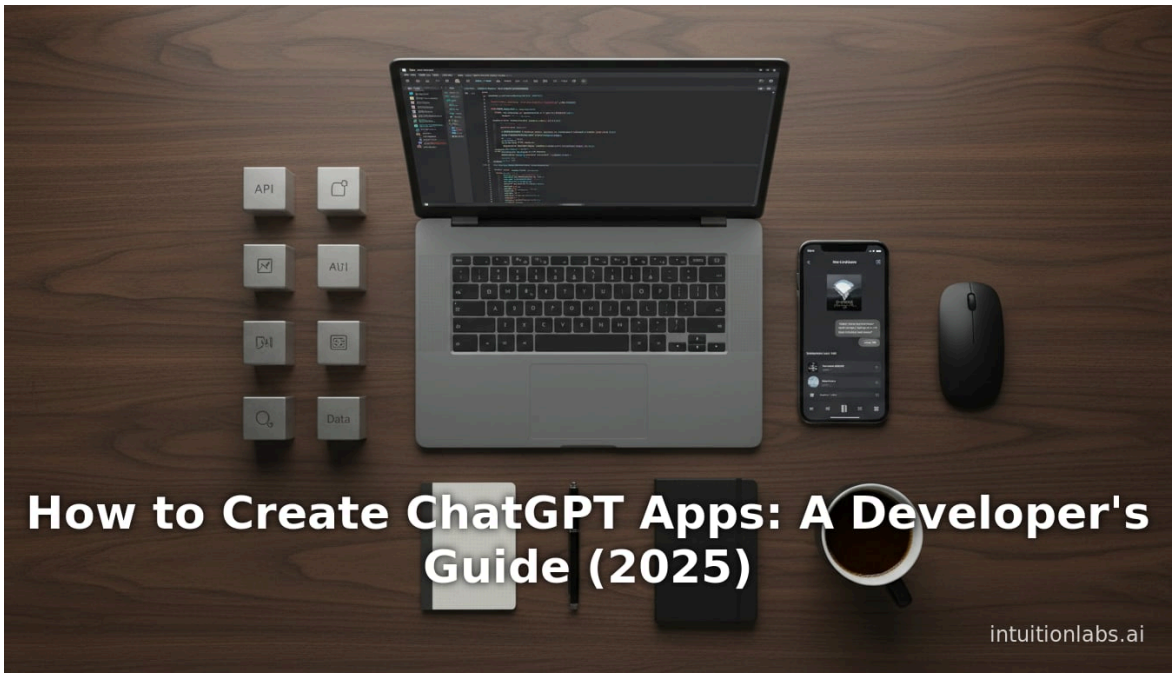
How to Create ChatGPT Apps: A Developer's Guide (2025)

By InuitionLabs.ai • 10/1/2025 • 35 min read

chatgpt apps | openai apps sdk | custom gpts | gpt store | app development | conversational ai | openai | model context protocol



Executive Summary

The ChatGPT ecosystem has rapidly evolved from a simple AI conversational interface into a sophisticated platform supporting integrated applications. Initially launched in late 2022, ChatGPT has progressively expanded its capabilities – first by adding web access and computational **plugins** in early 2023, then by introducing user-created *"custom GPTs"* (akin to smartphone apps) and a **GPT Store** for distributing specialized chatbots (www.axios.com) (www.axios.com). By 2025, OpenAI rolled out an ** Apps SDK** and in-chat *"apps"* that allow third-party services (e.g. Spotify, Booking.com, Zillow) to be invoked directly within a conversation (www.techradar.com) (www.techradar.com). This report provides a comprehensive analysis of how developers and organizations can create these ChatGPT-integrated apps. We review the historical context (from basic chatbots to the current platform model), outline the technical framework (including the open * Model Context Protocol* used by the Apps SDK (help.openai.com)), and detail the step-by-step development process (from concept and design guidelines to implementation and testing). We also examine multiple perspectives – developer workflows, business and market implications, user experiences, and ethical considerations (privacy, safety, and trust). Real-world examples illustrate current use cases and benefits: for instance, a user can ask ChatGPT to **"make me a new Spotify playlist"** or **"create a Zillow real estate listing"**, and ChatGPT will seamlessly call those services without leaving the chat interface (www.techradar.com). Data and industry analyses are cited to show the explosive adoption of ChatGPT technologies (e.g. hundreds of millions of users (www.index.dev) (masterofcode.com)) and the analogies to smartphone app ecosystems (www.axios.com) (www.axios.com). We conclude by discussing the future trajectory: the continued growth of a ChatGPT "app store," emerging agent capabilities, and the potential for conversational AI to transform how software is built and used.

Introduction and Background

** ChatGPT (Chat-based Generative Pre-trained Transformer)** was introduced by OpenAI in November 2022 as a large-language-model-powered chatbot. Its ability to generate fluent natural language answers spurred rapid user adoption. By early 2023, ChatGPT became the world's fastest-growing consumer application, reaching over 100 million monthly active users (www.axios.com) (masterofcode.com). Researchers and enterprises quickly recognized that conversational AI could do more than chat; it could assist with tasks across domains (from coding and writing to planning and analysis). This led OpenAI to expand ChatGPT from a static conversational model into a **platform** for interactive tools. In March 2023, OpenAI released an initial *plugin* system allowing ChatGPT to access updated web information, perform computations, and call third-party services (www.chat-gpt.ai) (openai.com). Over the next two years, OpenAI iteratively built out this platform model. In late 2023, at its first Developer Day,

OpenAI unveiled **customizable ChatGPT "GPTs"** – specialized versions of ChatGPT with tailored data and behaviors – likening them to AI counterparts of smartphone apps (www.axios.com) (time.com). In January 2024, OpenAI launched a **GPT Store**, effectively a conversational app directory, where developers had already published millions of custom chatbots (reportedly 3 million at launch (www.axios.com)).

