

see

~/Downloads/NoteGPT-All/Business_Analysis_20250501_162645.md and Master_20250501_154012 (Thur May 1, 4pm) or ~/Downloads/NoteGPT-All/MasterMDDocs folder

Chrome_Extension_PROJECT_PLAN_2025-05-05_14-42-02.md



Master Document: NoteGPT Clone – Full Project Organization (Draft)

Table of Contents (not up to date)

- 1. Web App Structure & Core TODOs**
- 2. Features Expected to be Missing in MVP**
- 3. PDF Detail/[id] Page Prototype and See SimplePDFViewer github repo for PDF File Uploader example**
- 4. Backend Server**
- 5. Environment Variables**

6. Firebase Authentication

7. Feature Table, MVP Priorities & Database Schema

8. Known Bugs and Suspected Issues

9. UI/UX Improvements & Layout Recommendations

10. Competitor Research

11. Suggested Roadmap

12. Suggestions from Perplexity

13. AI Chat with PDF or Video

0. Misc Notes

Please commit to Repo daily

Cursor Account: andrew.duquet.dev@gmail.com pwd: softwaredev

Preventing Cursor AI from Changing Hand-Edited Code

Cursor AI allows you to influence its behavior globally by using a special file in your project directory called `cursor.rules`. This file can be used to set instructions that Cursor will follow whenever it interacts with your codebase.

How to Set a Global Rule to Prevent Code Changes

ToDo List

Fix whatever auth mistakes I made (also maybe allow a user to experiment without logging in??)

PDF Detail Page buttons (see section: **PDF Detail / [id] Page below**) - **remove timestamp from Note**

PDF Uploader/Drag and drop (see section: **3b. SimplePDFViewer repo**)

PDF thumbnail for Notes Page

Sticky buttons and other UI/UX and Bugs (see **UI/UX Improvements & Layout Recommendations**)

streamline create page to add a note -> Notes Page -> Detail/[id] page

database

mind maps (see prototype ~/Downloads/NoteGPT-Clone-Apr29at1222)

user subscription level

browser tab title

header titles (Details)

Performance issues?

Dev Commits from Terminal (example: mindmaps)

```
1057 cd notegpt-clone
1058 ls
1059 git status
1060 git stash
1061 git status
1062 git branch
1063 git checkout -b dev
1064 git branch
1065 git stash list
1066 git stash apply
1067 git branch
1068 git add .
1069 git commit -m "MindMaps"
1070 git push origin dev
```

andewduquet@Andrews-MBP-2 notegpt-clone %

For example, did from

andewduquet@Andrews-MBP-2 NoteGPT-Clone-Apr29at1222 % for Mind Maps on Wed Apr30at4:11

See up to date MD Documents

**~/Downloads/NoteGPTClone Folder and messaged to
Ahmar on 4/25/25 at 5:30pm**

Also, on cursor, on my macbook:

~/Downloads/NoteGPT-All/

note-gpt-clone-main-Apr22at451pm (has token level transcript tracking)

latest zip backup: ~/Downloads/notegpt-clone-main-Apr22at451pm 3.zip (April 25,12:03)

~/Downloads/notgpt-pdfviewer2

~/Downloads/firebase-auth-master...

~/Downloads/testdir2 (has both PDF and Auth I think)

~/Downloads/TestPDF (has test.pdf and MD docs for PDF)

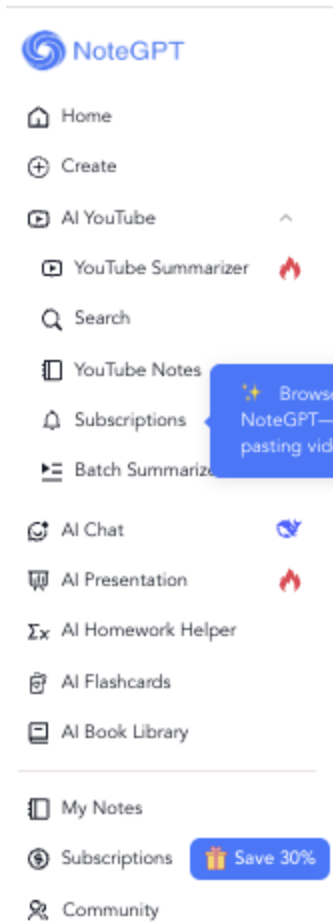
Ok for me to remove:

cache: failed SpaCy and NLTK effort

1. Web App Structure & Core TODOs

1.1 Webpages

- **Overview**



- **Home** (Say PDF as well as Youtube Video)

- **Create**

Tab option 2: PDF Uploader/Drag-Drop (screenshot also shows the PDF thumbnail image of a PDF Note in the middle of the 2nd row) - see



Upload or drag a file here.

PDF, Image, PPT, Word, TXT, eBook: max 30MB

Audio, Video: max 40MB

Supported file formats

OR

Upload to Summarize

- Home
- Create
- AI YouTube
- AI Chat
- AI Presentation
- AI Math & Homework
- AI Flashcards
- AI Book Library
- My Notes
- Subscriptions Save 30%
- Community

Home

Dark

Hi~ Let me help you get started!

Learned for 57 days. NoteGPT has saved you 11 hours.

Over 4 million users are using NoteGPT with you.

