E-Web: An Enterprise Benchmark for Web Agents

Senthil Kumaran Kandasamy*, Shyamkrishnan Adissamangalam, Amal Raj, Lakshmi Vanaja Chinnam, Ramya Radhakrishnan, Satendra Choudhary, Jyostna Parasabaktula, Shalini Padhma, and Prasenjit Dey†

Emergence AI

November 2024

Abstract

Intelligent AI agents are increasingly integrated into everyday tasks, both in consumer and enterprise environments. Their ability to autonomously complete tasks and enhance human productivity has led to a diverse ecosystem, leading to agents developed for various specialized capabilities. Web navigation has emerged as a particularly intriguing area, with web agents capable of autonomously navigating, reading, writing, and updating web pages to accomplish specific tasks. However, given the diverse capabilities of these agents, it can be challenging to identify their specific strengths and limitations, highlighting the need for standardized benchmarks to accurately assess and compare their effectiveness. While some consumer-focused benchmarks exist, comprehensive, enterprise-grade benchmarks for real-world, web-scale scenarios remain scarce. In this work, we introduce E-Web, a benchmark designed for evaluating web agents in enterprise contexts. E-Web includes realistic, tool-driven tasks where web agents must navigate one or more web applications to complete a task. Our benchmark follows a skill-centric approach rather than focusing solely on tools or domains, emphasizing core, transferable skills that remain stable across different environments. We believe that by mastering foundational skills, agents can better generalize across varied domains and tools, thus enhancing their overall adaptability. This skill-based benchmark approach addresses a significant gap in evaluation standards, offering insights into how agents perform foundational skills and how these skills affect overall task execution. The benchmark currently comprises 220 prompts covering 15 core skills across 15 enterprise domains and 26 relevant applications or websites. Additionally, we propose a comprehensive evaluation metric that considers both end-to-end task completion and individual skill performance, providing a balanced framework for assessing web agents' capabilities. This is the initial version of our benchmark, serving as the foundation for future enhancements, including the evaluation of advanced capabilities such as API integration, tool execution, and beyond.

1 Introduction

Benchmarking serves as a foundational practice to evaluate and compare the performance of systems in a structured manner—identifying areas for improvement, fostering advancements, and establishing performance standards across domains. For AI agents, benchmarking is especially critical; it measures not only task success rates but also the adaptability, efficiency, and robustness of these agents as they tackle complex, multi-step interactions. The latest advancements in AI agents reflect an exciting shift toward systems that go beyond basic language models, integrating the ability to interact autonomously with tools, APIs, and web pages. These agents represent a more advanced generation of AI, with several recent efforts pushing the boundaries in terms of performance and applications across diverse domains.

*Corresponding author: Senthil Kumaran Kandasamy (senthil@emergence.ai)
†Corresponding author: Prasenjit Dey (prasenjit@emergence.ai)

Web agents are specifically designed to interact with web pages and perform tasks that involve web navigation and data manipulation, enabling automation of various processes. In the consumer realm, web agents can assist with online shopping, social media management, content curation, and even personal finance tasks. In the enterprise context, they facilitate tasks such as customer interactions, project management, content generation, and data retrieval. This versatility underscores their potential to streamline processes across a variety of business and personal applications, making them valuable tools for enhancing productivity and efficiency in diverse scenarios.Early LLM models, such as GPT-2 and BERT, were limited to answering questions and generating text, but modern systems like OpenAI's GPT-4 [1], Anthropic's Claude [2], and others are designed to perform multi-step workflows autonomously. The key trends and advancements driving this evolution are a) Tool and API Integration, b) Web and Interface Navigation, and c) Multi-modal Capabilities.

As the capabilities of web agents [3, 4, 5] rapidly improve, effectively evaluating their performance has become increasingly challenging. Traditional evaluation methods may struggle to keep pace with the fast-evolving landscape especially as agents become more sophisticated and capable of complex tasks. This presents a significant hurdle in ensuring that benchmarks and evaluation criteria remain relevant and useful.

Recent benchmarks have emerged that thoroughly evaluate agents' specific capabilities, particularly their ability to complete tasks in an end-to-end manner [6, 7, 8, 9, 10]. However, there is a notable gap in the literature regarding the assessment of agents' proficiency in handling the fundamental components of these tasks—what we refer to as **atomic skills**. These atomic skills serve as the building blocks for more complex task execution, but they have not been systematically evaluated within existing benchmarks.

In this work, we propose a preliminary approach to developing an enterprise benchmark—**E-Web**—that focuses on capturing these atomic skills. By emphasizing these foundational skills, we aim to enhance understanding of how agents operate at a granular level, ultimately informing the design and improvement of more sophisticated web agents. This approach seeks not only to fill the existing gap in the evaluation landscape but also to provide a framework for understanding the interplay between atomic skills and overall task performance.

Although this work represents just a preliminary effort focused on benchmarking web agents, we plan to build upon this foundation by incorporating additional capabilities such as tool utilization and API calling. By expanding our benchmark to include these advanced functionalities, we aim to capture a broader spectrum of agent capabilities.

2 Related Work

The evolution of web benchmarking has included both server and client-side metrics since the early 1990s. Benchmarks like SPECweb [11] assessed the ability of web servers to handle HTTP requests. As web applications grew more complex, benchmarks such as TPC-C [12] and TPC-W [13] emerged, simulating transaction-heavy environments and web-based commerce, respectively, establishing a foundation for performance evaluation across both client and server interactions.

Browser-side benchmarking has become increasingly significant, particularly as user interactions evolved with the advent of more complex web applications. Early benchmarks primarily measured page load times, but as e-commerce expanded, benchmarks like TPC-W [13] began to assess user experiences through simulated browsing and purchasing processes. This transition highlighted the necessity of evaluating end-to-end task success rates in understanding browser performance in online retail scenarios. With the rise of cloud-based services in the 2010s, client-side benchmarks started to include metrics that reflected interactions with these services. SPEC Cloud [14] emerged to evaluate performance in multi-tenant environments, where client interactions with backend data were critical. This shift allowed

for a better understanding of how well browsers could manage real-time data retrieval and display, aligning performance metrics more closely with user-centric experiences.

