

AI Agent Browser User Guide

Aside AI Browser Practical Workflow Automation Use Cases

Concepts · Real-World Cases · Prompts

Attorney Kyung-jin Kim

Preface

The web browser is shifting from a tool manually clicked by humans into a tool where tasks are delegated through natural language. Aside is an AI agent browser that inherits your active browser's login state, allowing AI to view the screen, click, and enter text to carry out your workflows.

This book organizes real-world use cases of Aside across seven core domains: shopping and price comparison, blog and social media management, market research and data collection, internal administration, regular briefings and routines, and developer integrations—structured so you can follow along step by step. These cases are based on 65 user reviews, lecture videos, and official help center guides published between August 22 and October 22, 2026, with a complete reference list included at the end of the book.

Every section follows the same consistent structure. First, understand what task is being delegated through an Overview. Next, review how others accomplished it in Real-World Use Cases. Then, copy and adapt the Prompt Example to your own needs. Finally, consult the Expected Execution Flow and Precautions to manage costs and verify security safeguards.

Figures such as credit consumption, model names, and pricing plans reflect values at the time of publication and are subject to change. For irreversible actions like payments, public publishing, or sending emails and text messages, always include an explicit confirmation step in your prompts so that a human makes the final click.

Kyung-jin Kim

Table of Contents

Prologue. The Dawn of the AI Agent Era That Directly Operates Your Browser — p.8

Chapter 1. Aside Browser Fundamentals and Core Operating Principles — p.15

1.1 Web Session Inheritance and Secure Login: Importing Chrome Cookies/History and the Aside Vault Principle — p.15

1.2 The 6-Step Execution Process of AI Agents: Request, Planning, Execution, Authentication, Verification, and Result Reporting — p.18

1.3 Execution Permissions and Safeguards: Read-Only, Guard, Full Access Modes and Approval Procedures for Checkout and Publishing — p.22

Chapter 2. E-Commerce & Smart Shopping Use Cases — p.28

2.1 Multi-Mall Price and Review Comparison — p.28

2.2 Automated Cart Addition and Order Preparation — p.32

2.3 Building Financial Data Dashboards — p.35

Chapter 3. Blog & Multi-SNS Content Marketing Use Cases — p.41

3.1 Blog Writing Automation — p.41

3.2 Integrated Multi-SNS Channel Management — p.44

3.3 Content Performance Tracking Reports — p.48

Chapter 4. Market Research, Web Research & Data Collection Use Cases — p.54

4.1 Real Estate & Property Listing Monitoring — p.54

4.2 Community & News Issue Tracking — p.57

4.3 Corporate & Product Reference Analysis — p.61

Chapter 5. Internal Administration & Repetitive Office Automation Use Cases — p.67

5.1 Receipt & Expense Report Automation — p.67

5.2 Bulk Customer Notification & Message Dispatch — p.70

5.3 Internal ERP & Payroll System Processing — p.74

5.4 Supply Chain & Logistics Tracking — p.77

Chapter 6. Scheduled Briefings & Routines Automation Use Cases — p.83

6.1 Daily Morning Briefing — p.83

6.2 Registering and Managing Recurring Work Routines — p.86

Chapter 7. Developer & Advanced Automation Integration Use Cases (CLI / MCP / REPL) — p.93

7.1 Collaborating with Coding Agents (Integrating Claude Code, Codex, Cursor, etc. with Aside CLI/MCP) — p.93

7.2 Web Service UI/QA Testing (Post-Build Auto Login, Signup Screen Captures, and Anomaly Verification) — p.97

7.3 Deterministic Browser Automation (Script-Based Direct Web Inspection via REPL) — p.101

Epilogue. Risk Management and Guidelines for Effective AI Agent Use — p.107

References — p.114

Prologue. The Dawn of the AI Agent Era That Directly Operates Your Browser

① Overview

Most generative AI services we have experienced over the past few years have taken the form of a 'Chat UI.' Conventional chat-based AIs like ChatGPT, Claude, and Gemini remained in the role of an 'advisor' that answers text prompts with textual information, code, or drafting guides. No matter how brilliant the AI's analysis was, the user ultimately had to open the web browser themselves, enter usernames and passwords to log in, click buttons, and manually copy and paste data into Excel or internal enterprise systems.

An Agentic Browser fundamentally transforms this paradigm. Moving far beyond answering questions with text, it takes direct hold of the browser's mouse and keyboard just like a human, navigating websites, clicking buttons, filling out forms, downloading files, and moving across multiple sites to complete tasks.

Aside is a standalone desktop browser designed from the ground up for such AI agents. Rather than a simple extension that attaches a sidebar chatbot to Chrome, it is a dedicated agentic browser newly built on the Chromium engine. Developed by three Korean founders (Jun Kim CEO, Chanhee Lee CPO, Sanghoon Lee CTO) backed by Silicon Valley's premier startup accelerator Y Combinator (Fall 2025 F25 batch), it officially supports both macOS and Windows.

- Aside's most distinctive advantage lies in its design philosophy: "No separate API integration needed—the browser itself is the integration." Conventional AI agents functioned only when separate APIs or dedicated connectors were configured for tools like Notion, Gmail, Slack, and internal

ERPs; they were rendered useless on government portals, online banking, internal admin consoles, and e-commerce shopping malls where APIs were unavailable. In contrast, Aside seamlessly imports browsing history, bookmarks, and cookie sessions from your existing Chrome browser with a single click. Because the AI agent works directly inside a browser environment where the user is already logged in, immediate workflow automation becomes possible on virtually any web service without API integrations.

Furthermore, for security, it features a built-in encrypted password manager (Aside Vault), performing logins via browser autofill without ever exposing plaintext passwords to the AI model. Users can get started with 500 free credits provided every month, and can also link existing paid ChatGPT Plus/Pro or Claude Pro/Max accounts via OAuth login to run the browser agent freely without incurring additional costs.

② Real-World Use Cases

Coupage cordless vacuum cleaner price and review scraping with automated Google Sheets population: Navigated to Coupang to explore the top 3 cordless vacuums in the 200,000 KRW range with the highest review counts. The AI agent browsed product detail pages and reviews to extract product names, current prices, ratings, review counts, and key specifications. It then navigated to the user's Google Spreadsheet, populated the collected data into a structured table, and neatly formatted text wrapping and column alignments.

- Preparing Coupang daily necessity orders and pausing right before checkout: With an active Coupang login session, when prompted with "Order cleansing water" or "Order a 24-can pack of 355ml Coke Zero," the AI agent searched Coupang, precisely selected the correct volume options, added items to the account's cart, and proceeded all the way to the payment page.

It paused at the stage showing the shipping address and final payment amount to await human approval; once verified and manually clicked by the user, the package was delivered the next day.

- Batch recipient address entry for 10 book release event winners on Yes24: Delegated while keeping open a Google Forms response sheet containing addresses and contact details of 10 book launch event winners. The Aside browser opened 6 browser windows concurrently on Yes24 checkout pages, searched and populated each winner's address, and automatically entered recipient names and phone numbers, leaving the purchasing manager only having to enter the payment password.

Summarizing 5 Hacker News articles with benchmarked execution speed and credit usage: Requested to navigate to the tech community Hacker News homepage, select 5 trending AI posts, and compile a table with titles, one-line summaries, and original URLs. The Aside agent completed the table in 25 seconds, deducting 29 credits. When re-running the exact same prompt, it took 23 seconds and deducted 32 credits.

Editing personal information exposure segments in a study recording on YouTube Studio: In a 2-hour-21-minute study recording video, the agent identified 3 timestamps down to 0.25-second frame precision where attendees' phone numbers were exposed (including around the 1h 02m mark). The AI agent navigated to the logged-in YouTube Studio editor, cut the segments, and applied blur effects to complete the edits.

Bulk scraping of 104 community case studies by integrating Claude Code agent with Aside MCP: Connected Aside MCP to the Claude Code terminal to thoroughly collect 104 case study posts (approx. 440,000 characters) related to 'aside' from the GPters community. By extracting web session auth tokens through the Aside REPL and calling internal GraphQL APIs, it collected full-text content across all 104 articles out of 517 candidates without data

loss.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permissions: Guard Mode
- Model Configuration: Standard / High (Claude Opus or GPT series)
- Security Instruction: Ask First (User approval required before payment and final submission)

[User Prompt]

"Access Coupang and Naver Shopping respectively and search for '15.6-inch laptop privacy filter'.

[Collection Criteria]

1. From each shopping mall, collect information on the top 3 cost-effective products with a user review rating of 4.0 or higher (6 products total).
2. Target fields: Shopping mall name (Coupang/Naver), exact product name, listed price, shipping fee, rating, review count, product URL.
3. Calculate the 'Final Total Price' separately by adding the shipping fee to the listed price.

[Output and Storage Instructions]

- Organize the 6 collected products into a single comparison table, sorted in ascending order of 'Final Total Price'.
- Display the resulting table in the chat window and save it as 'laptop_privacy_filter_price_comparison.csv' in my computer's Downloads folder.
- Note: Never add any product to the cart or click checkout buttons under any circumstances."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon receiving instructions, the Aside agent opens background tabs and navigates sequentially to Coupang and Naver Shopping.

Enters the keyword '15.6-inch laptop privacy filter' into the search box and applies the 4.0+ rating filter.

Inspects product card DOM elements and detail page text to extract product names, prices, shipping fees, ratings, and review counts.

Accurately computes the total price including shipping in memory and sorts products in ascending order.

Generates and saves the CSV file to the designated Downloads folder, then outputs the comparison table alongside a completion summary in the chat.

- Precautions:

- Understanding credit consumption mechanics: Aside credits are consumed not based on the count of returned results, but in proportion to 'the number of times the AI navigates and opens web pages.' For instance, there is a documented benchmark where the AI searched for a keyword yielding 0 results and kept scrolling through dozens of pages without stopping, exhausting 150 credits (30% of the monthly free allowance). Always specify search bounds such as "search only within page 1" or "limit to top 5 results" in your prompts.

- Cost reduction via existing AI subscription linking: If you already subscribe to ChatGPT Plus/Pro or Claude Pro/Max, connect the account via OAuth in Settings > Models > Providers. This allows you to run the browser agent freely within your existing AI subscription limits without exhausting native Aside

credits.

- Safeguards for irreversible sensitive actions: Actions like payment, public publishing, or sending emails/SMS are irreversible sensitive tasks; maintain Ask First permissions and Final Confirm settings, adhering to the 'Human-in-the-Loop' principle where a human clicks the final confirmation button.

Chapter 1. Aside Browser Fundamentals and Core Operating Principles

1.1 Web Session Inheritance and Secure Login: Importing Chrome Cookies/History and the Aside Vault Principle

① Overview

The biggest hurdle when adopting a new workflow automation tool is having to enter your username and password, undergo two-factor authentication, and reconfigure your environment every time you access a website. Traditional web crawlers and legacy AI automation tools often cannot maintain logged-in web sessions or require building separate API connectors for each service, causing them to halt before login walls on internal ERPs, e-commerce admin panels, Gmail, Notion, and government portals.

The Aside AI browser's web session inheritance and secure login use cases were engineered to solve these inefficiencies. With a single click, users can import browsing history, bookmarks, and login cookies from their existing browsers such as Chrome, Edge, Safari, Arc, Dia, and Comet. Furthermore, through the built-in encrypted password manager Aside Vault, logins are safely handled via virtual browser autofill without ever exposing plaintext password strings to the AI model. This allows consumers and employees to immediately delegate tasks to AI in pre-authenticated browser environments without the hassle of re-entering credentials or engineering API integrations.

② Real-World Use Cases

- **Browser Data Import:** During initial setup and onboarding, users selected profiles from their existing Google Chrome, Microsoft Edge, Arc, Dia, and Comet browsers to seamlessly import browsing history, cookies, and bookmarks. As cookies were inherited, agent workflows immediately launched in an active authenticated state on frequently visited sites like Naver Blog, Coupang, Yes24, Notion, Gmail, Slack, and YouTube Studio without requiring re-authentication.
- **Safari Migration Specifics:** Due to macOS security policies, direct automated import is not supported for Safari; instead, browsing history and cookie data were securely brought into Aside via a ZIP export uploaded through Safari's File > Export menu.
- **Aside Vault Integration and External Password Manager Import:** By activating Aside Vault, users imported saved account credentials in bulk from Chrome, 1Password, Bitwarden, Apple Passwords, Dashlane, LastPass, or standard CSV files (name,url,username,password). The vault is unlocked via Touch ID fingerprint recognition on macOS, or Windows Hello biometrics and dedicated account master password gates on Windows.
- **Three-Tier Access Policy for AI Agents:** In Settings > Passwords > Access policy for AI agents, users configured credential usage policies across three modes tailored to their workflow: 'Always allow' (default), 'While unlocked' (active only when the user is at the machine with the vault unlocked), or 'Never' (completely blocking AI from autofilling passwords).
- **Plaintext Password Concealment and Incognito Restrictions:** When the AI agent reaches a login page, Aside Vault verifies the target URL against stored entries and populates browser DOM input fields via autofill. Throughout this process, raw password values are never passed into the AI agent's inference

context; the agent only observes that authentication succeeded. Additionally, for security, agent access to the Vault is entirely disabled in Incognito sessions.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permissions: Guard Mode (Allow access to designated folders)
- Password Policy: While unlocked (Autofill only when vault is unlocked)
- Security Instruction: Never output plaintext passwords in chat or files

[User Prompt]

"Navigate to my Naver Blog writing page.

If logged out, perform automatic login using the account credentials stored in Aside Vault.

[Task Instructions]

1. Once logged in, close the SmartEditor guide pop-up and enter 'Aside AI Browser Secure Login Hands-On' in the title field.
2. In the body, write the following sentence: 'Aside Vault logs in via autofill without exposing plaintext passwords to the AI.'
3. Never display or record my plaintext password or account credentials as text in the chat during the process.
4. When drafting is complete, do not click the 'Publish' button; click only the 'Save Draft' button and report the results."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon receiving instructions, the Aside agent navigates to the specified URL (Naver Blog editor).

When detecting a login page, Aside Vault verifies the domain URL and fills in the stored username and password via browser autofill.

After successful authentication, detects editor DOM elements and injects the title and body text.

Following instructions, clicks the top 'Save Draft' button and verifies that the draft count increments.

Outputs a chat summary confirming draft saving without password exposure and completes the task.

- Precautions:

- Domain URL accuracy verification: Aside Vault's autofill functions only when the stored entry's website URL strictly matches the actual login page URL. In corporate or academic environments using dedicated single sign-on portals, register the exact login form page URL in the Vault rather than just the generic domain root.

