# Cook Browser: Completing Work on the Live Web

**Austin Worthington**

Client Acquisition.io

trycook.ai/research

**Abstract**

We evaluated Cook Browser through the Cook desktop application on three browser-agent suites: Online-Mind2Web, BU Bench V1, and Odysseys. In September 2026 we ran the complete BU Bench V1 (100 tasks) and Odysseys (200 tasks) suites and graded them with the official graders published in Aside's benchmark repository. Counting each task's best graded run, Cook passed 94 of 100 BU Bench V1 tasks (94.0%) and met 95.6% of the Odysseys rubric items; on each task's first graded run the results were 84.0% and 70.8%. The Online-Mind2Web result, 93.4% after the cohort adjustment described below, is the August 2026 record. The work tested whether an agent can navigate live websites, retain the requested target, and return the result of a multi-step task. (Cook, 2026)

The research also produced concrete engineering improvements: stronger long-horizon persistence, explicit tab lifecycle management, correct final-answer extraction, and preservation of rich browser results. These mechanisms address failures in executing work, rather than only changes to the wording of the final response.

Cook.ai

## 1 Introduction and scope

Cook completed 268 tasks in the recorded 300-task Online-Mind2Web cohort. Its adjusted score excludes 13 tasks classified as impossible; its full-cohort score is 89.3%. The accepted result merges a baseline with targeted reruns. It is a best-known recorded result, not a fresh single-pass estimate or an all-benchmark state-of-the-art claim.

Report scope: the August 2026 evaluation record for Online-Mind2Web, and full-suite runs of BU Bench V1 and Odysseys executed on September 23 and 24, 2026 for this revision. External results are attributed to their publishers and presented as context.

## 2 Architecture: closing the loop between intent and evidence

A browser agent works in a changing environment. A page can load slowly, a new tab can become active, or a button can produce an intermediate state rather than the requested result. The useful unit of work is therefore a continuing relationship between a person's intent, the current page, and evidence of progress. Our architectural emphasis is on preserving that relationship across a sequence of actions.

### 2.1 Keep the intended target attached to the action

A browser can contain several similar pages at once. Selecting the right action is not enough if that action reaches the wrong tab or page. Requested-target binding makes the target part of the task's execution context, rather than leaving it as an assumption about whichever page happens to be active. The recorded benchmark work includes an explicit correction to this behavior. (Cook, 2026)

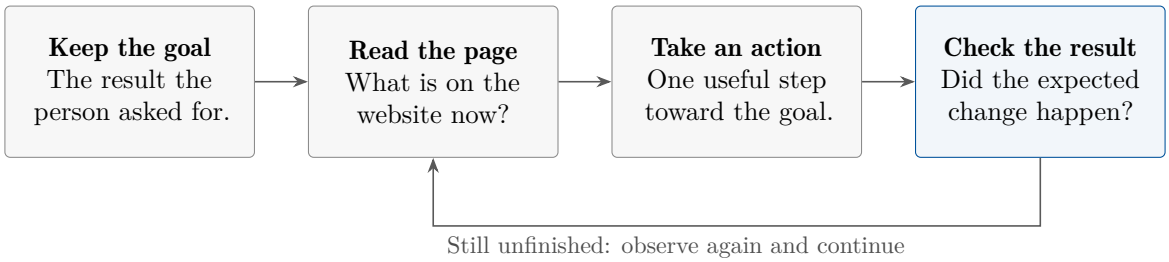



**Figure 1:** The browser task as a feedback loop. In plain English: keep the request in mind, look, act, and check. The diagram summarizes the architectural idea; it does not expose the execution policy.

For a nontechnical analogy, imagine asking an assistant to update a particular customer's record while several customer records are open. The instruction "change the address" is only useful when the assistant also retains *whose* address is being changed. The browser equivalent is keeping the intended page associated with the request. This example illustrates the mechanism, not a separately scored benchmark task.

