Meet Scheduler & Bigger Idea

I've conceptualized an automatic meeting scheduler. Here's how it works: when someone wants to meet with you, they send a message with their name and email address. The AI agent checks your Google Calendar, identifies an available time slot, schedules a Google Meet, and sends the meeting details (date, time, and link) to the client. For added convenience, clients can submit requests via voice or text message, and the AI agent will process and schedule the meeting accordingly.

Overview

It automatically schedules online meetings based on user input received via **Telegram**. It leverages **Google Gemini AI** for natural language understanding and integrates with **Google Calendar** to create and share meeting invites.

Key Features

1. **Voice and Text Input Handling**
    - The workflow detects whether the Telegram message is a **voice note** or **text message**.
    - For voice messages, the system downloads and transcribes the audio using **Google Gemini's transcription capabilities**.
    - Text messages are directly forwarded to the AI processing stage.
2. **AI-Powered Meeting Scheduling**
    - The system uses **Google Gemini (PaLM) API** as the LLM (Large Language Model) to understand natural language inputs.
    - It extracts relevant scheduling information such as:
        - Meeting date and time
        - List of participants
        - Meeting purpose (optional)
3. **Structured Data Extraction**
    - The **Structured Output Parser** node ensures that AI output is formatted in JSON containing:
    - {
    - "start_time": "start time",
    - "end_time": "end time",
    - "candidates": ["email1", "email2"]
    - }
    -
    - This standardized format enables seamless integration with the Google Calendar API.

4. **Google Calendar Integration**
    - Once the structured data is received, the workflow creates an event in **Google Calendar**.
    - A **Google Meet** (Hangouts) conference link is automatically generated.
    - Attendees are added to the event using the extracted email addresses.
5. **Automated Telegram Notification**
    - After successfully scheduling the meeting, the workflow sends a Telegram message back to the user:

"Your meeting is scheduled. Here is the meet link: [Google Meet Link]"

Workflow Architecture

The automation consists of the following major nodes:

Node Name	Function
Telegram Trigger	Detects incoming Telegram messages (text or voice).
Switch	Differentiates between audio and text messages.
Download Audio	Retrieves the audio file from Telegram servers.
Transcribe a Recording	Converts the voice input into text using Google Gemini AI.
Text Message / Transcribed Audio	Normalizes input data for processing.
AI Agent	Interprets the message, extracts meeting details, and outputs structured data.
Structured Output Parser	Validates and structures AI output for Calendar API compatibility.
Create an Event	Uses Google Calendar to create a meeting with attendees and a meet link.
Send a Text Message	Sends the confirmation and link back to the user on Telegram.

Technology Stack

- **n8n Workflow Automation Platform**
- **Google Gemini (PaLM API)** – AI model for natural language processing and audio transcription
- **Google Calendar API** – Event creation and Meet link generation
- **Telegram Bot API** – User interaction interface
- **JavaScript-based Node Configuration** within n8n

Working Process (Step-by-Step)

1. A user sends a **text or voice message** on Telegram such as:

"Schedule a meeting with John and Priya tomorrow at 4 PM."

2. The **Telegram Trigger** node captures the message.
3. The **Switch** node identifies if it's a **voice** or **text** message:
    - If voice → Audio is downloaded and transcribed.
    - If text → It's directly used.
4. The transcribed or original text is processed by the **AI Agent** powered by Google Gemini.
5. The **AI Agent** extracts date, time, and participant emails in structured JSON format.
6. The **Create Event** node sends this data to **Google Calendar**, generating a meeting and link.
7. Finally, a **Telegram message** is sent back to the user with the confirmation and the Google Meet URL.

Use Cases

- Professionals or teams wanting quick meeting scheduling through chat.
- Hands-free scheduling via voice messages.
- Automated calendar management for personal assistants or chatbots.
- Integration with business bots to handle meeting requests dynamically.

Advantages

- **Hands-Free Scheduling:** Users can speak naturally or type messages—no manual entry needed.
- **AI Understanding:** Automatically detects intent, timing, and participants.
- **Real-Time Calendar Sync:** Directly updates the Google Calendar with correct details.
- **Cross-Platform Convenience:** Works seamlessly on Telegram, accessible via mobile or desktop.
- **Error Reduction:** Structured AI output ensures accurate scheduling

Another Idea that I have is:

I've been thinking about this concept even before taking the course. Let me explain my idea:

I'm a big fan of Varun Mayya—a business genius and marketing expert who became a content creator and has dominated this space for 10 years. Why mention him? Because of his social media content process. On his Instagram profile, you'll see all his impressive content is created using AI agents—from writing and video production to his voice—all generated by AI and refined by his team.

I've always wanted to create something similar, and enrolling in this course now allows me to experiment with AI agents. I want to expand on this concept further. While Varun Mayya's approach and most AI content models focus on writing and voice/video enhancement, social media encompasses more than just videos. Photos are equally important, and I want to incorporate this element into my model.

The challenge lies in getting AI to edit photos. This typically requires APIs from photo editing software, but popular tools like Canva and Photoshop either don't offer APIs or use proprietary ones. My solution: browser-based photo editing software. We can use Photopea, a simplified Photoshop alternative, and leverage AI browsers like Comet by Perplexity to edit pre-made templates for carousels or individual photos.