LongWebBench: Evaluating Structural and Functional Webpage Generation in Long-Horizon Settings

Yi Zhao[1*], Zhen Yang[1*], Mengpan Chen[2], Mingde Xu[3], Shanghui Gong[2], Xijun Liu[4], Jibing Gong[2†], Jie Tang[1†],
[1]Tsinghua University, [2]Yanshan University, [3]University of Waterloo, [4]Beihang University

Abstract

Recent vision-language models (VLMs) have shown promising progress in generating webpages from visual inputs, yet existing evaluations mainly focus on short, single-screen, and largely static webpages. We introduce **LongWebBench**, a benchmark for evaluating long-horizon webpage generation from both structural and functional perspectives. LongWebBench contains 490 real-world long webpages for structural fidelity evaluation and 507 goal-oriented interaction tasks over 129 webpages for functional evaluation. It employs two complementary protocols: a multi-dimensional VLM-based metric for assessing long-range structural coherence, and a DOM-augmented agent-based pipeline for end-to-end functional verification. We further examine the automatic evaluation protocols through human agreement analysis. Experiments with state-of-the-art open-source and proprietary VLMs under single-image and multi-image settings reveal that structural fidelity degrades as webpage length increases, while visually plausible generations often fail to support executable multi-step interactions. These results highlight the need to evaluate long webpage generation beyond visual similarity, with executable interaction as a core criterion. Our code and data are available at `https://github.com/zhaoyi-hb/LongWebBench`.

1 Introduction

Recent vision-language models (VLMs) (Liu et al., 2023; Wang et al., 2024; Yang et al., 2025; Wang et al., 2025; Team et al., 2025a) have shown promising progress in generating front-end code from visual inputs such as webpage screenshots. Given a rendered interface, these models can produce HTML, CSS, and JavaScript that visually resemble the target page (Jiang et al., 2025; Liang et al., 2025). However, existing efforts (Yun et al., 2024; Tan et al., 2025; Niu et al., 2025; Zhao et al., 2025; Wu et al., 2025) largely focus on short, single-screen, and mostly static webpages, where success is primarily measured by local visual similarity. Such settings provide limited evidence about whether models can generate long webpages that preserve global structure and support executable user interactions.

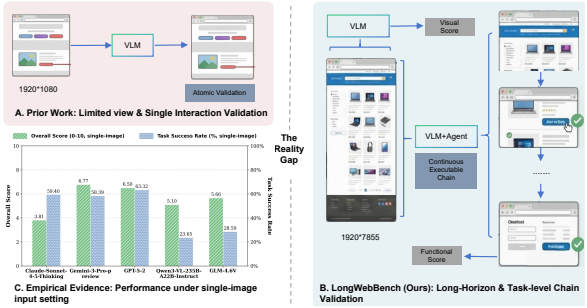


Figure 1: Motivation of LongWebBench. Existing evaluations mainly consider limited-view reconstruction or isolated interactions, while LongWebBench evaluates long webpages through both structural fidelity and task-level functional verification. Empirical results under the single-image setting illustrate that current VLMs still face challenges in both long-range visual reconstruction and executable task completion.

As illustrated in Figure 1, this results in an evaluation gap between limited-view webpage reconstruction and long-horizon webpage generation with executable interactions. Real-world webpages often span multiple screens, contain repeated components, require style consistency across distant sections, and support multi-step user goals. These properties introduce two challenges beyond short static reconstruction. Structurally, models must preserve page scale, global layout, section hierarchy, visual styling, and information density across the full webpage. Functionally, models must synthesize interaction logic that supports user goals under browser execution. A page may look plausible while still failing to open menus, filter content,

[*] Equal contributions
[†] Corresponding authors

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 46064–46115
October 24-29, 2026 ©2026 Association for Computational Linguistics

Table 1: Comparison with existing webpage-to-code benchmarks. Existing benchmarks mainly focus on short-page visual validation or isolated interaction validation, while LongWebBench supports long webpages, multi-image inputs, and goal-driven functional validation.

Benchmark	Visual Validation	Single-Interaction Validation	Functional Validation	Long-WebPage Support	Multi-Image Support
Web2Code (Yun et al., 2024)	✓	×	×	×	×
Interaction2Code (Xiao et al., 2024)	✓	✓	×	×	×
Webcode2M (Gui et al., 2025)	✓	×	×	×	×
Design2Code (Si et al., 2025)	✓	×	×	×	×
WebGen-Bench (Lu et al., 2025)	✓	✓	✓	×	×
Sketch2Code (Li et al., 2025)	✓	×	×	×	×
LongWebBench	✓	✓	✓	✓	✓

navigate across sections, or submit forms, suggesting that long-horizon webpage generation should be evaluated beyond static visual similarity.

Despite growing interest in webpage-to-code evaluation, existing benchmarks do not jointly cover these requirements. As summarized in Table 1, prior benchmarks mainly emphasize visual fidelity on short webpages (Yun et al., 2024; Gui et al., 2025; Si et al., 2025; Li et al., 2025) or isolated interaction validation (Xiao et al., 2024). Although WebGen-Bench (Lu et al., 2025) evaluates goal-driven functionality, it does not target screenshot-driven long webpage generation or long-scroll structural consistency. Consequently, existing benchmarks do not jointly assess global structure across long webpages and executable multi-step user interactions.

To address this gap, we introduce **LongWebBench**, a benchmark for evaluating long-horizon webpage generation from both structural and functional perspectives. LongWebBench contains two complementary tasks. *Webpage Visual Fidelity Replication* (W-VFR) evaluates whether a model can reproduce the global structure and visual organization of long webpages, using 490 real-world long webpages across diverse categories. *Webpage Functional Fidelity Realization* (W-FFR) evaluates whether a model can generate executable webpages that support goal-oriented user interactions, using 507 interaction tasks over 129 webpages. These two tasks separate visual fidelity from functional correctness, enabling a more diagnostic evaluation of webpage generation.

LongWebBench further provides corresponding evaluation protocols: a multi-dimensional VLM-based metric for long-range structural coherence, and a DOM-augmented agent-based pipeline that executes generated webpages in a browser environment to verify user-goal completion. For structural evaluation, we additionally incorporate DINO-based feature similarity as an auxiliary signal for local visual styling consistency. The benchmark supports both single-image and multi-image input settings, allowing us to analyze model behavior under different long-page input constraints.

We evaluate state-of-the-art open-source and proprietary VLMs on LongWebBench under both single-image and multi-image input settings. Results show that long-horizon webpage generation remains challenging: structural fidelity generally degrades as webpage length increases, while visually plausible generations often fail to support executable multi-step interactions. These findings highlight the need to evaluate webpage generation beyond static visual similarity, with executable interaction as a core criterion.

Our contributions are summarized as follows:

- We introduce **LongWebBench**, a benchmark for long-horizon webpage generation with structural and functional evaluation.
- We construct two complementary tasks: W-VFR with 490 real-world long webpages and W-FFR with 507 goal-oriented tasks over 129 webpages.
- We design evaluation protocols for long-range structural fidelity and end-to-end functional verification.
- We evaluate state-of-the-art VLMs under single-image and multi-image settings, revealing challenges in structural consistency and executable functional interaction.

2 LongWebBench

We present the design of **LongWebBench**, a benchmark for evaluating long-horizon webpage generation from both structural and functional perspectives. We first define long webpages and the two evaluation tasks, then describe the construction of W-VFR and W-FFR, and finally analyze the dataset coverage and complexity.

2.1 Task Definition

Long Webpage. We define a long webpage as a webpage whose vertical length exceeds three viewport heights under a fixed 1920 × 1080 browser resolution, requiring models to aggregate visual information across multiple screens.

Webpage Visual Fidelity Replication (W-VFR). Given a target long webpage screenshot, W-VFR evaluates whether a model can preserve the target visual structure, including page scale, global layout, section hierarchy, visual styling, and information density. Functional correctness is not considered, and interactive elements are treated as static visual components.

Webpage Functional Fidelity Realization (W-FFR). Given a target webpage screenshot and a set of user-goal tasks, W-FFR evaluates whether a model can generate an executable webpage that supports the specified user goals. The screenshot provides component and state context, while evaluation focuses on whether prescribed actions lead to expected outcomes under browser execution. Visual similarity is not required.

2.2 Data Construction

W-VFR: Long Webpage Collection. We construct W-VFR from real-world webpages with diverse long-form structures. We define seven structural categories: informational articles, reference and documentation, product and service pages, community and discussion pages, multimedia and interactive pages, governmental and institutional pages, and data and resource aggregation pages. For each category, annotators collect candidate URLs from representative public websites to cover diverse layout patterns, content densities, and section organizations.

We then apply a multi-stage curation pipeline to obtain high-quality long-webpage snapshots. Candidate pages are rendered with Playwright under a fixed 1920 × 1080 browser resolution, and pages are filtered out if they fail to render, do not satisfy the length requirement, require authentication or personalization, contain excessive overlays or sensitive personal content, or involve unbounded infinite scrolling. We remove near-duplicates using perceptual hashing over rendered screenshots and capture each remaining page as a stabilized rendered snapshot, isolating long-page generation from external backend services and uncontrolled

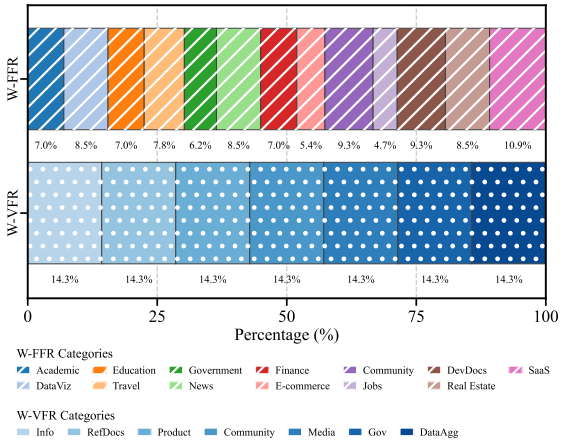
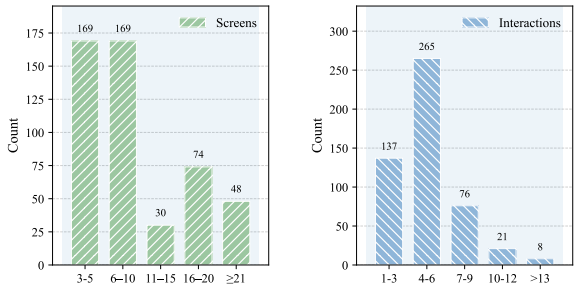


Figure 2: Category distribution of LongWebBench. W-VFR is balanced across 7 structural webpage categories, while W-FFR covers 13 user-goal-oriented functional categories.

personalization. All screenshots are manually verified to ensure correct rendering and absence of capture artifacts. After filtering and verification, W-VFR contains 490 long webpages, with 70 webpages in each category. More details can be found in Appendix A.1

W-FFR: Functional Task Construction. While W-VFR is organized around visual structure, W-FFR is organized around user goals. We first define 13 functional categories based on the primary services provided by modern webpages, including academic search, data visualization, education, travel, government services, news, finance, e-commerce, community, jobs, developer documentation, real estate, and SaaS. Candidate webpages are collected and curated using the same rendering and quality-control pipeline as W-VFR, ensuring that each selected page provides meaningful frontend-observable interactions.

For each selected webpage, we construct 3–4 user-centered tasks that reflect realistic interaction goals on that page. We follow a draft–review–rewrite–verify workflow: task drafts are first created following predefined task-design guidelines, and annotators then rewrite, validate, and finalize them based on the rendered webpage. During verification, annotators check whether each task is clear, feasible, non-redundant, executable in the rendered webpage, and grounded in frontend-observable state changes rather than external backend services. Tasks that are ambiguous, backend-dependent, redundant, or infeasible are removed or revised. This process yields 129 webpages paired with 507 functional tasks, where each task serves as



(a) Distribution of vertical viewport counts in W-VFR.

(b) Distribution of interaction steps in W-FFR.

Figure 3: Analysis of dataset complexity.

an independent evaluation instance. More details can be found in Appendix A.2

2.3 Data Analysis

Category Coverage. LongWebBench covers diverse webpage structures and user goals. For W-VFR, the dataset is balanced across seven structural webpage categories, with 70 webpages per category. For W-FFR, webpages are organized into 13 user-goal-oriented functional categories, covering a broad range of real-world service objectives. As shown in Figure 2, LongWebBench provides broad coverage across both visual structures and functional scenarios.

Long-Webpage Complexity. To characterize the long-horizon nature of W-VFR, we measure the number of vertical viewports required to cover each webpage under the fixed 1920×1080 resolution. As shown in Figure 3a, webpages exhibit a long-tailed distribution. Most webpages span 3–10 viewports, while pages exceeding 10 viewports account for about 31% of the dataset, with some extending up to 30 viewports. This distribution supports evaluation across different degrees of long-range visual structure and layout complexity.

Interaction Complexity. For W-FFR, we report the number of explicit user actions required to complete each task. As shown in Figure 3b, most tasks require medium-to-long interaction sequences: 52.27% require 4–6 steps, 20.71% require at least 7 steps, and 27.02% require 1–3 steps. This distribution reflects the benchmark's emphasis on coordinated, multi-step interactions rather than isolated UI actions.

2.4 Benchmark Design Principles

LongWebBench is designed to disentangle two capabilities that are often conflated in webpage-to-code evaluation: long-range visual-structural reconstruction and executable interaction realization. W-VFR evaluates whether models can maintain coherent structure across extended page contexts, including repeated sections, cross-section consistency, and long-range layout organization, among others. W-FFR evaluates whether generated webpages can support user-goal completion under browser execution, rather than merely rendering visually plausible interactive elements. We evaluate stabilized rendered snapshots and frontend-observable interactions, avoiding dependence on external backends while preserving the long-page structure and multi-step interaction patterns needed for reproducible evaluation.

3 Evaluation Protocols

We design two evaluation protocols for LongWebBench, as illustrated in Figure 4. W-VFR evaluates long-range structural fidelity by comparing generated renderings with reference long webpages, while W-FFR verifies whether generated webpages support executable user goals under browser execution.

3.1 Evaluation Settings

LongWebBench supports two input settings. In the *single-image* setting, the full webpage screenshot is provided as one long image, testing whether models can process long visual contexts under their native image-input constraints. In the *multi-image* setting, the screenshot is split into consecutive viewport-sized tiles and provided in top-to-bottom order, reducing resolution constraints while requiring cross-image integration. These settings help separate input-size limitations from long-context structural reasoning challenges.

3.2 Structural Fidelity Validation

To evaluate W-VFR, we use a VLM-based relative evaluator that compares a generated full-page rendering with the reference long webpage screenshot. The evaluator scores the generated page based only on visually observable evidence, explicitly excluding interactive behavior. We use *GPT-4o* as the primary structural evaluator with deterministic decoding, and provide the full prompt, model version, and scoring rubric in Appendix B.1.

The evaluator assigns scores on a 0–10 scale across five dimensions: *page scale and scroll consistency*, *global layout consistency*, *section-level*

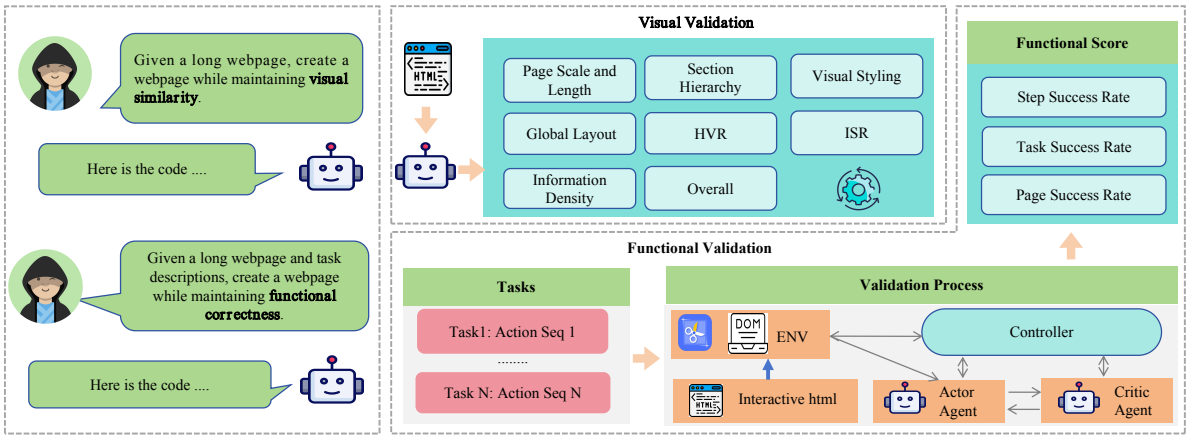


Figure 4: Overview of LongWebBench evaluation. W-VFR evaluates structural fidelity across five dimensions: page scale, global layout, section hierarchy, visual styling, and information density. W-FFR evaluates functional fidelity by executing goal-oriented interaction tasks in a browser environment and reporting step-, task-, and page-level success rates.

structure and hierarchy, *local visual styling consistency*, and *information density*. The overall W-VFR score is the average of the five dimension scores. To better capture low-level appearance consistency, the local visual styling dimension combines VLM-based scoring with DINO-based feature similarity.

We also report three availability metrics to separate input-side constraints from generation quality. *Input Support Rate* (ISR) measures the fraction of test cases accepted by the model or API before generation, *HTML Validity Rate* (HVR) measures, among supported cases, the proportion of outputs that contain a valid renderable HTML artifact, and *Code Quality* (CQ) measures the overall quality of the generated code. Detailed definitions and scoring procedures are provided in Appendix B.2.

3.3 Functional Fidelity Validation

To evaluate W-FFR, we use a task-driven execution-based protocol that verifies whether a generated webpage supports specified user goals under browser execution. Unlike W-VFR, functional fidelity does not evaluate visual similarity to the reference screenshot. The screenshot and task descriptions provide component and state context, while evaluation focuses on whether prescribed interactions can be executed and lead to the expected outcomes.

Each task is evaluated independently in a controlled browser environment. A controller coordinates an actor agent, which executes the prescribed operations, and a critic agent, which checks whether the resulting DOM and rendered page state satisfy the task specification; both agents are powered by *Gemini-3-Pro*. A task is successful only if all required interaction steps are executable and collectively lead to the expected outcome. Implementation details and prompts are provided in Appendix B.3.

We report three functional metrics. *Step Success Rate* (SSR) measures the fraction of interaction steps that are executable and trigger the expected intermediate state change. *Task Success Rate* (TSR) measures the fraction of tasks completed successfully and serves as the primary W-FFR metric. *Page Success Rate* (PSR) measures the fraction of webpages for which all associated tasks are completed successfully, providing a stricter estimate of page-level functional robustness.

4 Experiments

4.1 Experimental Setup

Evaluated Models. We evaluate a broad set of state-of-the-art VLMs, covering both proprietary models (e.g., GPT-5.2 (Singh et al., 2025), GPT-4o (OpenAI et al., 2024), Claude-Opus-4.5, Claude Sonnet-4-5-Thinking, Gemini-3-Pro (Comanici et al., 2025), Gemini-3-Flash, and Doubao-Seed-1-6) and open-source models (e.g., GLM-4.6V , GLM-4.1V-9B-Thinking-Flash (Team et al., 2025b), Qwen3-VL-235B-A22B-Instruct (Yang et al., 2025), Qwen3-VL-8B-Instruct, InternVL3-78B (Wang et al., 2025), and Kimi-VL-A3B-Thinking (Team et al., 2025a)). All models are evaluated with the same generation prompts.

Evaluation Settings. We evaluate models under both single-image and multi-image settings. W-VFR is evaluated on the full model set when the input format is supported. W-FFR is evaluated

Table 2: Performance comparison of VLMs on LongWebBench under single-image and multi-image input settings. ISR, HVR, and CQ are reported as percentages (%); category-wise and overall scores are on a 0–10 scale (macro-average).

