

Google Antigravity – Agentic Development Platform & Setup Guide

Installation, Configuration & Usage

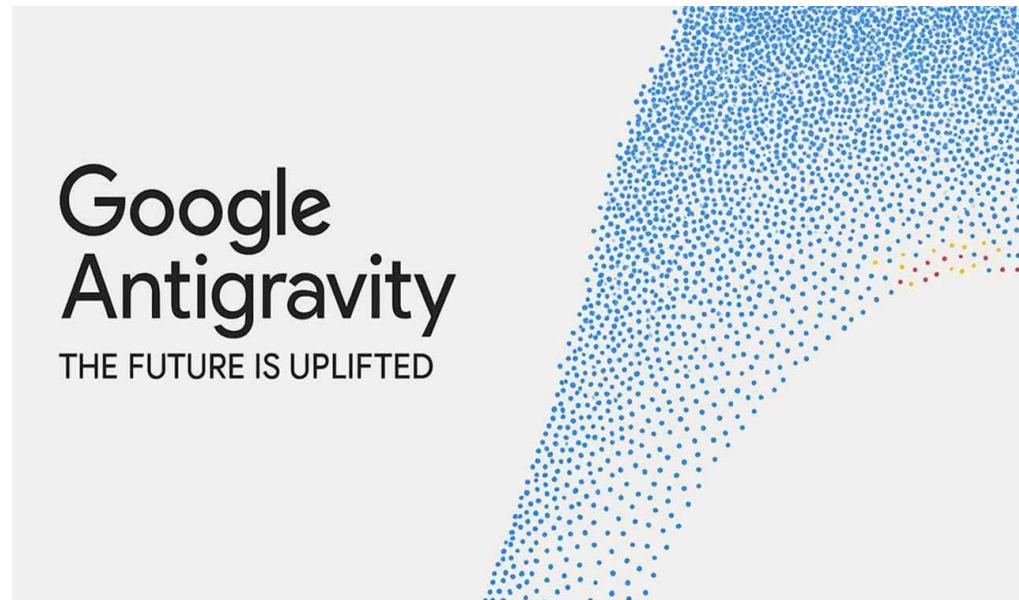


Table of Content

| | |
|---|----------|
| 1. Executive Summary..... | 1 |
| 2. Key Purposes..... | 1 |
| 3. Installation & Configuration..... | 1 |
| 3.1 System Requirements & Download..... | 1 |
| 3.2 Setup & Configuration | 1 |
| 4. Autonomous Agent | 1 |
| 5. Agent Manager..... | 1 |
| 6. Browser Manager..... | 1 |
| 7. Artifacts..... | 1 |
| 8. Conclusion..... | 1 |

1. Executive Summary

- **Google Antigravity:** Agent-based development platform that enables AI agents to plan, execute, and validate development tasks within a familiar IDE.
- Mission Control approach to manage multiple agents in parallel while keeping developers in control.
- Reduces repetitive effort and helps developers focus on decisions, design, and code quality.

2. Key Purposes

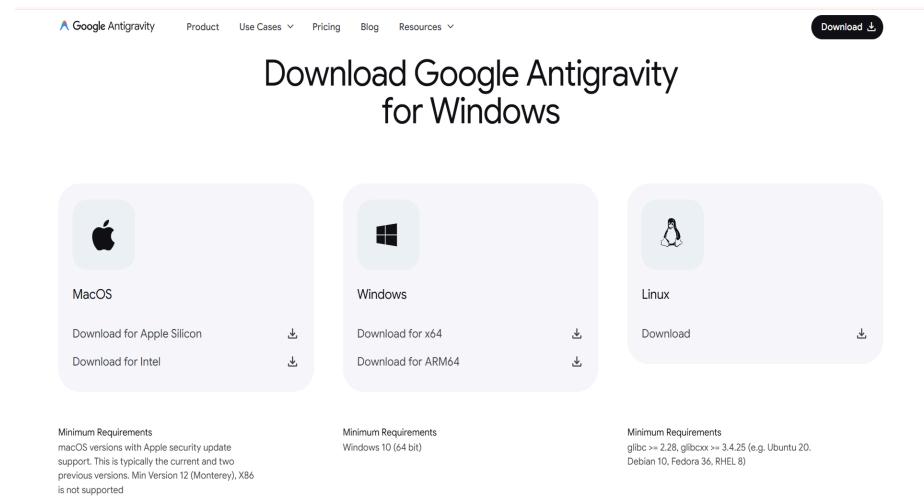
- **Autonomous Agents:** Empowers AI agents to handle full feature requests, plan, code, and verify, reducing developer micromanagement.
- **Mission Control:** A central interface to oversee and interact with these agents, creating workspaces and managing workflows.
- **Parallel Execution:** Enables developers to start new tasks without waiting for others, improving efficiency.
- **Contextual Awareness:** Agents remember context, allowing for seamless handoffs between different tasks
- **Browser Control:** Agents use browsers to gather information and perform visual feedback-driven frontend work
- **Knowledge Base:** Allows developers to build custom knowledge bases from code references.
- **Terminal Support:** Offers control over terminal execution for agents, with configurable permission levels.

3. Installation & Configuration

3.1 System Requirements & Download

Go to the downloads page and click on the appropriate Operating System version that is applicable to your case:

- **Source:** Download the official installer from <https://antigravity.google/download>
- **OS Support:** Available for macOS (Apple Silicon/Intel), Windows, and Linux.



With the installation completed, Google Antigravity is now ready to be launched on the local system.

3.2 Setup & Configuration

After installation, this screen allows users to configure how the agent is expected to behave.

How do you want to use the Antigravity Agent?

Select one of the options below.

Agent-driven development

Agent-assisted development RECOMMENDED

Review-driven development

Custom configuration

Terminal execution policy
Auto

Review policy
Agent Decides

Use the default allowlist for the browser

The agent will occasionally request for review.

Back Next ↗

- **Agent-driven development:** The AI agent takes the lead in the development process
- **Agent-assisted development:** The agent works alongside the developer, providing assistance and occasionally requesting a review.



- **Review-driven development:** The agent performs tasks but relies heavily on the user to review and approve actions.
- **Custom configuration:** Allows the user to manually define specific rules and workflows for the agent's behavior

The next step is to configure **the Editor**. Choose your preferences.

Configure Your Editor

Configure your editor settings below.