- Identity verification and Multi-Factor Authentication (MFA) intervention: While password entry is automated, the AI will pause when encountering SMS/push multi-factor authentication (MFA), passkey approval, or CAPTCHA challenges required by financial or enterprise portals. When an authentication prompt appears, complete it manually on your phone or fingerprint reader for the AI to proceed.

1.2 The 6-Step Execution Process of AI Agents: Request, Planning, Execution, Authentication, Verification, and Result Reporting

① Overview

When an autonomous AI agent executes tasks in a browser, understanding how a single user prompt is processed and completed internally is critical. Without visibility into this process, diagnosing why an AI paused becomes difficult, and inefficient web page navigation can lead to excessive credit consumption.

The Aside AI browser's 6-step execution process provides a clear pipeline architecture ranging from natural language prompt reception and sub-task decomposition to direct web UI control, secure authentication, human verification, and final deliverable generation. Beyond built-in models, Aside provides an open model ecosystem where users can link existing paid ChatGPT Plus/Pro, Claude Pro/Max, or GitHub Copilot accounts via OAuth, enabling virtually unlimited agent execution without consuming native Aside credits. Mastering this 6-step framework and cost-optimization setup empowers you to execute complex web workflows with precision at minimal cost.

② Real-World Use Cases

- Autonomous Agent 6-Step Execution Pipeline:
- Step 1 (Request Input): Enter instructions in natural language in the main address/command bar (Cmd+E or Cmd+T) or Side Panel. Attach active tabs or local files/folders using the @ key as needed.
- Step 2 (Task Planning): The connected AI model (GPT-5.6 Luna, Claude Opus 5.5, etc.) analyzes the request and breaks it down into sub-tasks.

- Step 3 (Browser Control & Execution): Dedicated Agent Tabs open to execute button clicks, scrolling, form inputs, and screenshot captures using Playwright-style JavaScript and visual intelligence (Visual UI).
- Step 4 (Secure Authentication): Upon encountering a login gate, autofills credentials via the Vault without exposing raw password text.
- Step 5 (Human Verification & Validation): Pauses immediately before irreversible actions such as payments, publishing posts, sending emails/SMS, or account cancellations to trigger a user approval prompt.
- Step 6 (Result Reporting): Compiles and presents findings in Markdown tables, CSVs, Google Sheets, or Word documents, saving them to the working directory.
- Three Model Integration Architectures for Cost Optimization:
 - Aside Default Tier Model: 500 free credits automatically replenished at the beginning of each month upon registration without payment (Free plan).
 - Existing Subscription Linking (Subscription): Connect already paid ChatGPT Plus/Pro, Claude Pro/Max, or GitHub Copilot accounts via Settings > Models > Providers > Connect. Runs for free within your existing subscription limits without deducting Aside credits.
 - External API Key Integration (BYOK API): Register API keys from Anthropic, OpenAI, Google, xAI, OpenRouter, DeepSeek, etc., for direct-billing large-scale inference.
 - Benchmarked Execution Speed and Credit Consumption Examples:

Collecting 5 top Hacker News stories and building a summary table took 25 seconds and deducted 29 credits (re-run took 23 seconds and deducted 32 credits).

Compiling a SaaS pricing comparison table took 22 seconds and deducted 16 credits.

Executing 35 multi-step tasks across enterprise portals consumed 340 credits in total over 5 days.

A YouTube search request yielding 0 results caused the agent to scroll through dozens of pages without stopping, consuming 150 credits (30% of the monthly free tier).

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permissions: Guard Mode
- Model Configuration: Task models -> Set default model to connected Claude or ChatGPT subscription model
- Search Bounds: Restrict to top 5 items within the first search results page

[User Prompt]

"Navigate to the Hacker News homepage and select the 5 most trending AI-related posts today.

[6-Step Pipeline Execution Instructions]

1. Extract the title, one-line summary, and original URL of each post.
2. Restrict the search scope to the top 5 items on the first page; avoid unnecessary scrolling or pagination.
3. If a bot verification (CAPTCHA) screen appears during navigation, do not force progression; halt immediately and await input.
4. Present the 5 collected articles as a single Markdown table in the chat, and save them as 'hackernews_top_stories.csv' in my Downloads folder."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Immediately upon prompt submission, the AI model decomposes the task into 4 sub-tasks: 'Enter Hacker News -> Parse DOM -> Summarize Text -> Save CSV'.

An agent tab opens in the background and scans headline elements on the Hacker News main page.

Extracts titles and links for 5 articles and summarizes the content of target pages.

Generates a Markdown table in the chat and creates the CSV file in the designated Downloads folder.

Completes the entire pipeline in approximately 25 seconds, displaying credit usage and a task completion message.

- Precautions:

- Must update Task models after connecting subscriptions: Even if you link your ChatGPT or Claude subscription under Settings > Models > Providers, you must manually change the default model under Task models below to the connected subscription model (e.g., Claude Opus 5.5 or GPT-5.6); otherwise, native Aside credits will continue to be deducted. Always verify Task models selection after linking.

- Explicit search bounds prevent credit waste: The agent will wander across dozens of pages until it finds items matching specified criteria. Specifying clear boundaries in your prompt, such as 'search page 1 only' or 'collect top 5 only,' prevents unnecessary credit consumption from runaway browsing.

1.3 Execution Permissions and Safeguards: Read-Only, Guard, Full Access Modes and Approval Procedures for Checkout and Publishing

① Overview

When an AI agent gains permission to click buttons on authenticated accounts and access your local file system to read and write files, robust safety safeguards become essential. Safeguards must prevent accidents such as an AI inadvertently deleting critical files, sending unauthorized emails to external contacts, or charging hundreds of dollars on an e-commerce platform.

The Aside AI browser's permissions and safeguards use cases provide triple security guardrails to isolate and control the agent's operating scope. It operates three session-level file permission modes (Read-Only, Guard, Full Access), granular tool-level rules (Allow, Ask, Deny), and a Human-in-the-Loop framework that never executes irreversible actions—such as payments, post publishing, message dispatch, or bulk data deletion—without manual user confirmation. This empowers both individual users and enterprises to maximize automation efficiency while rigorously protecting internal security policies and private data.

② Real-World Use Cases

- Three Session Permission Modes:
- Read-Only: The AI agent can only read browser screens and specified file contexts; all write operations that create, modify, or delete files on the local machine are strictly blocked.
- Guard (Default): The AI agent freely reads and writes files only within the designated project Working Folder; any attempt to access system folders or

directories outside this boundary triggers a user confirmation pop-up.

- Full Access: Permits unrestricted read and write access across all system paths for overnight unattended routines or bulk batch operations. Even in this mode, plaintext passwords stored in the Vault are never revealed to the AI.

- Tool-Level Action Rules (Allow, Ask, Deny): Configured per capability in Settings > Agents. 'Allow' executes without prompting, 'Ask' requires pop-up approval prior to execution, and 'Deny' strictly forbids the action. If rules conflict, 'Deny' takes ultimate precedence.

- Four Sensitive Operations Never Executed Automatically (Human-in-the-Loop):

Payments & Checkout

Publishing / Posts

Sending Messages (Email and SMS)

Bulk Data Deletion & Account Closure (Delete / Submit)

- Real-World Safety Guardrail Implementation Examples:
 - Pausing before Coupang checkout: Added daily necessities (Coke Zero, etc.) to the Coupang cart and navigated to the payment page; the agent halted immediately before clicking the final payment button to await user approval.
 - Drafting blog posts and saving drafts: Explicitly instructed 'Never publish publicly' in prompts for Naver Blog; verified that the agent clicked only the top 'Save Draft' button before wrapping up.

- Test dispatch and confirmation before bulk SMS delivery: When sending notification SMS to 31 recipients via Google Messages Web, dispatched a single test message to the operator's phone number to verify wording and schedule before receiving approval for the full run.
- Aborting within 25 seconds upon detecting Google search CAPTCHA: When encountered with a bot verification challenge (CAPTCHA) during Google search, halted operations within 25 seconds and notified the user rather than attempting bypasses that could trigger account bans.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permissions: Guard Mode (Allow access to Downloads folder)
- Safety Policy: Ask First and disable Skip confirmation
- Negative Constraints: "Never execute payments, publishing, email dispatch, or file deletion"

[User Prompt]

"Access Coupang and search for '355ml Coke Zero 24 cans'.

[Task and Safety Instructions]

1. Select the Rocket Delivery product with the highest review rating and add it to the cart.
2. Navigate to the order/checkout page and verify that the default shipping address is properly selected.
3. Proceed up to the payment method selection screen and immediately pause execution.
4. Under no circumstances click the 'Pay Now' button or enter payment passwords; freeze the screen and request my approval."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon prompt submission, Aside navigates to Coupang and searches for the product within Guard permission limits.

Selects the Coke Zero option, adds it to the cart, and enters the order form page.

Verifies the shipping address and reaches the screen displaying the final checkout amount.

The sensitive operation guardrail activates; the agent displays a 'Waiting for Payment Approval' pop-up and pauses execution.

Prompts the user to visually inspect the address and total amount, completing the flow by letting the user click the payment button directly.

- Precautions:

- Check the Final Confirm switch: When executing critical tasks like payments or email dispatch, ensure that the 'Skip confirmation' toggle in Aside settings or at the bottom of the chat interface is turned off (set to Ask First).

- Precautions for unattended background routines: When running morning routines while away from the machine, assign Guard or Full Access permissions, but structure prompts purely around safe queries, scraping, and file exports, completely excluding payment or deletion commands.

Chapter 2. E-Commerce & Smart Shopping Use Cases

2.1 Multi-Mall Price and Review Comparison

① Overview

In the online shopping process, consumers who want to find the lowest price or high-cost-performance products must visit multiple e-commerce platforms individually, such as Coupang, Naver Shopping, Amazon, and Danawa. Manually accessing each platform, entering search keywords, logging into member accounts, and cross-referencing prices, discount policies, shipping conditions, product specifications, user ratings, and review counts onto a separate note or Excel file is an extremely cumbersome and time-consuming task.

Aside AI Browser directly accesses Coupang, Naver, Danawa, Amazon, and more by inheriting the cookies and login sessions users previously maintained in their existing browsers. As if a human were visually browsing web pages, the AI agent navigates the sites to accurately identify and extract prices, actual total payment amounts including shipping fees, specifications, review ratings, and review counts. The collected information is automatically organized and output as a single Markdown table, CSV file, or Google Spreadsheet, allowing consumers and marketers to compare product details across multiple shopping malls at a glance within minutes and make optimal purchasing decisions.

② Real-World Use Cases

- **Danawa Audio Device Spec and Price Comparison:** A real case of directly accessing the Danawa website, applying a price filter under 100,000 KRW, and searching for earphone products. The AI agent extracted product names, selling prices, ratings, and review counts from the search results and summarized them into a clean table. Even when subsequently requested to add release dates or detailed option fields, it adjusted the table structure and updated it within seconds.
- **Coupage Cordless Vacuum Cleaner Research and Google Sheets Auto-Creation:** The agent searched Coupage for the top 3 cordless vacuum cleaners in the 200,000 KRW price range with the highest number of user reviews, collecting product names, current prices, ratings, review counts, and key features. Upon completing collection, it accessed the user's Google Spreadsheet, entered the gathered information into a tabular format, and automatically adjusted basic formatting such as line wraps and alignment.
- **Naver Shopping vs. Coupage Robot Vacuum Cleaner Price Comparison:** The agent simultaneously searched Naver Shopping and Coupage for a specific robot vacuum cleaner model released after 2026. Beyond simple unit price comparison, it cross-referenced final total payment amounts including shipping fees alongside actual product purchase links, completing a comparison table showing which shopping mall offered a lower actual purchase price.
- **Laptop Privacy Filter Comparison Across Coupage and Naver Shopping:** The agent searched for 15.6-inch laptop privacy filters based on cost-performance and customer reviews, comparing prices, shipping fees, review counts, and ratings across Coupage and Naver Shopping. It found a product priced in the 8,900 KRW range with 691 reviews and a 4.5 rating on Coupage, and a 12,500 KRW free-shipping product with 1 review and a 5.0

rating on Naver Shopping, clearly illustrating the differences in pricing and review distribution between the two platforms.

- Amazon Wireless Earbud Collection: The agent visited the Amazon website, searched for 3 earbud products priced under \$100, and organized their model names, prices, noise-canceling support status, wired/wireless type, and product links into a table.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Model Setting: Standard / High (Claude or GPT series)
- Save Location: My Computer's Downloads folder

[User Prompt]

"Go to Coupang and Naver Shopping respectively and search for '15.6-inch laptop privacy filter'.

Collect information on the top 3 products with high cost-performance and a review rating of 4.0 or higher from each shopping mall, totaling 6 products.

Items to collect:

1. Shopping mall name (Coupang or Naver)
2. Exact product name
3. Listed price and shipping fee (calculate total amount including shipping fee separately)
4. Rating and number of reviews
5. Direct product URL

Once collection is complete, organize and compare them in a single table sorted from lowest to highest final total price including shipping.

Display the results as a table in the chat window, and save it as

'laptop_privacy_filter_price_comparison.csv' in the Downloads folder.

However, do not add products to the cart or click the checkout/payment button under any circumstances."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

After user instructions are submitted, the Aside agent opens Coupang and Naver Shopping tabs sequentially or in parallel.

It enters the specified keywords into search boxes and applies filters for ratings of 4.0 or higher and price/popularity sorting.

It navigates search result listing pages and product detail pages, identifying product names, prices, shipping fees, ratings, and review counts.

It consolidates the parsed data, calculates total amounts including shipping fees, and sorts them.

It generates a Markdown table, saves a CSV file to the designated Downloads folder, and outputs a completion report.

- Precautions:

- Credit Consumption Optimization: Instructing the agent with broad scopes—such as "investigate every single review," "explore the entire product description," or "infinitely scroll to the end of the page"—leads to browsing dozens of pages, rapidly driving up free/paid credit consumption. When writing prompts, you must explicitly delimit the search scope, such as "top 3 to 5 products only" or "search first page only."

- Specify Payment Prevention Constraints: To prevent accidental misclicks during search and comparison tasks, it is safe to include negative constraints at the bottom of the prompt, such as "do not add to cart" and "do not proceed with payment."