- Summary
- YouTube
- AI Chat
- Presentation
- Writer
- AI Math
- Flashcards
- Diagram

YouTube File (PDF, Image, Audio & More) Link Long Text English



Upload or drag a file here.
PDF, Image, PPT, Word, TXT, eBook: max 30MB
Audio, Video: max 40MB
Supported file formats

Upload to Summarize

Continue Exploring

My YouTube Channels

Search YouTube Videos

AI Podcast Generator

Recent Notes

More



Harvard Professor reveals the Science of Happin...
April 14, 2025



Last Lecture Series: How to Live your Life at Full ...
April 14, 2025



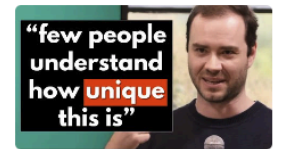
How to download any YouTube transcript (in 30 s...
April 8, 2025



Delta plane, Air Force jet nearly collide at Reaga...
April 7, 2025



1628F089-7A5A-4286-8F8B-70D9A79B4F93.pdf
March 28, 2025



Andrej Karpathy: This Is Elon Musk's Secret To S...
March 25, 2025



English in a Minute: Grow Like a Weed
March 21, 2025



Mariam Shengelia - Freedom | Georgia
March 21, 2025



PTI Refuses To Back Anti-Terror Operation | Ali A...
March 20, 2025

More

File (Image, Audio & More) (*not in MVP?*)

Webpage Link (*not in MVP?*)

Long Text File (*not in MVP?*)

Summary Settings (sprocket icon)

Language dropdown

Batch Summarize

Multi-Links

Continue Exploring (*not in MVP?*)

Recent Notes (with "More" button)

- **PDF Detail / [id] Page**

Upload and drag-drop from Create or Home Page

Top Buttons: Preview, etc.

Bottom Buttons: Copy, Download

Left Column Top Buttons:

- **Notes**
- **Summaries and custom prompt**
- **Quiz (working, with console error)**
- **PDF and Youtube Detail/[id] Page**
 - Copy to Markdown (all) - Upper Left Corner**
 - Sticky Buttons for PDFs and YouTube videos**
- **Login Pages**
- **Database (User settings - language, etc.)**
- **Languages (Multi-language dropdown)**
- **Sidebar Navigation**

Collapse/Expand partial sidebar (like NoteGPT.io)

Sections:

- **AI YouTube: Summarizer, Search, Notes, (Subscriptions not MVP), (Batch Summarize not MVP)**
 - **AI Flashcards**
 - **AI Chat (*Not in MVP*)**
 - **AI Presentation (*Not in MVP*)**
 - **AI Math/Homework (*Not in MVP*)**
 - **AI Book Library (*Not in MVP*)**
 - **My Notes**
 - **Subscriptions**
 - **Community**
- **Back Link next to filename.pdf or YouTube title.**
-

2. Features Expected to be Missing in MVP

- **AI Chat**
 - **Mindmaps** - See ~/Downloads/add-mindmap-feature-proposal.md shared in messages and via shared gdoc
 - **Video Tracking Off**
 - **YouTube Video Discover**
 - **Community**
 - **AI Book Library**
-

3a. PDF Detail/[id] Page Prototype

- **My PDF Prototype is Working**
Example URL: save file to /public/test.pdf first

`http://localhost:3000/workspace/detail/123?type=pdf&pdfUrl=/test.pdf`
- **Compressed Archive**
`notegpt-clone-main-Apr22at451pm` compressed

on 4/25/25 noon

- GitHub Repo (includes Markdown Implementation Guides)

<https://github.com/duquet/notegptwithpdf/>

3b. SimplePDFViewer repo giving example of how to upload and drag-drop PDF in case helpful for Create or homepage PDF Upload/Drag-Drop

- Repo: <https://github.com/duquet/simplepdfviewer>
 - Notes: Includes basic README.md documentation about
 - CORS issues
 - PDF.js handling
 - Test environment setup using a local **test.pdf**
-

4. Backend Server

Environment & directory structure

```
cd api
```

```
rm -rf env  
python3 -m venv env  
source env/bin/activate  
pip install -r requirements.txt  
python3 api.py
```

5. Environment Variables

```
NEXT_PUBLIC_FIREBASE_API_KEY="AlzaSyDhful1NDR  
ZKGaEXu5HUhyiwU9zz8YigdY"
```

```
NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN="notegpt-clone-56652.firebaseio.com"
```

```
NEXT_PUBLIC_FIREBASE_PROJECT_ID="notegpt-clone-56652"
```

```
NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET="notegpt-clone-56652.firebaseiostorage.app"
```

```
NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID="402711510015"
```

```
NEXT_PUBLIC_FIREBASE_APP_ID="1:402711510015:web:0f7e244f7cf089a78845ca"
```

**NEXT_PUBLIC_OPENAI_API_KEY="sk-proj-71Cg2JMYPa
EgQtlvSlrcnlWV0oEn5kO9q9wFaCttU9_DOTjOyPQOxqja4
Kven589e8oKlh6HFuT3BibkFJIU_INbJsJsoC-T8NSqjatkd
asXdmkvM0g97t-2RprbocRV5ljZb1awLCLVU6C88OedSVZ
inkcA"**

**DATABASE_URL="postgres://notegpt-clone_owner:npg_
Ln2CB3afoZpi@ep-empty-morning-a5heol5x-pooler.us-ea
st-2.aws.neon.tech/notegpt-clone?sslmode=require"**

6. Firebase Authentication

Login should be optional.

Firebase Auth currently supports both Google and Email/Password login.

Minor future improvements suggested (password visibility toggle, optional additional providers).