Keybindings
Configure your keybindings. Normal Vim

Extensions
Install popular language extensions. Language extensions are required for some Agent features. Other extensions can be found on the marketplace in the editor. Install 7 Extensions

Command Line
Install the command line tool to open Antigravity with `agy`. Install

Back Next ↩

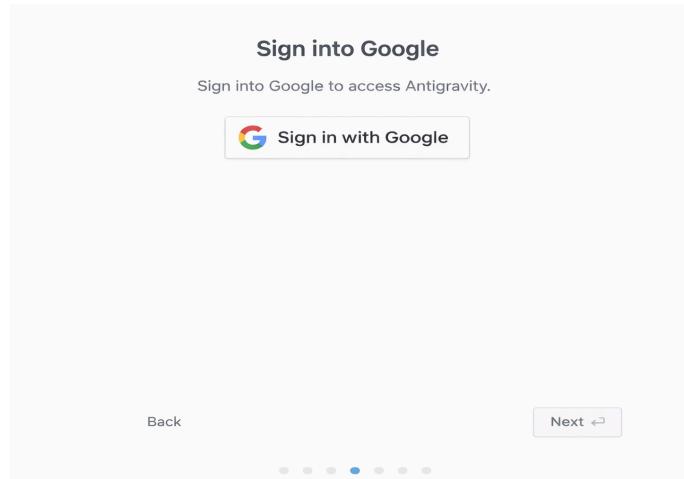
• • • • •

- **Keybindings:** Choose between standard or Vim-style keybindings.

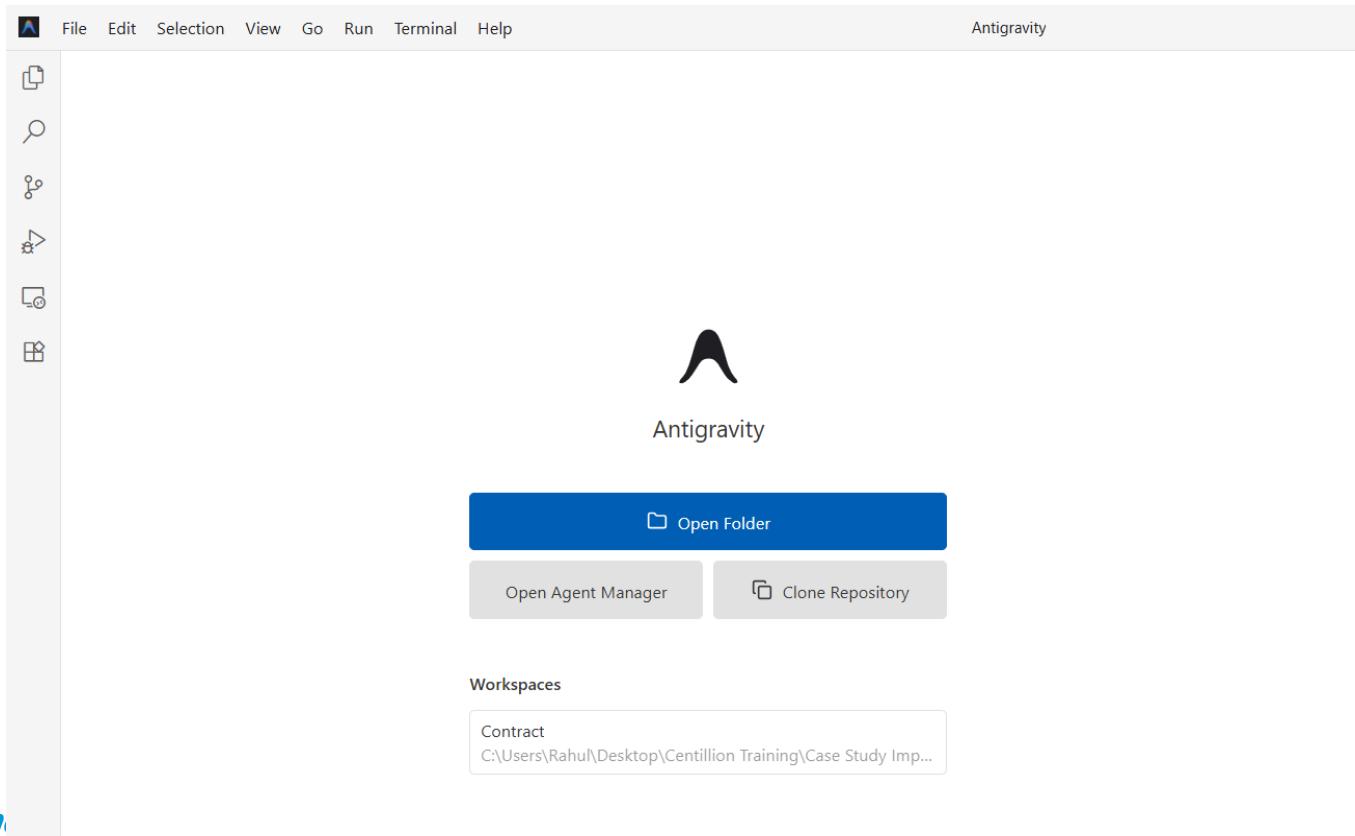


- **Extensions:** Installs essential language extensions required for agent features.
- **Command-Line Tool:** Enables the `agy` CLI for launching Antigravity from the terminal.