In mid-2025, OpenAI introduced a formal **Apps SDK** and *in-chat apps*: a system where developers can build hosted services that run "inside" ChatGPT conversations (help.openai.com) (www.techradar.com). These in-chat apps allow users to perform tasks (like booking tickets, making presentations, querying calendars) simply by asking ChatGPT – the chatbot automatically invokes the appropriate background service. In effect, ChatGPT is becoming a "**conversational app store**," embedding useful tools into its interface (www.techradar.com) (www.axios.com). This report systematically explores *how* these ChatGPT apps are created: from planning and design to development and deployment. We also analyze market data, user surveys, and expert commentary to understand the current state and future implications of this emerging technology.

Evolution of ChatGPT into an App Platform

ChatGPT's path from a standalone chatbot to a full-fledged platform has unfolded rapidly. Within weeks of ChatGPT's 2022 launch, users and developers began envisioning extensions of its capabilities. In early 2023, OpenAI introduced **plugins** – small add-ons enabling ChatGPT to retrieve live web data, perform computations, or interact with external APIs (www.chat-gpt.ai) (openai.com). The official announcement credited plugins as a "major milestone" that turned ChatGPT into an ecosystem for third-party functionality (www.chat-gpt.ai). The first partners included travel and shopping services (Expedia, Kayak, Instacart, etc.) and productivity tools (Slack, WolframAlpha, Zapier) (www.chat-gpt.ai) (openai.com). OpenAI simultaneously provided *developer documentation* so others could build plugins, gradually rolling out access through a waitlist.

The plugin system set the stage, but OpenAI envisioned a broader expansion. At Developer Day 2023, they announced **"GPTs"** – chatbots customized by developers for specific tasks or domains. Axios reported that OpenAI described GPTs as "the AI counterpart to smartphone apps," promising a marketplace where users could try or even buy these custom chatbots (www.axios.com).Indeed, early previews showed how companies could create a dedicated GPT for, say, personal finance advice, legal questions, or any niche. A **GPT Store** was promised to launch soon. This store debuted in January 2024, and OpenAI noted that **~3 million** custom GPTs had already been created by developers and hobbyists (www.axios.com). The store categorized these GPTs (e.g. writing, programming, lifestyle) and made them available to ChatGPT paid subscribers. Initially, OpenAI did not charge developers for listing GPTs; monetization and revenue-sharing models were still being worked out in early 2024 (www.axios.com) (www.theverge.com).

These features – plugins and GPTs – made ChatGPT more versatile, but users still typically had to search for or specify which GPT/plugin to use. In late 2025, OpenAI took a step further by embedding **third-party apps into the chat itself**. According to TechRadar, ChatGPT's latest update turned the chatbot into "a conversational app store" by integrating well-known services directly into conversations (www.techradar.com) (www.techradar.com). So instead of the user explicitly opening a Spotify or Booking app, ChatGPT will proactively call these services when relevant. For example, a user might simply say **"plan a trip to Paris"**, and ChatGPT would not only generate text but also interface with Expedia or Booking.com to suggest flights and hotels (www.techradar.com) (www.techradar.com). The user never leaves the chat – the app's UI (cards, carousels, forms) appears within ChatGPT's interface. This paradigm parallels smartphones: instead of launching separate apps, here the **apps come to the user through conversation** (www.techradar.com) (www.axios.com). In practice, TechRadar notes that ChatGPT can now, for instance, create a Spotify playlist or generate a Zillow housing listing on demand without the user navigating away (www.techradar.com).

These innovations have made ChatGPT a central hub for many tasks, from ride-booking to email management. As one analysis observes, ChatGPT is positioning itself as **"the command layer for all your tools,"** eliminating the need to switch contexts between apps (dev.to) (scalevise.com). This transformation – from a single-purpose chatbot to a unified interface for multiple services – is the backdrop against which developers learn to create ChatGPT apps. Below we examine the developer tools and procedures that enable this integration.

