

NOVA Fashion Store

Frontend Documentation

1. Project Overview

NOVA is a fully client-side fashion e-commerce store built in vanilla HTML, CSS, and JavaScript. It supports bilingual content (English + Arabic with RTL), dark/light mode, a full cart and wishlist system, product modals with image zoom, checkout via EmailJS, and analytics tracking.

There is no build step, no framework, and no backend dependency. Every file is static and deployable directly to Vercel or any static host.

File Structure

File	Role	Can Edit Safely?
index.html	Main HTML shell, all DOM structure, inline init script	Yes — layout & config
styles.css	Core design system: variables, layout, components	Yes — styling only
extras.css	Additional feature styles (engagement, modal extras)	Yes — styling only
products.js	Product catalogue — all items, prices, images, stock	YES — this is your main data file
i18n.js	All UI text in English + Arabic	Yes — text changes
lang.js	Language switching logic	No — logic
cart.js	Cart + wishlist state and UI	No — logic
shop.js	Product grid rendering, filter, search, sort	No — logic
modal.js	Product detail modal, zoom, size guide, related items	No — logic
checkout.js	Checkout form + EmailJS order submission	Yes — EmailJS IDs
darktheme.js	Dark/light mode toggle and persistence	No — logic
trustbar.js	Trust badges strip rendering	No — logic
reviews.js	Customer reviews data + rendering	Yes — add/edit reviews
infosections.js	Delivery info & return policy rendering	No — logic

engagement.js	Promo banner, newsletter popup, recently viewed, countdown	YES — config at top
analytics.js	GA4 + Plausible event tracking	YES — config at top

2. Configuration Files — What to Edit

products.js — Your Product Catalogue

This is the most important file to edit. Each product follows this structure:

```
{
  id: 'shoes-001',                // unique ID, used in cart & URLs
  name: { en: 'White Sneakers', ar: 'أبيض حذاء' },
  category: 'Shoes',              // must match a category in il8n.js
  price: 89.99,
  oldPrice: 119.99,              // null if no discount
  desc: { en: '...', ar: '...' },
  colors: [
    { name: { en: 'White', ar: 'أبيض' }, hex: '#F5F5F5',
      images: ['url1', 'url2', 'url3'] }
  ],
  sizes: ['36', '37', '38'],
  stock: 18,
  sizeStock: { '36': 2, '37': 0, '38': 4 } // optional per-size stock
}
```

Key rules for products:

- stock: 0 marks the item as "Sold Out" across cards and modal
- sizeStock lets you mark individual sizes as out of stock (value 0)
- oldPrice creates the discount badge and % calculation automatically
- First image in each color is the card image; second is the hover image
- Images can be Unsplash URLs or your own hosted URLs (Cloudinary recommended)

engagement.js — Promo Banner, Countdown, Newsletter

All configuration is at the very top of the file in three objects:

```
PROMO_CONFIG  - banner text (EN+AR), CTA label, active: true/false
NEWSLETTER_CONFIG - popup delay (ms), heading, body text, active:
true/false
SALE_CONFIG   - endDate (ISO string), label text, active: true/false
```

analytics.js — Tracking Setup

Edit ANALYTICS_CONFIG at the top:

```
ga4: {
  enabled: true,
  measurementId: 'G-XXXXXXXXXX' // your GA4 ID
},
plausible: {
  enabled: false,
  domain: 'yourdomain.com'
},
debug: false // set true during testing
```

checkout.js — EmailJS Order Submission

At the top of index.html, find and replace these three constants:

```
const EMAILJS_PUBLIC_KEY = 'your_public_key';
const EMAILJS_SERVICE_ID = 'your_service_id';
const EMAILJS_TEMPLATE_ID = 'your_template_id';
```

Your EmailJS template must use these exact variable names: {{order_id}}, {{customer_name}}, {{customer_address}}, {{customer_phone}}, {{order_details}}, {{order_total}}, {{order_date}}

i18n.js — UI Text

All displayed text is defined here in both languages. To add a new translation key:

1. Add the key to both the en: {} and ar: {} objects
2. Use it in HTML with data-i18n="your.key"
3. Or in JS with t('your.key')

reviews.js — Customer Reviews

REVIEWS_DATA is an array at the top of the file. Each review object has: product (en+ar name), stars (1-5), text (en+ar), name, avatar (pravatar URL), date (en+ar).

3. Feature Index — What Is Built

#	Feature	File	Status
1	Second image on card hover	shop.js	✓ Done
2	Quick-add size selector on card hover	shop.js	✓ Done
3	Stock urgency badge on card	shop.js	✓ Done
4	Image zoom on hover + lightbox click	modal.js	✓ Done
5	Size guide popup with chart	modal.js	✓ Done
6	You Might Also Like (related products)	modal.js	✓ Done
7	Bottom nav bar (mobile only)	engagement.js	✓ Done
8	Swipeable category pills on mobile	extras.css	✓ Done
9	Swipe gestures on modal gallery	index.html	✓ Done
10	Dismissible promo banner	engagement.js	✓ Done
11	Newsletter popup (30s delay)	engagement.js	✓ Done
12	Recently viewed products section	engagement.js	✓ Done
13	Sale countdown timer	engagement.js	✓ Done
14	Scroll progress bar	index.html	✓ Done
15	Category breadcrumb	index.html	✓ Done
16	Blur-up image placeholder	index.html	✓ Done
17	Back to top button	index.html	✓ Done
18	Shareable URLs (category + product)	index.html	✓ Done
19	Share product button (native + copy)	modal.js	✓ Done
20	GA4 + Plausible analytics	analytics.js	✓ Done

4. Analytics Events Reference

Event Name	When It Fires	Key Parameters
page_view	Page loads	auto (GA4 built-in)
view_item	Product modal opens	item_id, item_name, category, price

add_to_cart	Modal add OR quick-add on card	item_id, size, price, method
remove_from_cart	Remove clicked in cart panel	item_id, qty, size
begin_checkout	Proceed to Checkout clicked	value, items (JSON)
purchase	Order placed successfully	transaction_id, value, items
wishlist_add	Heart icon — not yet wishlisted	item_id, category, price
wishlist_remove	Heart icon — already wishlisted	item_id
search	800ms after user stops typing	search_term
filter_apply	Category pill or nav tab clicked	category

5. Adding a New Product (Step by Step)

4. Open products.js
5. Copy an existing product object and paste it at the end of the PRODUCTS array (before the closing `])`)
6. Give it a unique id (e.g. "shoes-003")
7. Fill in name, desc in both languages
8. Set category to one of: Accessories / Shoes / Leggings / T-Shirts / Arm Sleeves
9. Set price and oldPrice (or null for no discount)
10. Replace the image URLs with your product photos
11. Set stock to your actual quantity
12. Save — the product appears immediately on the site

6. Adding a New Category

13. Add entries to i18n.js in both en and ar sections: cat.NewCat and nav.NewCat
14. Add a `<button class="cat-pill" data-cat="NewCat">` in index.html (category pills section)
15. Add a `<button class="cat-nav" data-cat="NewCat">` in the header nav
16. If you want a size guide for it, add an entry to SIZE_CHARTS in modal.js

7. Script Load Order (index.html)

The scripts must be loaded in this exact order or the store will break:

```
1. products.js    — data must exist before any logic
```

2. `i18n.js` — translations must exist before rendering
3. [all other `.js` files]
4. `engagement.js` — must be near last (patches `openProduct`)
5. `analytics.js` — MUST be last (patches all other functions)