WebQuest: A Benchmark for Multimodal QA on Web Page Sequences

Maria Wang [*1] **Srinivas Sunkara** [*1] **Jason Lin** [1] **Gilles Baechler** [1] **Fedir Zubach** [1] **Lei Shu** [1] **Yun Zhu** [1] **Jindong Chen** [1]

Abstract

The growing power of multimodal large language models (MLLMs) is turning autonomous web agents that assist users into a reality. To accurately assess these agents' capabilities in real-world scenarios, we introduce WebQuest. This new benchmark dataset challenges MLLMs with cross-page question-answering that requires complex reasoning, such as arithmetic and sorting, across diverse website categories. Unlike existing web agent benchmarks that focus on multi-step web navigation and task completion, WebQuest evaluates information extraction, multimodal retrieval and composition of information from many web pages at once. We provide three dataset splits: Single Screen QA, Multi Screen QA, and Trace QA based on navigation traces. We evaluate leading proprietary multimodal models like GPT-4o, Gemini 1.5 Pro, Claude 3.5 Sonnet, and open source models like InternVL2.5, Pixtral and Qwen2.5-VL on our dataset, revealing a significant gap between single-screen and multi-screen reasoning. We also explore techniques like Chain-of-thought prompting to address this gap.

1. Introduction

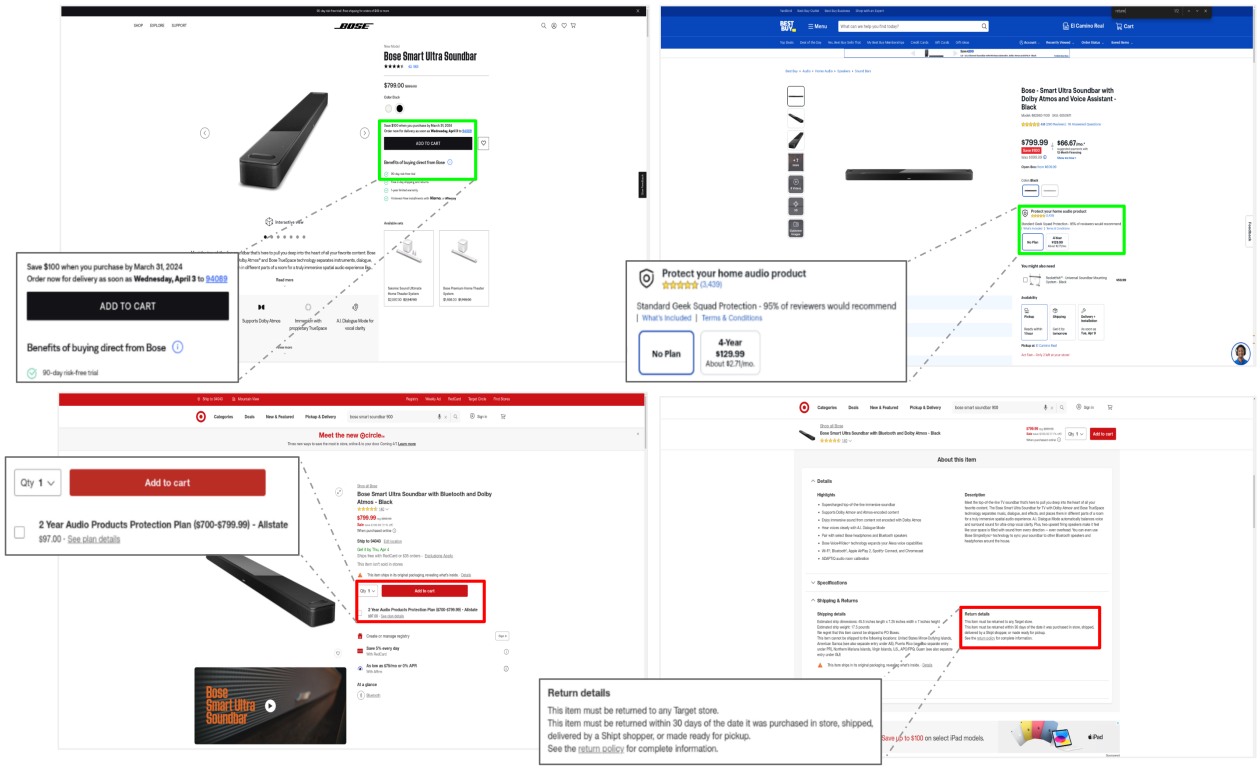
Web User-Interfaces (UIs) act as a crucial bridge between computers and humans, allowing users to acquire information, complete tasks, engage in social interactions and perform various other daily activities. These UIs are inherently multimodal, rich in semantic knowledge and structure, presenting information through text, images, interactive elements, etc. The pervasive nature of web UIs has driven research into multimodal, assistive technologies designed to simplify user journeys. At the same time, recent MLLMs (Team, 2024; OpenAI et al., 2024; Anthropic, 2024) with strong capabilities across a variety of tasks have sparked interest in building end-to-end agents adept at assisting users with various complicated tasks.

One of the challenges in web agent modeling is to interpret spatial and semantic relations among structured components in sequences of screens. Traditional computer vision tasks like summarization, image captioning, visual question answering have been extended to document images (Mathew et al., 2021), infographics (Masry et al., 2022; Mathew et al., 2022) and UIs (Chen et al., 2021; Hsiao et al., 2022; Liu et al., 2024; Wang et al., 2021; Li et al., 2020). However, all of these tasks consist of single or multi image setups where the relevant information is on at most one of the images, where SoTA MLLMs have saturated at over 85% zero shot performance. Motivated by the progress of MLLMs in single-page UI tasks, the research community has created a variety of multi-step benchmarks focusing on UI Automation (Deng et al., 2023; Koh et al., 2024), demanding task-driven UI navigation (Xue et al., 2025). The majority of these benchmarks however are self-contained within a single website to complete the given task.

Real world web usage often require combining and comparing information from multiple websites and pages. This high-level reasoning across different aspects e.g. style, visually-situated text, arithmetic is seldom studied in literature and uncommon in Internet corpus, yet lends itself to a QA format intuitive to humans. Queries like *"how much more [quality] is a [adjective] item than [adjective] item"* , grounded in multimodal input are pervasive in our daily routines but rarely documented. It can therefore be limiting to develop and evaluate web agents using only the standard QA formulation found in various UI automation datasets. Motivated to address this gap, we reformulate the QA task used in existing UI datasets in two ways. Firstly, we include questions requiring aggregation/reasoning across information extracted from a screen. Secondly, our dataset includes questions spanning multiple websites and pages. In Figure 1, we provide an example from the dataset, illustrating the same product presented in different websites. In this case, the ability to synthesize over information extracted from multiple webpages is essential to compare the different offerings and make an informed choice.

[*]Equal contribution [1]Google DeepMind. Correspondence to: Maria Wang <mariawang@google.com>, Jindong Chen <jdchen@google.com>.

Workshop on Computer-use Agents @ ICML 2025, Vancouver, Canada.



Question: What is the longest return window available for the bose ultra soundbar, and from which site?
Answer: 90 days from bose.com
Question: What is the protection plan with the lowest per year cost?
Answer: Standard Geek Squad protection plan which costs $32.5

Figure 1. A multi screen question and answering example in WebQuest. The first question requires visual information extraction and numerical comparison, while the second question necessitates arithmetic analysis grounded in non-uniform structured web screens. Both questions involve multi-step reasoning over extracted information and are designed to emulate human decision-making process. We color-coded parts of the images which are relevant to answering the questions in zoomed-in callouts, where the final answer is located in green.

WebQuest enables detailed model benchmarking through three distinct splits: *Single Screen QA* includes multimodal content from a single web screen; *Multi Screen QA* extends this to multiple screens; and *Trace QA* involves sequences of screens navigated by humans in web browsing sessions. We evaluate various state-of-the-art proprietary models GPT-4o (OpenAI et al., 2024), Gemini 1.5 Pro (Team, 2024), Claude 3.5 Sonnet (Anthropic, 2024), and open source models like Qwen2.5-VL (Bai et al., 2025), InternVL-2.5 (Chen et al., 2025), Pixtral (team, 2024) and InstructBLIP (Dai et al., 2023) via prompt engineering and Chain-of-thought prompting, revealing a significant gap between single and multi-page reasoning. Finally, we run text-only and UI element grounding evaluations to better characterize MLLM performance in understanding semantic information and leveraging visual structures in long screen sequences.

The main contributions of our work are multifold. We believe it will advance the field of UI understanding.

- WebQuest is the first QA benchmark requiring synthesizing information across various parts of a single UI or multiple UIs.
- WebQuest is the first dataset with QA over long interaction sequences focused on information extraction and arithmetic reasoning across the sequences.
- WebQuest includes 3 subsets of data containing single-screen, multi-screen and navigation traces, enabling detailed benchmarking of the capabilities of different models.