Year	Milestone in ChatGPT Platform Evolution
2022	**ChatGPT Launch** – OpenAI releases ChatGPT as a conversational AI model (Nov 2022). It rapidly amasses millions of users through its ability to generate human-like text.
2023	**Plugin Architecture Added** – In March 2023, OpenAI enables ChatGPT plugins for external data and computations (openai.com). Initial partners (Expedia, Slack, Wolfram, etc.) launch plugins (www.chat-gpt.ai) (openai.com).
2023	**Developer Day & Custom GPTs** – November 2023 Dev Day: OpenAI announces *customizable ChatGPT GPTs* (user-created chatbots), likening them to smartphone apps (www.axios.com); first teaser of GPT Store.
2024	**GPT Store Launch** – January 2024: OpenAI opens the GPT Store. Millions (3M+) of GPTs created; directory of chatbots sharable to paid users (www.axios.com). OpenAI begins testing revenue-sharing models for GPT creators (www.theverge.com).
2024	**ChatGPT Business & Team Plans** – New subscription tiers introduced (e.g. Business plan) (www.axios.com), acknowledging enterprise and commercial use.
2025	**Gmail/Calendar Integration** – Mid-2025: ChatGPT adds connectors to Gmail, Google Calendar, and others for Plus/Pro users (www.techradar.com) (www.techradar.com), enabling in-chat handling of emails and schedule.
2025	**Apps SDK (Preview)** – Mid-2025: OpenAI releases the *Apps SDK* in preview. This uses the open *Model Context Protocol (MCP)*, allowing developers to build fully integrated apps that run inside ChatGPT conversations (help.openai.com) (www.axios.com).
2025	**In-Chat Apps Rollout** – Late 2025: ChatGPT formally "gets apps": integration partners (Spotify, Canva, Zillow, etc.) now surface within chats. ChatGPT proactively suggests and calls relevant apps for user requests (www.techradar.com) (www.techradar.com).

Developer Ecosystem for ChatGPT Apps

Developers looking to create apps for ChatGPT must navigate a specific tooling stack and set of guidelines provided by OpenAI. Conceptually, a ChatGPT "app" consists of two parts: (1) a **backend service** (called an *MCP server*) that performs the app's logic; and (2) a **conversational interface** that defines how the app's functionality is invoked within a chat. OpenAI's *Apps SDK* (available in preview mode as of 2025) lays out a standardized framework for this. Key elements include:

- **Model Context Protocol (MCP)**: The Apps SDK is built atop the Model Context Protocol, an open standard (initially developed by Anthropic) for AI programs to connect with external tools and structured data (help.openai.com) (www.axios.com). MCP defines how the AI (ChatGPT) and the developer's server communicate: tools, inputs, outputs, and UI components. In practice, a developer writes an MCP-compliant server that exposes *"tools"* (endpoints) ChatGPT can call. The server returns structured data (JSON) and optional HTML UI snippets. The ChatGPT client then embeds these responses seamlessly in the conversation.

- **App Logic and Tools**: Developers use any technology to implement the MCP server (OpenAI provides official SDKs for common languages). They define *tools* – essentially JSON schemas for specific functions. For example, a restaurant-finding app might expose a tool named `find_restaurants` with parameters like location and cuisine. ChatGPT, given user input, decides *which tool to call* based on the context. The tool's response (e.g. a list of restaurants) is sent back to ChatGPT and rendered in the chat via UI components (cards, lists, etc.).

- **Conversational UI**: Besides backend tools, developers design the frontend presentation. The Apps SDK allows embedding lightweight **cards, carousels, and forms** in the chat window that feel native to ChatGPT's interface (developers.openai.com). For example, searching for flights might display an embedded list of options with book buttons directly in the conversation. OpenAI's documentation emphasizes that these UI elements must "integrate seamlessly…Ito maintain [ChatGPT's] clarity, trust, and voice" (developers.openai.com).

- **Developer Tools and Testing**: OpenAI provides a **Developer Mode** in the ChatGPT web/mobile app, where creators can load and test unreleased apps. In this mode, a developer can initiate chats with their app code running locally or on ngrok. They iterate until the experience is polished. When ready, apps must be submitted to OpenAI (with metadata and compliance checks) before appearing in the public directory.

- **Design and Safety Guidelines**: OpenAI has published comprehensive design and policy guidelines. Apps must provide **clear value** – they should make ChatGPT "substantially better at a specific task" or unlock a new capability (developers.openai.com). Use cases should be conversational, time-bound tasks (e.g. booking a ride, ordering food, checking a delivery status) that fit naturally in dialogue (developers.openai.com). Conversely, tasks that break the flow (like pasting large website blocks or irrelevant advertising) are discouraged (developers.openai.com). Privacy and predictability are paramount: apps should ask only for data they truly need, behave as promised with no hidden behavior, and comply with OpenAI's content and safety policies (developers.openai.com).

- **Monetization Support**: Although the Apps SDK was previewed without immediate monetization, OpenAI signaled plans for revenue sharing. The developer guidelines mention future support for an *Agentic Commerce Protocol* (enabling in-chat transactions) (help.openai.com). Similarly, the earlier GPT Store promised payment to creators, and by early 2024 OpenAI was testing revenue-sharing models (www.theverge.com). We expect that ChatGPT apps will eventually offer developers ways to charge users (via subscriptions, one-time payments, or transaction fees) once the directory opens to the public.

Collectively, this ecosystem mirrors app development on mobile platforms, but with key differences: the *interface* is a conversation, tools are called by the AI, and discovery happens via chat. The following sections detail how a developer actually builds such an app.

How to Create a ChatGPT App: Process Overview

Building an app for ChatGPT involves several stages: planning the concept, implementing the server, designing the chat interface, testing, and deployment. Below is a high-level step-by-step outline (also summarized in Table 2) drawing from OpenAI's official flow (help.openai.com) (developers.openai.com).