2.2 Automated Cart Addition and Order Preparation

① Overview

When moving beyond simple information gathering to the actual purchase stage in shopping malls, tedious repetitive tasks arise, such as selecting options (color, specifications, quantity, etc.), entering shipping addresses, and filling in recipient contact information. In particular, for corporate event giveaways, bulk purchases of team daily supplies, or sending gifts to event winners where individual orders must be placed for multiple identical or distinct delivery destinations, the manager must open address lookup windows and fill out forms dozens of times.

Aside AI Browser utilizes an encrypted password vault (Vault) and existing cookie sessions to automatically log into relevant commerce portals, handling specification option selection, address lookups, recipient information input, and coupon applications that humans previously had to handle one by one. This use case prepares order sheets up to the exact step prior to final payment authorization (the screen awaiting payment method selection and password entry) and pauses, providing tremendous convenience where the user only needs to verify the address and amount prepared by the AI and press the final payment button.

② Real-World Use Cases

- Coupang Cart Addition and Checkout Page Entry: While logged into Coupang, the agent searched for specific daily necessities such as Coca-Cola Zero (355ml, 24 cans) or cleansing water, selected the exact volume and options, added them to the user's shopping cart, and navigated all the way to the payment checkout page.
- YES24 Batch Order Creation for 10 Event Winners: The agent used winner names, addresses, and contact numbers gathered via Google Forms to send

books as gifts to 10 publication event winners. Aside Browser opened 6 browser windows simultaneously on the YES24 purchase page, looked up and entered each winner's address, auto-filled phone numbers and recipient names, and prepared them in a payment-pending state. The purchasing manager simply opened each window, verified the address accuracy, and entered the payment password to complete all orders.

- Amazon Login Session-Based Cart Addition: Using passkeys and Vault auto-fill, the agent logged into an Amazon account, matched the quantities and options for protein shakes or daily supplies requested by the user, added them to the cart, and reported completion.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Security Policy: Ask First (Await approval prior to payment step)
- Reference Materials: Currently active tab (@Response Sheet or Winner List)

[User Prompt]

"Go to the Coupang site. If not logged in, log in using Aside Vault credentials. Search for 'Coca-Cola Zero 355ml 24 cans' and select the direct-purchase Rocket Delivery product. Set the quantity to 1, add it to the cart, and proceed directly to the order/checkout page.

[Notes]

- Verify that the default delivery address is correctly selected.
- Navigate up to the payment method selection screen and pause the task.
- Never enter payment passwords or click the 'Pay Now' button; pause the screen and wait for approval."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon receiving instructions, Aside accesses Coupang and proceeds with Vault-based login.

After searching for the product, it selects the exact product card matching the specified options (355ml, 24 cans, Rocket Delivery).

On the product detail page, it verifies the quantity and clicks 'Add to Cart' or 'Buy Now'.

It transitions to the order sheet page and auto-completes the default shipping address and recipient details.

At the step immediately preceding payment where the final checkout amount is displayed, the agent pauses and presents a popup notification to the user stating "Payment preparation completed."

- Precautions:

- Safety Guardrails and Approval Policies: Aside enforces Human-in-the-Loop guardrails for financial and sensitive actions such as payments, publishing posts, and transmitting messages. Even if payment prohibition conditions are specified in the prompt, the user must personally verify that the AI has not misentered the address and check the final payment amount before completing the payment via manual input.

- Handling Address Lookup Popups: When street name address search popups or identity verification popups appear, the AI might misidentify address form elements; thus, you should verify in advance whether the address text in the source (Excel/Google Forms) is accurate.

2.3 Building Financial Data Dashboards

① Overview

Time deposit and installment savings interest rates across commercial banks and savings banks fluctuate frequently depending on market conditions. However, conventional interest rate comparison websites and monochrome tables often show only simple nominal rates, making it difficult for consumers to obtain a practical answer to: "If I deposit 500,000 KRW monthly for one year, what is the actual maturity payout I will receive after taxes?"

Aside AI Browser directly accesses the Financial Supervisory Service's (FSS) Integrated Financial Product Comparison Disclosure API and official disclosure web pages of individual financial institutions to collect interest rate data, synchronizing the gathered data with web dashboard environments such as Supabase and Netlify. Furthermore, it automatically verifies simple/compound interest calculation logic, monthly contribution limit exception handling, and ordinary taxation (15.4%) calculation rules, continuously keeping the dashboard UI up to date. By configuring scheduled morning Routines, the latest interest rate and maturity payout dashboard is maintained automatically without human intervention.

② Real-World Use Cases

- Building a Dashboard Integrated with FSS Disclosure API Data: Utilizing the Financial Supervisory Service's Integrated Financial Product Comparison Disclosure API, the agent collected 767 deposit and installment savings products and 4,327 interest rate/maturity period options from commercial banks and savings banks, storing them in a Supabase database. Based on this data, a real-time deposit and savings dashboard web service was deployed to Netlify.

- Decoupled Architecture Separating UI Modifications from Data Ingestion: To prevent unnecessary traffic from re-invoking the FSS API every time the user edits web dashboard UI designs or copy, the disclosure ingestion/sync server function was completely decoupled from user screen query functions. A structure was created that compares product hash values and only updates new or modified products in the database.

- Daily Scheduled Auto-Collection Using Aside Routines: A task titled 'Periodic Deposit/Savings Disclosure Collection · Daily at 9:00 AM' was registered in Aside Routines. At 9:00 AM every morning, Aside automatically sends an authenticated collection request to the server function to refresh the latest interest rate snapshot.

- Contribution Cap Exception Handling and Maturity Payout Calculation Logic: Default calculation amounts were set to 1,000,000 KRW monthly for installment savings and 10,000,000 KRW for time deposits; however, for products with contribution caps (e.g., installment savings capped at 500,000 KRW monthly), the default input amount was automatically reduced to the cap of 500,000 KRW. A UI rule was implemented to display a warning and revert to the previously permitted amount if a user enters 600,000 KRW, exceeding the cap.

Validating Financial Double-Application Errors in Compound Interest Calculations: In the process of validating savings bank disclosure rate tables, 'contracted interest rate' was distinguished from 'annual average yield' that already factored in compounding. Catching an error where interest was duplicated if the disclosed annual average yield was fed into the monthly compound formula again, the dashboard calculation logic was updated to accurately differentiate the applied contracted rate (e.g., 4.00%) from the disclosed compound yield (e.g., 4.07%).

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Connected Services: Netlify deployed web page and Supabase dashboard
- Routine Registration: Run daily at 9:00 AM

[User Prompt]

"Update the data on the currently deployed deposit and savings dashboard website.

Fetch new disclosure data for time deposits and installment savings from commercial banks and savings banks via the Financial Supervisory Service disclosure API, and synchronize it with the Supabase DB.

Calculation and UI Application Rules:

1. Set default maturity calculation amounts to 1,000,000 KRW per month for installment savings and 10,000,000 KRW for time deposits (12 months, ordinary taxation at 15.4%).
2. If a product contribution limit is lower than 1,000,000 KRW per month, automatically set that product's maximum limit as the default value.
3. For compound interest products from savings banks, verify the logic to calculate monthly compound interest based on the 'contracted interest rate' rather than the 'disclosed annual average yield'.
4. Rather than displaying the estimated maturity payout extensively on the main card, modify it to appear as a popup window when each bank card is clicked.

Once synchronization is complete, verify whether the saved snapshot timestamp has updated properly and report back."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

- At the scheduled routine time (9:00 AM) or upon user instructions, Aside invokes the financial disclosure ingestion API and server function.

After ingesting FSS disclosure data, it cross-references existing DB product hash values to classify modified, new, or discontinued products.

It updates and stores only changed products and interest rate options into the Supabase DB.

It executes validity checks on simple/compound interest calculation logic and contribution limit exception handling (running approximately 100 regression test scenarios).

It verifies the final update completion timestamp, validates the dashboard deployment status, and outputs a report.

- Precautions:

- Avoid Direct Access to 2FA-Protected Financial Accounts: Having an AI directly log into personal online banking dashboards to calculate payout amounts easily triggers session timeout errors due to the financial sector's characteristic 15-minute two-factor authentication (Push/OTP) challenges. Therefore, rather than logging directly into personal accounts, integrating official disclosure APIs or public database ingestion methods is far more stable.

- Calculation Disclaimer Notice: Because minor discrepancies with actual bank maturity payouts may occur depending on deposit dates, individual bank rounding/truncation methods, and variable interest rate adjustments, a disclaimer notice must be clearly displayed at the bottom of the dashboard stating: "This calculation result is an estimated amount for comparison based on fixed monthly contributions and fixed interest rates, and may differ from the actual payout."

Chapter 3. Blog & Multi-SNS Content Marketing Use Cases

3.1 Blog Writing Automation

① Overview

Blog marketing is an exceptionally powerful tool for brand promotion and personal branding, yet regularly producing high-quality posts daily or weekly demands considerable labor and time. Completing a single post involves complex manual steps: topic research and drafting the body text, maintaining the existing blog's writing style and tone, collecting and editing images, positioning images in harmony with text context, organizing subheadings and readability, saving drafts, and final publishing.

Aside AI Browser directly accesses the editor interfaces of major blogging platforms such as Naver Blog and Tistory where the user is already logged in. The AI agent inspects the author's previous posts to autonomously learn frequently used phrasing, vocabulary, sentence lengths, and structural writing patterns. It pulls photographic data stored in Google Drive or local folders, inserts images at locations that visually match the narrative context, arranges subheadings, and completes the article. By completely taking over the process of drafting and editing on the browser, the author only needs to review the draft and imagery prepared by the AI and click final publish, or run the entire pipeline through fully automated routines.

② Real-World Use Cases

Learning Naver Blog Writing Style and Auto-Posting with Google Drive Photo Integration: The agent accessed the author's Naver Blog, analyzing and parameterizing the style, context, vocabulary, writing formulas, and patterns from 6 to 7 recent posts. Concurrently, it accessed an Osaka trip photo folder in Google Drive, downloaded 17 photos, filtered out duplicates or blurry shots, and drafted a travel journal aligned with scenes such as the first night's cityscape, Osaka Castle, and Nara Deer Park. In the Naver Blog editor, it wrote the article mirroring the author's distinctive sober tone and manner, organized subheadings, rearranged photo placements, and completed a private draft publishing. Furthermore, after completing the task, it saved this entire procedure as a skill file so that it could be executed with a single command next time.

- Drafting and Saving Household Budget and Reading Log Posts: The agent was instructed to write a September household budget settlement post and a September reading log post in the Naver Blog editor. After drafting the body content, per user request, it did not publish publicly right away; instead, it clicked only the 'Save Draft' button, verified that the draft count incremented, and finalized the session. The author refined the title and select sentences of the AI-generated draft into their own personal voice before final publication.

- Replacing Blog Skin and Profile Logo: Moving beyond basic post creation, the agent entered the Naver Blog admin settings page while logged in, uploaded and updated a new blog profile logo image, and automatically replaced the top blog skin design as well.

- Cross-Platform Post Migration and Bulk Link/Image Corrections: The agent migrated 110 articles from the Brunch platform—which does not offer an open API—to Naver Blog, and traversed all 306 posts published across 3

distinct blogging services to audit and repair expired links and broken images all at once.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode (access to Google Drive and local workspace folder)
- Model Setting: Standard / High (Claude Opus or GPT series)
- Reference Materials: My Naver Blog session and Google Drive @Osaka_Photos folder

[User Prompt]

"Navigate to my Naver Blog post writing page. If the login session is not open, log in using Vault credentials.

[Step 1: Learn Writing Style and Patterns]

Read the 5 most recent posts on my blog and analyze my frequently used phrasing, sentence lengths, subheading structures, and vocabulary characteristics.

[Step 2: Gather Materials and Draft Body Text]

Check the photos uploaded today in the 'Osaka_Photos' folder on Google Drive. Recognize the locations and situations in the photos, and draft a '3 Nights 4 Days Osaka Travel Journal' blog post adhering strictly to my analyzed writing style. Distribute photos throughout the article at positions matching the context, and divide sections with readable subheadings.

[Step 3: Save and Guardrails]

Once writing and image insertions are complete, never click the 'Publish' button; make sure to click only the 'Save Draft' button and report the result."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Aside assigns parallel agents that simultaneously manipulate the open Naver Blog tab and Google Drive tab.

It collects existing post texts to extract writing style data formatted as JSON and skills.

It downloads image files from the Google Drive folder and analyzes their visual scenes.

It enters Naver Blog SmartEditor to inject the title, subheadings, body text matching the analyzed tone, and downloaded images sequentially.

It clicks the 'Save Draft' button at the top of the editor, verifies the updated draft count, and prints a summary in the chat window.

- Precautions:

- Handling Naver Search Algorithm Policies: If 100% AI-generated posts are automatically published publicly as spam, there is a risk of incurring exposure penalties (low-quality classification) by search engine algorithms. Rather than full auto-publishing, it is safer to specify 'Private Publishing' or 'Save Draft' in the prompt and establish a pipeline where a human conducts final review (Human Touching) to adjust the title and select expressions before going public.

- Image Rendering Latency: Uploading a large volume of high-resolution images into the editor at once can cause browser DOM rendering delays, potentially leading the AI to misclick buttons; therefore, images should be compressed or limited to under 10 photos, or uploaded in stages via prompt instructions.

3.2 Integrated Multi-SNS Channel Management

① Overview

Marketers and solo creators who simultaneously operate multiple social media channels—such as Instagram, Threads, YouTube Studio, Naver Blog, and Notion—suffer severe workflow fragmentation every day. Individually logging into each platform, drafting content copy, posting Instatoon captions, pasting Reels/Shorts voiceover scripts, and updating schedules on Notion content calendars is routine labor that incurs massive time and mental fatigue.

Even for social media platforms that do not support API integrations or present difficult access barriers, Aside AI Browser seamlessly leverages sessions where the user is already logged in across browser tabs. With just a single natural language instruction, it traverses tabs across Instagram, Threads, YouTube, and Notion simultaneously, handling Instatoon caption uploads, script drafting, profile information updates, and Notion calendar organization all at once. Through this, marketers can establish a multi-channel operational pipeline publishing 3 or more times per week without intricate software development or third-party automation tool setups.

② Real-World Use Cases

- Establishing an Integrated Operational Pipeline Across 5 SNS Channels: Aside was deployed to an operational pipeline running 3 times weekly across 5 platforms: Instagram, Threads, YouTube, Naver Blog, and Notion. It drafted Instatoon captions and Reels voiceover scripts, published Threads posts, batch-organized Notion content calendar schedules, and executed multi-channel management including simultaneously updating profile logos and header skin images across Naver Blog and Threads.

