WebMMU: A Benchmark for Multimodal Multilingual Website Understanding and Code Generation

Rabiul Awal[1,2,3*] **Mahsa Massoud**[1,2,4*] **Zichao Li**[1,2,4*] **Aarash Feizi**[1,2,4]
Suyuchen Wang[1,2,3] **Christopher Pal**[1,2,6] **Aishwarya Agrawal**[2,3] **David Vazquez**[1]
Siva Reddy[1,2,4] **Juan A. Rodriguez**[1,2,5] **Perouz Taslakian**[1] **Spandana Gella**[1]
Sai Rajeswar[1,2]
[1]ServiceNow, [2]Mila, [3]Université de Montréal, [4]McGill University,
[5]École de Technologie Supérieure (ETS), [6] Polytechnique Montréal

Abstract

Understanding diverse web data and automating web development presents an exciting challenge for agentic multimodal models. While existing benchmarks address isolated web-based tasks such as website-based Visual Question Answering (VQA) and UI-to-code generation, they lack a unified evaluation suite for assessing web agents that interact with and reason about web environments. We introduce WebMMU, a large-scale benchmark for evaluating web agents across multilingual website understanding, HTML/CSS/JavaScript code editing, and mockup-to-code generation. WebMMU provides a comprehensive evaluation suite with real-world website data, multi-step reasoning tasks, and functional UI understanding. Benchmarking state-of-the-art multimodal models on WebMMU reveals significant limitations in web-based reasoning, layout understanding, and structured code generation, particularly in preserving UI hierarchy, handling multilingual content, and producing robust and functional code. While existing models are optimized for English settings, WebMMU highlights the challenges of cross-lingual adaptation in real-world web development. These findings expose critical gaps in current models' ability to understand website structures, execute user instructions, and generate high-quality web code, underscoring the need for more advanced multimodal reasoning in AI-driven web understanding and development. The dataset and code are publicly available at webmmu.github.io.

1. Introduction

The web is an integral part of daily life, facilitating information access, commerce, and communication. Artificial Intelligence (AI) models capable of reasoning over the *Visual Web* could enable intelligent web agents that assist users in web understanding—extracting multi-faceted insights, supporting decision-making (e.g., identifying shopping items within a budget), and adapting to multilingual environments [16]. Beyond understanding, AI also holds promise in web design and development, automating front-end creation, UI modifications, and code generation[3]. Unlike unimodal tasks that process only text or natural images, Visual Web reasoning requires AI to integrate structured UI elements, spatial layouts, textual content, interactive components, and embedded visuals. While recent advancements in multimodal web understanding and web-agentic systems [27, 45] enable AI to interact with and extract knowledge from web interfaces, AI-driven web comprehension and development remain underexplored. Web automation methods [22, 26] have improved accessibility and streamlined front-end workflows, yet existing approaches focus on isolated tasks. As such, despite growing interest in multimodal, agentic, and code-generation AI models [28, 45], current benchmarks remain fragmented, lacking a unified framework for evaluating AI's capabilities in web-based reasoning, structured interaction, multilingual adaptation, and full-stack development.

Existing datasets have attempted to tackle specific aspects of web-based AI, but they remain fragmented and insufficient for a comprehensive evaluation of Visual Web understanding. Website VQA datasets, such as WebQA[9] and WebSRC[12], primarily focus on textual content retrieval, neglecting reasoning over UI structures, interactive elements, and multilingual web content. Similarly, design-to-code datasets, such as Pix2Code[8] and HTML/CSS generation benchmarks[41], lack real-world fidelity, often producing brittle code that fails to generalize across diverse website layouts. The challenge extends further to hand-drawn sketch interpretation, where limited datasets like Sketch2Code [26] fail to support diverse UI structures and real-world web variability. Moreover, existing benchmarks lack multilingual and cross-domain

*Co-first author

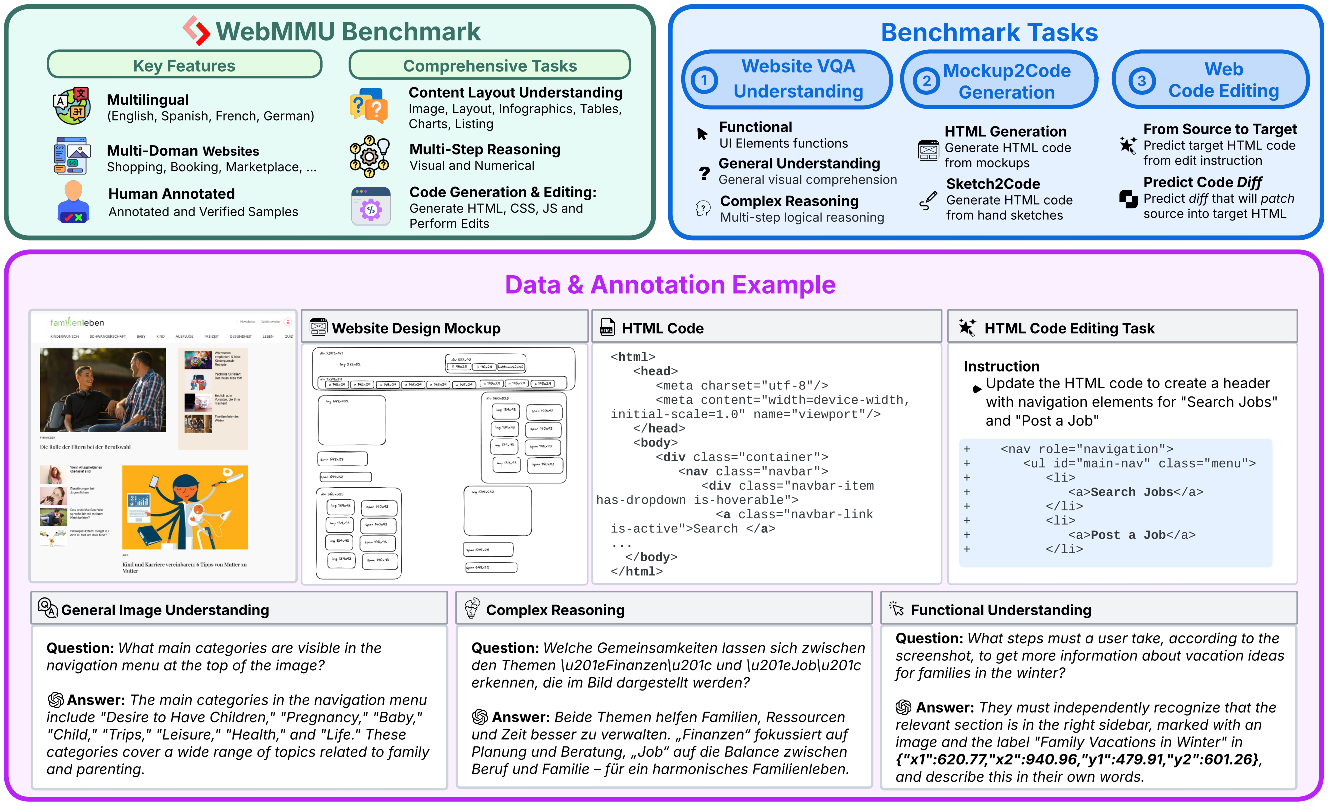


Figure 1. **WebMMU Benchmark Overview.** WebMMU is designed to evaluate AI models on diverse web-based tasks, including website-based VQA, multilingual understanding, sketch-based web development, and automated code generation from mockups. It challenges models to interpret complex website layouts, generate structured web code, and answer functional reasoning questions.

adaptability, restricting their applicability to non-English and domain-specific web scenarios. These limitations emphasize the need for a unified benchmark that integrates multiple web-related AI tasks, offering structured evaluation criteria, multimodal capabilities, and cross-lingual generalization to support both web development and expert-level web reasoning.

To address these challenges, we introduce **WebMMU** (Figure 1), a multimodal, **M**ultilingual, and **MU**lti-task benchmark for evaluating multimodal large language models (MLLMs) in understanding and generating content for the Visual **Web** across English, Spanish, German, and French. WebMMU encompasses three core tasks: **Website VQA (WebQA)**, which enables fine-grained evaluation of functional understanding, general visual comprehension, and multi-step reasoning through visual question-answer pairs; **Mockup2Code Generation**, which assesses design-to-code generation by aligning UI mockups and sketches with structured web layouts, covering both simple UI designs and complex layouts with nested elements; and **Web Code Editing**, which evaluates precise and context-aware code editing through user-requested HTML/CSS/JavaScript modifications, including feature additions, UI adjustments, and bug fixes. WebMMU spans a diverse range of web domains, including *shopping*, *booking*, *sports*, *technology*, and more, ensuring broad applicability across real-world web usage.

We benchmark state-of-the-art multimodal AI models across three core tasks, evaluating both open-source and closed-source models. Our results reveal significant challenges in action grounding and complex reasoning, structured layout understanding, and accurate web generation. While models (in particular, closed-source ones) exhibit strong general image understanding in WebQA, they struggle with complex reasoning, with most scoring below 50% and some as low as 2% (e.g., Fuyu-8B in English), alongside notable multilingual performance drops (Figure 2). In Web Code Editing, even top-performing models like Gemini-2.0-Flash and Claude-3.5-Sonnet outperform open-source counterparts yet still struggle with maintaining logical structure and syntactic correctness, highlighting the need for more structure-aware code-editing techniques, particularly for complex modifications. Similarly, in Mockup2Code, models such as OpenAI-o1 and Claude-3.5 achieve a high LLM-as-Judge score (4/5) on simple layouts but fail with nested element structures, revealing limitations in UI hierarchy comprehension. These findings emphasize the need for improved multimodal alignment, UI-aware modeling, and cross-lingual robustness to bridge the gap between vision-language models and real-world web interaction.

WebMMU sets a new standard for evaluating MLLMs in web reasoning, UI comprehension, and automated web generation, driving progress in web understanding and development. Our contributions are as follows:

- **Comprehensive Multi-Task Web Benchmark**: Unifying website VQA, web design-to-code generation, and code editing into a standardized evaluation framework.
- **Diverse and Multi-Lingual Web Pages**: Supporting various domains with multilingual interactions and structured annotations for functional and UI reasoning.
- **Rich Annotations for Web Understanding and Development**: Offering fine-grained question-answer pairs, modified HTML/CSS/JavaScript code for editing requests, and sketch annotations aligned with web UI layouts.
- **Benchmarking State-of-the-Art AI Models**: Evaluating leading multimodal AI models on web-based reasoning, code editing accuracy, and generalization to diverse web structures.

2. Related Work

Web Understanding and Agentic MLLMs. Multimodal learning has become central to web UI understanding, integrating visual, textual, and structural modalities to support both web comprehension and agentic navigation. Early work, such as Screen2Words [36], parsed web screenshots into UI elements, later influencing MLLM pretraining[25]. Recent advances leverage patching strategies[4], grounding[14], text-structural alignment[5, 38], and context-aware UI representations[24]. These innovations have expanded MLLM applications in web agents, enabling models to navigate and manipulate websites based on user instructions[14, 39, 43]. However, existing benchmarks often rely on limited artificial websites[16, 45] or focus solely on English data[10, 28, 42], lacking diversity and real-world complexity. WebMMU addresses these gaps by incorporating real-world websites and multilingual queries, requiring models to perform complex reasoning and UI grounding, making it a more comprehensive evaluation framework for MLLM-driven web understanding and navigation.