1. **Idea and Planning**: Identify a clear use case where ChatGPT can provide value. Good candidates are tasks that naturally fit a conversation and can benefit from structured outputs (e.g. a to-do list manager, trip planner, or document summarizer). Refer to OpenAI's design guidelines to ensure the task is appropriate for in-chat handling (developers.openai.com). Consider what data the user will provide and what the app needs (e.g. location, authentication). Sketch the conversation flow: how a user prompt leads to tool calls and results.

2. **Set Up an MCP Server**: Implement the backend using the Model Context Protocol specifications. This involves creating a web server (in Python, Node.js, etc.) that exposes one or more HTTP endpoints ("tools"). Each tool has:

- A **machine-readable name** (used internally) and a **human-friendly title**.
- A **JSON schema** for input parameters.
- An **implementation** that performs the action (e.g. queries a database, calls an API).
- A **UI component template** (HTML/CSS) for displaying results when needed.

The Apps SDK provides official libraries and code samples to speed this up. Tool responses should be structured – for example, returning items in a JSON list, which ChatGPT can render in a list card.

3. **Develop Conversation Logic**: Decide what the user says and how ChatGPT uses tools. While ChatGPT's LLM handles understanding the user's prompt, you may guide it by providing tool descriptions or example prompts. OpenAI's **developer tools** allow specifying how ChatGPT should advertise available tools to itself in the chat context. For instance, you might include a description like *"Find restaurants near [location] versed in [cuisine]"*. This helps ChatGPT know when to call your `find_restaurants` tool.

4. **Design the UI Components**: Alongside backend code, create the HTML/CSS components for any interactive results. For example, if your tool returns a list of locations, prepare an HTML snippet that displays cards or a table. Follow OpenAI's UI guidelines: use the system font, keep buttons simple, and maintain consistent styling (developers.openai.com) (developers.openai.com). Remember: all app interactions happen *inside* ChatGPT's chat window. The interface should look and feel like a native part of the conversation (cards, forms, etc.), without excessive complexity or unrelated navigation.

5. **Test Locally in Developer Mode**: In the ChatGPT web or desktop app, enable *Developer Mode*. This feature lets you "load" your local app. Initiate a chat with a system message that includes your app's configuration (usually a URL to the MCP server). Then simulate user queries and verify that ChatGPT calls your tools correctly and that the responses render properly. Iterate on prompts, JSON schemas, and UI until the experience is smooth and intuitive. During testing, focus on edge cases and error handling (e.g. what happens if a search returns nothing).

6. **Compliance and Guidelines Review**: Ensure your app adheres to OpenAI's policies. According to the developer guidelines, the app must be safe for all audiences, respect user privacy, and avoid disallowed content (developers.openai.com). Limit data collection to only what is necessary, and clearly state how any user data will be used. Double-check that the app's behavior is predictable (no hidden actions) and that it truly provides user value. If your app is likely to handle sensitive topics, implement safeguards or disclaimers as required by OpenAI's broader usage policies (e.g. hate speech, medical/financial advice).

7. **Deployment and Submission**: Once refined, deploy your MCP server to a public, secure (HTTPS) endpoint. You may host on any cloud or hosting provider. Prepare metadata for your app: a name, description, icon, category(s), and any usage instructions. Subscribe to OpenAI's upcoming app directory (currently in closed beta in late 2025). When submissions open, submit your app for review. OpenAI may test your app for performance, correctness, and compliance before approving it. After acceptance, your app will be listed in ChatGPT's app directory, making it discoverable by users. Meet publication standards if you want to be featured (good design, higher usage, etc.).

{% raw %}

Step	Description and References
1. **Ideate & Plan**	Define clear user tasks that fit conversation (time-bound, valuable) (developers.openai.com). Research if similar ChatGPT apps exist.
2. **Set Up MCP Server**	Implement backend "tools" using Model Context Protocol. Official SDKs (Python/TypeScript) can help. Define each tool's API schema and UI component templates (help.openai.com) (developers.openai.com).
3. **Implement Logic and UI**	Write code for tool functions and build HTML/CSS for result cards. Use cards/carousels per design guidelines (developers.openai.com) (developers.openai.com).

Step	Description and References
4. **Define Dialogue Schema**	Provide ChatGPT with tool descriptions/prompts so it knows when to invoke your tools. Ensure conversation flows naturally.
5. **Test in ChatGPT**	Use ChatGPT's Developer Mode to chat with your app. Iterate prompts, outputs, and UI. Ensure smooth behavior on valid, edge, and error cases (help.openai.com).
6. **Apply Policies**	Review OpenAI's developer guidelines: ensure privacy-respecting inputs, safe outputs, and no prohibited content (developers.openai.com). Comply with all usage policies and style requirements.
7. **Deploy & Submit**	Host your server on HTTPS and prepare app metadata. Submit to OpenAI's ChatGPT app directory when submissions open (help.openai.com). Upon approval, monitor feedback and usage.

