ChatGPT

Notebook LM in Education: Vision and Open Notebook Implementation

The "Notebook LM" Educational Vision

NotebookLM represents an AI-first approach to learning and research – essentially a "thinking partner" that integrates large language models (LLMs) into your note-taking or study materials [1]. Instead of passively reading textbooks or documents, a NotebookLM system lets learners actively engage with their content: asking questions, generating summaries, creating study aids, and drawing connections, all grounded in the user's own notes and sources. Google's experimental NotebookLM (formerly "Project Tailwind") exemplifies this vision – it can analyze uploaded documents, turning complexity into clarity by synthesizing information and answering questions with direct references to the sources [2] [3]. The goal is to **transform passive reading into active learning**, for example by instantly generating flashcards, quizzes, or summaries from your materials [4] [5]. In short, the NotebookLM approach envisions an AI-enhanced notebook that helps students and researchers **understand complex material faster** and retain knowledge more effectively [6].

This vision emphasizes *personalization and interactivity*. The AI becomes a "cognitive partner" working with the learner's own notes, rather than a generic chatbot. Ideally, it provokes deeper thinking – for instance, by explaining a concept in simpler terms, answering follow-up questions, or even debating a point using the provided sources [7]. Importantly, a notebook LM is **document-grounded**: it bases its responses on the content you provide (lecture notes, articles, slides, etc.), which helps maintain accuracy and relevance. It's a natural fit for education because it can adapt to any subject matter the user is studying. Google's NotebookLM, for example, has features like a "Learning Guide" (an AI tutor for a given topic) and can pull in trusted academic content (e.g. OpenStax textbooks) to enhance understanding [5]. By integrating AI into the learning workflow, the notebook LM approach aims to turn a **note repository into an interactive learning environment**.

Open Notebook: An Open-Source NotebookLM Implementation

Open Notebook is a community-driven project that embodies this NotebookLM vision in an open-source, privacy-first manner. Described as *"a private, multi-model, 100% local, full-featured alternative to NotebookLM"* [8], Open Notebook was created to give learners and researchers a self-hosted AI partner for knowledge work. Its educational philosophy is clear: Open Notebook is *"the cognitive partner you always wanted"* – an assistant for researchers, students, and professionals to enhance their learning while **maintaining control over their data and workflows** [9]. Founder Luis Novo's motivation aligns with democratizing learning; he emphasizes that powerful AI study tools shouldn't be limited to big tech or require trading privacy for functionality [10] [11]. Open Notebook draws inspiration from learning science (citing Feynman's techniques and Zettelkasten note-taking) to encourage deep understanding through curiosity and connecting ideas [12].

Key goals and vision of Open Notebook include: (1) **Data privacy and ownership:** unlike Google's cloud-based service, Open Notebook keeps your notes and AI processing local – *"your data stays under your control*

— *no cloud dependencies*" [13]. (2) **Model flexibility:** it avoids vendor lock-in by working with many AI models/providers – over 16 are supported, from OpenAI and Anthropic to local LLMs via Ollama or LM Studio [14] [15]. (3) **Empowering learners:** it's designed for *learning enthusiasts* and *independent thinkers* who want an AI to help them go beyond superficial understanding [16] [17]. The system encourages users to form their own insights, with the AI as a customized tutor or challenger rather than just an answer bot [18] [19]. (4) **Workflow control:** users decide exactly how AI is used in their note-taking process – from choosing which models to use, to specifying what content the AI can see [20]. The overarching mission is to eventually provide *"a cognitive partner for every person"* – a personalized learning assistant that augments human knowledge without sacrificing autonomy [21].

System Architecture and Components

Open Notebook's architecture is a modern web application, split into a front-end UI and a back-end API, with an integrated database for storing content. The front-end is built with **Next.js/React** and provides the user interface (running on port 8502 in the default setup) [22]. The back-end is a **Python FastAPI** service (running on port 5055) that handles all AI requests, data processing, and integration logic [23]. Notably, the Next.js front-end proxies API calls to the FastAPI server, so they work together seamlessly [24].