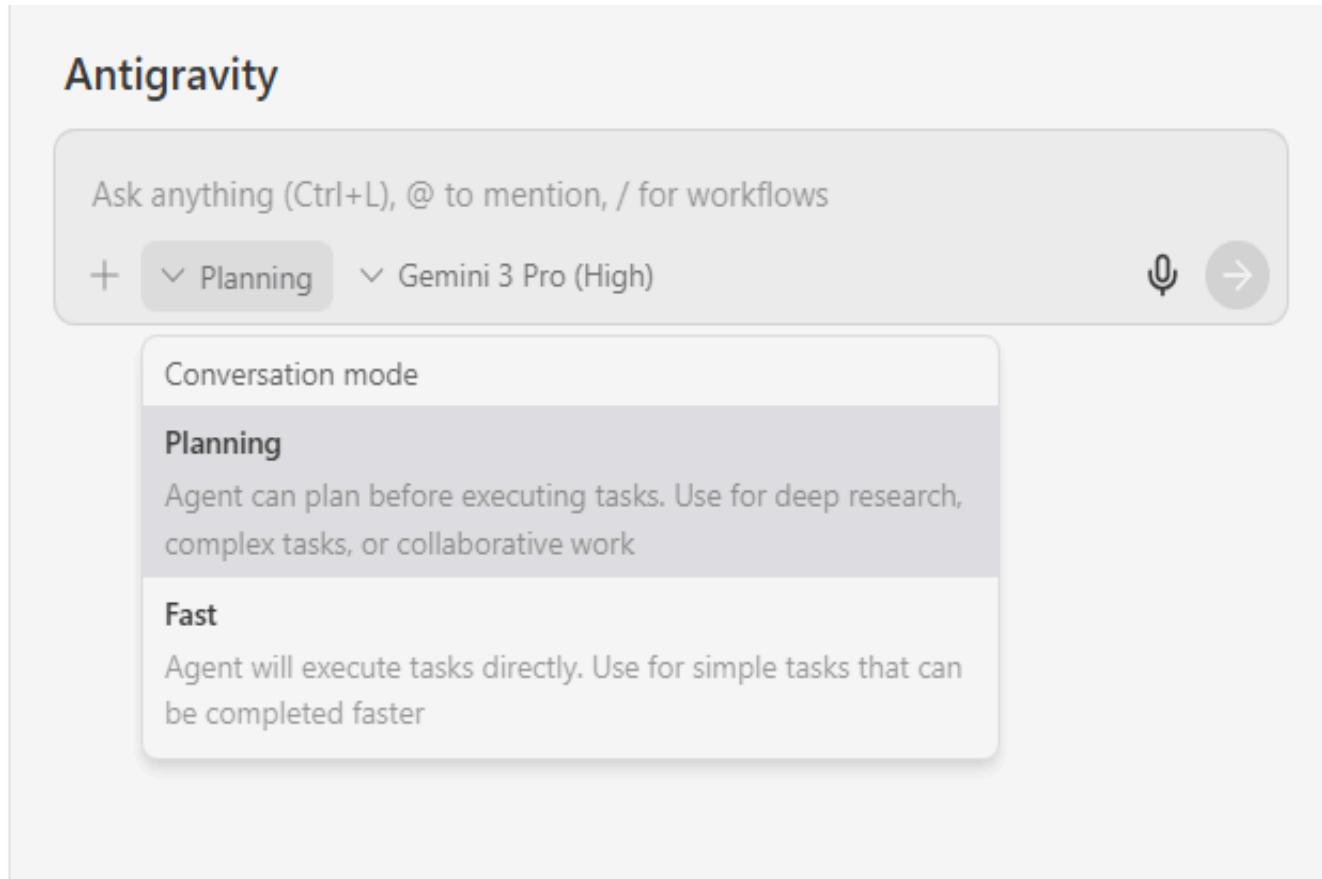
The next step requires signing in with a Google account.



After signing in with Google, Antigravity asks you to open a folder or to clone.



After that, the agent window opens automatically. It can also be opened manually using **Ctrl + Alt + B** or by clicking the toggle icon in the top-right corner.



- **Planning Mode**

The agent plans before execution.

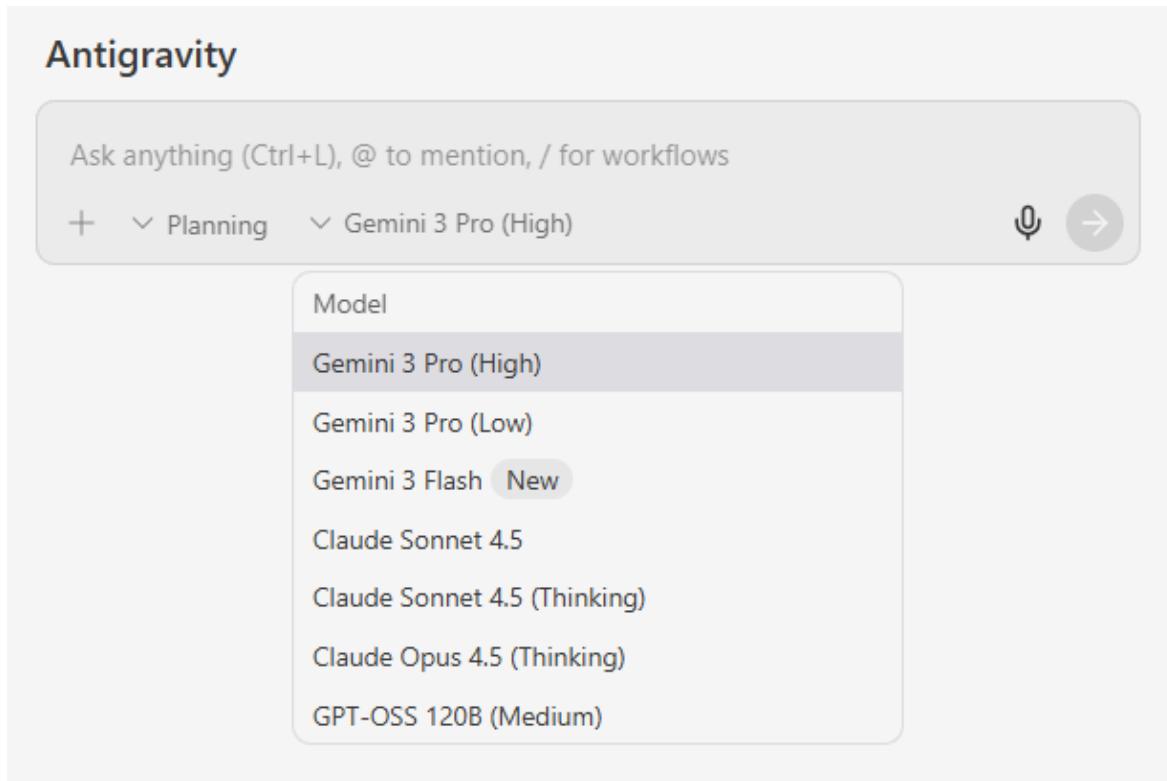
Best for: Complex or research-heavy tasks.

- **Fast Mode**

The agent executes immediately.

Best for: Quick, simple tasks.

Antigravity allows users to choose different AI models based on task complexity,



- **Gemini 3 Pro (High / Low):** Advanced reasoning with selectable performance levels
- **Gemini 3 Flash :** Lightweight model optimized for speed and efficiency
- **Claude Sonnet 4.5 (Thinking):** Balanced reasoning with structured responses
- **Claude Opus 4.5 (Thinking):** High-end model for deep and complex reasoning
- **GPT-OSS 120B (Medium):** Large open-source model for flexible use

4. Autonomous Agent

The agent window provides two distinct conversation modes.

