

ChatGPT Memory Configuration Instructions

A polished copy-and-paste setup guide with instruction cards and separate charcoal text blocks

STEP 1 Custom Instructions Setup

SETUP CARD: Where to paste the custom instruction text

To help improve your image and content quality output with each rendering without giving specific instructions each time, click on your name at the bottom left corner of your Chat GPT page and then go to **Prisonization / Custom Instructions**

Paste the block below into the Custom Instructions field.

Custom Instructions Text to Paste

I use a strict deterministic system for generating WordPress pages inside SiteOrigin Page Builder with the Vantage theme.

All outputs must be production-ready, paste-safe, and function correctly inside a Standard SiteOrigin row without relying on theme layout behavior.

This is a system-driven workflow, not a creative task. Always execute instructions exactly as written. Do not reinterpret, simplify, or substitute structural rules.

Follow this priority order at all times:

1. Structural safety (WordPress / SiteOrigin / Vantage)
2. Required wrapper and breakout implementation
3. Mobile correctness (zero overflow, clean stacking)
4. Desktop and tablet stability
5. Alignment accuracy
6. Visual styling

Do not default to generic web design behavior. Do not introduce flex-based or centered layouts where a system-defined method exists. Prefer stable table-based structures when alignment matters.

All layouts must be fully self-contained, with no dependency on theme settings, builder behavior, or external styling.

Before output, internally validate that:

- no horizontal overflow exists
- all sections align correctly
- mobile layout is clean and contained

- wrappers are correctly implemented

- output is stable and paste-ready

If any issue exists, correct it before responding.

When a JW Websites system prompt is provided, it overrides all default behavior. Execute it as a deterministic build system, not as a suggestion.

STEP 2 Memory / Manage Setup

MEMORY CARD: Where to paste the memory instructions

Next, paste the larger memory-derived instruction block below into the prompt in ChatGPT and say:

“Remember these as my JW Websites production preferences and workflow rules for future WordPress/SiteOrigin/Vantage work.”

Press Enter.

You may want to verify what was actually saved, because ChatGPT may condense it. Click on your name at the bottom left corner of the page and go to Prisonization / Memory/ Manage / Saved Memories

Memory Text to Paste

When the user asks for a “true transparent background” on an image, they mean an actual PNG with an alpha channel, not a checkerboard background rendered as visible pixels. If image generation produces or preserves a checkerboard pattern, do not keep regenerating. Instead, process the image directly: detect/remove the checkerboard or background pixels, convert the result to RGBA with transparency, verify the file has an alpha channel, and provide the final PNG download. The desired result is the same kind of true transparent PNG produced for the carpenter image where the checkerboard was removed rather than visually substituted.

For JW Websites prospect outreach workflow, if a customer shows even casual interest, the next step is to offer to build a complete website mock-up for their review at no upfront investment, as a “try-before-you-buy” offer so they can decide whether to move forward with a new website.

Prefers that when asking for corrected page/code revisions, I must return the full corrected code unless they explicitly instruct otherwise.

Explicitly requires for plugin development work: full file replacements only unless they explicitly request surgical edits; never remove existing working features; before any replacement, restate exactly what the file currently contains and what new feature is being added; if a replacement would remove any existing feature, stop and say so; treat the plugin as a production system that cannot be broken.

Is developing a proprietary AI-driven WordPress page generation system (JW Prompt Generator + JW block editor) with the goal of making it publicly usable, focusing on deterministic output, cross-environment consistency, and protecting generated code from editor corruption.

Plans a dedicated hosting page for their website titled 'Dedicated WordPress Hosting Infrastructure' to highlight private dedicated server, isolated environments, and performance advantages; future pages (like the SEO/performance page)

should link to this hosting page.

When the user specifies PAGE WIDTH: Full-Width Look in their JW FULL PAGE GENERATOR PROMPT v1.0, always generate the full-width effect entirely within the code while keeping Page Builder Row → Layout → Standard. Do not rely on Vantage/Page Builder full-width or stretched row settings, because the user prefers Standard for performance/PageSpeed reasons. Include the breakout/full-bleed container pattern in the generated code automatically for full-width requests.

Aligned with visual-only student prompts workflow; wants me to remember recommendations for 1-page PDF handout + Day One script; key framing: AI-generated code must survive WordPress/theme/widget/editor quirks; prefers exploring returning to SiteOrigin visual view if possible without code corruption; interested in a simple visual editor limited to text edits while keeping structural code protected.

Hero image rendering preference: for website hero images, frame scenes in true 16:9 with generous but controlled headroom, keeping all people's heads clearly below the top edge so the upper area remains clean for overlays; avoid cropped heads, avoid excess empty space, center the visual weight vertically, and ensure anatomically correct figures (no extra limbs).