I also created a <https://github.com/duquet/firebase-auth> which uses a .env.local

7. Feature Table, MVP Priorities & Database Schema

7.1 Feature Table (Summary)

Feature	NoteG PT.io	MV P	MVP 2.0
YouTube Transcript Summarization	✓	✓	✓
PDF Summarization	✓	✓	✓
Image/PPT Summarization	✓	?	✓
Mindmaps	✓	?	✓
AI Flashcards	✓	✓	✓
AI Chat	✓	?	✓
Timestamped Notes	✓	✓	✓

Subscription/Monetization	✓	?	✓
Chrome Extension	✓	X	✓
API Key management (user-supplied)	?	?	✓

7.2 Database Support Needed

Feature	Required?
Authentication	✓
Subscriptions	✓
Usage Tracking vs Subscription Level	✓
Summaries	✓
Flashcards	✓

Notes (YouTube, PDF, Text) ✓

Mindmaps X
(later)

Bug Reports TBD

Batch Summarization TBD

Settings (Language, Theme) ✓

8. Known Bugs and Suspected Issues

8.1 Known Bugs

- "Add Note" timestamp often reads 00:00.00 (even when playing video)
- Detail/[id] page header incorrectly shows "Dashboard" instead of "Details"
- "Summary Edit" button currently does nothing
- Quiz (Flashcards) browser console error
- Excessive ytdl calls on every /video POST
- Summary reload required before Flashcard generation
- Missing shimmer loading (non-MVP)
- Flashcards header formatting mismatch with NoteGPT.io
- Chrome export error: "The default export is not a React Component in /workspace/detail/[id]/page"

8.2 Bugs Observed on NoteGPT.io

- Can't delete Default Summaries — regenerates immediately after Delete
 - Homework Helper sometimes fails to load
-

9. UI/UX Improvements & Layout Recommendations

- Sticky transcript controls while scrolling transcript
 - Sticky top-right control panel while scrolling Summary Cards
 - Sidebar: show collapsed buttons instead of fully hiding sidebar
 - Breadcrumb: add < before YouTube video title (with ellipsis)
 - Summaries and Flashcards: consistent title styling with NoteGPT.io (rounded blue box bottom left)
 - AI Flashcards page: center Q/A content
 - Dark/Light mode toggle: replace moon icon with slider switch
 - Add tooltips to buttons
 - Summary: "Key Insights" should be bullet points, not far-left numbers
 - Search suboption under AI YouTube (Sidebar)
 - Smooth onboarding: Summary always visible or toggleable on Detail/[id] page
 - Global header polish: remove underline under logo; align bottom borders
 - Suggestions Panel: add 3 suggested next actions at right-column bottom
-

10. Competitor Research

10.1 Key Competitors

- Otter.ai
- Descript
- Fireflies.ai
- Tactiq
- Sonix
- Rev
- Zoom AI Companion
- Notion AI
- Mem

10.2 Observations & Opportunities

- **Mind Maps** are a differentiator (NoteGPT has it, few others do)
- **Multi-Model Support** would differentiate (OpenAI, Claude, Gemini, Mistral)
- **Interactive Summaries**: allow follow-up questions

- **Prompt Marketplace:** importable/exportable templates
 - **Smart Tagging & Knowledge Graphs**
 - **Team Collaboration Tools**
 - **Batch Summaries** for researchers/students
 - **Offline Access and Mobile Apps** (future)
-

11. Suggested Roadmap

11.1 MVP Core

- Add Note (with timestamp)
- Login/Auth system
- Flashcards and Summaries integration
- Basic Create page: YouTube, PDF
- Sidebar navigation polish
- Free/Pro subscription system

11.2 MVP 1.1 (First Improvements)

- Save Summaries and Flashcards into Database (reduce API costs)
- Soft-delete user content (is_deleted field)
- Multi-Language UI toggle
- Sticky controls for smoother UX
- Better transcript segment UI (collapsible, timestamps only)

11.3 MVP 2.0 (Extended)

- Mind Maps generation
- Chrome Extension
- PDF upload refinements (multi-page tests)
- AI Chat/Q&A module
- Mobile-responsive improvements

12. Non-Software To-Dos

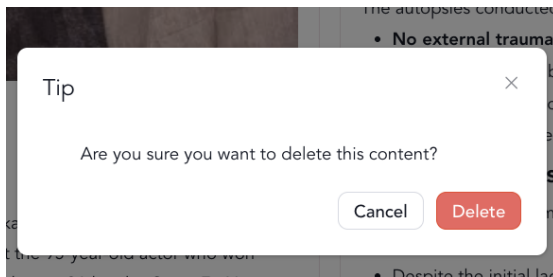
- Build a community on Discord (and similar platforms)
- Produce YouTube videos showcasing our clone

- Figure out competitive positioning (very important!)
-

13. Questions

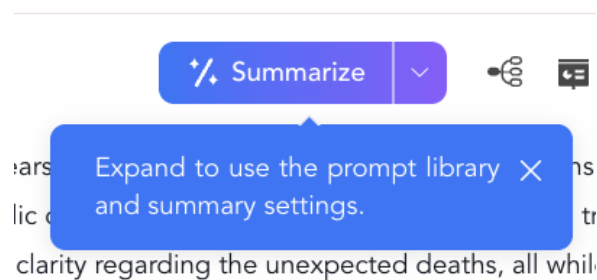
- Why does NoteGPT.io append `?from=yt` in its `detail/[id]` URLs?
- I don't understand what "Auto-snap during playback" means in context of NoteGPT.io (which supports it).
- How hard is it to add MindMaps (see gDoc), Chat, PDF upload, non-YouTube video support, etc.?
- Discuss Database deletion strategie such as Soft Delete / Retention
- **Global header polish:** remove underline under logo; align borders, change icon for collapse sidebar to show right and left in the middle horizontal line of the 3 horizontal lines
- **Vertical Scroll bar on the Details/[id] right column shouldn't hide top buttons**
- **Button indicator:** shouldn't show circle next to "Add Note"
- **Suggestions panel:** add three suggestions at right-column bottom
- **Transcript UI:** show only start time; collapsible segments
- **Dark/Light toggle:** slider instead of moon icon
- **Video controls:** add "i" icon and 3-dot menu
- **Breadcrumb back link:** add `<` before youtube video title; ellipsize
- **Flashcards title:** rename to "AI Flash Cards" and put in blue box in lower left
- **Nav arrows:** ensure `< 1/10 >`
- **Sidebar collapse:** show sidebar buttons instead of full collapse of sidebar
- Flashcards: Center content of questions and answers

- Summaries: When user clicks Delete, add a NoteGPT.io -like Alert



NOTE: NoteGPT.io bug: Doesn't delete the Summary. All Summaries and Flashcards are automatically regenerated after the Alert's red Delete button is clicked. This means you can never delete the default summary or other summaries!