A **SurrealDB** database instance is bundled as the data store (running on port 8000) [25]. SurrealDB is used to keep track of user notebooks, documents (sources), notes, and possibly vector embeddings for search. It's **auto-configured** in the default Docker deployment, so users don't have to manage the DB separately [25]. SurrealDB also supports advanced queries, which Open Notebook leverages for features like full-text search across all content [26]. For AI operations, Open Notebook integrates the popular **LangChain** framework [27]. LangChain is used under the hood to orchestrate tasks like document text splitting, embedding generation, and constructing prompts for the LLM, which simplifies building the retrieval-augmented generation workflow (RAG) [27].

Multi-model support is a standout architectural feature. Open Notebook uses an adapter library called *Esperanto* to interface with a wide range of AI providers [28]. Thanks to this, the user can plug in different LLM backends just by providing the appropriate API keys or endpoints. Out of the box it supports "16+ providers including OpenAI, Anthropic, Ollama, LM Studio, and more" [15]. For example, you can use OpenAI's GPT-4 or GPT-3.5 (with your API key) for high-quality answers, or switch to **Ollama** to run a local model on your machine for free [14]. Google's PaLM via Vertex AI, Anthropic's Claude, and other services are also supported [29]. Many providers can also handle embeddings for document similarity search (e.g. OpenAI, Google) or specialized tasks like speech-to-text and text-to-speech in some cases [29]. This modular design (using providers that are *"OpenAI-compatible"* or specific integrations like ElevenLabs for TTS) gives the user freedom to choose or even self-host the AI model [30] [31]. In practice, Open Notebook's back-end will route requests – such as generating an answer or embedding a document – to whichever model provider is configured, then return the result to the front-end.

Deployment and openness: Technically, Open Notebook is delivered as a Dockerized application for easy setup. The project provides a one-command Docker run or a docker-compose recipe to launch everything (frontend, backend, DB) on a local machine or server [32] [33]. This makes it straightforward for users to self-host. Since it's open-source (MIT licensed), advanced users can also run it in development mode (`make start-all`) to customize the code [34]. An important part of the architecture is the presence of a **REST API** that mirrors all functionality of the UI [35]. This means Open Notebook can be integrated into other tools or scripted – aligning with its extensibility goals (more on that later). The architecture summary in the

documentation highlights these points: Python + React app, with SurrealDB for storage and LangChain to tie AI components together [27]. In essence, Open Notebook's technical design prioritizes **flexibility** (multi-provider AI), **privacy** (local runtime, user-controlled data), and **extensibility** (open API and source).

Features and Workflow

On the user-facing side, Open Notebook provides a rich set of features that map to an effective learning workflow. When you open a notebook in the interface, it is conceptually divided into three main sections: **Sources, Notes, and Chat** [36].

- **Sources:** This is where you add and manage your reference materials. Open Notebook supports a broad range of content: you can import PDFs and Word documents, add text notes or copy-pasted excerpts, include PowerPoint slides, and even integrate multimedia sources like YouTube videos or web pages [37] [38]. For web links and YouTube, Open Notebook will fetch the content (e.g. scrape the article text or retrieve the video transcript) so that it becomes part of your notebook's knowledge base. This **multi-format support** acknowledges that "learning doesn't happen in one format, and neither should your notes" [38]. (Audio/video file support is mentioned as coming soon, and YouTube is already handled via transcripts [39].) All source content is indexed for search and can be used by the AI when answering questions. You have fine control here: each notebook is a container of sources on a particular topic or project, keeping unrelated content separate.

- **Notes:** Open Notebook lets you create your own notes and also generate AI-assisted notes. You can type out your understanding or ideas as you normally would in a note-taking app. But uniquely, you can also ask the AI to **summarize a source, extract key points, or even draft an explanation** of a concept for you [40]. This is often done via "Transformations" – custom or built-in actions that process content (for example, summarizing a lengthy PDF or converting a raw text into a structured outline) [41]. The notes area is a space where human and AI-generated content can coexist. For instance, after adding a research paper as a source, you might click a "Summarize" transformation to generate an outline of that paper, which you can then edit or annotate further. Open Notebook even supports *"recursive summarization"* [42] – breaking down large documents into hierarchical summaries – to deal with information overload in a structured way. All notes you create (AI or manual) are saved in the notebook and can later be reviewed or searched.