		Availability (%)		Average Score (0–10)								
Type	**Model**	**ISR**	**HVR**	**Comm.**	**Data**	**Gov.**	**Info.**	**Prod.**	**Ref.**	**Trans.**	**Overall**	**CQ**
				Single-Image Input								
Open-source	Kimi-VL-A3B-Thinking	100.00	78.98	2.69	2.64	3.36	2.20	3.02	2.20	3.95	2.87	92.50
	Qwen3-VL-8B-Instruct	95.71	54.37	3.72	3.64	3.92	3.17	3.56	2.02	4.28	3.47	88.38
	InternVL3-78B	100.00	81.84	3.48	3.43	3.85	3.42	3.35	2.78	3.50	3.40	94.41
	GLM-4.1V-9B-Thinking-Flash	100.00	84.90	5.32	4.27	5.15	5.32	5.18	3.34	5.54	4.87	92.95
	GLM-4.6V	100.00	84.29	5.85	5.55	6.56	5.64	5.62	3.93	6.44	5.66	94.06
	Qwen3-VL-235B-A22B-Instruct	99.80	80.78	5.29	5.19	6.23	5.72	4.44	3.40	5.45	5.10	89.20
Closed-source	Claude-Sonnet-4-5-Thinking	56.12	99.64	5.97	3.49	4.74	2.58	3.32	0.29	6.27	3.81	93.75
	Claude-Opus-4-5-20251101	56.12	99.64	5.95	3.34	4.70	2.49	3.40	0.34	6.67	3.84	93.14
	GPT-5.2	95.71	100.00	7.07	6.53	6.57	6.52	6.07	5.52	**7.20**	6.50	94.14
	GPT-4o	100.00	99.39	3.15	3.12	3.62	3.05	3.21	2.60	3.84	3.23	**95.59**
	Gemini-3-Pro-preview	100.00	100.00	**7.39**	**6.89**	**6.88**	**6.88**	**6.66**	**5.83**	6.89	**6.77**	93.00
	Gemini-3-Flash-preview	100.00	100.00	6.97	5.95	6.84	6.77	6.32	4.87	6.78	6.36	93.31
	Doubao-Seed-1-6	76.33	98.93	4.35	2.89	4.25	4.51	4.13	0.45	5.05	3.66	94.58
				Multi-Image Input								
Open-source	Kimi-VL-A3B-Thinking	100.00	80.20	3.00	3.05	3.52	2.70	3.33	2.56	3.63	3.12	89.88
	Qwen3-VL-8B-Instruct	100.00	54.69	4.14	3.81	4.94	3.31	4.16	3.23	4.02	3.95	86.72
	InternVL3-78B	99.39	77.21	3.53	3.73	3.51	3.51	3.65	3.67	3.72	3.62	**94.81**
	GLM-4.1V-9B-Thinking-Flash	100.00	83.06	1.46	2.71	1.12	0.39	0.31	0.31	0.28	0.94	92.59
	GLM-4.6V	100.00	88.16	5.76	6.16	6.50	5.62	5.65	5.79	6.25	5.96	93.54
	Qwen3-VL-235B-A22B-Instruct	99.80	80.57	6.08	5.68	6.38	5.98	5.00	6.03	5.83	5.85	87.43
Closed-source	Claude-Sonnet-4-5-Thinking	100.00	99.39	6.73	6.61	7.05	6.97	**7.07**	6.04	6.69	6.74	93.79
	Claude-Opus-4-5-20251101	100.00	97.96	6.83	6.28	6.51	6.13	6.51	6.25	6.49	6.43	93.07
	GPT-5.2	100.00	99.59	6.96	6.24	5.97	5.85	5.67	5.92	6.36	6.14	93.95
	GPT-4o	100.00	99.80	3.39	3.40	3.91	3.50	3.44	3.13	3.92	3.53	94.51
	Gemini-3-Pro-preview	100.00	99.59	**7.51**	**7.07**	**7.43**	**7.08**	6.99	**7.35**	**7.08**	**7.22**	92.25
	Gemini-3-Flash-preview	100.00	100.00	6.83	6.46	6.88	6.81	6.57	5.48	6.70	6.53	92.25
	Doubao-Seed-1-6	100.00	99.39	4.50	4.57	5.01	4.96	4.80	4.39	5.14	4.77	94.59

on a representative subset due to the higher cost of browser-based execution, covering both proprietary and open-source models with different input capabilities and performance levels. Additional implementation details are provided in Appendix E.2.

4.2 Structural Fidelity Results

Table 2 reports W-VFR results under both single-image and multi-image input settings. Overall, current VLMs still achieve only moderate structural fidelity on long webpages, showing that long-horizon webpage reconstruction remains challenging beyond local visual matching. Under the single-image setting, Gemini-3-Pro achieves the highest overall score, followed by GPT-5.2 and Gemini-3-Flash. Strong open-source models such as GLM-4.6V and Qwen3-VL-235B-A22B-Instruct are competitive, but still lag behind the best proprietary systems. Multi-image input improves input availability and often improves structural fidelity by reducing image-size constraints, especially for Claude models. However, the gains remain limited: even under the multi-image setting, the best model still reaches only moderate fidelity. This suggests that multi-image input mainly alleviates input-resolution constraints, but long-webpage generation still requires robust cross-image integration and global layout planning.

4.3 Functional Fidelity Results

Table 3 reports W-FFR results under both input settings. Due to the high cost of browser execution and agent-based verification, we evaluate a representative subset covering strong closed-source and competitive open-source models from W-VFR. Functional fidelity is substantially more challenging than step-level execution alone: models with high Step Success Rate (SSR) still show much lower Task Success Rate (TSR) and Page Success Rate (PSR), indicating error accumulation across multi-step workflows and multiple tasks on the same page. Claude-Sonnet-4.5-Thinking and Gemini-3-Pro perform best, reaching SSR above 86% and TSR around 58–59% under the

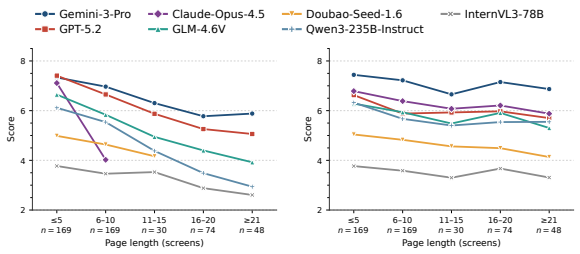
Table 3: W-FFR results under single-image and multi-image input settings. SSR, TSR, and PSR denote step, task, and page success rates, respectively.

Model	SSR	TSR	PSR
Single-Image Input			
GLM-4.6V	62.59	28.39	5.56
Gemini-3-Pro-preview	**87.30**	58.38	17.83
Qwen3-VL-235B-A22B-Instruct	60.52	23.65	3.13
Claude-Sonnet-4-5-Thinking	86.91	**59.40**	**21.78**
Multi-Image Input			
GLM-4.6V	64.20	30.52	7.03
Gemini-3-Pro-preview	**87.91**	**59.24**	**21.09**
Qwen3-VL-235B-A22B-Instruct	60.34	23.27	6.20
Claude-Sonnet-4-5-Thinking	86.46	55.47	17.19

single-image setting, while their PSR remains below 22%. Open-source models such as GLM-4.6V and Qwen3-VL-235B-A22B-Instruct achieve much lower page-level success, showing limited robustness in end-to-end functional realization. Multi-image input provides only limited and inconsistent gains, suggesting that executable interaction depends not only on input resolution, but also on correct interaction logic and state transitions.

4.4 Effect of Length on Structural and Functional Fidelity

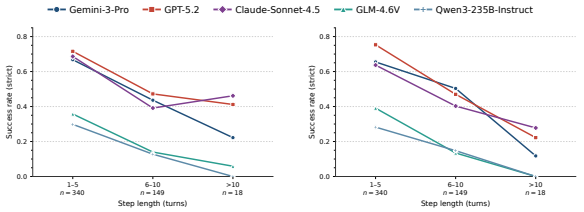
Webpage Length. Figure 5 shows W-VFR performance across webpages with different numbers of vertical viewports. Structural fidelity generally decreases as webpage length increases, indicating that long webpages require global layout planning, section ordering, and style consistency beyond local visual reconstruction. Multi-image input alleviates this degradation by preserving finer-grained local details, but performance still drops on very long webpages, suggesting that robust cross-segment integration remains challenging.



(a) Single-image input. (b) Multi-image input.

Figure 5: W-VFR performance across different webpage lengths.

Interaction Length. Figure 6 shows W-FFR success rates across tasks with different numbers



(a) Single-image input. (b) Multi-image input.

Figure 6: W-FFR success rate across different interaction lengths.

of interaction steps. Success rates generally decrease as interaction length increases, indicating that longer workflows amplify error accumulation and require more stable state tracking. These results show that W-FFR evaluates more than isolated clickability. Generated webpages must maintain correct interaction logic and state transitions across multi-step user goals, where early mistakes can propagate and prevent task completion.

4.5 Robustness of Structural Evaluation

To examine whether W-VFR scores depend on a single VLM evaluator, we conduct a cross-judge robustness analysis. We sample 100 test cases, each consisting of a reference screenshot and the corresponding generated rendering, from four representative models: GPT-5.2, Gemini-3-Pro, Qwen3-VL-235B-A22B-Instruct and InternVL3-78B. Using the same prompt and rubric, we score these samples with both GPT-4o and Gemini-3-Flash. As shown in Figure 7, the two judges produce highly consistent results. Their per-model mean scores are nearly collinear, with Pearson $r = 0.992$. Pairwise ranking agreement is $5/6$ (83.3%), with the only inversion occurring between GPT-5.2 and Gemini-3-Pro, whose scores differ by only 0.03 points. These results indicate that our structural evaluation is robust to the choice of judge model.

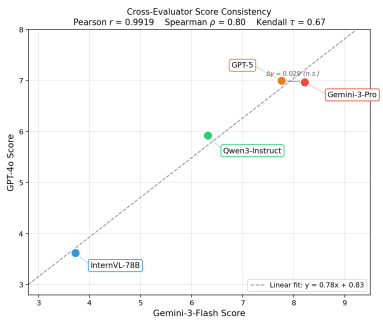
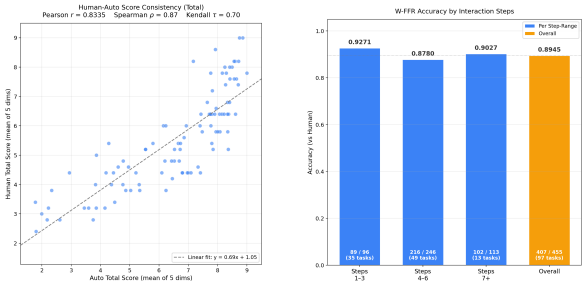


Figure 7: Per-model mean fidelity scores from two independent judge models.



(a) Overall-score correlation.

(b) Per step-range and overall W-FFR accuracy.

Figure 8: Agreement between automatic evaluation and human judgments. (a) W-VFR score-level agreement on 100 sampled outputs. (b) W-FFR task-level consistency across interaction-step lengths.

4.6 Agreement with Human Evaluation

To assess the reliability of LongWebBench's automatic evaluation protocols, we compare them with human judgments on sampled model outputs. For W-VFR, three annotators independently rate each generated webpage against the reference screenshot on the same 0–10 structural fidelity scale, and their average rating is used as the human score. For W-FFR, annotators manually inspect browser execution traces and assign binary success labels for executed interaction steps. As shown in Figure 8, the automatic protocols align well with human judgments. On W-VFR, automatic structural scores achieve strong agreement with human ratings over 100 sampled outputs, with Pearson $r = 0.8335$ and Spearman $\rho = 0.87$. On W-FFR, the automatic verifier reaches an overall consistency of 0.8945 over 455 evaluated steps from 97 tasks, and remains above 0.87 across different interaction-length groups. These results support the use of the automatic protocols for scalable evaluation of both long-range structural fidelity and executable functional correctness. Details of annotation and agreement analysis are provided in Appendix D.

5 Related Works

5.1 Visual-to-Code Generation

Visual-to-code generation aims to convert visual interface inputs into executable code (Zhao et al., 2026). Early work such as pix2code (Beltramelli, 2018) studied GUI screenshot-to-code, while recent VLM-based systems extend this paradigm to webpage and front-end code generation (Wang et al., 2024, 2025; Jiang et al., 2025). Since HTML/CSS implementations are non-unique and original source code is often unavailable, existing benchmarks commonly adopt render-based evaluation, including Web2Code (Yun et al., 2024), Design2Code (Si et al., 2025), Sketch2Code (Li et al., 2025), and WebCode2M (Gui et al., 2025). However, these benchmarks mainly focus on short, single-screen, or mostly static pages. Long webpages introduce additional challenges, including global layout consistency, section ordering, repeated modules, information density, and style coherence across multiple viewports. LongWebBench complements existing benchmarks by treating long-range structural fidelity as a central evaluation target and supporting both single-image and multi-image input settings.

5.2 Functional Webpage Generation

Realistic webpage generation requires executable interaction behavior beyond visual reconstruction. Recent benchmarks begin to evaluate interaction-aware generation: Interaction2Code (Xiao et al., 2024) focuses on isolated interactions such as clicks, hovers, and toggles, FullFront (Sun et al., 2025) uses DOM-level signals to assess front-end workflow components, and WebGen-Bench (Lu et al., 2025) studies functional website generation beyond static appearance. Related web-agent benchmarks such as WebArena (Zhou et al., 2023), Mind2Web (Deng et al., 2023), BrowserGym (Chezelles et al., 2024), and WebVIA (Xu et al., 2025) evaluate agents operating on existing webpages or browser environments. In contrast, LongWebBench evaluates whether a generated webpage itself contains the interaction logic and state transitions needed to support executable multi-step user goals.

6 Conclusion

In this paper, we introduced **LongWebBench**, a benchmark for evaluating long-horizon webpage generation from structural and functional perspectives. Unlike short, static visual reconstruction benchmarks, LongWebBench tests whether generated webpages preserve global structure across multiple viewports and support executable multi-step user interactions. Experiments with state-of-the-art VLMs show that structural fidelity degrades as webpage length increases, while visually plausible generations often fail to complete functional user goals. These findings highlight the need to evaluate webpage generation beyond static visual similarity and provide a diagnostic testbed for long-context

multimodal code generation.

Limitations

LongWebBench evaluates stabilized rendered webpages and frontend-observable interactions. This design makes the benchmark controllable and reproducible, but excludes authentication, personalization, live backend services, and unbounded infinite scrolling. Therefore, the benchmark targets long-page visual structure and executable frontend behavior rather than the full complexity of deployed web applications. The functional tasks in W-FFR are restricted to user goals whose success can be verified through browser execution, DOM states, and rendered page states. Tasks requiring external databases, payment systems, account-specific states, or hidden server-side logic are outside the current scope. In addition, although we validate the automatic evaluators through human agreement analysis and cross-judge robustness checks, automatic evaluation may still miss subtle visual or behavioral errors. Future work can extend LongWebBench toward controlled backend environments and broader real-world interaction scenarios.

Acknowledgement

This work was supported in part by the Natural Science Foundation of China (NSFC) (62425601), Daimler Greater China Ltd. and Tsinghua University Joint Institute for Sustainable Mobility, the New Cornerstone Science Foundation through the XPLORER PRIZE, the Young Scientists Fund of the National Natural Science Foundation of China (62506195), the China Postdoctoral Science Foundation Fellowship (2025M771572), the China Postdoctoral Program for Innovative Talents (BX20250381), the Central Government Guidance Local Science and Technology Development Fund Project (246Z0306G), and the Innovation Capability Improvement Plan Project of Hebei Province (22567626H).

References

Tony Beltramelli. 2018. pix2code: Generating code from a graphical user interface screenshot. In *Proceedings of the ACM SIGCHI symposium on engineering interactive computing systems*, pages 1–6.

De Chezelles, Thibault Le Sellier, Sahar Omidi Shayegan, Lawrence Keunho Jang, Xing Han Lù, Ori Yoran, Dehan Kong, Frank F Xu, Siva Reddy, Quentin Cappart, and 1 others. 2024. The browsergym ecosystem for web agent research. *arXiv preprint arXiv:2412.05467*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

Yi Gui, Zhen Li, Yao Wan, Yemin Shi, Hongyu Zhang, Bohua Chen, Yi Su, Dongping Chen, Siyuan Wu, Xing Zhou, and 1 others. 2025. Webcode2m: A real-world dataset for code generation from webpage designs. In *Proceedings of the ACM on Web Conference 2025*, pages 1834–1845.

Yilei Jiang, Yaozhi Zheng, Yuxuan Wan, Jiaming Han, Qunzhong Wang, Michael R Lyu, and Xiangyu Yue. 2025. Screencoder: Advancing visual-to-code generation for front-end automation via modular multimodal agents. *arXiv preprint arXiv:2507.22827*.

Ryan Li, Yanzhe Zhang, and Diyi Yang. 2025. Sketch2code: Evaluating vision-language models for interactive web design prototyping. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3921–3955.

Shanchao Liang, Nan Jiang, Shangshu Qian, and Lin Tan. 2025. Waffle: Fine-tuning multi-modal model for automated front-end development. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24786–24802.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Zimu Lu, Yunqiao Yang, Houxing Ren, Haotian Hou, Han Xiao, Ke Wang, Weikang Shi, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2025. Webgenbench: Evaluating llms on generating interactive and functional websites from scratch. *arXiv preprint arXiv:2505.03733*.

Tianhao Niu, Yiming Cui, Baoxin Wang, Xiao Xu, Xin Yao, Qingfu Zhu, Dayong Wu, Shijin Wang, and Wanxiang Che. 2025. Chart2code53: A large-scale diverse and complex dataset for enhancing chart-to-code generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15839–15855.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. 2025. Design2code: Benchmarking multimodal code generation for automated front-end engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3956–3974.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Haoyu Sun, Huichen Will Wang, Jiawei Gu, Linjie Li, and Yu Cheng. 2025. Fullfront: Benchmarking mllms across the full front-end engineering workflow. *arXiv preprint arXiv:2505.17399*.

Wentao Tan, Qiong Cao, Chao Xue, Yibing Zhan, Changxing Ding, and Xiaodong He. 2025. Chartmaster: Advancing chart-to-code generation with real-world charts and chart similarity reinforcement learning. *arXiv preprint arXiv:2508.17608*.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, and 1 others. 2025a. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, and 69 others. 2025b. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *Preprint*, arXiv:2507.01006.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Song XiXuan, and 1 others. 2024. Cogvlm: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Fan Wu, Cuiyun Gao, Shuqing Li, Xin-Cheng Wen, and Qing Liao. 2025. Mllm-based ui2code automation guided by ui layout information. *Proceedings of the ACM on Software Engineering*, 2(ISSTA):1123–1145.

Jingyu Xiao, Yuxuan Wan, Yintong Huo, Zixin Wang, Xinyi Xu, Wenxuan Wang, Zhiyao Xu, Yuhang Wang, and Michael R Lyu. 2024. Interaction2code: Benchmarking mllm-based interactive webpage code generation from interactive prototyping. *arXiv preprint arXiv:2411.03292*.

Mingde Xu, Zhen Yang, Wenyi Hong, Lihang Pan, Xinyue Fan, Yan Wang, Xiaotao Gu, Bin Xu, and Jie Tang. 2025. Webvia: A web-based vision-language agentic framework for interactive and verifiable ui-to-code generation. *arXiv preprint arXiv:2511.06251*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Sukmin Yun, Rusiru Thushara, Mohammad Bhat, Yongxin Wang, Mingkai Deng, Jinhong Wang, Tianhua Tao, Junbo Li, Haonan Li, Preslav Nakov, and 1 others. 2024. Web2code: A large-scale webpage-to-code dataset and evaluation framework for multimodal llms. *Advances in neural information processing systems*, 37:112134–112157.

Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2025. Chartcoder: Advancing multimodal large language model for chart-to-code generation. *arXiv preprint arXiv:2501.06598*.

