

ChatGPT Operator: A Digital Proxy for Human Interaction?

[Assem Hijazi – Experience, Knowledge & Ideas](#)

Recently, OpenAI announced the release of **Operator**, a new feature integrated into ChatGPT. When I first heard about it, I thought, *Why not write about it?*—especially since the announcement didn't make things as clear as they could be.

If you've seen the announcement, the name **Operator** might give you several impressions, but it doesn't quite explain what the feature does. Is it about running operations? Helping manage systems? Or something entirely different? The truth is, it offers far more than the name suggests—and not in the way you might expect.

This led me to wonder: *Is Operator really acting on behalf of a human?* Could it be considered a **digital proxy or representative of a human**? And if so, how does it accomplish tasks like booking a reservation or filling out forms online without having integration with specific websites? The idea of Operator being an **assistant that explores websites, interacts with them, and acts as a digital extension of human senses**—like sight and interaction—raises several questions about how it works and whether it truly acts like a human.

The feature feels incredibly promising, but let's not shy away from pointing out a recurring trend: **ChatGPT's naming conventions for its features often fail to convey their true potential**. Operator might sound intriguing, but without clear explanations, it risks being misunderstood or underestimated. So, let's dive deeper to understand what Operator is, how it works, and whether it truly acts as a digital human proxy.

The List of Features

With a clearer understanding of how Operator functions, let’s dive into its core features. Below is a concise list of what the Operator can do, along with a quick description for each feature.

Note based on initial search of public data and ChatGPT.

Feature	Description
1. Autonomous Task Execution	Performs tasks like filling out forms, booking reservations, and making purchases, step by step.
2. Web Navigation	Mimics human interaction by clicking, typing, scrolling, and interacting with web interfaces.
3. Vision-Based Understanding	Analyzes website layouts visually, identifying buttons, fields, and menus without needing integration.
4. Context-Aware Actions	Adjusts its behavior based on the information you provide, making it adaptable to various workflows.
5. Approval for Sensitive Tasks	Requests user confirmation before actions like entering passwords or making payments.
6. Collaboration with Websites	Works seamlessly with partnered platforms (like OpenTable or Instacart) for enhanced task execution.
7. Task Prioritization	Completes tasks in a logical sequence, ensuring steps like form submission or payment are accurate.
8. Error Handling	Notifies users if it encounters challenges, such as unclear forms or security barriers like CAPTCHAs.
9. Consistency and Speed	Eliminates human errors and performs repetitive tasks much faster than a human could.

Limitations of ChatGPT’s Operator

While Operator introduces groundbreaking capabilities, it’s essential to recognize its current limitations. Here's a detailed list of what Operator cannot yet do or areas where it faces challenges:

Note based on initial search of public data and ChatGPT.

Limitation	Details
1. Limited Vision Accuracy	Operator simulates human sight but struggles with overly complex or unconventional website layouts, making it less reliable on non-standard interfaces.
2. No Video Understanding	It cannot watch or interpret videos, meaning it cannot follow video-based instructions or understand dynamic visual content.
3. Basic Image Recognition Only	While it can analyze simple images, it struggles with intricate or abstract visuals, such as unlabeled icons or complex diagrams.
4. Challenges with Interactive Media	Operator cannot interact with dynamic content like video players, animations, or interactive graphics (e.g., sliders, embedded games).
5. No Emotional Intelligence	Lacks the ability to interpret human emotions, context, or subjective elements, which are often crucial in decision-making or nuanced tasks.
6. Dependency on User Input	Requires clear and specific instructions from the user to complete tasks, making it less autonomous in unfamiliar scenarios.
7. Inability to Handle CAPTCHAs	Security mechanisms like CAPTCHAs block Operator entirely, requiring human intervention to proceed.
8. Limited Contextual Flexibility	Operator may fail to adapt if the task involves ambiguous workflows, vague instructions, or hidden dependencies.
9. No Built-in Integration with Websites	Without direct integrations or APIs, Operator relies solely on vision and reasoning, which can lead to errors on advanced or secured platforms.
10. Limited Multistep Workflow Understanding	While it can handle simple sequences, complex, multi-layered workflows with conditional logic may confuse it.
11. Security and Verification Barriers	Struggles with tasks involving multi-factor authentication, email verifications, or other manual security checks.
12. Static Learning for New Sites	Operator does not dynamically learn from unique website designs, meaning it may face repetitive challenges on non-partnered platforms.
13. Performance on Dynamic Websites	Websites with constantly changing layouts or content can confuse Operator, leading to incomplete or incorrect actions.

Filling Out a Registration Form on a Website

Step-by-Step Process:

1. You Specify the Task:

- You type a command into ChatGPT, such as:
"Can you fill out the registration form on [examplewebsite.com] for me?"
 - You either provide the website URL directly (e.g., *"https://examplewebsite.com/register"*) or clearly name the site.
 - If you're already on the site, you may describe it in context, but giving the link ensures Operator knows exactly where to go.

2. Operator Accesses the Website:

- Using its built-in browser capabilities, Operator navigates to the website.
- If you've provided a specific link (like the registration page), it goes directly there.

3. Operator Analyzes the Web Page:

- It scans the page visually, identifying key elements like:
 - Form fields (e.g., "Name," "Email," "Password").
 - Buttons (e.g., "Submit").
- It interprets these elements using vision and reasoning to map out the layout.

4. You Provide Information:

- Operator may ask for details it needs to complete the task. For example:
 - *"What name should I use?"*
 - *"What email address should I enter?"*
 - *"Would you like me to generate a secure password?"*
- You respond with the required details, or you might pre-provide them in your initial command.

5. Operator Fills Out the Form:

- It clicks into each field, types the information, and completes the form step by step:
 - Types your name in the "Name" field.
 - Enters your email in the "Email" field.
 - Generates and fills a password if requested.
- It ensures every field is correctly filled before proceeding.

6. Operator Submits the Form:

- Once all fields are complete, Operator clicks the "Submit" button (or equivalent).

7. Follow-Up Actions:

- If additional actions are needed (e.g., verifying an email, solving a CAPTCHA), Operator pauses and notifies you to complete those steps manually.
-

Clarifications: Does Operator Need a Link?

- **Providing a Link:** If you give a direct link (e.g., the registration page), Operator can navigate there directly, saving time.
 - **Naming the Website:** If you provide just the website name, Operator will attempt to locate the relevant page. However, this may take longer or require follow-up instructions to specify the exact location.
 - **Being Already on the Website:** If you're describing a site you're currently using (without providing a link), Operator may ask for more context to ensure it finds the correct page.
-

Why This Matters:

Being clear about how you provide instructions helps set the stage for seamless interactions. While Operator is smart enough to infer some details, direct links or clear task descriptions greatly improve its accuracy and efficiency.