Dashlane's Trustworthy Web Pages Checklist

Don't Look Phishy: 5 Quick Fixes

Phishing kits often copy the surface of a site better than its real functionality

1 Non-Functional Page Components

Issue	Examples
Menu items that go nowhere	Home, Products, Pricing, Support, Track order
Footer links that are only decorative	Privacy Policy, Terms of Service, Cookie Settings, Help Center, Contact Us
Search bars that are not real	Fake search box or icon /"search products" field
Category buttons that all behave the same	Electronics, Fashion, Orders, Account, Gift cards
Account or **language** selectors that are fake	Eng / Fr / De, Personal / Business, Sign in with another account
"**Remember me**" or **checkbox** options that do not matter	Remember this device, Keep me signed in, Save card for future use
Buttons with generic labels that all trigger the same action	Continue, Verify, Confirm, Next, Resolve issue
Pagination, **tabs**, or **step indicators** that are fake/not functional	Step 1 of 3, Shipping / Payment / Review, Overview / Billing / Security
Product cards or **order summaries** that are just scenery/non-functional	Non-functional order summaries/invoice previews/customer account panels
Social media or **help icons** that are dead	Facebook/X/Instagram/LinkedIn icons, Live chat bubble, Support/help widget

2 Hostname Setup

- ✓ Stable, readable, brand-consistent, logically structured
- ✗ Random, overly long, stuffed with extra words, numbers, or subdomains, and only loosely related to the brand it claims to be
- ⚠ A random-looking hostname is **NOT** the same as a URL with random-looking parameters

Phishing often starts with the address itself: attackers rely on throwaway domains and brand imitation at scale

3 Meaningful Content

- ☐ Navigation labels
- ☐ Help text
- ☐ Policy links
- ☐ Legal information
- ☐ Account explanations
- ☐ Contact information
- ☐ Product descriptions
- ☐ Accessible labels and Instructions

A page can look convincing with very little real content

Trustworthy pages usually follow one visual language; phishy pages often feel cobbled together

4 Styling Consistency

- ✓ One **consistent styling system** across buttons, forms, navigation, and layout
- ✓ Styling mostly **integrated into the product**, not patched in case by case
- ✓ **Inline styling limited** to small, justified cases
- ✓ **Third-party CSS limited** to clear **vendor features** such as CAPTCHA, payment, or maps
- ✓ **External widgets visually match the rest of the page –** typography, spacing, and button patterns
- ✓ **Reused the same rules** for colors, fonts, spacing, fields, and states
- ✓ Feels like part of the same website, **not an isolated screen**

- ✗ **Looks surprisingly isolated** from the rest of the site with **almost no CSS**
- ✗ **Relies heavily on inline styling** instead of reusable styling patterns
- ✗ Different **sections** look as if they were **built with different styling systems**
- ✗ A **simple** login or payment **page** pulls **CSS from multiple unrelated sources**, suggesting copied assembly
- ✗ Buttons, fields, icons, or checkboxes **do not follow the same visual language**
- ✗ Looks **cobbled together from mismatched pieces**
- ✗ Visual polish relies on **screenshots or image-based UI** instead of real interface components
- ✗ **Styling shortcuts** make the page **feel like a template** with a brand pasted on top

5 Frameworks That Support Trustworthy Web Development

- **Choose modern frameworks:** React / Next.js, Vue, Angular, Svelte, Lit, Bootstrap
  Avoid jQuery-heavy pages, disconnected plugins, and manual DOM patchwork
- **Respect HTML semantics:** use real buttons, links, forms, and labels
  Avoid clickable s and fake form elements
- **Use standard routing:** rely on built-in routers for internal navigation
  Avoid manual redirects, hash hacks, and hidden-frame navigation
- **Follow security conventions:** escape untrusted content and prevent code injection
  Avoid raw HTML insertion and unsafe script loading
- **Keep CSS reusable:** use one consistent styling system across the app
  Avoid fragmented styles, excessive inline CSS, and mismatched UI blocks
- **Align with MDN:** follow Mozilla Developer Network guidance and modern web standards

The easier a fake page is to assemble from reusable parts, the easier it is to scale.