

EPROJECT: PASCAL IMITATION JEWELLERY

DOCUMENTATION [DEVELOPER GUIDE]

Detail	Value
Project Title	Pascal Imitation Jewellery - E-commerce Site
Document Type	Developer Guide & Technical Specifications
Technology	HTML5, CSS3, Vanilla JavaScript
Target Audience	Project Developers / Maintainers
Version	1.0 (Initial Release)

SECTION 1: SYNOPSIS AND ARCHITECTURE

The Pascal Imitation Jewellery project is a **Static Multi-Page Application (MPA)** developed using foundational web technologies (HTML, CSS, JavaScript). Its primary goal is to provide a royal and luxurious e-commerce experience centered around imitation jewelry.

1.1 Project Objectives

- 1. Immersive UI:** Implement a dark-mode, luxury theme using a gold color palette for visual appeal.
- 2. Dynamic Product Display:** Centralize product data in JavaScript for easy filtering and rendering on category pages.
- 3. Core Features:** Provide essential e-commerce tools like Product Filtering, Comparison, and Wishlist functionality.
- 4. UX Enhancement:** Utilize a custom pre-loader for an engaging initial loading experience on the homepage.

1.2 File Structure Overview

The project is organized with clear separation between structure, styling, and logic:

Component	Files	Description
Structure	* .html (12 files)	Defines all page layouts (Home, Products, Rings, Compare, etc.).
Styling	CSS/style.css	Global styling, variables, animations, and responsive design.

Component	Files	Description
Core Logic	JAVASCRIPT/script.js	Contains all product data, filtering logic, and feature implementation.
Utility Logic	JAVASCRIPT/loader.js	Dedicated script for the custom animated loading screen.

SECTION 2: TECHNICAL SPECIFICATIONS

2.1 Styling Standardization (`CSS/style.css`)

The design relies heavily on centralized CSS variables defined in the `:root` selector for maintainability.

CSS Variable	Default Value	Usage
--gold	#D4AF37	Primary accent for buttons, headings, and interactive elements.
--dark	#0a0a0a	Background color for the dark mode theme.
--text-primary	#ffffff	Standard text color.
--glass	rgba(255, 255, 255, 0.05)	Used for subtle, translucent backgrounds (e.g., card backgrounds).

SECTION 3: CORE FUNCTIONALITY (JAVASCRIPT)

3.1 Custom Pre-loader (*loader.js*)

The `loader.js` script provides a simulated, multi-step loading experience on `home.html`.

Functionality	Implementation Details
Progress Simulation	Uses <code>setInterval</code> to increment the <code>progress</code> variable by a random amount (2-6%) every 200ms.
Step Tracking	Updates three status steps (<code>step1</code> , <code>step2</code> , <code>step3</code>) based on progress milestones (30% for 'Processing', 70% for 'Ready').
Completion	Once <code>progress >= 100</code> , the <code>completeLoading()</code> function executes, which hides the loader (<code>#loader</code>) with a fade-out effect.

3.2 Product Features (*script.js*)

Feature Name	Primary Function(s)	Description
Category Display	displayRings (collection)	Filters the products array by category and injects the corresponding HTML cards into the product listing container.
Filtering & Sorting	setupRingsFilterListeners ()	Listens for changes in the Price Range, Rating, and Brand filters, and re-renders the filtered product set.
Product Comparison	initComparePage (), addToComparison ()	Manages product IDs stored in local storage and

Feature Name	Primary Function(s)	Description
		populates the comparison table on compare.html using the product's features array.
Wishlist & Notifications	addWishlistListener(), showBraceletsNotification()	Adds/removes items from the wishlist (toggling the heart icon) and displays a small notification banner for user feedback.

SECTION 4: TASK SHEET / MAINTENANCE LOG

Tas k Ref.	Task Description	Status	Note for Next Developer
1.01	Standardize Footer/Header	Pending	The Footer HTML is repeated across all 12 HTML files (aboutus.html, home.html, etc.). Consider using a template or server-side include for easy updates.
1.02	Contact Info Update	Complete	Contact details (+92 333 0101010, info@pascaljewellery.com) are present in all page footers.
1.03	products.html Logic	Incomplete	The products.html file seems to be a gateway page. Logic needs to be added to dynamically show all products, not just categories.
1.04	Image Placeholder Review	Complete	Placeholder images (e.g., https://picsum.photos/) have been replaced with relative paths (images/EPROJECTS

Tas k Ref.	Task Description	Status	Note for Next Developer
			IMAGES/ . . .) in the products array.
1.05	Legal Pages	Complete	Dedicated pages for policy.html and terms.html are available and linked in the footer.