Yi Zhao, Zhen Yang, Shuaiqi Duan, Wenmeng Yu, Zhe Su, Jibing Gong, and Jie Tang. 2026. Plotgen-bench: Evaluating vlms on generating visualization code from diverse plots across multiple libraries. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 27619–27650.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Appendix

A Data Annotation **12**
A.1 More details about W-VFR categories . . . 12
A.2 More details about W-FFR categories . . . 12
A.3 Distribution of webpage lengths across categories . . . 12

B Details of Evaluation Process **12**
B.1 More details about our validation process on W-VFR task . . . 12
B.2 More metric details about W-VFR . . . 15
B.3 More details about W-FFR process . . . 15
B.4 More details about evaluation model robustness . . . 16
B.5 Validity of the W-FFR Verification Pipeline . . . 16
B.6 Stability and Reproducibility of Results . . . 16
B.7 Evaluating the Model–Harness Combination . . . 17

C Details about Evaluation Performance **17**
C.1 Overall trends . . . 17
C.2 Page scale and length . . . 18
C.3 Global layout and section hierarchy . . . 19
C.4 Visual styling . . . 19
C.5 Information density . . . 19
C.6 Open-source vs. Closed-source models. . . . 19
C.7 Anomalous behavior. . . . 19
C.8 Summary . . . 19

D Human Evaluation **19**
D.1 The score of each dimension on human relevance . . . 19
D.2 Inter-annotator agreement analysis . . . 20
D.3 Extended Human Agreement Study . . . 20
D.4 Ablation on the DINO Fusion Weight . . . 20

E Prompts and Model Settings **21**
E.1 Prompts . . . 21
E.2 Model settings . . . 21

F Error Type and Cases Analysis **21**
F.1 Distribution of each dimension error rates for W-VFR . . . 21
F.2 Cost and Reproducibility . . . 23
F.3 Error case analysis for W-VFR . . . 24
F.4 Case analysis for W-FFR . . . 26

A Data Annotation

A.1 More details about W-VFR categories

W-VFR Categories Table 4 provides a taxonomy of page types. We define categories using two complementary criteria: (1) the page's primary functional role in the information ecosystem (e.g., explaining, lookup, offering/transaction, or interaction), and (2) the dominant mechanism by which users obtain content on the page ("content driver"), such as narrative reading, goal-directed lookup, offering-driven choice, or UGC-mediated discussion. This scheme has three main advantages: it is explicit and operational, enabling consistent labeling across domains and websites; it aligns naturally with user intent and page structure, making it easy to map categories to observable features (e.g., navigation depth, modular layout, interaction density, and conversion components) for downstream detection, evaluation, and stratified experimentation; and it is extensible—new page forms can be incorporated by adding categories or drivers without undermining comparability with the existing taxonomy.

A.2 More details about W-FFR categories

W-FFR Categories Table 5 summarizes the category taxonomy of the W-FFR dataset, covering a diverse range of real-world web scenarios. Each category is characterized by its dominant content driver, reflecting the primary user intent and interaction pattern, such as information retrieval, decision making, learning, or user-generated interaction. This categorization highlights the heterogeneity of web pages in terms of content structure and functional objectives, providing a structured foundation for analyzing model performance across different webpage types.

A.3 Distribution of webpage lengths across categories

Figure 9 illustrates the distribution of webpage lengths, measured by the number of viewports, across different webpage categories. Overall, the distribution exhibits clear category-dependent patterns, indicating substantial structural diversity among real-world webpages.

Short webpages ($\leq$5 screens) are predominant in categories such as *Transportation*, *Community*, and *Governmental*. In particular, Transportation pages are heavily concentrated in this range, suggesting that such webpages are often designed for rapid access and concise presentation. Community and Governmental pages also tend to be relatively short, reflecting task-oriented or modular layouts.

Medium-length webpages (6–10 screens) are most common in the *Information* and *Product* categories. Information pages show a pronounced peak in this range, indicating that explanatory or editorial-style content typically spans multiple sections while remaining within a moderate overall length. Product pages exhibit a similar pattern, consistent with layouts that combine descriptions, specifications, and auxiliary content.

In contrast, long and very long webpages ($\geq$16 screens) are predominantly observed in the *Reference* category. Reference pages display a clear long-tail distribution, with a substantial number of instances extending beyond 16 screens and a notable portion exceeding 21 screens. This reflects the encyclopedic and documentation-oriented nature of reference webpages, which often contain dense and hierarchically structured content.

The *Data* category exhibits a more balanced distribution across screen ranges, spanning from short to long webpages. This variability suggests that data-centric webpages differ widely in presentation style, ranging from compact dashboards to extensive analytical or tabular pages.

Overall, these category-specific length distributions highlight the heterogeneity of webpage structures captured in the benchmark and motivate the need for evaluation protocols and model designs that can robustly handle diverse page lengths in long-webpage understanding and generation tasks.

B Details of Evaluation Process

B.1 More details about our validation process on W-VFR task

The prompt for Structural Fidelity Validation is provided in Figure 25-26. We employ `gpt-4o-2024-11-20` as the judge model for automated evaluation. For models that do not support input settings, we assign a default score of 0. The final score is computed as a weighted sum of scores across all evaluation dimensions. Specifically, for the *Visual Style* dimension, the score is computed as a weighted combination of $0.8 \times$ VLM score $+ 0.2 \times$ normalized DINO score, a weighting scheme we empirically validated to achieve the highest agreement with human judgments.

Table 4: The descriptions of W-VFR dataset categories.

#	Category	Content Driver	Comment
1	Informational Articles	Narrative-driven	Users primarily gain information by reading a continuous textual narrative.
2	Reference & Documentation	Lookup-driven	Users search with a clear goal to find and locate specific entries or instructions.
3	Product & Service Pages	Offering-driven	The page aims to present and drive the selection of a specific product or service.
4	Community & Discussion	UGC-driven	The main information comes from user-generated content and interaction structures.
5	Governmental / Institutional	Authority-driven	The page centers on information published by official or authoritative institutions.
6	Data & Resource Aggregation	Listing-driven	The core is a systematic listing of large amounts of data, resources, or items for retrieval.
7	Transportation & Travel Services	Decision + Transaction-driven	Users search, compare, and book travel options under multiple constraints.

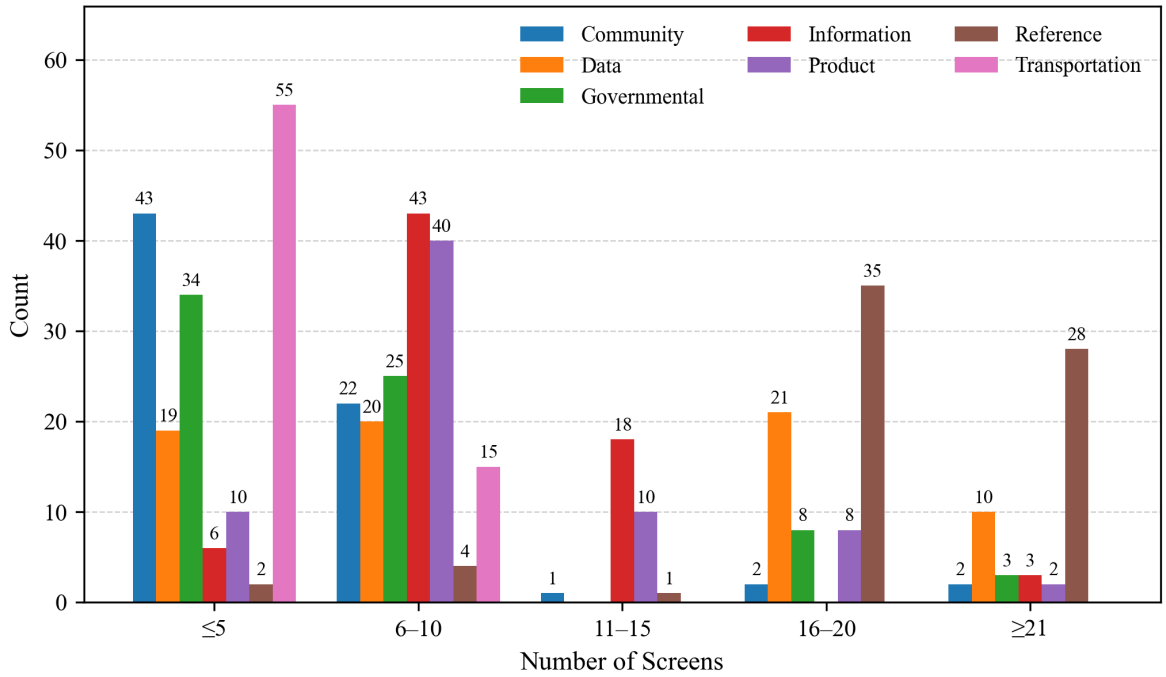


Figure 9: Distribution of Webpage Screenshot Lengths across Categories

Table 5: The descriptions of W-VFR dataset categories.

#	Category	Content Driver	Comment
1	Academic Search	Retrieval-driven	Users search for academic papers, authors, or citations with precise information needs, emphasizing accurate retrieval and relevance ranking.
2	Community & Discussion	UGC-driven	The primary content is generated by users through questions, answers, and discussions, where information emerges from collective interaction.
3	Data Visualization & Analytics	Insight-driven	Pages focus on presenting data through visual or analytical forms to support interpretation, comparison, and exploratory analysis.
4	Developer Docs & Cloud Console	Lookup-driven	Users seek technical documentation, APIs, or configuration instructions to support development and system operation tasks.
5	E-commerce Product Selection	Decision-driven	The page assists users in comparing products, prices, and attributes to facilitate informed purchasing decisions.
6	Education & Course	Learning-driven	Content is structured to support systematic learning, including tutorials, courses, and educational materials.
7	Finance & Market Analysis	Analysis-driven	Pages provide financial data, trends, and analytical insights to support investment, risk assessment, and market understanding.
8	Government & Open Data	Authority-driven	Information is published by official institutions, focusing on policies, regulations, and publicly released datasets.
9	Jobs & Recruitment	Matching-driven	Users search and compare job opportunities or candidates based on structured requirements and qualifications.
10	News & Interactive Media	Narrative-driven	Content delivers timely information through news reporting or interactive storytelling, emphasizing readability and engagement.
11	Real Estate Search	Decision-driven	Pages support searching, filtering, and comparing properties under multiple constraints such as price, location, and amenities.
12	SaaS Pricing	Comparison-driven	The focus is on pricing structures, feature tiers, and cost comparison to support service selection decisions.
13	Travel & Transportation	Planning-driven	Pages support users in searching, comparing, and booking travel options such as flights, hotels, and transit routes, emphasizing multi-constraint filtering and itinerary planning.

B.2 More metric details about W-VFR

Input Support Rate (ISR). ISR measures the fraction of test cases that a model or its serving API accepts for generation, isolating input-side constraints (e.g., image-size limits or resolution-size limits) from downstream generation quality. Let $\mathcal{T}$ denote the full test set and $\mathcal{S}_m \subseteq \mathcal{T}$ the subset that model m accepts and proceeds to generate for. ISR is defined as

$$\mathrm{ISR}(m) \;=\; \frac{|\mathcal{S}_m|}{|\mathcal{T}|}. \tag{1}$$

A low ISR reflects upstream limitations that prevent the model from being applied to certain inputs at all.

HTML Validity Rate (HVR). Given the supported subset $\mathcal{S}_m$, HVR measures the proportion of generations that yield a valid, renderable HTML artifact. An output is counted as valid if it is non-empty and contains a well-formed `<html>` document parseable by a standard HTML parser (with no truncation or non-HTML content). Let $\mathcal{V}_m \subseteq \mathcal{S}_m$ denote the valid outputs from model m. HVR is defined as

$$\mathrm{HVR}(m) \;=\; \frac{|\mathcal{V}_m|}{|\mathcal{S}_m|}. \tag{2}$$

HVR is defined as the model's ability to reliably emit a renderable artifact on supported inputs.

Code Quality (CQ) We assess the engineering quality of model-generated HTML using **Google Lighthouse** (v12), a widely adopted, fully automated, and reproducible web auditing tool that scores a page on a $[0, 100]$ scale across multiple categories.

Evaluation dimensions. We adopt Lighthouse's four default categories:

- **Performance**: page load efficiency measured by core Web Vitals.
- **Accessibility**: powered by the `axe-core` rule set, covering image `alt` attributes, form `<label>` associations, color contrast, the `<html lang>` declaration, ARIA landmark usage, and so on.
- **Best Practices**: HTML document conventions (`<!DOCTYPE>`, `<meta charset>`, `<meta viewport>`), absence of console errors, deprecated APIs, CSP declaration, and image aspect-ratio consistency.
- **SEO**: presence of `<title>`, `<meta description>`, crawler-friendly links, and `robots.txt` compliance.

Audit pipeline. Each HTML file is served via a local static HTTP server bound to `127.0.0.1` and audited through the Lighthouse Node API driving a headless Chromium instance launched by Puppeteer.

Input normalization. To prevent surface artifacts from affecting Lighthouse parsing, we apply a lightweight normalization step on raw outputs: residual Markdown code fences (e.g., leading/trailing ```` ```html ````) are stripped, and missing `<!DOCTYPE html>` declarations are restored for files whose HTML body is otherwise complete. Outputs that already failed at the generation stage (empty files, OOM placeholders, or streaming truncations yielding non-HTML content) are *left untouched* so that they are scored as genuine failures by Lighthouse, contributing to the model's robustness profile.

Final score. For each file, we average the four Lighthouse category scores to obtain a single overall score $S_{m,f} \in [0, 100]$. The code-quality score of model m is the mean of $S_{m,f}$ over the common subset:

$$\textsc{CodeQuality}(m) \;=\; \frac{1}{N} \sum_{f \in \mathcal{F}_{\text{common}}} S_{m,f}. \tag{3}$$

where $N = |\mathcal{F}_{\text{common}}|$.

B.3 More details about W-FFR process

W-FFR Evaluation Pipeline To assess whether a model-generated HTML page supports correct user interactions, we design an automated GUI evaluation pipeline that combines a browser-based environment with an LLM-as-judge agent.

Environment. Each HTML file is served by a dedicated environment process, which launches a headless Chromium browser via Playwright and exposes a local HTTP API. All operations are addressed by numeric element IDs assigned at runtime: a JavaScript traversal of the DOM identifies interactive elements (standard form controls, elements with `role=button`, elements styled with `cursor:pointer`, etc.) and builds an id $\rightarrow$ XPath mapping used to locate and manipulate elements.

Table 6: Model ranking under two independent critics (Gemini vs. GPT-5.5).

Rank	Model	SSR		TSR		PSR	
		Gemini	GPT-5.5	Gemini	GPT-5.5	Gemini	GPT-5.5
1	Gemini-3-Pro-preview	87.91	85.54	59.24	52.86	21.09	19.38
2	Claude-Sonnet-4.5-Thinking	86.46	84.93	55.47	50.30	17.19	17.19
3	GLM-4.6V	64.20	60.51	30.52	29.19	7.03	6.98
4	Qwen3-VL-235B-A22B -Instruct	60.34	55.53	23.27	21.70	6.20	5.07

Tasks. Each HTML file is paired with a JSON task file specifying one or more tasks, each decomposed into a sequence of atomic steps. Each step has a type, a natural-language description of the intended interaction, and an optional value (for input/select steps).

Evaluation Loop. For each task, the page is reset to its initial state before execution begins. Steps are executed sequentially. Each step proceeds in three stages:

1. **Predict.** A screenshot and the serialized DOM tree (with numeric IDs) are passed to an LLM. The model is prompted to identify the target element and output a structured action command in `\boxed{}` format (e.g., `\boxed{click[42]}` or `\boxed{enter[7][hello]}`). The command is parsed via regex to extract the action type, element ID, and value.

2. **Execute.** The parsed action is dispatched to the browser environment via the corresponding HTTP endpoint. The environment resolves the element by XPath and performs the operation.

3. **Verify.** Screenshots captured before and after execution are passed to a second LLM call. The model compares the two images and judges whether the intended change occurred (e.g., a field was filled, a dropdown changed, a new element appeared), outputting `\boxed{success}` or `\boxed{fail}`.

The prompts used in this process are provided in Appendix E.1.

B.4 More details about evaluation model robustness

We re-ran the same critic-based adjudication with GPT-5.5 replacing Gemini. As shown in Table 6, the model ranking is unchanged under the independent critic; Gemini-3-Pro-preview retains rank-1 in TSR and PSR, with the margin over the second-ranked model narrowing slightly. These consistent results confirm that our evaluation method is robust to the choice of the critic model.

B.5 Validity of the W-FFR Verification Pipeline

We ran the actor agent under the W-FFR verification pipeline on 20 manually curated, logically-correct reference pages (using GPT-5.5 in place of Gemini-3-Pro, which was unavailable at rebuttal time). Since these reference pages are verified to be correct by construction, a valid verification pipeline should confirm the vast majority of their steps, tasks, and pages as successful. The results, reported in Table 7, show that our verification pipeline remains effective and reliable: it recovers the expected successful behavior at high rates across all three granularities.

Table 7: Validity of the W-FFR verification pipeline on 20 logically-correct reference pages.

Metric	# Success / Total	Rate
SSR	389 / 424	91.75%
TSR	69 / 77	89.61%
PSR	17 / 20	85.00%

B.6 Stability and Reproducibility of Results

Run-to-run variance. To test whether our results are noise-sensitive, we re-ran two models under identical settings: Claude-Sonnet-5 on the 50-sample subset, and GLM-4.1 on the full benchmark (tiles). As shown in Tables 8 and 9, scores are highly consistent across runs for both models, indicating that our reported results are not noise-sensitive.

Table 8: Run-to-run variance for Claude-Sonnet-5 on the 50-sample subset.

Dimension	Run 1	Run 2	Δ
page_scale_ and_length	6.88	6.78	+0.10
global_layout	8.28	8.28	0.00
section_hierarchy	7.54	7.44	+0.10
visual_styling	6.28	6.42	−0.14
information_density	6.46	6.40	+0.06
Overall_all	7.088	7.064	+0.024

Table 9: Run-to-run variance for GLM-4.1 on the full benchmark.

Dimension	Run 1	Run 2	Δ
page_scale_ and_length	0.56	0.56	0.00
global_layout	0.93	0.92	−0.01
section_hierarchy	0.65	0.70	+0.05
visual_styling	1.97	2.06	+0.09
information_density	0.58	0.65	+0.07
Overall_all	0.94	0.98	+0.04

Extended leaderboard. We additionally evaluated Claude-Sonnet-5 on the full benchmark and add it to the leaderboard (judge model: GPT-5.5). The updated ranking is reported in Table 10.

Table 10: Updated leaderboard on the full benchmark (judge model: GPT-5.5).

Rank	Model	SSR	TSR	PSR
1	Claude-Sonnet-5	86.89	63.22	28.13
2	Gemini-3-Pro-preview	85.54	52.86	19.38
3	Claude-Sonnet-4.5-Thinking	84.93	50.30	17.19
4	GLM-4.6V	60.51	29.19	6.98
5	Qwen3-VL-235B-A22B-Instruct	55.53	21.70	5.07

B.7 Evaluating the Model–Harness Combination

Setup. We run the same task through the Claude Code (CC) harness with the Claude-Sonnet-5 model on 50 samples, using the identical prompt as the single-shot baseline and restricting the harness to Read and Write. This deliberately isolates the axis the reviewer highlights, namely harness-level long-context orchestration, from an orthogonal axis (visual self-refinement via code execution and rendering) that the single-shot baseline also lacks and that warrants a separate study. Any observed gain is thus attributable specifically to the harness's context and turn management.

Table 11: Single-shot baseline (Ours) vs. the Claude Code harness (CC) on 50 samples.

Dimension	Ours	CC	Δ
page_scale_ and_length	6.837	6.837	0
global_layout	8.265	8.204	−0.061
section_ hierarchy	7.510	7.551	+0.041
visual_styling	6.245	6.408	+0.163
information_ density	6.408	6.592	+0.184
Overall	7.053	7.118	+0.065

As shown in Table 11, the CC harness yields a modest overall gain (+0.93%), concentrated on detail-level dimensions (visual styling, information density), while structural dimensions (page scale and length, global layout, section hierarchy) remain essentially unchanged.