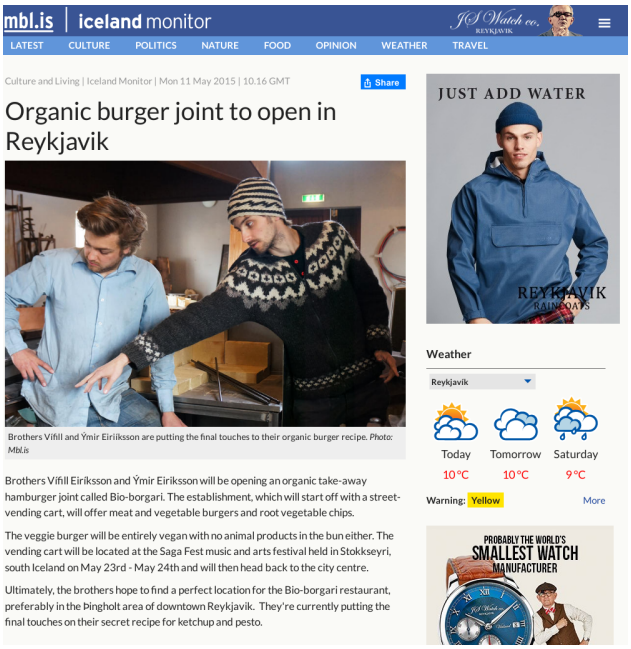
The dataset, the metrics computation scripts and the prompts that we used for our evaluations will be

made available on github at: https://github.com/google-deepmind/webquest.

2. Related Work

Visual Question Answering The problem of answering questions based on the contents of an image has been studied extensively by the research community from VQA (Antol et al., 2015) on natural images, to TextVQA (Singh et al., 2019) of text in natural images and DocVQA (Mathew et al., 2021) containing documents with text, tabular structures and figures. Datasets like ChartQA (Masry et al., 2022) and InfographicVQA (Mathew et al., 2022) emphasize complex numerical and compositional reasoning in documents and UIs. For web and UI understanding, QA datasets like WebSRC (Chen et al., 2021) focused on desktop screens and ScreenQA (Hsiao et al., 2022) focused on mobile app screens; both contain questions which require extracting relevant information from a single screen. While related, our dataset contains questions necessitating more sophisticated and cross-screen reasoning. Similar to our dataset in having multiple images as input, Multi-page DocVQA (Tito et al., 2023) requires visual extraction from up to 20 pages to answer questions. However, each question om Multi-page DocVQA requires only a single page that contains the answer to be identified whereas in WebQuest, the answer necessitates extracting and aggregating information from multiple screens. To the best of our understanding, there are no QA benchmarks spanning multiple websites or images in a task-oriented sequence, where a subset of pages are necessary for joint reasoning. Other datasets like TaT-DQA (Zhu et al., 2022) and DUDE (Landeghem et al., 2023) contain questions over multiple pages, but are based on documents with tables and not rich UIs.

Web Navigation and Agent Benchmarks Recent research in agent datasets that focus on performing various tasks on digital screens have received widespread attention. Earlier efforts introduced simulated web and mobile environments, such as MiniWob++ (Liu et al., 2018), MoTIF (Burns et al., 2022), Mind2Web (Deng et al., 2023) and WebShop (Yao et al., 2023). To facilitate autonomous web agents, Online-Mind2Web (Xue et al., 2025), WebArena (Zhou et al., 2024), VisualWebArena (Koh et al., 2024) and WebLinx (Lù et al., 2024) pair interactive instructions with DOM trees, HTML and pixel-based environments. Most of these datasets contain tasks that are limited to one website per task or one website per task category (e.g. shopping, travel). Recently released MMInA (Zhang et al., 2024) is the first to collect multi-hop, sequential navigation across websites. While they focus on long-range reasoning with only one website per task category, WebQuest has multiple websites per category, closer to real world workflows. GAIA (Mialon et al., 2023) is a generalist agent benchmark



Question: How many more days will it be cloudy than rainy?
Answer: 1

Figure 2. An example of Single screen QA, where the task is to count how many cloudy days before a rainy one, and the weather conditions are depicted by pictograms. Note we do not include partial screenshots in our dataset, so a model needs to identify relevant parts of the screen to answer the question.

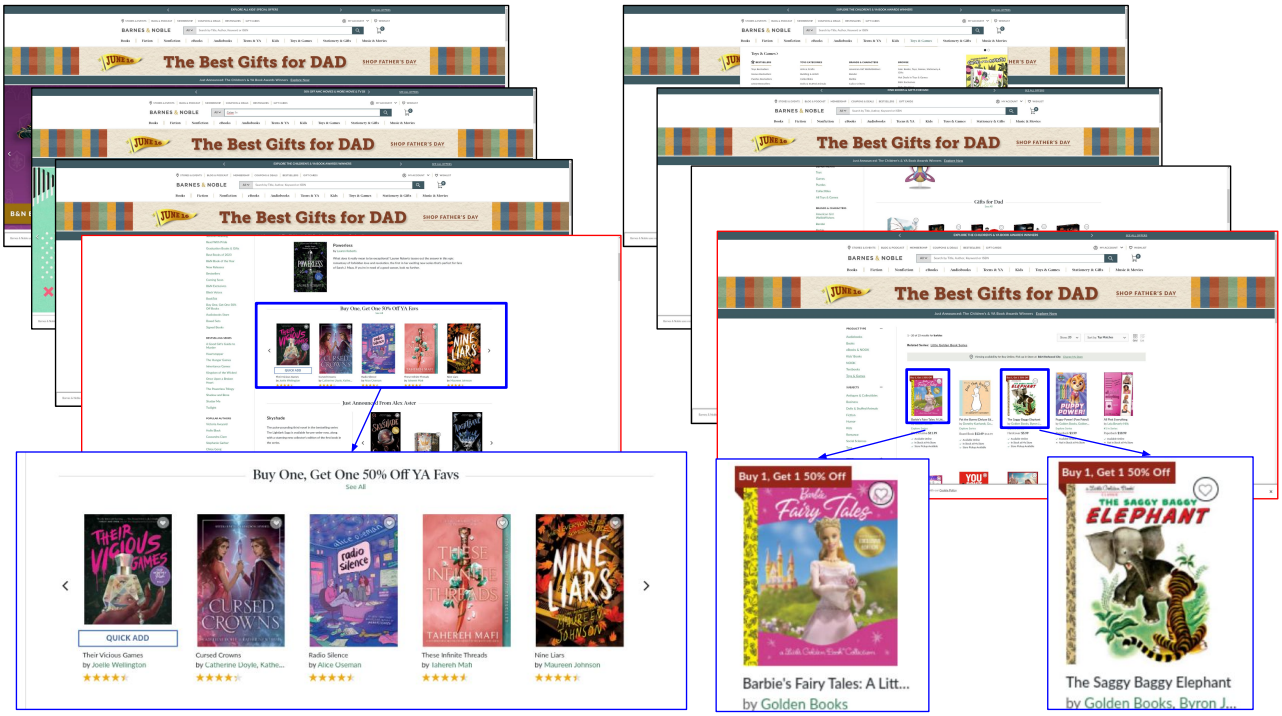
that requires a range of skills spanning complex tool-use and multi-step reasoning. As with MMInA, WebQuest differs from GAIA in that it uses multiple websites per question in each category.

3. WebQuest Benchmark

WebQuest is a multimodal benchmark for question answering based on the contents of web UIs. In contrast to many existing VQA datasets, WebQuest enhances the scope of the QA task by focusing on reasoning questions across multiple related webpages and websites. To better understand the capabilities and characterize the performance of various modeling approaches, we split the dataset into three categories: *Single Screen QA*, *Multi Screen QA*, and *Trace QA*.

3.1. Single Screen QA

This category comprises question-answer pairs, each derived from the content of a single web UI. In contrast to other UI-based QA datasets (Chen et al., 2021; Hsiao et al., 2022; Liu et al., 2024), answering questions in this set often necessitates use of arithmetic and logical operations across



Question: How many books I've viewed have Buy 1, Get 1 50% Off discount? **Answer:** 7

Figure 3. This figure provides an example of Trace QA, specifically for a task involving counting number of items satisfying given conditions. It shows a complete browsing session as a sequence of screens. However, to answer the question, one only needs to focus on certain screens, extracting and analyzing relevant visual and semantic information from them. We marked the key images with a red border, and used zoomed-in callouts with blue borders to highlight specific portions of these images.

information extracted from multiple screen elements. For example, determining the number of rainy days might involve analyzing pictograms as illustrated in Figure 2. The distribution of operations required for these questions is detailed in Figure 7. This split comprises 542 examples, gathered by having raters follow these instructions when exploring a pre-selected list of popular websites:

1. Explore the website.
2. Consider the kind of questions users might be interested in when using the website, focusing on arithmetic reasoning.
3. Capture a screenshot and formulate a corresponding question-answer pair.

Throughout the data collection process, the generated examples underwent frequent review by the authors to maintain high quality and ensure diversity. Furthermore, two rounds of data validation were conducted to eliminate screenshots containing personally identifiable information (PII). A breakdown of website categories within the dataset can be found in Figure 4. Exact rating instructions and more details are provided in Appendix B.