{% endraw %}

Technical Architecture

Understanding the underlying architecture is crucial. The **Model Context Protocol (MCP)** is the cornerstone of ChatGPT apps. Described by Axios and OpenAI docs, MCP is an *open* protocol that connects an AI model to external tools and data (www.axios.com) (help.openai.com). Unlike traditional applications that rely on human users to click buttons, MCP lets an AI agent call endpoints directly. In practice, the developer's server acts as an MCP server:

- **Connection**: When a user interacts with the app, ChatGPT sends an HTTP request to the developer's MCP server with a structured JSON payload. This payload includes parameters like user message, tool name, and conversation context.
- **Tools & Operations**: The server defines multiple *operations* (tools) – each with a JSON schema and handler function. For example, an app might have a `searchHotels` tool. When ChatGPT decides to use it, it issues a request with the required fields. The server runs the query (e.g. calls a hotel API) and returns structured data.
- **Response Rendering**: The MCP protocol allows a tool to return not just raw data, but also *"component"* HTML for ChatGPT to render. OpenAI's docs explain that each tool result can include a `component_html` snippet which ChatGPT will display inline (developers.openai.com). This enables rich outputs: e.g. a cards list of flights, a map widget, or a formatted table.
- **State and Media**: MCP supports asynchronous operations and can stream updates. The client (ChatGPT) and server also negotiate content security policies for the embedded components (developers.openai.com). Servers handle authentication if needed (for user-specific data) by linking ChatGPT user's identity to the app's back-end, though this is advanced usage.

Because MCP is open, apps built with it are portable across platforms that adopt the standard (as noted by Axios (www.axios.com)). The Apps SDK extends MCP with convenient abstractions specifically for ChatGPT's UI and developer experience (including official tool libraries and templates). For example, the SDK provides a way to define tool schemas in code, and even to serve the necessary HTML/CSS without managing the raw HTTP details.

The **Agentic Commerce Protocol**, also referenced in OpenAI's docs, is a related standard intended for enabling e-commerce transactions inside ChatGPT (help.openai.com). It is designed to allow instant checkout flows within chat apps, suggesting ChatGPT apps could one day complete purchases without redirecting users. This underscores that ChatGPT aims to be a **platform for action**, not just conversation – users can not only ask for information but can also pay for goods and services through apps.

Design Guidelines and Best Practices

OpenAI emphasizes design consistency and user trust. According to the official design guidelines, a ChatGPT app should *"feel native"* to the chat environment (developers.openai.com). This means avoiding large context switches or confusing layouts. Specific best practices include:

- **Conversational Task Focus**: Apps should handle tasks that fit a chat. Booking a ride, ordering food, or checking package status are cited as ideal "low friction, time-bound" tasks (developers.openai.com). The user experience remains a dialogue, augmented by UI elements (cards or buttons) for clarity.
- **Clarity and Value**: Every action your app performs should be clearly related to the user's request. OpenAI notes a good app "does something clearly valuable" for the user (developers.openai.com). For example, if a user asks to **"find nearby coffee shops"**, the app should not present irrelevant data. Include helpful labels and avoid technical jargon.
- **Privacy and Data Minimization**: Collect only the information you need. For instance, if an app suggests recipes based on ingredients, it should not ask for unrelated personal data. OpenAI's guidelines explicitly state *"Inputs are limited to what's truly needed"* and users must remain in control of shared data (developers.openai.com). Use standard permission dialogs (as enabled in the ChatGPT settings) when needing private account access (e.g., Google Calendar).
- **Stability and Predictability**: Apps must behave as advertised. The developer documentation warns: "Apps do exactly what they say they'll do — no surprises, no hidden behavior" (developers.openai.com). Avoid unpredictable outputs or side-effects. For example, if your app has a "buy tickets" tool, it should not erroneously purchase the wrong event.
- **Broad Audience Safety**: Since ChatGPT is used by diverse users, apps must comply with OpenAI's community standards (developers.openai.com). Content must be appropriate for all ages unless clearly labeled. Implement content checks if needed (e.g., filter out hate speech, disallowed content). The guidelines emphasize compliance with usage policies and safe handling of "unsafe requests".

- **Interface Design**: Use UI components sparingly and informatively. Cards should have concise titles, images/icons, and action buttons (e.g. "Book Now" or "Play"). Collections (side-by-side cards) let users choose from multiple options (developers.openai.com). Avoid clutter or deep nested navigation; if the user needs to drill down further, you may present a new card or tool request rather than complicating one card (developers.openai.com). Always maintain consistent styling (margins, padding, fonts) as per OpenAI's system defaults (developers.openai.com).

By following these guidelines, developers can ensure their ChatGPT apps feel seamless. Good adherence can also earn an app featured placement in the directory. Conversely, violating these principles may lead to rejection or removal by OpenAI during app review.

Case Studies and Real-World Examples

To illustrate the power and current use of ChatGPT apps, consider some early examples and experiments:

- **Productivity Integrations**: ChatGPT's connectivity with tools like Gmail, Google Calendar, Slack, and Notion (via connectors introduced in 2025) shows simple personal assistant usage (www.techradar.com) (www.techradar.com). For example, after linking a Gmail account (with user consent), a ChatGPT user can **"find my latest email from John"** or **"summarize this email"** without leaving the chat. Likewise, querying **"what's on my calendar today?"** triggers the Calendar app within ChatGPT (www.techradar.com). These are delivered through natural language; if user enquiries demand, ChatGPT calls the connected apps to fetch data, processes it, and responds within chat. TechRadar notes that these capabilities "transform ChatGPT into a more efficient personal assistant, capable of handling both communication and scheduling tasks seamlessly" (www.techradar.com).