Today, agent-based benchmarks represent the latest evolution in client-side evaluation, simulating AI-driven agents navigating web applications. Frameworks like WebShop [15] and WebArena [9] focus on end-to-end task success. The rise of agents has also prompted the development of several benchmarks, each designed to assess different aspects of their performance[3, 16, 17, 9]. Some agents operate on the web, while others access multiple tools or even interact directly with computers[2], leading to a wide range of possibilities. As a result, benchmarks vary in focus—some prioritize task completion or accuracy, while others emphasize adaptability, scalability, or multi-step interactions—making it difficult to establish standardized evaluation methods across all domains and use cases.

At a high level, benchmarks for agents can be categorized along several key dimensions:

1. Enterprise vs. Consumer-Focused Benchmarks: Enterprise-focused benchmarks evaluate agents performing complex, domain-specific tasks, such as managing workflows[16, 10], automating business operations, or integrating with ERP, CRM, and other enterprise tools (e.g., Salesforce or SAP). These agents are expected to prioritize accuracy, compliance, and scalability. Consumer-focused benchmarks, on the other hand, test agents on tasks like personal assistance or e-commerce navigation. [3]

2. Open-Web vs. Simulated/Controllable/Reproducible Sandbox Environments: In open-web environments, agents operate across real, dynamic websites with evolving content and layouts, testing their ability to handle unexpected changes, new features, and real-time interactions. These environments offer the advantage of real-world relevance but are often difficult to control or reproduce, complicating benchmarking[7]. Simulated or sandbox environments provide controlled conditions where interactions can be carefully monitored, recorded, and reproduced. These benchmarks are useful for experiments requiring consistency across runs or for training agents on specific workflows without disruptions caused by live website changes[9].

3. Single-Website vs. Multi-Website Interactions: Single-website benchmarks test agents within the confines of one platform, focusing on their ability to navigate a single web interface, such as filling out forms or managing profiles. These tests assess task-specific proficiency but may not reflect the challenges agents face in broader, cross-platform workflows [3]. Multi-website benchmarks evaluate agents' ability to coordinate tasks across multiple domains or websites—such as booking a flight on one platform, reserving a hotel on another, and syncing calendar entries. These benchmarks test interoperability, multitasking, and cross-context adaptation, reflecting the complexity of real-world scenarios that span multiple services and systems [6].

4. Evaluation Method: Automated vs. Manual Assessment: Benchmarks can be categorized based on their evaluation methods. Automated evaluation relies on predefined metrics and algorithms to assess agent performance, allowing for consistency and scalability in benchmarking. This method is efficient but may overlook nuanced aspects of agent behavior[6]. In contrast, manual evaluation involves human reviewers assessing the agent's outputs, providing insights into qualitative aspects such as user experience, contextual understanding, and error handling. While this method can be more comprehensive, it is resource-intensive and may also introduce variability in assessment[3].

5. Domain-Specific vs. General-Purpose Benchmarks: Domain-specific benchmarks focus on agents designed for particular industries or applications, such as healthcare, finance, or customer service [16, 17]. These benchmarks evaluate how well agents perform within specialized contexts and tasks. In contrast, general-purpose benchmarks assess agents intended to operate across a wide range of topics or tasks, evaluating their versatility and adaptability in diverse scenarios [3].

3 Our Approach

Our benchmark focuses on capturing the fundamental atomic skills required for engaging with a variety of web elements. These interactions are essential for executing more complex workflows, such as form submissions, data entry, and navigation. Additionally, by emphasizing enterprise workflows, our approach addresses the specific needs of businesses, enabling them to automate and optimize tasks that are often overlooked in traditional benchmarks.

3.1 Essential Operations: Read vs Read/Create/Update

Consumer web agents primarily focus on read-heavy tasks, retrieving information from public-facing websites [3]. These agents automate processes such as web scraping, extracting product prices, summarizing articles, or fetching weather updates, with their operations largely confined to reading and navigating content without altering the underlying web pages. In contrast, enterprise web agents engage in dynamic page manipulation and execute complex, state-changing operations. Beyond data retrieval, they perform create, update, and delete (CRUD) operations across various platforms, such as web-based ERPs, CRMs, and SaaS tools. These agents may, for example, fill and submit forms, update records through web interfaces, manage user accounts, or delete obsolete data, integrating deeply with web-based systems to maintain and modify business-critical information. Thus, in constructing our benchmark, we have focused on tasks that involve create and update operations in addition to read only tasks.

3.2 Skill Based vs Task Based Benchmarks

In evaluating web agents, task-based benchmarks and skill-based benchmarks represent two distinct (among other possible) approaches, each with different scopes and objectives. Task-based benchmarks focus on evaluating an agent's ability to complete end-to-end workflows that reflect real-world scenarios. For example, a task might involve filling out a multi-page form, booking a flight, or navigating a checkout process across multiple steps. These benchmarks test the agent's ability to manage complex interactions, including maintaining state, handling dependencies between steps, and adapting to dynamic content. However, they often do not capture how well an agent performs at the atomic level, where the specific interaction with individual UI elements (such as a radio button or date picker) is critical.

While our benchmarks also evaluate agents on workflows, they differ from traditional task-based benchmarks in that we construct these workflows by patching together distinct atomic capabilities. Rather than focusing on high-level task completion alone (e.g., placing an order or booking a flight), our approach emphasizes the modular components that constitute these workflows. Each workflow in our benchmark is decomposable into smaller, skill-based interactions with atomic UI elements, such as selecting a radio button, interacting with a date picker, using a drop-down menu, or dragging and dropping items. A related work [18] emphasizes capturing essential intermediate actions or states necessary for task completion, referred to as key nodes. They developed a framework and evaluation metric that addresses both task completion and these intermediate steps. However, the identified key nodes differ from the atomic skills defined in our research, suggesting that while both approaches evaluate task performance, they focus on different components of the task execution process.

Our approach ensures that the agent's proficiency is evaluated not just on the final outcome of a workflow but on its ability to reliably perform each step that contributes to that outcome. By combining these atomic capabilities into synthetic workflows, we maintain the realism of task-based benchmarks while also delivering fine-grained insights into the agent's core strengths and weaknesses. This method ensures that an agent cannot mask deficiencies in UI-specific skills by relying on broader, high-level heuristics. Moreover, this modular approach makes our benchmarks more adaptable: as new UI elements and interaction patterns emerge, we can easily integrate them into existing workflows, keeping the benchmarks relevant and reflective of modern web interfaces.