Visual Question Answering for Web. Progress in web-based VQA has been driven by benchmarks like WebSRC [12], WebQA [9], WebQuest [34], VisualWebBench [27], and WebWalkerQA [37] covering tasks such as captioning, webpage QA, and element grounding. Compared to traditional VQA on natural images [40], web-based VQA additionally requires understanding structured webpage layouts, the relationships between UI elements, and their functional roles within web environments. However, these existing benchmarks focus on narrow task sets, limiting generalization across diverse web scenarios. WebMMU addresses these gaps by spanning 20 domains in four languages and introducing fine-grained categories—action, multi-step reasoning, and general understanding questions for a more comprehensive evaluation on a browsing session or a single view screenshot.

Automatic Web Design and Development. Code generation and editing have been widely studied across programming languages, with benchmarks evaluating code generation [11, 23, 31, 32] and code editing based on natural language instructions [21, 33]. However, most previous studies focus on general-purpose programming, neglecting web design and development. To bridge this gap, [20, 41] explore generating HTML/CSS from web screenshots. In contrast, WebMMU introduces Web Code Editing, which involves multilingual tasks for modifying website visual and functional features based on user instructions, better reflecting real-world web development use cases. Additionally, WebMMU includes Mockup2Code. Unlike prior work [6, 22] that relies on simplistic and artificial sketches drawn by researchers, our sketches are extracted from real-world websites, preserving complex element hierarchies.

3. WebMMU Benchmark

We introduce WebMMU, an ongoing effort designed to evaluate AI on real-world Visual Web tasks that integrate text, images, and structured code. By unifying challenges such as visual question answering (WebQA), Web Code Editing, and Mockup2Code generation, WebMMU offers a holistic, multilingual testbed (English, Spanish, German, French) for web-based reasoning. In the remainder of this section, we describe WebMMU's task formulation, data collection, annotation process, and present an overview of benchmark tasks.

3.1. Data Collection and Annotation

Website Selection and Data Capture. To construct WebMMU, we curated a diverse set of webpage URLs from the FineWeb dataset [30] and applied domain-specific heuristics to ensure coverage across 20 popular, content-rich, and feature-rich web domains (e.g., shopping, booking, technology). We selected webpages in four languages—English, German, French, and Spanish—considering linguistic diversity, annotator availability, and budget constraints. To capture full browsing sessions on a single webpage, we generated collages combining multiple snapshots taken at different scroll depths and interaction states within the page. A viewport-specific snapshot was retained alongside relevant HTML and assets (e.g., CSS, JavaScript). Selection strictly adhered to web crawling policies (e.g., `robots.txt`).

Annotation Process. Annotators were provided with webpage screenshots, corresponding HTML, and asset files and were tasked with three objectives: (1) generating open-ended and multiple-choice questions that capture real-world usage, including highlighting, clicking, and multi-step reasoning; (2) creating UI mockups of varying complexity and formats to support design-to-code workflows; and (3) formulating code edit requests that require programming expertise. A structured training phase ensured annotation consistency and quality.

Quality Control and Annotator Demographics. A 100% quality assurance framework was implemented in three stages: *Trainer Review*, where experienced annotators performed initial annotations; *Primary QA (QA1)*, where independent specialists verified accuracy, completeness, and adherence to guidelines; and *Secondary QA (QA2)*, ensuring consistency with expert-level annotation criteria. The dataset was annotated by 127 professionals across North America, South America, Europe, Africa, and Asia, representing diverse linguistic and domain expertise. English annotators primarily came from Asia, German and French from Europe, and Spanish from Latin America. Annotators held qualifications ranging from bachelor's to advanced degrees for specialized tasks and were compensated above fair market wages, ensuring ethical labor practices and high-quality results.

3.2. Tasks Overview

WebMMU introduces a comprehensive task evaluation suite for visual web-based environments, integrating grounding, multi-step reasoning, and structured code generation and editing across three core tasks: WebQA, Mockup2Code, and Web Code Editing.

3.2.1. Web Question Answering (WebQA)

The WebQA task in WebMMU evaluates models' ability to extract, integrate, and ground structured UI elements, numerical data, and graphical components from web screenshots while reasoning over hierarchical layouts, predicting actions, and ensuring spatial grounding. It consists of three categories: **Agenctic Action**, which focuses on web navigation and action execution without feedback from the environment, requiring models to understand UI elements like buttons, menus, and hyperlinks, identify elements (e.g., *"Where can I find the coaching plans?"*), and execute actions (e.g., *"How can I save this drill?"*) while handling *spatial grounding* and distinguishing *static vs. interactive elements* across multilingual UIs; many of these tasks also require *coordinate-based reasoning* to localize UI components accurately. **Multi-step Reasoning** involves multi-step inference, numerical calculations, and comparisons across UI components (e.g., *"If a customer were to buy all the camera models mentioned on the bottom of this page in Ëxpert Camera Reviewsẗable, what would be the grand total?"*), requiring models to integrate text, numerical values, and layout structures from structured web content, where hierarchical reasoning is essential despite being constrained to single-frame snapshots; and **General Visual Comprehension**, which assesses a model's ability to extract and synthesize structured and unstructured data from web screenshots, including OCR-extracted text, images, graphical elements, and UI components (e.g., *"How many brand logos are in the Featured Brands section?"*), emphasizing *semantic comprehension* beyond standard OCR-based extraction.

	En	Es	De	Fr	Total
Website Images	392	133	130	131	**786**
WebQA	1476	484	379	456	**2795**
Mockup2Code	180	93	85	78	**436**
Web Code Editing	165	75	67	68	**375**
Total	**2213**	**785**	**661**	**733**	**4392**

Table 1. **Multilingual Statistics.** Language-wise dataset breakdown across tasks. We report the number of web images per language. English (En), Spanish (Es), German (De) and French (Fr).

3.2.2. Mockup2Code

The Mockup2Code task in WebMMU advances design-to-code by translating hand-drawn wireframes and high-fidelity digital mockups into structured code. Unlike text-based UI generation, it evaluates a model's ability to interpret spatial hierarchies and UI structures from visual inputs. The dataset includes low-fidelity sketches and digitally created mockups, challenging models to generalize across abstraction levels in web design while tackling component recognition, spatial alignment, and structured code synthesis. Unlike prior design-to-code datasets, WebMMU incorporates real-world web layouts, ensuring models generate syntactically correct, semantically meaningful code aligned with modern web development practices.

3.2.3. Web Code Editing

Web Code Editing is a novel task, which evaluates a model's ability to modify webpage code while preserving functional and structural integrity, given a screenshot, source code, and a user edit request. To perform well, models must complete three sub-tasks: (1) understand the provided inputs, including the webpage codebase, visual elements in the screenshot, and the requested modification; (2) identify the relevant code snippets that require modification; and (3) generate the appropriate HTML, CSS, or JavaScript edits to implement the requested change. These sub-tasks require an advanced understanding of webpage development and realistic code editing capabilities.

The modification requests span a broad range of visual and functional changes. Visual edits include adjusting font size and colors, repositioning elements, and adding headers

Task	Metric	Evaluation Details
WebQA	LLM-as-Judge	Measures accuracy; 0 (incorrect) / 1 (correct).
Mockup2Code	LLM-as-Judge	Assesses layout fidelity on a 1-5 scale (layout, spacing, grid).
Code Editing	BLEU [29], TreeBLEU [20]	Evaluates structural correctness by matching ground truth differences.
	LLM-as-Judge	Scores functional accuracy on a 1-5 scale (functional correctness).

Table 2. **Evaluation Metrics** used in WebMMU.

or footers. Functional modifications involve adding interactive components such as buttons or forms and enhancing user experience with dynamic UI elements. The task is multilingual, aligning with the broader scope of WebMMU. Given the length of webpage source code, models are prompted to output only the necessary code differences rather than rewriting the entire codebase. This improves both practicality and efficiency, ensuring that the generated edits remain concise and targeted. More details on the prompt formulation are provided in Appendix 8.2.

3.3. Dataset Statistics

WebMMU consists of 786 webpage images spanning domains such as e-commerce, education, news, and finance. It includes 2795 WebQA samples, 436 Mockup2Code instances, and 375 Web Code Editing cases. Unlike previous datasets that focus on predefined UI layouts, WebMMU captures full-page web snapshots, requiring models to reason over dynamic content, nested structures, and multimodal dependencies. The dataset supports English, Spanish, German, and French, ensuring linguistic diversity in web comprehension. Table 1 provides a detailed breakdown. As data collection progresses, we aim to expand coverage across languages, task complexity, and real-world web navigation.

4. Evaluation

We evaluate state-of-the-art multimodal AI models across both closed-source and open-source categories. Model inference for WebQA, Mockup2Code, and Web Code Editing follows standardized prompts (Appendix 8). Evaluation combines LLM-as-Judge [44] scoring with established automatic metrics, as summarized in Table 2. To ensure fair and structured assessments, each rating criterion is explicitly defined for each task, preventing subjective biases in scoring.

LLM-as-Judge is used to evaluate WebQA, where model responses receive binary correctness scores (0 or 1) based on predefined criteria for semantic accuracy and reasoning completeness (Appendix 8.5). This structured approach ensures consistency and prevents arbitrary grading. Since automated metrics fail to capture layout fidelity in Mockup2Code, evaluation instead relies on LLM-as-Judge, assessing the alignment between input sketches and rendered outputs across three key dimensions: *layout structure, spacing, and grid consistency* (Appendix 12). Each aspect follows well-defined scoring guidelines, ensuring reproducible and fair assessments. For Web Code Editing, we evaluate both *structural correctness* and *functional accuracy*. The former is measured using BLEU [29] and TreeBLEU [20], ensuring syntactic validity and adherence to coding conventions. The latter relies on LLM-as-Judge, where functional equivalence between reference and predicted edits is rated on a 1-5 scale. To avoid arbitrary scoring, rating criteria explicitly define correctness levels based on functional preservation and intended user modifications. Since web functionalities can be implemented in multiple ways, the evaluation accounts for semantically valid alternatives, preventing undue penalization of syntactically different but functionally correct edits. For all LLM-as-Judge evaluations, we use GPT4o-1120, which has demonstrated strong alignment with human judgment and diverse scoring behavior [18], ensuring robustness across tasks.

5. Results

In this section we present results of state-of-the-art (SOTA) models on our proposed WebMMU benchmark, including results on Web Visiual Question Answering (WebQA) Mockup2Code Generation and Web Code Editing. Main results are presented in Table 3, and Figures 2-4.

5.1. WebQA Performance

Table 3 presents model accuracy (%) across three question types—*Multi-step Reasoning* ( ), *Agenctic Action* ( ), and *General Visual Comprehension* ( )—evaluated in four languages. Closed-source models, such as Gemini 2.0 Flash and Claude 3.5 Sonnet, outperform open-source alternatives across all tasks but still struggle with agentic action, particularly in predicting spatial coordinates for interactive elements. Among open-source models, larger architectures (>30B parameters) like Qwen2VL-72B and Internvl2.5-38B perform better in general image understanding and UI recognition, while smaller models (<8B) exhibit poor generalization across tasks.

