You'll be developing a Google Calendar-based backend tool that supports a **Pending Reservation Assistant** workflow for use inside the ChatGPT SDK. This tool is designed to help users detect calendar events they've received but haven't responded to — and quickly act on them — via a clean in-chat experience powered by ChatGPT's app platform.

ChatGPT itself cannot manage Google Calendars, so your backend will expose authenticated endpoints for ChatGPT to call, including logic for identifying unresolved calendar invites and enabling users to accept or reject them conversationally.

The goal is to provide a fast, simple utility: **"Show me calendar invites I haven't responded to in the past 14 days"** → **"Accept this one, reject that one."**

Reference Materials

- **ChatGPT SDK Overview**: https://openai.com/index/introducing-apps-in-chatgpt/
- **Example SDK Code**: https://github.com/openai/openai-apps-sdk-examples
- **Me Going Through The Requirements: https://drive.google.com/file/d/1wxCx7KYkCI_NjQ-86y3ZxgqKE7Qt3i-o/view?usp=sharing**
- **Example Video**: https://www.youtube.com/watch?v=wsPUYCa2TCw

---

Authentication

- You'll implement **OAuth 2.0 authentication** with Google Calendar on behalf of the user.
- Store and manage access tokens securely. Users must stay within the ChatGPT app — no external site redirection allowed after auth.
- You must support token refreshing to maintain authorization for repeated interactions.

Functionality to Support

✅ `GET /pending_reservations`

Purpose: Fetch unresolved calendar invites from the user's Google Calendar.
Behavior:

- Default date range: events scheduled between **now and 14 days from now**.
- ChatGPT can customize this range by passing `start_date` and `end_date` (ISO 8601 format).

- Only include events:
  - Where the **user is an attendee** (not organizer).
  - And the user’s **RSVP status is “needsAction”** (i.e. not accepted/declined/tentative).
- Return the following info per event:
  - Event ID, summary/title, description
  - Organizer email
  - Start and end time
  - Location
  - Attendees list (with emails + statuses)
  - Link to Google Calendar event
- Response should include a **UI panel** showing these events clearly, with previewable details per event.

✅ `POST /respond_to_invite`

Purpose: Accept or reject a pending calendar invitation on behalf of the user.
Payload:

```
{
  "event_id": "string",
  "response": "accepted" | "declined" | "tentative"
}
```

Behavior:

- Use Google Calendar API to update the user’s attendee status for the given event.
- Once successful, return confirmation payload.
- User can initiate this command in natural language like:
  - “Accept the first invite”
  - “Reject the one from Bob on Friday”
- UI must include a **button-based selection** for each event with response options: ✅ Accept | ❌ Decline | 🤷 Tentative.

UI Requirements

- Must expose a clean interactive panel inside ChatGPT for:
  - Listing unresolved invites in the selected date range.
  - Offering buttons to quickly respond per event.
- **Progress indicators are not required** — operations are atomic.

- Use inline messaging confirmations for actions (“You accepted the meeting with Alice on Dec 22 at 3 PM”).

Technical Notes

- **Avoid unnecessary re-fetching**: ChatGPT apps may call the same endpoint multiple times. Include safeguards (e.g. memoization, timestamp-limited caching) to avoid duplicate work or performance issues.
- All actions must work **within the conversation context** — user should never leave the chat.
- You are expected to write clean, maintainable code and work incrementally.
- You must provide **code access during development**, and implementation should be complete **within 7 days**.