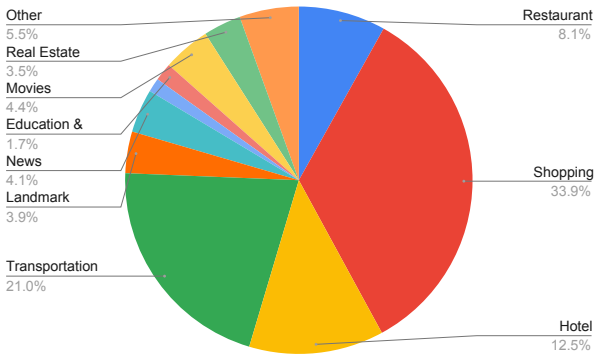


Figure 4. Distribution of website categories of Single Screen QA examples.

3.2. Multi Screen QA

In the Multi Screen QA task, we aim to capture instances where users are browsing the web with the aim of gathering information focused on a particular task, e.g., shopping for a sound bar as depicted in Figure 1. This could involve tasks

like comparing product attributes across websites, filtering travel options etc. The overall goal is similar to the Trace QA described in Sec 3.3, however for this case, the screens containing relevant information for answering the questions are already filtered. This occurs in cases where a user has explored several relevant webpages, opened them in different tabs or taken screenshots, and then seeks answers pertaining to these pages. It is important to note that the different screens in each QA pair are semantically related and can come from webpages in the same or alternative websites.

This split contains 307 QA pairs. The overall guidance provided to raters is similar to the one in section 3.1. One major difference is that now the raters are instructed to capture multiple screenshots from different webpages. To ensure that related websites are explored for each QA pair, raters are provided a curated list of popular websites grouped into categories. Each rater is first assigned a domain (e.g., travel, shopping etc) for each question, and are asked to follow the instructions:

1. Choose 2-5 websites from the assigned category.
2. Imagine a user researching a product, service, or location online, comparing information across webpages and websites.
3. Take 3-5 screenshots from websites related to the chosen use case.
4. Formulate a question-answer pair based on the combined information in all screenshots, focusing on arithmetic reasoning and reflecting common user journeys.

Please refer to Appendix B for rating details. The authors reviewed the data and provided feedback; two validation rounds prevented inappropriate content. Figure 7 shows the distribution of necessary math operations. Website category statistics are in Appendix A.

3.3. Trace QA

The Trace QA split addresses scenarios where users browse the web to gather information for a specific task, requiring them to synthesize data from multiple websites and pages. Unlike Multi-Screen QA, where all screens in a session are relevant to the question, Trace QA datasets contain screen traces where only a subset of screens contain information necessary to answer the question. This reflects a more realistic information-seeking behavior. A Trace QA example is provided in Figure 3.

To create this dataset, we provided raters with a list of popular websites and their associated categories, similar to the Multi-Screen QA data collection. Raters then browsed these websites and recorded their screen traces. They were instructed to follow these steps for the *Trace QA* split (See Appendix B for more rating details):

1. Roleplay a user researching information across different websites and pages. Identify potential questions this user might have.
2. Begin recording the screen trace from the initial webpage and navigate through the different webpages as part of the trace.
3. Generate a question-answer pair relevant to the browsing session.
4. Ensure the question requires arithmetic reasoning and integrates information extracted from approximately 3-5 screenshots within the trace.

The distribution of operations present in the questions is depicted in Figure 7. The mean trace length is 16 screens. Website category breakdown and the distribution of screens per trace can be found in Figure 5 and Fig 6. Finally, the data underwent rigorous validation to remove any inappropriate screens and was manually reviewed to ensure accuracy and correctness.

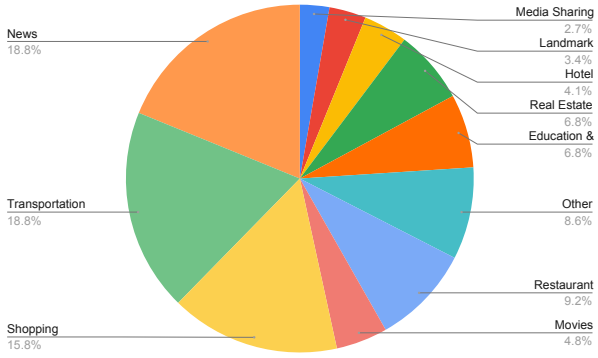


Figure 5. Distribution of categories of Trace QA examples. News, Transportation and Shopping categories contain the most number of traces.

4. Experiments

4.1. Metrics

To evaluate the performance of various models on the WebQuest benchmark, we utilize a variant of the Relaxed Accuracy metric used for the ChartQA (Masry et al., 2022) dataset. Since over half the answers are numerical, we extract numbers from predictions and ground truths. For floating-point answers, we allow a ± 0.05 margin, while integer and string answers require an exact match after SQuaD (Rajpurkar et al., 2016) pre-processing. To account for answer variations, we use multiple ground truth variants

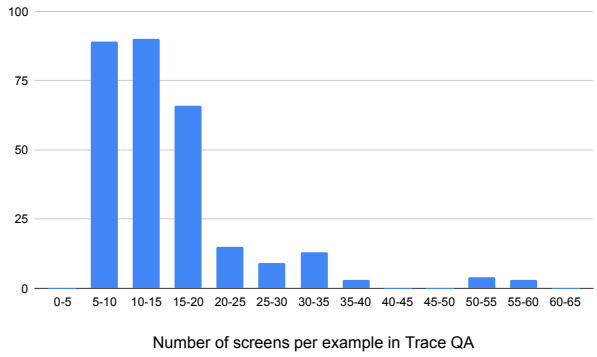


Figure 6. Distribution of the length of traces for the Trace QA benchmark. There are a few very long traces (> 50 screens), while a majority of the traces contain fewer than 20 screens.

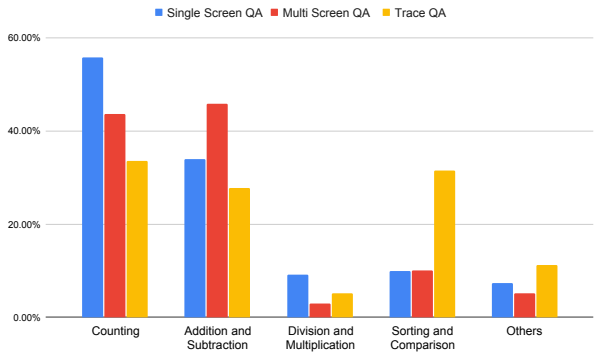


Figure 7. Distribution of the operations needed for each split. Please note that some of the questions may require multiple operations. Single and Multi Screen splits have a higher proportion of questions involving counting and arithmetic operations. Trace QA split shows an even distribution across counting, arithmetic and sorting/comparison operations.

generated by Gemini 1.5 Pro (Team, 2024) and report the highest score. Although the Trace QA split has actions associated with each step, we do not evaluate action accuracy. This dataset is focused on addressing the gaps in question answering across multiple websites and we envision it being used in conjunction with other UI Control datasets like WebArena (Zhou et al., 2024), VisualWebArena (Koh et al., 2024), Mind2Web (Deng et al., 2023).

4.2. Baselines

We evaluate the performance of a number of state-of-the-art MLLMs and leading open source models on our benchmark. We use out-of-the-box parameters and default settings of API-based models for evaluation. The model input includes the screen image(s) and question. Inference time techniques like prompt engineering and Chain-of-thought prompting (Wei et al., 2022) were utilized to improve the model performance. In our Chain-of-thought (Wei et al., 2022) prompt, we include stages (steps) to analyze the screenshot and question, extract information from each screen, analyze extracted information, then finally to generate the answer in short phrases. We evaluated various state of the art models like GPT-4o (OpenAI et al., 2024), Gemini 1.5 Pro (Team, 2024), Claude 3.5 Sonnet (Anthropic, 2024). For Claude 3.5 Sonnet (Anthropic, 2024) Trace QA evaluations, due to memory constraints, we resize the images preserving aspect ratio such that the shorter side is 1440 pixels. For GPT-4o (OpenAI et al., 2024) and Gemini 1.5 Pro (Team, 2024) evals we use the original image resolutions for all questions. We also evaluate open sourced models, including BLIP2 (Li et al., 2023), InstructBLIP (Dai et al., 2023) variant based on Flan-T5-XXL language model (Chung et al., 2022), Qwen2.5-VL-7B (Bai et al., 2025), InternVL2.5-8B(Chen et al., 2025) and Pixtral-12B (team, 2024). For InstructBLIP, InternVL and Pixtral, we resize the images such that the longer side is 672 pixels and 1024 pixels respectively. For Qwen2.5-VL, we use flash-attention implementation for the Multi Screen QA and Trace QA splits.

4.3. Main Results

In this section, we present a comprehensive comparison of different MLLMs on our WebQuest benchmark. To better understand the capabilities and failure cases of different models we present results on the various splits described in section 3. The best performance for each MLLM on the WebQuest splits can be seen in Table 1. We analyze the results in the sections below.