Performance varies significantly by question type. General image understanding is the easiest, as it primarily involves visual recognition without complex reasoning. In contrast, complex reasoning remains difficult, with most models scoring below 50.0% and some as low as 2.0% (e.g., Fuyu-8b in English), indicating challenges in retrieving and reasoning over structured webpage content. **Agentic action is the most difficult, with even the strongest models rarely exceeding 10% accuracy**. This task requires precise spatial grounding—models must not only recognize interactive elements (e.g., “About Me” in a menu bar) but also predict **approximate bounding box coordinates**. While many models detect interactive components, they struggle with precise local-

Model	English			French			German			Spanish		
Gemini2.0 Flash	44.3	1.2	59.2	41.6	9.0	**52.8**	18.2	12.8	29.1	46.1	12.0	36.1
Claude3.5 Sonnet	**51.4**	**3.7**	**64.1**	**53.0**	**12.7**	51.2	**26.9**	**15.6**	**31.6**	**63.8**	**15.9**	**41.9**
Phi3.5-VI-4b [1]	8.90	1.80	31.60	2.20	6.90	39.00	8.40	13.00	23.90	3.00	10.20	32.00
UI-Tars-7b	**19.30**	**8.10**	47.60	7.70	8.90	47.60	7.80	14.30	28.40	**20.90**	**14.00**	**38.80**
Molmo-7b [15]	12.30	3.80	32.90	7.00	7.50	47.60	8.30	13.70	**31.90**	15.10	10.30	32.00
Qwen2VL-7B [35]	18.00	2.90	**57.10**	10.10	10.20	**52.00**	**10.70**	**17.60**	26.30	19.30	14.00	36.50
Fuyu-8b [7]	1.60	0.40	14.30	0.00	1.30	17.50	1.00	5.60	15.70	0.70	1.50	10.90
Internvl2.5-8b [13]	16.30	1.90	46.30	**11.00**	**13.30**	40.00	7.40	16.00	25.90	13.80	11.90	31.10
Glm4V-9b [19]	15.30	8.10	41.80	11.40	**13.90**	48.10	14.70	13.80	25.00	21.60	13.40	35.60
Llama-3.2-11B-Vision [17]	**27.10**	7.90	**53.20**	11.60	11.30	48.10	11.80	14.30	**33.60**	17.50	11.80	37.90
Pixtral-12b [2]	27.10	**9.20**	44.90	**17.70**	11.30	**53.40**	**19.50**	**19.30**	21.70	**28.70**	**17.80**	**40.20**
Internvl2.5-38b [13]	22.90	3.80	**59.30**	**20.90**	**15.30**	**65.70**	**18.00**	**20.10**	**39.70**	**36.20**	**14.90**	**41.40**
Qwen2VL-72B [35]	**23.60**	**4.30**	53.70	16.90	13.90	54.50	15.30	17.50	36.20	29.10	12.70	41.00

Table 3. **Performance on Web VQA.** Model accuracy (%) by question type across four languages. : Multi-step Reasoning, : Agenctic Action, : General Visual Comprehension. The best model within each size category is highlighted in **bold**, and the runner up is underlined. Models are categorized by size: **blue** indicates models with fewer than 8B parameters, **orange** represents models with 8B to 12B parameters, **green** denotes models with more than 12B parameters, and **gray** marks proprietary models.

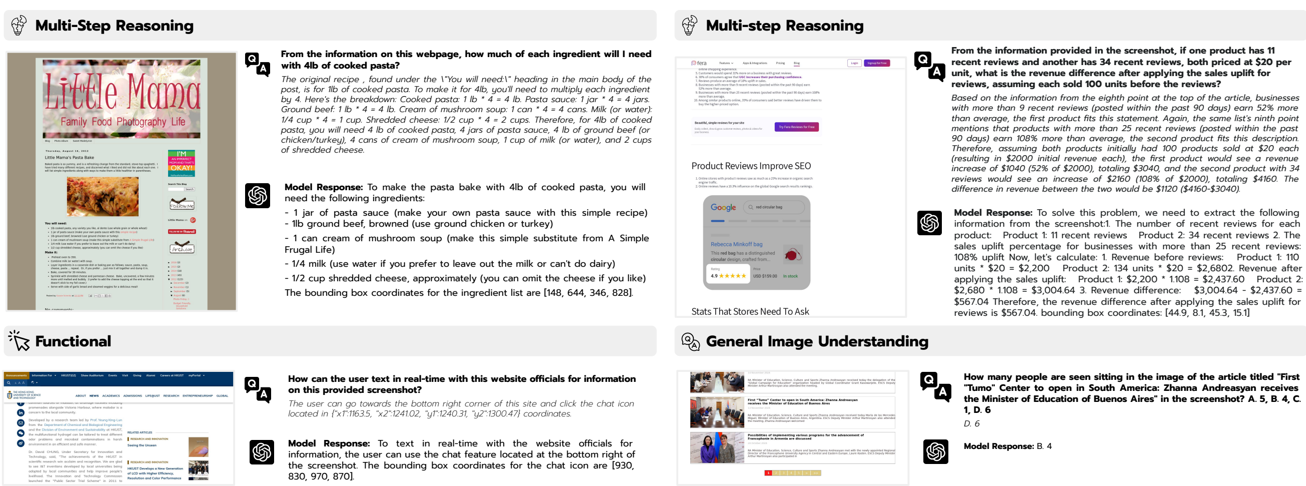


Figure 2. **Failure Cases in Web Visual Question Answering (WebQA)** for the top-performing open-source model (InternVL-38B).

ization, leading to low scores.**Error Analysis.** Figure 2 highlights common failure patterns. Models frequently **miscalculate numerical values or fail to integrate relevant information in multi-step reasoning**. In functional understanding, inaccurate bounding box predictions explain the poor performance in agentic action tasks. Additionally, multilingual generalization remains a challenge despite the dataset covering resource-rich languages. These findings underscore the need for improved spatial reasoning, numerical comprehension, and cross-lingual adaptation to bridge the gap between vision-language models and real-world web interaction.

5.2. Mockup2Code Generation

Figure 3 evaluates the Mockup2Code task, reporting scores for each dimension and overall performance. Open MLLMs such as Phi3.5 VI, Fuyu-8B, and GLM4V-9B generally perform poorly across all metrics. Notably, Phi3.5 and Fuyu-8B score nearly 1 across all dimensions, indicating a complete failure on this task. Nevertheless, performance improves with model scale. For instance, Qwen2VL's score rises from 1.90 to 3.39 when scaling from 7B to 72B, while InternVL

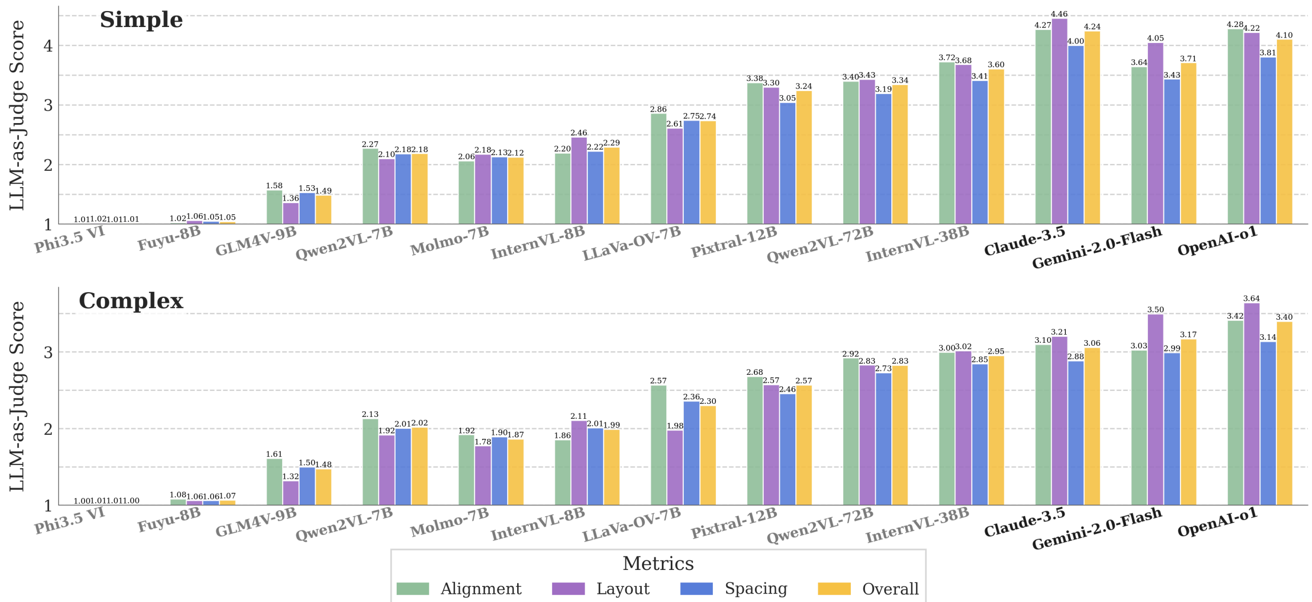


Figure 3. **Mockup2Code Performance.** LLM-as-Judge evaluation scores for simple and complex UI mockups across three key dimensions: *alignment, layout, and spacing*, along with overall performance. Higher scores indicate better fidelity between the generated and reference web designs. Closed-source models outperform open-source alternatives, particularly in complex cases, yet challenges remain in precise layout reproduction.

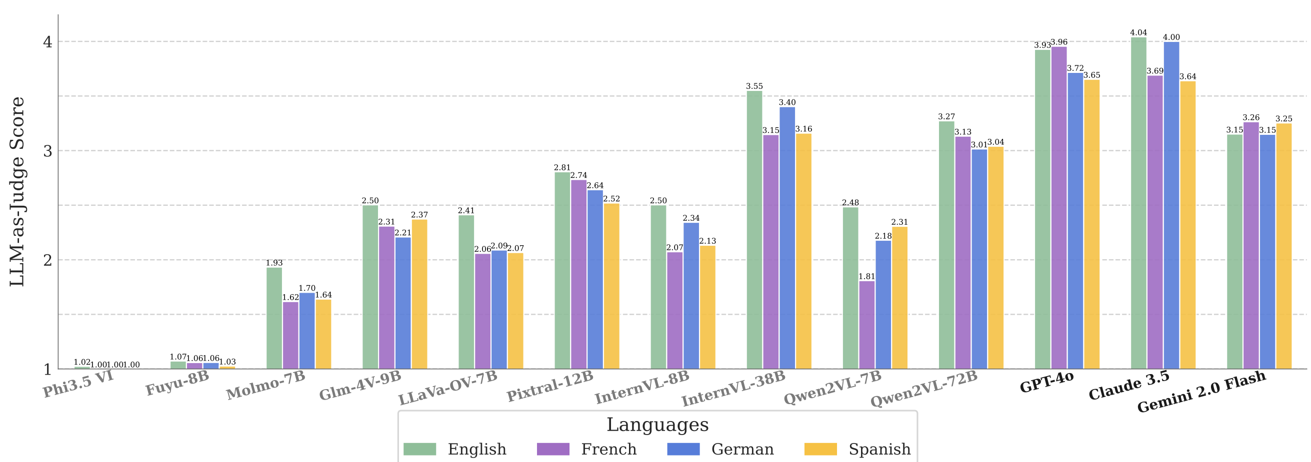


Figure 4. **Performance on Code Edits.** LLM-as-Judge metric, on a scale of 1-5, used to evaluate functional correctness of code edits. All models, including closed-source models, struggle with the Web Code Editing task of WebMMU. Refer to Table 7 for full results, including BLEU and TreeBLEU scores, of all models.