- Summary tool tip:



14. Sticky - Scrolling behavior on Detail/[id] page

15. Ai Chat

To build a powerful AI Chat feature for your NoteGPT clone (based on YouTube transcripts and PDF contents), here's a structured breakdown of how to think about implementation and architecture:

Core Use Case

Enable users to chat with AI based on:

1. YouTube video transcripts (already parsed and segmented)
2. Uploaded PDF documents (parsed and enhanced)

Chat should be context-aware, fast, and scalable — ideally with tools like OpenAI **gpt-4-turbo** + retrieval + memory.

High-Level Strategy

1. Define Chat Context Scope

- Chat should only use:
 - YouTube transcript (enhanced or raw)
 - PDF extracted text (paragraphs, sections, metadata)
 - Optional: summary/flashcards (if well structured)
- Avoid:
 - Non-textual PDF content (images, charts, scanned pages unless OCR'd)
 - Full document history unless helpful (adds token bloat)

2. Store Sources Cleanly

- For each **videoId** or **pdfId**, store:
 - **full_transcript.json** or **full_pdf_text.json**
 - **enhanced_transcript.json** or **cleaned_pdf_text.json**
 - **summary.md** or **summary.json**
 - Optional: **metadata.json** with author, title, sections
-

Implementation Steps

1. Frontend: Trigger Chat Mode

- From **/detail/[id]** (for both videos and PDFs):
 - Show “Ask AI” or “Chat about this content”
 - On click: open chat UI and send content identifier (**type=pdf|yt + id**) to backend.
-

2. Backend: Prepare Context Window

A. Basic Strategy

- At first load, fetch relevant content (transcript or PDF text)

Create a **system** prompt like:

You are an AI assistant helping the user understand the following [video|document]. Refer to its content when answering.

-
- Inject full cleaned text or smartly chunked context (see below).

B. Optional Enhancements

- Summarize each section with OpenAI (**gpt-3.5-turbo** is cheap)
- Use embeddings + vector search for retrieval-augmented generation (RAG) later
- Store context chunks in a fast cache (e.g., Redis or file-based)

3. Chunking and Token Management

You need to stay under OpenAI token limits. So:

- ◆ For YouTube:

- **Chunk by timestamp groups (~500 tokens per chunk)**
- **Include speaker attribution if useful**
- **Optionally summarize sections before chat**

◆ **For PDFs:**

- **Chunk by:**
 - **Page**
 - **Heading/Section titles**
 - **Paragraphs**
- **Strip out footers, page numbers, image tags, etc.**

4. Deciding What to Include

Content Type	Include ?	Reason
Enhanced Transcript Text		Primary chat material

Summary Page	✓ (optional)	Helps AI answer better but shouldn't replace raw context
PDF Extracted Text	✓	Use cleaned, readable sections
PDF Metadata	✓	Inject into prompt for more context (title, author, etc.)
PDF Images/Charts	✗*	Only include if you OCR or caption them; ignore for initial version

5. AI Chat Prompt Design (Initial MVP)

[

{

role: "system",

content: "You are a helpful assistant answering questions about the following [video/document]. Refer to the content provided."

},

{

```
    role: "user",  
  
    content: "What was the main argument in section 2?"  
  
  },  
  
  {  
  
    role: "assistant",  
  
    content: "Based on the transcript, section 2 covers..."  
  
  }  
]
```

Later, enhance with memory, citation links, or chunk-aware retrieval.

Testing Suggestions

- Pick 2-3 test documents/videos and run:
 - "What is this about?"
 - "Summarize section 2"
 - "What did the speaker say about X?"

- "Who is the author and what is their main claim?"

Compare answers using:

- Transcript-only
 - Transcript + Summary
 - Transcript + Summary + Metadata
-



Next Steps

1. Create utility functions:

- `getChatContext(id, type)` → returns cleaned content
- `getSystemPrompt(type)` → returns dynamic system prompt

2. Add frontend chat modal

3. Set up backend `/api/chat` route

4. Use `gpt-4-turbo` with streaming and context batching

5. Add test cases to verify quality

Would you like a code scaffold for the `/api/chat` handler and frontend chat modal next?

Feedback from Perplexity:

Your todo list for a NoteGPT.io clone is impressively comprehensive and well-structured. Based on a deep analysis of NoteGPT's public features, user reviews, and technical differentiators¹²³, here are targeted suggestions and areas for improvement or addition:

Additions & Improvements

1. Feature Parity and Differentiators

- **Custom Template Creation & Output Customization:** NoteGPT allows users to create and select summary templates and customize output formats directly from the settings panel². Consider adding:
 - A “Custom Templates” section in your UI and API.
 - Settings for users to define and save their own summarization prompts.
 - Output format options per content type (YouTube, PDF, PPT, etc.).
- **Smart Tagging System:** NoteGPT uses intelligent tags for organizing notes and building a personal knowledge base². Add:
 - Tag creation, editing, and search/filter features.
 - Automatic tag suggestions based on content analysis.
- **Multi-Language Support:** NoteGPT processes content in 40–50+ languages and offers one-click translation for summaries and transcripts²³. Ensure:
 - Multilingual summarization and transcript generation.
 - UI and content translation features (not just tooltips).
- **API Integration:** NoteGPT provides advanced API access for custom workflows and integration with multiple AI models (ChatGPT, Claude, DeepSeek, Mistral, Groq, etc.)².