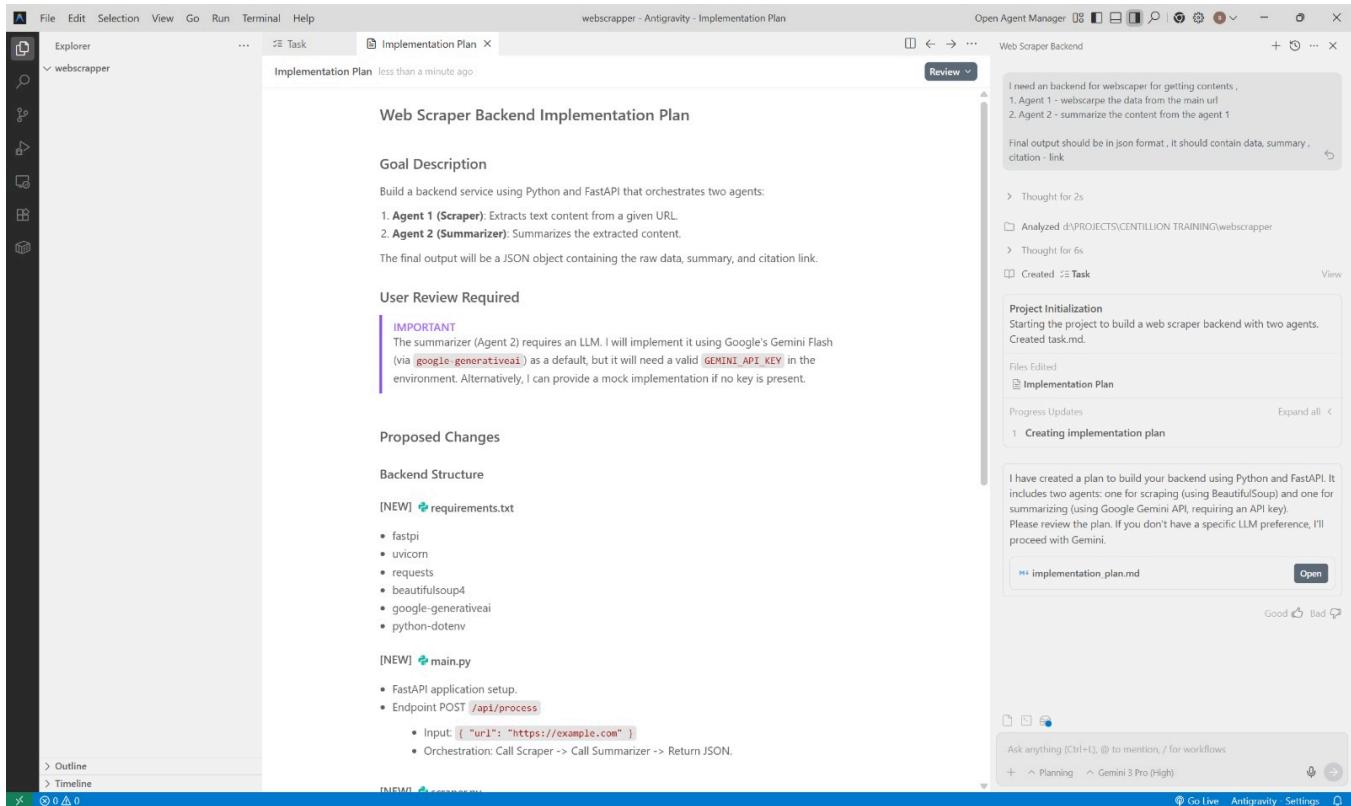
- Planning Mode
- Fast Mode

The agent presents assigned tasks along with a structured implementation plan.

Tasks

- Initialize Project Structure
 - Create virtual environment setup (instructions)
 - Create `requirements.txt`
- Implement Agent 1 (Scraper)
 - Create scraper module
 - Implement `extract_data` function
- Implement Agent 2 (Summarizer)
 - Create summarizer module
 - Integrate with LLM (Placeholder/Gemini)
- Build Main Backend API
 - Setup FastAPI app
 - Create endpoint `/scrape`
 - Orchestrate Agent 1 and Agent 2
- Verification
 - Test with a sample URL

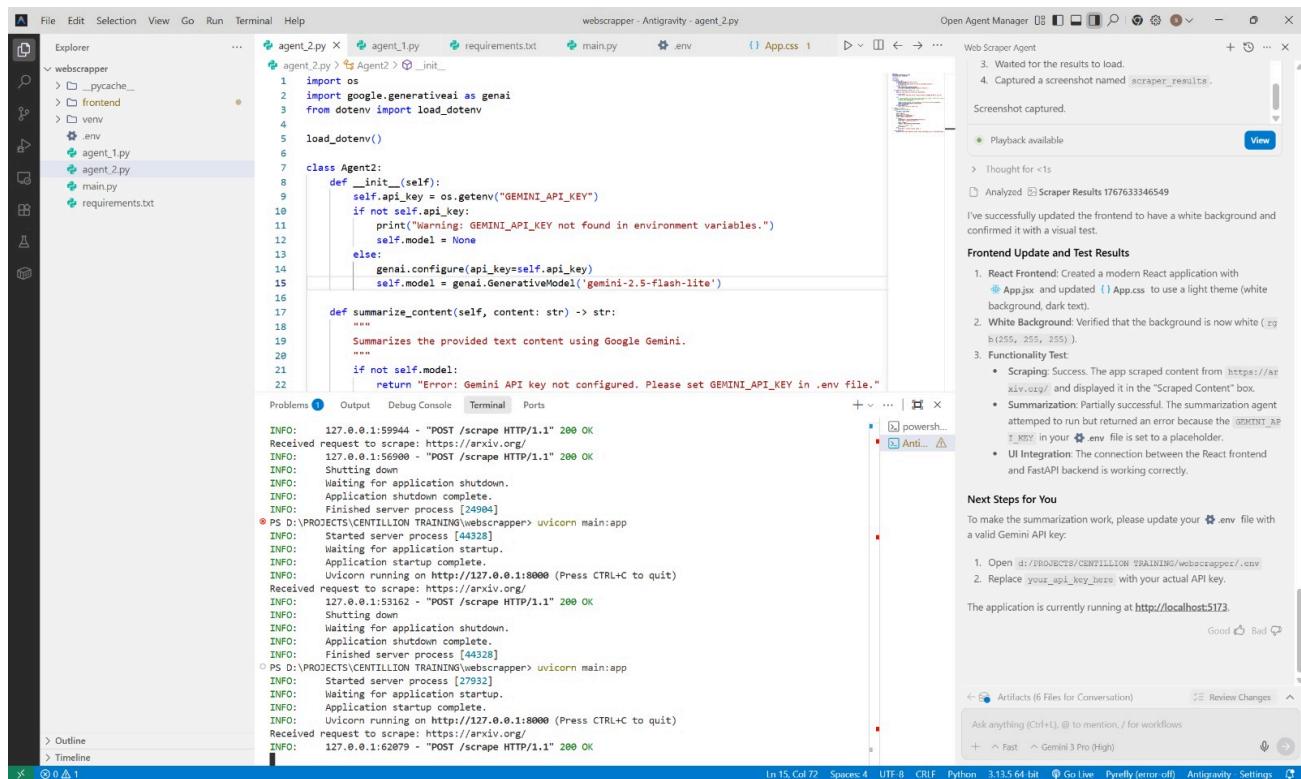
Implementation Plan:



In planning mode, the agent generates tasks and a detailed implementation plan as reviewable artifacts before execution.