improves from 2.34 to 3.61 when scaling from 8B to 38B. Additionally, Pixtral-12B outperforms all 7B/8B models. Still, even the best open MLLMs struggle, especially with complex designs—InternVL-38B, the highest performer, scores only 2.98 out of 5.In contrast, proprietary models like Claude-3.5, Gemini-2.0-Flash, and OpenAI-o1 perform significantly better, particularly on simple UI designs, where they achieve LLM-as-Judge scores above 4. However, their performance declines in complex variants, with top scores reaching only 3.4 out of 5. Across all evaluation dimensions, both proprietary and large-scale open VLMs struggle most with spacing, which requires accurately setting element dimensions and margins based on sketch input. **Case Analysis.** Figure 5 illustrates both successful and failed cases with the best-performing model OpenAI-o1on Mockup2Code. As seen, OpenAI-o1handles simple, flat layouts well, even when

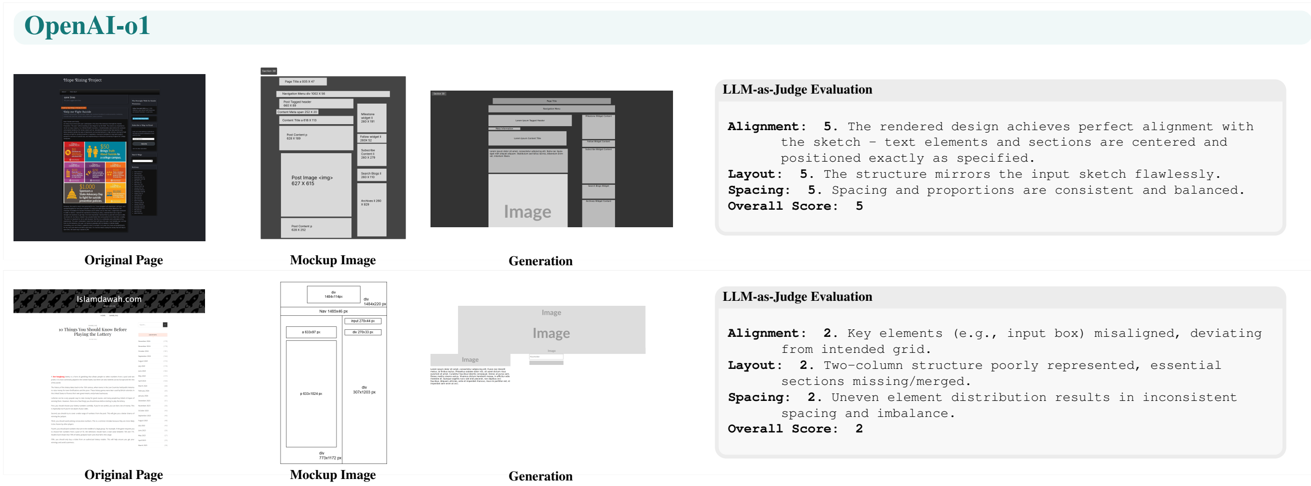


Figure 5. **Success (Top) and Failure (Bottom) Cases for Mockup2Code Generation from OpenAI-o1.**

the number and variety of elements are moderate. However, it struggles with nested structures, failing to replicate element hierarchy and spacing accurately. Concretely, OpenAI-o1's generation is significantly misaligned, failing to preserve the basic structure when <p> and <a> elements are embeded within <div> tags. Similar issues appear in other samples and with different models, as shown in Figures 14 and 16.

5.3. Code Editing Performance

Figure 4 presents results for Web Code Editing, evaluated using LLM-as-Judge (all metrics in Table 7). While proprietary models achieve the highest functional accuracy, their advantage over large open-source models is relatively small, **indicating that both struggle with preserving functional correctness while ensuring syntactic consistency**. Smaller models, such as Phi3.5-VI and Fuyu-8b, perform the worst, often failing to generate valid code (LLM-as-Judge score <1.5). Performance improves with model size, with Qwen2VL-72B and InternVL-38B achieving results competitive with closed-source models. However, even the strongest models exhibit notable limitations in producing structurally correct edits that fully preserve intended functionality. Multilingual performance remains stable* for high-performing models, while smaller ones show greater variance, highlighting challenges in adapting code modifications across different languages.Notably, a key limitation across all models (in particular, open-source) is the failure to **automatically generate valid patch files** for seamless integration into the source code. Despite having access to full source files, no model successfully produced patch file contents that could be directly applied without manual intervention. As a result, human oversight remains essential, underscoring a fundamental challenge in automating web code edits effectively.

6. Discussion and Conclusion

While WebMMU provides a comprehensive evaluation of web-based AI reasoning and code generation, it has several limitations. First, it is restricted to single-screenshot web reasoning, capturing static snapshots rather than supporting interactive environments or multi-turn navigation. Although multi-step reasoning tasks are included, they rely solely on single-page information, limiting evaluation in dynamic web exploration. Second, linguistic coverage is constrained to four languages—English, French, German, and Spanish—due to annotator availability, which may limit generalization to underrepresented languages and regional web structures. Finally, while Mockup2Code and Web Code Editing cover core web technologies such as HTML, CSS, and JavaScript, modern frontend frameworks like React, Angular, and Vue.js are not explicitly evaluated. Future extensions could incorporate these frameworks to better reflect real-world development practices.

We introduce WebMMU, a comprehensive benchmark for multimodal and multilingual website understanding and generation, spanning WebQA,Web Code Editing, and Mockup2Codetranslation. Unlike prior benchmarks, WebMMU evaluates models in a structured web environment that requires reasoning over complex layouts, functional elements, and cross-lingual content. Our results reveal fundamental challenges: Web VQA models struggle with UI comprehension and multilingual generalization, code editing models fail to maintain structured logic beyond syntactic correctness, and UI generation models exhibit a trade-off between spatial precision and semantic fidelity. Despite progress in visual and structural understanding, hierarchical reasoning, functional accuracy, and real-world UI adaptation remain open problems. These findings underscore the need for better multimodal alignment, UI-aware architectures, and cross-lingual adaptation to advance web agents.

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 6

[2] Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khandelwal, Timothée Lacroix, Guillaume Lample, Diego Las Casas, Thibaut Lavril, Teven Le Scao, Andy Lo, William Marshall, Louis Martin, Arthur Mensch, Pavankumar Muddireddy, Valera Nemychnikova, Marie Pellat, Patrick Von Platen, Nikhil Raghuraman, Baptiste Rozière, Alexandre Sablayrolles, Lucile Saulnier, Romain Sauvestre, Wendy Shang, Roman Soletskyi, Lawrence Stewart, Pierre Stock, Joachim Studnia, Sandeep Subramanian, Sagar Vaze, Thomas Wang, and Sophia Yang. Pixtral 12b, 2024. 6

[3] AI Anthropic. Claude 3.5 sonnet model card addendum. *Claude-3.5 Model Card*, 3(6), 2024. 1

[4] Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. Screenai: A vision-language model for ui and infographics understanding. *arXiv preprint arXiv:2402.04615*, 2024. 3

[5] Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, et al. Uibert: Learning generic multimodal representations for ui understanding. *arXiv preprint arXiv:2107.13731*, 2021. 3

[6] Somoy Subandhu Barua, Imam Mohammad Zulkarnain, Abhishek Roy, Md Golam Rabiul Alam, and Md Zia Uddin. Sketch2fullstack: Generating skeleton code of full stack website and application from sketch using deep learning and computer vision. *arXiv preprint arXiv:2211.14607*, 2022. 3

[7] Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. Introducing our multimodal models, 2023. 6

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

[9] Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16495–16504, 2022. 1, 3

[10] Dongping Chen, Yue Huang, Siyuan Wu, Jingyu Tang, Liuyi Chen, Yilin Bai, Zhigang He, Chenlong Wang, Huichi Zhou, Yiqiang Li, et al. Gui-world: A dataset for gui-oriented multimodal llm-based agents. *arXiv preprint arXiv:2406.10819*, 2024. 3

[11] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. 3

[12] Xingyu Chen, Zihan Zhao, Lu Chen, Jiabao Ji, Danyang Zhang, Ao Luo, Yuxuan Xiong, and Kai Yu. Websrc: A dataset for web-based structural reading comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4173–4185, 2021. 1, 3

[13] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 6

[14] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*, 2024. 3

[15] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, Yen-Sung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Jen Dumas, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna,

Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024. 6

[16] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[17] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 6

[18] Aarash Feizi, Sai Rajeswar, Adriana Romero-Soriano, Reihaneh Rabbany, Spandana Gella, Valentina Zantedeschi, and João Monteiro. Pairbench: A systematic framework for selecting reliable judge vlms. *arXiv preprint arXiv:2502.15210*, 2025. 5

[19] Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Jingyu Sun, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024. 6

[20] Yi Gui, Zhen Li, Yao Wan, Yemin Shi, Hongyu Zhang, Yi Su, Shaoling Dong, Xing Zhou, and Wenbin Jiang. Vision2ui: A real-world dataset with layout for code generation from ui designs. *arXiv preprint arXiv:2404.06369*, 2024. 3, 5

[21] Jiawei Guo, Ziming Li, Xueling Liu, Kaijing Ma, Tianyu Zheng, Zhouliang Yu, Ding Pan, Yizhi Li, Ruibo Liu, Yue Wang, et al. Codeeditorbench: Evaluating code editing capability of large language models. *arXiv preprint arXiv:2404.03543*, 2024. 3

[22] Vanita Jain, Piyush Agrawal, Subham Banga, Rishabh Kapoor, and Shashwat Gulyani. Sketch2code: transformation of sketches to ui in real-time using deep neural network. *arXiv preprint arXiv:1910.08930*, 2019. 1, 3

[23] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. Swe-bench: Can language models resolve real-world github issues? In *ICLR*, 2024. 3

[24] Jihyung Kil, Chan Hee Song, Boyuan Zheng, Xiang Deng, Yu Su, and Wei-Lun Chao. Dual-view visual contextualization for web navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14445–14454, 2024. 3

[25] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023. 3

[26] Ryan Li, Yanzhe Zhang, and Diyi Yang. Sketch2code: Evaluating vision-language models for interactive web design prototyping. *arXiv preprint arXiv:2410.16232*, 2024. 1

[27] Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. Visualwebbench: How far have multimodal llms evolved in web page understanding and grounding? *arXiv preprint arXiv:2404.05955*, 2024. 1, 3

[28] Xing Han Lù, Zdeněk Kasner, and Siva Reddy. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*, 2024. 1, 3