For your clone:

- Plan for modular AI backend support.
- Expose APIs for content upload, summarization, and note retrieval.
- Allow users to connect their own API keys for different LLMs.
- **Cloud Sync & Device Consistency:** NoteGPT syncs notes across devices and saves all custom settings/content in a personal library². Consider:
 - Real-time sync for notes, settings, and templates.
 - Cloud-based storage and retrieval for user data.

2. UI/UX & Accessibility

- **Advanced Settings Panel:** Many NoteGPT features are hidden in a robust settings menu². Your clone should:
 - Expose all customization and template options in a dedicated settings area.
 - Allow users to toggle advanced features (e.g., token-level tracking, output format).
- **Mind Map Visualization:** NoteGPT's mind map creation is a major selling point¹³. Prioritize:
 - Mind map generation from summaries and notes.
 - Export and share options for mind maps.
- **Note Organization:** Folders, tags, and advanced search are key for power users¹. Ensure:
 - Folder management, tag-based filtering, and search by content type, date, or tag.
- **Chrome Extension Parity:** The extension supports sidebar and embedded summarization with consistent settings¹². Plan for:
 - Chrome extension with sidebar and in-page summarization.
 - Sync extension settings with web app.

3. Security & Privacy

- **Data Handling Practices:** NoteGPT emphasizes privacy (no PII collection, auto-deletion of uploads), but lacks GDPR compliance²³. For your clone:
 - Make privacy policies clear (auto-delete files, no PII storage).
 - If targeting EU users, consider GDPR compliance or at least transparent data handling documentation.
- **User Data Export:** Allow users to export all their notes, summaries, and mind maps in common formats (Markdown, PDF, CSV).

4. Performance & Reliability

- **Batch Processing:** NoteGPT supports batch summarization and handles high traffic². Your MVP should:
 - Support batch summary requests for multiple videos or documents.
 - Optimize for speed and reliability, especially for academic and legal content.

5. Documentation & Community

- **User Guides for Advanced Features:** Since many users miss advanced settings², include:
 - In-app tooltips or onboarding for customization features.
 - FAQ and troubleshooting sections for common issues.
- **Comparison Table:** You already mention this, but ensure it's up-to-date and highlights unique features and pricing versus competitors like Otter.ai, Descript, Fireflies.ai, etc.²³.

Table: Key Feature Comparison & Recommendations

Feature Area	NoteGPT.io Status	Your TODO List	Recommendation/Addition
Custom Templates/Prompts	✓ (advanced, per content)	Partial	Add user-defined templates & prompt mgmt

Smart Tagging	✓	Partial	Add auto-tagging and tag-based search
Mind Maps	✓	Planned	Make a priority for MVP 2.0
Multi-language Summarization	✓ (40–50+ languages)	Partial	Expand language support & UI translation
API Integrations	✓ (multi-LLM, open API)	Partial	Modular AI backend, user API keys
Output Customization	✓	Partial	Per-content output format options
Cloud Sync	✓	Partial	Real-time sync for notes/settings
Chrome Extension	✓	Planned	Ensure feature parity with web app
Privacy/Auto-Delete	✓ (no PII, auto-delete)	Partial	Add clear privacy policy, auto-delete
Batch Processing	✓	Partial	Optimize for batch summaries
User Data Export	✓	Not listed	Add export options (MD, PDF, CSV)
Advanced Settings Panel	✓	Partial	Centralize all customization options

Minor Suggestions

- **Accessibility:** Add screen reader support and keyboard navigation for all major UI components.
- **Error Handling:** Robust error messages for failed uploads, API issues, or unsupported formats.
- **Usage Analytics:** Track feature usage (anonymously) to inform future improvements.

Questions for Further Clarification

- **Speaker Identification:** NoteGPT currently lacks speaker ID in transcripts³. Is this a priority for your clone, especially for meeting or multi-speaker audio?

- **GDPR Compliance:** If you plan to serve EU users, is GDPR compliance on your roadmap?
- **Offline Support:** NoteGPT is internet-dependent³. Is offline note access or local caching needed for your users?

Summary:

Your todo list covers most core and advanced features of NoteGPT.io, but could be improved by prioritizing custom template management, smart tagging, advanced output customization, robust multi-language support, API integration, cloud sync, and privacy features. Adding user data export, a comprehensive advanced settings panel, and ensuring Chrome extension parity will help you match and potentially surpass NoteGPT's current offering¹²³.

Citations:

1. <https://notegpt.io>
2. <https://clauideai.guru/notegpt-review-the-hidden-features-most-users-miss/>
3. <https://www.meetjamie.ai/blog/notegpt>
4. https://chromewebstore.google.com/detail/notegpt-yt-summary-webpag/baecjmoceaobp_nffgnlkloccenkoibbb
5. <https://notegpt.io/blog/how-to-summarize-youtube-videos>
6. <https://notegpt.io/interview/dinesh-journey-with-notegpt>
7. <https://notegpt.io/blog/pdf-summarizer>
8. <https://notegpt.io/blog/how-to-use-the-notegpt-workplace>
9. <https://www.aitechsuite.com/tools/notegpt.io>
10. <https://aijourney.so/tool/notegpt-io>
11. <https://notegpt.io/faq>
12. <https://www.eliteai.tools/tool/notegpt>
13. <https://github.com/Adarsreg/Notegpt>
14. <https://www.revoyant.com/product/notegpt>
15. <https://updf.com/chatgpt/notegpt/>
16. <https://notegpt.io/blog/how-to-use-the-notegpt-workplace>
17. <https://notegpt.io/blog/summarize-with-1-click>
18. <https://notegpt.io/blog/ai-tool-to-summarize-youtube-video>
19. <https://notegpt.io/blog/what-is-pdf-ai-chat-and-how-does-it-work>
20. <https://10web.io/ai-tools/notegpt/>
21. <https://notegpt.io/chatgpt/documents-summary>
22. <https://notegpt.io/blog/benefits-of-knowledge-sharing>
23. <https://www.aitechsuite.com/tools/notegpt.io>
24. <https://www.revoyant.com/compare/notegpt>
25. <https://notegpt.io/blog/no-results-found-on-transcript-on-youtube>
26. <https://notegpt.io/blog/gamma-ai-presentation>
27. <https://noiz.io/blog/best-notegpt-alternatives/>