- **Tasks:** A clear list of actions required to complete the work
- **Implementation Plan:** An end-to-end project outline covering goals, folder structure, and relevant files

Fast Mode:



The screenshot shows the Antigravity IDE in Fast Mode. The code editor displays a Python script named `agent_2.py` with the following content:

```

agent_2.py
agent_2.py > Agent2 > __init__
1 import os
2 import google.generativeai as genai
3 from dotenv import load_dotenv
4
5 load_dotenv()
6
7 class Agent2:
8     def __init__(self):
9         self.api_key = os.getenv("GEMINI_API_KEY")
10        if not self.api_key:
11            print("Warning: GEMINI_API_KEY not found in environment variables.")
12            self.model = None
13        else:
14            genai.configure(api_key=self.api_key)
15            self.model = genai.GenerativeModel('gemini-2.5-flash-lite')
16
17     def summarize_content(self, content: str) -> str:
18         """
19             Summarizes the provided text content using Google Gemini.
20         """
21         if not self.model:
22             return "Error: Gemini API key not configured. Please set GEMINI_API_KEY in .env file."

```

The terminal output shows the application running on port 8000 and receiving requests from `arxiv.org`.

The right-hand panel displays the following sections:

- Frontend Update and Test Results**
 - React Frontend: Created a modern React application with `App.jsx` and updated `App.css` to use a light theme (white background, dark text).
 - White Background: Verified that the background is now white (255, 255, 255).
 - Functionality Test:
 - Scraping: Success. The app scraped content from `https://arxiv.org/` and displayed it in the "Scraped Content" box.
 - Summarization: Partially successful. The summarization agent attempted to run but returned an error because the `GEMINI_API_KEY` in your `.env` file is set to a placeholder.
 - UI Integration: The connection between the React frontend and FastAPI backend is working correctly.
- Next Steps For You**

To make the summarization work, please update your `.env` file with a valid Gemini API key:

 - Open `PROJECTS/CENTILLION TRAINING/webscraper/.env`
 - Replace `your_api_key_here` with your actual API key.

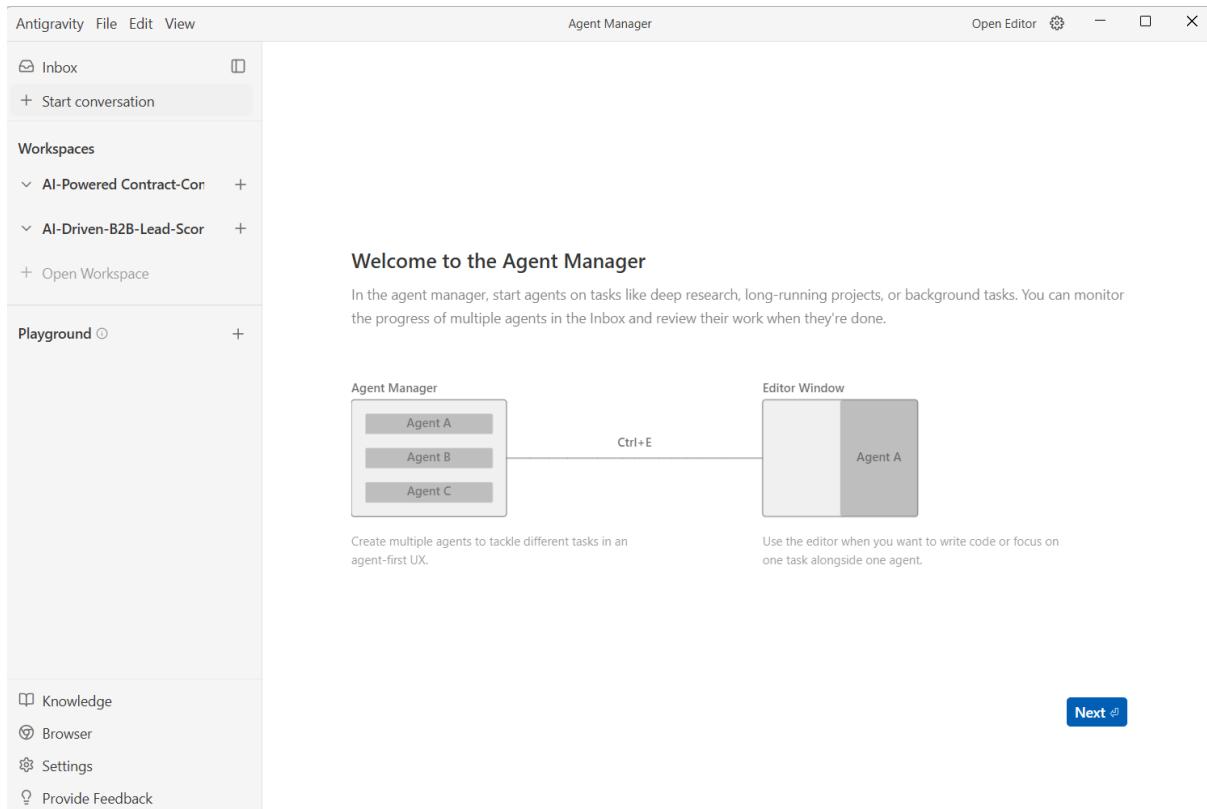
The application is currently running at `http://localhost:5173`.

In fast mode, the agent skips detailed planning and proceeds directly with execution.

- **Execution:** Tasks are implemented immediately without prior plan review
- Suitable for quick changes, simple fixes, and short-lived tasks

5. Agent Manager

The Agent Manager : It serves as a central dashboard to monitor and direct the activities of multiple AI agents simultaneously.