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 5

[30] Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. 3

[31] Juan Rodriguez, Xiangru Jian, Siba Smarak Panigrahi, Tianyu Zhang, Aarash Feizi, Abhay Puri, Akshay Kalkunte, François Savard, Ahmed Masry, Shravan Nayak, Rabiul Awal, Mahsa Massoud, Amirhossein Abaskohi, Zichao Li, Suyuchen Wang, Pierre-André Noël, Mats Leon Richter, Saverio Vadacchino, Shubbam Agarwal, Sanket Biswas, Sara Shanian, Ying Zhang, Noah Bolger, Kurt MacDonald, Simon Fauvel, Sathwik Tejaswi, Srinivas Sunkara, Joao Monteiro, Krishnamurthy DJ Dvijotham, Torsten Scholak, Nicolas Chapados, Sepideh Kharagani, Sean Hughes, M. Özsu, Siva Reddy, Marco Pedersoli, Yoshua Bengio, Christopher Pal, Issam Laradji, Spandanna Gella, Perouz Taslakian, David Vazquez, and Sai Rajeswar. Bigdocs: An open and permissively-licensed dataset for training multimodal models on document and code tasks. *arXiv preprint arXiv:2412.04626*, 2024. 3

[32] Juan A. Rodriguez, Abhay Puri, Shubham Agarwal, Issam H. Laradji, Pau Rodriguez, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. Starvector: Generating scalable vector graphics code from images and text, 2024. 3

[33] Runchu Tian, Yining Ye, Yujia Qin, Xin Cong, Yankai Lin, Yinxu Pan, Yesai Wu, Haotian Hui, Weichuan Liu, Zhiyuan Liu, et al. Debugbench: Evaluating debugging capability of large language models. *arXiv preprint arXiv:2401.04621*, 2024. 3

[34] Maria Wang, Srinivas Sunkara, Gilles Baechler, Jason Lin, Yun Zhu, Fedir Zubach, Lei Shu, and Jindong Chen. Webquest: A benchmark for multimodal qa on web page sequences. *arXiv preprint arXiv:2409.13711*, 2024. 3

[35] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6

[36] Jason Wu, Xiaoyi Zhang, Jeff Nichols, and Jeffrey P Bigham. Screen parsing: Towards reverse engineering of ui models from screenshots. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 470–483, 2021. 3

[37] Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Deyu Zhou, Pengjun Xie, and Fei Huang. Webwalker: Benchmarking llms in web traversal. *arXiv preprint arXiv:2501.07572*, 2025. 3

[38] Linhui Xu, Yawen Zhang, Yawen Li, Yawen Zhang, and Yawen Li. Hierarchical multimodal fine-grained modulation for visual grounding. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1–9. ACM, 2024. 3

[39] Ori Yoran, Samuel Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. Assistantbench: Can web agents solve realistic and time-consuming tasks? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8938–8968, 2024. 3

[40] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024. 3

[41] Sukmin Yun, Haokun Lin, Rusiru Thushara, Mohammad Qazim Bhat, Yongxin Wang, Zutao Jiang, Mingkai Deng, Jinhong Wang, Tianhua Tao, Junbo Li, et al. Web2code: A large-scale webpage-to-code dataset and evaluation framework for multimodal llms. *arXiv preprint arXiv:2406.20098*, 2024. 1, 3

[42] Ziniu Zhang, Shulin Tian, Liangyu Chen, and Ziwei Liu. Mmina: Benchmarking multihop multimodal internet agents. *arXiv preprint arXiv:2404.09992*, 2024. 3

[43] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. In *Forty-first International Conference on Machine Learning*. 3

[44] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023. 5

[45] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023. 1, 3

WebMMU: A Benchmark for Multimodal Multilingual Website Understanding and Code Generation

Supplementary Material

7. Task Samples

The Table 4, 5, and 6 present representative examples from the WebMMU dataset, covering VQA, Sketch2HTML, and Code Edition tasks. The VQA task (Table 4) evaluates a model's ability to interact with webpage elements, recognize visual content, and perform complex reasoning based on structured UI components. The Sketch2HTML task (Table 5) illustrates how webpage screenshots are converted into structured HTML representations, distinguishing between basic layout sketches and detailed UI component mappings. The Code Edition task (Table 6) demonstrates automated HTML modifications, providing before-and-after visual transformations based on functional and design-driven prompts. These task samples comprehensively showcase the challenges in webpage understanding, layout structuring, and automated UI refinement within the WebMMU benchmark.

8. Prompt Formulation

This section provides details on the prompt formulations used throughout this work. These prompts guide the multimodal large language models in generating and evaluating responses across different tasks. The prompts are categorized based on their usage, including code modification, VQA evaluation, and UX scoring.

8.1. VQA Generation Prompt

VQA Generation Prompt: This prompt directs the model to analyze a webpage screenshot and answer visual questions about its content or structure. The prompt template can be seen in Figure 6.

8.2. Code Edition Generation Prompt

Code Edition Generation Prompt: This prompt guides a model in modifying the source code based on a modification instruction given by the user. The model outputs changes using the `git diff` format, highlighting additions and deletions with '+'s and '-'s respectively. This ensures clear and structured documentation of code edits. The prompt template can be seen in Figure 7.

8.3. VQA Judge Prompts

VQA Judge Prompts: These prompts are used for evaluating model responses in VQA tasks. The model rates answers as 1 (Correct and Complete) or 0 (Incorrect or Irrelevant) based on factual accuracy and completeness. Example cases are provided to guide the evaluation. The prompt template can be seen in Figure ??.

8.4. UX Score Prompt

UX Score Prompt: This prompt is used for evaluating the UX score in the HTML2Sketch task. It assesses how well a webpage layout, generated from HTML, aligns with a given design sketch. Ratings range from 1 (Poor Match) to 5 (Excellent Match), based on criteria such as layout accuracy, alignment, and typography. The model provides a structured analysis before assigning a score. The prompt template can be seen in Figure 12.

8.5. LLM-as-Judge Prompts

Code Edit Judge Prompts Used for evaluating model responses in Code edition tasks. The model rates answers as 1-5 (5 refers to the most correct and complete, and 0 refers to incorrect or Irrelevant) based on factual accuracy and completeness. Example cases guide the evaluation. The prompt template can be seen in Figure ??.

Web-screenshot Analysis

Analyze the website screenshot and provide a detailed answer to the question. If the question involves locating or interacting with specific elements on the screen, include the bounding box coordinates [x_min, y_min, x_max, y_max] in your response.

Figure 6. VQA Generation Prompt for model inputs

Web Code Editing Generation Prompt

You are an expert web developer specializing in identifying and applying modifications to web code. You will receive a website's screenshot and a combination of it's HTML, CSS, and/or JavaScript code, formatted as follows:

- **HTML Code:** `html_code`
- **CSS Code:** `css_code`
- **JavaScript Code:** `javascript_code`

You will also receive a modification prompt describing the required changes. Your task is to produce the necessary code modifications using 'git diff' format, even if some or all sections are missing. Follow these guidelines:

1. **Input code:** `<input_code>`
2. **Modification Prompt:** `<edit_prompt>`
3. **Output Diff:**
   - Use '+' for additions and '-' for deletions.
   - Modify only the relevant parts while preserving structure.
   - In case the code is missing, generate the necessary block of code from scratch.
   - Ensure readability and correctness in the modifications.

Only output the necessary diff; do not repeat the input code.

Figure 7. Code edition generation prompt

UX Generation Prompt

You are an expert website developer. Analyze the provided webpage sketch and generate a single, fully structured HTML file with embedded CSS that accurately reflects the design.

The output must be a self-contained HTML document with internal `<style>` tags for CSS. Ensure all elements are structured exactly as seen in the sketch—**no extra elements, no missing elements**.

HTML Requirements:
- **Components:** Include all necessary components such as headers, paragraphs, buttons, forms, and images, maintaining the correct hierarchy and placement.
- **Images:** Use images generated from `https://placehold.co/` with exact dimensions matching the sketch, a neutral background color, and centered "Image" text. For example:

```
<img src="https://placehold.co/300x200?text=Image&bg=cccccc" alt="Placeholder Image">
```

- **Placeholder Text:** Use Lorem Ipsum for placeholder text where needed.

CSS Requirements:
- Implement CSS directly within the HTML file (inside a `<style>` block) to match the sketch, covering spacing, font sizes, colors, alignments, and element positioning.
- Use CSS Grid or Flexbox where appropriate to replicate the exact design layout.
- Apply styling for readability and interactive elements (e.g., fonts, button appearance).
- Ensure placeholder images maintain proper dimensions and design consistency.

Code Output:
- Provide a single, complete HTML file with internal CSS (do not separate them into different files).
- Do not include explanations, comments, or any extra formatting outside the code itself.

Figure 8. UX Generation Prompt: It takes input sketch and outputs HTML/CSS code of the given input

Web-screenshot Analysis

```
 examples = [
   {
     "INPUT": {
       "question": "What is the capital of France?",
       "model_answer": "Paris",
       "ground_truth": "Paris",
     },
     "OUTPUT": {
       "rating": 1,
       "rationale": "The model's answer matches the reference answer exactly."
     }
   },
   {
     "INPUT": {
       "question": "What is in the left of the image?",
       "model_answer": "A bus is in the left of the image.",
       "ground_truth": "A dog is in the left of the image.",
     },
     "OUTPUT": {
       "rating": 0,
       "rationale": "The model's answer is incorrect because the reference answer is 'A
dog'."
     }
   },
   {
     "INPUT": {
       "question": "Where is the burger on the table? Tell me the coordinates.",
       "model_answer": "The burger is on the table.",
       "ground_truth": "The burger is on the table at (50, 10, 150, 60).",
     },
     "OUTPUT": {
       "rating": 0,
       "rationale": "The predicted answer is incomplete because it does not provide the
coordinates as requested in the question."
     }
   }
]
test_case = {
   "INPUT": {
     "question": question,
     "model_answer": model_answer,
     "ground_truth": ground_truth
   }
}
```

You are evaluating a Visual Question Answering (VQA) system's response. Compare the model's answer with the ground truth and rate its accuracy.
Rating Scale (1 or 0): 1 - **Correct and Complete**: - The predicted answer fully matches the ground truth. - No factual errors or missing details. - Addresses the question with the correct level of specificity.
0 - **Incorrect or Irrelevant**: - Any factual errors or mismatches with the reference answer. - Does not address the question properly. - Provides misleading or irrelevant information.
Examples for reference: json.dumps(examples, indent=4)
Question, Model Answer, and Ground Truth: json.dumps(test_case, indent=4)
You must provide your evaluation in the following JSON format (without any extra text): json.dumps("rating": 0 or 1, "rationale": "[Brief explanation of why this rating was chosen]") """

Figure 9. LLM-as-judge prompt for VQA task using few shot examples

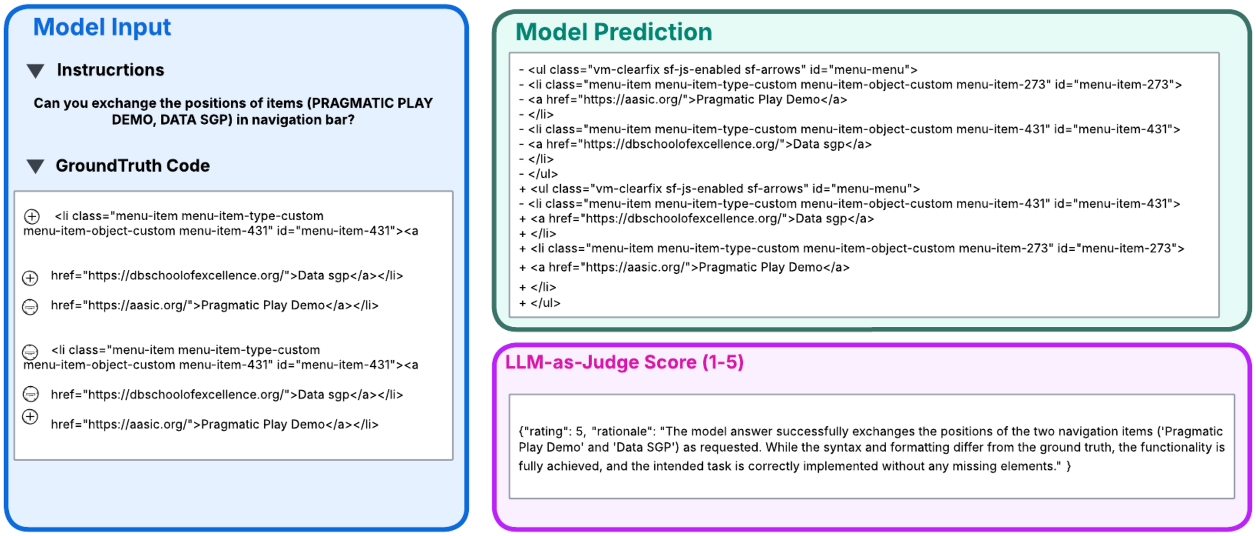


Figure 10. Code edit case study: **Success (Claude-3.5)**. The generated code accurately addresses the instructions given as input.