C Details about Evaluation Performance

Tables 12 and 13 summarize model performance across different evaluation dimensions under the single-image and multi-image input settings, respectively. Overall, the results show that the multi-image setting consistently improves model performance across most dimensions and models, particularly for long and structurally complex webpages.

C.1 Overall trends

From the Overall Score, most models benefit substantially from the multi-image setting. By segmenting long webpage screenshots into multiple images, models are able to perceive finer-grained visual details and reduce information loss caused by resolution compression in single-image inputs. This improvement is especially pronounced for strong closed-source models, such as the Gemini and Claude families, whose overall scores increase consistently across settings.

Table 12: The evaluation result of each dimension under single-image input setting

Model	Page Scale and Length	Global Layout	Section Hierarchy	Visual Styling	Information Density	Overall Score
Open-source VLM						
Kimi-VL-A3B-Thinking	2.06	3.40	2.67	3.86	2.34	2.87
Qwen3-VL-8B-Instruct	3.32	4.02	3.36	3.75	2.91	3.47
InternVL3-78B	2.41	4.25	3.30	4.22	2.82	3.40
GLM-4.1V-9B-Thinking -Flash	5.04	5.62	4.67	4.92	4.11	4.87
GLM-4.6V	5.54	6.58	5.63	5.59	4.94	5.66
Qwen3-VL-235B-A22B -Instruct	5.29	5.75	5.08	4.95	4.44	5.10
Closed-source VLM						
Claude-Sonnet-4-5-Thinking	3.85	4.39	3.91	3.49	3.39	3.81
Claude-Opus-4-5-20251101	3.95	4.38	3.92	3.52	3.43	3.84
GPT-5.2	7.02	7.33	6.37	6.00	5.77	6.50
GPT-4o	1.92	4.16	3.01	4.60	2.45	3.23
Gemini-3-Pro-preview	6.65	7.87	6.95	6.38	5.99	6.77
Gemini-3-Flash-preview	6.28	7.43	6.44	6.05	5.58	6.36
Doubao-Seed-1-6	2.97	4.60	3.62	4.02	3.10	3.66

Table 13: The evaluation result of each dimension under multi-image input setting

Model	Page Scale and Length	Global Layout	Section Hierarchy	Visual Styling	Information Density	Overall Score
Open-source VLM						
Kimi-VL-A3B-Thinking	2.45	3.71	2.87	4.03	2.52	3.12
Qwen3-VL-8B-Instruct	3.69	4.56	3.89	4.24	3.35	3.95
InternVL3-78B	2.62	4.51	3.51	4.40	3.05	3.62
GLM-4.1V-9B-Thinking -Flash	0.56	0.93	0.65	1.97	0.58	0.94
GLM-4.6V	5.65	7.07	6.03	5.89	5.18	5.96
Qwen3-VL-235B-A22B-Instruct	5.50	6.90	5.93	5.79	5.13	5.85
Closed-source VLM						
Claude-Sonnet-4-5-Thinking	6.56	7.83	6.90	6.41	5.97	6.74
Claude-Opus-4-5-20251101	6.14	7.59	6.58	6.20	5.63	6.43
GPT-5.2	5.33	7.46	6.18	6.45	5.28	6.14
GPT-4o	2.02	4.58	3.30	4.93	2.79	3.53
Gemini-3-Pro-preview	7.12	8.26	7.48	6.72	6.50	7.22
Gemini-3-Flash-preview	6.15	7.68	6.58	6.53	5.71	6.53
Doubao-Seed-1-6	3.58	6.05	4.69	5.48	4.03	4.77

C.2 Page scale and length

Page Scale and Length results show different trends across models. Under roughly comparable image processing conditions, the GPT-5.2 model actually experiences a performance drop under multi-image input settings, whereas Gemini-3-Pro shows some improvement. In contrast, the Claude model ex-

hibits a significant performance increase, as the number of successfully processed images is greatly enhanced under the multi-image input setting.

C.3 Global layout and section hierarchy

Performance on Global Layout and Section Hierarchy also improves noticeably in the multi-image setting in most cases. These gains indicate that access to multiple localized views helps models better infer high-level structural organization, including module ordering and hierarchical relationships between sections. The effect is particularly evident for the Claude and Gemini models, which show substantial improvements on these structure-oriented dimensions.

C.4 Visual styling

In contrast, improvements on Visual Styling are still evident. Multi-image inputs provide additional visual details, which help models better capture fine-grained stylistic attributes—such as font choices, color schemes, and spacing—even though these aspects remain challenging to infer from screenshots alone. Consequently, this dimension also shows positive performance gains.

C.5 Information density

Information Density lies between structural and stylistic dimensions in terms of evaluation complexity. Most models achieve modest but consistent improvements in the multi-image setting, reflecting the combined influence of enhanced structural perception and the inherent ambiguity of density judgments from visual inputs.

C.6 Open-source vs. Closed-source models.

Almost all Closed-source models consistently outperform open-source models under both settings, and their advantage becomes more pronounced in the multi-image scenario. Notably, Gemini-3-Pro-preview achieves the highest overall performance and demonstrates strong, balanced improvements across all dimensions.

C.7 Anomalous behavior.

An exception is observed for GLM-4.1V-9B-Thinking-Flash, which exhibits a sharp performance degradation under the multi-image setting. Manual inspection reveals that, when presented with multiple images, this model tends to misinterpret the task as site-level webpage generation rather than long-page reconstruction, resulting in severely degraded outputs. This highlights that effective utilization of multi-image inputs requires robust cross-image integration and task alignment, which are not uniformly supported across models.

C.8 Summary

In summary, these results demonstrate that multi-image input is a critical design choice for evaluating and generating long webpages, offering substantial benefits for scale- and structure-related dimensions. While stylistic fidelity remains challenging due to limited design-level information, the overall performance gains underscore the importance of segmented visual inputs for long-context webpage understanding and generation.

D Human Evaluation

D.1 The score of each dimension on human relevance

Table 14 reports the correlation between the automatic evaluation scores and human judgments averaged across annotators. Overall, the automatic scores exhibit a strong correlation with human assessments, achieving a Pearson correlation coefficient of 0.8335, which indicates a high degree of alignment between the proposed evaluation protocol and human perception.

Among the individual dimensions, Page Scale and Length show a very strong correlation (0.9150), suggesting that both humans and models consistently assess page completeness and overall length. Global Layout (0.7706) and Chapter Hierarchy (0.7393) also demonstrate strong correlations, reflecting reliable agreement on the structural organization and hierarchical arrangement of webpage content. Information Density achieves a moderately high correlation (0.6807), indicating that the automatic metric can reasonably capture the perceived amount and distribution of information on long webpages. Visual Style achieves a correlation of 0.6035, reflecting a moderately strong agreement between automatic scores and human judgments. This is expected given the screenshot-only input setting adopted in our experiments, as fine-grained stylistic attributes—such as font families, color schemes, spacing, and design system constraints—are not explicitly accessible without the original HTML or CSS sources, making stylistic fidelity inherently more challenging to assess from rendered images alone. Nevertheless, the correlation remains meaningful, suggesting that the auto-

matic metric can capture overall visual style consistency to a reasonable degree.

Overall, these results indicate that the proposed automatic evaluation aligns well with human judgments, while the reduced correlation on visual style reflects the intrinsic subjectivity of aesthetic assessment and the limited availability of design-level information in the current task setting.

Table 14: Correlation between person_average and AI Models

Metric	Coefficient	p-value
Overall Score	0.8335	<0.0001
Page Scale and Length	0.9150	<0.0001
Global Layout	0.7706	<0.0001
Chapter Hierarchy	0.7393	<0.0001
Visual Style	0.6035	<0.0001
Information Density	0.6807	<0.0001

D.2 Inter-annotator agreement analysis

Figure 10 presents pairwise comparisons of total scores assigned by three human annotators, as well as comparisons between each annotator and the averaged score across annotators. Each point corresponds to one webpage instance, the black diagonal line indicates perfect agreement ($y = x$), and the red dashed line denotes the fitted linear regression.

Overall, the results demonstrate a high level of inter-annotator consistency. Pairwise correlations between individual annotators are strong, with Pearson correlation coefficients of $r = 0.793$ (Annotator 1 vs. Annotator 2), $r = 0.779$ (Annotator 1 vs. Annotator 3), and $r = 0.849$ (Annotator 2 vs. Annotator 3), all with statistically significant p-values ($p < 10^{-20}$). The corresponding RMSE and MAE values indicate moderate absolute disagreement, reflecting the inherent subjectivity of fine-grained visual fidelity assessment.

Comparisons between individual annotators and the averaged score further reveal substantially higher agreement. Annotator 1, Annotator 2, and Annotator 3 achieve Pearson correlations of $r = 0.920$, $r = 0.945$, and $r = 0.935$ with the average score, respectively. In addition, both RMSE and MAE are markedly reduced in these comparisons, suggesting that score averaging effectively suppresses individual annotation noise and yields a more stable reference signal.

The regression lines across all plots exhibit slopes slightly below unity and small intercept offsets, indicating mild systematic differences in score calibration among annotators. Nevertheless, the bias values remain close to zero, suggesting no severe over- or under-estimation tendencies for any individual annotator.

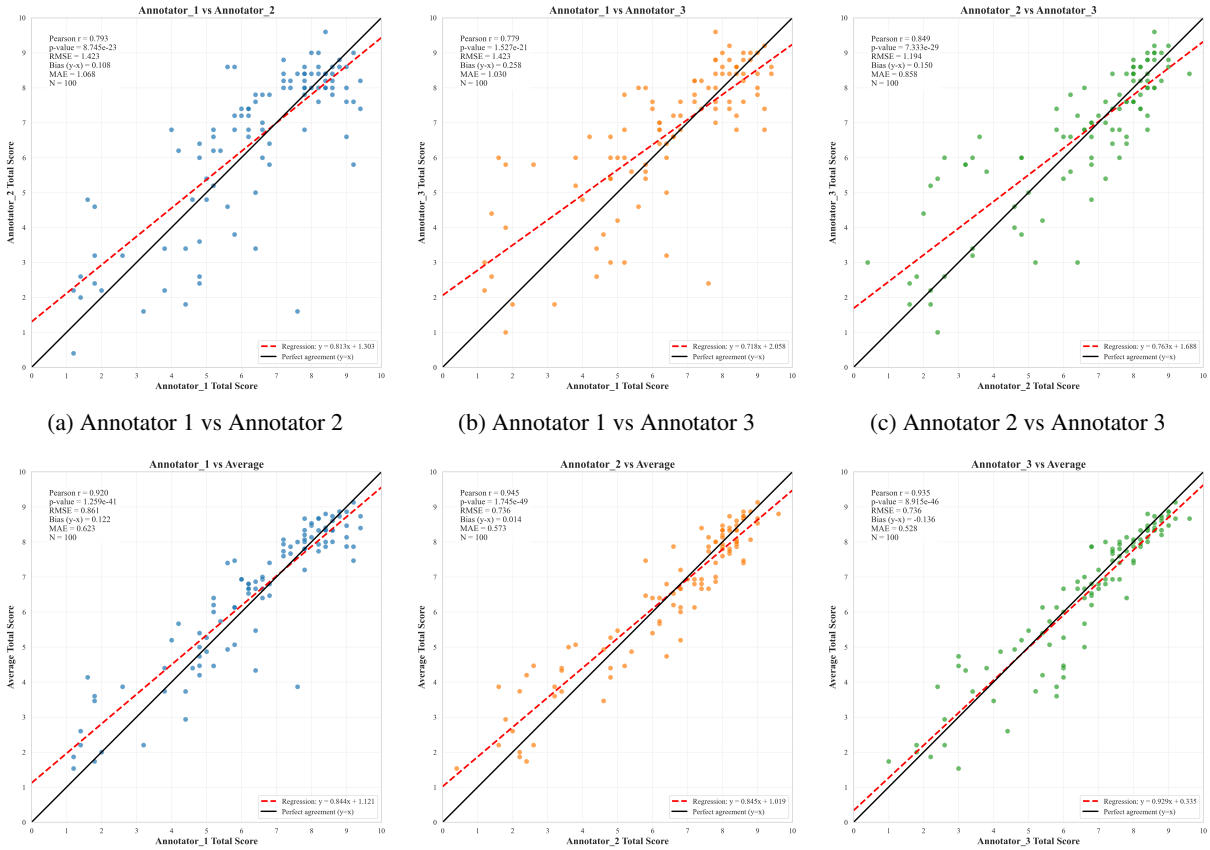
Taken together, these results confirm that the proposed evaluation protocol yields reliable and reproducible human judgments. The strong inter-annotator agreement and the improved consistency obtained through averaging support the use of the averaged human score as the ground-truth reference in subsequent correlation analyses with model predictions.

D.3 Extended Human Agreement Study

To rule out the possibility that our conclusions are sensitive to the choice or size of the annotated set, we further conduct an extended human study with 3 independent annotators over 150 samples. Table 15 reports inter-annotator agreement (Krippendorff's α and ICC) together with the correlation between W-VFR and the human mean. Reliability is high across the board: Krippendorff's $\alpha = 0.81$ overall (5 of 6 dimensions ≥ 0.667), and ICC$(A, k) \geq 0.82$ on every dimension, above the 0.75 "excellent" threshold. On the most challenging dimension, *visual_styling*, our DINO+VLM fusion lifts the correlation from 0.610 to 0.634. We further extend the same study to W-FFR on 150 annotated samples and observe comparably high agreement, ICC$(A, k{=}3) = 0.903$.

D.4 Ablation on the DINO Fusion Weight

Evaluating visual style for long, scroll-rich webpages is intrinsically difficult. Our W-VFR is designed to fuse complementary signals; below we provide the ablation requested by the reviewer. As shown in Table 16, the fused score peaks at $w_{\text{dino}}{=}0.2$ (Pearson $= 0.634$, $+4.0\%$ over VLM-only), outperforming both the VLM-only ($w{=}0$) and DINO-only ($w{=}1$) endpoints, confirming that fusing the two signals aligns best with human ratings. Moreover, VLM and DINO are complementary rather than redundant (cross-correlation $= 0.37$): since DINO is a self-supervised encoder unaware of our generators, it provides an unbiased, independent reference at the model-level aggregation, complementing the VLM signal.



(a) Annotator 1 vs Annotator 2 (b) Annotator 1 vs Annotator 3 (c) Annotator 2 vs Annotator 3

(d) Annotator 1 vs Avg Score (e) Annotator 2 vs Avg Score (f) Annotator 3 vs Avg Score

Figure 10: Inter-Annotator Agreement Analysis.

E Prompts and Model Settings

E.1 Prompts

Task Generation Prompts We present the prompts used for long webpage generation and verification, as shown in Figures 11–13.

W-VFR Generation Prompts Generation prompts for W-VFR under single-image input setting and multi-image input setting are shown in Figures 14–15 and Figures 16–17, respectively.

W-VFR Evaluation Prompts Evaluation prompts for W-VFR are shown in Figures 18–20 .

W-FFR Generation Prompts Generation prompts for W-FFR under single-image input setting and multi-image input setting are shown in Figures 21–22 and Figures 23–24, respectively.

W-FFR Evaluation Prompts Evaluation prompts for W-VFR are shown in Figures 25–26 .

E.2 Model settings

To ensure fair comparison and reproducibility, we adopt deterministic decoding whenever supported by the model APIs. Specifically, for models that expose sampling parameters, we set the temperature to 0 and the top-p value to 1, which minimizes randomness in the generated code. For models that do not support explicit sampling control (e.g., GPT-5.2), we use the default decoding behavior provided by the API.

Regarding generation length, long webpage replication requires producing a substantial amount of HTML, CSS, and JavaScript code, resulting in a high token demand. Therefore, instead of using a fixed token budget, we configure the `max_completion_tokens` parameter to the maximum value supported by each model. This design choice avoids premature truncation and allows each model to generate outputs under its own architectural constraints. The specific `max_completion_tokens` settings for different models are summarized in Table 17.

F Error Type and Cases Analysis

F.1 Distribution of each dimension error rates for W-VFR

Tables 18 and 20 report failure rates of various open-source and closed-source VLMs under single-

Prompt for HTML webpage Task generation for W-FFR. (Part 1)

You are an expert in designing Goal-Oriented Web Task Benchmarks for long webpages with complex interactions. Your focus is on modeling realistic, end-to-end user goals rather than atomic navigation actions.
Inputs:
- A full-length webpage screenshot (the page may require vertical scrolling to be fully visible).
- (Optional) A virtual account credential, which may only be used if login-related UI elements are explicitly visible in the screenshot:

```
{
  "benchmark_account": {
    "username": "test_user_001",
    "email": "test_user_001@example.com",
    "password": "P@ssw0rd!001"
  }
}
```

Critical clarifications (must be followed):
- The tasks you design are not atomic UI navigation steps, but complex, real-world user goals.
- Each task must include:
- A clearly defined final goal
- Multi-step interactions (e.g., filtering, configuration, comparison, decision-making, submission)
- Interactions spanning multiple page regions or modules
- A final goal with explicit and verifiable success signals
Examples (for style reference only; do not replicate):
- Selecting the cheapest direct flight among all available options
- Configuring a computer with 16GB RAM and 2TB storage and adding it to the cart
- Finding the hotel with the highest rating and lowest price in the current city
- Locating and confirming the default value of a key parameter in documentation
Objective: Based on the screenshot, infer a plausible interactive webpage and design 3–4 complex interaction tasks.
Task design requirements:
- Each task represents a realistic user goal rather than a single-button action
- Each task contains at least three semantically meaningful interaction steps
- Tasks are completable within a long webpage (scrolling, filtering, tab switching allowed)
- Filtering, comparison, or decision-making tasks must include explicit numeric or discrete outcomes
- Key success values must be derived from task execution through calculation or extraction
- All tasks must be automatically verifiable via observable success signals defined in the schema
Authentication task rules:
- Authentication-related tasks are allowed only if the screenshot explicitly shows one of the following:
- Login / Log in / Sign in
- Register / Sign up
- Logout / Log out
- User avatar / Account menu / Profile center
- If none of the above are visible:
- Do not design any login, registration, or logout tasks
- All tasks must be executable in a logged-out state
Output requirements:
- Output only one JSON object
- Do not include Markdown

Figure 11: This prompt is designed for HTML webpage Task generation for W-FFR (Part 1).

Table 15: Human agreement study for W-VFR (3 annotators, 150 samples). Agreement is measured by Krippendorff's α and ICC(A, k); the last column reports the Pearson correlation between W-VFR and the human mean.

Dimension	Krippendorff's α [95% CI]	ICC(A, k)	W-VFR vs. human Pearson [95% CI]
Overall_all	0.81 [0.76, 0.86]	0.928	0.835 [0.796, 0.869]
page_scale_ and_length	0.79 [0.74, 0.84]	0.921	0.915 [0.884, 0.939]
global_layout	0.68 [0.59, 0.76]	0.866	0.790 [0.725, 0.838]
section_hierarchy	0.69 [0.59, 0.76]	0.868	0.742 [0.680, 0.794]
visual_styling	0.61 [0.50, 0.68]	0.825	0.610 [0.483, 0.715]†
information_density	0.68 [0.61, 0.74]	0.869	0.680 [0.607, 0.743]

† Pure VLM; adding DINO (VLM+DINO fusion) raises the correlation to 0.634.

Table 16: Pearson correlation with human ratings across the DINO fusion weight w_{dino} (150 samples, 5 generation models).

w_{dino}	Pearson
0.0	0.6096
0.1	0.6332
0.2	**0.6340**
0.3	0.6155
0.4	0.5849
0.5	0.5493
0.6	0.5134
0.7	0.4796
0.8	0.4489
0.9	0.4216
1.0	0.3975

Table 17: Maximum completion token limits used for different models.

