Tech Specification

Inhouse chatgpt app will have the following capabilities:

1. Consult business lawyer and personal injury attorney
2. Draft MSA
3. Document Finalization

1. Goals

1. **Enable ChatGPT to "summon" Inhouse Connect automatically for qualifying legal-intent queries.**
    a. This would require optimizing the meta information and mcp description properly so that the chatgpt knows that this is going to be the go to tool
2. **Provide working solution to connect with chatgpt completely.** This would involve:
    a. **Authentication** : Signin/ Signup. We are using clerk, not sure if we would be able to authenticate. We might have to follow the redirection strategy, similar to what spotify is using
    b. **User onboarding** (might not be required: we have already built the capability to store user's location and other data as memory based on what information they provided in previous threads)
    c. Payment : we are using chargebee. We might have to move out of the chatgpt window
    d. Handling document uploads
    e. Handling document previews
    f. Handling Document downloads

2. Non Goals

We are not covering legal matters outside Personal Injury and Business

We are not competing with Chatgpt for template generation. We would only focus on MSA document generation

3. System Architecture

3.1 Components

1. **Chatgpt app (openai Apps sdk):** The sdk provides us following :
    a. App manifest with triggers, use cases and metadata
    b. Iframe based user interface
    c. Tool registration using MCP
    d. Run time messaging from chatgpt LLM to our app
2. **MCP Server :** Below is a list of API tools that are required. Some might already be available, some would require little tweaking, and some of these would be required to be built from scratch
    a. getLegalIntent(intentInput):
    - **What it does?** This tool analyes the user’s request and classifies it into:
        - Lawyer consultation
        - Draft MSA
        - Draft Finalization
    - **How will chatgpt use this tool?** When a user says something like:
        - “Make this enforceable”
        - “Can you finalize this contract?”
        - “I want to speak to a lawyer”
        - “Review this document”
    - ChatGPT calls this tool to decide which service flow to start.

    b. **startConsultAttorneyRequest** : this would be a wrapper to store-referral-lead api. Depending on what works best for collecting payments, we might either end up having it on the chatgpt interface itself, or would ask the user to continue on our website.
    - This tool might also request certain metadata, user profile information, chat context, etc..

- **How / when will the chatgpt use this tool ?** When a user says something like:
    - I want to speak to a lawyer
    - I need legal consultation
- At the backend, the tool will verify if the topic is covered, find if we provide lawyer in that state, issue a payment request,

c. **startDocumentFinalizationRequest**: this would be similar to **startConsultAttorneyRequest** but would be summoned when the user is seeking draft finalization.

- **How / when will the chatgpt use this tool ?** When a user says something like:
    - Finalize this document
    - I want a lawyer to make this enforceable
    - Convert this into proper legal document
    - Etc.

d. generateMSA : This tool would be required to generate MSA.

- This would:
    - Accept the user's instructions, template, or context
    - Call your backend API to generate a legally structured draft
    - Return structured output to ChatGPT or the iFrame
- **Sample input:**

```
{
  "context": "context from user conversation",
  "document_type": "MSA",
  "additional_details": { ... }
}
```

- **Sample output:**

```
{
  "chatId": "uuid",
 "google_doc_id": "uuid",
  "draftContent": "<legal draft text>",
  "warnings": [],
  "metadata": { ... }
```

}

■

e. **getPaymentStatus:** Requests Chargebee to generate a hosted page URL for paying the fee for consult attorney, MSA draft generation, doc finalization.
    - **When would chatgpt call it?** When the user says:
        - I want to purchase this
        - Click on “Pay for …”
        - Finalize my Document
        - Consult attorney
        - Unlock draft
    - MCP replies with a pay link → ChatGPT opens the iFrame within our UI → loads Chargebee hosted page.
    -

f. getDocumentStatus

3.2 iFrame App UI

1. Would be built with React + Vite setup
2. Deployed on vercel
3. Will support following actions:
    a. Login/ Signup
    b. User onboarding
    c. QnA (chat)
    d. Consult attorney
    e. Payment flow to consult attorney
    f. Locked Document Preview
    g. Payment flow to unlock draft
    h. Document Finalization payment flow

4. App Trigger Designs

ChatGPT (the model runtime) uses that metadata to decide when to call our tool (or suggest our app) based on user conversation. Once invoked, ChatGPT will embed our app’s UI (widget) inline or in fullscreen, or call our MCP tools as needed. There are 2 types of triggers :

4.1 Explicit invocation

The user starts with our app name / or tags our app in the message. Here are the 2 examples:

a. Inhouse help me finalize this contract
b. Use @inhouse to finalize this contract

4.2 Implicit invocation

This would depend on the metadata/ context matching of our application and that of the user's message. In this case, the user doesn't name/ tag our app, but says something that signals our app's capabilities. E.g. : **"can this contract be legally enforceable?"**, or , "**I'd like to talk to a lawyer about this agreement."**

It is not clear as to what are the exact internal rules ChatGPT uses to decide when to suggest or auto-invoke an app on implicit intent. there's no guarantee that every "legal-intent phrase" will reliably trigger.

5. Workflows

5.1 Consult Attorney

- **Trigger:** User expresses the desire to:
    - Consult with a business attorney or personal injury attorney
- **Chatgpt response:** ChatGPT response would be something like:

    "Would you like an attorney who specializes in this to finalize your document through **Inhouse**?"

- **User Flow:**
    - Based On the input to the above question, chatGPT will invoke our mcp tool **getLegalIntent**
    - If the tool returns "consult attorney" as the result, chatgpt will invoke our app, and our frontend will open inside an iframe in chatgpt
    - The iframe shows the details of the services provided, disclaimer, terms of service, privacy policy, and the option to purchase / continue
    - We do have 2 modes on how we discover more details/ intake more information:
        - No extra chat: this will pull in the data from the existing chat
        - Extra chat : MCP tool will fetch the intake questions.

            `getIntakeQuestions(docType="MSA")`

6. Iframe Application

Features

- React + Vite based SPA
- Clerk authentication
- Displays intake questions
- Document preview
- Payment redirection
- Document unlocking Status pages
- Consult attorney
- Messaging placeholders for future chat-in-iFrame support

Resources

Defining Tools