28. <https://assignmentgpt.ai/blog/notegpt-review>
29. <https://notegpt.io/pricing>
30. <https://screenapp.io/alternatives/notegpt>
31. <https://notegpt.io/blog/black-friday-ai-deals>
32. <https://notegpt.tenereteam.com>
33. <https://notegpt.io/free-tools>
34. <https://www.g2.com/products/notegpt/competitors/alternatives>
35. https://www.reddit.com/r/NoteTaking/comments/1d2sutz/review_of_best_ai_summarizers/
36. <https://notegpt.io>
37. <https://notegpt.io/faq>
38. <https://notegpt.io/blog/introduction-paragraph-generator>
39. <https://www.meetjamie.ai/blog/notegpt>
40. <https://notegpt.io/blog/how-to-use-vimeo-summary>
41. https://www.reddit.com/r/learnprogramming/comments/1jao875/notegptio_explanation/
42. <https://notegpt.io/blog/note-library>
43. <https://notegpt.io/ai-notes-generator>
44. <https://notegpt.io/ai-music-generator>
45. <https://notegpt.io/privacy>
46. <https://chromewebstore.google.com/detail/notegpt-yt-summary-webpag/baecjmoceaobpnffgnlkloccenkoibbb>
47. <https://notegpt.io/blog/pdf-summarizer>
48. <https://notegpt.io/blog/transcript-not-showing-up-for-long-video-on-youtube>
49. <https://notegpt.io/chatgpt/reviews-summary>
50. <https://www.semrush.com/website/notegpt.io/competitors/>
51. <https://www.scamadviser.com/check-website/notegpt.io>

Answer from Perplexity: pplx.ai/share

Absolutely! Here's a deeper dive into additional areas to consider for your NoteGPT clone, focusing on **innovation**, **user experience**, **scalability**, and **long-term differentiation**. I'll break this down into actionable sections with suggestions, best practices, and "pro" ideas inspired by both NoteGPT.io and broader industry trends.

1. Innovation & Differentiation

A. Advanced AI Features

- **Multi-Model Support:** Let users choose between different LLMs (OpenAI, Claude, Gemini, Mistral, etc.) for summaries and chat. This can be a paid "Pro" feature.

- **Prompt Marketplace:** Allow users to share, rate, and import custom summary/chat prompts.
- **Interactive Summaries:** Enable users to ask follow-up questions or request clarifications on summaries, using conversational AI.

B. Content Types & Input Sources

- **Audio Uploads:** Accept MP3, WAV, M4A, etc. for transcription and summarization (not just YouTube).
- **Document Parsing:** Support DOCX, EPUB, HTML, and more, not just PDF.
- **Browser Integrations:** Besides Chrome, consider Firefox and Edge extensions.

C. Collaboration & Sharing

- **Team Workspaces:** Allow users to invite collaborators, share notes, and co-edit summaries/mind maps.
- **Public/Private Note Links:** Users can publish notes with privacy controls (public, unlisted, team-only).
- **Commenting & Reactions:** Add threaded comments or reactions on notes and summaries.

2. User Experience & Engagement

A. Onboarding & Guidance

- **Interactive Onboarding:** First-time user walkthroughs highlighting key features (e.g., “Try summarizing a YouTube video!”).
- **Feature Discovery:** Tooltips or “Did you know?” popups for advanced features.

B. Personalization

- **Custom Themes:** Let users pick color schemes or UI layouts.
- **Personal Dashboard:** Show recent activity, favorite notes, and quick actions.
- **Smart Recommendations:** Suggest videos, articles, or documents to summarize based on user interests.

C. Accessibility & Inclusivity

- **Voice Commands:** Allow users to control the app or add notes via speech.
- **Font & Contrast Options:** Support dyslexia-friendly fonts, high-contrast modes, and text resizing.

3. Performance, Scalability & Security

A. Scalability

- **Microservices Architecture:** Consider breaking out transcript, summary, and user management into separate services for easier scaling.
- **Queue/Job System:** Use background jobs (e.g., Celery, Sidekiq) for heavy tasks like transcription and summarization.

B. Security

- **OAuth 2.0 & SSO:** Support Google, Microsoft, and Apple sign-in for enterprise and education users.
- **Audit Logs:** For team and enterprise plans, provide logs of who accessed or edited what.
- **End-to-End Encryption:** For sensitive notes, offer optional encryption.

C. Reliability

- **Auto-Retry & Fallbacks:** If a summary fails (e.g., OpenAI API down), retry or use a backup model.
- **Monitoring & Alerts:** Set up error tracking (Sentry, Datadog) and uptime alerts.

4. Monetization & Growth

A. Pricing Models

- **Freemium with Usage Tiers:** Offer limited free summaries per month, then paid tiers for power users.
- **Pay-as-You-Go:** Credits for one-off summarizations or API calls.
- **Enterprise Licensing:** Custom plans for schools, businesses, or research groups.

B. Viral Growth Features

- **Referral Program:** Users get extra credits for inviting friends.
- **Social Sharing:** One-click sharing of notes/mind maps to Twitter, LinkedIn, etc.