- Threads Trend Analysis and Thumbnail Planning Integration: The agent accessed the Threads website and extracted 3 trending topics gaining traction among users through scrolling exploration. Based on the gathered trend topics, it generated YouTube thumbnail proposals and creative briefs, automating thumbnail production via integration with the AutoClick service.

Bulk Informational SMS Dispatch to Offline Lecture Applicants from Google Forms: The agent opened a Google Sheet listing 30 applicants who signed up for an offline seminar via Google Forms, screened for duplicate and invalid phone numbers, and calculated actual recipients excluding cancellations. Opening a Google Messages web tab paired with a personal mobile phone number, it sequentially auto-sent informational text messages to 31 recipients without manual copy-pasting.

- Automated Event Scheduling Bridging Gmail and Google Calendar: After extracting date, time, location, and attendee information from meeting invitation emails in the Gmail inbox, the agent immediately navigated to the Google Calendar tab, opened a new event creation modal, and scheduled the meeting automatically.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Login Sessions: Pre-logged into Instagram, Threads, YouTube Studio, and Notion tabs
- Reference Materials: Notion content calendar page (@Notion_Calendar)

[User Prompt]

"From the open Notion calendar tab, read the content topic and draft marked as 'Scheduled for release this Wednesday'.

[Step 1: Draft Copy for Instagram and Threads]

- Based on the topic, draft captions for an Instatoon (including 5 hashtags) and script text for Reels voiceover.
- Draft a short-form text post for Threads (within 300 characters) and prepare it in the Threads input box.

[Step 2: Update Notion Calendar Status]

- Once drafts are complete, update the status of the corresponding entry on the Notion calendar to 'Completed' and attach the draft links.

However, do not click the final 'Post' button on Instagram and Threads; pause and wait for my approval."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

It accesses the Notion tab to read the day's scheduled content topic and key focus terms, storing them in memory.

It opens Instagram and Threads tabs to draft captions and text tailored to the formatting standards of each platform.

The agent returns to the Notion calendar view and updates the progress status property to 'Completed'.

It pauses execution immediately before posting and prompts the user to review the caption content.

- Precautions:

- Platform Account Penalties (Bot Detection) Caution: On platforms like Instagram, Threads, and X (Twitter), if repetitive clicks or bulk posting occur in a uniform pattern over a short period of time, the account may be flagged as spam and suspended. When formulating prompt instructions, you must specify guidelines such as "introduce a random wait time of 3 to 5 seconds between actions" and "limit the number of posts per day."

- Mandatory Approval for SMS and Email Dispatches: To prevent damages resulting from accidental erroneous transmissions when sending bulk SMS or emails, a human-in-the-loop approval (Ask) step must be conducted prior to sending to visually verify recipient lists and body copy.

3.3 Content Performance Tracking Reports

① Overview

The core of marketing activities does not end with uploading content; it lies in tracking metrics on how widely published content has been exposed across each channel, identifying age brackets and keywords where drop-offs occur, and incorporating that feedback. However, YouTube Studio audience retention graphs, Naver Cafe and Blog external mention volumes, and Instagram view counts either lack open APIs or are provided solely as charts, previously requiring managers to manually inspect screens, take screenshots, and transcribe figures into Excel.

Aside AI Browser directly accesses administrator dashboards—such as YouTube Studio, Naver Cafe, Threads, Naver Search Advisor, and Google

Search Console—while maintaining authenticated sessions. The AI agent visually and textually interprets on-screen graphs, view counts, impressions, click-through rates (CTR), drop-off points, and comment counts, extracting them into CSV files or unified summary reports. In particular, when registered as a scheduled Routine every Monday morning, users can receive last week's channel report card directly via Slack or Google Sheets without manually logging into multiple admin consoles each time.

② Real-World Use Cases

- **Automated Weekly Performance Tracking Routine Across 5 SNS Channels:** Running an Aside routine every Monday morning automatically collected the previous week's performance across Instagram, Threads, YouTube, Naver Blog, and Notion. Gathering view counts, likes, and comment reaction metrics across each channel, it tracked ongoing performance by compiling channel-specific metrics—including 1,135 Threads followers, 726 YouTube subscribers, and 1,905 Naver Blog neighbors—into a consolidated operations spreadsheet.
- **YouTube Studio Drop-Off Segment and Viewer Retention Analysis:** Logging into the YouTube Studio management console, the agent analyzed viewer retention graphs and drop-off rate data across 7 long-form videos. Analyzing retention graphs unavailable through public APIs, it identified a pattern where half of viewers dropped off within the first 60 seconds, leading to concrete content improvement proposals such as shortening intro lengths.
- **Brand External Mention Tracking and Keyword Collection:** To collect external mention volumes for the GPTERS brand, the agent traversed Naver Blog, Naver Cafe, YouTube, and Threads in an unauthenticated state. It discovered 3 actual mention posts on Naver Blog (consuming 55 credits) and 11 mentions over 1 month on Naver Cafe (consuming 145 credits), compiling a table detailing exact publication timestamps (e.g., 2026.08.25 00:08),

original URLs, and one-line summaries.

Detecting and Editing Personally Identifiable Information Exposure in YouTube Studio Recordings: In a 2-hour 21-minute study session recording, the agent detected 3 timestamps where students' phone numbers were exposed (e.g., 1:02:36–1:04:16) with 0.25-second frame precision. In the YouTube Studio video editor screen, it cut the sections and applied blur effects, significantly reducing editing turnaround time.

- **Naver Search Advisor Search Analytics Capture and Cross-Referencing:** Logging into Naver Search Advisor—inaccessible to Claude or Codex—the agent retrieved summary screens, 20 search keywords, and click/impression metrics, saving 39 screenshots. Without arbitrarily altering ambiguous numbers from the tables (e.g., 2.3k impressions), it preserved source data verbatim to complete a weekly search analysis report.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Login Sessions: Pre-logged into YouTube Studio and Naver Search Advisor
- Routine Setting: Recurring execution every Monday at 9:00 AM

[User Prompt]

"Access my YouTube Studio and Naver Search Advisor dashboards to analyze content performance over the past 7 days.

[Collection and Analysis Items]

1. YouTube Studio: View counts, click-through rate (CTR), average view duration, and initial 60-second viewer drop-off rate for the 3 most recently uploaded videos.
2. Naver Search Advisor: Total impressions, total clicks, top 10 ranked search queries, and click-through rates over the past 7 days.

[Report Generation]

- Based on the collected metrics, produce a comprehensive performance summary table.
- If any numbers are ambiguous or not clearly displayed on screen, do not fabricate them; mark them as 'Unverified'.
- Present the generated report in the chat window, and save it as 'weekly_content_performance_report.csv' in my workspace folder."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:
 - At the scheduled routine time (Monday 9:00 AM), Aside runs in the background to access YouTube Studio and Naver Search Advisor tabs.

It navigates dashboard views and analytics chart DOM elements to extract textual metrics and graph data.

It standardizes collected metrics and categorizes key drivers such as top-ranked keywords and drop-off rates.

It generates and saves a CSV file in the designated workspace folder, then outputs a completion report.

- Precautions:

- Gauging Credit Consumption from Page Traversal: During basic search or data collection, if an agent performs infinite scrolling or traverses dozens of pages to locate results, credits deplete rapidly. For example, in an observed case, checking 0 search results on YouTube consumed 150 credits; therefore, when writing prompts, you must set an upper bound on page navigation, such as "collect from page 1 only" or "check data within the past 7 days only."

- Role Division Between API Utilization and Browser Agents: For standard data harvesting where official APIs are available—such as Naver Blog/Cafe search or public open data—API integration is considerably faster and less costly. Deploying the Aside browser agent specifically to areas where 'no API exists and human authenticated visual observation is strictly necessary'—such as YouTube Studio chart analysis and extracting private metrics behind authenticated dashboards—yields the highest cost efficiency.

Chapter 4. Market Research, Web Research & Data Collection Use Cases

4.1 Real Estate & Property Listing Monitoring

① Overview

When monitoring real estate investment or residential properties, the greatest challenge lies in manually verifying constantly changing listing information, real-time prices, and gap investment amounts (purchasing property with an existing lease deposit/jeonse) on a daily basis. Major real estate portal sites such as Naver Real Estate do not display property listings all at once; instead, they use infinite scrolling, and brokerage advertising modal popups frequently obscure the screen upon access, making data collection difficult using standard web scraping scripts or APIs alone.

The Aside AI browser utilizes computer vision and browser DOM recognition technology to access real estate websites just as a human would view the screen and move the mouse. It autonomously detects and closes brokerage advertising modal popups appearing on screen, scrolls down infinitely, and collects listing information from designated areas (e.g., Gaepo Jugong, Mok-dong, Yeongdeungpo, etc.) without missing a single item. Furthermore, beyond simply scraping listing catalogs, it cross-references purchase prices with lease (jeonse) prices to calculate estimated gap investment amounts for properties subject to lease succession conditions, outputting the result into a CSV file in real time. This allows real estate brokers and investors to receive an organized dataset of the lowest-priced listings matching their desired criteria in just a few minutes every morning.

② Real-World Use Cases

Naver Real Estate Gaepo Jugong listing collection and gap investment estimation: In this use case, the Naver Real Estate listing page URL was provided to the Aside browser with the prompt, "Review Gaepo Jugong listings to identify properties purchasable with the lowest amount of capital, and save the complete listing catalog as a CSV." The Aside agent opened the browser window, performed infinite scrolling, and exhaustively collected listing data. Among the gathered listings, it identified properties subject to lease succession conditions, calculated and provided the estimated gap amount required for purchase, and cleanly saved the complete catalog into a CSV file.

Collection of 20 apartment listings in Mok-dong and Yeongdeungpo with automated modal popup handling: In a browser environment connected to Claude, the agent was instructed to access Naver Real Estate, collect 20 apartment listings across Mok-dong and Yeongdeungpo, and log them into a CSV file in real time. While sequentially exploring listings—including 9 in Mok-dong and 5 in Yeongdeungpo—brokerage advertising modal boxes popped up across the website; the AI autonomously recognized and properly closed them, proceeding seamlessly with the collection task.

③ Prompt Example

PROMPT

[Request Conditions]

- Execution Permission: Guard mode
- Target Site: Naver Real Estate (@active_tab or URL)
- Save Path: Downloads folder on my computer

[User Prompt]

"Access the Naver Real Estate website and explore apartment listings across Mok-dong and Yeongdeungpo.

[Collection and Analysis Conditions]

1. Scroll down the listings infinitely in these areas and collect information on the top 20 listings.
2. If brokerage ad popups or modal windows obstruct the screen midway, close them immediately and continue working.
3. Collected items: Apartment complex name, building/floor, exclusive area, sale price, lease price (if available), property feature description.
4. For listings with both sale price and lease price, calculate (Sale Price - Lease Price) and add an 'Estimated Gap Amount' column.

Sort the collected 20 listings in ascending order of 'Estimated Gap Amount', save them to the Downloads folder as 'real_estate_listings_monitoring.csv', and display the top 5 key listings in a table within the chat window."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon instruction submission, the Aside agent immediately creates a Naver Real Estate tab and navigates to the specified regional listings page.

During page loading, it visually detects the location of the close button (X) on any event/promotional modal popup and clicks it.

It sequentially scrolls downward, collecting text elements from property cards (complex name, area, price, description).

It converts sale price and lease price strings into numeric data, then calculates the difference (gap amount).

It sorts the collected data in ascending order by gap amount, exports it to a CSV file, and presents a completion report.

- Precautions:
 - Setting infinite scroll limits: Given the nature of real estate portals, specifying nationwide or excessively broad areas can load thousands of listings, overloading browser memory and consuming excessive credits. You must explicitly specify an upper limit in the prompt, such as 'top 20 listings' or 'up to 3 pages maximum.'
 - Numeric unit conversion and validation: In real estate pricing, mixed notation such as '15.5 billion KRW' or '820 million KRW' is common; therefore, it is prudent to manually cross-check the final CSV file once to ensure the AI-converted gap amount figures correspond accurately to the original sale prices.

4.2 Community & News Issue Tracking

① Overview

In fast-evolving industries such as IT, AI, and marketing, staying informed in real time about viral topics across diverse communities and news portals—including Hacker News, X (Twitter), Reddit, Naver Cafe, and IT news sections—is critical. However, manually visiting dozens of sites every day, scrolling through articles, and checking comments consumes an immense amount of time. Furthermore, for communities lacking search APIs or portals with restricted search conditions, humans had no alternative but to type queries and navigate manually.

The Aside AI browser navigates directly to various communities and news portals specified by the user, regardless of login status. The AI agent scans recently published or popular posts, extracts key takeaways, engagement metrics, and exact publication timestamps (accurate down to the minute), and organizes them into a unified Markdown table alongside original source links. Moreover, users can configure Routines to automatically trigger news aggregation and blog/social media topic recommendations at a set time every morning. It also features smart detection capabilities: should bot verification like Google CAPTCHA appear, it safely pauses execution immediately and alerts the user rather than fruitlessly burning credits.

② Real-World Use Cases

- Hacker News 5 AI articles summary and pricing comparison test: The agent was instructed to access Hacker News, select 5 AI-related posts, and create a table with titles, one-line summaries, and original links. In just 25 seconds, the Aside agent accurately extracted titles, summaries, and links for all 5 articles and formatted them into a table, consuming a total of 29 credits. When rerun on the same task, it completed in 23 seconds and consumed 32 credits.
- Establishment of daily morning AI news briefing routines: A recurring routine was scheduled for 8:00 AM or 9:00 AM every morning to extract 5 major AI articles from the Naver News IT/Science section and overseas tech communities. The AI was configured not only to organize article titles, press outlets, one-line summaries, and source links, but also to recommend 3 topic ideas and candidate headlines suitable for writing on blogs or social media that day. Upon execution, it autonomously added a cautionary note explaining that publication times for overseas articles might differ by a day due to time zone differences.

- Curation of 5 trending AI news items on X (Twitter): The agent scanned trending posts from hundreds of accounts flooding the X (Twitter) platform daily, selected 5 must-watch news items (such as ElevenLabs MCP expansion, Gemini 3.1 Pro vs GPT-6 Astra SVG comparison, and AGI-related computer use videos), and compiled screenshots and summary text to create newsletter content.
- Brand external mention volume tracking and CAPTCHA detection pause: To discover external mentions of a brand ('GPters'), the agent browsed Naver Blog, Naver Cafe, Google, YouTube, and Threads in a logged-out state. It accurately identified 3 posts over 7 days on Naver Blog (consuming 55 credits, capturing publication timestamp down to the minute: '2026.08.25.00:08') and 11 posts over 1 month on Naver Cafe (consuming 145 credits), matching the ground-truth benchmark with 100% precision. Conversely, when date filter interactions during Google search triggered bot detection and presented a CAPTCHA screen, Aside did not waste credits or fabricate results; instead, it halted execution immediately within 25 seconds and reported the situation to the user.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Read only mode
- Routine Configuration: Can be scheduled for recurring execution at 8:00 AM daily
- Safety Rules: Immediate halt upon CAPTCHA detection; hallucinations (fabrications) strictly prohibited

[User Prompt]

"Access the Naver News IT/Science section and the Hacker News main page respectively.