3.3 Evaluation

The evaluation of our benchmark operates at two levels: high-level task completion and skill-level performance. At the high level, we assess whether the agent successfully completes the entire task end to end, yielding a pass/fail outcome. If only the task completion is taken into account, the primary overall metric here is the task completion rate: the ratio of successfully completed tasks to the total attempted tasks in the benchmark. This provides a clear measure of the agent's ability to achieve the desired end-to-end outcomes. However, relying solely on this binary metric may overlook cases where the agent performs intermediate steps correctly but fails at certain points, preventing task completion. Given the complexity of many workflows, where multiple atomic interactions are chained together, the failure of one element does not necessarily reflect poor performance across all steps.

To complement the task-level evaluation, we introduce a Skill Execution Score (SES), which measures the agent's ability to execute individual atomic UI interactions within each task. Each workflow is broken down into core interactions, such as filling text fields, selecting from dropdowns, managing pagination, or interacting with date pickers. The SES is defined as the ratio of successfully executed skills to the total number of required skills in a task (e.g., $3/5 = 0.6$ for a task with five skills, three of which were performed correctly). However, a key challenge arises when an agent employs alternative strategies or uses skills not explicitly captured in the benchmark. For example, the agent might bypass an expected dropdown interaction by directly entering a value, or it may automate form submissions in ways not accounted for in the original workflow design. Given the diversity of possible approaches, automatically evaluating such behavior is difficult, as success may not always align with predefined benchmarks. To address this, the SES metric needs to remain flexible, allowing agents to receive partial credit for using unanticipated but valid skills or alternative workflows.

By integrating these two metrics into a combined execution score, we can create a comprehensive evaluation framework that not only captures the agent's success in fulfilling tasks but also highlights its adaptability in utilizing various skills. This approach ensures that agents are rewarded for creativity and adaptability, even when they deviate from the expected paths, while still identifying areas for improvement at a granular skill level. We propose a **Combined Execution Score** metric per task:

$$\mathcal{T}_{\text{exec}} = \frac{w_{\text{E2E}} \cdot \mathbb{1}(\mathcal{E}_{\text{E2E}}) + \sum_{i=1}^{n} w_i \cdot \mathbb{1}(\mathcal{S}_i)}{w_{\text{E2E}} + \sum_{i=1}^{n} w_i}$$

Where,

- $\mathcal{T}_{\text{exec}}$: Combined execution score (ranging from 0 to 1).
- $\mathcal{E}_{\text{E2E}}$: End-to-end task execution score (1 if successful, 0 otherwise).
- $\mathcal{S}_i$: Partial skill execution score for skill i (1 if successful, 0 otherwise).
- w_{E2E}: Weight assigned to the end-to-end score.
- w_i: Weight assigned to skill i.
- $\mathbb{1}(x)$: Indicator function that is 1 if x is true (successful) and 0 otherwise.

We can determine the weights for both end-to-end task completion and individual skills using heuristics. Specifically, the weight for each skill can be proportional to its relative difficulty, ensuring that more complex skills contribute more to the overall evaluation. These weights can also be learned dynamically, leveraging data from agent performance over time. This adaptive weighting system allows the framework to better reflect real-world conditions, where some skills may require more effort and expertise. Fine-tuning these weights ensures a balanced assessment that fairly evaluates both holistic task success and granular skill execution.

3.4 Example Task: Create new opportunities in Salesforce

Prompt:

Create new opportunities for the following accounts, but only if their current opportunities are in the 'Closed - Lost' stage: 1. 'Global Enterprises', contact: 'Jane Smith', opportunity value: $300,000, product: 'GC5000 series', close date: July 10, 2025. 2. 'Future Tech Solutions', contact: 'Michael Brown', opportunity value: $175,000, product: 'GC3000 series', close date: August 15, 2025.""

The task involves creating new opportunities for two specified accounts if their current opportunities are marked as "Closed - Lost." The skills required for this task include:

- Text Fields: Inputting or modifying account details, such as names and contact information.
- Dropdown Menus: Selecting relevant options for opportunity status or product type.
- Date Pickers: Choosing specific close dates for the new opportunities.
- Buttons: Executing commands to save or submit the new opportunity entries.

4 Design and Methodology

The framework for our benchmark is hierarchically structured into three levels: tool type or domain, tool/application/website, and skills.

- **Tool-Type/Domain**: This is the broadest category, encompassing various domains such as CRM (Customer Relationship Management), SaaS (Software as a Service), project management tools, and more. Each type serves specific enterprise needs.
- **Tool/Application/Website**: Within each domain, specific applications like Salesforce, JIRA, or Microsoft Teams are identified. These tools are commonly used in business contexts and reflect real-world usage.
- **Skills**: At the most granular level, we define the essential skills required for interaction, which include UI elements like date pickers, sliders, checkboxes, and dropdown menus. These atomic skills are the foundational components that enable users to perform complex tasks within the applications.

This hierarchical framework not only organizes our benchmark but also allows for targeted evaluation of agent capabilities, ensuring a comprehensive understanding of how well agents can navigate and manipulate various tools. Since our benchmark will primarily focus on enterprise tasks and workflows, our methodology to design the benchmark broadly consisted of the following steps:

1. Identifying the key business functions, domains and tools
2. Identifying the key atomic web skills
3. Prompt generation and dataset refinement

4.1 Domain and Tool Selection

Our benchmark development began with a focus on enterprise tasks, selecting key business functions such as collaboration and communication, project and task management, ERP, CRM, and business analytics as the core scope. To operationalize this, we identified a few representative and widely used applications and websites, such as Salesforce, Microsoft suite, and Atlassian suite, among others. A comprehensive list of platforms used is provided in Table 1.