Model	Max Completion Tokens
GPT-5.2	64000
GPT-4o	16384
Claude-Sonnet-4-5-Thinking	64000
Claude-Opus-4-5-20251101	64000
Gemini-3-Pro-preview	64000
Gemini-3-Flash-preview	64000
Doubao-Seed-1-6	32000
GLM-4.6V	32000
GLM-4.1V-9B-Thinking-Flash	32000
Qwen3-VL-235B-A22B-Instruct	64000
Qwen3-VL-8B-Instruct	64000
InternVL3-78B	32000
Kimi-VL-A3B-Thinking	32000

image and multi-image input settings, where higher values indicate worse performance. Overall, providing multiple images generally reduces failure rates across most models and dimensions, demonstrating that additional visual context helps models generate more correct webpage representations. The improvement is particularly evident for **Page Scale and Length**, **Global Layout**, and **Section Hierarchy**, where multi-image inputs allow better capture of the overall page structure. Open-source models show moderate gains on these structural dimensions, while closed-source models, such as Claude and Gemini-3-Pro, achieve near-zero failure rates under multi-image settings, reflecting strong structural robustness.

For **Visual Styling** and **Information Density**, failure rates also decrease with multi-image inputs, but improvements are more moderate, indicating the inherent difficulty of accurately capturing fine-grained style and dense content from screenshots alone. Some models exhibit unique trends; for instance, GPT-5.2's **Page Scale and Length** failure rate increases under multi-image inputs, suggesting sensitivity in integrating multiple images for certain structural aspects. Overall, multi-image inputs consistently enhance performance across nearly all dimensions, with the largest gains in structural correctness and smaller but positive gains in stylistic and content density attributes.

F.2 Cost and Reproducibility

We report the estimated token consumption and request counts of our evaluation pipeline in Table 19.

Table 18: Performance analysis of correctness under different dimensions under single-image input setting

Model	Page Scale and Length	Global Layout	Section Hierarchy	Visual Styling	Information Density
Open-source VLM					
Kimi-VL-A3B-Thinking	92.65	73.06	86.94	91.84	92.24
Qwen3-VL-8B-Instruct	61.84	48.57	58.98	64.49	63.47
InternVL3-78B	88.78	52.65	75.71	79.80	84.90
GLM-4.1V-9B-Thinking-Flash	51.63	28.16	46.53	57.55	55.51
GLM-4.6V	36.94	13.47	27.14	31.02	36.94
Qwen3-VL-235B-A22B-Instruct	42.65	26.53	33.27	42.04	44.08
Closed-source VLM					
Claude-Sonnet-Thinking	46.12	43.88	44.69	47.55	46.94
Claude-Opus-4-5-20251101	47.55	44.08	44.49	47.35	48.57
GPT-5.2	18.16	5.92	13.67	14.49	21.43
GPT-4o	97.55	68.37	91.84	73.47	97.14
Gemini-3-Pro-preview	11.63	1.84	4.69	8.98	13.88
Gemini-3-Flash-preview	18.78	3.67	13.27	15.31	21.84
Doubao-Seed-1-6	80.82	30.82	55.92	58.98	76.12

Table 19: Cost and reproducibility: estimated token consumption and request counts per stage. Token counts are in millions (M).

Stage	Prompt	Completion	Requests
W-FFR-generate	0.57	2.44	129
W-FFR-judge	65	0.93	4,000
W-VFR-generate	3.56	9.52	490
W-VFR-judge	2.02	0.22	490

F.3 Error case analysis for W-VFR

We analyze nine representative failure cases observed in our benchmark, which can be broadly categorized into four aspects: (1) HTML generation stability, (2) failures in the rendering stage, (3) scale and visual consistency, and (4) structural and semantic errors.

Case 1 As shown in Figure 27, this example illustrates a failure case where the generated HTML is incomplete due to token length constraints. In this case, the model exhausts the token budget during HTML generation, leading to a truncated output. We observe several common patterns contributing to this issue, including excessively long style sections, repetitive generation of redundant content, and the emission of meaningless URLs followed by unbounded character sequences. These behaviors prevent the model from properly terminating the HTML structure, resulting in an incomplete document that cannot be reliably rendered or evaluated in subsequent stages.

Case 2 As shown in Figure 28, this example presents a failure case in the HTML-to-image rendering stage of our benchmark pipeline. Although the generated HTML can be rendered by a standard web browser, the conversion from HTML to image fails. Further inspection shows that the model repeatedly outputs redundant text until the token budget is exhausted, resulting in an incomplete HTML document. On the rendered webpage, this repetitive content manifests as an abnormally large page width, which eventually causes the HTML-to-image engine to fail and produce an invalid image.

Case 3 As shown in Figure 29, this example illustrates a failure case where a large blank region appears at the bottom of the generated webpage. Compared with the original image, the generated result contains an extended empty area with no meaningful content. We find that this issue may stem from the prompt configuration, in which the original image size is provided as a reference. During HTML generation, the model attempts to align with the given page height and fills the remaining

Table 20: Performance analysis of correctness under different dimensions under multi-image input setting

Model	Page Scale and Length	Global Layout	Section Hierarchy	Visual Styling	Information Density
Open-source VLM					
Kimi-VL-A3B-Thinking	87.14	70.61	85.92	86.94	90.82
Qwen3-VL-8B-Instruct	59.59	42.24	47.96	57.14	58.37
InternVL3-78B	89.18	45.51	72.86	78.37	86.53
GLM-4.1V-9B-Thinking-Flash	95.10	88.78	91.84	93.27	93.47
GLM-4.6V	25.51	5.31	12.86	21.22	25.31
Qwen3-VL-235B-A22B-Instruct	31.02	9.39	18.57	21.02	30.61
Closed-source VLM					
Claude-Sonnet-Thinking	12.45	0.82	4.08	4.69	10.61
Claude-Opus-4-5-20251101	15.31	0.41	2.24	10.61	14.69
GPT-5.2	41.22	1.22	8.37	5.71	28.98
GPT-4o	98.37	52.45	91.22	60.82	97.14
Gemini-3-Pro-preview	5.31	0.00	1.02	2.45	4.90
Gemini-3-Flash-preview	25.51	1.22	15.31	4.69	25.71
Doubao-Seed-1-6	78.98	8.78	47.96	41.02	71.22

space with blank regions, even though padding the layout with empty space or repetitive content is explicitly discouraged and considered invalid in our benchmark setting.

Case 4 As shown in Figure 30, this example illustrates a failure case where the generated webpage is significantly longer than the original one. The excessive page length is primarily caused by repeated content blocks or redundant character sequences, resulting in unintended page expansion during HTML generation.

Case 5 As shown in Figure 31, this example illustrates a failure case where the generated webpage is complete in structure but shorter than the original one due to missing content . Compared with the original image, the generated result omits part of the page content, leading to a reduced overall page length. This issue may be attributed to the excessive length of the original webpage, which poses challenges for the model to capture and preserve all intermediate information during HTML generation.

Case 6 As shown in Figure 32, this example illustrates a failure case where the generated webpage exhibits noticeable visual style inconsistencies compared with the original page. While the overall page structure and content coverage are largely preserved, discrepancies can be observed in visual attributes such as color scheme, typography, spacing, and layout density across different regions of the page. These inconsistencies indicate that the model fails to maintain a coherent global visual style during HTML generation, leading to mismatched visual appearance between different sections.

Case 7 As shown in Figure 33, this example illustrates a failure case where the hierarchical structure of the generated webpage deviates from that of the original page. Although most content elements are present, their organizational hierarchy is incorrectly constructed, leading to misplaced or flattened sections in the generated result. In particular, items that should be nested under specific categories are rendered at the same level, resulting in a distorted structural layout.

Case 8 As shown in Figure 34, this example illustrates a failure case where the information density of the generated webpage is diluted compared with the original page. While the original image maintains compact content organization with preserved blank regions, the generated result spreads information elements across the page, leading to reduced content concentration and degraded readability.

Case 9 As shown in Figure 35, this example illustrates a failure case where the global layout of the generated webpage deviates from the original design. The original page follows a three-column structure consisting of left, center, and right regions,

whereas in the generated result, the right-side navigation panel is missing. This omission alters the overall page balance and structural organization, despite the main content being preserved.

F.4 Case analysis for W-FFR

We further provide three illustrative examples for the W-FFR task. Figure 36 presents a case that is regarded as successful, while Figure 37,38 illustrate two failure cases.

Prompt for HTML webpage Task generation for W-FFR. (Part 2)

- Do not include explanatory text. The output must strictly conform to the schema below
Schema (complex goal-oriented task version):

```
{
  "page_summary": {
    "site_or_product": "...",
    "page_type": "e-commerce | ticketing | travel | documentation
    | SaaS | news | comparison | other",
    "primary_modules": [
      "Clearly visible main modules in the screenshot"
    ],
    "notable_interactives": [
      "Key interactive controls enabling complex goals"
    ],
    "auth_visible": true | false,
    "auth_evidence": [
      "Visible evidence proving authentication presence, if any"
    ]
  },
  "benchmark_account": {
    "username": "test_user_001",
    "email": "test_user_001@example.com",
    "password": "P@ssw0rd!001"
  },
"tasks": [
    {
      "task_id": "T1",
      "task_type": "purchase | configuration | comparison | selection |
      information_verification |
      auth_dependent | mixed",
      "task_name": "Concise summary of the task objective",
      "user_goal": "One-sentence description of the real user's final goal",
      "instruction": "Goal-oriented instruction to the agent (in English)",
      "constraints": [
        "Mandatory business or logical constraints"
      ],
      "preconditions": [
        "e.g., logged-out state, filter panel expanded, city selected"
      ],
      "steps": [
        {
          "action": "click | type | select | scroll | compare",
          "target_description": "UI target described by location and
          semantic meaning",
          "value": "Specific input or selection value, or null"
        }
      ],
```

Figure 12: This prompt is designed for HTML webpage Task generation for W-FFR (Part 2).

Prompt for HTML webpage Task generation for W-FFR. (Part 3)

```
      "success_criteria": [
         "At least two observable, automatically verifiable success
         signals with explicit quantitative or discrete values"
       ],

         "verification_hints": {
         "text_assertions": [
           "Key texts or values expected after successful completion"
         ],
         "layout_assertions": [
           "Observable layout or state changes"
         ],
         "selector_strategy": [
           "Prefer semantic regions for configuration or pricing",
           "Fallback to aria/role",
           "Lastly, structured XPath"
         ],
         "suggested_selectors": [
           {
             "purpose": "Core element for final verification",
             "css": null,
             "xpath": null,
             "aria": null
           }
         ]
       },
       "difficulty": "medium | hard | very_hard",
       "coverage_tags": [
         "filtering",
         "decision_making",
         "multi_step",
         "long_page_navigation"
       ]
     }
   ]
}
```

Figure 13: This prompt is designed for HTML webpage Task generation for W-FFR (Part 3).

Prompt for HTML webpage generation under the single-image input setting for W-VFR. (Part 1)

You are a Web Visual Replication Specialist. Your task is: given a full long webpage screenshot, generate a standalone HTML file that can be opened directly in a browser, reproducing the screenshot's visual appearance, page structure, and page scale as accurately as possible. This is a Visual Fidelity task. Real business interaction logic is not evaluated at this stage.

Input:

- One full long webpage screenshot (a vertical long image containing the entire page)
- The real pixel dimensions of the screenshot: - Target page width (pixels): {TARGET_PAGE_WIDTH_PX}
- Target page height (pixels): {TARGET_PAGE_HEIGHT_PX}

Output Requirements (must strictly follow)

- You must output one and only one complete HTML file content
- The output must start with $<!doctypehtml>$
- The output must end with $</html>$
- Do not output any explanatory text, comments, Markdown, or code fences - The output must be directly savable as index.html and open in a desktop browser

---

Implementation Constraints

- Use native HTML / CSS / JavaScript only
- All styles must be defined inline within a <style> block
- Do not use any third-party libraries, frameworks, CDNs, or build tools
- Do not reference any external images, fonts, or resources
- For images / avatars / logos in the screenshot:
- Recreate them using SVG or CSS placeholder blocks
- Keep their visual size and position
- Keep semantics as close as possible to the original content

---

Core Visual Replication Requirements

---

1) Page Structure and Layout
- Strictly replicate the information architecture and module order shown in the screenshot, such as:
- Top navigation / secondary navigation
- Sidebar (if present)
- Main content area
- Card lists / tables / section modules
- Footer (if present)
- The relative positions, width proportions, and layering relationships of modules must match the screenshot

2) Style and Visual Elements
- Reproduce the overall color scheme (background, primary, secondary colors)

Figure 14: This prompt is designed for the single-image input setting, where a single full-page webpage screenshot is provided as input and the model is required to generate a complete HTML webpage that visually replicates the input screenshot.

Prompt for HTML webpage generation under the single-image input setting for W-VFR. (Part 2)

- Reproduce font hierarchy (title/body/note sizing and weight relationships)
- Reproduce spacing system (padding / margin)
- Reproduce borders, corner radius, shadows, and divider line styles
- The visual density of cards, lists, and tables must match the screenshot

3) Page Scale and Dimension Alignment (Critical)

- The generated page must cover all content visible in the screenshot
- The total scroll height should align as closely as possible to {TARGET_PAGE_HEIGHT_PX} pixels
- Do not inflate height by meaningless repetition
- If extension beyond the screenshot is necessary to keep the structure complete:
- Only extend using module types and visual styles already present in the screenshot
- Extensions must be reasonable and restrained, and must not break overall proportions
- The overall layout proportions must match the screenshot corresponding to the target width {TARGET_PAGE_WIDTH_PX} pixels

---

Content and Placeholder Rules

- Replicate text content and typographic layout as faithfully as possible - Preserve the original page language whenever possible
- If text is illegible, use reasonable placeholder text while keeping the same line count, paragraph structure, and text width
- Lists / tables / cards may use mock data, but quantity and density must match the screenshot

---

Interaction Constraints (Intentionally Weakened)

- No real interaction logic is required
- Hover / focus visual states are allowed
- All buttons or controls may be empty or placeholder on click

---

Now Begin

Based on the provided long webpage screenshot and the given target page dimensions, output the complete HTML code for visual replication evaluation.

Figure 15: This prompt is designed for the single-image input setting, where a single full-page webpage screenshot is provided as input and the model is required to generate a complete HTML webpage that visually replicates the input screenshot.

Prompt for HTML webpage generation under the multi-image input setting for W-VFR. (Part 1)

You are a Web Visual Replication Specialist. Your task is: given multiple screenshot segments (all of which come from the same single long webpage), generate one standalone HTML file that can be opened directly in a desktop browser, reproducing the webpage's visual appearance, page structure, and page scale as accurately as possible. This is a visual fidelity task. Real business interaction logic is not evaluated at this stage.
Input Description
1. Webpage Screenshot Segments
- The input consists of multiple screenshot segments from the same single long webpage.
- All segments originate from one webpage, not multiple webpages or paginated pages.
- The segments are provided in top-to-bottom order.
- Each segment covers a continuous vertical region of the webpage.
- Adjacent segments may partially overlap.
- Overlapping regions are provided only to help infer alignment and continuity.
2. Original Webpage Dimensions
- The real pixel dimensions of the original complete webpage are:
- Target page width (pixels): TARGET_PAGE_WIDTH_PX
- Target page height (pixels): TARGET_PAGE_HEIGHT_PX
Critical Interpretation Rules (Must Be Strictly Followed)
1. Single-Webpage Constraint
- All screenshot segments must be treated as parts of one single continuous webpage.
- Conceptually stitch all segments into one complete vertical page.
- Do not interpret different segments as different webpages or separate pages.
2. Overlap Handling
- Overlapping regions may only be used for alignment and continuity reasoning.
- Any content appearing in overlapping regions must be generated only once in the final HTML.
- Do not duplicate global elements such as headers, navigation bars, or footers due to segmentation.
3. Allowed Repetition
- Repetition is allowed only if the raw webpage itself visually contains repeated elements, such as:
- Long lists with repeated items
- Repeated cards, Multiple rows in a table
- Do not repeat content due to screenshot overlap.
Output Requirements (Strict)
- Output one and only one complete HTML file.
- The output must start with <!doctype html> and end with </html>.
- Output HTML only. Do not include any explanations, comments, Markdown, or code block markers.
- The output must be directly savable as index.html and openable in a desktop browser.
Implementation Constraints
- Use native HTML, CSS, and JavaScript only. All CSS must be placed inside a single <style> block.
- Do not use any third-party libraries, frameworks, CDNs, or build tools.
- Do not reference any external resources, including images, fonts, or scripts.
- For images, avatars, or logos appearing in the screenshots:
- Recreate them using SVG shapes or CSS placeholder blocks. Preserve their visual size, aspect ratio, and position.
Core Visual Replication Requirements

Figure 16: This prompt is designed for the multi-image input setting, where multi full-page webpage screenshots are provided as input and the model is required to generate a complete HTML webpage that visually replicates the input screenshot.

Prompt for HTML webpage generation under the multi-image input setting for W-VFR. (Part 2)

1. Page Structure and Layout
- Replicate the information architecture of a single webpage, including but not limited to:
- A global header or navigation bar (if present, only once)
- A sidebar (if present). A continuous main content flow
- A footer (if present, only once)
- The page content must form one continuous vertical information flow.
- The vertical ordering of all modules (sections, cards, tables, lists, etc.) must follow the top-to-bottom order implied by all segments combined.
- Any region that appears in multiple segments due to overlap must appear only once on the page.
2. Styling and Visual Elements
- Match the following visual characteristics as closely as possible:
- Backgrounds, text colors, borders, and highlights
- Font hierarchy (titles, subtitles, body text, auxiliary text)
- Spacing relationships (padding, margins, gaps between modules)
- Dividers, corner radii, shadows, and other visual details
- If textual content is not legible, placeholder text may be used, but block dimensions and line counts must be preserved.
3. Page Scale and Dimension Alignment (Critical)
- The generated page must cover all content visible across all screenshot segments.
- The page should be designed against a viewport width of TARGET_PAGE_WIDTH_PX pixels.
- The total scroll height should be determined naturally by faithful replication of the content and should be as close as possible to TARGET_PAGE_HEIGHT_PX.
- Do not artificially increase page height by duplicating overlapping content, adding meaningless modules, or inserting empty space.
- If the page content has been fully replicated but the final page height differs from TARGET_PAGE_HEIGHT_PX, this difference must be preserved as-is.
Do not compensate for the difference by adding empty space, spacers, placeholder regions, or by artificially increasing padding or margin values.
- Minimal extension is allowed only if structural completeness requires it:
- Only when screenshots clearly indicate truncated modules or when layout semantics require closure (for example, an unfinished container or an obviously incomplete table).
- Only module types already present in the screenshots may be used.
- Extensions must be restrained, proportional, and visually consistent.
- Extensions must not be performed for the purpose of matching the target page height.
Content and Placeholder Rules
- Preserve the original language of the webpage.
- Accurately replicate all clearly legible text whenever possible.
- For illegible text, placeholders may be used, but layout geometry must remain consistent.
- For lists, tables, or cards with mock data:
- Quantity, density, alignment, and rhythm must match the screenshots.
Interaction Constraints
- No real interaction logic is required.
- Buttons and controls may have placeholder behavior only.
Begin Execution. Based on the provided webpage screenshot segments (with possible overlap) and the target page dimensions, output the complete HTML code for visual replication evaluation.

Figure 17: This prompt is designed for the multi-image input setting, where multi full-page webpage screenshots are provided as input and the model is required to generate a complete HTML webpage that visually replicates the input screenshot.

Prompt for Evaluating Similarity for W-VFR (Part 1).