4.3.1. Single Screen QA

For the Single Screen QA split, we notice that all the MLLMs perform relatively well compared to Multi Screen QA and Trace QA. Chain-of-thought (Wei et al., 2022) prompting improves the scores by 13% on average. We also notice a significant performance difference between proprietary models and some open source models like InstructBLIP, Pixtral but the gap is much lower when compared to recent models like Qwen2.5-VL, even though these models are smaller in size. When comparing performance of the highest scoring proprietary and open-source models, the biggest difference is in questions involving arithmetic where Qwen2.5-VL scores 10% less than Claude-3.5 Sonnet.

4.3.2. Multi Screen QA

For this split, we notice that all the MLLMs perform worse than the single screen split. With some MLLMs, we notice considerable performance improvement with Chain-of-thought (Wei et al., 2022) prompting and few shot evaluations. However, the performance for all the MLLMs drops

Table 1. Performance of MLLMs on the *WebQuest benchmark* variants, using Relaxed Accuracy. Prompt tuning was conducted separately for each model. All MLLMs perform much better on the *Single Screen QA* split and worse as the number of screens increase. *Trace QA* split is considerably more challenging than the *Multi Screen QA* split. Furthermore, models perform better with CoT. Note the last 4 are open-source models.

Model	**Single Screen QA**		**Multi Screen QA**		**Trace QA**	
	no CoT	CoT	no CoT	CoT	no CoT	CoT
Claude-3.5 Sonnet	43.5	61.6	38.4	56.5	37.7	40.4
Gemini 1.5 Pro	42.3	52.4	16.9	31.6	18.5	29.1
GPT-4o (2024-11-20)	40.2	57.0	35.8	52.4	32.2	35.6
InstructBLIP-Flan-T5 XXL	9.1	-	7.5	-	-	-
InternVL2.5-8B	24.9	45.9	12.4	28.0	-	-
Pixtral-12B	18.1	37.8	7.5	27.5	-	-
Qwen2.5-VL-7B-Instruct	41.5	50.9	13.8	36.2	21.2	22.3

Table 2. Performance of Claude 3.5 models with CoT prompting, comparing image and text-only evaluations (Accuracy %).

Data Split	**Image**	**Text only**
Single Screen QA	61.6	52.2
Multi Screen QA	56.5	42.4
Trace QA	40.4	39.7

Table 3. Performance of Gemini 1.5 Pro models on UI element grounding task requiring the identification of UI elements relevant to answering a question.

QA Type	**Answer Acc**	**BBox F-1 @IOU=0.1**	**Answer Acc@IOU=0.1**
Single Screen QA	40.1	31.1	10.7
Multi Screen QA	14.8	29.3	4.3

to on average 34.1% on this split, highlighting the difficulty models have in reasoning across multiple images. We further analyze the performance gap in section 4.6. Compared to Single Screen QA, we notice a significant performance gap between open-source and proprietary models. We also noticed larger gaps for open-source models on questions related to arithmetic, while comparison questions were handled better.

4.3.3. TRACE QA

The Trace QA split has an average of 15.8 screens per example, of which on average 3.2 screens contain relevant information. All models performed worse on TraceQA than Single Screen and Multi Screen QA. With Chain-of-thought prompting (Wei et al., 2022), proprietary MLLMs on average achieve a score of 35.2%. Claude 3.5 Sonnet (Anthropic, 2024) achieves the best performance with a score of 40.4% on this dataset. There is a gap of ¿15% comparing open-source models to the best proprietary models. We did not display the results for InternVL-2.5, Pixtral-12B and InstructBLIP models as it needed aggressive resizing of the images and hence was not reflective of model capabilities.

4.4. Text only Evaluations

To better understand the importance of visual information in our benchmark, we evaluated Claude-3.5 Sonnet using only OCR-extracted text. We employed Chain-of-thought prompting (Wei et al., 2022) and, for the Multi Screen, Trace QA splits, we included screenshot indices to preserve image separation and screenshot sequence.

As shown in Table 2, performance on Single and Multi Screen QA tasks dropped by more than 9% when input screenshots were removed. This suggests that visual context is beneficial even when all information is on one screen, likely by overcoming OCR errors or leveraging layout. Accurate numerical extraction and calculation remain the key challenge for both approaches. For Multi Screen QA, visual layout and structure can be used to correctly associate information spread across pages, whereas the OCR only model struggles much more with tasks depending on layout or image properties (e.g., failing to extract the correct prices to be summed across different menus).

For Trace QA, the pixel-based model holds only a slight accuracy edge over the OCR-only model (40.41% vs 39.73%). While OCR excels at extracting explicit text if the relevant pages are found (e.g., defining "ripples in spacetime"), it fails on visual-only tasks (like identifying the "Lion" on a jacket). We notice both models find numerical questions significantly harder than non-numerical ones on this benchmark. The pixel model's minor advantage likely stems from using visual cues to potentially better locate relevant pages or answer visual queries, but the core difficulty of filtering irrelevant pages, combined with shared weaknesses in numerical reasoning, seems to equalize performance substantially.

4.5. Grounding evaluations

In addition to evaluating the groundtruth answers for the questions, we also evaluated the models' ability to identify the location of information relevant to answering the questions with UI element grounding annotations for the Single

Screen QA and Multi Screen QA splits. These annotations provide bounding box locations for the UI elements relevant to the questions. We conducted preliminary evaluations using Gemini 1.5 Pro (Team et al., 2023) on these grounded annotations, reporting bounding box F1 (with IoU=0.1) to assess the accuracy of the identified visual elements. We also compute the accuracy of the final answer and answer accuracy with IoU=0.1, which requires both the correct bounding boxes and the final answer to align. These initial evaluations offer insights into the model's ability to identify relevant visual information. The results are presented in Table 3 indicating that even for Single Screen QA, identifying the location of the answers on the images is a challenging task for current VLMs.

4.6. Results Analysis

In this section, we compare the performance of different MLLMs on WebQuest in Table 1. We highlight our key findings below.

4.6.1. Single screen vs multiple screens

Across all models, we consistently observed higher performance on single screen tasks than multi screen tasks. Furthermore, Chain-of-thought prompting (Wei et al., 2022) significantly improved results, yielding average gains of 11.1% on the *Multi screen QA* split and 5.7% on the *Trace QA* split. Our Chain-of-thought (Wei et al., 2022) prompt decomposed the original questions into four reasoning steps: question analysis, screenshot analysis, information extraction, analysis of extracted information, and answer generation. The example in Figure 9 of Appendix D where the models are tasked with counting number of drink options illustrates some interesting differences between the models. Claude-3.5 Sonnet (Anthropic, 2024) and Gemini 1.5 Pro (Team, 2024) did not mis-recognize the food items in image-3, suggesting they leverage strong common sense knowledge. However, all models fail to recognize the color and font difference of the menu item subheadings and details in image-2, which are categories or constituents of the drinks and should not be counted. Gemini 1.5 Pro was poorer at counting, while understanding the task better. GPT-4o (OpenAI et al., 2024) tended to hallucinate and make the incorrect associations in its reasoning attributed to reading error, i.e. mistaking price as number of options and missing the wine option on page 3. Gemini 1.5 Pro performed worse than GPT-4o and Claude-3.5 Sonnet on Multi Screen and Trace QA. In our analysis of 50 randomly selected Multi Screen QA samples indicated that a significant portion of the proprietary models errors stemmed from extracting incorrect or insufficient information across the screenshot sequence. We present more examples in Appendix D.

4.6.2. Common errors made by various models

- **Analysis of UI Interfaces** We observed that all proprietary models tend to be confused by visual/layout differences in UI interfaces serving the same purpose. As displayed in Figure 10, when presented with multiple checkout screens, one containing itemized prices and the other one only the total price, GPT-4o misread the individualized price as total price, an example of *visual oversight*.

- **Counting instances under a given criteria** Another common failure mode is counting instances under a given criteria or association. These questions often involve counting and filtering based on both semantic and visual attributes, such as dining options, costs, colors and style. We found Chain-of-thought step hints to be particularly helpful on decoupling conflated errors across information extraction and reasoning, providing tremendous value in understanding the model's errors and comparing performance between models.

- **Considering questions non answerable** We also observe many errors to be "the question is not answerable" across proprietary models. In these cases, we observe that the main reason is that the model fails at information extraction required from the screens, which can potentially be complemented by text-based inputs.

5. Conclusions

We introduce WebQuest, the first multimodal question answering benchmark designed to evaluate arithmetic and comparative reasoning across multiple screens. Our benchmark encompasses three distinct settings: single-screen, multi-screen, and trace-based, enabling a comprehensive assessment of model capabilities. Evaluations of state-of-the-art MLLMs, including Gemini 1.5 Pro (Team, 2024), Claude 3.5 Sonnet (Anthropic, 2024), and GPT-4o (OpenAI et al., 2024), reveal a significant performance gap between single-screen and multi-screen reasoning tasks. Notably, Chain-of-thought prompting (Wei et al., 2022) proves effective for navigating complex, multi-screen scenarios by facilitating information extraction and synthesis across multiple screens. Future research directions include leveraging richer screen information, such as OCR, DOM, and screen annotations from tools like ScreenAI (Baechler et al., 2024), to potentially enhance model performance and even replace raw screen images in cases of long screen sequences. Furthermore, a compelling area for future research is extending WebQuest for personalized, multi-turn dialogues, which would enable agents to assist users in dynamic, context-aware conversations.