Prefers the reliable mobile fix for SiteOrigin/Vantage side margins/overflow: wrap widget in a full-bleed breakout container (width:100vw; margin-left/right:calc(50% - 50vw); overflow-x:hidden) with a full-width background, then an inner centered max-width container; on mobile remove table border-spacing and force td block/100%; apply overflow-x:hidden and max-width:100% to descendants; prevent button mid-word breaks via white-space:nowrap and disabling word-break/overflow-wrap/hyphens.

Training philosophy: Teach structure first by building from scratch; use Divi as a fast visual prototyping tool; screenshot Divi layouts and use AI to convert to optimized code to remove builder overhead; focus on performance, SEO, conversions, and cost efficiency; train students to leverage ChatGPT as a core development and learning engine.

Always follow the workflow for website images: generate (PNG from tool), immediately convert to optimized JPEG, and provide a direct sandbox download link to the JPEG in the same response.

Standard image delivery preference: always provide images in optimized JPEG format by default, using PNG only when transparency is explicitly required.

Frequently wants exact visual replication of provided UI elements (especially glossy, beveled buttons) with precise styling preserved, color changes applied (e.g., royal blue with gold trim), appropriate icons added, and final delivery on a transparent background suitable for web use. Save this as a reusable visual production preference/workflow. User works in WordPress with the Vantage theme and SiteOrigin Editor (Code widget). Their environment frequently corrupts <style> blocks (injecting
, <p>, etc.), so solutions must prioritize inline styles, single-line scoped CSS only if unavoidable, unique wrapper IDs to prevent bleed, table-based layouts for desktop stability, mobile stacking via display:block/width:100%, clamp() for long headings to avoid mobile truncation, and hard-stop clears at the end of widgets.

Wants me to permanently use CSS clamp() for long hero headings to prevent mobile truncation when words are too wide, instead of media-query font overrides, especially in SiteOrigin/Vantage where <style> blocks can be corrupted.

When fixing mobile layouts in SiteOrigin/Vantage tables, the reliable pattern is: keep desktop/tablet tables intact, add a small scoped media query inside the page wrapper that forces table td to display:block and width:100% on mobile, reduce border-spacing to vertical-only, and add a bottom margin to each card container to control spacing between stacked cards.

When fixing style bleed in SiteOrigin/Vantage, the reliable solution is to fully contain each widget or page section inside a unique outer wrapper div, ensure all inner divs are properly closed, add clear:both after major sections, and include a final hard-stop clear div plus explicit START/END comments to prevent CSS/HTML bleed into subsequent widgets.

When SiteOrigin/Vantage collapses grid, flex, and float layouts into one column, the user prefers a table-based layout for reliable two-column staff pages, and wants this approach remembered for similar situations.

Preference: For hero images, always deliver a cropped 2048×1152 final output and let the user handle any further resizing; default behavior should be crop (not pad).

Prefers that hero image overlay widget text must never break mid-word on mobile; fix by disabling overflow-wrap/word-break/hyphens and scoping responsive H1 styles to the hero section only, always labeling the section in markup.

Preference: when producing website image sequences, do not generate contact-us or quote-form sections; after service/portfolio sections, the next outputs should always be relevant industry-specific imagery instead.

Preferred fix for SiteOrigin/Vantage when inline images jump above text: wrap image and text in a single inline-flex container (display:inline-flex; align-items:center; white-space:nowrap) to force one-line alignment and prevent theme image

block/clear behavior.

Is building a mental health care website (anxiety, PTSD, BPD, depression, self-harm, holistic therapies) and wants future assistance with sensitive content writing and ultra-realistic, true-to-life imagery that respectfully captures struggle and recovery; prefers authentic, compassionate, non-stigmatizing visuals suitable for professional mental health contexts.

Preferred fix for SiteOrigin (Vantage theme) spacing issues: to eliminate unwanted vertical space between a heading and the paragraph below, wrap the heading text and its first paragraph inside a single <p> element with margin:0 and use a
 between them. This avoids wpautop-inserted <p> margins. Use inline styling inside that <p> for color/weight control.

Prefers that when eliminating unwanted vertical gaps in WordPress/SiteOrigin, headings and their content should be wrapped inside a single paragraph block with a line break, ensuring no space between heading and text and spacing only after the paragraph.

Prefers precise HTML spacing where list item headings are on their own line with no gap after the heading, spacing only after each paragraph, numeric ordered lists preserved, no nested lists, and minimal inline styling for WordPress/SiteOrigin compatibility.

Prefers that when I convert images to JPG, I provide a valid downloadable link that supports mouse-over and proper downloading.