Tool Type	Website
Accounting	Xero
CRM	Salesforce, JIRA
Cloud Storage and File Sharing	Dropbox, Google Drive
Code Collaboration	Github
Collaboration and Communication	Teams, Outlook, Google Calendar, Slack, Confluence
Customer Support	Zendesk
Digital Transaction Management	Docusign
Expense Management	Expenisfy
Graphic Design	Canva
Knowledge Management	Confluence
Online Survey and Data Collection	Google Forms
Productivity Suite	Microsoft - Excel, PowerPoint, Word, Outlook, Viva, Forms, Sharepoint
Project Management	JIRA, Asana, Trello
Social Media	LinkedIn
Visual Collaboration and Whiteboarding	Miro

Table 1: Domains/Tool-Types and corresponding Tools/Websites

4.2 Skill selection

Once the relevant domains and tools were selected, our next step was to identify the fundamental atomic skills that form the building blocks of typical web interactions. These atomic skills represent the basic actions that web agents must master to perform more complex workflows. They include common UI elements such as text fields for data entry, autocomplete for predictive text, date pickers for selecting dates, checkboxes for multi-option selections, buttons for triggering actions and drag-and-drop or reorder operations, which are often used for organizing items or prioritizing tasks.

The comprehensive list of these atomic skills is presented in Table 2. While not exhaustive, this selection of skills captures the most frequently encountered UI interactions in modern web applications. By focusing on these core elements, our benchmark ensures that web agents are evaluated on essential tasks that underpin both consumer and enterprise workflows. Mastery of these atomic skills is critical, as they collectively enable agents to perform more advanced operations across diverse platforms.

Skill Type	Description
text fields	Areas where users input free-form text.
dropdown menus	Menus with selectable options displayed on click.
date pickers	Widgets for selecting specific dates.
buttons	Clickable elements triggering actions or events.
pop-ups	Temporary overlays with additional content.
file upload	Interfaces to upload files from local storage.
search box	Fields to enter queries for searching content.
toggle switches	Switches to enable or disable settings.
drag drop and reorder	Elements rearranged by dragging and dropping.
checkboxes	Boxes to select multiple options.
radio buttons	Circular options allowing single selection.
nested menus	Hierarchical menus with multiple levels.
sliders	Controls to adjust values across a range.
autocomplete	Input fields with suggested completions.
pagination controls	Navigate through multi-page content.

Table 2: Descriptions of different skill types used in web agents. We have considered a total of 15 skills.

4.3 Prompt generation

After identifying the relevant domains, tools, and atomic skills, we proceeded to manually curate a set of realistic prompts designed to simulate typical multi-step web tasks. Each task prompt was carefully crafted to involve several steps, with each step corresponding to one or more atomic skills (such as

entering data in text fields, interacting with dropdown menus, using sliders, etc.). These prompts aim to reflect the types of complex interactions web agents would need to perform in real-world scenarios, especially within enterprise settings, ensuring that the agents are tested on both their individual skills and their ability to integrate them into workflows.

The generation of prompts was primarily performed by our in-house team of data annotators, who possess practical experience with both enterprise workflows and various web platforms. In the initial experimental phase, each annotator was assigned the task of generating multiple prompts for specific skills across a variety of tools and websites. This process allowed us to identify patterns regarding which tools or workflows commonly required particular primitive skills, such as date pickers or search boxes. With these insights, we proceeded to design more complex workflow-based prompts by combining relevant skills, simulating real-world scenarios encountered in enterprise settings. This structured approach ensured that the benchmark captured both fundamental interactions and intricate, task-specific workflows, Once the prompts were generated (each with one or many underlying skills), they underwent a validation stage. Three independent validators reviewed the prompts to confirm that they indeed invoked the intended atomic skills. This multi-step process ensured both accuracy and alignment, validating that the tasks were realistic and that the correct UI elements were accessed. By following this approach, we ensured that the curated prompts are representative of the skills that web agents are expected to demonstrate.

Given the enterprise focus of our benchmark, certain prerequisites must be met to ensure that some of the prompts function effectively. Specifically, access to various tools may necessitate valid credentials or accounts. This requirement is particularly pertinent for proprietary applications and services, such as Salesforce or JIRA, where user authentication is critical to accessing features and functionalities essential for task completion. Moreover, certain prompts may require additional resources, including specific usernames, configurations, or even particular files, to yield meaningful results. This complexity is a stark contrast to consumer benchmarks, which typically involve tasks accessible via the open web and do not require special permissions or resources. For every task in the benchmark, we capture the prerequisites.

The implications of these prerequisites extend beyond mere access; they influence the reproducibility and scalability of the benchmark. For instance, the dependency on specific user accounts can introduce variability in performance results, as different users might experience different levels of access or functionality based on their role within an organization. Hence, while our benchmark aims to capture the nuances of enterprise applications, it is essential to acknowledge these challenges, as they can impact both the design and the outcomes of the evaluation process.

4.4 Evaluation

We performed very preliminary evaluation of the benchmark using Three agents - Agent-E -[4], Multion [5, 19]. and Claude-3.5 with computer-use capabilities. [2].

5 Results and Discussion

5.1 Analysis of the Benchmark

The highest level details of the benchmark are presented in Table 3. We have a total of 220 prompts, across 15 domains and 26 websites, with 15 underlying skills. A few representative prompts are provided in the appendix A. We are also making public a representative subset (65 prompts) available on Github.

In Figure 1, we present the distribution of essential operations—create, update, and read—across benchmark prompts. Unlike consumer applications, where read operations dominate, our dataset contains a more balanced mix. Specifically, over 60% of the prompts involve at least one update operation, and nearly 70% feature at least one create operation. Most prompts involve a combination of read, update and create operations. We have not created any delete operations in this benchmark dataset, and that