[Task Instructions]

1. Select a total of 5 trending AI-related articles and posts from both sites today.
2. Collected items: Article title, press outlet/source, key content one-line summary, original URL.
3. Based on the 5 collected articles, draft 2 content topic ideas and suggested headlines suitable for writing on blogs or social media today.
4. If a CAPTCHA or bot verification prompt appears during search, do not force progression under any circumstances; halt immediately within 25 seconds and notify me.
5. Do not fabricate unverified article content; output only the collected results in a clean table."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:
- According to the routine schedule (8:00 AM) or user prompt, Aside opens Naver News and Hacker News tabs asynchronously to access them.

It reads article list DOM elements and headline text to filter for AI-related keywords.

It navigates into each original article page to identify core sentences, capturing publication timestamps down to the minute and URLs as

snapshots.

It monitors for security challenges such as CAPTCHAs, and if no issues arise, returns a comprehensive summary table and blog topic recommendations.

- Precautions:

- Excessive credit consumption during '0 results' searches: AI agent credit costs are incurred not by the number of collected results, but by the 'process of navigating browser pages to find results.' In practice, an observed test case consumed 150 credits (30% of a monthly free tier limit) while deeply navigating YouTube just to confirm 0 mentions. To prevent infinite exploration on platforms with no results, explicitly state in the prompt: 'If no mentions appear on the first search page, report 0 results immediately.'

- PC power state during Routine execution: Aside Routines operate while the computer and the Aside application remain powered on. While they function normally under Windows Lock screen (Win+L) or simple display-off states, Routines will not run if the PC enters full Sleep Mode or if the application is closed; therefore, automatic PC sleep settings must be disabled.

4.3 Corporate & Product Reference Analysis

① Overview

When launching a new product or formulating a marketing strategy, conducting corporate reference research to cross-analyze pricing policies, discount rates, shipping terms, feature sets, customer review ratings, and pros/cons across top 10 market competitors is an indispensable process. However, because each competitor site has a completely different page structure, requires logins or closing popups, and presents product spec tables in inconsistent formats, manually compiling a comparison matrix wastes dozens of hours per week.

The Aside AI browser is specialized for multi-site research; with a single natural language command such as 'Compare prices, shipping policies, and review ratings for the top 10 brands in this category and compile them into a table,' it sequentially opens and navigates dozens of competitor websites. When necessary, it performs logins or selects option menus to accurately extract required values, visualizing extracted specs and pricing in a unified cross-comparison matrix table. Furthermore, based on collected and analyzed data, it generates executive-ready Word (.docx) report files with a single click and saves them directly to your desktop.

② Real-World Use Cases

Cross-comparison of pricing and shipping policies across top 10 competitors: In this use case, the agent was instructed to access 10 competitor websites sequentially, crawl subscription plans, prices, free shipping thresholds, and return/refund policies, and perform comparative analysis. The Aside agent opened all 10 sites directly, extracted required figures, and compiled them into a single comparison table, saving the researcher at least 5 to 6 hours of work per week.

- Cross-check of lowest-cost model pricing between OpenAI and Anthropic: The user requested a table comparing lowest-cost model prices with the official pricing pages of OpenAI and Anthropic open side-by-side. In approximately 22 seconds, it output a comparative pricing table for GPT-6 Luna and Claude Haiku 4.5 models complete with accurate source links, consuming 16 credits for the task.

Danawa and Amazon cross-spec comparison of earphones under 100,000 KRW / \$100: The agent accessed Danawa and Amazon, applied filters for under 100,000 KRW (under \$100 on Amazon), and explored 3 wireless earphone products. It gathered model names, prices, wired/wireless type, active noise cancellation support, user ratings, review counts, and product links into an organized table. When subsequently instructed to 'add release dates or additional spec criteria,' it intelligently updated and restructured the existing table in just 3 seconds.

Competitor product analysis report generation as Word (.docx) file and desktop saving: Based on collected and analyzed competitor product data, the agent was instructed to author an executive-level analysis report. The Agent structured an executive summary, review status analysis, satisfaction vs dissatisfaction ratios, interpretation, and marketing implications into a Word document (.docx), created a folder on the desktop to save it, and automatically opened the folder window to display the completed file.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode (Desktop report file save permission)
- Model Setting: Standard / High (Claude Opus or GPT series)
- Exploration Targets: 3 designated competitor service pricing pages (@Site_A, @Site_B, @Site_C)

[User Prompt]

"Access the pricing pages of the 3 designated competitor websites directly and cross-compare their latest service rates and spec terms.

[Research and Collection Items]

1. Service/Plan name and monthly/annual subscription prices (including discount rates)
2. Availability of free tier and major restrictions
3. Core feature support (API access, team collaboration features, capacity limits, etc.)
4. Shipping policy or refund terms
5. Direct URLs to original pricing pages

[Output and Document Saving Instructions]

- Construct a 3-way pricing cross-comparison matrix table based on the collected data.
- Beyond a simple table, provide 3 single-line strategic takeaways highlighting pricing advantages our service can leverage.
- Compile the entire analysis into a formatted Word document ('competitor_pricing_analysis_report.docx'), save it to the desktop, and open the destination folder window on screen."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon instruction submission, the Aside agent opens tabs for the 3 designated competitor websites sequentially.

It navigates to the Pricing and Features pages of each site, parsing table formats and pricing text.

It standardizes collected rates and feature options into a common structure across all 3 competitors to generate matrix comparison data.

The AI agent drafts comprehensive takeaway paragraphs written in a professional market analysis tone.

Using Python or the built-in document generator, it creates and saves the .docx report file on the desktop, popping up the folder window to finalize the task.

- Precautions:

- Validating differing pricing tiers: Occasionally, the AI might conflate pricing with different billing cycles (e.g., monthly billing rate vs. monthly equivalent under annual billing) and misalign them in tables. You must specify in the prompt: 'Clearly distinguish between annual discounted pricing and single-month billing rates.'

- Excessive out-of-scope product spec searching: If the AI continuously crawls external blogs to forcefully fill in missing specs that do not match given criteria, credit consumption escalates. Adding a negative constraint at the bottom of the prompt—such as 'Mark unverified spec items as "Unverified" rather than guessing forcefully'—is a key technique to guarantee output reliability.

Chapter 5. Internal Administration & Repetitive Office Automation Use Cases

5.1 Receipt & Expense Report Automation

① Overview

Among internal administrative duties for corporations and sole proprietors, the most frequent, repetitive, and cumbersome task is organizing receipts, proof documents, and drafting expense reports. To process monthly travel costs, team dinners, office supply purchases, and consumable payments, administrators must open dozens of PDF receipts, image files, and digital receipts individually. Manually inspecting transaction dates, merchant names, payment amounts, and payment methods (credit card, cash receipt, etc.) to transcribe them by hand into Excel or internal expense filing systems represents a quintessential waste of time.

The Aside AI browser provides powerful file exploration and OCR processing capabilities that directly handle receipt files gathered in local computer folders. The AI agent sequentially reads bundles of receipt PDF files in the Downloads folder or designated working directories, accurately extracting essential items. Gathered items are standardized into highly readable Markdown tables, CSV files, or Excel spreadsheets. Furthermore, it can interface with national tax portals (such as Hometax) or internal accounting portals to verify cash receipt issuance and attach supporting documents in email replies. This completely eliminates manual transcription labor from expense filing while maximizing precision.

② Real-World Use Cases

Receipt PDF extraction from Downloads folder and automated table generation: In this case, the Aside agent directly read receipt PDF files accumulated in the computer's Downloads folder. By recognizing text and layouts within the files, it accurately categorized transaction dates, store names (merchant trade names), payment amounts, and payment methods, summarizing them into a clean table.

National Tax Service Hometax cash receipt issuance and email reply integration: When an administrative clerk received an email requesting 'Please issue a cash receipt,' they completed the workflow using the Aside browser. With the email open, the user prompted the agent, prompting the AI to navigate directly into the logged-in National Tax Service Hometax portal, issue the cash receipt, and download the issuance proof PDF. Subsequently, it returned to the email composer window, attached the downloaded proof file, and drafted a confirmation reply email—executing an integrated administrative pipeline end-to-end.

- Vendor invoice downloading and consolidated expense reporting: The agent gathered invoices and receipts for the current month dispersed across multiple web services and vendor portals into a single folder. By navigating to each portal, downloading billing statements, and generating an internal expenditure report table, it executed the entire pipeline without requiring manual copy-pasting by staff.

③ Prompt Example

PROMPT

[Request Conditions]

- Execution Permission: Guard mode (Allow Downloads folder access)
- Target Files: All receipt PDF files in my computer's Downloads folder

[User Prompt]

"Read all receipt PDF files in my computer's Downloads folder and create an expense settlement table.

[Extraction Items]

1. Transaction Date (YYYY-MM-DD format)
2. Merchant Name (Store name or business trade name)
3. Payment Amount (Numeric value and KRW currency)
4. Payment Method (Credit card, cash receipt, bank transfer, etc.)
5. Receipt File Name

[Output and Saving Instructions]

- Sort collected items chronologically by transaction date and present them as a Markdown table in the chat window.
- Calculate the sum of all receipt amounts on the bottom line labeled 'Total Expenditure Amount'.
- Save the organized results into the Downloads folder as '2026_October_Expense_Report.csv'.
- Do not modify or delete the original receipt PDF files under any circumstances."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon instruction submission, Aside scans the specified Downloads folder within Guard permission scope.

It ingests PDF receipt files in the folder as individual snapshots and text data.

It parses patterns for dates, merchant names, amounts, and payment methods into structured data, calculating the grand total.

It generates a summary table in the chat window and saves a CSV file to the designated path.

It reports file creation completion alongside confirmation of original file integrity, concluding the task.

- Precautions:

- Image receipt resolution and OCR error verification: For crumpled or dim scanned image receipts rather than digital PDFs, OCR might misread the number of zeros in amounts or misinterpret merchant names. Always verify whether the total expenditure in the final CSV matches the withdrawal total on original bank or card statements.

- Verifying Guard permissions for original file preservation: If receipt processing involves renaming or moving files, ensure the permission mode remains set to Guard and explicitly stipulate 'Do not delete original files' to prevent loss of original source records.

5.2 Bulk Customer Notification & Message Dispatch

① Overview

When hosting offline seminars, corporate events, book launch events, or product briefings, administrative staff face severe repetitive toil when dispatching individual notifications detailing event dates, directions, required materials, or virtual links to dozens of registrants. Copying dozens of names and phone numbers one by one from Excel or Google Forms response sheets to paste into web SMS windows or Google Messages often leads to misdirection incidents—such as omitting recipients or typing wrong names.

The Aside AI browser simultaneously manipulates the Google Sheets tab containing registrant lists and the Google Messages Web tab (synchronized between mobile and PC) within a unified workspace. The AI agent reads the spreadsheet roster, automatically screens out duplicate numbers, typographical errors, and cancellations, and tallies only verified recipients. It then sequentially injects recipient phone numbers and message copy into the dispatch interface, executing bulk dispatches in mere minutes. Crucially, before dispatching, it self-checks for textual inaccuracies, consent records, and data leak prevention constraints, awaiting final human confirmation to completely eliminate bulk misdelivery risks.

② Real-World Use Cases

Validation of 31 offline seminar registrants from Google Forms and bulk SMS dispatch: In preparing a free offline seminar, registrant records submitted via Google Forms were reviewed in Google Sheets. The user commanded the Aside agent to ingest names and phone numbers for over 30 registrants from the sheet.

- Automated screening of duplicates and cancellations: The AI agent thoroughly inspected the sheet roster, filtered out duplicate submissions and malformed numbers, excluded canceled registrants, and reported the verified recipient count (31 recipients).
- Google Messages Web synchronized dispatch: Opening the Google Messages Web tab linked to the user's personal mobile phone, it flawlessly automated individual notification texts to 31 recipients without requiring manual copy-pasting.
- Pre-dispatch schedule mismatch detection and legal consent validation: When instructed to send, the agent autonomously identified a contradiction between the original Google Form seminar schedule (1:30 PM–4:00 PM) and

the scheduled text message draft (1:30 PM–3:30 PM), flagging the discrepancy for user confirmation. Furthermore, it conducted intelligent verification to confirm whether legal message reception consent had been secured in accordance with communications regulations.

- Compliance with security and leak-prevention constraints: Following prompt guidelines, it strictly observed safety guardrails: leaving the original Google Sheets data untouched, refusing to expose recipient phone directories within the AI chat dialogue, and avoiding transmissions to external sites.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Reference Tabs: Open Google Sheet (@registrant_list) and Google Messages Web (@Google_Messages)
- Safety Policy: Ask First (Confirm receipt of test message before full dispatch)

[User Prompt]

"Review the offline seminar registrant list in the open Google Sheet and send out notification text messages via Google Messages Web.

[Step 1: Roster Cleansing and Verification]

1. Verify the names and mobile phone numbers of applicants.
2. Filter out duplicate phone numbers, malformed numbers, and cancellations, then report the verified headcount for actual dispatch.
3. Verify whether the seminar date and time recorded on the sheet match the schedule stated in the notification message text.

[Step 2: Testing and Dispatch Guidelines]

1. Never modify the original Google Sheets data.
2. For personal data protection, do not output the full phone number directory in the chat window.
3. Send 1 test text message to my phone number first and await my approval.
4. Upon approval, dispatch bulk messages sequentially in batches of 5."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Aside accesses the Google Sheets tab to scan recipient phone number and name columns.

It screens for duplicate entries and malformed patterns, tallies the refined recipient headcount, and cross-checks for schedule discrepancies within the message copy.

It navigates to the Google Messages Web tab, dispatches 1 test notification text to the test phone number, and pauses execution.

Once the user checks the test text message on their mobile phone and clicks 'Approve Dispatch', it proceeds with bulk sending in 5-message batches.

It reports the final dispatched count and concludes the workflow.