Impact Statement

This paper presents a benchmark aimed at evaluating question answering capabilities of various models on UI screens focusing on arithmetic and logical reasoning questions across multiple related webpages and websites. The questions, answers, and navigation traces collected may reflect cultural and demographic biases of the human raters.

References

Anthropic. Introducing the next generation of claude, 2024. URL https://www.anthropic.com/news/claude-3-family.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. VQA: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Baechler, G., Sunkara, S., Wang, M., Zubach, F., Mansoor, H., Etter, V., Cărbune, V., Lin, J., Chen, J., and Sharma, A. ScreenAI: A vision-language model for UI and infographics understanding, 2024.

Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., and Lin, J. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Burns, A., Arsan, D., Agrawal, S., Kumar, R., Saenko, K., and Plummer, B. A. A dataset for interactive vision language navigation with unknown command feasibility. In *European Conference on Computer Vision (ECCV)*, 2022.

Chen, X., Zhao, Z., Chen, L., Zhang, D., Ji, J., Luo, A., Xiong, Y., and Yu, K. WebSRC: A dataset for web-based structural reading comprehension. *arXiv preprint arXiv:2101.09465*, 2021.

Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Cui, E., Zhu, J., Ye, S., Tian, H., Liu, Z., Gu, L., Wang, X., Li, Q., Ren, Y., Chen, Z., Luo, J., Wang, J., Jiang, T., Wang, B., He, C., Shi, B., Zhang, X., Lv, H., Wang, Y., Shao, W., Chu, P., Tu, Z., He, T., Wu, Z., Deng, H., Ge, J., Chen, K., Zhang, K., Wang, L., Dou, M., Lu, L., Zhu, X., Lu, T., Lin, D., Qiao, Y., Dai, J., and Wang, W. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025. URL https://arxiv.org/abs/2412.05271.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., and Hoi, S. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. URL https://arxiv.org/abs/2305.06500.

Deng, X., Gu, Y., Zheng, B., Chen, S., Stevens, S., Wang, B., Sun, H., and Su, Y. Mind2Web: Towards a generalist agent for the web, 2023.

Hsiao, Y.-C., Zubach, F., Wang, M., et al. ScreenQA: Large-scale question-answer pairs over mobile app screenshots. *arXiv preprint arXiv:2209.08199*, 2022.

Koh, J. Y., Lo, R., Jang, L., Duvvur, V., Lim, M. C., Huang, P.-Y., Neubig, G., Zhou, S., Salakhutdinov, R., and Fried, D. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks, 2024.

Landeghem, J. V., Tito, R., Łukasz Borchmann, Pietruszka, M., Józiak, P., Powalski, R., Jurkiewicz, D., Coustaty, M., Ackaert, B., Valveny, E., Blaschko, M., Moens, S., and Stanisławek, T. Document understanding dataset and evaluation (dude), 2023. URL https://arxiv.org/abs/2305.08455.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL https://arxiv.org/abs/2301.12597.

Li, Y., Li, G., He, L., Zheng, J., Li, H., and Guan, Z. Widget captioning: Generating natural language description for mobile user interface elements, 2020. URL https://arxiv.org/abs/2010.04295.

Liu, E. Z., Guu, K., Pasupat, P., Shi, T., and Liang, P. Reinforcement learning on web interfaces using workflow-guided exploration. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/abs/1802.08802.

Liu, J., Song, Y., Lin, B. Y., Lam, W., Neubig, G., Li, Y., and Yue, X. VisualWebBench: How far have multimodal LLMs evolved in web page understanding and grounding? *arXiv preprint arXiv:2404.05955*, 2024.

Lù, X. H., Kasner, Z., and Reddy, S. WebLINX: Real-world website navigation with multi-turn dialogue, 2024.

Masry, A., Long, D., Tan, J. Q., Joty, S., and Hoque, E. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177.

Mathew, M., Karatzas, D., and Jawahar, C. DocVQA: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Mathew, M., Bagal, V., Tito, R., Karatzas, D., Valveny, E., and Jawahar, C. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

Mialon, G., Fourrier, C., Swift, C., Wolf, T., LeCun, Y., and Scialom, T. Gaia: a benchmark for general ai assistants. *arXiv preprint arXiv:2311.12983*, 2023.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., et al. GPT-4 technical report, 2024.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., and Rohrbach, M. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Team, G. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024.

Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

team, M. A. Announcing pixtral 12b, 2024.

Tito, R., Karatzas, D., and Valveny, E. Hierarchical multimodal transformers for multipage DocVQA. *Pattern Recognition*, 144:109834, 2023.

Wang, B., Li, G., Zhou, X., Chen, Z., Grossman, T., and Li, Y. Screen2Words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 498–510, 2021.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Xue, T., Qi, W., Shi, T., Song, C. H., Gou, B., Song, D., Sun, H., and Su, Y. An illusion of progress? assessing the current state of web agents, 2025. URL https://arxiv.org/abs/2504.01382.

Yao, S., Chen, H., Yang, J., and Narasimhan, K. WebShop: Towards scalable real-world web interaction with grounded language agents, 2023.

Zhang, Z., Tian, S., Chen, L., and Liu, Z. MMInA: Benchmarking multihop multimodal internet agents, 2024.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., Alon, U., and Neubig, G. WebArena: A realistic web environment for building autonomous agents, 2024.

Zhu, F., Lei, W., Feng, F., Wang, C., Zhang, H., and Chua, T.-S. Towards complex document understanding by discrete reasoning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 4857–4866, 2022.

Appendix

We first provide additional statistics for the datasets in Appendix A and then describe the detailed rating instructions in Appendix B. In Appendix C, we detail the different prompts used for evaluation. We illustrate some error cases in Appendix D and provide more examples from the dataset in Appendix E.

A. Statistics of the Dataset

A.1. Single Screen QA

We show detailed statistics about the website categories for Single Screen QA in Table 4.

Category	Num Examples
Restaurant	44
Shopping	184
Hotel	68
Transportation	114
Landmark	21
News	22
Media Sharing	7
Education & Information	9
Movies	24
Real Estate	19
Other	30
Total	**542**

Table 4. Distribution of the 542 examples of Single Screen QA, per categories and the number of screens in the example.

A.2. Multi Screen QA

We show detailed statistics about website categories in Fig 5.

Table 5. Distribution of the 307 examples of Multi Screen QA per categories.

Category	Num Examples
Shopping	209
Restaurant	72
Transportation	12
Hotel	12
Landmark	1
Other	1
Total	**307**

Table 6. Distribution of the 307 examples of Multi Screen QA per the number of screens in the example.

3 screens	4 screens	5 screens	Total
54	131	122	**307**

A.3. Trace QA

The distribution of the examples by website category for Trace QA is presented in Table 7.

Table 7. Distribution of examples of Trace QA for each category

Category	Num Examples
Shopping	46
Restaurant	27
Media Sharing	8
News	55
Movies	14
Transportation	55
Real Estate	20
Hotel	12
Education & Information	20
Landmarks	10
Other	24
Total	**292**

B. Rating Details

For the *Single Screen QA* and *Multi Screen QA* tasks, we used 10 raters of ages between 25 and 35 based in India. The raters were employed as contractors to help with a variety of data collection tasks apart from this task. They are paid by a fixed hourly wage independent of the number of data annotations. They are instructed to prioritize generating high quality and diverse examples over quantity. They use Chrome as their primary browser and and set their browser locations to the United States. After taking screenshots, they generate the question-answer pair in a text file and upload them along with the screenshots. The authors of the paper review the annotated data every week and provide on-going feedback. The major feedback provided to the raters include the following:

1. The questions should reflect what users commonly try to find out from viewing the websites.

2. Include questions for different arithmetic operations, such as addition, subtraction, comparison and counting.

3. Avoid repeating questions even when the screenshots are different. Please ensure the questions are as diverse as possible.

4. For counting questions, consider adding filtering conditions that you may have in mind while viewing the pages.

5. Avoid common knowledge questions (questions that can be answered without the screenshots).

For the *Trace QA* task, the authors of the paper collected the data. They are based in the United States and Europe. They employ a Chrome Extension plugin to record screenshots, actions, DOM trees and other metadata from their browsing sessions. The extension enables the raters to start a recording and then proceed to browse the web normally. All actions taken by the raters are recorded. The raters can pause or stop the recording at any time. Once the recording is stopped, the raters can review the interaction trace captured in the UI shown in For a visualization of the trace collection user interface, please see Figure 8. Below are the instructions given to the raters for using the Chrome Extension plugin.