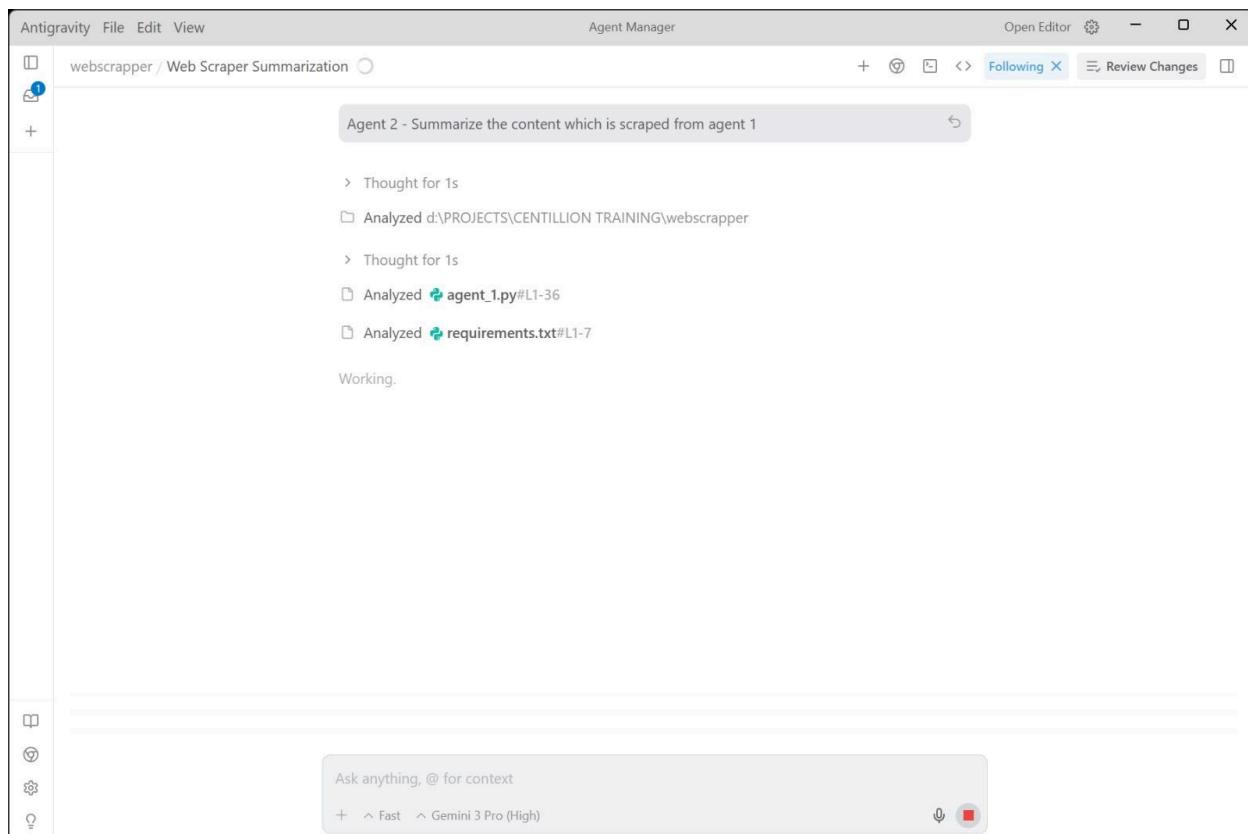
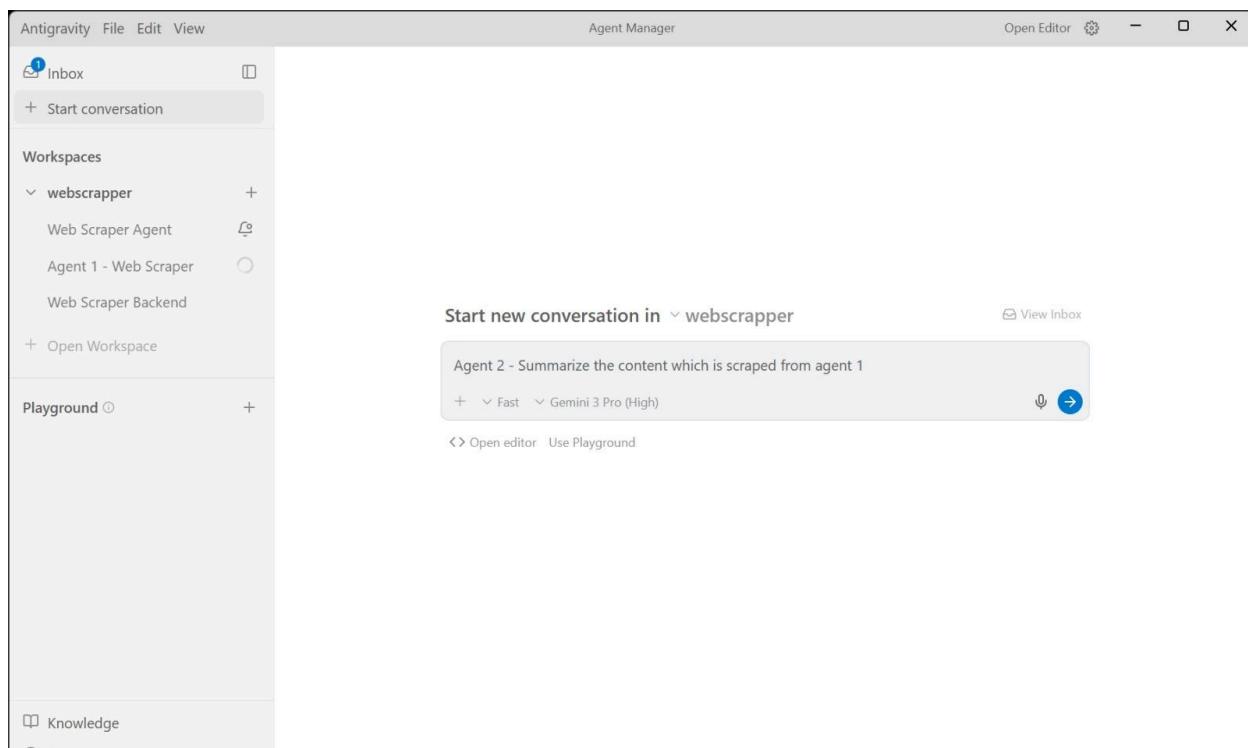
- Create and assign tasks to multiple agents in parallel
- Monitor agent progress and execution status in real time
- Review outputs and artifacts before approval
- Switch seamlessly between the Agent Manager and editor view



Agent 1 - Web Scraper

The screenshot shows the Antigravity application interface. The left sidebar contains a navigation menu with sections: 'Inbox' (with 3 notifications), 'Workspaces' (including 'webscrapper', 'Hr recruiter application', 'antigravity browser', 'hr recruiter application v0', and 'Playground'), and 'Knowledge'. The main area is titled 'Agent Manager' and contains a 'Start new conversation in' button followed by a dropdown menu for 'hr recruiter application v0 ai'. The dropdown menu includes the instruction '- Scrape contents from this url (https://arxiv.org/)', a speed setting 'Fast', and a model selection 'Gemini 3 Pro (High)'. Below the dropdown are 'Open editor' and 'Use Playground' buttons. The bottom right of the main area has a microphone icon and a blue arrow icon.

The screenshot shows the 'Webscraper / Web Scraper Agent' workspace in Antigravity. The left sidebar shows a list of files: 'webscraper / Web Scraper Agent' (selected), 'requirements.txt', 'agent_1.py', and 'index.html'. The main area displays a log of the agent's actions: 'Thought for 3s', 'Analyzed d:\PROJECTS\CENTILLION TRAINING\webscraper', 'Thought for <1s', 'Analyzed requirements.txt#L1-7', 'Thought for 6s', and 'Edited agent_1.py +36 -0'. A 'Generating.' message is visible at the bottom. The bottom right of the main area has a microphone icon and a red square icon.