- **Chat (Assistant):** This is the interactive Q&A interface, analogous to chatting with an AI tutor. The chat is **context-aware**, meaning you can ask questions and the AI will answer by drawing from the sources in your notebook [26] [43]. For example, if you have uploaded several articles about a topic, you can ask the assistant to compare their viewpoints or clarify a term mentioned in them. Because the system uses retrieval augmented generation, it will pull relevant source snippets (via vector similarity search and full-text search) and include them in the prompt to the LLM [26]. The answer you get is thus grounded in your content, often with **citations linking back to the original sources** [44]. Open Notebook places a strong emphasis on citing sources for transparency and to maintain research integrity (citing sources comprehensively rather than just giving a generic answer) [44]. The chat interface also isn't limited to Q&A; you can ask the AI to generate things like a study guide or brainstorm ideas based on the notebook content, effectively leveraging the AI as a collaborator.

A powerful aspect across these features is **fine-grained context control**. Open Notebook recognizes that sometimes you want the AI to have access to all your sources for a broad question, but other times you

might want to restrict it to avoid irrelevant context or reduce token usage. It provides *"3 granular levels"* of context sharing [45]. In practice, this means you can toggle how much of your notebook's content the AI assistant sees: e.g., *"no context"* (treat it like a generic model, for a purely conceptual question), *"partial context"* (perhaps share just the currently open source or a subset of notes), or *"full context"* (allow the assistant to draw on everything in the notebook) [45]. Google's NotebookLM had an all-or-nothing approach (either the AI sees the whole notebook or nothing), whereas Open Notebook lets the user decide per query or note what is exposed [45]. This not only protects privacy for sensitive notes but also helps manage the cost (you might not always want to send an entire library to the model for a simple question). The UI provides controls for this, so the user is always aware of what the AI can access [46] [43].

Another marquee feature is the **Podcast Generator**. Open Notebook can transform notes or selected content into an audio podcast, with support for multiple AI-generated voices and speakers [47] [48]. For example, you could take your summarized notes or an article and have the system produce an "episode" where two AI voices discuss the material. This is presented as a way to review or learn in a more engaging format (auditory learning). Technically, it uses text-to-speech models (the project integrates with providers like ElevenLabs for very realistic voices, or even open-source TTS models) [31] [49]. Open Notebook allows 1–4 speakers with custom voice profiles, whereas Google's NotebookLM (with its "Audio Overviews") only had a basic 2-speaker setup [50]. This flexibility means you can customize the tone (e.g. one speaker could take the role of a tutor, the other a student asking questions) and create a more dynamic podcast. It's a novel way to "listen" to your notes on the go, or reinforce learning by hearing material explained in conversation form.

Behind the scenes, Open Notebook implements intelligent **search** features to support all of the above. It indexes all added content for **full-text search** as well as generates vector embeddings for **semantic search** [26]. This means you can quickly find where a term was mentioned in your sources, or ask the assistant a question and know it will retrieve semantically related chunks even if exact keywords differ. The search is unified in the UI – a student could, for instance, search their entire notebook for "Hegel dialectic" and find every highlight or note about it, or rely on the assistant to bring up the most relevant source excerpts when discussing that concept. This combination of search and chat is key to making the notebook's content accessible and useful.

Finally, Open Notebook exposes all core functions via a **RESTful API** (and even offers a pre-built ChatGPT plugin to guide installation) [51]. This means the platform is not just a monolithic app; users or developers can programmatically add documents, query the assistant, fetch notes, etc. Such extensibility is valuable in educational settings – for example, a school could integrate Open Notebook's API with a Learning Management System (LMS) to automatically populate a student's notebook with course materials, or a power-user could write scripts to ingest new papers daily and summarize them. The open-source nature ensures that if a feature is missing, the community can build it.

Strengths and Current Capabilities

Open Notebook's capabilities are already robust for a 1.x release, and it has several strengths that differentiate it in the NotebookLM space:

- **Privacy and Local Control:** Perhaps the biggest strength is its privacy-first design. All data stays on the user's machine or chosen server when self-hosted. There is no requirement to send your personal notes to a cloud service [13]. Even the AI processing can be done locally if you use local

models. For anyone uneasy with Big Tech reviewing their study materials or research notes, this is a critical advantage. As XDA Developers put it, one can replace Google's NotebookLM with Open Notebook *"because I don't want Google having more of my data."* With Open Notebook, **the user has complete data sovereignty** – you decide where your notebook lives and who/what can access it [52].

