

Plan: QuickerSite Code Audit Findings + 2026 Improvement Backlog

This is a read-only audit. Below are confirmed findings (bugs, dead code, one security note) and a prioritized "low-hanging fruit" list. ****No files were changed.**** I can turn any subset into implementation tasks on your approval.

A. Confirmed bugs

| # | Location | Issue | Severity |

|---|-----|-----|-----|

| 1 | `includes/FormField.asp:377` | `public property get sFileURL(mode)` has an ****empty body**** — always returns "". It's defined but consumed nowhere, so it's both a bug and dead. | Low (dead) |

| 2 | `includes/functions.asp` `GetEmailValidator()` | Body is ****duplicated**** verbatim (`Set...New RegExp` + `.Pattern` + `.IgnoreCase` block appears twice). Harmless but the first allocation is dead. | Low |

| 3 | `includes/submission.asp` | `On Error Resume Next` is set (x2) and ****never**** reset with `On Error Goto 0` — errors swallowed for the rest of the page. | Med |

| 4 | `customer.asp` (delta 8), `poll.asp`, `catalogField.asp`, `FormField.asp` | `On Error Resume Next` left active across method boundaries (more `resume next` than `goto 0`). Hides real failures; makes debugging hard. | Med |

I did ****not**** find another QS_iFrameYT-style "assigns to wrong variable" bug. The 200+ "never assigns return value" hits from my first scan are almost all `Pick/Save/remove/getRequestValues` used as subs — legitimate.

B. Dead code / orphan files

| Item | Evidence | Confidence |

|-----|-----|-----|

| `includes/freeASPUpload - Copy.asp` | 0 inbound references anywhere — a leftover backup of `freeASPUpload.asp`. | High |

| Legacy editor configs `C_FCKEDITOR`, `C_CUTEEDITOR`, `C_INNOVAEDITOR` (`begin.asp`) | Only 1 ref each = the `const` declaration itself. Folders absent on disk (only `ckeditor445` exists); `QS_EDITOR=4` hardwires CKEditor. The `createFCKInstance` path always uses CKEditor. | High |

| `sFileURL` (above) | Defined, never called. | High |

| `GetEmailValidator` duplicate block | Second assignment shadows the first. | High |

Worth a deeper sweep before deletion: a full orphan-file pass across all `bs_*/`/`ad_*/` pages (my spot checks found `pb.asp` IS referenced once; `ad_upgrade2/3/4` need confirming). I'd run this as part of implementation, not guess now.

C. Security (critical-only)

- **No critical vulnerabilities confirmed.** The two surfaces that looked critical are mitigated:

- `newsletterCategory.asp:144/170/173` concatenates `request.querystring("sEmail")` into SQL **without** `cleanUp()`. But it's gated by `checkEmailSyntax()` (anchored regex `^...\$` that rejects quotes) before the query runs, so injection isn't reachable. **Recommendation** (defense-in-depth, not critical): **still wrap with `cleanUp()`** — relying on a validator as your only SQL defense is fragile.

- `bs_editcode.asp` writes arbitrary file content from `request("path")`, but is behind backsite auth, blocks second-admins, enforces a non-executable extension allow-list (`js/html/htm/txt/css`), and requires the path to be under the userfiles vdir. Acceptable.

- One hardening note (not critical): the `bs_editcode.asp` path check is a raw string-prefix compare, not canonicalized — fine today because of the extension allow-list, but worth a `..` guard if that list ever widens.

D. Low-hanging fruit to make QuickerSite more versatile in 2026

Tier 1 — small, high impact

1. **Modern editor config cleanup** — drop the 3 dead editor consts; document CKEditor as the only editor.
2. **`On Error` hygiene pass** — add `On Error Goto 0` where missing (start with `submission.asp`); surfaces real bugs cheaply.
3. **`cleanUp()` on the newsletter `sEmail` queries** — defense-in-depth.
4. **Remove orphan files** (`freeASPUUpload - Copy.asp`, dead stubs) — reduces confusion/attack surface.
5. **Security headers via `web.config`** — add `X-Content-Type-Options`, `Referrer-Policy`, `X-Frame-Options`/CSP, HSTS. Zero code change, big hardening win.

Tier 2 — moderate, broadens usefulness

6. **JSON helper for constants** — a small `toJSON()` / `Server.CreateObject("MSXML2.ServerXMLHTTP")` wrapper so custom scripts can build modern AJAX/API endpoints cleanly (the docs already hint at this capability).
7. **Native `` YouTube/embed helper done right&amp;amp;lt;/b&amp;amp;gt; (you fixed `QS_iFrameYT`; make it return-correct and support Vimeo/maps).&amp;amp;lt;/li&amp;amp;gt;&amp;amp;lt;li&amp;amp;gt;8. &amp;amp;lt;b&amp;amp;gt;Lazy-loading &amp;amp;amp;amp; responsive `&amp;amp;lt;img&amp;amp;gt;`&amp;amp;lt;/b&amp;amp;gt; in gallery/catalog/page-media output (`loading="lazy"`, `srcset`) — instant performance/SEO.&amp;amp;lt;/li&amp;amp;gt;&amp;amp;lt;li&amp;amp;gt;9. &amp;amp;lt;b&amp;amp;gt;Open Graph / Twitter Card meta tags&amp;amp;lt;/b&amp;amp;gt; from page `sSEOTitle` / `sDescription` / first image — better social sharing.&amp;amp;lt;/li&amp;amp;gt;&amp;amp;lt;li&amp;amp;gt;10. &amp;amp;lt;b&amp;amp;gt;`prefers-color-scheme` dark-mode hook&amp;amp;lt;/b&amp;amp;gt; in the default template/theme CSS.&amp;amp;lt;/li&amp;amp;gt;&amp;amp;lt;/ol&amp;amp;gt;&amp;amp;lt;/div&amp;amp;gt;&amp;amp;lt;div data-bbox="113 825 419 843" data-label="Section-Header"&amp;amp;gt;&amp;amp;lt;h4&amp;amp;gt;**Tier 3 — larger, future-proofing**&amp;amp;lt;/h4&amp;amp;gt;&amp;amp;lt;/div&amp;amp;gt;&amp;amp;lt;div data-bbox="113 855 773 893" data-label="List-Group"&amp;amp;gt;&amp;amp;lt;ol&amp;amp;gt;&amp;amp;lt;li&amp;amp;gt;11. &amp;amp;lt;b&amp;amp;gt;Parameterized query helper&amp;amp;lt;/b&amp;amp;gt; (`db.q(sql, args())`) to migrate off string concatenation gradually.&amp;amp;lt;/li&amp;amp;gt;&amp;amp;lt;/ol&amp;amp;gt;&amp;amp;lt;/div&amp;amp;gt;**

12. **SMTP/TLS modernization** for CDO mail (the `C_SMTP`` fields exist; ensure SSL/587 works with modern providers).

13. **`sitemap.xml` + structured data (JSON-LD)** generation from the page tree.

14. **Static-asset cache-busting** (version query string) tied to `updatedTS``.

Suggested next step

I'd recommend implementing **Tier 1** as a single low-risk cleanup pass (dead code + error hygiene + the `cleanUp`` hardening + `web.config` headers), since it's safe and improves the codebase immediately.

A few decisions for you before I'd start:

1. Should I do a **full orphan-file sweep** (every `bs_`/`ad_`` page) and include a delete list, or leave file deletions out and only fix code?
2. For Tier 1, do you want it as **one batch** or separate changes per concern (easier to review/revert)?
3. Any of the Tier 2/3 items you want pulled forward into the first implementation round?