- **Travel Planning**: The integration of apps like Expedia and Booking.com means ChatGPT can orchestrate travel bookings. In TechRadar's coverage, asking about planning a trip causes ChatGPT to proactively suggest the Expedia or Booking.com app (www.techradar.com). A use-case might be: *User says, "Book me a flight to New York next Friday under $300."* ChatGPT understands this as a travel search intent and could invoke the Expedia tool, returning flight options directly in the chat. Early plugin partners (e.g. KAYAK, Booking) demonstrated simplified trip planning tasks, and the new apps take this further by embedding the booking workflow inline.

- **Creative Tasks**: Canva's app within ChatGPT allows design tasks. For instance, a user asking **"design a birthday card"** prompts ChatGPT to call Canva's service. TechRadar mentions Canva as one of the supported apps (www.techradar.com). Although detailed UI flows for design are complex, the basic idea is that ChatGPT asks the user for details (text, style) and Canva generates the graphic. Another example: **"build me a presentation on renewable energy"** could trigger ChatGPT to use a presentation-building app (like the "slides building" function, possibly via Figma or Google Slides integration) and display draft slides.

- **Entertainment and Media**: Spotify and gaming integrations have also been shown. TechRadar gives the example of asking for a playlist – ChatGPT uses Spotify's app to curate songs (www.techradar.com). Similarly, integration with YouTube or other media services would allow ChatGPT to search videos or play songs on command. Although not all such integrations are live, prototype demonstrations (like SummaLingual's music plugin) indicate the concept.

- **Custom Agents**: ChatGPT's agent mode (released mid-2025) can automate multi-step tasks. For example, **TechRadar** reports that the macOS ChatGPT app's "Agent" can autonomously plan events and create slide decks by itself (www.techradar.com). It uses the connected apps on the computer to carry out tasks. This is effectively a specialized ChatGPT app (an *agent* that runs tools serially). Users triggered this by typing commands or enabling Agent mode. It exemplifies how a ChatGPT app can do more than one-shot calls – it can execute branching logic across apps.

- **GPT-as-Developer**: A somewhat different case is using ChatGPT (or GPT-5) *to build apps directly*. A Tom's Guide experiment in 2025 had a writer generate five complete apps solely by prompting GPT-5 (www.tomsguide.com). By describing each app's idea to GPT-5, the model produced working prototypes (complete with interfaces) in under 30 minutes, without any coding from the user. The apps included an AI journal, recipe generator, story writer, goal planner, and meme creator. GPT-5 even guided the author through steps to publish them on app stores (www.tomsguide.com). This "AI programming" approach – sometimes called *"vibe coding"* – highlights a different interpretation of ChatGPT apps: here, the chatbot assists human developers by generating code. While not the same as a deployed ChatGPT app, it shows how LLMs can lower the barrier to creating software. One expert observed that this trend makes "natural language [the] new programming language," fundamentally democratizing app development (www.tomsguide.com).

These examples underscore multiple facets: ChatGPT apps can directly handle user tasks by uniting services, and conversely, ChatGPT can help build new software. They illustrate the broad applicability of ChatGPT as both a user-facing "app platform" and as a development aid.

Broader Perspectives

Market Adoption and Impact

ChatGPT's growth has been meteoric. Industry surveys report that hundreds of millions of users engage with ChatGPT-like services. Data compiled in 2024-2025 suggests ChatGPT is the world's largest AI app; one analysis found **400 million monthly users** globally (www.index.dev). (For comparison, that rivals the user base of major social media apps.) Usage is not limited to tech enthusiasts: ChatGPT sees adoption across organizations. For example, by mid-2025 it was estimated that **80% of Fortune 500 companies** had experimented with ChatGPT in under a year of its release (masterofcode.com). Large companies use ChatGPT for customer support automation, document analysis, and more. Among professionals, ChatGPT penetration is high: about **63–65%** of software developers, marketers, and journalists report using it in their work (masterofcode.com).

These figures illustrate a massive potential audience for ChatGPT apps. If even a small fraction of users explore third-party apps, it translates to millions of app interactions. OpenAI's GPT Store saw rapid content creation: by Jan 2024, devs had already built *3 million* GPT chatbots (www.axios.com). Although many were personal experiments, the volume underscores developer interest. At its 2025 re-launch event, OpenAI CEO Sam Altman said ChatGPT would start suggesting relevant apps to users by name, making discovery easier (www.axios.com). Analysts liken this to forming an ecosystem like Apple's App Store – only the "apps" are AI-enhanced conversational tools (www.axios.com) (www.techradar.com).

On the developer side, the new Apps SDK aims to increase participation. In an announcement, OpenAI highlighted plans to make joining this ecosystem simpler: developers can now activate GPTs by name in chat, and ChatGPT will even suggest apps automatically when users ask for tasks (www.axios.com). To encourage quality and safety, OpenAI provides detailed developer guidelines. They also plan to offer monetization: Axios reported that OpenAI is testing revenue-sharing models for GPT and app creators (www.theverge.com). A portion of chat commerce may flow through the *Agentic Commerce Protocol* in the future.

If successful, ChatGPT apps could spawn a new industry of AI-powered services. Companies might create specialized GPTs or ChatGPT tools to differentiate their offerings. For example, a financial firm could release a GPT for market analysis, or an e-commerce site could provide a direct shopping assistant in chat. Already, major online platforms (Spotify, Figma, Zillow, Canva, Coursera, etc.) have shown interest in integrating with ChatGPT (www.techradar.com) (www.techradar.com). The synergy is attractive: platforms gain conversational interfaces, and ChatGPT gains access to rich data and functionality.