1. Install Chrome Plugin Extension.
2. Go to the first web page you want to start the trace.
3. Zoom in or out to adjust screen font sizes.
4. Click on the Chrome Extension emoji and select *Start* to begin recording.
5. Browse the websites with click, type and scroll actions. If you need to pause the recording and resume later, click on the Chrome Extension emoji and select *Pause*.
6. Click on *Stop* on the Chrome Extension to end the recording of the trace.
7. Click on *See Local Sessions* on the Chrome Extension and review the screenshots. Remove the ones that wasn't intended to be added.
8. Enter the question-answer pair.
9. Upload the example.

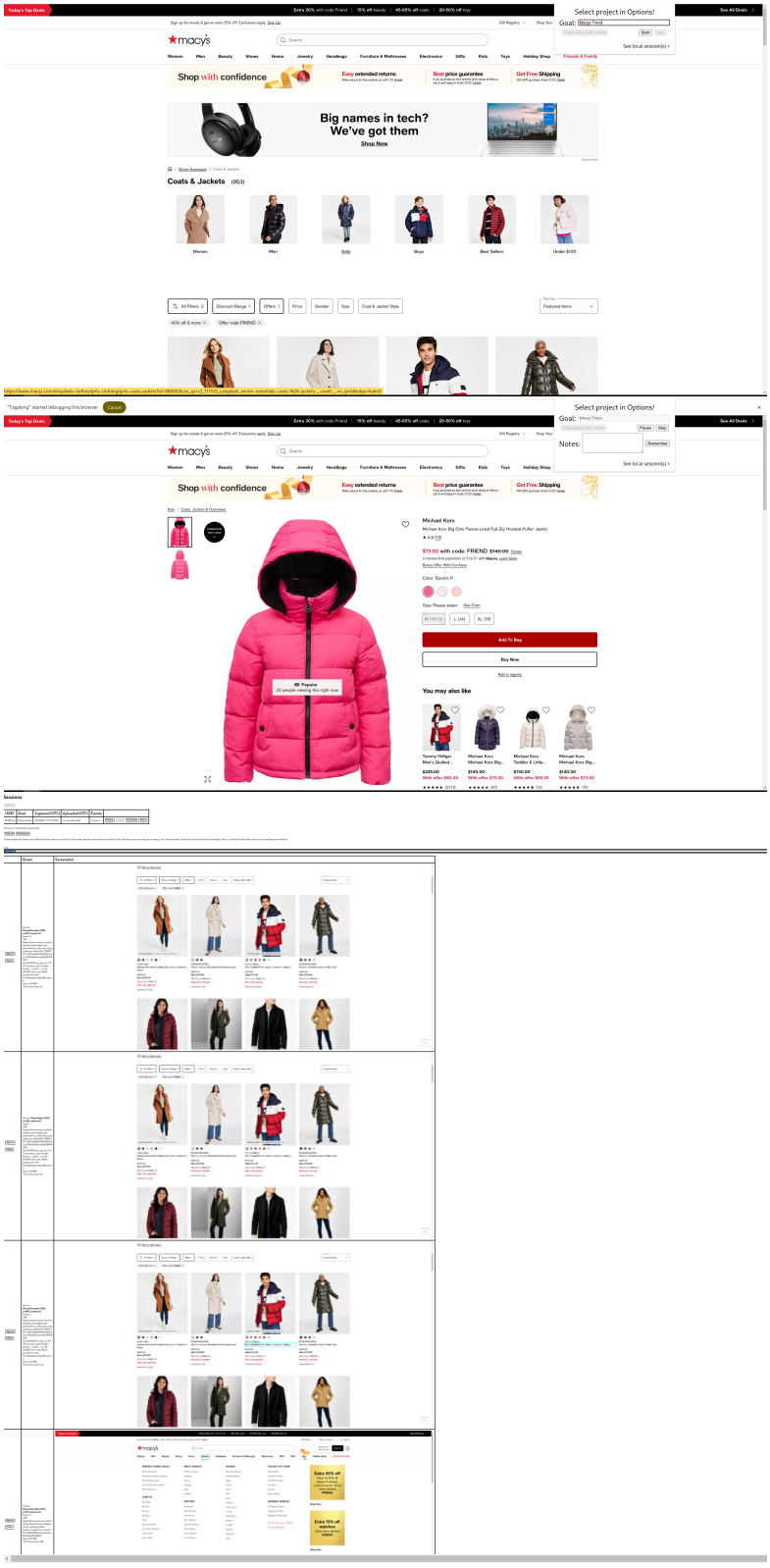


Figure 8. The raters record traces through the chrome extension shown here. They first navigate to the first page of the trace and then click Start. Then, they can either pause or stop the recording of their traces. Finally, they can review the screens collected, delete any unintentionally captured screens and add question-answer pairs for the trace.

C. Prompts

In this section, we show the prompts used for evaluating the different MLLMs. These prompts correspond to the results presented in Table 1.

C.1. Zero-shot prompt

This prompt is used for evaluating the models by directly predicting the final answer. The prompt is as follows:

```
You are given a sequence of screens and
a question. Answer the question using
only the information on the screens.

You should directly tell me your answer
in the fewest words possible, and do not
output any explanation or any other
contents.

Question: <question>
```

C.2. Chain-of-thought prompt

This prompt encourages the models to retrieve the relevant information and then combine it to arrive at the final answer. The prompt is as follows:

```
You are given <num screen> screenshots
and a question. Your goal is to answer
the question according to the screen
information only. Please follow the
below steps to answer the question.

Question:
<question>

(Screenshots Analysis)
First, analyze the contents of each
screenshot and list them here.

(Question Analysis)
What kind of information is needed
from each screenshot to answer the
question?

(Information Extraction)
Now, according to the question and
analysis, please extract the
relevant information from each
screen and list them here.

(Analyze Information)
Based on the question, please
analyze how to answer the question
using the information above.

(Answer)
Please generate the answer with the
information above. Please exclude any
other additional words and information.
```

C.3. Chain-of-thought prompt for open source models

For both Pixtral-12B, Qwen2.5-VL and InternVL-2.5 models, we used a different prompt based on successful prompts used in the respective papers for achieving good results on VQA tasks. The prompt is as follows:

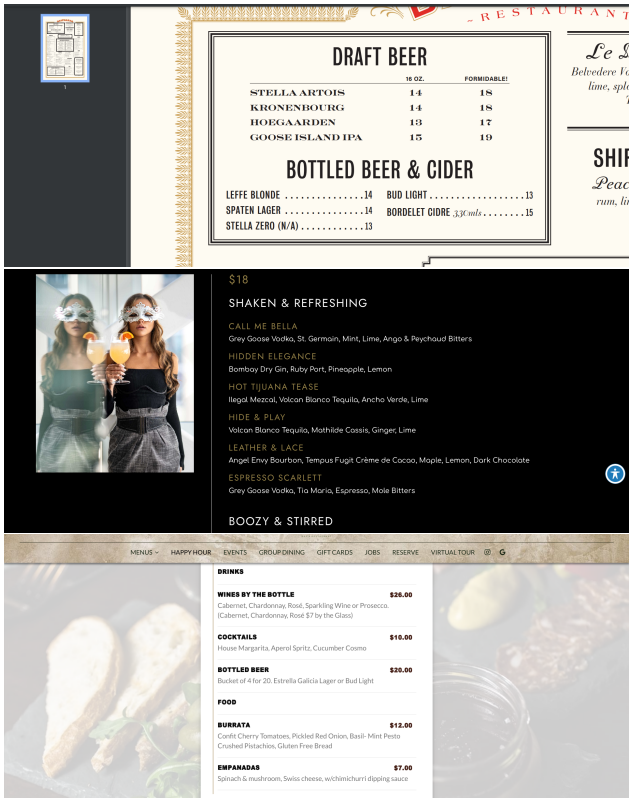
```
Analyze the images and the question
carefully, using step-by-step
reasoning.

First describe the images provided
in detail. Then, present your
reasoning. And finally your
final answer in the format:

Final Answer: <answer>, where
<answer> is a single word,
short phrase or a number.

IMPORTANT: Remember to end your
answer with Final Answer: <answer>.
```

D. Results Analysis Examples

In this section, we further explore the performance for various MLLMs on the different splits of WebQuest. Figure 9 presents a question involving counting across 3 screens where different proprietary models make different errors with all models failing to recognize the color and font differences across categories and constituents of items on the menus. Figure 10 illustrates an example of layout differences across screens of similar functionality causing model prediction errors.

