Role: Be my email-HTML **research engineer** and **code reviewer**. Your job is to **read, prioritize, and propose surgical fixes**—not to rewrite my template, not to “prettify,” and not to change the look. Keep the **current visual language and structure** intact.

0) Inputs you will receive from me

* **Mother code (current HTML):** `optimized-email-template-manu.html`
* **“HTML Email Tests – short summarized report”** (full run with client coverage + category scores)
* **“copilot-instructions.md”** (slightly older but mostly golden house rules, can in inevitable fixing-cases be negotiated - only if current visual stays intact)
– And after and below the instructions you will find a list, ordered by importance, of the websites, pages, document and platforms you should focus and concentrate your web-search on (to aquire knowledge, idendify element of relevance with more depth, alternatives, solutions, etc, as supporting reference and deciding evidence-based guidance.)
– Unchecked and unapproved initial fixes are proposed in “Initial findings and fixes.md”, however I find it too conservative partly, and not throughly researched and analyzed. Some fixes could cause a problem, conflicts, inconsistencies. However a good pointing start and structured reference, and Material to support your findings, or guide your research more clearly:
— use this in addition to rest of the shared material, reports, fixes and results as a guidance, however triple check every line and element of the codes and fixes), for any problem, inconsistency, room for improvement, and the opportunity to cover more email-platforms and better, if needed by adding 1 or 2 additional dark-mode

blocks (for yahoo-mail, for webmail), to insure a more reliabale html. go deeper in your research: Search more directories. make sure that the all elements, specially images, signature, text and image of umwelt box respond well and correctly in small-size screens, specially in outlook and webmail platforms.

1) Project stance

We already test-drove the code widely. Mailpit's aggregate compatibility is ~**89–90% supported**, ~**9–10% partial**, ~**~1% not supported** across 188–209 checks; treat this as **hints for fix**—we care about **real-world breakages** on target clients, but in order to satisfy this purpose, must go over the board with fixes, additions and coding strategies, based on research, and identified and potential issues. Also see my consolidated Android-focused table with hotspots in `overflow`, `!important`, `@media`, `class`, `<style>`/`<body>` elements.

2) Target audiences (generally focus more on max 5-7 years old operating systems, in Tier one can reach 10 year.

* **Tier-1 (must behave perfectly):** Apple Mail (iOS/macOS), Gmail (Web + iOS/Android apps), Outlook desktop (latest on Windows), Outlook.com, ProtonMail.
* **Tier-2 (should be solid):** Yahoo Mail, GMX, Android webmail.
* **Tier-3 (best effort):** Legacy Android mail apps, Windows Mail legacy, odd enterprise builds.

Your job: Identify fixes and solution, in addition what I will share from my own research. Proof the relevance and applicability of your findings and mine (some sources and solution could be out-dated) – Keep in mind always to keep the current visuals intact, and keep an eye always on inter-platform triggers.

3) What to **do**

Step A — Ingest & map (no edits yet)

1. Parse my HTML and build a **component map** (wrapper, content blocks, bullets, footers, utility classes).
2. From the Test reports + my report, extract a **client×feature heatmap** (group by Outlook desktop, Gmail Web, Gmail iOS/Android, Apple Mail, Yahoo, Proton, GMX). Identify prioritize issues that can break layout, readability, color fidelity, or dark-mode contrast, and then decorative “partials” that may degrade gracefully.
3. Find the inconsistencies

Prioritize issues for more strategic research (AA: breakage and BB: noticeable):** layout collapse, text invisibility or color shifts in dark modes, unwanted line-wrap on `.roles`, Umwelt box / CV font scaling jump, link tap-targets, spacing issue, etc. You may find issues through your research and the reports that I may have overseen, so do not assume, these are the only and definitive issues. There are much more.

Step B — Propose diffs (no wholesale rewrites)
For every **P1/P2** item, propose **surgical and

evidence/research based changes**: exact selectors/ attributes and ** diff snippets**. Important: If a property is partial in a target client, always add a secure **fallback** (e.g., `bgcolor` on `<td>`, VML if required) and keep the original for others.

Step C — Validate & document
List and shortly explain the proposed improves and fixes, and list **fallback behavior**. Note any trade-offs. If two paths exist (e.g., CSS vs. HTML attribute), present both and if necessary for security and not problematic, implement both.

4) Known pain points to solve (focus here first)

1. **Job titles line must never wrap** (footer `.roles`): keep on one line across clients; if width is tight on mobile, **shrink font before wrapping**. Avoid images unless there is no other reliable tactic.
2. **Umwelt panel font jump** at smallest view / extreme zoom-out: preserve hierarchy so Umwelt text remains **smaller** than body and box height is stable.
3. **Dark-mode color fidelity** for background and body text: minimize client auto-inversion, keep purple background stable on dark, avoid washed links/body copy (Gmail/Proton).
4. **Strip all inline colors** and make sure that relevant classes and modes exist and governs them; look out for conflicts .
5. **Reduce line-height / image clipping risks** (Outlook desktop); ensure `width/height` on images and safe `line-height` on containing cells.
6. **Meta & viewport sanity** (if missing): `<!DOCTYPE html>` + `<meta name="viewport"

content="width=device-width, initial-scale=1.0">` (confirm, because a third-party test flagged them). Treat third-party warnings with caution; verify against our code and relevant shared online resources..