Category	Count
Domains	15
Applications/Tools/Websites	26
Skills	15
Prompts	220

Table 3: Overview of Benchmark Composition and Key Statistics

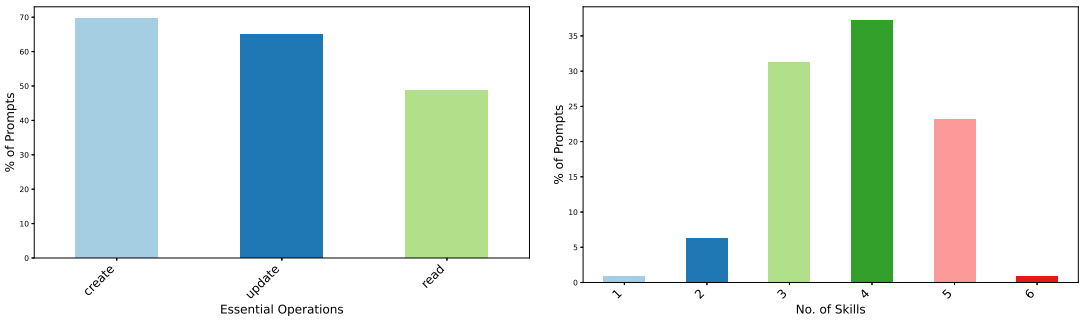

Figure 1: Distribution of essential operations, skill types and number of skills per task

would be a task for future efforts. Having a mixture of create, read and update operations demonstrates the practical focus of the benchmark, reflecting real-world enterprise workflows that emphasize dynamic task execution. The inclusion of complex operations challenges agents to perform beyond mere data retrieval, highlighting their adaptability and deeper functional capacity. In this figure, we also illustrate the distribution of the number of skills required per task. The minimum skill count for any prompt is 1, while the most complex tasks require up to 6 skills. The median number of skills per prompt is 4, with an average of approximately 3.8. This distribution highlights the variability in task complexity, reflecting both simple and multi-step workflows.

In Figure 2 we present the distribution of websites and skill types across the prompts. The data exhibits a roughly Pareto-like distribution, with a small set of core skills (such as text fields, dropdown menus, date pickers, and buttons) appearing frequently, while others are less common. Similarly, most prompts are concentrated around key platforms like Teams, JIRA, and Salesforce, reflecting the focus on enterprise workflows. This skewed distribution highlights both the prevalence of essential interactions and the importance of specific platforms in real-world business operations. For the websites, a better distribution of prompts (ensuring more balanced coverage across various tools) will be explored as part of future work. This improvement would aim to reduce the heavy reliance on a few platforms, creating a more diverse and representative benchmark. Expanding the dataset to include more tools will enhance the generalizability of the agents' performance.

To further illustrate the distribution of skills across various tool types, we present a heatmap matrix in Figure 3. This visualization indicates that there is significant potential for improvement in distributing skill types more evenly across different tool categories. Such enhancements would contribute to better generalization of agent performance and ensure a more comprehensive evaluation of their capabilities across diverse applications. Similarly, the distribution of skills across websites, as shown in Figure 4, indicates that there is room for improvement in skill distribution. This unevenness suggests a need for further exploration and refinement in future work to ensure a more balanced representation of skills across different websites.

5.2 Performance of various agents

Our preliminary evaluation of several agents [4, 19, 2] on this benchmark has demonstrated its significantly higher difficulty level compared to existing benchmarks like WebVoyager [3], particularly in end-to-end

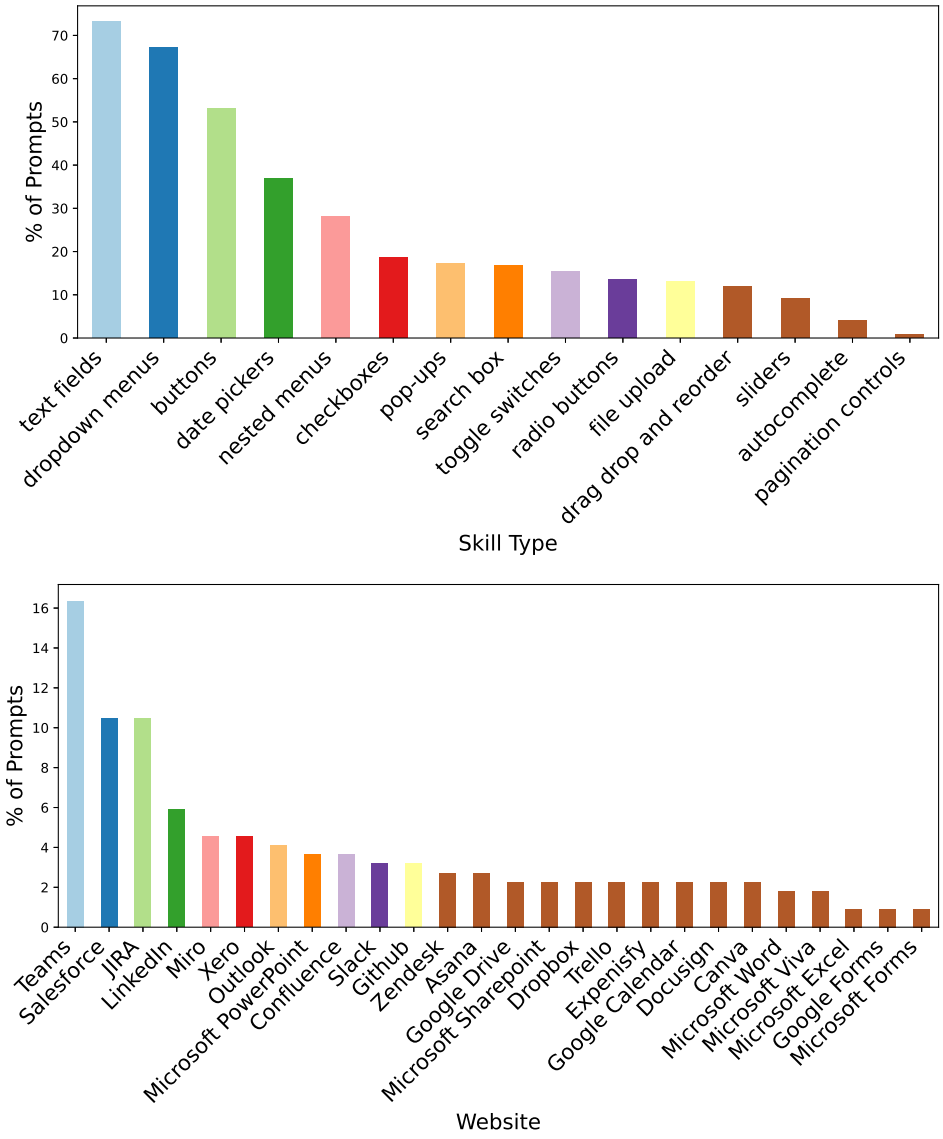

Figure 2: Distribution of Skills and Websites across prompts.