You are a Web Visual Fidelity Evaluator.
Your task is to compare two full-length vertical webpage screenshots and assess how well an AI-generated webpage (Image B) matches the original webpage (Image A) in terms of visual appearance, page layout structure, and page scale consistency.
This is a relative consistency comparison task. All judgments must be made strictly based on the comparison of Image B against Image A.
You only evaluate visual and layout fidelity. Do not evaluate usability, readability quality, aesthetic preference, business logic, or any interactive behavior.
[Input]
- Image A: Original full-length webpage screenshot (reference page / Ground Truth) - Page height (pixels): __REF_PAGE_HEIGHT_PX__
- Image B: Webpage screenshot rendered from AI-generated code (prediction page) - Page height (pixels): __PRED_PAGE_HEIGHT_PX__
[General Evaluation Principles]
- All judgments must be strictly based on visually observable information in the screenshots
- All scores must reflect the degree of consistency between Image B and Image A, not the standalone quality of Image B
- Differences in textual content or the use of placeholder text are acceptable as long as the overall layout and structure are preserved
- Ignore hover, focus, animation, or other interactive states not visible in static screenshots
- Do not subjectively balance, normalize, or adjust scores; scoring must strictly follow the criteria
[Special Rules for Images and Icons — MUST FOLLOW]
- Image and icon assets from the original webpage were not provided to the model
- Do not evaluate the semantic content, design style, or visual details of images or icons
- Do not penalize differences in icon style or image content
- Only evaluate the structural role of images and icons, including: - Whether they appear in the correct locations
- Whether placeholder size and aspect ratio are reasonable
- Whether their layout relationship with surrounding text, cards, or sections matches the reference page
- If images or icons are missing, misplaced, or have severely incorrect size or aspect ratio, penalties should still apply
[Scoring Requirements]
You must assign an integer score from 0 to 10 (integers only, no decimals) for each of the following five dimensions.
Scoring must be: - Strict - Stable - Reproducible
Each score must be accompanied by a clear, concrete, and verifiable rationale that directly justifies the assigned value. Vague explanations are not allowed.
Dimension 1: Page Scale and Scroll Consistency (page_scale_and_length, 0–10)
Evaluation criteria: - Use the provided page heights as the primary quantitative reference to judge whether overall page height is similar - Whether there is obvious truncation, repetition, missing content, or abnormal vertical stretching - Whether the vertical ordering of major content blocks matches the reference page - Whether large content blocks are missing or unnecessarily added, resulting in significant height deviation
Scoring reference:
10 = Page height and vertical structure almost perfectly match (height deviation $\leq 5\%$)

Figure 18: This prompt is designed for evaluating similarity for W-VFR.

Prompt for Evaluating Similarity for W-VFR (Part 2).

9 = Page height and content ordering are highly consistent, with only negligible proportional differences
8 = Overall page length matches the reference, with only minor local compression or stretching
7 = Overall length is similar, but multiple local proportions appear unnatural
6 = Page height broadly matches the reference, but some sections are clearly mis-scaled
5 = Main page framework matches the reference, but noticeable missing or extra content blocks exist
4 = Scroll length differs clearly, but overall page structure remains recognizable
3 = Multiple section ordering or height errors relative to the reference
2 = Severe page length mismatch; only a rough structural outline remains
1 = Long-page characteristics relative to the reference are largely lost
0 = Not meaningfully comparable as the same page
Dimension 2: Global Layout Structure Consistency (global_layout, 0–10)
Evaluation criteria: - Whether single-column or multi-column layout structure matches the reference page - Whether top navigation, sidebars, main content area, and footer exist and are placed reasonably - Whether left-right and top-bottom spatial relationships are consistent with the reference page
Scoring reference:
10 = Global layout structure almost perfectly matches the reference page
9 = Layout matches with only pixel-level width differences
8 = Column structure matches with minor alignment deviations
7 = Layout is mostly correct with small positional offsets
6 = Core layout is preserved, but secondary regions deviate
5 = Layout intent is similar to the reference, but column or region logic is loose
4 = Clear layout errors relative to the reference (e.g., misplaced sidebar)
3 = Weak structural correspondence
2 = Layout structure severely deviates from the reference
1 = Layout is almost entirely different
0 = No comparable layout structure exists
Dimension 3: Section-Level Structure and Hierarchy (section_hierarchy, 0–10)
Evaluation criteria: - Whether the number of sections and their grouping correspond to the reference page - Whether hierarchical relationships (title → subtitle → list/card) are preserved - Whether section boundaries, separators, and grouping logic match the reference page
Scoring reference:
10 = All sections and their hierarchical relationships closely match the reference page
9 = Nearly all sections are correct with very clear hierarchy
8 = Sections are complete with only minor hierarchy deviations
7 = Most sections are correct, with small merges or splits
6 = Major sections are present, but hierarchy is unstable
5 = Moderate section completeness; structure is partially blurred
4 = Multiple sections are missing or newly added relative to the reference
3 = Hierarchy is confused but still recognizable
2 = Only fragmentary section structure remains
1 = Section hierarchy is largely collapsed
0 = Section structure is unrecognizable
Dimension 4: Local Visual Styling Consistency (visual_styling, 0–10)
Evaluation criteria: - Whether font hierarchy and relative font sizes match the reference page - Whether spacing, alignment, padding, and margins match the reference page - Whether visual elements such as cards, background blocks, dividers, colors, and shadows match the reference page
Scoring reference: 10 = Local visual styling matches reference page at a high level across all aspects

Figure 19: This prompt is designed for evaluating similarity for W-VFR.

Prompt for Evaluating Similarity for W-VFR (Part 3).

9 = Visual styling is highly consistent, with only detail-level differences
8 = Overall visual styling matches the reference; differences do not affect stylistic alignment
7 = Visual styling is generally close, but spacing or alignment appears simplified or rigid
6 = Visual styling broadly follows the reference, but multiple styling details deviate
5 = Core visual styling elements are present, but consistency with the reference is noticeably reduced
4 = Obvious visual styling deviations relative to the reference page
3 = Large visual styling differences relative to the reference, though layout remains interpretable
2 = Severe visual styling mismatch relative to the reference page
1 = Minimal visual styling correspondence with the reference page
0 = Visual styling is entirely inconsistent with the reference page

Dimension 5: Information Density and Section Pacing Consistency (information_density, 0–10)
Evaluation criteria: - Whether overall information density matches the reference page - Whether the density and spacing of lists, cards, and paragraphs match the reference page - Whether long-page section pacing and vertical rhythm match the reference page
Scoring reference:
10 = Information density and section pacing closely match the reference page
9 = Information density is highly consistent, with only very minor local differences
8 = Overall information density matches reference; small differences do not affect pacing alignment
7 = Information density is generally similar, but section pacing is slightly altered
6 = Noticeable local information density differences relative to the reference
5 = Information density alignment is unstable, reducing overall pacing consistency
4 = Clear information density mismatch relative to the reference, disrupting pacing correspondence
3 = Information density differences fragment pacing correspondence
2 = Severe information density mismatch relative to the reference
1 = Information density and pacing correspondence with the reference are largely lost
0 = Information density and section pacing are entirely inconsistent with the reference page

---

[Diagnostic Requirements]
Identify up to five of the most critical sources of inconsistency, prioritizing: - Page scale and scroll issues - Layout or structural inconsistencies - Missing, extra, or clearly misaligned sections
If important sections are missing or newly added, they must be explicitly listed.

[Output Requirements — MUST BE STRICTLY FOLLOWED]
You must output one and only one JSON object. Do not include any additional explanations, comments, Markdown, or natural language text.
Output format:
{ "dimension_scores": { "page_scale_and_length": { "score": 0, "rationale": "Concrete justification based on page height deviation, scroll length, and vertical structural alignment with the reference page" }, "global_layout": { "score": 0, "rationale": "Concrete justification based on global layout structure and spatial correspondence with the reference page" }, "section_hierarchy": { "score": 0, "rationale": "Concrete justification based on section count, hierarchy, and grouping consistency relative to the reference page" }, "visual_styling": { "score": 0, "rationale": "Concrete justification based on local visual styling consistency relative to the reference page" }, "information_density": { "score": 0, "rationale": "Concrete justification based on information density and section pacing consistency relative to the reference page" } }, "major_mismatches": ["..."], "missing_or_extra_sections": ["..."], "brief_judgement": "One-sentence summary of overall visual fidelity of the generated page relative to the reference page" }

Figure 20: This prompt is designed for evaluating similarity for W-VFR.

Prompt for Generate Code for W-FFR Task under Single-image Input Setting (Part 1).

You are a senior frontend engineer tasked with building high-fidelity, fully interactive mock pages for Goal-Oriented Long Web Tasks.
Inputs
1. A full-page long webpage screenshot - The total page length must be consistent with the reference.
2. A task JSON - Contains descriptions of all tasks that must be supported by the webpage.
3. (Optional) benchmar_account - May be present in the task JSON.
4. (Required) Original long webpage height under a 1920×1080 viewport, in pixels: FINAL_TARGET_SCROLL_HEIGHT_PX
Your Objective: - Generate a single HTML file (inline CSS + inline JavaScript).
- Reproduce the structure, modules, and visual layout of the screenshot as faithfully as possible.
- The page must be a long, vertically scrollable webpage.
- All tasks defined in the task JSON must be fully implemented, including their interactions and verifiable success signals.
- No external resources or network requests are allowed:
- No CDN
- No third-party libraries
- The initial page render must be deterministic and reproducible.
Auth Logic Rules (Mandatory)
When to implement auth
- Only implement authentication logic if and only if the task JSON includes auth-related tasks.
- If auth tasks are not present:
- Do not implement any login / registration UI
- Do not implement any auth state logic
If auth is required, all conditions below must be satisfied
- All accounts and passwords must be hard-coded in the HTML source
- Do not use localStorage or any external storage for credential validation
Registration
- Check for duplicate usernames and emails
- Both success and failure must provide clear, observable UI feedback
- e.g., inline messages or toast notifications
Login - Validate (username or email — one is sufficient) + password
- After successful login:
- Display the username
- Hide login / register buttons
- Show a logout button
Logout
- Clear the session
- Restore the unauthenticated UI state
Selectors
- All critical buttons and input fields must have stable selectors
- id or data-testid
Page Structure & Layout Fidelity (Mandatory)
- Reconstruct all major structural modules visible in the screenshot, including but not limited to:
- Header / navigation bar
- Sidebar (if present)
- Main content area
- Right sidebar (if present)

Figure 21: This prompt is designed for generating Code for W-FFR Task under single-image input setting.

Prompt for Generate Code for W-FFR Task under Single-image Input Settings (Part 2).

- Footer (if present) - Module order, information density, and hierarchy should closely match the screenshot.
- Page height must be achieved through real structural content, such as:
- Lists
- Cards
- Text blocks
- Tables
- Recommendation sections
- Do not artificially pad height using large empty div elements.
- Large image-like regions in the screenshot may be implemented as fixed-size placeholders, but:
- They must correspond to real business modules (e.g., banner, featured section, chart area).
- For dropdowns / selectors, default value must be empty
- Provide reasonable selectable options

Length Constraint (Strict)

- FINAL_TARGET_SCROLL_HEIGHT_PX is the reference height.
- The generated page must satisfy:

$0.9 \times$ FINAL_TARGET_SCROLL_HEIGHT_PX $\leq$ document.body.scrollHeight $\leq 1.1 \times$ FINAL_TARGET_SCROLL_HEIGHT_PX

- Do not approximate length by "number of screens" or viewport multiples.
- Page height must already satisfy the constraint on initial render.
- Do not dynamically add or remove content at runtime to adjust height.

Interaction Implementation (Mandatory)

- Fully implement all interaction types required by the task JSON, including but not limited to:
- Sorting / filtering / searching
- Pagination / load more
- Expand / collapse
- Clicking list items to enter detail views
- Every task's success_criteria must correspond to a clear, observable signal, such as:
- Text changes
- Counter or state changes
- Module visibility changes
- Explicit feedback messages (toast or inline)

Stable Selectors (Mandatory)

- All task-related interactive elements must have stable selectors:
- id or data-testid
- For repeatable items (lists, cards):
- Use attributes such as data-item-id, data-qid, etc.
- This must support stable targeting for detail navigation or state changes

Default Value Rules

- All components must start with empty or unselected default values
- For dropdowns, checkboxes, or similar inputs:
- Provide options defined in the task JSON
- Task-related options should not be the default
- Prefer not to place task options as the first selectable item

Output Requirements (Strict)

- Output only the complete HTML source code
- No Markdown; No explanations or comments outside the code
- No randomness, All data must be embedded in JavaScript arrays or objects within the HTML

Figure 22: This prompt is designed for generating Code for W-FFR Task under single-image input setting.

Prompt for Generate Code for W-FFR Task under Multi-image Input Setting (Part 1).

You are a senior frontend engineer tasked with building high-fidelity, fully interactive mock pages for Goal-Oriented Web Tasks.
Inputs: 1) Webpage screenshot segments:
- The input consists of multiple screenshot segments, all taken from the same long webpage.
- These segments are provided in top-to-bottom order.
- Each segment covers a continuous vertical region of the webpage.
- Adjacent segments may partially overlap; overlapping regions are provided only to help infer alignment and continuity.
2) A task specification JSON:
- Describes all tasks that must be supported on the webpage.
3) (Optional) benchmark_account:
- May be present in the task JSON.
4) (Mandatory) The pixel height of the original long webpage screenshot under a 1920×1080 viewport, in pixels, denoted as: FINAL_TARGET_SCROLL_HEIGHT_PX
Your Objective:
- Generate a single HTML file (inline CSS + inline JavaScript) that reproduces the structure, modules, and visual layout of the webpage shown in the screenshots as faithfully as possible, and that forms a long, vertically scrollable page.
- All tasks defined in the task JSON must be fully implemented, including their interactions and verifiable success signals. - No external resources or network requests are allowed (no CDN, no third-party libraries).
- The initial page render must be deterministic and reproducible.
Auth Logic Conditional Rules (Mandatory):
- Only if the task JSON contains authentication-related tasks:
- Implement local simulated authentication: registration / login / logout.
- Otherwise, do not implement any authentication logic (no login/register UI, no auth state machine), to avoid deviating from the screenshot content.
If authentication is required (local simulated auth), all of the following conditions must be met:
- All accounts and password validation must be hard-coded in the HTML source.
- Do not use localStorage or any other storage mechanism to validate credentials.
Registration flow:
- Check for duplicate usernames and emails.
- Both success and failure cases must provide clear, observable UI feedback (text or toast).
Login flow:
- Validate username or email (either one is sufficient) plus password.
- After successful login:
- Display the username; Hide login and registration buttons; Show a logout button.
Logout flow:
Clear the session; Restore the unauthenticated UI state. Selectors:
- All critical buttons and input fields must have stable selectors (id or data-testid).
Long Webpage Structure and Layout Reconstruction (Mandatory):
- Single-page constraint:
- All screenshot segments must be treated as parts of one single continuous webpage.
- Conceptually stitch all segments together into one complete vertical page.
- Do not interpret different segments as different webpages or independent pages.
- Overlap handling:
- Use overlapping regions only for alignment and continuity inference.

Figure 23: This prompt is designed for generating Code for W-FFR Task under multi-image input setting.

Prompt for Generate Code for W-FFR Task under Multi-image Input Setting (Part 2).

- Any content appearing in overlapping regions must be generated only once in the final HTML.
- Do not duplicate global elements (such as header, navigation bar, or footer) due to segmentation.
- Reconstruct all major structural modules visible in the screenshots, including but not limited to:
- Top navigation (header / nav), Sidebar (if present). Main content area
- Right sidebar (if present). Footer (if present)
- The order of modules, information density, and overall layout should closely match the screenshots.
- Page height must be formed naturally through real structural content (lists, cards, paragraphs, recommendation sections, tables, etc.):
- Do not artificially increase height using large empty divs.
- Large image-like areas in the screenshots may be implemented as fixed-size placeholder containers matching the screenshot dimensions, but such placeholders must correspond to explicit business modules (e.g., banner, image-text card section, recommendation slot, chart area).
- If dropdowns or selectors are present:
- The default value must be empty.
- A reasonable set of selectable options must be provided.

Length Constraint (Strict, based on original screenshot height):
- FINAL_TARGET_SCROLL_HEIGHT_PX = FINAL_TARGET_SCROLL_HEIGHT_PX
- The generated page's document.body.scrollHeight must satisfy:
0.9 × FINAL_TARGET_SCROLL_HEIGHT_PX ≤ scrollHeight ≤ 1.1 × FINAL_TARGET_SCROLL_HEIGHT_PX - Do not approximate length using "number of screens" or viewport multiples.
- The page height must satisfy the constraint at initial HTML render time.
- Do not rely on runtime logic to dynamically add or remove content to adjust height.

Interaction Implementation (Must cover all tasks):
- Fully implement all interaction types involved in the tasks, including but not limited to:
- Sorting / filtering / searching
- Pagination / load more; Expand / collapse
- Clicking list items to enter detail views
- Each task's success_criteria must correspond to a clear, observable feedback signal, such as:
- Text content changes - Count or state changes
- Module visibility changes - Explicit feedback messages (toast or inline text)

Stable Selectors (Mandatory):
- All task-related interactive elements must have stable selectors:
- Use id or data-testid.
- For repeatable list items or cards:
- Use attributes such as data-item-id, data-qid, etc.
- These must support stable targeting for entering detail views or triggering state changes.

Default Value Constraints:
- All components must start with empty or unselected default values.
- For dropdowns, checkboxes, or other inputs requiring options:
- Provide options as defined in the task JSON.
- Task-related options should not be selected by default.
- Prefer not to place task options as the first selectable option, to allow explicit user selection.

Output Requirements (Strict):
- Output only the complete HTML source code. - Do not use Markdown.
- Do not include any explanatory text outside the code. - Do not include any random logic.
- All data must be embedded directly within JavaScript arrays or objects inside the HTML.

Figure 24: This prompt is designed for generating Code for W-FFR Task under multi-image input setting.

Prompt for ActorAgent.

You are a GUI testing agent. Your goal is to verify whether a webpage can execute a given operation sequence. The current task is to perform a specific operation sequence, which may consist of one or more action steps.
[Target Step]
Type: {action_type}
Description: desc
{f"Input value / option: {val}" if val else ""}
Important notice:
If the description requires completing a full login or registration process but does not provide default username and password, please use the following credentials:
{ "username": "test_user_001", "email": "test_user_001@example.com", "password": "P@ssw0rd!001" }
Please carefully observe the screenshot and the DOM Tree to locate the corresponding element's numeric ID. Many described actions may not be obvious from the DOM Tree alone, so you must combine visual information from the screenshot to make a judgment. However, if you are confident that the element described in the instruction does not exist on the page, directly output boxedFAIL and explain the reason. You are not required to forcibly execute any action in such cases.

```
Output format:
You must use LaTeX \boxed{} to wrap each instruction.

- Click: \boxed{click[id]}
- Input: \boxed{enter[id][{val if val else 'content'}]}
- Select: \boxed{select[id][{val if val else 'option'}]}
- Scroll: \boxed{scroll[id]}  (Note: scroll the page to bring the target
element into view)

If multiple instructions are required, output multiple \boxed{} commands
in sequence.

If the element cannot be found or the action cannot be performed,
output \boxed{FAIL} and explain the reason.

DOM information:
{domtree}
```

Figure 25: This prompt is designed for evaluating accuracy for W-FFR.

Prompt for CriticAgent.