### 2.2 Let the work determine when a task is finished

Long tasks often contain legitimate stretches of investigation: comparing pages, following links, waiting for a state change, or collecting several pieces of evidence. Stopping because an interaction has become long can leave a task incomplete even when its remaining steps are feasible. One of the recorded improvements addressed premature winding down and increased persistence on long-horizon work. (Cook, 2026)

Persistence is useful when it remains tied to the original objective. Repeating an ineffective action indefinitely would merely turn a failure into a slower failure. Figure 1 expresses the design intent: another action should follow from an updated view of the task, and completion should be supported by the observed result. The benchmark record supports the importance of persistence as an engineering target; it does not isolate how many percentage points that change contributed.

### 2.3 Preserve what the browser actually returned

Browser observations are not always plain sentences. They can include structured page information, screenshots, and the result of an interaction. If this material is flattened, lost, or associated with the wrong step, later reasoning operates on an incomplete account of what happened. The research included fixes that preserved rich browser results and images throughout the task trail. (Cook, 2026)

The practical distinction is between *being told that a page changed* and *retaining the evidence of the change*. Evidence makes it possible to revisit a decision, understand an unexpected transition, and evaluate an action after the fact. A successful action whose result disappears between system components can be as unhelpful to the final answer as an action that never succeeded.

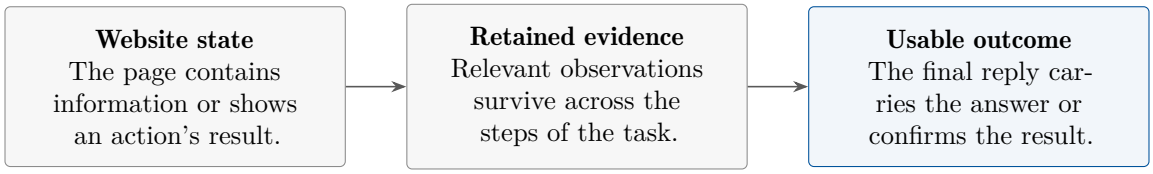
### 2.4 Treat browser resources as part of task reliability

Open pages consume resources and remain part of the agent's working environment. Unmanaged tabs can accumulate across tasks, increasing memory pressure and making subsequent behavior harder to interpret. The recorded work added explicit lifecycle management and cleanup for agent-created browser tabs. The relevant principle is ownership: the system should know which browser resources belong to its work and release those resources when appropriate. (Cook, 2026)

This is a systems contribution rather than a new reasoning capability. It does not teach the model a new fact. It helps preserve the conditions in which the model can use the capabilities

Cook.ai

it already has. A repeated evaluation is particularly sensitive to this distinction, because one task's leftover state can otherwise affect the next task.




A failure in any link can hide work that succeeded elsewhere. Better reasoning alone cannot repair missing evidence.

**Figure 2:** Why evidence preservation and final-answer handling matter. Completing an action, retaining its evidence, and delivering its result are distinct responsibilities.

### 2.5 Deliver the result, not just a trace of activity

A long task trail can contain intermediate findings, progress messages, and several candidate answers. The research corrected final-answer handling and preserved explicit answers during publishing workflows. These changes matter because the useful output is the answer or completed state requested by the person, not simply a record that tools were used. (Cook, 2026)

Together, the design choices address different failure points: target confusion, early termination, loss of observations, resource accumulation, and loss of the final result. They explain *how* a capable model can be made more effective inside a browser without assuming that the model alone is responsible for every change in the final score.

## 3 Evaluation Results

| Benchmark | Headline result | Stricter view |
|---|---|---|
| Online-Mind2Web, Aug. 2026 | 268 / 287 = 93.4% (13 impossible tasks excluded) | 268 / 300 = 89.3% (full cohort) |
| BU Bench V1, Sept. 2026 | 94 / 100 = 94.0% (best graded run per task) | 84 / 100 = 84.0% (first graded run) |
| Odysseys, Sept. 2026, rubric items | 1,171 / 1,225 = 95.6% (best graded run per task) | 867 / 1,225 = 70.8% (first graded run) |
| Odysseys, Sept. 2026, tasks with every item met | 182 / 200 = 91.0% (best graded run) | 105 / 200 = 52.5% (first graded run) |

**Table 1:** Cook Browser results. The September rows cover the full suites; a task's best graded run is the best of up to 29 graded runs (1.64 per BU Bench V1 task and 4.22 per Odysseys task on average). Task and rubric denominators are distinct.