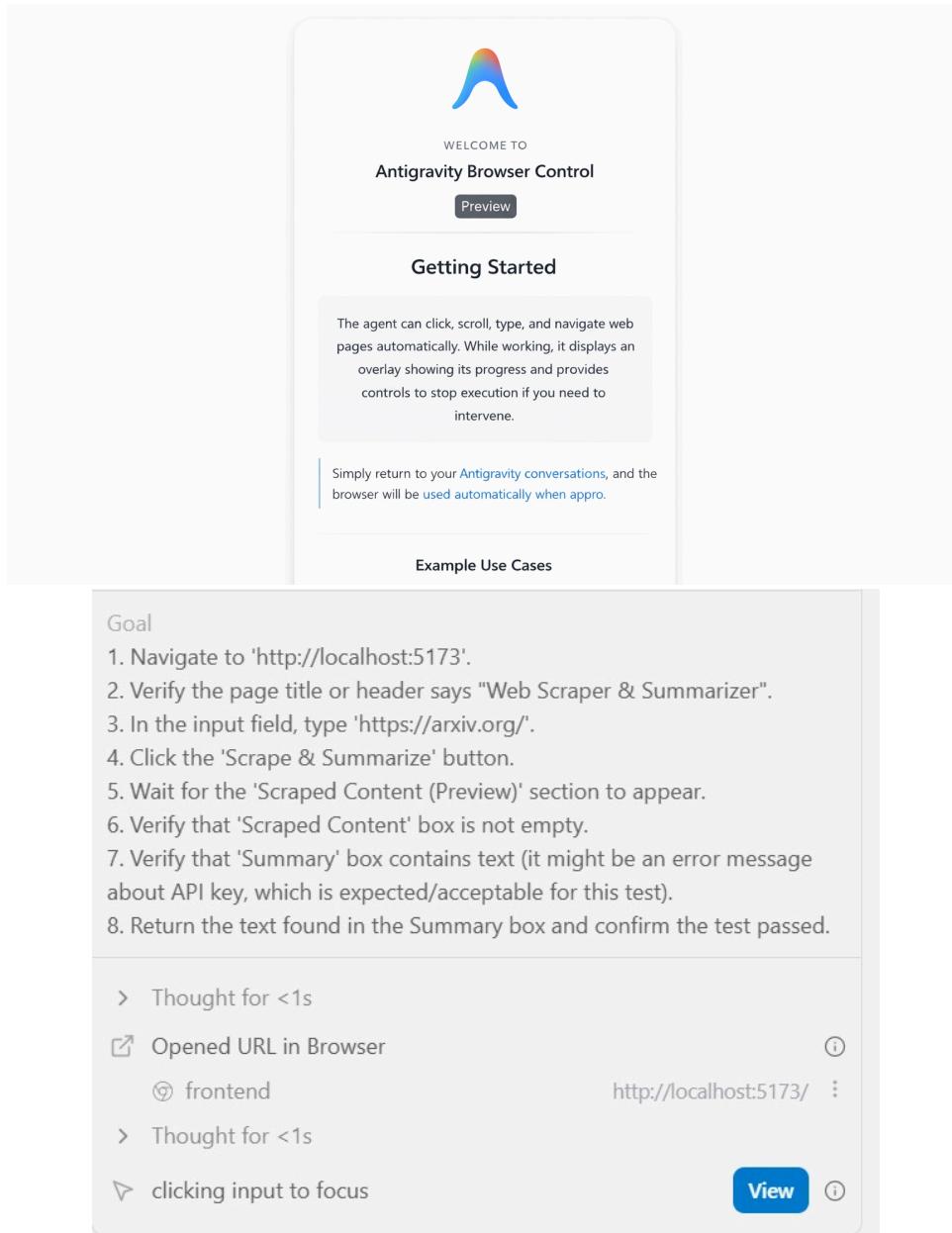


6. Browser Agent

Web-related tasks are handled by a dedicated browser subagent.

- Performs secure web interactions
- Safely reads and controls web pages

This subagent uses built-in browser tools and transparently displays its actions, allowing you to work in other tabs without disruption.



The screenshot shows the Antigravity Browser Control interface. At the top, there is a logo and the text "WELCOME TO Antigravity Browser Control" with a "Preview" button. Below this is the "Getting Started" section, which contains a description of the agent's capabilities and a note about returning to Antigravity conversations. The "Example Use Cases" section is expanded, showing a "Goal" list and a timeline of events. The goal list includes steps such as navigating to a URL, verifying the page title, and interacting with an input field. The timeline shows events like "Thought for <1s", "Opened URL in Browser" (with a URL of "http://localhost:5173/"), "Thought for <1s", and "clicking input to focus". A "View" button is located at the bottom right of the timeline.

WELCOME TO
Antigravity Browser Control
Preview

Getting Started

The agent can click, scroll, type, and navigate web pages automatically. While working, it displays an overlay showing its progress and provides controls to stop execution if you need to intervene.

Simply return to your [Antigravity conversations](#), and the browser will be [used automatically when appro](#).

Example Use Cases

Goal

1. Navigate to 'http://localhost:5173'.
2. Verify the page title or header says "Web Scraper & Summarizer".
3. In the input field, type 'https://arxiv.org/'.
4. Click the 'Scrape & Summarize' button.
5. Wait for the 'Scraped Content (Preview)' section to appear.
6. Verify that 'Scraped Content' box is not empty.
7. Verify that 'Summary' box contains text (it might be an error message about API key, which is expected/acceptable for this test).
8. Return the text found in the Summary box and confirm the test passed.