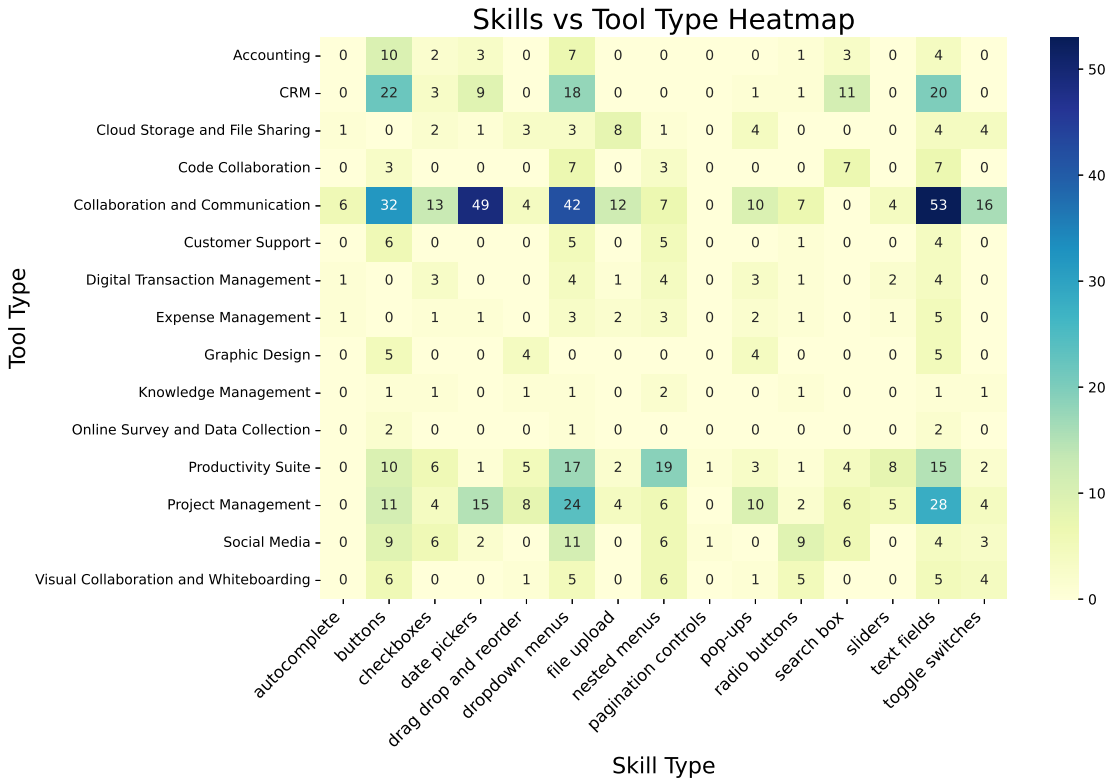

Figure 3: Number of skills for each tool type.

task completion metrics. The benchmark's design emphasizes both complex, multi-step tasks and fine-grained skill assessments, making it a more rigorous test of agent capabilities. Moving forward, we plan to expand our evaluations to include additional agents, measuring their overall task completion rates and individual skill execution scores. This dual approach will provide a clearer picture of each agent's proficiency in both holistic workflows and specific skill areas, guiding improvements in agent development and performance metrics.

We found that certain skills, such as filling text fields or selecting checkboxes, are significantly easier for agents to manage compared to more complex tasks like using date pickers. Ongoing evaluations and in-depth analyses of skill performance will contribute to a better understanding of the inherent difficulty levels associated with different skills. This knowledge will be instrumental in developing improved metrics for our evaluation framework, enhancing its overall effectiveness in assessing agent capabilities.

6 Summary and Future Work

In this work, we introduce the first version of a web agent benchmark **E-Web** that evaluates agent performance based on specific skills. We summarize the comparison of key aspects of our benchmark with existing benchmarks in Table 4. Our findings indicate that the benchmark is notably challenging, thereby providing a robust foundation for further research in agent capabilities. This benchmark can facilitate the development of more sophisticated agents by identifying areas for improvement. In our analysis, we observed a significant variance in the difficulty levels associated with different skills required by web agents. For instance, fundamental tasks like filling out text fields or selecting checkboxes seem to be relatively straightforward, whereas tasks such as date picking present inherent challenges. This disparity highlights the need for tailored training strategies to enhance agent capabilities across the skill spectrum.