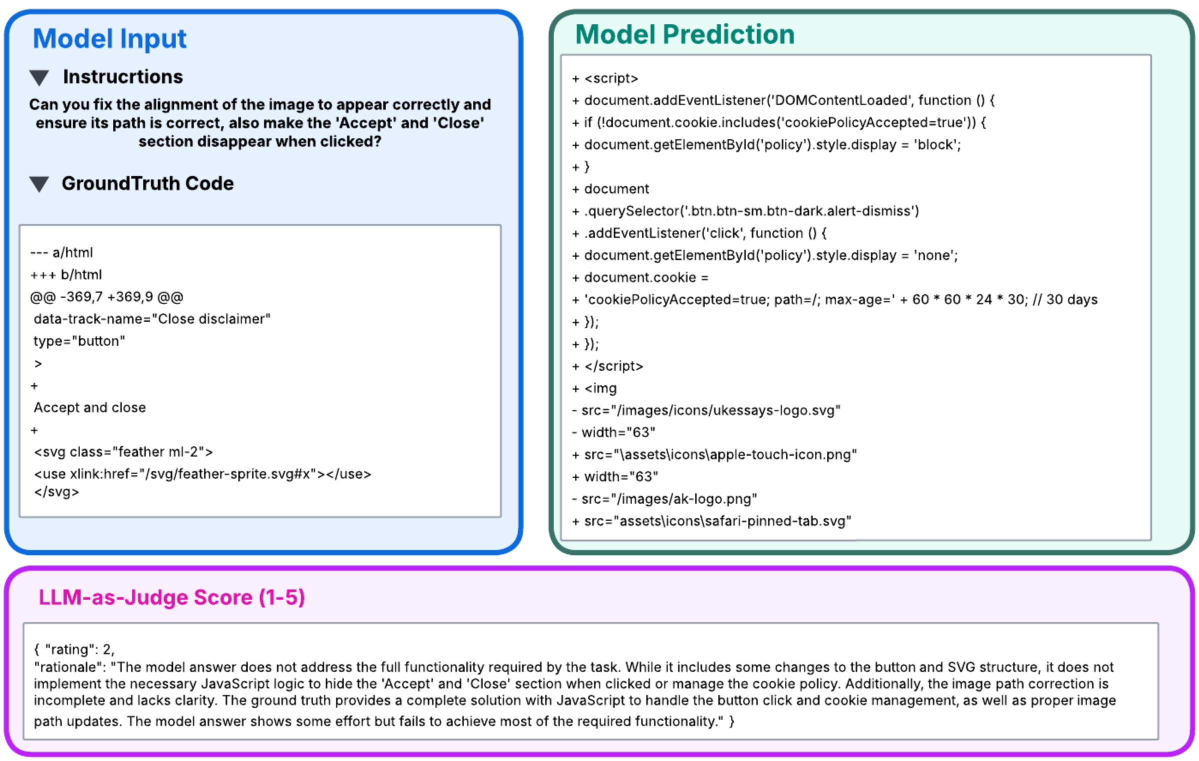


Figure 11. Code edit case study: **Failure (Gemini-2.0-Flash)**. The generated answer by the model skips main modifications requeseted by the user.

UX Evaluation Prompt

Task Overview:
Your task is to evaluate the accuracy of an AI-generated website by comparing it against a provided input sketch. The AI-generated website is provided as an image rendering of the HTML/CSS output. Your goal is to assess how well this rendered image replicates the intended layout from the sketch.

Provided Inputs:
You will receive two images:
1. **Input Sketch** – A wireframe illustrating the intended layout.
2. **Predicted AI-Rendered Website Image** – A screenshot of the website generated from AI-created HTML/CSS based on the sketch.
Since the AI-generated website is provided as an image, your evaluation must be based entirely on visual accuracy, disregarding the underlying code implementation.

Step 1: Detailed Description of Both Images
For each image (**Input Sketch** and **AI-Rendered Website**), provide a highly-detailed breakdown based on the following categories. Ensure that descriptions follow the same format for both images to facilitate a precise comparison.

1. Identify All Structural Sections:
Describe in detail the overall structure of the webpage layout, covering the following:
- **Header** – Does it contain a logo, navigation menu, search bar, or other elements?
- **Navigation Bar** – Describe the menu items. How many items are there? Is the navigation horizontal or vertical?
- **Main Content Area** – Identify distinct sections such as hero banners, text areas, images, or interactive components.
- **Sidebars (if applicable)** – Is there a sidebar for additional navigation, filters, or widgets?
- **Footer** – What content is present (e.g., links, social icons, contact information)?

For the AI-rendered website, note any differences compared to the sketch (e.g., missing sections, extra sections, missing items, misplaced content).

2. List and Describe All Elements:
List all key elements present in the **Input Sketch** and **AI-Rendered Website**:
- **Text Elements** – Titles, paragraphs, labels, lists, captions.
- **Images & Icons** – Identify all image placeholders and their intended placement.
- **Buttons & Links** – Describe all interactive elements like CTAs, navigation links, or form buttons.
- **Forms & Inputs** – Search bars, text fields, dropdowns, checkboxes, radio buttons, etc.
- **Tables & Lists** – If present, describe their structure and formatting.

For the AI-rendered website, specify any elements that are missing, added, or incorrectly placed.

3. Layout & Positioning Details:
Describe and analyze the spatial arrangement of elements in both images:
- **Column Structure** – Is the design single-column, multi-column, or grid-based?
- **Alignment** – Are elements aligned left, center, right, or justified?
- **Spacing & Proportions** – Are elements evenly spaced? Are margins, padding, and gaps consistent?
- **Relative Proportions** – Are certain sections (e.g., hero banners, sidebars) larger than others?

For the AI-rendered website, describe any deviations from the sketch (e.g., elements' size differences, elements too large/small, uneven spacing, misalignments).

Step 2: Evaluation of the AI-Rendered Website
After describing both images, evaluate the AI-generated website's accuracy using the following criteria. Assign a score from 1 to 5 for each.

1. Layout Structure Accuracy (1-5):
Does the generated HTML structure strictly follow the wireframe in layout, hierarchy, and element grouping? This includes the correct placement, nesting, and semantic usage of standard structural elements: `<header>`, `<nav>`, `<main>`, `<section>`, `<aside>`, `<article>`, `<footer>`, `<div>`, and content containers like `<img>`, `<p>`.
- **5** → 100% match. All elements are correctly placed, properly nested, fully grouped, and semantically accurate. No missing, misplaced, or extra elements.
- **4** → Mostly accurate, but minor structural inconsistencies exist (e.g., an unnecessary wrapper, slightly misplaced section, or minor redundancy). No missing elements.
- **3** → Some structural errors — at least one missing or misused element, multiple misplaced sections, or noticeable grouping issues.
- **2** → Major deviations — multiple missing, misplaced, or incorrectly nested elements, affecting hierarchy and readability.
- **1** → Severe structural failure — multiple core sections are absent or completely misstructured, making the output unrecognizable compared to the wireframe.

2. Spacing & Proportions (1-5):
Do margins, paddings, and element dimensions (e.g., `width`, `height`, `max-width`, `min-width`, `max-height`, `min-height`, gap for flex/grid layouts) precisely match the wireframe?
- **5** → 100% correct. All elements have precise margins, paddings, widths, heights, and spacing. No deviations.
- **4** → Minor inconsistencies exist (e.g., slightly incorrect padding/margin values or minor width/height variations).
- **3** → Noticeable discrepancies — some elements are too large, too small, or unevenly spaced, affecting visual balance.
- **2** → Significant spacing issues — multiple elements have incorrect dimensions, margins, or paddings, leading to a visibly distorted layout.
- **1** → Severe inaccuracies — most elements have incorrect proportions or spacing, making the layout visually broken and inconsistent with the wireframe.

3. Alignment & Grid Consistency (1-5):
Are elements precisely aligned according to the wireframe, following the expected grid/flex structure and ensuring uniform positioning?
- **5** → Perfect alignment. Every element follows the wireframe's grid, flex, or positioning structure exactly. No misalignments.
- **4** → Mostly aligned, but minor deviations exist (e.g., slightly off-center text or small pixel variations in placement).
- **3** → Some clear misalignments — at least one noticeably off-grid or misplaced element that affects overall balance.
- **2** → Major alignment issues, with multiple elements misaligned, overlapping, or not following the expected structure.
- **1** → Severe disorganization — the output fails to follow the wireframe's grid or positioning, making the layout appear chaotic.

Final Score Calculation:
Final Score = (Layout Structure Accuracy + Spacing & Proportions + Alignment & Grid Consistency) / 3

Output Format:
Your response must follow this JSON structure:

```
{
  "descriptions": {
    "input sketch": "provide the description of sketch here",
    "AI-rendered website": "provide the description of website here"
  },
  "scores": {
    "layout_structure_accuracy": [1-5],
    "spacing_proportions": [1-5],
    "alignment_grid_consistency": [1-5]
  },
  "final_score": [calculated average score],
  "reasoning": "[Concise evaluation highlighting key strengths and weaknesses]"
}
```

Figure 12. LLM-as-Judge input prompt: It evaluates the model output and the groundtruth among some detailed criteria given in prompt

Web Code Editing Evaluation Prompt

```
  You are evaluating a system that generates HTML code based on a given task. Compare the predicted
      code with the ground truth code and rate its correctness based on functionality rather than
      exact syntax. If the code performs the intended task correctly, even if formatted differently
       or using a different approach, it should receive a high score.
    ### Rating Scale:
    5 - PERFECT
    - Fully achieves the required functionality as described in the reference output.
    - May have differences in syntax or structure, but effectively performs the same task with no
        missing elements.
    4 - CORRECT BUT WITH MINOR ISSUES
    - Achieves the intended functionality but has small flaws (e.g., slightly different behavior,
        minor inefficiencies).
    3 - PARTIALLY CORRECT
    - Achieves part of the intended functionality but is missing key aspects or has notable issues.
    2 - MOSTLY INCORRECT
    - Fails to accomplish most of the required functionality but shows some partial effort.
    1 - COMPLETELY INCORRECT
    - The solution does not fulfill the required functionality or is entirely off-target.

    ### Examples for reference:
        [{
            "INPUT": {
            "question": "Change the header's background color to blue.",
            "model_answer": "+<style>\n  header { background-color: blue; }\n</style>\n<header>
                Welcome</header>",
            "ground_truth": "<header style='background-color: blue;'>Welcome</header>"},
            "OUTPUT": {
            "rating": 5,
            "rationale": "The model answer correctly implements the change by ensuring the header
                displays with a blue background. Despite using a style tag in the model answer
                versus inline styling in the ground truth, both approaches deliver the exact
                intended functionality."}
        },

    }
]

    ### Task for Evaluation:
        {
    "INPUT": {
        "question": "question",
        "model_answer": "model_answer",
        "ground_truth": "ground_truth"
    }
}

Provide your evaluation in the following JSON format (using
json
delimiters, do not include any extra text):
{
  "rating": "1 or 2 or 3 or 4 or 5",
  "rationale": "[Brief explanation of why this rating was chosen]"
}
```

Figure 13. LLM-as-judge prompt for Web Code Editing task using few shot examples.