C. Analytics for Users

- **Usage Dashboard:** Show users how many notes, summaries, and flashcards they've created.
- **Progress Tracking:** For students, track learning progress and flashcard mastery.

5. Community & Ecosystem

A. Community Building

- **User Forums:** Built-in or Discord/Reddit integration for Q&A and feature requests.

- **Feature Voting:** Let users vote on new features or bug fixes.
- **Events & Webinars:** Host live demos, Q&A sessions, or “how to get the most from NoteGPT” events.

B. Integrations

- **Learning Management Systems (LMS):** Integrate with Canvas, Moodle, Blackboard for education.
- **Productivity Tools:** Connect to Notion, Evernote, Google Drive, Dropbox for import/export.
- **Zapier/IFTTT:** Enable automation for power users.

6. Long-Term Vision

A. AI-First Knowledge Base

- **Semantic Search:** Let users search notes and summaries by meaning, not just keywords.
- **Knowledge Graph:** Visualize connections between notes, topics, and sources.
- **Auto-Organize:** Use AI to suggest folders/tags and group related notes.

B. Research & Citation Tools

- **Auto-Citation:** Generate citations for summarized content (APA, MLA, etc.).
- **Plagiarism Checker:** Scan notes/summaries for originality.

C. Mobile Experience

- **Mobile App:** Native iOS/Android or PWA with offline support.

- **Push Notifications:** Remind users to review flashcards or check new summaries.

7. Checklist for a Standout MVP

- Customizable AI prompts/templates
- Multi-language support (UI + content)
- Batch processing & cloud sync
- Mind map generation & export
- Chrome extension with feature parity
- Team/collaboration features
- Robust privacy & export controls
- API access for power users
- Community and feedback mechanisms
- Mobile-friendly design

Final Thoughts

Your current list is strong for a 1:1 clone, but these enhancements can help you:

- **Match NoteGPT.io's best features**
- **Outpace competitors with unique, sticky features**
- **Build a community and ecosystem for long-term success**

If you want a prioritized roadmap or technical architecture sample for any feature, let me know!

Answer from Perplexity: pplx.ai/share

16. AI Chat with PDF or Video

ChatGPT response to: “In my NoteGPT clone, I want to add soon the Ai Chat feature which will all the user to chat with openai based upon youtube video transcripts and PDF contents (2 different types based upon URL and detail/[id] page contents. Can you explain the best way to think about how to implement this and program this? For example, Should I include the content of any Summary pages? Should I include non-text PDF content?”

To build a powerful **AI Chat feature** for your NoteGPT clone (based on YouTube transcripts and PDF contents), here’s a **structured breakdown** of how to think about implementation and architecture:



Core Use Case

Enable users to **chat with AI** based on:

1. YouTube video transcripts (already parsed and segmented)
2. Uploaded PDF documents (parsed and enhanced)

Chat should be context-aware, fast, and scalable — ideally with tools like OpenAI **gpt-4-turbo** + retrieval + memory.



High-Level Strategy

1. Define Chat Context Scope

- Chat should only use:
 - YouTube transcript (enhanced or raw)
 - PDF extracted text (paragraphs, sections, metadata)
 - Optional: summary/flashcards (if well structured)
- Avoid:

- Non-textual PDF content (images, charts, scanned pages unless OCR'd)
- Full document history unless helpful (adds token bloat)

2. Store Sources Cleanly

- For each `videoId` or `pdfId`, store:
 - `full_transcript.json` or `full_pdf_text.json`
 - `enhanced_transcript.json` or `cleaned_pdf_text.json`
 - `summary.md` or `summary.json`
 - Optional: `metadata.json` with author, title, sections

Implementation Steps

1. Frontend: Trigger Chat Mode

- From `/detail/[id]` (for both videos and PDFs):
 - Show “Ask AI” or “Chat about this content”
 - On click: open chat UI and send content identifier (`type=pdf|yt + id`) to backend.

2. Backend: Prepare Context Window

A. Basic Strategy

- At first load, fetch relevant content (transcript or PDF text)

Create a `system` prompt like:

You are an AI assistant helping the user understand the following [video|document]. Refer to its content when answering.

-
- Inject full cleaned text or smartly chunked context (see below).

B. Optional Enhancements

- Summarize each section with OpenAI (`gpt-3.5-turbo` is cheap)
 - Use embeddings + vector search for retrieval-augmented generation (RAG) later
 - Store context chunks in a fast cache (e.g., Redis or file-based)
-

3. Chunking and Token Management

You need to stay under OpenAI token limits. So:

♦ For YouTube:

- Chunk by timestamp groups (~500 tokens per chunk)
- Include speaker attribution if useful
- Optionally summarize sections before chat

♦ For PDFs:

- Chunk by:
 - Page
 - Heading/Section titles
 - Paragraphs
 - Strip out footers, page numbers, image tags, etc.
-

4. Deciding What to Include

Content Type	Include?	Reason
Enhanced Transcript Text	✓	Primary chat material
Summary Page	✓ (optional)	Helps AI answer better but shouldn't replace raw context
PDF Extracted Text	✓	Use cleaned, readable sections
PDF Metadata	✓	Inject into prompt for more context (title, author, etc.)
PDF Images/Charts	✗*	Only include if you OCR or caption them; ignore for initial version

5. AI Chat Prompt Design (Initial MVP)

```
[
  {
    role: "system",
    content: "You are a helpful assistant answering questions about the following [video/document]. Refer to the content provided."
  },
  {
    role: "user",
    content: "What was the main argument in section 2?"
  },
  {
    role: "assistant",
    content: "Based on the transcript, section 2 covers..."
  }
]
```

Later, enhance with memory, citation links, or chunk-aware retrieval.