- Precautions:
 - Preventing platform spam blocking during rapid transmissions: If over 50 text messages are sent out continuously within a few seconds via carrier-linked portals or Google Messages Web, mobile telecom carriers may flag them as spam and block message dispatch. You must include instructions in your prompt such as 'Wait 3 seconds after every 5 messages' and 'Proceed in staged batches.'
 - Specifying privacy negative constraints: To prevent the agent from exposing customer names and full phone directories in the chat output or external memory, always include the constraint 'Do not print full phone number rosters in the chat window' in the prompt.

5.3 Internal ERP & Payroll System Processing

① Overview

Internal ERPs, HR and payroll systems (e.g., Rippling), national tax services (Hometax), and government portals often provide no public APIs for security and privacy reasons, remaining hidden deep behind authenticated login sessions. As a result, administrative personnel have had to log in with their own accounts, navigate menus, and download documents manually every month for payroll stub retrieval, VAT and cash receipt audits, and purchase order (PO) delivery schedule checks.

The Aside AI browser inherits the user's regular browser cookie sessions and encrypted password vault, effortlessly navigating past security login walls on internal ERP and payroll portals that lack API integrations. In an authenticated state, it retrieves current-month payroll statements or verifies parity between internal ERP dashboard data and external documentation in seconds. Plaintext passwords are injected via cryptographic accelerators without exposure to AI models, and human confirmation is required during sensitive steps like payments or filings—enabling administrative automation that strictly complies with enterprise security policies.

② Real-World Use Cases

- Rippling payroll service login and automated payroll statement download: The agent accessed Rippling, a prominent US payroll management platform, automatically downloaded this month's payroll statements, and organized them into designated folders.
- National Tax Service Hometax VAT filing and document verification: The agent navigated the Hometax portal (which does not support API integrations), audited the VAT filing entry status, allowed the user to visually inspect the completed forms, and had the user press only the final submission button, demonstrating a safe human-in-the-loop workflow.

Reconciliation of delivery dates between cloud ERP and vendor email purchase orders (Open PO): The agent accessed the procurement dashboard of an internal cloud ERP system, cross-checking delivery schedules for 10 recent purchase orders against promised delivery dates received in vendor emails. Through this process, it detected 4 critical discrepancies where parts shipments had been postponed by 6 days without being updated in the internal ERP, preventing supply chain disruptions proactively.

- US visa application drafting and Hometax cash receipt issuance proxy: The agent filled out complex input fields on US visa application forms and completed cash receipt issuance and file downloads on Hometax. In both cases, the AI populated all form fields while leaving only the final 'Confirm/Submit' button for human execution.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Access Targets: Internal cloud ERP dashboard and Gmail inbox (@Gmail_Tab)
- Security Guidelines: Utilize Vault-stored credentials (strict prohibition against printing plaintext passwords)

[User Prompt]

"Cross-check my internal ERP procurement management page and Gmail inbox.

[Audit Instructions]

1. Read the 'Expected Delivery Date' for the 10 most recent Purchase Orders open on the ERP dashboard.
2. Locate the 'Promised Delivery Date' in shipment confirmation emails sent by the respective vendors this week in my Gmail inbox, and cross-reference them.
3. If there are any discrepancies between the ERP expected delivery date and the email promised delivery date, compile the PO number, vendor name, and difference in days.

[Reporting and Saving]

- Display the list of discrepancies in a table within the chat window, and save it to my work folder as 'ERP_PO_Delivery_Discrepancy_Audit_Report.csv'.
- Do not click modify or delivery date change buttons in the ERP; perform read-only auditing only."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon instruction submission, Aside accesses the ERP portal and Gmail tab, completing automated login via Vault.

It parses PO numbers and expected delivery date text for the 10 purchase orders on the ERP procurement dashboard.

It switches to the Gmail tab to extract promised delivery dates from vendor email bodies matching the respective PO numbers.

It cross-references the two date datasets to flag delayed or mismatched schedules.

It outputs a summary report as a table, exports a CSV file to the designated folder, and concludes.

- Precautions:

- User intervention upon multi-factor authentication (MFA): Internal ERP or payroll portals frequently require phone push notifications or OTP secondary verification upon login. If the AI pauses here, do not worry; the user simply needs to complete the MFA step on screen for the workflow to proceed to subsequent automation steps.

- Confirm settings for data modifications and submission buttons: Because data in an ERP cannot be easily rolled back once modified or submitted, explicitly stipulate 'Do not click data edit/submit buttons' in the prompt and keep the Final Confirm toggle enabled in Aside settings.

5.4 Supply Chain & Logistics Tracking

① Overview

For import/export, logistics, and supply chain management (SCM) personnel, every morning begins with intense manual lookup toil. To track estimated times of arrival (ETA), current positions, and customs hold statuses for dozens of incoming cargo containers, staff must visit multiple ocean freight carrier portals such as Maersk, MSC, and CMA CGM individually. Because each carrier portal features distinct page layouts, search query forms, and strict session timeouts, coordinators have spent over 45 minutes every day copying container numbers and pasting them into Excel tracking sheets.

The Aside AI browser inherits carrier portal login sessions and cookies, injecting dozens of container IDs into search fields in batch and automatically extracting tracking data. The AI agent awaits asynchronous table rendering across each carrier site, captures vessel names, current ports, discharge ETAs, and customs hold statuses, and compiles them into a single Markdown table or consolidated Excel tracking file. Configured as a routine every morning, logistical milestones are fully refreshed before coordinators arrive at the office, saving over 30 minutes of manual labor daily.

② Real-World Use Cases

Simultaneous tracking across 3 ocean carrier portals and automated extraction of 18 container milestones: A field production test deployed the Aside browser across 3 carrier portals—Maersk, MSC, and CMA CGM—to replace manual inquiry workflows that took 45 minutes every morning.

- Batch tracking of 18 import container IDs: The coordinator provided a list of 18 container IDs and prompted: 'From the active carrier tracking tabs, extract the vessel name, current port, discharge ETA, and customs hold

status for each container ID into a Markdown table.'

- Awaiting asynchronous rendering and completion in 2 minutes 40 seconds: The Aside agent sequentially manipulated carrier search forms and awaited data loading, extracting milestone data for all 18 containers into a flawless Markdown table in just 2 minutes and 40 seconds.

100% success rate across 85 container queries over one week, saving 35 minutes daily: Over 5 consecutive days of production testing across 14 carrier tracking runs and 85 container lookups, it achieved a 100% success rate without session drops, saving 35 minutes of working time every morning.

Application of asynchronous load wait rules and CAPTCHA interrupt guardrails: Strict stability rules were applied: waiting 3 seconds for asynchronous table rendering after clicking search, recording 'NOT_FOUND' and immediately advancing to the next number when search results yielded nothing, and instantly halting operations to alert a human rather than forcing execution if bot verification (CAPTCHA) occurred.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Target Tabs: Open carrier portal tabs (@Maersk, @MSC, @CMA_CGM)
- Target Container IDs: [ID-1001, ID-1002, ID-1003, ID-1004, ID-1005]

[User Prompt]

"Sequentially search tracking information for the 5 provided container IDs across the open carrier portal tabs.

[Extraction Items and Execution Rules]

1. Extraction items: Container ID, Vessel Name, Current Port, Discharge ETA, Customs Hold Status.
2. After clicking the search button, wait 3 seconds for asynchronous table loading to complete.
3. If no search results are returned, enter 'NOT_FOUND' and immediately proceed to the next container ID.
4. If a CAPTCHA or identity verification window appears during search, halt operations immediately and notify me in the chat window.

Arrange the collected results into a Markdown table sorted by earliest Discharge ETA, and save it to my work folder as 'container_arrival_milestones.csv'."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon receiving instructions, Aside navigates to the designated carrier portal tab and locates the search input field.

It sequentially enters container IDs one by one, clicks the search button, and waits the designated 3 seconds for asynchronous loading.

It accurately extracts vessel names, current locations, ETAs, and customs status text from the milestone table rendered on screen.

It sorts the collected data in ascending order by ETA date, formatting it into a Markdown table and CSV file.

It reports the total number of inquiries and success rate, concluding the workflow.

- Precautions:

- Specifying asynchronous rendering (AJAX) wait time: On carrier portals, clicking search typically triggers partial data table loading rather than a full page refresh. Explicitly specifying 'Wait 3 seconds after clicking search' in the prompt prevents the agent from reading prematurely and returning errors, serving as a core technique for achieving a 100% success rate.

- 'Record Not Found' exception handling: When entering canceled or erroneous container numbers, portals may return error screens. You should include exception handling instructions directing the agent to enter 'NOT_FOUND' and immediately advance to the next number, preventing infinite retry loops on identical IDs.

Chapter 6. Scheduled Briefings & Routines

Automation Use Cases

6.1 Daily Morning Briefing

① Overview

Reviewing major issues across dozens of news sites, IT communities, financial disclosures, and social media right after arriving at work every morning is essential for knowledge workers, marketers, and executives, but it consumes 20 to 30 minutes or more of valuable morning time each day. Manually visiting each site, entering search terms, reading articles, and taking summary notes is not only cumbersome, but depending on one's condition, it is also easy to miss specific news or lose energy and focus to unnecessary gossip articles.

The scheduled briefing use case of the Aside AI browser completely automates this repetitive morning research task. Instead of the user opening the browser and searching, the AI agent autonomously opens the browser in the background at a designated time (e.g., 8:00 AM or 9:00 AM every day) and navigates to specified press IT sections, Hacker News, X (formerly Twitter), the Korea Exchange (KRX) disclosure site, and more. The AI selects 3 to 5 core topical news items of the day, organizing the original article links, publication timestamps, concise one-line summaries, and recommended writing topics applicable to the day's tasks or content creation into a neat presentation delivered directly via the chat window or as a file.

② Real-World Use Cases

Automated summary routine for 5 Naver News IT/Science articles every morning at 8:00 AM: A case of registering a scheduled Cron routine in Aside's Routines menu that runs every morning at 8:00 AM. At 8:00 AM every day, the AI agent autonomously navigates to the Naver News IT/Science section, extracts 5 major articles, and provides them to the user organized in a clean table with article titles, one-line summaries, and direct links to the original articles. The next morning at 8:00 AM, the results were successfully compiled without any human intervention.

Comprehensive Hacker News and IT community briefing with blog topic recommendations: Configured as a daily 9:00 AM routine to browse both the Hacker News main page and domestic Korean IT news simultaneously. It collected core summaries and links for 5 major articles in just 25 seconds (consuming 29 credits). Beyond simply delivering the news, it also generated and provided '3 great blog writing topics and recommended titles for today' based on the gathered articles. In addition, when collecting overseas articles, the AI independently recognized that publication dates might differ by one day due to time zone differences with local time, adding an explanatory cautionary note at the bottom of the briefing.

- KRX (Korea Exchange) and market condition briefing integration: Successfully operated a scheduled routine that accesses Korea Exchange (KRX) disclosures and market indicator websites every weekday at 8:00 AM to automatically conduct major market reviews and financial briefings.
- Terminal-based morning news collection using Aside CLI: Successfully executed the task of selecting 5 articles from the Naver News IT/Science section and organizing them into a table using a single terminal command line (aside "...") in just 59 seconds.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Read only mode
- Routine Schedule: Every day at 8:00 AM (Register as Cron Routine)
- Constraints: No button clicks or page modifications permitted; pause immediately if a CAPTCHA appears

[User Prompt]

"Set up a scheduled briefing routine that runs every morning at 8:00 AM.

[Collection and Briefing Guidelines]

1. Navigate to both the Naver News IT/Science section and the Hacker News main page.
2. Select the 5 most important AI and technology-related news stories published today.
3. Data fields to collect: Article title, source (media outlet/site name), publication timestamp down to the minute, one-line summary of key contents, original article URL.
4. Based on the gathered news, suggest 2 content topics and recommended titles that would be great for our team to post on our blog or social media today.

[Output Format]

- Present all results in an easy-to-read Markdown table.
- For overseas articles, include a note at the bottom stating that time zone differences with local time may exist.
- Run a test execution right now to show me the results, then officially register it to the Routines list."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

- When the scheduled time arrives (8:00 AM every morning), Aside's internal scheduler engine automatically creates a new task.

The agent asynchronously navigates to Naver News and Hacker News pages to parse DOM headline text.

It opens the original articles, reads the core text, and extracts one-line summaries and URL snapshots.

Based on the collected data, it constructs recommended writing topics and suggested titles, recording them in Markdown table format within the briefing chat window.

- Precautions:
 - PC power and sleep mode management: Aside routines run on the user's local computer while the Aside browser app is active, not on a cloud server. While routines execute normally during Win+L screen lock or simple display-off states, if the computer enters full Sleep mode or the Aside app is closed, routines will not run; therefore, PC sleep mode settings should be disabled.
 - Free Plan credit consumption calculation: The Free Plan provides 500 credits per month and allows up to 3 active routines. Since a single briefing collecting 5 articles consumes approximately 29 to 32 credits, running a daily briefing requires roughly 900 credits per month, which may exceed the default free credit limit. By logging into and connecting your existing ChatGPT Plus/Pro or Claude Pro/Max subscription in Aside settings (Settings > Models), you can run daily morning briefings for free within your personal AI subscription quota without deducting Aside credits.

6.2 Registering and Managing Recurring Work Routines

① Overview

Relying on human memory to issue identical prompts each time for web browser tasks that repeat daily, weekly, or monthly (e.g., weekly performance metrics collection, monthly report exports at midnight on the 1st of each month, regular course registrations and ticket bookings, daily data synchronization at designated hours, etc.) is highly inefficient. As manipulation procedures or conditional judgements grow more complex, people are prone to omitting instructions or phrasing prompts differently each time, disrupting the consistency of the results.

Aside AI browser's routine management system provides not only a simple Cron method that triggers at fixed times, but also a Heartbeat method that wakes up existing conversations to preserve prior context, as well as an intelligent routine recommendation feature (Find Routine Suggestions) where the AI analyzes user behavior patterns to proactively identify and suggest recurring tasks. Furthermore, developers and power users can integrate Aside CLI and MCP to bypass the Free Plan's limit of 3 active routines, invoking Aside agents in the background via macOS launchd or external schedulers to implement sophisticated, enterprise-grade automation orchestration.