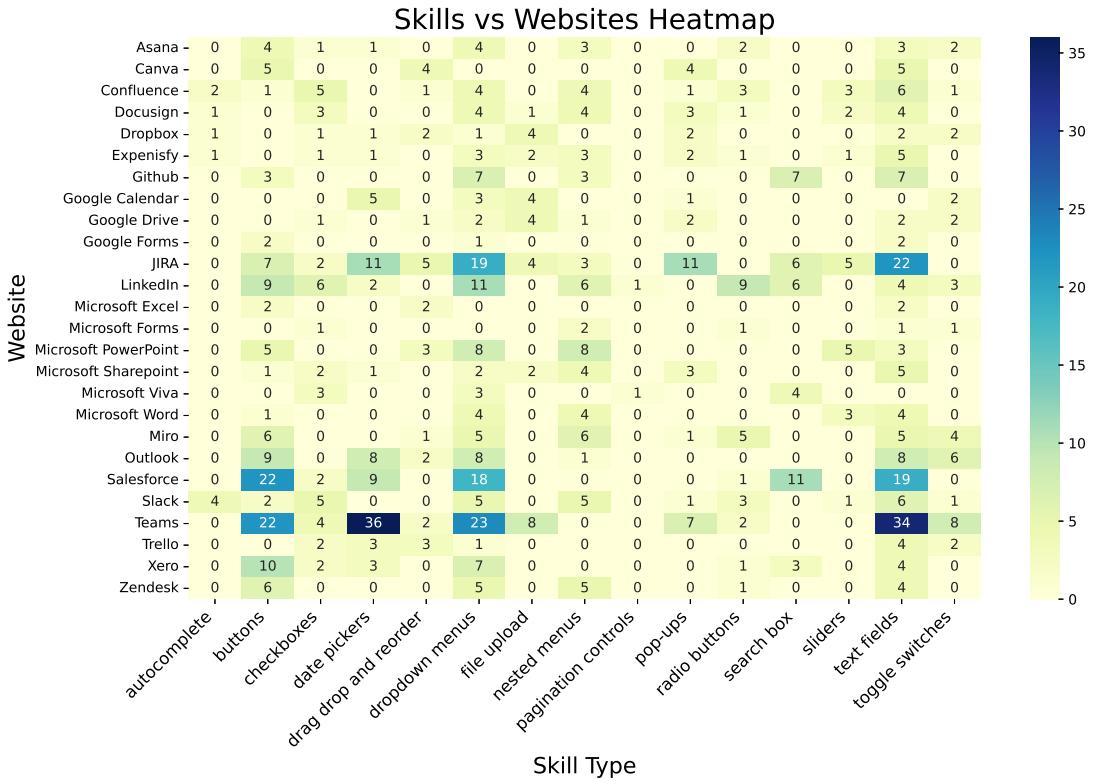

Figure 4: Number of websites for each skill

Benchmark	Online Evaluation	Live Web	Enterprise Workflows	Skill-Based	Evaluation Metric	Tasks	Enterprise tools
WebVoyager	x	✓	x	x	Success Rate	643	-
Mind2Web	✓	x	x	x	Success Rate	2350	-
WorkArena	✓	x	x	x	Success Rate	29	-
WebArena	✓	x	x	x	Success Rate	812	-
WebCanvas	✓	x	x	Partial	Success Rate, Key Nodes	542	-
ST-WebAgentBench	✓	x	x	x	Success Rate, CuP, Risk	234	-
Spider2-V	✓	x	✓	x	Success Rate	494	20
E-Web	x	✓	✓	✓	Success Rate, Skill Execution Score, Combined Execution score	220	26

Table 4: Comparison of Web Benchmarking Frameworks

Future directions include expanding the dataset to achieve more robust distributions across domains and tools, ensuring it reflects a wider variety of real-world applications. Additionally, we aim to introduce multi-step, cross-platform workflows, which represent scenarios where agents interact with multiple tools and services to complete complex tasks. These scenarios are crucial because modern enterprise tasks often span multiple platforms, requiring seamless transitions between different applications and coordinated use of various atomic skills. By evaluating such capabilities, we can assess how well agents can handle dependencies, manage state across applications, and recover from errors when switching contexts. Many tasks that can be performed via web interfaces can just as effectively be accomplished using API endpoints, providing greater flexibility in execution. To ensure fair and meaningful evaluations, we will develop benchmarks that focus on outcomes rather than the method of completion, remaining agnostic to whether a task is executed through a web interface or an API. This approach aligns with

the rapidly evolving capabilities of agents, which increasingly extend beyond the web, enabling more versatile and powerful solutions.

If skills can operate independently, agents may be flexible enough to reorder tasks based on context. However, if task success depends on a specific skill sequence—such as navigating to a specific page before interacting with certain elements—then sequence integrity becomes crucial. This implies that benchmarks should test not only skill execution but also skill chaining. Tool-specific dependencies are also significant. For example, different enterprise tools may implement similar functions in distinct ways, potentially altering skill execution paths for an agent. Dependencies in tool design and UI layout may require customizations in how an agent performs even routine tasks, which is especially relevant in settings with highly varied software interfaces. As part of future work, testing these aspects can improve our understanding of agents' adaptability across workflows, setting the groundwork for more sophisticated evaluation metrics that address sequence sensitivity and platform-specific nuances.

An inherent challenge associated with enterprise workflows is the existence of potential prerequisites to achieve a specific task. Another future direction is to implement a two-step mechanism for benchmark evaluation to enhance usability and user experience. The first step will involve prompts to help users set up prerequisites, such as creating accounts or preparing specific data. The second step will involve executing the main benchmark evaluation, on tasks that assess web agent performance using the previously established accounts and data. By segmenting the benchmark into setup and evaluation phases, we can create a supportive environment that encourages successful interactions and leads to more reliable assessments of agent performance, thereby reducing friction and facilitating easier adoption in future implementations.

Key Takeaways

- Benchmarking provides a systematic way to measure, compare, and improve the performance of AI agents, setting a clear standard for development and advancement. Evaluating AI Web agents presents unique challenges due to their variability in task execution, environment sensitivity, and reliance on task structure. Key difficulties include quantifying multi-step workflows, assessing adaptability to different contexts, and ensuring fairness across diverse tools and applications. These challenges make it essential to develop metrics that accurately reflect agent capabilities in both skill execution and end-to-end task fulfillment.
- Our benchmark - **E-Web** - introduces an innovative evaluation framework that assesses web agents on both skill-specific and task-based criteria. The framework distinguishes between atomic skills (e.g., data entry, button-clicking) and full task completion (e.g., navigating a multi-step workflow), providing a layered understanding of an agent's performance. A combined metric captures both dimensions, assessing agents on precise skill execution and holistic task outcomes.
- The benchmark emphasizes enterprise-relevant tools, ensuring that AI agents can handle essential applications like project management and customer relationship management. With broad skill coverage, the benchmark represents the core skills needed for professional environments, making it an effective measure of an agent's readiness for enterprise tasks and practical deployment.
- Future broader directions include expanding the scope of the benchmark to evaluate tasks across a wider range of interfaces, including APIs and tools that go beyond web interfaces.

7 Author Contributions

- Conceptualization and Study Design - SKK and PD

- Early Explorations and Prompt Generation - SA, AR, LVC, RR, SC, JP and SP
- Data Curation and Benchmark Development - SA, AR and LVC
- Annotations and Validations - AR, LVC, RR, SC, JP and SP
- Analysis - SA and SKK
- Writing – Original Draft and Revisions - SKK and PD
- Supervision and Project Management - SKK and PD

8 Disclosure

AI language models, primarily Chat-GPT, were used in assistance with text refinement and editing of this document. The contributions of these tools are acknowledged, and all content has been reviewed by the authors to ensure accuracy and relevance.