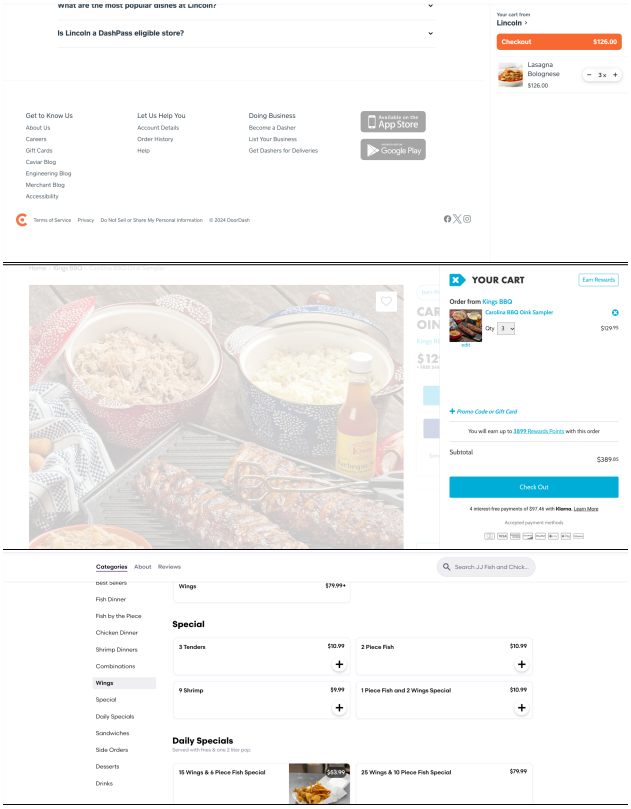
E. Dataset Examples

In this section, we present a few more examples of the WebQuest dataset. In Figure 11, we show the relationship between the 3 different splits Single Screen QA, Multi Screen QA and Trace QA. Then, we illustrate a few examples from each of the splits: Single Screen QA in Figures 12, 13 and 14, Multi Screen QA in Figures 15 and 16 and Trace QA in Figure 17.



Question: Total, how many drinks are shown on these site pages?
Answer: 18

Figure 9. A challenging example that all models failed to answer correctly. The counting task is complicated by the stylistic diversity of the menu. Each models' Chain-of-thought gets it wrong at different steps.



Question: The prices of lasagna and Carolina BBQ Oink Sampler together are how much more than the prices of 3 tenders and 2 pieces of fish together?
Answer: $493.02

Figure 10. UI style differences can confound models even with Chain-of-Thought prompting, despite serving the same checkout functionality of a shopping cart.

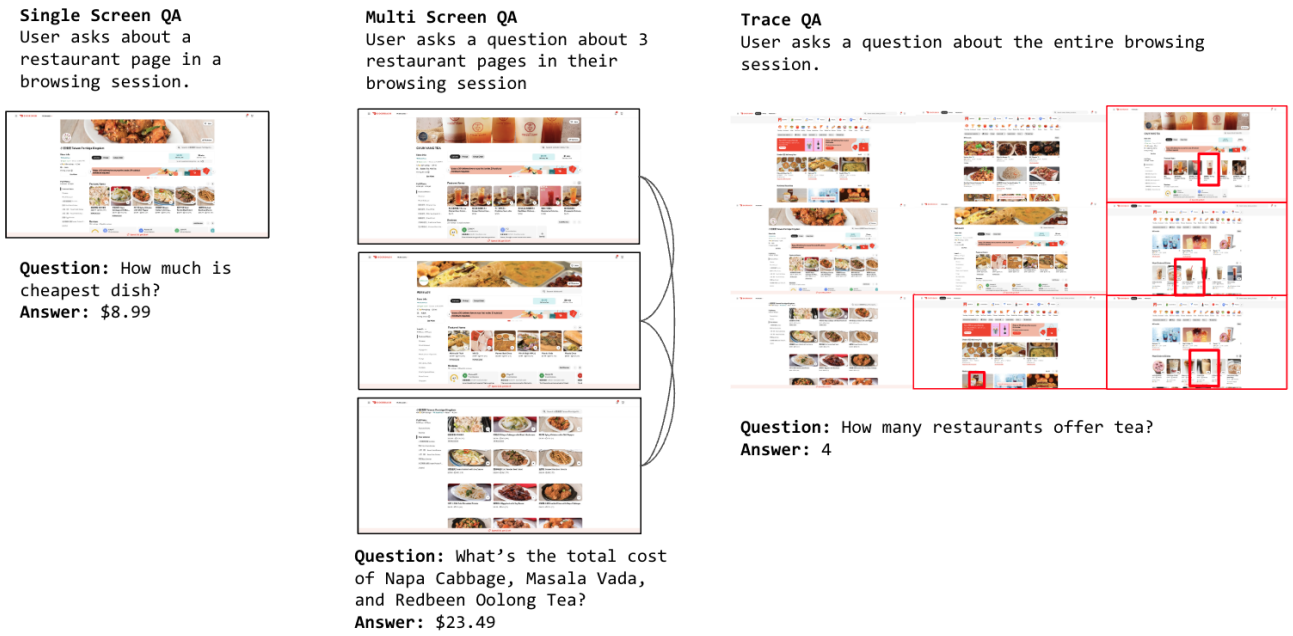
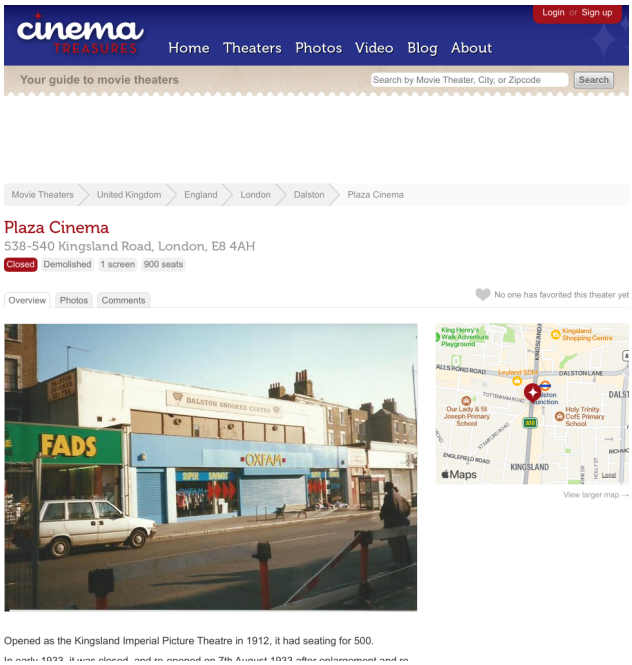


Figure 11. This figure demonstrates the relationships among Single Screen QA, Multi Screen QA, and Trace QA. Differing in both number of screens and types of questions asked, Single Screen QA focuses on a single page, Multi Screen QA focuses on multiple pages within a browsing session, and Trace QA focuses on the entire browsing session. Each task dimension unlocks new types of questions, e.g. from arithmetic and OCR to cross-page reasoning.

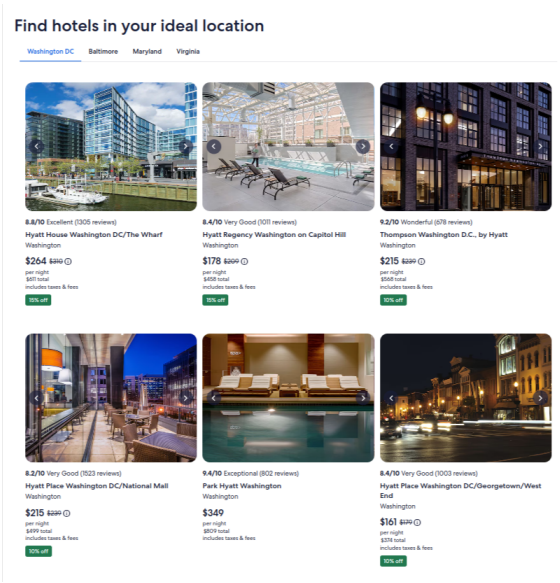


Question: How many more seats does the Plaza Cinema have than it had when it opened?
Answer: 400

Figure 12. The task is to retrieve and compute the difference between the number of seats mentioned in different context of the screen layout.

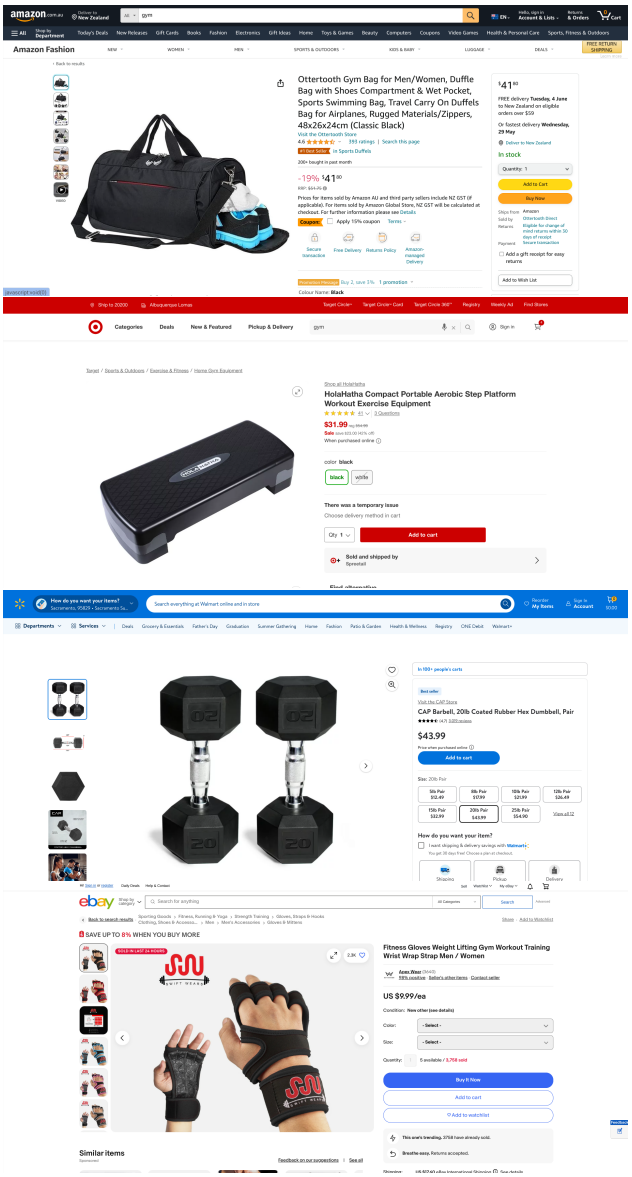
Question: How many days are there between the two gigs?
Answer: 16

Figure 13. The task is to retrieve 2 dates and compute the difference between them in days. Note that the dates are in different colors of font and background, and in different date format. Also note the web page display show two images are missing, which is a common case the QA task is able to handle. We intentionally include screenshots as such.