② Real-World Use Cases

- Three practical paths for routine creation and management: There are three primary methods to create routines in Aside. First, the UI method, where you click Settings > Routines > Create to directly enter the time and prompt. Second, the natural language method, where you prompt during a conversation, 'Repeat this task every Monday at 9:00 AM,' having the AI directly register the schedule into the settings DB. Third, the automatic discovery method, where you click the 'Find Routine Suggestions' button in Settings, allowing the AI to analyze recent conversation histories and browsing patterns to independently identify and recommend tasks with high repetition potential.
- Regular weekly Saturday 7:00 AM KTX/SRT train seat reservation and payment routine: A real-world case of a user who needs to book train tickets for specific time slots every week to attend lecturing engagements. The user kept Aside running on a Mac mini and configured the routine while lowering the credit card limit used for payments. The moment seat reservations opened one month in advance at 7:00 AM on Saturday, the Aside routine automatically connected, reserving the seat and completing payment within seconds. The user automated securing train seats without having to wake up early and wait in line.
- Daily 9:00 AM deposit and savings financial disclosure data collection and storage verification: A case of a regular routine collecting Financial Supervisory Service (FSS) disclosure data every day at 9:00 AM. Rather than simply dispatching a request and concluding, it implemented rigorous routine verification logic that does not treat an HTTP 202 response code as simple completion, but verifies all the way until the DB snapshot's 'last saved timestamp' is actually updated.

Integrating external agents (Claude Code + launchd) to bypass the 3-routine limit: A case leveraging Claude Code and the macOS launchd scheduler to overcome the Free Plan's constraint of 3 active routines. On the 1st of every month at 12:30 PM, launchd wakes up an execution script, which calls Aside MCP/CLI to automatically extract monthly internal KPI metrics and cumulatively store them into an Excel file spanning 26 sheets, completing seamless background automation.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Routine Type: Cron Routine (Every Monday at 09:00 AM)
- Monitoring & Management Guidelines: Skip overlapping previous execution if an existing routine is still running

[User Prompt]

"Officially register the task just completed as a recurring routine that runs every Monday at 9:00 AM.

[Routine Execution Guidelines]

1. Routine Name: 'Weekly Social Media Performance Collection and Notion Calendar Update'
2. Execution Time: Every Monday at 09:00 AM (Korea Standard Time)
3. Actions: Navigate to YouTube Studio and Naver Search Advisor, extract last week's views, clicks, and impressions, and update the Notion calendar property to 'Done'.
4. If the previous week's routine task is still running, do not run concurrently; skip this execution.
5. Once configured, verify in the Settings > Routines list that the routine has been properly created, and inform me of the next scheduled execution time."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

The routine schedule and execution prompt are saved to the configuration DB via the user's prompt instruction or the Create button in Settings.

When the designated time arrives, the scheduler initiates a new conversation (Cron) or wakes up an existing conversation (Heartbeat).

The agent visits the specified websites to execute tasks, presenting a Final Confirm guardrail immediately prior to sensitive actions such as payments or transmissions.

Once the task finishes, results are recorded, and the countdown to the next scheduled execution time is updated.

- Precautions:

- Distinction between Cron and Heartbeat routines: When creating a routine, select 'Cron Routine' to open a clean new chat window for independent task execution each time, or select 'Heartbeat Routine' to preserve previous conversation context and browsing history while carrying out new sequential instructions.

- Security limit settings for automating payments and sensitive actions: When delegating autonomous payments to a routine, as in the KTX booking case, set a low per-transaction payment limit on the relevant card or properly incorporate Aside's Final Confirm approval popup policy to prevent accidental large incorrect charges.

- Limitations of routine control via CLI/MCP: While current versions of Aside MCP and CLI tools fully support directly executing the browser agent (aside exec, aside repl), they do not provide APIs to programmatically 'create' or

'modify' routines inside the Aside app itself. Therefore, registering new routines must be designed either by interacting directly within the Aside app UI or by invoking Aside CLI commands from external OS schedulers (Windows Task Scheduler, macOS launchd, Linux cron).

Chapter 7. Developer & Advanced Automation Integration Use Cases (CLI / MCP / REPL)

7.1 Collaborating with Coding Agents (Integrating Claude Code, Codex, Cursor, etc. with Aside CLI/MCP)

① Overview

In modern software development, the use of coding agents (harnesses) such as Claude Code, OpenAI Codex, Cursor, and OpenCode—where AI directly writes and modifies source code in terminal or editor environments—is growing rapidly. However, while these coding agents demonstrate outstanding performance in local file system manipulation and code builds, they face a fatal barrier when entering authenticated web services (internal admin portals, Google Sheets, external communities, SaaS behind login walls, etc.) to fetch data directly or manipulate screens due to browser authentication hurdles.

The Aside AI browser systematically supports command-line interface (CLI), Model Context Protocol (MCP), and Skill integration features for developers. Through a single installation command (`curl -fsSL https://releases.aside.com/install.sh | bash` or `aside skills install`) or MCP server configuration (`aside mcp` or `claude mcp add aside -- aside mcp`), coding agents operating in the terminal can freely invoke the Aside browser as their powerful hands and feet (tools).

This enables coding agents to leverage the `exec` tool for manipulating browsers with maintained login sessions, the `repl` tool for deterministic script

control, and the `memory_search` tool imbued with user business context, implementing integrated automation orchestration that seamlessly completes tasks from collecting data hidden behind web sessions to local code modifications and verification.

② Real-World Use Cases

Automated collection and analysis of 104 GPTers use cases using Claude Code and Aside MCP: A case where a management and research manager at a small-to-medium manufacturing enterprise connected Aside MCP to Claude Code to automatically collect 104 study posts (approximately 440,000 characters) containing the keywords 'aside' or its Korean transliteration from the GPTers community. By extracting webpage session authentication tokens via Aside REPL and calling the internal GraphQL search API, 104 case post bodies were precisely gathered out of 517 candidates. Cross-analyzing the 9 identified automation archetypes with the user's business context accumulated in Aside Memory (Head of Manufacturing, factory equipment dashboard, weekday 8:00 AM KRX briefing routine), the agent recommended 3 optimal tailored automation initiatives, including 'cumulative Excel export of monthly internal system metrics'.

- AutoClick service development and browser testing automation: A case where a former Meta senior engineer (Silicon Valley Developer) integrated Aside CLI/MCP and the `/aside` skill with Codex and Claude Code during the development of their proprietary service, AutoClick. By having the Aside browser interact with the app, the agent generated a UI/UX improvement report from an agentic perspective. After modifying code and creating a Pull Request (PR) in Codex, the developer invoked the Aside browser skill to verify live web browser interface behavior.
- Collecting Coupang power banks with a single terminal command: A developer invoked Aside CLI in a terminal environment to execute the

command: 'Find 5 power banks on Coupang with 10,000mAh, 20W or higher, and under 30,000 KRW.' An Aside agent session ran in the background, browsing through 16 pages of search results to parse the names, prices, capacities, URLs, and screenshot information for 5 qualifying products, returning neatly organized results to the Claude Code terminal.

- Terminal-based smoke testing and session control: Instantly initiated smoke tests on a local development server from the terminal using the command `aside "Open localhost:3000 and run a smoke test"`, resumed control over existing browser task sessions via the `--session <session-id>` option, or switched between multi-account operations using the `--account` option.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Integration Method: Invoke /aside skill or MCP tool in Claude Code terminal
- Target Tools: Aside CLI and my local project workspace

[User Prompt]

"Use Aside MCP tools in the Claude Code terminal.

[Step 1: Session and Data Collection]

1. Open active tabs for Naver Search Advisor and Google Search Console where I am logged in.
2. Extract search impressions, clicks, and top 10 keywords for the last 30 days.

[Step 2: Local Code Integration and Report Generation]

1. Convert the collected data into JSON format and save it as 'src/data/search_metrics.json' in my project folder.
2. Verify whether data definitions and numbers match 100% with existing dashboard source code ('src/components/Dashboard.jsx').
3. If verification reveals no discrepancies, generate a metrics summary report Markdown document and provide a completion report."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

The coding agent (such as Claude Code) receives instructions and loads the built-in aside mcp or /aside skill manual.

The agent invokes the Aside CLI/MCP process in the background to launch an Aside desktop browser session.

The Aside browser navigates to websites with active login sessions to perform specified data extraction and screen verification.

The collected data structure is transferred to the coding agent terminal via the MCP channel.

Based on the received data, the coding agent modifies local files (JSON, JSX, Python, etc.), verifies that formulas and metrics match, and outputs a final completion report.

- Precautions:

- Windows install key permission errors (aside.install.sig): When a coding agent (such as Codex) executes the CLI under a separate sandboxed virtual account in Windows environments, login key file permission access errors may occur. In this case, run commands from a standard user-permission PowerShell window outside the sandbox, or grant the agent permission to execute outside the sandbox.

- Specifying credentials path (ASIDE_CLI_AUTH_PATH): When running CLI batch operations on another computer, remote server, or sandboxed environment, always preset the environment variable export `ASIDE_CLI_AUTH_PATH="/allowed/custom/path/aside-auth.json"` to ensure credentials files import properly.

7.2 Web Service UI/QA Testing (Post-Build Auto Login, Signup Screen Captures, and Anomaly Verification)

① Overview

When developing web applications or front-end services, the QA (Quality Assurance) testing process—where developers manually open browsers to fill out signup forms, attempt automated logins, and verify whether modal popups trigger properly upon button clicks after adding new features or refactoring code—is arduous and repetitive. Traditional browser test scripting methods like Playwright or Selenium require developers to hand-code every DOM selector and wait logic, suffering from chronic maintenance headaches where test scripts break whenever website designs or class names change slightly.

The Aside AI browser combines the AI agent's ability to visually read and manipulate web interfaces just like a human, delivering a mature automated UI/QA testing environment. When a developer deploys a new build to a local development server (such as localhost:3000), the coding agent invokes Aside to complete signup flows, inspect form input states across diverse aspect ratios, and automatically capture screenshots at points of failure to submit alongside inspection notes. In particular, leveraging the built-in password repository (Vault) and dedicated testing profiles (Profiles) enables building safe, fully automated QA pipelines without corrupting production data.

② Real-World Use Cases

- AutoClick service UI/UX improvement and browser QA testing pipeline: A case where the developer of the thumbnail generation service AutoClick refactored source code and integrated Codex with Aside browser skills to conduct browser QA testing. By having the agent navigate to the service interface, fill out forms, click thumbnail generation buttons, and observe rendering processes, the agent proactively proposed UI structural improvements (form input scripting, clarifying label selection states, etc.) to optimize token efficiency. Based on these proposals, the developer opened code improvement PRs and conducted cross-model reviews with Claude, enhancing deployment stability.
- Automated testing and screenshot submission upon GitHub errors: When errors occurred during GitHub issue triage or code builds during developer workflows, the Aside agent navigated directly to the test site, executed clicks and form entries, captured screenshots of the exact screens where malfunctions occurred, and provided them to the developer as root-cause analysis assets.
- 30-minute interval dashboard error rate monitoring routine post-deployment: Executed a Launch Checks routine immediately following a new service deployment, automatically accessing the admin dashboard every 30 minutes to inspect and record system error rates and response latencies.

Frame-level verification of personal information exposure in study session recordings: Detected three timestamps where student phone numbers were exposed down to 0.25-second frame intervals across a 2-hour-21-minute study recording, navigated into YouTube Studio Editor, and verified via screen capture screenshots that trimming and blurring operations for those sections were accurately completed.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Read only or Guard mode
- Target Server: My local development server (@localhost:3000)
- Profile Configuration: Specify Testing Profile

[User Prompt]

"Navigate to my local development server 'http://localhost:3000' and execute a UI/QA test for the complete signup and service usage workflow.

[Testing and Verification Scenarios]

1. Click the 'Sign Up' button on the main screen and fill out the registration form with test account details.
2. After registration, navigate to the login page and attempt automated login using the test credentials stored in Vault.
3. Upon entering the dashboard, click through 3 major menu sections sequentially, checking for screen loading glitches or console errors.
4. If there are areas where screen layouts break or error messages appear, capture full screenshots of those screens.

Once testing is complete, compile a QA report summarizing pass/fail status, elapsed time, and a list of captured error screenshot files."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon receiving the prompt, the Aside agent navigates to the specified local development server URL (localhost:3000).

It visually detects screen DOM elements and button coordinates, injecting test data into signup and login input forms.

During page transitions at each step, it checks asynchronous rendering states while monitoring browser console errors and network response status codes.

Upon detecting anomalies or layout discrepancies, it takes screenshots and saves them as image files in the local workspace directory.

After completing the entire scenario, it returns a comprehensive QA report detailing the test pass rate and paths to captured screenshot files.

- Precautions:

- Separating dedicated test profiles and Vault accounts: To avoid corrupting data through production cookies or live credentials, always create a separate 'Testing Profile' under Settings > Profiles when conducting QA tests, running automation in an isolated environment.

- Pausing at identity verification and CAPTCHA checkpoints: If the signup flow includes mobile identity verification (e.g., PASS) or Google CAPTCHA, AI cannot autonomously bypass them. For these sections, explicitly specify in the prompt: 'Pause upon encountering identity verification screens and alert the human user,' preventing the agent from being hard blocked or stalled.

7.3 Deterministic Browser Automation (Script-Based Direct Web Inspection via REPL)

① Overview

Autonomous LLM-based agentic browsing is useful for freely exploring unfamiliar websites or variable web interfaces, but because it relies on autonomous decisions at every step, operational paths can vary slightly each run, incurring token and credit consumption. Conversely, for advanced developer workflows that require clicking specific DOM elements with precise JavaScript commands on the same site daily, parsing network tokens to call APIs directly, or capturing pixel-perfect cropped screenshots of designated areas, zero-error deterministic control is strictly necessary.

Aside REPL (`aside repl`) provides an interactive execution environment to run Playwright-style JavaScript scripts directly inside the browser, meeting these exact developer needs. Rather than relying solely on stochastic AI reasoning, it utilizes precision browser automation functions like `openTab`, `querySelector`, and `gql` to manipulate web interfaces directly at sub-second speeds, flawlessly executing screenshots, file downloads, and network data parsing. This allows developers to construct an ultimate web automation toolkit combining autonomous AI flexibility with deterministic script precision.

② Real-World Use Cases

- High-volume collection of 104 GPTers use cases using Aside REPL: A real-world case utilizing Aside REPL to avoid the inefficiency of clicking 'Load More' hundreds of times when gathering use case posts from the GPTers community. Inside the REPL, a JavaScript script directly parsed session auth tokens (`await gql('query { tokens ... }')`) and called the site's internal GraphQL search API. This enabled accurately gathering 104 post bodies (approximately 440,000 characters) containing the keywords 'aside' or its Korean transliteration from 517 candidates in just seconds without data loss.

Precision manipulation of specific DOM elements via deterministic scripting: Used the command `aside repl "const p = await openTab('https://example.com')"` to open browser tabs without latency, directly reading target ID and class elements on the page to capture screenshots or complete high-speed file downloads.