### 3.1 Online-Mind2Web

The cohort contains 300 tasks. The accepted merge records 268 passes, 19 failures, and 13 impossible tasks. Within the adjusted cohort, performance was 76/77 on easy tasks (98.7%), 125/136 on medium tasks (91.9%), and 67/74 on hard tasks (90.5%). The benchmark evaluates task execution on live websites. (Cook, 2026; Xue et al., 2025)

### 3.2 BU Bench V1

In September 2026 Cook ran all 100 BU Bench V1 tasks, graded by the suite's official judge as published in Aside's repository, which reads the trajectory and the final answer. The first graded run of each task passed 84 tasks. Failed tasks were run again, several after the product changes described in Section 5.1; counting each task's best graded run, 94 of 100 passed. The

Cook.ai

remaining failures are bot checks or login walls on OpenReview, Vrbo, Apartments.com and Temu, a check page on Newegg, and one long research question. An August 2026 run of a frozen 20-task subset, the previous published figure, passed 17 of its 18 eligible tasks. (Cook, 2026; Browser Use, 2026)

### 3.3 Odysseys

Cook ran all 200 Odysseys tasks, graded by the official per-rubric judge with the trajectory's screenshots. Each task lists several rubric items; the task file used here lists 1,225 in total, while Aside reports 1,182 for its run. Counting each task's best graded run, Cook met 1,171 of 1,225 rubric items (95.6%) and every item of 182 tasks; on each task's first graded run it met 867 items (70.8%) and every item of 105 tasks. Unmet items concentrate on sites whose bot defenses refused the desktop browser, tasks that need dozens of pages open at the same time, and tasks that ask for schedules or prices that are not published yet. (Cook, 2026; Jang et al., 2026)

Arithmetic: a best-run score counts a task once any of its graded runs passed; rubric totals take each task's best graded run. A first-run score uses each task's earliest graded run. Until September 24, repeat attempts at a task could run in one of four shared benchmark workspaces, where files, carts and open tabs from an earlier attempt were still reachable. An LLM judge (Gemini 2.5 Pro, each verdict backed by a verbatim quote from the trajectory) read the first 40 steps of every repeat run from those workspaces that passed (BU Bench V1) or met every rubric item (Odysseys) and asked whether it used material an earlier run of the same task left behind; the one BU Bench V1 run and 28 Odysseys runs it flagged are not counted anywhere in this paper. Repeat runs that met only some rubric items were not judged. From September 24 each attempt ran in its own workspace, with the drive's chat, scratch and memory folders (and, from 15:40 UTC, its saved screenshots) emptied and a new browser profile. The Online-Mind2Web adjusted score divides passes by the eligible denominator; its full-cohort score keeps all 300 tasks. Percentages are rounded only for display.

## 4 Experimental Methodology

### 4.1 Execution path

Tasks were executed through the real Cook desktop browser. The September runs used Cook's desktop build for Linux on cloud machines, one task per machine at a time, browsing through United States ISP connections. One counted BU Bench V1 pass came from Cook's desktop build for macOS on a rented Mac, through the same kind of connection. Four counted Odysseys tasks with every item met came from Google Chrome on a rented Mac, driven by Cook's agent through Chrome's debugging port instead of Cook's own browser, through the same kind of connection. Evaluation considered the browser trajectory, available observations, final answer, and the relevant task rubric. Recording the path matters because a short final response may omit a page state that the browser evidence establishes, while a confident response alone may not establish that an action occurred. (Cook, 2026)

This report describes the system's design principles and recorded evaluation outcomes. Internal prompts, action-selection policies, recovery rules, resource budgets, and implementation details are outside its scope.