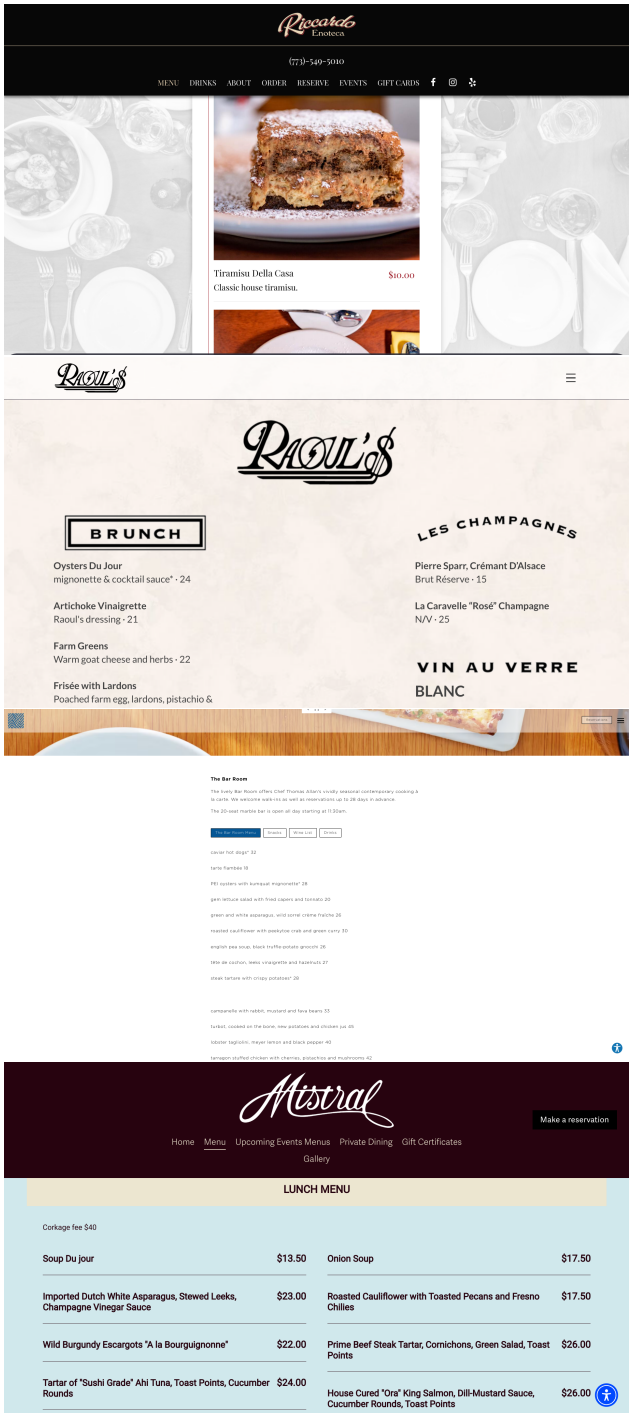
Question: What is the maximum discount?
Answer: 15% off

Figure 14. The task is retrieve all the discounts on the screen and compare them against each other.



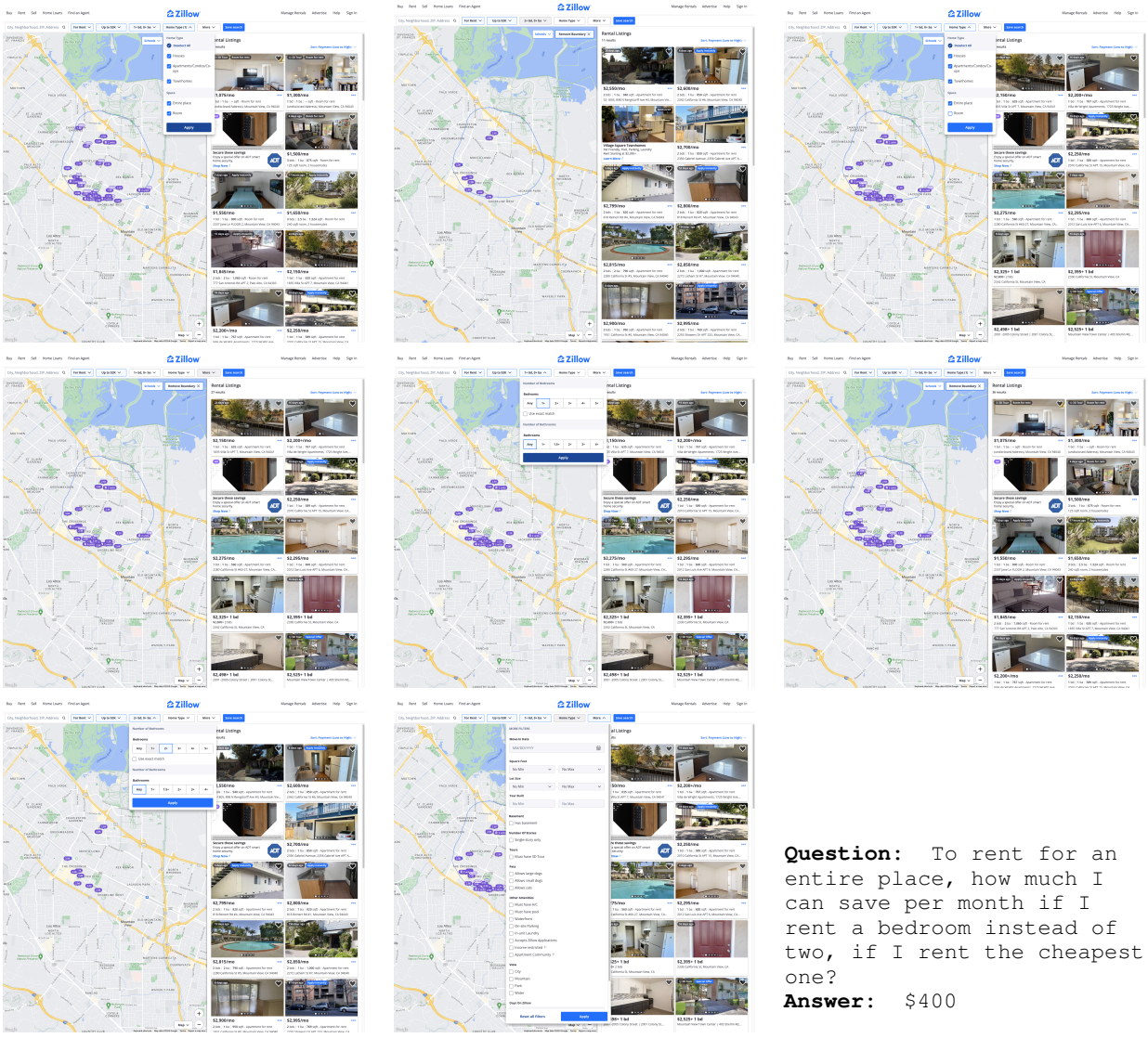
Question: How much do all the gym accessories cost in total?
Answer: $127.77

Figure 15. An example of Multipage QA, about prices of selected gym accessories, each from a different online shop. Notice the layouts and font sizes vary among them.



Question: The sum of the prices of soup do jour, caviar hot dogs, and oysters do jour is how much more than the price of Tiramisu Della Casa?
Answer: $59.5

Figure 16. An example of Multipage QA, about prices of food items over 4 websites of restaurant menus. Notice the layouts and font sizes vary significantly among them.



Question: To rent for an entire place, how much I can save per month if I rent a bedroom instead of two, if I rent the cheapest one?
Answer: $400

Figure 17. An example of Trace QA. The example, situated in a rental exploration scenario, is about price differences between different options. Notice the trace captures the steps taken in the exploration.