- **Flexibility in AI Models and Cost Control:** Open Notebook supports an impressive range of model providers and allows mixing and matching. This means you can optimize for quality, speed, or cost as needed. For instance, use a free local LLM for quick brainstorming, but an OpenAI model for generating a high-quality summary. The ability to switch providers (or use multiple) is a stark contrast to closed systems that tie you to one model [50]. It also future-proofs the platform – new open-source LLMs or services can be added via the provider interface. From a cost perspective, Open Notebook is transparent: it doesn't charge subscription fees; you *"pay only for AI usage"*, i.e. the API calls or the compute you run [53]. This can be far cheaper than proprietary services. You could even run entirely free by sticking to local models (Ollama with Llama2, etc.) [14]. This flexibility is a huge benefit for students or researchers on a budget, or institutions that want to avoid recurring costs.

- **Comprehensive Document Handling:** Open Notebook is very capable at ingesting and processing diverse content. It supports PDFs, text, Office documents, web pages, and YouTube out of the box [38]. It can fetch content from URLs and has built-in converters (for example, PDF to text, PPT to text outline, etc.). This eliminates a lot of manual prep work – you don't have to copy-paste everything. It also performs *document transformations* like splitting into chunks and summarizing, which lets it handle large documents by summarizing them recursively [54]. By integrating external tools (e.g., for web scraping or YouTube transcription when needed), it ensures your knowledge base is as rich as your learning resources. One can effectively build a personalized digital library and ask questions across all of it. This broad content integration meets learners where they are – whether you're studying a PDF textbook, an online article, or a lecture video, the system can work with it.

- **Interactive and Supportive UX:** Users have noted that Open Notebook's interface feels familiar to those who tried Google's NotebookLM [55]. There's a clear workflow: upload sources, ask questions, and take notes. It doesn't require programming knowledge to use; the UI provides buttons for common actions like "Summarize this document" or "Extract key points." The design encourages actively working with notes, not just storing them. For example, as one reviewer described, Open Notebook can "suggest connections, highlight themes, or even generate new ways of looking at the problem" from your notes [56]. This means the AI might proactively surface links between ideas in different sources or prompt you with questions, emulating the behavior of a good tutor. Such features help users get more out of their existing notes (turning "a dusty library of information" into a dynamic dialogue) [57]. The inclusion of creative outputs – like being able to turn notes into a podcast or a structured summary – also sets it apart as a tool not just for consumption but for content creation and review [40].

- **Citations and Integrity:** Unlike some chatbots, Open Notebook consistently provides citations for AI-generated content, and it can produce thorough references. This is a strength for academic or professional use, where tracing the source of information is important. The documentation even highlights *"comprehensive [citations] with sources"* as a feature, versus Google's basic references [44]. In practice, when you ask a question in the Chat, the assistant's answer will include footnote-style

links to your documents where the information came from. This builds trust and allows you to verify answers – an important aspect when using AI in education.

- **Openness and Extensibility:** Being open source, Open Notebook invites community contributions and customization. Users can create plugins or add new "transformation" scripts for custom analyses of content. For example, one could implement a transformation to generate quiz questions from a text – something Google's NotebookLM does in its closed system – and share it with others. The availability of a full REST API means integrations into other systems or automation is straightforward [35]. Also, advanced users can inspect how prompts are constructed, how data is processed, etc., for full transparency. This openness not only means the tool can improve rapidly with community help, but also that power-users can adapt it to niche workflows (imagine a medical researcher adding a custom module to parse and summarize clinical trial data, for instance). The absence of a black-box is a strength for those who want to **see exactly how their content is being handled** and trust the tool [58].

- **Current Maturity:** As of late 2025, Open Notebook has reached a 1.0+ version and has a growing user community (over 9k stars on GitHub) [59]. It has implemented most of the core features envisioned in its roadmap: multi-notebook management, multi-model integration, transformations, search, AI notes, context control, and podcast generation. The team has also been responsive in adding features; for instance, multi-provider support was extended (to Anthropic, Vertex, etc.) early on [60], and they continue to incorporate feedback (the project's update logs show rapid enhancements). This active development and community support are a strength, as the tool is quickly evolving toward the needs of its users.