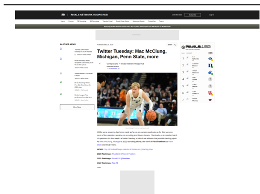
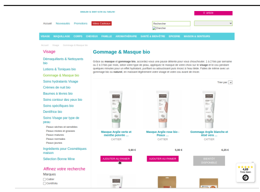
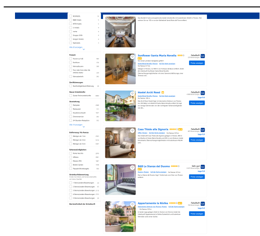
	How can I find more information about the player A.J. Dybantsa? Answer: Click on the ”RIVALS150 ranking” at the lower left and select ”A.J. Dybantsa” at (x1:230.34, x2:297.32, y1:1049.92, y2:1083.07).	How many players are visible in the 4th image on the left side? A) 2 B) 1 C) 4 D) 3 Answer: D) 3	Which user pays less when subscribing annually vs. monthly? Answer: - Yearly: $99.95 - Monthly: $9.95 × 12 = $119.40 - Savings: $19.45
	¿En qué área puede el usuario hacer clic para ver publicaciones anteriores? A) Categorías B) Lo más visto C) Archivo del blog Respuesta: C) Archivo del blog.	¿A qué tipo de público está dirigido este blog de recetas? Respuesta: Personas interesadas en repostería y cocina casera.	¿Cuál sería el efecto en la navegación si la sección ”Lo más visto recientemente” estuviera al inicio? Respuesta: Aumentaría la accesibilidad de los artículos populares.
	Je cherche un soin pour le visage. Où dois-je aller ? Réponse: Dans la barre latérale sous ”Soins visage par type de peau”.	Quel est le prix total des articles si l’on exclut celui avec 5 étoiles ? A) 58,70 € B) 62,85 € C) 45,50 € D) 51,90 € Réponse: B) 62,85 €.	Avec un budget de 15 €, quels produits puis-je acheter ? Réponse: - Masque Argile verte et menthe poivrée bio (5,90 €) - Masque Argile rose bio - Peaux sensibles (5,90 €)
	Comment filtrer les hôtels qui acceptent les animaux ? Réponse: Dans la section ”Ausstattung”, cliquez sur ”Haustiere erlaubt”.	Combien de chaises sont visibles dans l’image de ”Sunflower Santa Maria Novella” ? Réponse: 4 chaises.	Quelles sont les meilleures options d’hébergement près de Via Faenza ? Réponse: ”Sunflower Santa Maria Novella” avec une note de 8.9.

Table 4. **WebMMU VQA Task Samples.** This table presents diverse Visual Question Answering (VQA) task samples from the WebMMU dataset, categorized into three types: (1) Functional (interaction with webpage elements), (2) General Understanding (visual recognition within webpage images), and (3) Complex Reasoning (logical inference and numerical computation). Each row showcases an input webpage image alongside representative questions and answers.

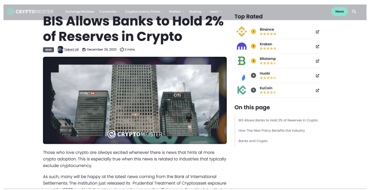
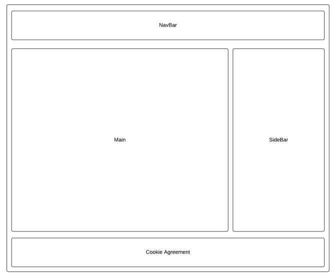
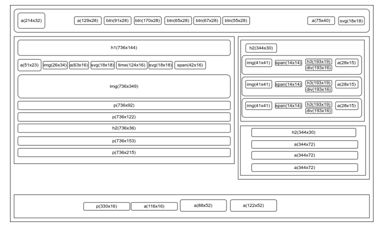
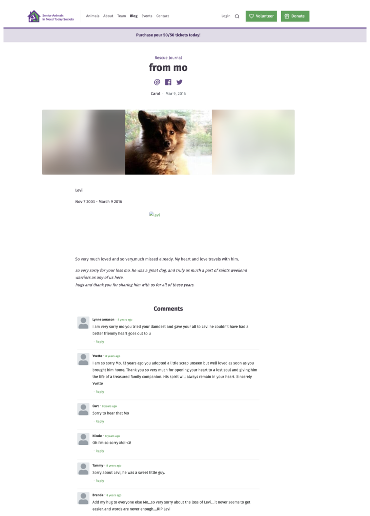
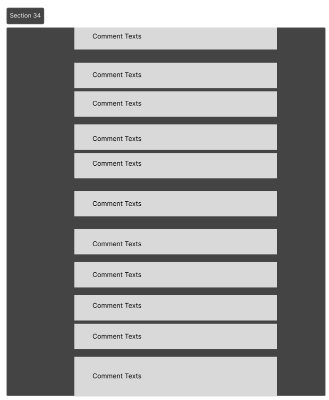
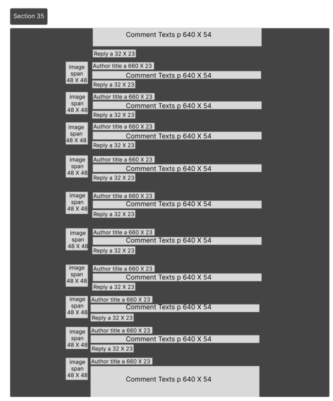
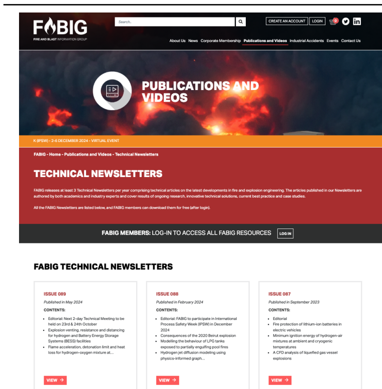
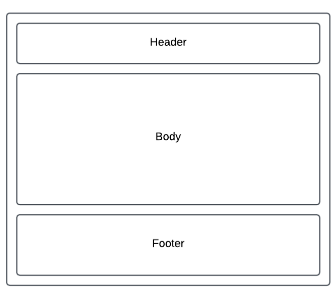
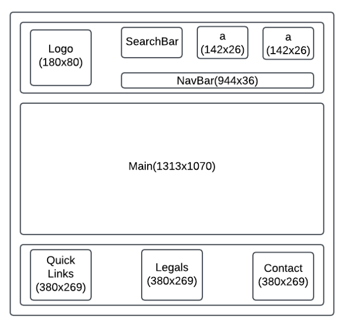
Input Image	Basic Layout Sketch	Detailed UI Representation
		
		
		

Table 5. **Mockup2Code Task Samples.** This table showcases examples from the Mockup2Code task, illustrating the transformation of webpage images into structured representations. Each row includes: (1) an Input Image (webpage screenshot), (2) a Simple Sketch (basic layout structure), and (3) a Complex Sketch (detailed UI components and text placements).

9. Case Studies for the Mockup2Code Task

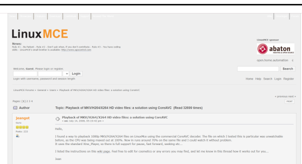
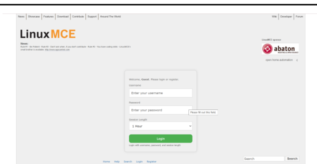
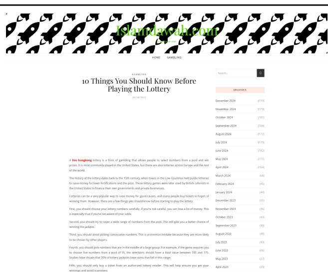
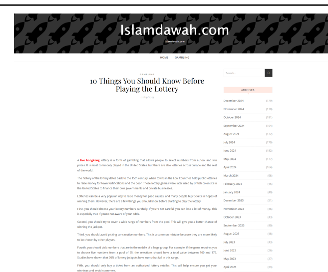
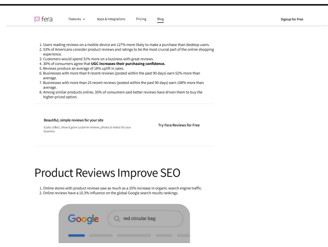
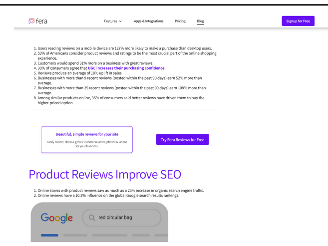
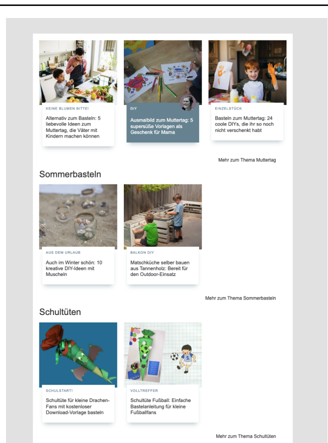
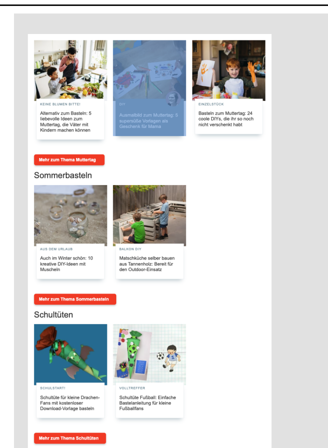
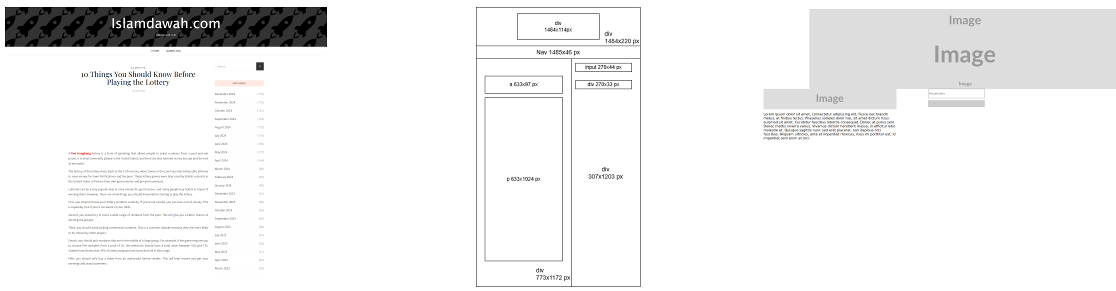
Input Image	Task Description	Rendered Image
	Comment faire pour afficher les différentes sections de l'article "La fête de la prune" en colonne et agrandir les images ?	
	Which changes should be made in the HTML code to improve the UI of the login form and navbar?	
	How can I fix the header element by adding a black overlay over the image, changing the font color to white, and setting the font family to "Lucida Sans"?	
	For a visually appealing design, enhance the navbar with hover and shadow effects, add hover interactions to buttons and links, and apply a card effect to containers.	
	Mache die drei Felder "link-next" auffälliger, indem du ihre Farbe, Größe oder Schriftstil anpasst.	