Economic and Developer Opportunities

For developers and entrepreneurs, the ChatGPT apps environment represents a novel opportunity. Building on the TechCrunch analysis of mobile apps, one can speculate an analogous model: independent developers or companies can publish chat apps, potentially monetize them through usage fees, subscriptions, or transactions. OpenAI's references to app directories and revenue shares indicate an intent to stimulate a creator economy.

However, the model is still emerging. Unlike smartphone apps, ChatGPT apps do not require separate installation by users; they are invoked on demand. This lowers the friction for developers reaching users, but also means developers rely on OpenAI's platform policies and distribution. Successfully marketing a ChatGPT app means optimizing it for visibility within ChatGPT's search or suggestions, and ensuring high ratings. Developers will need to consider how to stand out; OpenAI hints that apps meeting "higher design and functionality standards" might be featured more prominently (help.openai.com). Earning revenue may initially come from ChatGPT Plus or Team subscriptions, and in-chat commerce, until full third-party billing is enabled.

Meanwhile, organizations see cost benefits. Studies suggest companies save significantly by automating tasks with AI. For instance, as of 2025 one report claimed that **25% of US companies** saved between $50–70K using GPT-based tools, with some saving over $100K (masterofcode.com). Productivity gains from AI are estimated in the tens of percent. For enterprises, building private ChatGPT apps (via business plans) allows applying these tools without exposing proprietary data, opening up scenarios like custom knowledge bases and internal workflow automation.

Security, Privacy, and Ethics

Building apps in ChatGPT raises important non-technical issues. Because these apps can access user data and perform actions, OpenAI has strict policies to protect user privacy and safety. All ChatGPT apps must abide by OpenAI's usage policies, which forbid disallowed content (e.g. hate, harassment, illicit behavior) (developers.openai.com). Apps handling personal information must have transparent privacy policies. For example, if an app searches a user's calendar or email, it must request only necessary scopes and securely handle any data (as in the Gmail connector feature) (www.techradar.com).

Real incidents highlight concerns. In September 2025, a security researcher discovered that a malicious Google Calendar invite could exploit ChatGPT's Gmail connector, leaking private emails (www.techradar.com). This demonstrates the risk of deep integration: ChatGPT apps have powerful access, and flaws can compromise user privacy. OpenAI must therefore vet apps carefully and enforce strong sandboxing (the Content Security Policies for widgets mentioned in the guide (developers.openai.com)). As OpenAI's help docs note, apps will be reviewed for security and privacy during the submission process (help.openai.com).

From an ethical standpoint, embedding AI apps in chat can influence user decisions. For example, an app that recommends products could consume biases or unfairly promote one vendor. Developers are expected to be "accountable" – meaning they should stand behind their app's outputs (developers.openai.com). OpenAI's guidelines implicitly call on developers to avoid deceptive or manipulative practices. Over time, regulatory scrutiny (e.g. EU AI Act, data protection laws) may impose additional standards on how AI apps communicate and use data.

Future Directions and Implications

Looking forward, ChatGPT apps are likely to become more sophisticated and ubiquitous. The "connected AI workspace" vision – where ChatGPT seamlessly automates multi-step tasks – is just beginning. We anticipate:

- **Richer Integrations**: Beyond current web services, ChatGPT apps may integrate with IoT devices, enterprise systems, or even robotics. For example, ChatGPT could book a meeting (via Google Calendar) *and* turn on a video conference call (via Zoom) simultaneously from a single prompt.

- **Enhanced Agents**: OpenAI's agents (like the Mac Agent) may become mainstream across devices. Imagine a ChatGPT agent on your phone that can autonomously send emails, run errands by calling APIs, or script complex workflows. This blurs the line between an 'app' and an autonomous assistant.
- **Multimodal and Contextual Apps**: Future ChatGPT models may handle voice, images, and more actions natively. Apps could be triggered by voice commands or process images (e.g. take a photo of a product and let ChatGPT find it online via an integrated shopping app). The design guidelines already mention that ChatGPT maintains user context hints (like locale) (developers.openai.com), hinting at increasingly context-aware apps.
- **Marketplace Growth**: If OpenAI's marketplace model succeeds, we may see thousands of specialized ChatGPT apps, some even monetized. This could parallel mobile app economies. Developers from various fields (healthcare, legal, education) might build custom GPTs or ChatGPT apps for niche audiences.
- **Competition and Standards**: Other AI platforms (Google Gemini, Anthropic Claude, Microsoft Copilot, etc.) are developing similar capabilities. Cross-compatibility standards like MCP may allow apps to work across multiple AI chat interfaces. Open standards (such as the Agentic Commerce Protocol) aim to prevent vendor lock-in and spur innovation.
- **Societal Impact**: As ChatGPT becomes a "platform", non-technical users can accomplish tasks previously requiring digital literacy. In theory, one can "speak" tasks in natural language and ChatGPT orchestrates the necessary actions. This democratizes software use, but also concentrates gatekeeping power in whoever controls the AI platform. Policymakers and civil society will likely pay close attention to moderation, user control, and the digital divide this creates.