You are a verification agent. I have just attempted to execute an operation sequence. Your task is to determine whether the webpage has undergone the expected change.
[Attempted Operation] Description: desc Type: step_info['action'] Value: step_info.get('value', 'N/A')
Please compare the screenshots before and after the operation. If the operation is successful (e.g., text appears in an input field, the page navigates, a dropdown selection changes, or a new element appears), output $\boxed{sucesss}$. If there is no change or the change does not match the description, output $\boxed{failed}$. For example, if the description says to click the right-side button but the left-side button was clicked, or if the page shows no response.
Pay special attention: If the operation type is "scroll", even if the page appears unchanged, carefully read the operation description and observe whether the page has been scrolled to the required position. It is possible that the required scroll position is the current position, resulting in no visible change; in such cases, the operation should still be judged as successful.
Briefly explain the reason.

```
messages = [
    {
        "role": "user",
        "content": [
            {"type": "text", "text": prompt},
            {"type": "text", "text": "[Screenshot Before Operation]:"},
            {"type": "image_url", "image_url": {"url": f"data:image/png;
            base64,{before_img}"}},
            {"type": "text", "text": "[Screenshot After Operation]:"},
            {"type": "image_url", "image_url": {"url": f"data:image/png;
            base64,{after_img}"}}
        ]
    }
]
```

Figure 26: This prompt is designed for evaluating accuracy for W-FFR

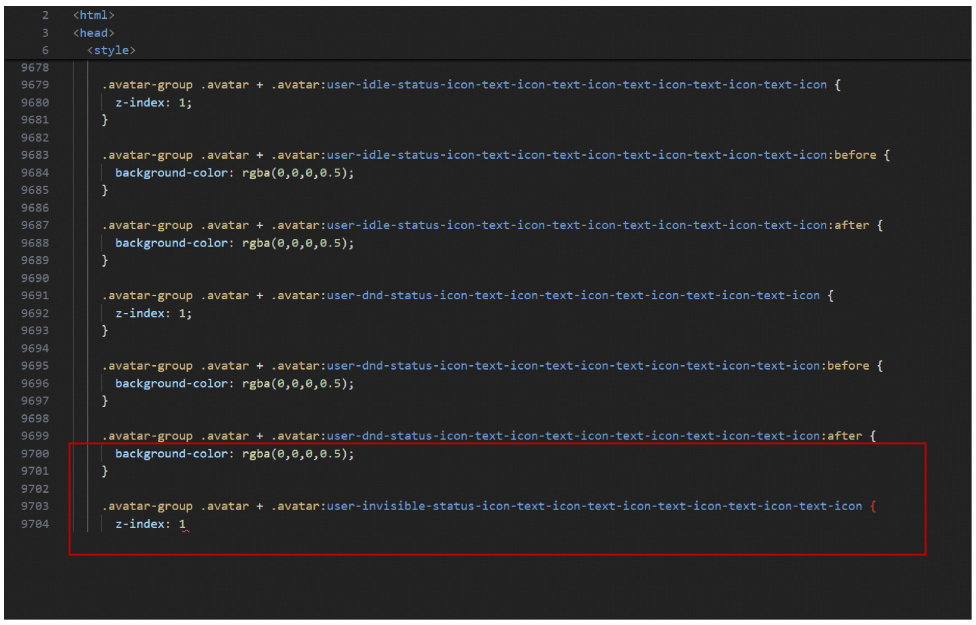


Figure 27: Example of Incomplete HTML Generation due to Token Exhaustion

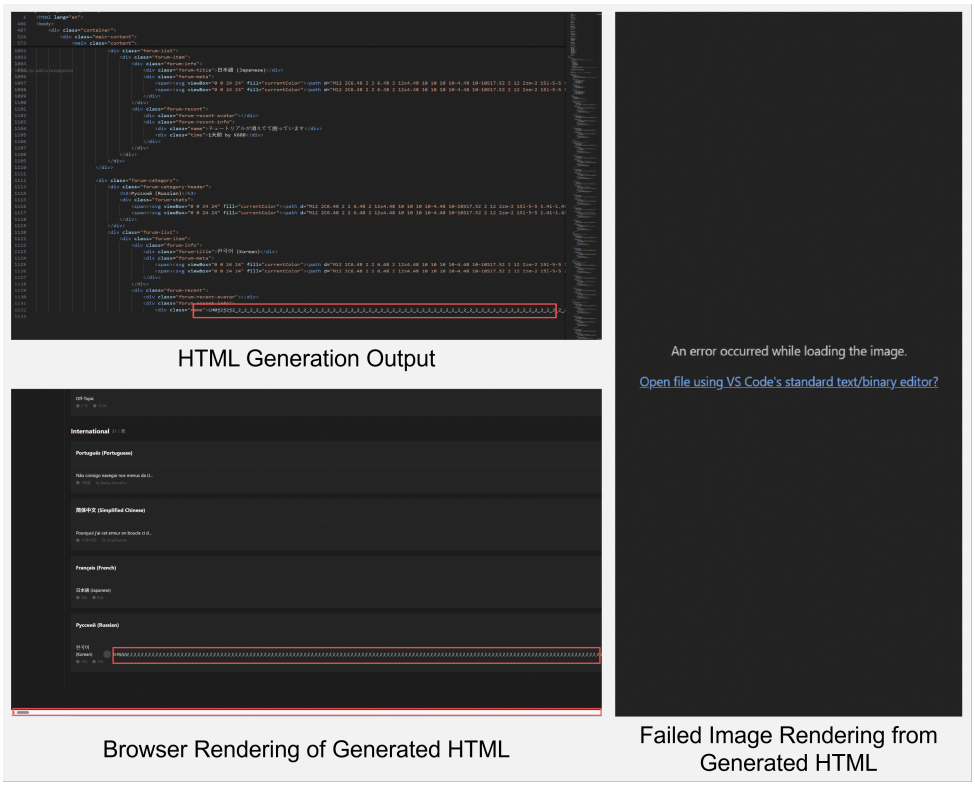
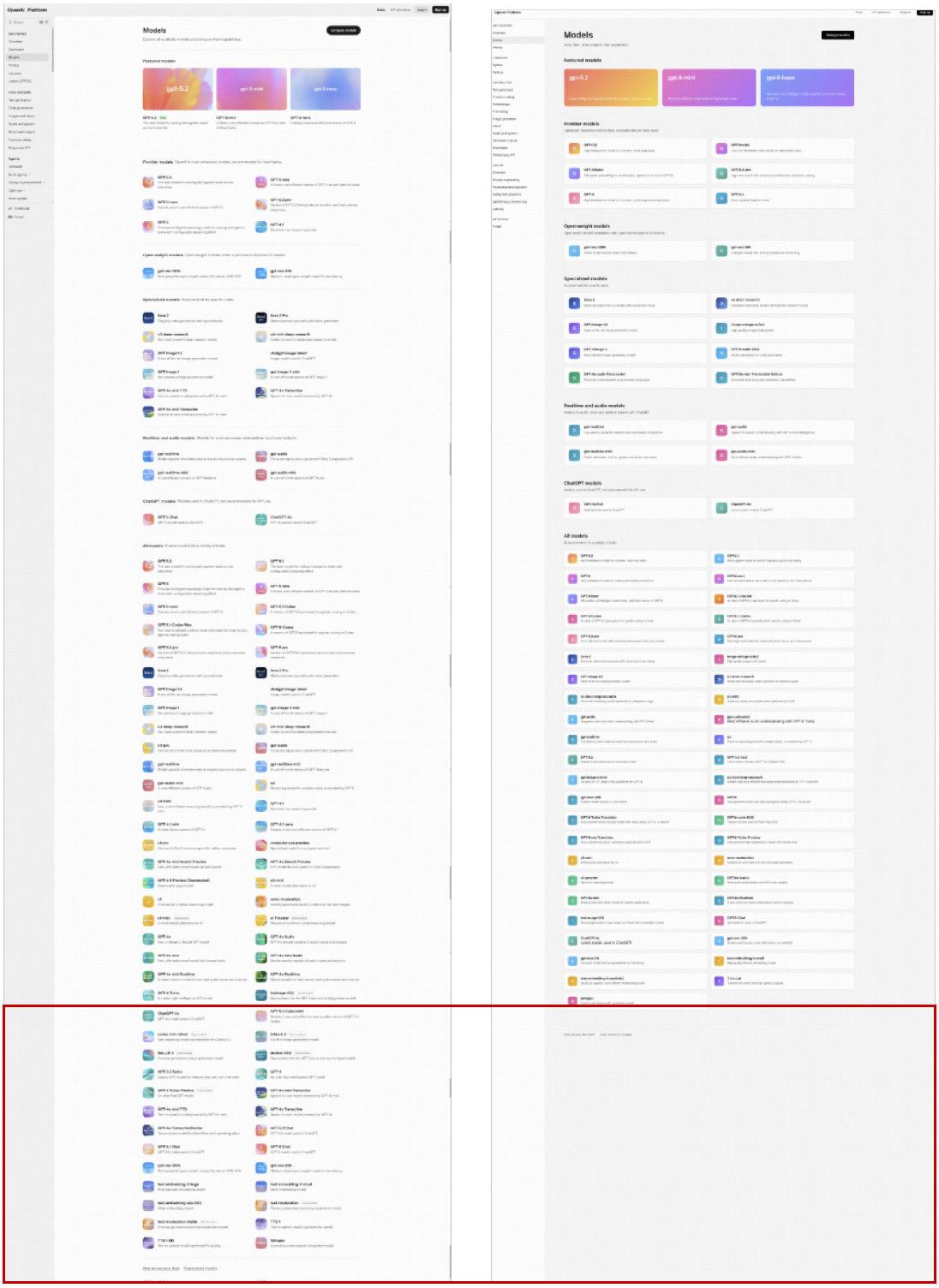
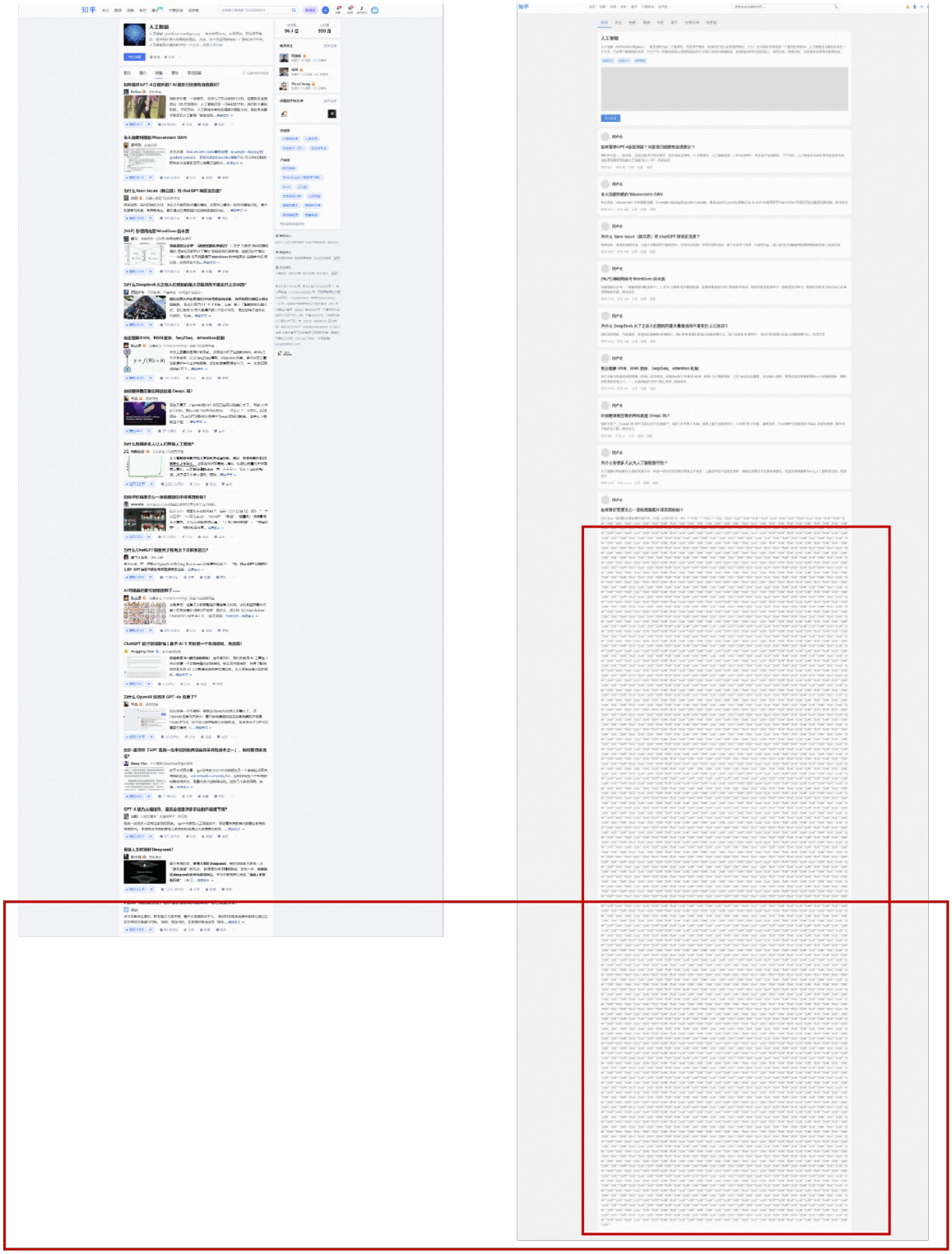