Table 6. Code edition before and after screenshot of webpage.

Model	English			French			German			Spanish		
	BLEU	**TreeBLEU**	**LLM-as-Judge**	**BLEU**	**TreeBLEU**	**LLM-as-Judge**	**BLEU**	**TreeBLEU**	**LLM-as-Judge**	**BLEU**	**TreeBLEU**	**LLM-as-Judge**
QwenVL-7B	9.02	28.91	2.48	4.11	22.17	1.81	5.41	24.02	2.18	7.22	14.19	2.31
Molmo-7B	1.98	11.91	1.93	3.12	5.77	1.62	1.01	12.30	1.70	1.82	4.21	1.64
Phi-3.5-VI	0.00	0.00	1.02	0.00	0.00	1.00	0.00	0.00	1.00	0.01	0.00	1.00
Fuyu-8B	0.02	0.09	1.07	0.00	0.00	1.06	0.00	0.00	1.06	0.00	1.11	1.03
InternVL-2.5-8B	10.46	25.96	2.50	6.61	14.03	2.07	9.68	23.40	2.34	5.73	15.03	2.13
Glm-4v-9B	6.09	21.74	2.50	4.75	15.23	2.31	4.60	17.00	2.21	5.12	7.38	2.37
Llava-OV-7B	8.08	27.98	2.41	3.32	16.76	2.06	5.78	17.63	2.09	4.42	11.34	2.07
Pixtral-12B	12.16	26.59	2.81	6.28	14.52	2.74	11.07	23.67	2.64	6.04	14.36	2.52
InternVL-2.5-38B	15.84	36.19	3.55	8.01	**26.77**	3.15	14.12	**33.75**	3.40	10.14	18.55	3.16
QwenVL-72B	16.00	38.38	3.27	9.40	25.34	3.13	14.16	30.41	3.01	10.36	19.97	3.04
Claude	**22.80**	**38.92**	**4.04**	**16.57**	24.66	3.69	**20.61**	32.61	**4.00**	**13.65**	**22.79**	3.64
Gemini-2-Flash	14.34	24.80	3.15	11.11	13.10	3.26	11.62	23.14	3.15	10.71	18.49	3.25
GPT-4o	18.94	35.11	3.93	11.81	12.47	**3.96**	15.47	25.23	3.72	10.89	15.14	**3.65**

Table 7. Results of Web Code Editing on different languages.

OpenAI-o1

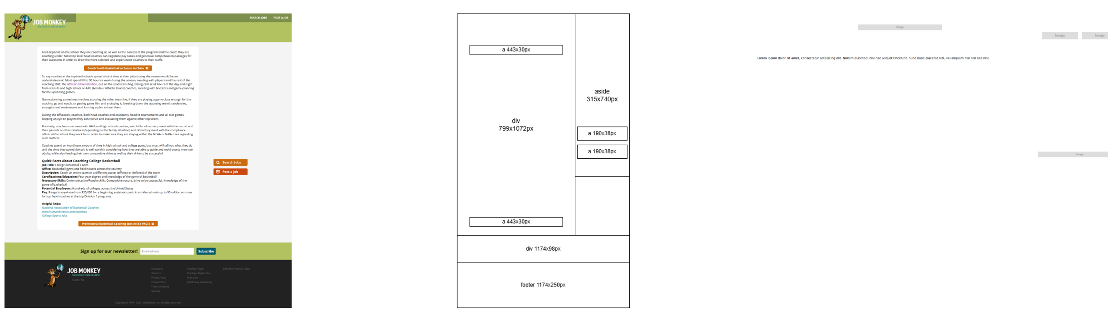


Original Page | Mockup Image | Generation

GPT Score Evaluation

```
Alignment:  2. Key elements (e.g., input box) misaligned, deviating from intended grid.
Layout:  2. Two-column structure poorly represented, essential sections missing/merged.
Spacing:  2. Uneven element distribution results in inconsistent spacing and imbalance.
Overall Score:  2
```

InternVL-38B



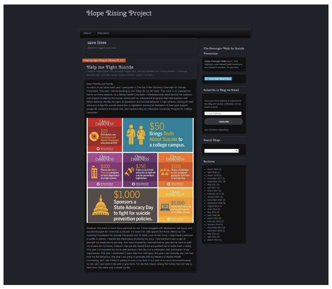
Original Page | Mockup Image | Generation

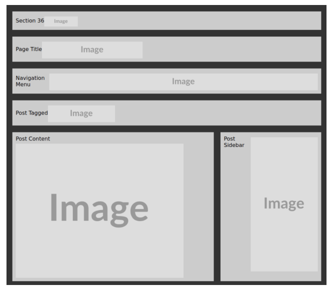
GPT Score Evaluation

```
Alignment:  2. The main content is misaligned; also, key elements like the aside and buttons are missing.
Layout:  2. The intended two-column structure is replaced by a single-column layout, significantly deviating
     from the sketch.
Spacing:
     2. Inconsistent spacing and proportions arise from the absence of the aside and smaller placeholders.
Overall Score:  2
```

Figure 14. Examples of the failure cases on the Mockup2Code task for the best closed-source model (OpenAI-o1) and the best open-source model (InternVL-38B).

OpenAI-o1



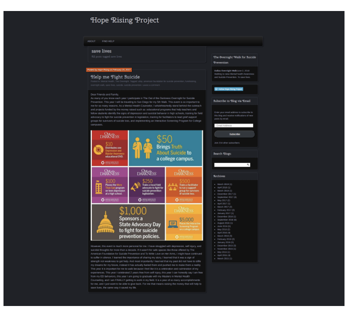


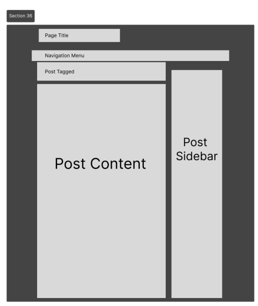
Original Page **Mockup Image** **Generation**

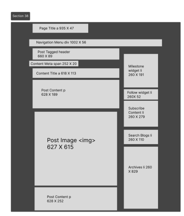
GPT Score Evaluation on Simple sketch

```
Layout:  5. Structure perfectly replicated.
Spacing:  4. Minor inconsistencies in spacing.
Alignment:  5. Precise alignment maintained.
Overall Score:  4.67
```

OpenAI-O1







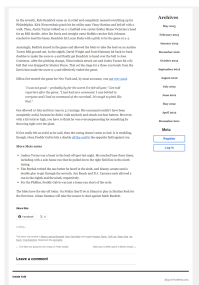
Original Page **Mockup Image** **Generation**

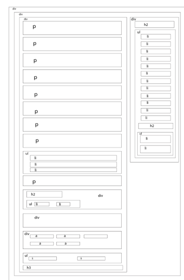
GPT Score Evaluation on Complex Sketch

```
Layout:  5. Structure and sections perfectly replicated.
Spacing:  5. Spacing and proportions are consistent.
Alignment:  5. Alignment is precise and matches the sketch.
Overall Score:  5
```

Figure 15. Examples of the Success cases on the Mockup2Code task for the best closed-source model (OpenAI-o1) for both simple and complex mockups.

Internvl2.5-8b



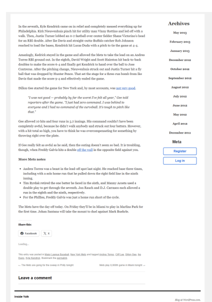


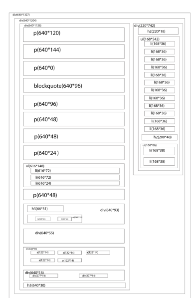
Original Page | Mockup Image | Generation

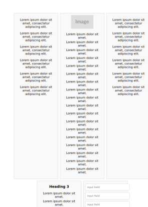
GPT Score Evaluation on Simple Sketch

Layout:
2. Many key sections, headers, and lists are missing, leading to a poor structural match with the sketch.
Spacing: 2. Uneven gaps and inconsistent spacing cause improper grouping and distorted proportions.
Alignment: 2. Misaligned elements and an irregular grid result in a disorganized layout.
Overall Score: 2

Internvl2.5-8b







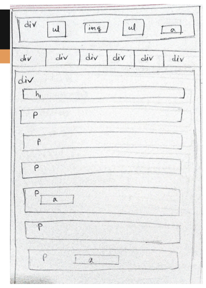
Original Page | Mockup Image | Generation

GPT Score Evaluation on Complex Sketch

Layout: 4. Two-column layout is largely reproduced with most elements accurately placed, though an extra image placeholder appears.
Spacing: 3. Inconsistent spacing leads to uneven gaps that affect the overall balance.
Alignment: 3. Some elements are misaligned, disrupting the intended grid structure.
Overall Score: 3.33

Figure 16. Examples of the Failure cases on the Mockup2Code task for the best closed-source model (Internvl2.5-8b) for both simple and complex mockups.

OpenAI-o1

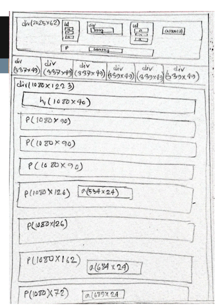


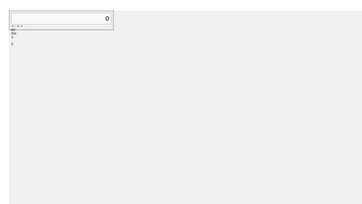
Original Page | Mockup Image | Generation

GPT Score Evaluation

```
Alignment:  2. Key elements (e.g., input box) misaligned, deviating from intended grid.
Layout:  2. Two-column structure poorly represented, essential sections missing/merged.
Spacing:  2. Uneven element distribution results in inconsistent spacing and imbalance.
Overall Score:  2
```

OpenAI-O1





Original Page | Mockup Image | Generation

GPT Score Evaluation

```
Alignment:  2. The main content is misaligned; also, key elements like the aside and buttons are missing.
Layout:  2. The intended two-column structure is replaced by a single-column layout, significantly deviating
     from the sketch.
Spacing:
     2. Inconsistent spacing and proportions arise from the absence of the aside and smaller placeholders.
Overall Score:  2
```

Figure 17. Examples of the Success cases on the Mockup2Code task for the best closed-source model (OpenAI-o1) for both simple and complex mockups.