### 4.2 Models and accepted results

Each task's first graded run in September used GPT-6 Sol as the actor, and so did the run behind each of the 94 BU Bench V1 passes. Odysseys reruns also used Claude Opus 5.5, GPT-6 Astra and GPT-6 Luna (GPT-6 Luna is Cook's own default model: one run fell back to it because its desktop connection named another workspace than the one holding the subscriptions); of the 182 tasks with every item met, the earliest run that met them used GPT-6 Sol for 164, Claude Opus 5.5 for 14, GPT-6 Astra for 3 and GPT-6 Luna for 1. Grading used the official graders ported

Cook.ai

from Aside's public repository with their prompts unchanged: Gemini 2.5 Flash for BU Bench V1, and Gemini 3.1 Flash-Lite for each Odysseys rubric item, with screenshots. The actor and the graders therefore come from different model families; the grading is still automated, not an independent human evaluation. Online-Mind2Web combined a Claude Sonnet 5 baseline with targeted Opus 5 reruns in August; that result is a merged record, not a single uninterrupted run. These model and evaluation details describe the measurements, not a specification of the current product's serving configuration. (Cook, 2026; Aside, 2026)

### 4.3 Comparison with Aside

Aside publishes 99.0% on 300 Online-Mind2Web tasks, 93.0% on 100 BU Bench tasks, and 88.8% rubric-item success on 200 Odysseys tasks. Its Odysseys task-average rubric score is 86.5%; these two aggregations differ. Cook and Aside did not run in one controlled evaluation with identical actors, graders, cohorts, and website conditions. (Aside, 2026)

Cook's September runs cover the same full BU Bench V1 and Odysseys task sets and use the same published graders, but Cook's headline figures count each task's best graded run, and the actors, run policies and website conditions differ. The research page therefore identifies the external bars as published references. The Mind2Web reruns and exclusions and the different Odysseys rubric totals also prevent a direct all-suite ranking. All chart values and notes are downloadable at trycook.ai/research/results.json.

## 5 Discussion and Limitations

### 5.1 Engineering findings

The benchmark work led to long-horizon persistence improvements, model-pin controls, tab cleanup, correct final-answer extraction, requested-target binding, preservation of explicit answers during publishing, and preservation of rich REPL results and images. The accepted record connects these changes to browser execution; it does not isolate a numerical contribution for each fix. The architecture discussion provides a mechanism-based explanation, while the tables provide aggregate outcome evidence. A controlled ablation would be required to turn each proposed mechanism into an individual causal score claim. (Cook, 2026)

The September runs added three product changes. The agent now reads and acts inside cross-origin frames, such as embedded editors, that had finished loading before it first looked at the page; those frames present the same browser identity as the page around them; and when a request leaves an ordinary detail open, such as trip dates or a store, the agent picks one, says which, and completes the task instead of stopping to ask. The runs also exposed a fault in the evaluation harness itself, which delayed each site's first connection by up to 73 seconds; graded runs made before its correction are included in the results above.

### 5.2 Limitations and future work

Nineteen Online-Mind2Web failures remain from the August record; twelve are concentrated on apartments.com and petfinder.com. In the September runs, sites protected by one commercial bot-defense service refused Cook's desktop browser on the first request, while a stock Chrome browser on the same machine and connection loaded them; tasks that need 20 to 50 pages open at once exceeded the memory of the evaluation machines; and some tasks ask for schedules that are not published yet. The best-run figures are not single-run estimates; each task's first graded run is reported beside them. Anti-bot interruptions, changing inventory, and geographic restrictions remain material to live-web evaluation.

Next research step: repeat the full suites in one fixed configuration, report each run's score, and keep development reruns distinct from confirmation runs; publish full-cohort and adjusted scores together.

Cook.ai

## References


Cook. Browser evaluation records: Online-Mind2Web, August 5, 2026; BU Bench V1 and Odysseys full-suite runs, September 23–24, 2026. Engineering research notes. Public chart summary: https://trycook.ai/research/results.json.

Xue et al. Online-Mind2Web benchmark. https://github.com/OSU-NLP-Group/Online-Mind2Web.

Browser Use. BU Bench V1. https://github.com/browser-use/benchmark.

Jang et al. Odysseys: Benchmarking Web Agents on Realistic Long Horizon Tasks. https://arxiv.org/abs/2604.24964.

Aside. Published benchmark results. Accessed September 24, 2026. https://github.com/at-inc/aside-benchmarks.


**Data availability.** Accepted aggregates and scoring notes are reproduced here. Full Cook trajectories are retained internally; this paper is not a public per-task reproduction package.

Cook.ai