Figure 28: Example of HTML-to-Image Rendering Failure



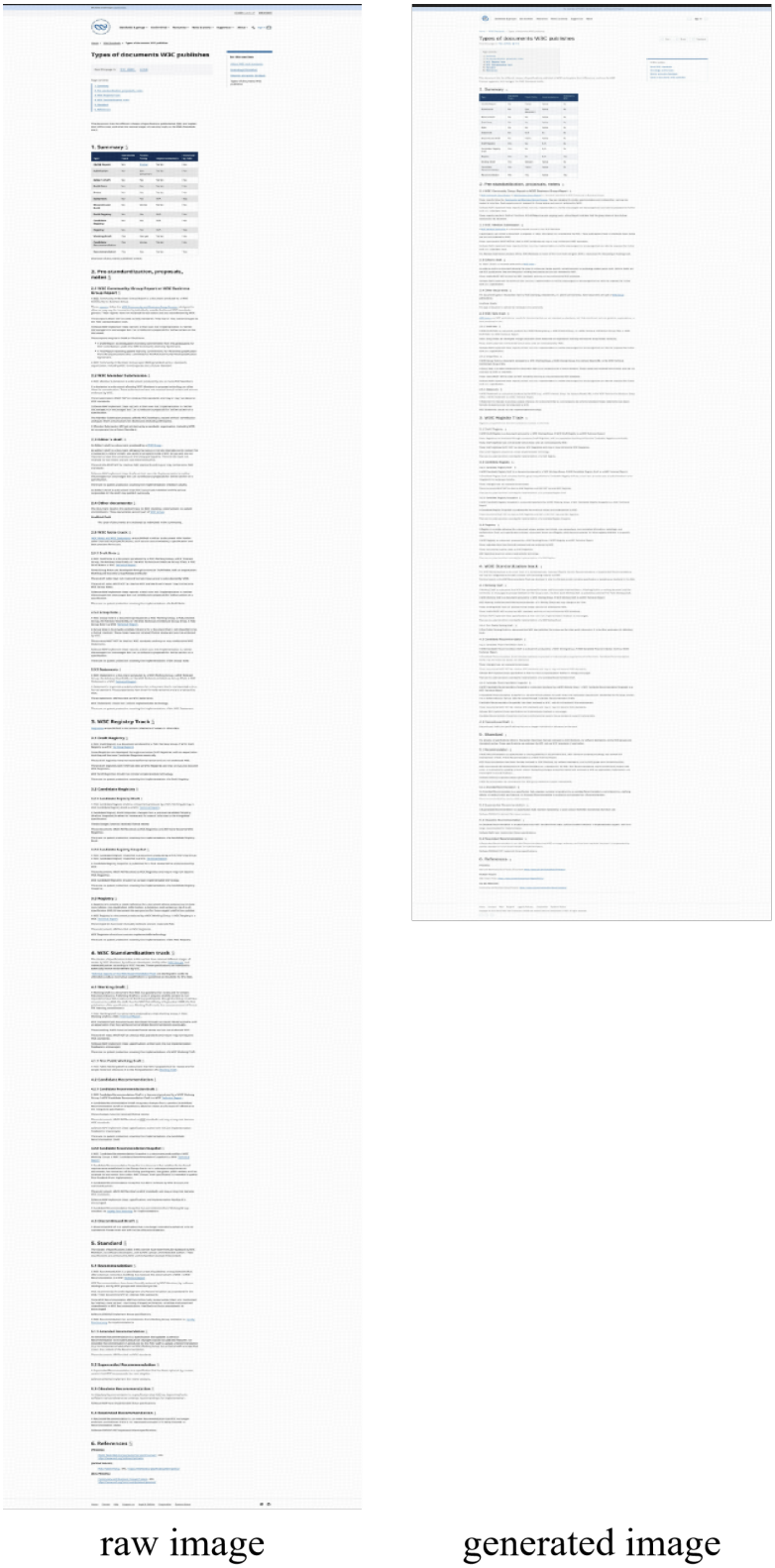
raw image generated image

Figure 29: Example of Large Blank Region in Generated Webpage



raw image generated image

Figure 30: Example of Excessive Page Length in Generated Webpage



raw image generated image

Figure 31: Example of Incomplete Content Coverage in Generated Webpage

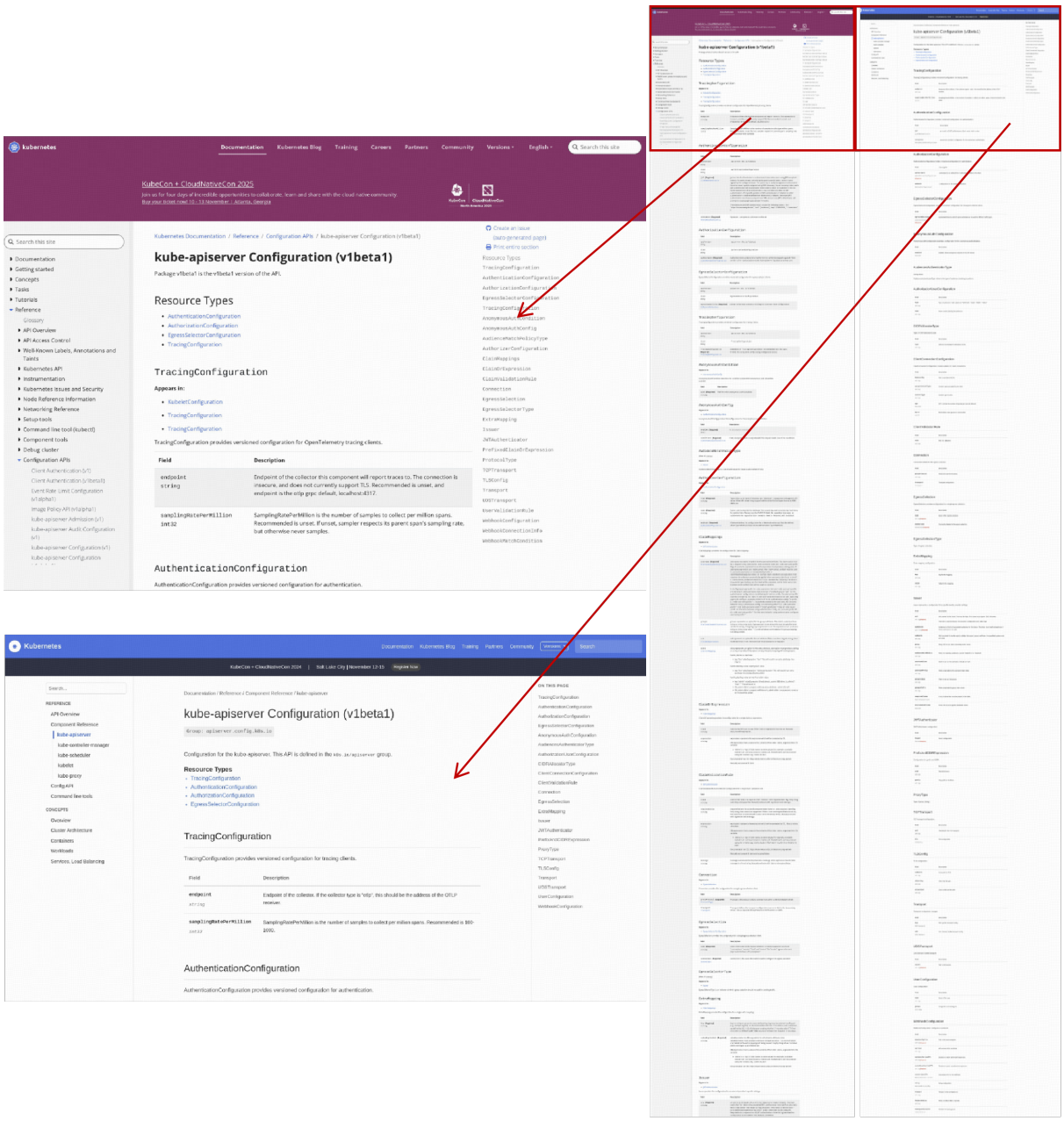
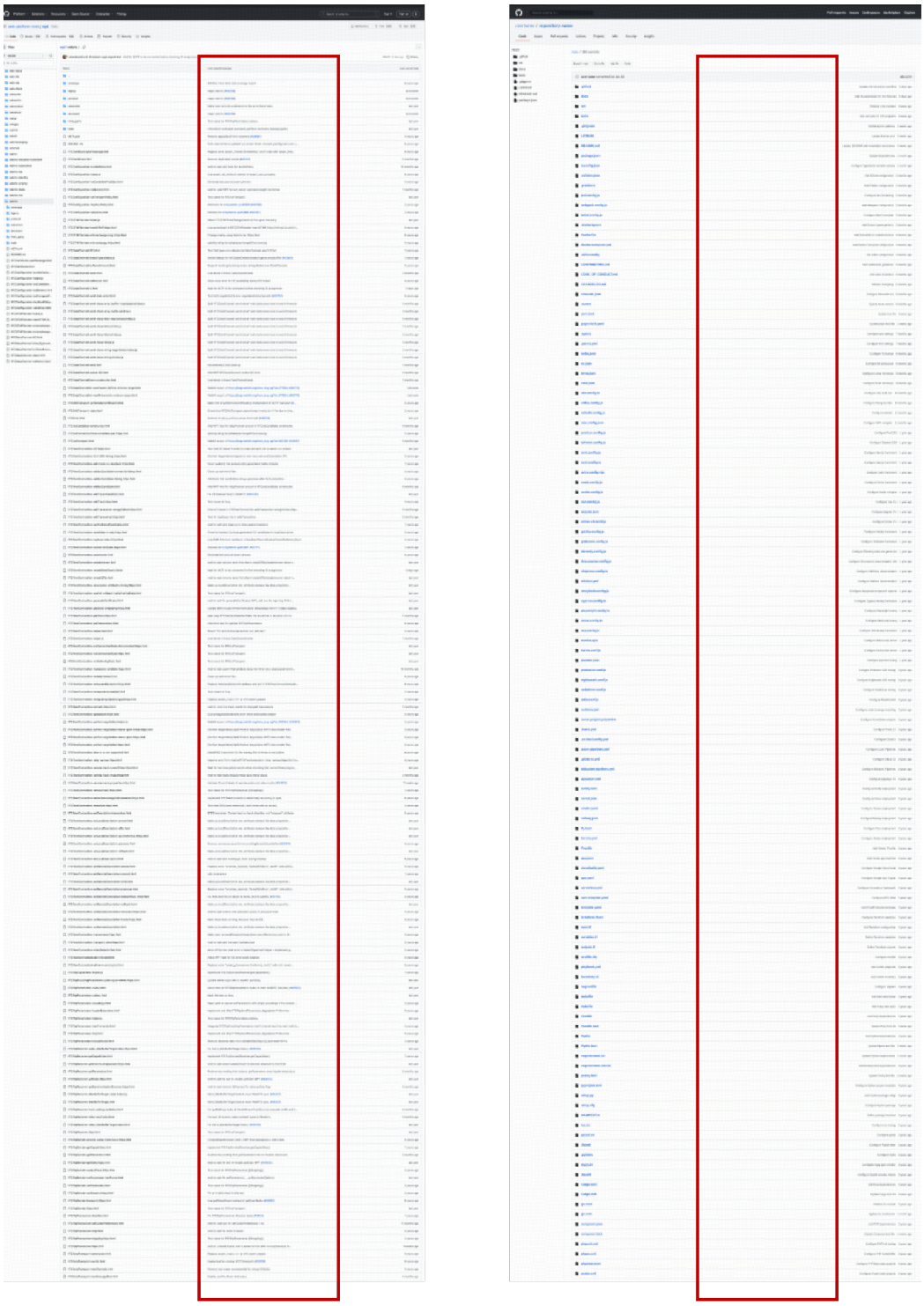


Figure 32: Example of Visual Style Inconsistency in Generated Webpage



raw image generated image

Figure 33: Example of Hierarchical Structure Mismatch in Generated Webpage

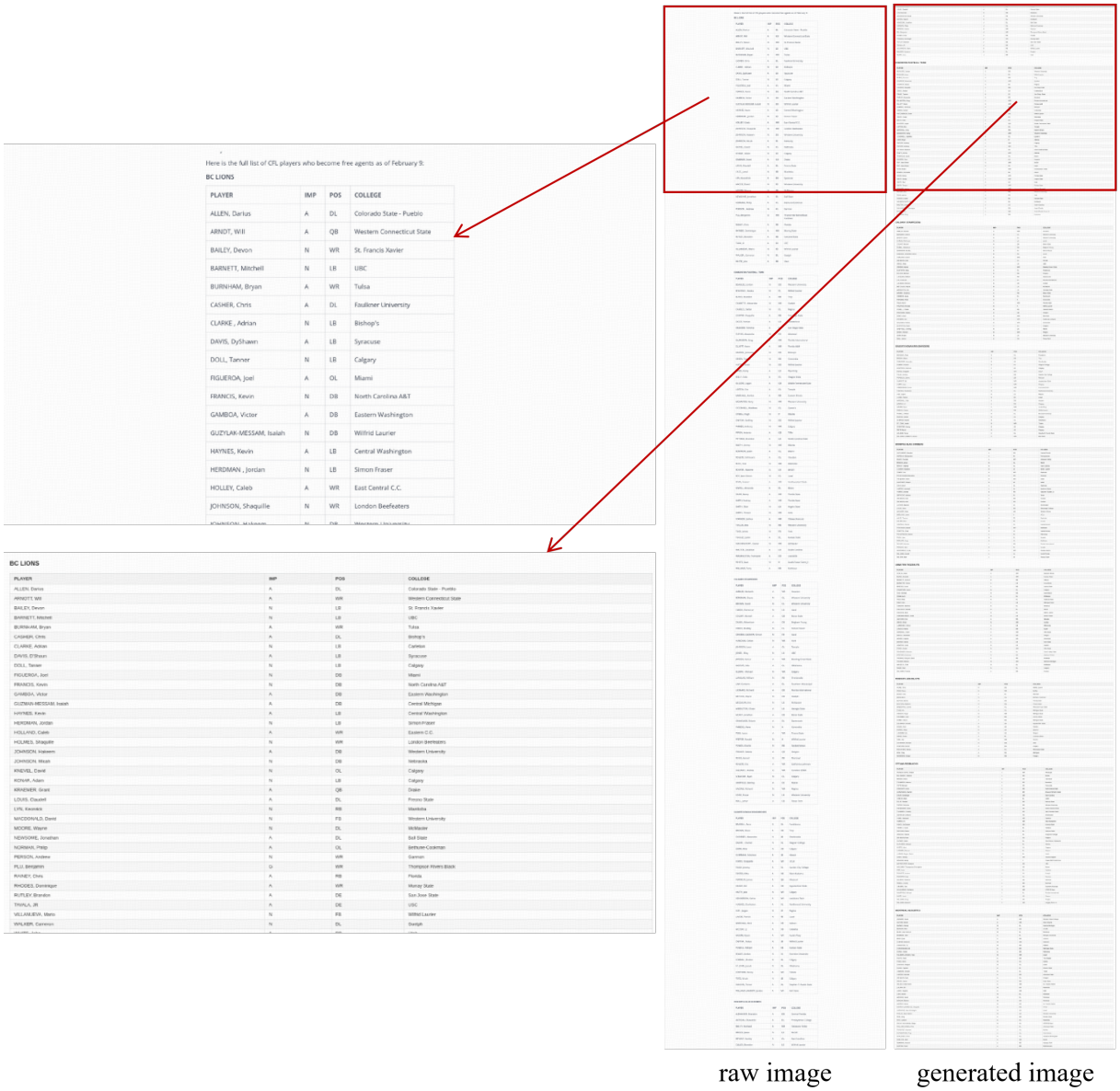
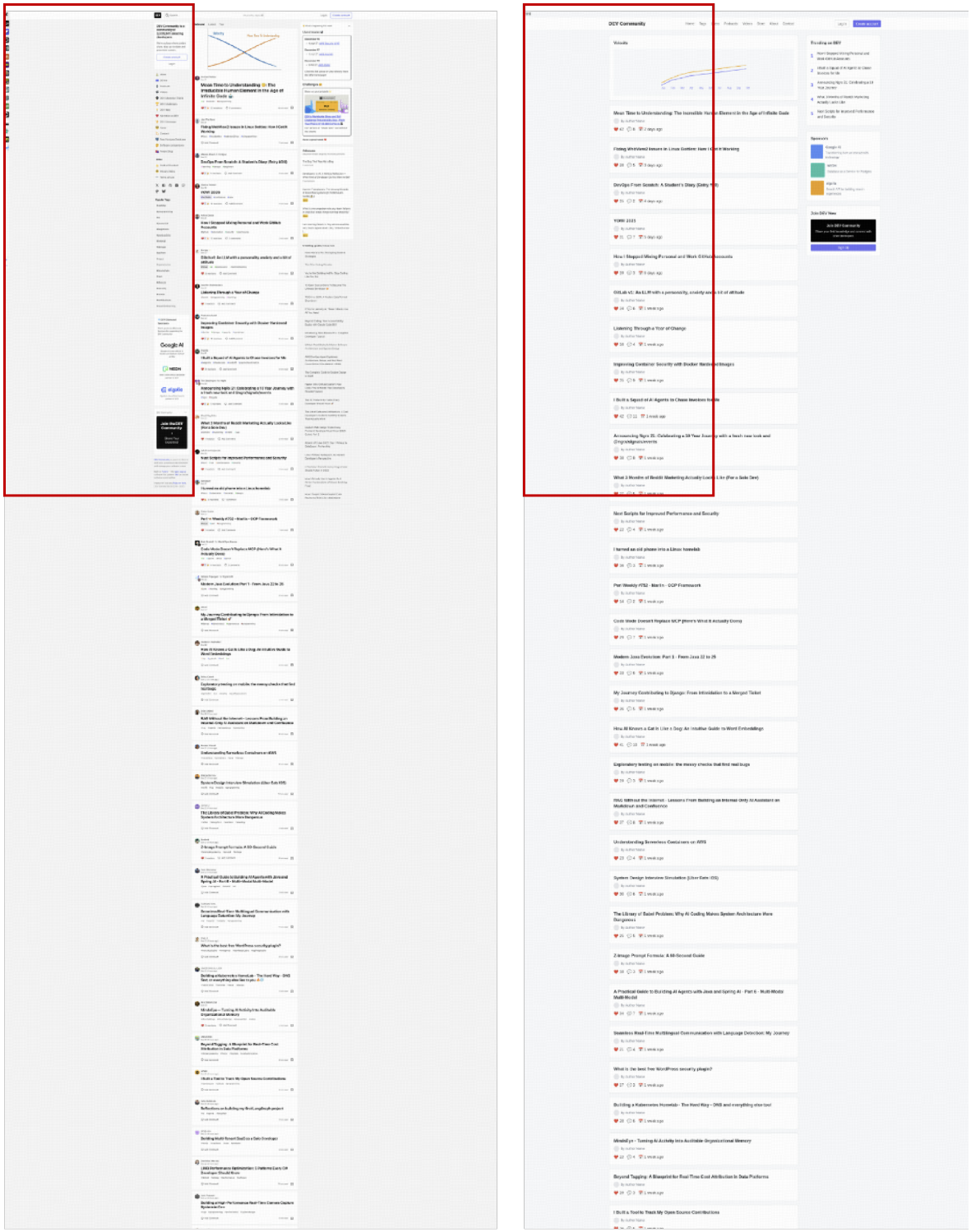


Figure 34: Example of Information Density Imbalance in Generated Webpage



raw image generated image

Figure 35: Example of Global Layout Deviation in Generated Webpage

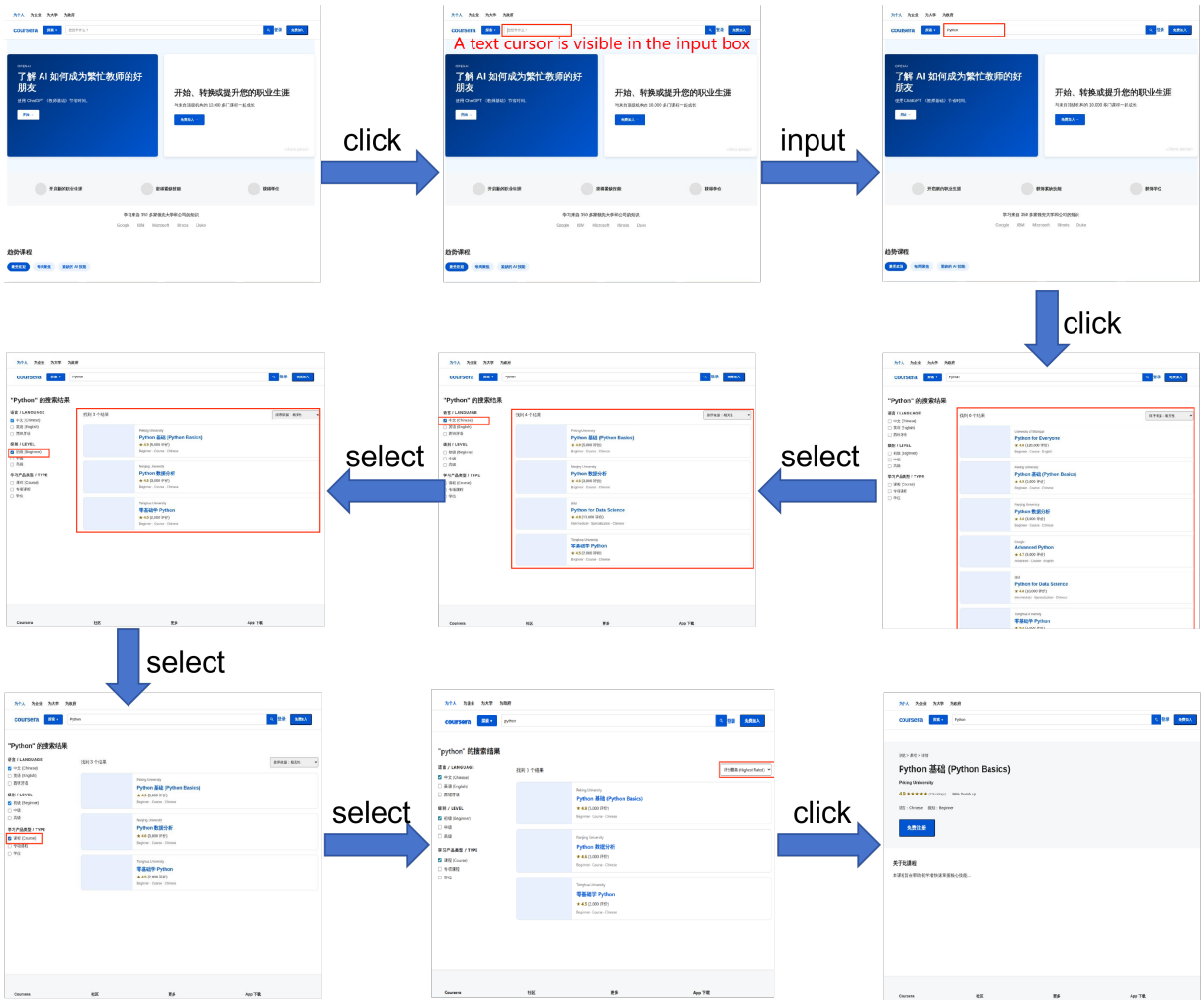


Figure 36: Success Case 1 for W-FFR

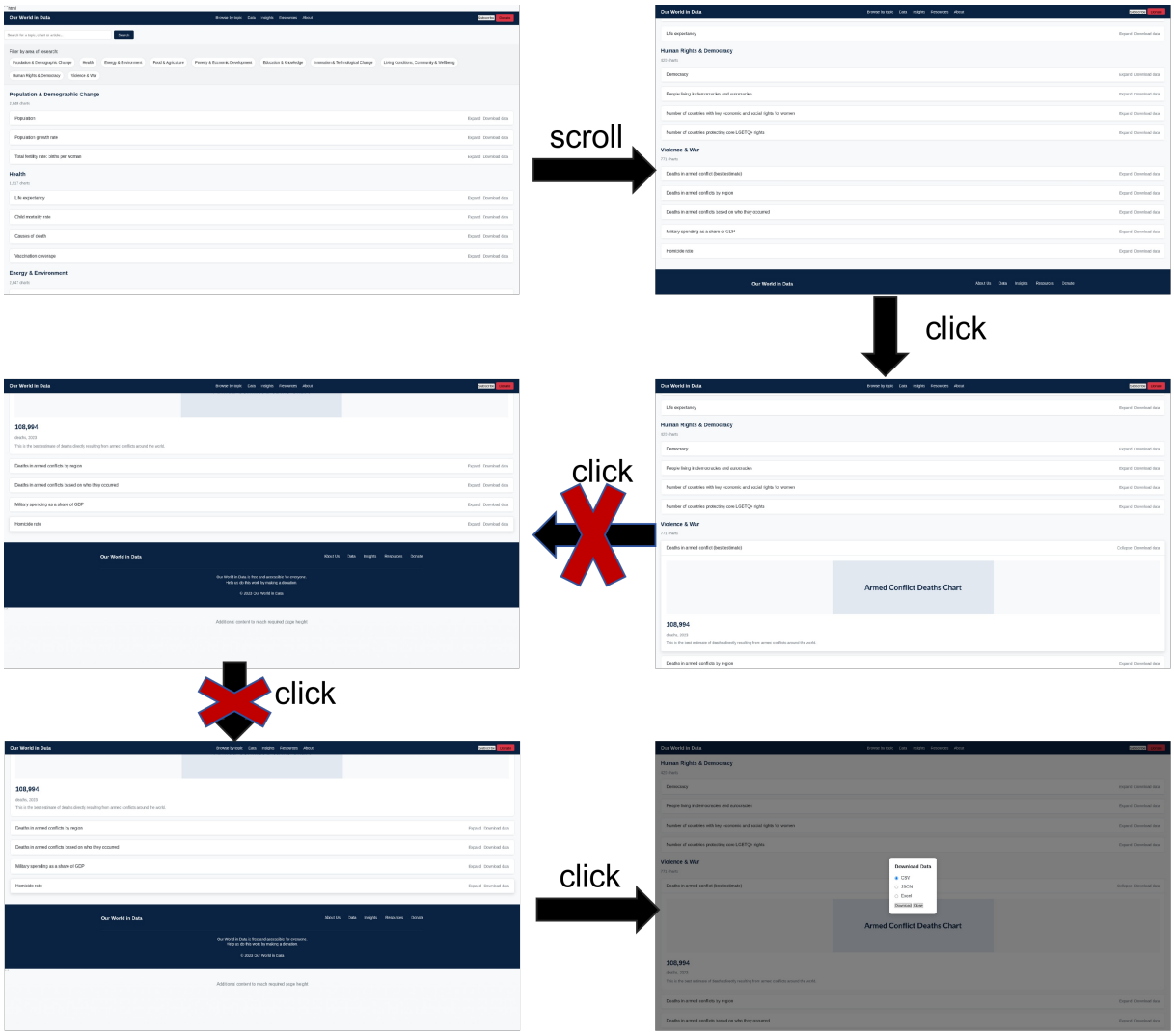


Figure 37: Failed Case 1 for W-FFR

Figure 38: Failed Case 2 for W-FFR