content:** This is ideal for an ACF Repeater field or a custom Gutenberg block, allowing the client to edit the icon, title, and description for each feature.

### **Section 6: Qurbani Countdown (#qurbani-countdown-section)**

* **Description:** A promotional banner with a countdown timer and a progress bar.
* **WordPress Implementation:**
  + **Content:** The title and text should be editable. The target date for the countdown and the progress bar percentage should be stored in custom fields.
  + **Interactivity (single-product.js):**
    - The script calculates the time remaining until a target date and updates the day/hour/minute/second elements.
    - It also dynamically sets the width of the progress bar fill (#qurban-progress-fill).

### **Section 7: How It Works (#how-it-works-section)**

* **Description:** A 3-step process explanation.
* **WordPress Implementation:**
  + **Content:** Similar to the "Why Trust Us" section, this can be managed with an ACF Repeater or a custom Gutenberg block.

### **Section 8: FAQ (#faq-section)**

* **Description:** An accordion-style FAQ section.
* **WordPress Implementation:**
  + **Content:** Create an "FAQ" Custom Post Type or use an ACF Repeater field on the page. Each item would have a question and an answer.
  + **Interactivity (single-product.js):**
    - The script toggles a class (active) and uses slideToggle() on the answer content when a question is clicked.

### **Section 9: Other Campaigns (#other-campaigns-section)**

* **Description:** A 4-column grid of cards linking to other campaign pages.
* **WordPress Implementation:**
  + **Content:** These cards could be populated by querying other Pages or Posts (perhaps from a "Campaigns" category) and pulling their featured image, title, excerpt, and permalink.

## **4. WordPress Development Checklist**

1. **Theme Setup:**
   * [ ] Set up a new WordPress theme or child theme.
   * [ ] Create a functions.php file.
2. **Enqueue Scripts & Styles:**
   * [ ] In functions.php, create a function to enqueue styles.
     + Enqueue Google Fonts (https://fonts.googleapis.com/css2?family=Inter...).
     + Enqueue the compiled Tailwind CSS (/output.css).
     + Enqueue the custom CSS (/css/single-product.css).
   * [ ] In functions.php, create a function to enqueue scripts.
     + Enqueue the custom JavaScript file (/js/single-product.js). Make sure to list 'jquery' as a dependency.
3. **Create Page Template:**
   * [ ] Create a custom page template file (e.g., template-donation-page.php).
   * [ ] Copy the <body> content from single-product.html into this file.
   * [ ] Assign this template to a new page in the WordPress admin.
4. **Template Partials:**
   * [ ] Break down the page template into smaller, manageable template parts for each section (e.g., template-parts/section-faq.php).
   * [ ] Use get\_template\_part() to call them into the main page template.
5. **Dynamic Content:**
   * [ ] Choose a strategy for managing content (ACF, Custom Post Types, Gutenberg).
   * [ ] Implement the chosen strategy for each section as detailed above.
   * [ ] Replace all static text and images with the appropriate WordPress functions (the\_title(), the\_content(), get\_field(), the\_post\_thumbnail(), etc.).
   * [ ] Update all href="#" and src="images/..." paths with dynamic WordPress links using get\_permalink() and get\_template\_directory\_uri().
6. **Final Testing:**
   * [ ] Verify that all Tailwind styles are applying correctly.
   * [ ] Test all jQuery-powered interactive elements (carousel, tabs, accordion) to ensure they work with the dynamically loaded content.
   * [ ] Test the page's responsiveness on mobile, tablet, and desktop.