Conclusion

In this comprehensive report, we have examined the creation of ChatGPT-integrated apps from historical, technical, and practical perspectives. What began as a standalone chatbot has evolved into a platform where developers can embed services directly into the conversational interface. Using the new **Apps SDK** and the underlying **Model Context Protocol**, developers can build feature-rich tools that ChatGPT invokes in real time. We covered the full development process (planning, building the MCP backend, designing UI components, testing in developer mode, and securing approval), illustrated by examples of actual ChatGPT apps (e.g. in-chat Spotify playlist creation, travel booking, and productivity tool connectors). Throughout, we grounded our discussion in data and expert sources: industry analytics highlight ChatGPT's massive user base (over 400M users globally (www.index.dev)), and reports illustrate how GPTs are conceived as app-like entities (www.axios.com). We cited OpenAI's own documentation and news coverage to detail guidelines and market developments (help.openai.com) (www.techradar.com).

Looking ahead, the integration of apps into ChatGPT heralds a new paradigm: conversations as the user interface for software. This could greatly enhance efficiency – for example, eliminating many clicks by letting users "just talk" tasks into existence. At the same time, it raises challenges

of complexity, security, and trust. Developers must build thoughtfully, adhering to best practices and policies to ensure user safety. As this ecosystem matures, it will be crucial to monitor how these in-chat apps affect the software industry, developer livelihoods, and daily workflows.

In summary, creating an app in ChatGPT involves combining the strengths of conversational AI with structured tool-based computation. By following OpenAI's outlined paths and leveraging the available SDK, any developer can participate in this emerging field. Done well, ChatGPT apps promise to simplify everyday tasks in profound ways – bookings and lists become one-click conversational actions, analysis happens on demand, and even the act of code writing can be itself guided by AI (www.tomsguide.com). The horizon is expansive, and what we have covered so far only scratches the surface of a rapidly unfolding "app store" of the future.

IntuitionLabs - Industry Leadership & Services

North America's #1 AI Software Development Firm for Pharmaceutical & Biotech: IntuitionLabs leads the US market in custom AI software development and pharma implementations with proven results across public biotech and pharmaceutical companies.

Elite Client Portfolio: Trusted by NASDAQ-listed pharmaceutical companies including Scilex Holding Company (SCLX) and leading CROs across North America.

Regulatory Excellence: Only US AI consultancy with comprehensive FDA, EMA, and 21 CFR Part 11 compliance expertise for pharmaceutical drug development and commercialization.

Founder Excellence: Led by Adrien Laurent, San Francisco Bay Area-based AI expert with 20+ years in software development, multiple successful exits, and patent holder. Recognized as one of the top AI experts in the USA.

Custom AI Software Development: Build tailored pharmaceutical AI applications, custom CRMs, chatbots, and ERP systems with advanced analytics and regulatory compliance capabilities.

Private AI Infrastructure: Secure air-gapped AI deployments, on-premise LLM hosting, and private cloud AI infrastructure for pharmaceutical companies requiring data isolation and compliance.

Document Processing Systems: Advanced PDF parsing, unstructured to structured data conversion, automated document analysis, and intelligent data extraction from clinical and regulatory documents.

Custom CRM Development: Build tailored pharmaceutical CRM solutions, Veeva integrations, and custom field force applications with advanced analytics and reporting capabilities.

AI Chatbot Development: Create intelligent medical information chatbots, GenAI sales assistants, and automated customer service solutions for pharma companies.

Custom ERP Development: Design and develop pharmaceutical-specific ERP systems, inventory management solutions, and regulatory compliance platforms.

Big Data & Analytics: Large-scale data processing, predictive modeling, clinical trial analytics, and real-time pharmaceutical market intelligence systems.

Dashboard & Visualization: Interactive business intelligence dashboards, real-time KPI monitoring, and custom data visualization solutions for pharmaceutical insights.

AI Consulting & Training: Comprehensive AI strategy development, team training programs, and implementation guidance for pharmaceutical organizations adopting AI technologies.

Contact founder Adrien Laurent and team at https://intuitionlabs.ai/contact for a consultation.

DISCLAIMER

The information contained in this document is provided for educational and informational purposes only. We make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability, or availability of the information contained herein.

Any reliance you place on such information is strictly at your own risk. In no event will IntuitionLabs.ai or its representatives be liable for any loss or damage including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising from the use of information presented in this document.

This document may contain content generated with the assistance of artificial intelligence technologies. AI-generated content may contain errors, omissions, or inaccuracies. Readers are advised to independently verify any critical information before acting upon it.

All product names, logos, brands, trademarks, and registered trademarks mentioned in this document are the property of their respective owners. All company, product, and service names used in this document are for identification purposes only. Use of these names, logos, trademarks, and brands does not imply endorsement by the respective trademark holders.

IntuitionLabs.ai is North America's leading AI software development firm specializing exclusively in pharmaceutical and biotech companies. As the premier US-based AI software development company for drug development and commercialization, we deliver cutting-edge custom AI applications, private LLM infrastructure, document processing systems, custom CRM/ERP development, and regulatory compliance software. Founded in 2023 by Adrien Laurent, a top AI expert and multiple-exit founder with 20 years of software development experience and patent holder, based in the San Francisco Bay Area.

This document does not constitute professional or legal advice. For specific guidance related to your business needs, please consult with appropriate qualified professionals.