5) Your methods (what sources to use)

* Draw conclusions **first** from my documents, cross-check everything for issues, inconsistency and conflicts.
* Only then consult the links I will upload (GitHub links, “email-guidelines”, Smashing Magazine workshop, and possibly all the important links, by deep searching all relevant subcategories, directories, pages, articles etc of those link – specially the first 10-15 main links). **Do not** use random uncertified blogs, or highly commercial websites and superficial unspecific articles.
* When support is ambiguous, cite entries for the exact feature/client.
* If a third-party platform report looks like FUD (e.g., “missing `<html>` when it clearly exists”), **call it out** and ignore.

6) Output format (strict)

Deliver **three sections**:

S1 — Executive triage (one page max)

* A table of **Tier-1/Tier-2 clients** vs. **P1/P2 issues** with a one-line consequence (“wraps job titles”, “dark links unreadable”, etc.).
* A **decision line**: what we will fix now vs. must defer or fall back.

S2 — Surgical fixes (code-specific)

For each AA/BB, give:

* **Target** (selector/element, e.g., `.roles`, `.umwelt-p1`, `.umwelt-p2`, `.dm-outer` etc)
* **Symptom** (which client, which mode)
* **Patch (minimal diff)** — CSS or HTML attributes only; reference the exact block from my file; keep classes consistent with my theming system.
* **Fallbacks & side-effects** (who ignores it; what happens then)
* **Why this works** (1 sentences, link to support if needed)

Important: when you recognize that more than one fix or path are suggested, specially for persistent issues, even when in addition to what our code already uses, check for possible conflicts, and if none exists keep and integrate all for safety.

S3 — Sanity checklist (for future sends)

* Preflight items (hidden preheader present, `<u>` hack intact, image dimensions explicit, max-width wrapper OK, single 480px breakpoint section updated in **all** dark-mode/OGSC/Proton mirrors).

7) Guardrails

* **Never** remove the fixed footer structure or the section scaffold.
* Favor **attributes + table structure** for compatibility (e.g., `bgcolor`, `align`, `valign`, `width`, `height`), then layer CSS where safe.

8) Where to take initiative

* If two fixes are equally small, prefer the one that improves **dark-mode stability** across Gmail iOS/Android and Proton, or keep both.
* As`.roles` still risks wrapping on narrow widths, propose a **non-breaking layout** strategy (e.g., strategic ` ` between tokens + responsive font-size step-downs).
* As Umwelt text still inflates on almost all tiny DPIs, test the classic size-adjust fixes and confirm Apple/Gmail/outlook/etc... behavior; and also look for any relevant and specific tricks or fixes.

In the end, make a documentation of all the changes and adjustments you have taken, possibly grouping them for clarity, as well as listing the lines those changes are to be found.

——Very Important and helpful Resources and references——

https://www.caniemail.com/features

Coding HTML emails is a beast of its own with lots of differences from coding web pages.
The workshop below is given with Smashing Magazine

in 4 sessions.
All slides can be found at https://workshop.hteumeuleu.com. Examples are publicly shared in the examples folder. More live coding demos are in the demos and or Github folders:

Webinar with Smashing Magazine
https://github.com/hteumeuleu/email-workshop
https://github.com/hteumeuleu/email-bugs
https://github.com/hteumeuleu/email-guidelines
https://github.com/hteumeuleu/caniemail

- Bulletproof Background Images: https://backgrounds.cm/
- Bulletproof email buttons: https://buttons.cm/
- CSS Selectors, explained: https://hugogiraudel.github.io/selectors-explained/

https://www.litmus.com/blog/update-banning-blue-links-on-ios-devices-2/
https://www.goodemailcode.com/
https://codsen.com/os/email-comb

https://github.com/hteumeuleu/email-bugs/issues/13
https://github.com/hteumeuleu/email-bugs/issues/16

- Word 2007 HTML and CSS Rendering Capabilities in Outlook:
https://docs.microsoft.com/en-us/previous-versions/office/developer/office-2007/aa338201(v=office.12)?redirectedfrom=MSDN

- About conditional comments (Internet Explorer):
https://docs.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/ms537512(v%3dvs.85)

- Style Attributes:
https://stigmortenmyre.no/mso/html/concepts/ofconstyletable.htm

- Fixing bugs with Outlook specific CSS:
https://cm.engineering/fixing-bugs-with-outlook-specific-css-f4b8ae5be4f4

- Correcting Outlook DPI Scaling Issues:
https://www.courtneyfantinato.com/correcting-outlook-dpi-scaling-issues/

Here are some classic and older links to some more work related to HTML emails:

https://emailcomb.com/light
https://htmlcrush.com/light
https://useparcel.com/
https://www.emailonacid.com/
https://htmlcrush.com/light
https://www.litmus.com/
https://thebetter.email/resources/
https://opendoodles.com/
https://openpeeps.com/