> Thought for <1s

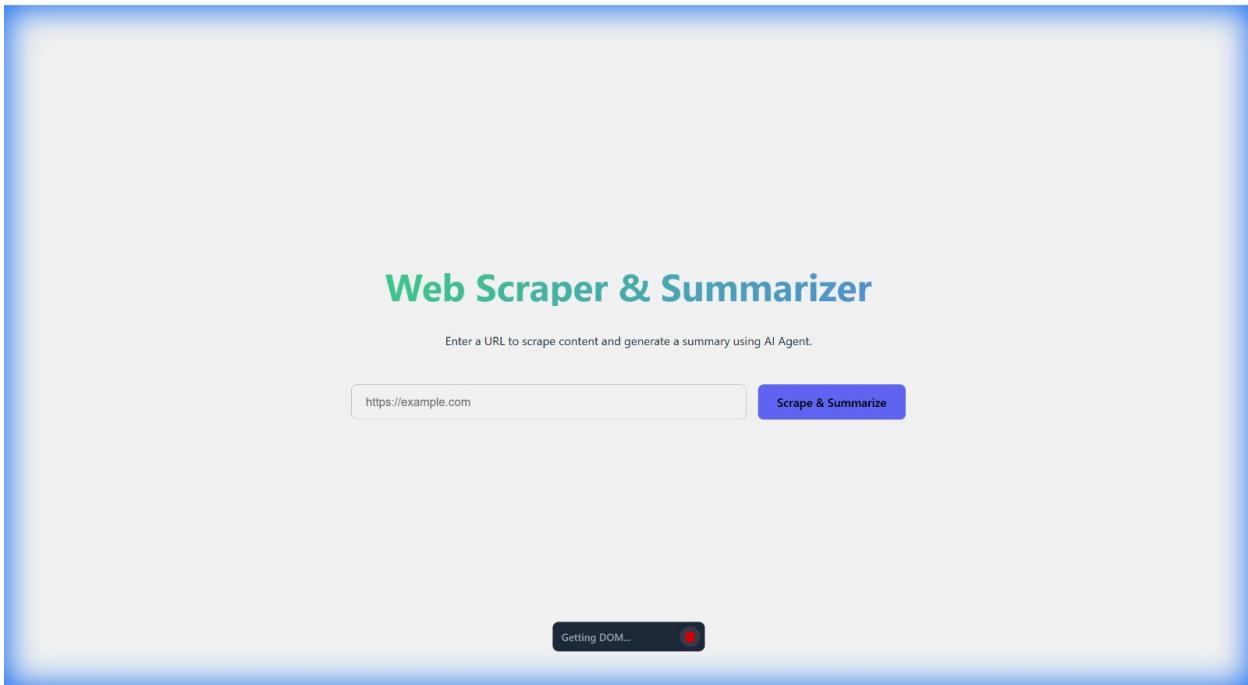
↗ Opened URL in Browser ⓘ
(frontend) http://localhost:5173/ ⋮

> Thought for <1s

▽ clicking input to focus View ⓘ



Agent interacting with the browser:



📸 **Verification:**

I verified the updated frontend build using a browser subagent. The screenshots confirm the white background theme and the functional scraping interface.

| Clean UI (White Background) | Scraping Results |
|-----------------------------|------------------|
| | |

You can find the source files in your workspace:

- frontend/src/App.jsx
- frontend/src/App.css
- main.py
- agent_1.py
- agent_2.py

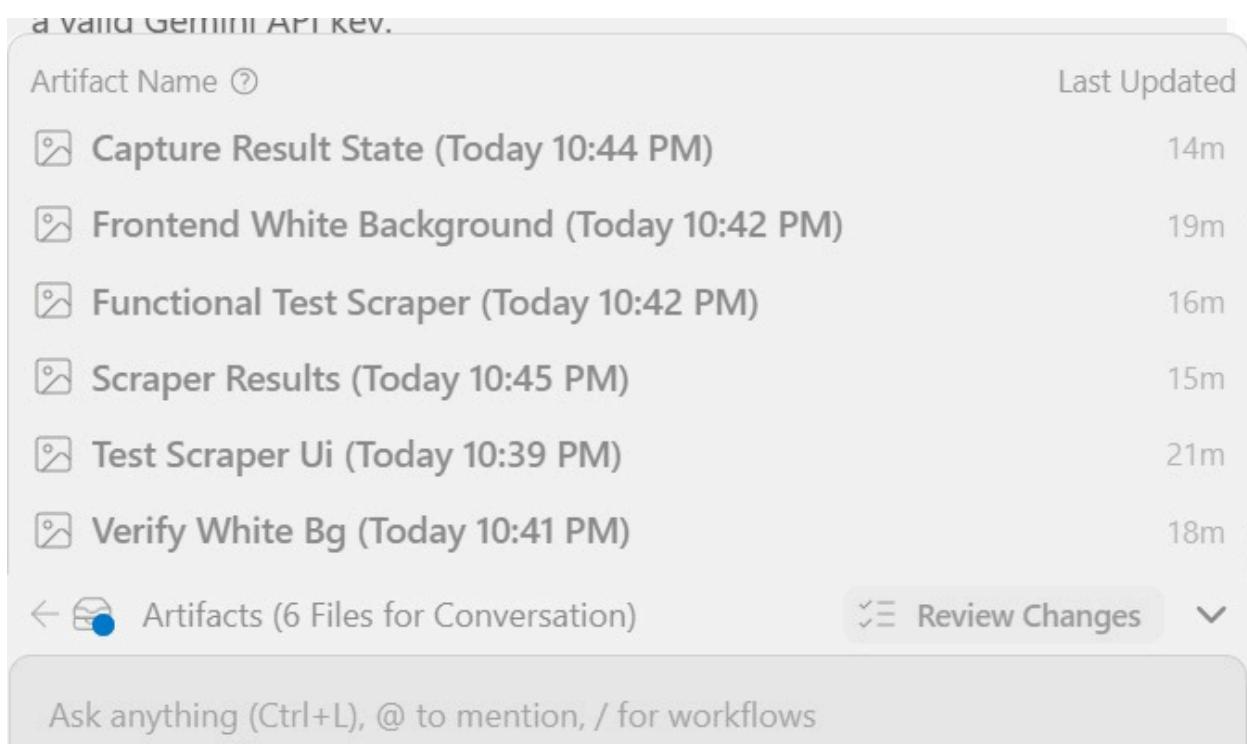
Would you like me to add more features or adjust the design further?

7. Artifacts

Artifacts are outputs such as task plans and screenshots that help the AI agent clearly show its progress and results to the developer.

- **Screenshots:** Visual evidence of actions, progress, and final results.
- **Tasks:** A clear list of actions required to complete the work
- **Implementation Plan:** An end-to-end project outline covering goals, folder structure, and relevant files

Artifacts:



| Artifact Name | Last Updated |
|--|--------------|
| Capture Result State (Today 10:44 PM) | 14m |
| Frontend White Background (Today 10:42 PM) | 19m |
| Functional Test Scraper (Today 10:42 PM) | 16m |
| Scraper Results (Today 10:45 PM) | 15m |
| Test Scraper Ui (Today 10:39 PM) | 21m |
| Verify White Bg (Today 10:41 PM) | 18m |

← Artifacts (6 Files for Conversation) Review Changes

Ask anything (Ctrl+L), @ to mention, / for workflows

8. Conclusion

Google Antigravity represents a practical shift toward agent-driven development by embedding autonomous AI agents directly into the developer workflow. By combining a familiar IDE experience with a centralized agent manager, it enables teams to delegate execution while retaining full control through review and oversight. Features such as planning and fast execution modes, browser subagents, and transparent artifacts ensure clarity, flexibility, and trust in AI-driven actions. When used with the right level of supervision, Antigravity can significantly improve productivity, reduce repetitive effort, and allow developers to focus on design, decision-making, and code quality.