Script parsing AWS official documentation RDS specifications and converting to Excel: When accessing the AWS official documentation site to gather latest RDS instance type specifications, REPL control was used to accurately parse specification table text structures, converting and saving them as JSON and Excel (.xlsx) files in a local directory.

Precision collection of 39 Naver Search Advisor and GA4 dashboard screenshots: Captured 39 exact interface screenshots across Naver Search Advisor, Google Search Console, and GA4 admin screens via deterministic scripting, collecting ambiguous units such as '2.3 hundred' impressions verbatim without distortion to cross-reference with Excel numerical tables.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Execution Tool: Aside REPL (`` aside repl ``)
- Destination Path: My project workspace folder

[User Prompt]

"Use the Aside REPL environment to execute deterministic script-based data parsing on the specified webpage.

[REPL Execution Script Guidelines]

1. Open the page using the command ``const tab = await openTab('[suspicious link removed]')``.
2. Accurately extract article titles (`'.titleline > a'`) and score (`'.score'`) text among DOM elements using `querySelectorAll` scripts.
3. Structure the extracted top 10 article items into a JSON format.
4. Capture the main table region on the current screen using the ``tab.screenshot()`` function and save it to my workspace folder as `'hacker_news_snap.png'`.

Once complete, display the extracted JSON data text in the chat window and finish the task."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon submission of user instructions, the coding agent or terminal invokes the aside repl environment.

The REPL engine directly injects Playwright-style JavaScript code (`openTab`, `screenshot`, DOM parsing syntax) into the browser process.

Bypassing LLM inference processes, it directly accesses the browser DOM and network sessions to execute high-speed manipulations and screenshots at sub-second speeds.

It returns parsed data and file storage results as REPL execution outputs.

It outputs the collected structured data to the chat window and verifies local file creation to conclude.

- Precautions:

- Script errors upon website DOM structural changes: Because deterministic scripts rely on designated CSS selectors or GraphQL query structures, any modifications to target website HTML class names or API schemas will trigger execution errors. Prior to authoring scripts, it is advisable to include validation logic that briefly verifies DOM presence within REPL.

- Compliance with robots.txt and Terms of Service: When using REPL for rapid API calls or high-volume crawling, introduce request delays and strictly adhere to legal guidelines so as not to violate target website robots.txt rules, server rate limits, or terms of service.

Epilogue. Risk Management and Guidelines for Effective AI Agent Use

① Overview

The Aside AI browser inherits authenticated live browser sessions left open by humans to directly handle tasks deep inside the web, including payment checkout pages, internal ERPs, social media, and financial portals. However, entrusting your logged-in accounts and file system control to an AI agent offers formidable convenience while simultaneously demanding a clear risk management framework in terms of security and cost.

Representative risk factors include multi-factor authentication (MFA) challenges and bot checks (CAPTCHAs) encountered during website access, security session expirations, the risk of account suspensions if attempting to bypass CAPTCHAs, and rapid credit depletion caused by indiscriminate web browsing.

Risk management and guidelines for effective AI agent use are necessary to proactively prevent these issues. By configuring Aside Vault-based password security policies, isolating dedicated profiles, establishing human-in-the-loop protocols upon encountering CAPTCHA or MFA challenges, and delineating clear division of labor between public data (APIs) and authenticated data (browser agents), organizations can maximize cost efficiency while securely embedding browser automation into standard operations.

② Real-World Use Cases

Google Search CAPTCHA detection and 25-second immediate abort safety mechanism: A case where Google systems detected bot-like behavior and presented a CAPTCHA challenge during an attempt to search Google with date range filters to find brand mentions. Rather than forcibly bypassing the CAPTCHA in violation of terms or fabricating false results, the Aside agent aborted execution within 25 seconds and accurately reported the situation to the user. This prevented wasteful credit depletion and averted account blocking risks.

- Corporate banking portal 15-minute session timeouts and MFA barriers: During Windows environment testing spanning 35 practical tasks over 5 days, an attempt was made to download account statement PDFs from a corporate banking portal for daily financial settlements. Due to the financial portal's strict 15-minute inactivity timeout and recurring SMS/push MFA challenges required upon navigating menus, sessions expired four consecutive times while the agent waited for human approval. Ultimately, this high-security financial task was redefined to be manually handled by a human in 90 seconds rather than through autonomous AI execution.

- Role division between APIs and browser agents when scraping Naver Cafe: Exploring Naver Cafe for 3 minutes and 4 seconds using the Aside browser agent to collect 11 brand mentions consumed 145 credits (roughly 30% of the monthly Free Plan quota). However, utilizing the Naver Open API endpoint demonstrated that identical Cafe post searches could be handled instantly with a single API call without deducting credits. Consequently, a 4-stage hybrid pipeline was built—delegating public data scraping to APIs, semantic analysis of collected data to AI models, and notifications to Slack—dramatically reducing operating costs.

100% tracking success on shipping carrier portals via dedicated profile isolation: Over 5 days of conducting 14 runs and tracking 85 containers, creating a dedicated corporate profile instead of using the default browser cluttered with personal tabs prevented cookie collisions. As a result, operations across three carrier portals—Maersk, MSC, and CMA CGM—achieved a 100% success rate without a single session disconnect, saving 35 minutes of working time every morning.

③ Prompt Example

P R O M P T

[Request Conditions]

- Execution Permission: Guard mode
- Security Policy: Ask First (Mandatory human approval required for payments, message transmissions, or account closures)
- Password Policy: While unlocked (Allow automated login only while the vault remains unlocked)

[User Prompt]

"Navigate to our internal materials management portal and overseas supplier websites to inspect scheduled incoming shipments for this week.

[Risk Management and Operational Rules]

1. Restrict Search Scope: Do not perform infinite scrolling or browse all pages; collect only within the top 10 items on the first search results page.
2. Apply Delays: After clicking dropdown menus or search buttons, always wait 3 seconds until screen spinners disappear before reading data.
3. Response to MFA and CAPTCHA: If an OTP, multi-factor authentication (MFA), or CAPTCHA screen appears during login, never attempt to bypass it; pause the task and alert me within 25 seconds.
4. Negative Constraints: Aside from data collection, never click 'Submit', 'Modify', or 'Pay' buttons.

Once collection is complete, organize the scheduled incoming dates and items in a Markdown table, and save it to my workspace folder as 'material_receipt_inspection_report.csv'."

④ Expected Execution Flow and Precautions

- Expected Execution Flow:

Upon receiving the prompt, Aside verifies the designated corporate profile and Guard permission status.

The agent navigates to the target portal address and, upon detecting login forms, automatically injects credentials via the password vault without

exposing plaintext.

Following click actions, it adheres to the designated 3-second DOM delay, reliably extracting data.

If a CAPTCHA or MFA popup occurs midway, it pauses operations immediately and hands screen control over to the user.

Once the user completes authentication, it resumes the task from the point of interruption to generate and save structured reports.

▪ Core Guidelines and Precautions:

Category	Primary Risk Factors	Response Guidelines and Operational Best Practices
Profile Management	Cookie conflicts with ~40 personal tabs and browser extensions	Isolate sessions by creating a dedicated Corporate Profile for automation
Password Security	Plaintext password leaks to AI models and unintended actions	Leverage Aside Vault and configure AI access policy to While unlocked, permitting access only while the user is present
Authentication & CAPTCHA	15-minute session timeouts and bot detection blocks	For financial portals with heavy 2FA/MFA, have a human complete pre-authentication prior to batch execution before handing control to AI
Cost (Credits)	Excessive credit consumption from infinite scrolling on 0-result searches	Explicitly declare search upper bounds such as "collect within top 1 page" or "limit to max 10 items", and offload API-capable scopes to APIs

▪ Three Principles for Division of Labor:

- Structured high-volume data collection: Leverage official APIs (Naver Search API, Public Data API) to gather data without consuming credits.
- Web screen manipulation behind login sessions: Concentrate deployment of the Aside browser agent on areas requiring authenticated sessions, such as YouTube Studio audience retention curves, internal ERPs, and secure portals.
- Irreversible sensitive actions: Maintain a Human-in-the-Loop structure where a human reviewer personally inspects and performs the final click for payments, public postings, and bulk SMS transmissions.

References

The cases and explanations in this book were compiled based on the following videos, blog posts, and official help documentation.

1. "No Need for Humans Anymore": How to 100% Automate Naver Blog with AI (feat. Aside)
2. "The Ultimate Automation": All-in-One Summary from Setup to Installation of the Trending AI Aside Browser
3. From Interest Rate Comparison to Maturity Payout Calculation: Building a Savings Dashboard with Aside
4. How to Change the Translation Button to Korean | AI Browser Aside Pro Tip
5. Recommended Browser Automation, First Setup on Windows — AI Browser Aside
6. Still Doing It Manually? Aside Automation Even AI Beginners Can Use
7. Aside Browser Korean Language Settings, 7 English Menus and Usage
8. Aside Browser: What Makes It Different? Everything from Installation to Precautions
9. How to Use Aside: This Is How It Works — AI Even Sends Text Messages
10. Escaping Complete Beginner Status with Aside | From Installation to Every Single Menu, Delegating Browser Tasks to AI
11. Aside AI Browser Usage Guide, Summarized from 18 Pages of Official Documentation #Aside #AIBrowser
12. Must-Read Before Adopting Aside! Deciding If It Fits You in Just 5 Minutes!
13. Aside Browser — Delegating Browser Work to AI | AidenAI
14. Aside Review: I Even Delegated Cutting 3 Personal Information Segments in a Study Recording
15. Aside AI Browser Practical Usage | From Market Research to Coupang Price Comparison and Checkout [#ChatGPTEducation #AIEducation #ChatGPTInstructorRecommendation]
16. Aside + JEV Demonstration and Easy Explanation
17. Registering the Taekwon V Site to 4 Search Engines with Aside

18. How Far Can AI Browsers Go These Days? I Tried Aside Myself | Installation, Concepts, and Complete Usage Guide [Aside Browser]
19. Now Delegate Browser Control to Aside~ For Those Frustrated When Delegated Tasks Halt on Browser Control!!
20. Hire This AI Assistant Before Hiring Employees | Aside AI Browser #shorts
21. I Switched After Using Chrome for 20 Years | Practical Experience with Aside AI Browser
22. Aside AI Browser Commanded by Words Instead of Chrome Part 1 | Comprehensive Beginner's Guide to Installation, UI, and Pricing
23. Why Do You Need an AI Browser When Chrome Works? A Thorough Comparison of Aside, ego lite, and Computer Use
24. Claude Code Browser Automation — Aside CLI Usage | AI Browser Aside
25. An AI Browser That Moves Up Clock-out Time by 30 Minutes! Works Automatically from Search to Click | Aside x Claude (Free Prompts Provided)
26. Comparing Aside and Claude Code Across Five Harness Elements
27. Organizing Scattered Aside Tasks into a Single Repository
28. If You Could Only Use One AI, Use This | Aside Beginner Guide from Installation to Hands-on Practice
29. I Tried the AI Browser 'Aside'... Managing 5 Social Media Channels Changed Completely
30. You Must Try an AI Browser. It's a Whole New World! (Includes Easy Connection Methods feat. ego lite)
31. AI Browser Aside Usage Guide Part 2 | Browser Fundamentals: New Tabs, Side Panel, Split Tabs
32. AI Browser Aside Usage Guide: From 5-Step Setup to Delegating Tasks via Text in One Go
33. Revealing Real Practical Usage of AI Browser Aside
34. AI Browser Aside Review: Lessons Learned After Exhausting Free Credits in 20 Minutes

35. AI Browser Aside Released on Windows! Plus Combinations That Cut Credit Consumption in Half
36. I Made an AI Browser Use My App | Practical Aside Usage by a Senior Meta Engineer
37. I Delegated Naver Blog to AI | Free Beginner Lecture from Installation to Automated Publishing
38. I Assigned Real Tasks to AI | Aside Browser Collecting Even Real Estate and Social Media Data
39. Aside Browser Usage: Pros and Cons, Pricing, and Reviews of the AI Agent Browser
40. Aside Browser Installation and Initial Setup: From Download to Your First Task | Hapni
41. Aside Browser | Aside Help Center
42. Aside browser just got 90% lighter and 14x cheaper to run
43. Aside Browser: The New Agentic Browser You Can't Miss!
44. Aside, the AI Browser That Works Across Sites You're Signed Into: How Far Does the Free Tier Go? | Promptating
45. Connecting Aside to Claude Code Skills | BLUEBUG'S BLOG
46. Connecting Aside to Claude Code: Selecting My Automation Tasks and Automating the First Task Monthly
47. Components | Aside Help Center
48. Configure AI providers | Aside Help Center
49. Configure password autofill | Aside Help Center
50. Get started | Aside Help Center
51. I Asked an AI Browser to Do My Work | Aside #Shorts
52. I Tested Aside on Windows for 5 Days: What Happened When I Let an AI Browser Run My Daily Portals – SmartWorkBase
53. Install the Aside AI Browser: From Windows and macOS Setup to Your First Task
54. Manage memory | Aside Help Center
55. Manage privacy and local data | Aside Help Center
56. Manage your Aside subscription | Aside Help Center

- 57. Run tasks | Aside Help Center
- 58. Set agent permissions | Aside Help Center
- 59. Troubleshoot Aside | Aside Help Center
- 60. Use Aside as your browser | Aside Help Center
- 61. Use Password Manager | Aside Help Center
- 62. Use routines | Aside Help Center
- 63. Use the CLI, MCP, and REPL | Aside Help Center
- 64. Use the side panel | Aside Help Center
- 65. Use Ultrabrowse | Aside Help Center

Aside AI Browser Practical Workflow Automation Use Cases

ISBN |

Author | Kyung-jin Kim

Publisher | Kyung-jin Kim

Published by | Attorney Kyung-jin Kim Publishing

Registration | March 10, 2025 (No. 2025-000015)

Address | Suite 304, Baekil Bldg., 91 Jeonnong-ro, Dongdaemun-gu, Seoul, Korea

Tel | +82-2-6338-1905

Email | kimkj008@gmail.com

© Kyung-jin Kim 2026

All rights reserved. Unauthorized reproduction or distribution of this publication is prohibited.

Note: Parts of the content and editing process of this book were assisted by artificial intelligence tools.

Support This AI Library Book

If this book was useful, you can support future translations, public editions, and open AI knowledge projects through PayPal.



PayPal

<https://paypal.me/kimkjki>

Scan the QR code or open the link above.