9 Acknowledgements

We thank Nimish Soni and Ravi Kokku for valuable discussions and feedback.

References

[1] OpenAI, "How openai's gpt-4 uses plugins to enhance its capabilities," 2024, accessed: 2024-10-29. [Online]. Available: https://www.openai.com/research/gpt-4-plugins

[2] Anthropic, "Introducing computer use, a new claude 3.5 sonnet, and claude 3.5 haiku," https://www.anthropic.com/news/3-5-models-and-computer-use, accessed: 29 October 2024.

[3] H. He, W. Yao, K. Ma, W. Yu, Y. Dai, H. Zhang, Z. Lan, and D. Yu, "Webvoyager: Building an end-to-end web agent with large multimodal models," 2024. [Online]. Available: https://arxiv.org/abs/2401.13919

[4] T. Abuelsaad, D. Akkil, P. Dey, A. Jagmohan, A. Vempaty, and R. Kokku, "Agent-e: From autonomous web navigation to foundational design principles in agentic systems," 2024. [Online]. Available: https://arxiv.org/abs/2407.13032

[5] P. Putta, E. Mills, N. Garg, S. Motwani, C. Finn, D. Garg, and R. Rafailov, "Agent q: Advanced reasoning and learning for autonomous ai agents," 2024. [Online]. Available: https://arxiv.org/abs/2408.07199

[6] O. Yoran, S. J. Amouyal, C. Malaviya, B. Bogin, O. Press, and J. Berant, "Assistantbench: Can web agents solve realistic and time-consuming tasks?" 2024. [Online]. Available: https://arxiv.org/abs/2407.15711

[7] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su, "Mind2web: Towards a generalist agent for the web," 2023. [Online]. Available: https://arxiv.org/abs/2306.06070

[8] S. Shlomov, B. wiesel, A. Sela, I. Levy, L. Galanti, and R. Abitbol, "From grounding to planning: Benchmarking bottlenecks in web agents," 2024. [Online]. Available: https://arxiv.org/abs/2409.01927

[9] S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, T. Ou, Y. Bisk, D. Fried, U. Alon, and G. Neubig, "Webarena: A realistic web environment for building autonomous agents," 2024. [Online]. Available: https://arxiv.org/abs/2307.13854

[10] O. Styles, S. Miller, P. Cerda-Mardini, T. Guha, V. Sanchez, and B. Vidgen, "Workbench: a benchmark dataset for agents in a realistic workplace setting," 2024. [Online]. Available: https://arxiv.org/abs/2405.00823

[11] S. P. E. Corporation, "Specweb 99," 1999, accessed: 2024-10-31. [Online]. Available: https://www.spec.org/web99/

[12] T. P. P. Council, "Tpc-c benchmark," 2000, accessed: 2024-10-30. [Online]. Available: https://www.tpc.org/tpcc/default5.asp

[13] D. Menasce, "Tpc-w: a benchmark for e-commerce," *IEEE Internet Computing*, vol. 6, no. 3, pp. 83–87, 2002.

[14] S. P. E. Corporation, "Spec cloud iaas 2018," 2018, accessed: 2024-10-31. [Online]. Available: https://www.spec.org/cloud_iaas2018/

[15] S. Yao, H. Chen, J. Yang, and K. Narasimhan, "Webshop: Towards scalable real-world web interaction with grounded language agents," 2023. [Online]. Available: https://arxiv.org/abs/2207.01206

[16] R. Cao, F. Lei, H. Wu, J. Chen, Y. Fu, H. Gao, X. Xiong, H. Zhang, Y. Mao, W. Hu, T. Xie, H. Xu, D. Zhang, S. Wang, R. Sun, P. Yin, C. Xiong, A. Ni, Q. Liu, V. Zhong, L. Chen, K. Yu, and T. Yu, "Spider2-v: How far are multimodal agents from automating data science and engineering workflows?" 2024. [Online]. Available: https://arxiv.org/abs/2407.10956

[17] I. Levy, B. Wiesel, S. Marreed, A. Oved, A. Yaeli, and S. Shlomov, "St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents," 2024. [Online]. Available: https://arxiv.org/abs/2410.06703

[18] Y. Pan, D. Kong, S. Zhou, C. Cui, Y. Leng, B. Jiang, H. Liu, Y. Shang, S. Zhou, T. Wu, and Z. Wu, "Webcanvas: Benchmarking web agents in online environments," 2024. [Online]. Available: https://arxiv.org/abs/2406.12373

[19] Multion, "Multion playground," https://platform.multion.ai/playground, accessed: 29 October 2024.

Appendix

A Representative prompts

Below, we present a few representative prompts from our benchmark dataset

1. **Prompt:** Adjust the probability of closing for the opportunity at Gamma Corp (contact: 'Linda Taylor') to 70%. The opportunity is for the 'GC5000 series', currently in the 'Negotiation' stage .

   Tool: Salesforce
   Skills: Text Fields, Dropdown Menus, Buttons

2. **Prompt:** Create a new Bug in the Agent AI Playground project titled 'User Profile Loading Error'. Description: Users encounter an error when loading their profiles. Set the Priority to High and the Due Date for November 15, 2025. .

   Tool:Jira
   Skills: Dropdown Menus, Text Fields, Date Pickers, Buttons

3. **Prompt:** Schedule a webinar titled 'Future of AI in Business' for March 10, 2025, at 2 PM. Set the duration to 1 hour and configure it to allow attendees to join via phone. Disable the Q and A feature. Enable registration and set it to Your Organization access. .

   Tool:Teams
   Skills: Text Fields, Date Pickers, Dropdown Menus, Radio Buttons, Toggle Switches, Buttons

4. **Prompt**: Create an event on LinkedIn titled "Holistic Health Workshop." The event will be virtual and is scheduled for 10th November in the UK time zone. This event will end on 10th November, and the comment control should be set to "No one." .

   Tool: Linkedin
   Skills: Text Fields, Dropdown Menus, Radio Buttons, Date Pickers, Buttons

5. **Prompt**: Move all files related to "2024 Conference Materials" into a new folder titled "Archived Events." Disable shared access and enable version history. .

   Tool: Dropbox
   Skills: Drag Drop and Reorder, Toggle Switches, File Upload