Testing Suggestions

- Pick 2-3 test documents/videos and run:

- "What is this about?"
- "Summarize section 2"
- "What did the speaker say about X?"
- "Who is the author and what is their main claim?"

Compare answers using:

- Transcript-only
 - Transcript + Summary
 - Transcript + Summary + Metadata
-

Next Steps

1. Create utility functions:
 - `getChatContext(id, type)` → returns cleaned content
 - `getSystemPrompt(type)` → returns dynamic system prompt
 2. Add frontend chat modal
 3. Set up backend `/api/chat` route
 4. Use `gpt-4-turbo` with streaming and context batching
 5. Add test cases to verify quality
-

Would you like a code scaffold for the `/api/chat` handler and frontend chat modal next?

16b. Ai Chat : Open Source PDF and Youtube video github repos

Perplexity question: is there a popular open source github repo that has a Chat Ai feature where the user can chat with PDF documents or youtube video transcripts or some other similar long content

There are several popular open-source GitHub repositories that enable users to chat with long-form content such as PDF documents and YouTube video transcripts using AI chat features. Here are some notable projects:

Chat with PDF Documents

- **pdfGPT**

This open-source project allows users to chat with the contents of their PDF files using GPT capabilities. It intelligently breaks down large documents, generates embeddings, and performs semantic search to provide relevant answers. Notably, it can cite the page number of the referenced information, improving credibility and ease of navigation. The repository is actively maintained and aims to support various LLMs and features like OCR and multiple PDF support^{[1](#)}.

- **AIAnytime/Chat-with-PDF-Chatbot**

This chatbot enables interactive conversations with PDF documents using a fully open-source stack-no OpenAI API required. It uses tools like Langchain for text processing, Sentence Transformers for embeddings, ChromaDB for vector storage, and a local language model for inference. The app is built with Streamlit for an easy-to-use web interface^{[43](#)}.

- **DocChatAI**

This project provides a customizable chatbot application that leverages large language models (LLMs) to process and interact with PDF files. It uses a retrieval-augmented generation (RAG) pipeline, storing document chunks in a vector database and retrieving relevant content to answer user queries[7](#).

- **AI PDF chatbot agent
(mayo00ear/ai-pdf-chatbot-langchain)**

A template for building AI chatbots that ingest PDF documents, store embeddings in a vector database, and enable chat-based retrieval and question answering[6](#).

Chat with YouTube Video Transcripts

- **YouTube Transcript Chatbot
(khaledsoudy-1/youtube-transcript-chatbot-langchain)**

This Streamlit web app uses OpenAI's language models and LangChain to allow users to ask questions about the content of any YouTube video. It fetches the transcript, splits it into manageable chunks, and provides context-aware responses to user queries[2](#).

General-Purpose Chat with Uploaded Files

- **Lobe Chat**

Lobe Chat is a modern, open-source chat UI/framework supporting file uploads (including documents, images, audio, and video) and knowledge base features. Users can upload various file types and interact with their content using integrated LLMs. It supports both local and cloud LLMs, multi-modal input, and knowledge base querying⁵.

Summary Table

Repository/Project	Content Type	Key Features	Open Source	LLM Requirement
pdfGPT	PDF	Chat with PDFs, semantic search, page citation	Yes	Optional (supports local and cloud LLMs)
Chat-with-PDF-Chatbot	PDF	Fully open-source, no API key needed, Streamlit UI	Yes	No (uses local models)

DocChatAI	PDF	RAG pipeline, vector DB, Streamlit UI	Yes	Yes
YouTube Transcript Chatbot	YouTube transcripts	Chat with YouTube video content, Streamlit UI	Yes	Yes (OpenAI API)
Lobe Chat	Multiple (PDF, images, audio, video)	File upload, knowledge base, multi-modal, plugins	Yes	Optional

These projects are widely used and actively maintained, making them strong choices for anyone looking to build or use an AI chatbot that can interact with long-form content like PDFs or YouTube transcripts¹²⁴⁵⁷.

Citations:

1. <https://github.com/bhaskatripathi/pdfGPT>
2. <https://github.com/khaledsoufy-1/youtube-transcript-chatbot-langchain>
3. <https://www.youtube.com/watch?v=N7wk1E1I0Qo>
4. <https://github.com/AIAnytime/Chat-with-PDF-Chatbot>
5. <https://github.com/lobehub/lobe-chat>
6. <https://github.com/mayooear/ai-pdf-chatbot-langchain>
7. <https://github.com/deepak7376/DocChatAI>
8. https://www.reddit.com/r/SideProject/comments/18alo8m/i_built_an_ai_assistant_to_chat_with_your_pdf_and/
9. https://www.reddit.com/r/SideProject/comments/1ecg9f0/ive_created_a_free_tool_for_extracting_youtube/
10. <https://discuss.streamlit.io/t/github-qa-chatbot-with-langchain/48118>
11. <https://github.com/sindresorhus/awesome-chatgpt>
12. <https://github.com/jacoblee93/fully-local-pdf-chatbot>
13. <https://github.com/Devanshu-17/TranscriptIQ>
14. <https://github.com/f/awesome-chatgpt-prompts>
15. https://www.reddit.com/r/OpenAI/comments/126sm2m/i_built_a_chatbot_that_lets_you_talk_to_any/
16. <https://discuss.huggingface.co/t/chat-with-a-pdf/53967>
17. <https://github.com/ganesh3/rag-youtube-assistant>
18. <https://github.com/sigoden/aichat>

19. https://www.reddit.com/r/MachineLearning/comments/129sqba/p_i_built_a_chatbot_that_lets_you_talk_to_any/
20. https://www.reddit.com/r/webdev/comments/1brogx1/feed_your_github_repo_as_text_to_ai_chat/

Answer from Perplexity: pplx.ai/share