Limitations and Challenges

Despite its strengths, Open Notebook is not without limitations. Being aware of these helps set the right expectations:

- **LLM Limitations (Hallucinations, Accuracy):** Open Notebook inherits the typical limitations of large language models. The AI can sometimes **"overreach" or produce *hallucinations*** – plausible-sounding but incorrect statements not supported by the sources [61]. For example, if your documents don't contain an answer, the model might still attempt one, which could be wrong. The use of retrieval and citations mitigates this (since you can check sources), but it doesn't eliminate the issue entirely. Users must still apply critical thinking and verify important outputs. This is a limitation for any AI-driven tool in education: it's a powerful assistant, but not an infallible oracle.

- **Learning Curve and Workflow Adjustment:** Adopting Open Notebook effectively may require a shift in how one studies or takes notes. As one reviewer noted, it takes time to get used to treating your notes as *"a living, interactive system instead of a static archive"* [61]. Students and researchers must develop new habits, like regularly consulting the AI, updating notes based on AI suggestions, or chunking information for optimal AI interaction. There is a bit of complexity in managing context settings and understanding what the AI can or cannot see. For less tech-savvy users, the initial setup (running a Docker container, configuring API keys) is another hurdle, though well-documented. In short, while the UI is user-friendly, getting the most out of Open Notebook requires **active engagement and some experimentation**, which might be daunting for those expecting a plug-and-play magic solution.

- **Technical Resources and Performance:** Running Open Notebook, especially with local models, can be resource-intensive. The web UI and database are lightweight, but if you choose to run a large LLM on your machine (via Ollama or Local GPU), you need sufficient hardware (CPU/GPU and memory). Alternatively, using cloud APIs (OpenAI, etc.) requires internet and incurs cost. Google's NotebookLM offloads all AI processing to Google's servers (which are very powerful), whereas Open Notebook's performance depends on what infrastructure *you* have. In practice, using GPT-4 via API will yield excellent results but costs money, whereas using a small local model is free but may produce lower-quality answers. This flexibility is a double-edged sword: novices might find it confusing to decide which model to use when. Also, indexing very large amounts of data can lead to heavy storage use (SurrealDB stores embeddings and text, which could be gigabytes if you added a huge library). These are not issues in small-scale usage, but it's a consideration for power users (e.g., someone attempting to load thousands of documents may need to plan for database size and memory accordingly).

- **Feature Gaps vs. Proprietary Tools:** Although Open Notebook covers core needs, it currently lacks some of the polished features that emerging proprietary systems have. For example, Google's NotebookLM recently introduced one-click flashcard and quiz generation from notes [62] [63], and Notion AI can do things like automatic summaries inside a note-taking app. Open Notebook doesn't (yet) have a dedicated flashcard generator or a mind-map style visualization of concepts. Its focus has been on the foundational capabilities (which it does provide well). Some of these advanced study tools (quizzing, visual knowledge graphs, etc.) would need to be manually created or are still on the roadmap. That said, because Open Notebook is open and scriptable, the community could implement these given time – but as of now, a user might need to use the general chat to *ask* for flashcards or summaries, rather than pressing a specialized button for it.

- **Collaboration and Multi-User Support:** Open Notebook is primarily a single-user application at this stage. Notebooks are not multi-tenant – there isn't a built-in way to have multiple people working on the same notebook in real-time or sharing an AI session (aside from manually sharing the web UI over a network). In contrast, a tool like SurfSense (discussed below) focuses on team knowledge bases with real-time collaboration. The Open Notebook team has expressed interest in shared knowledge spaces, and third parties like NoodleFlow have experimented with shareable notebooks/apps on top of Open Notebook [64], but the feature is not native or seamless yet. So, for a classroom or group project scenario, Open Notebook might not (yet) replace a collaborative platform. It's best suited to individual use, with collaboration possible by exporting notes or sharing content after the fact.

- **Stability and Maintenance:** As with any fast-evolving open source project, there can be bugs or breaking changes. In fact, the jump to version 1.0 introduced significant changes (switching from Streamlit to Next.js frontend, etc.), requiring users to migrate data and adjust configurations [65]. New users won't feel this, but early adopters had to occasionally troubleshoot. The documentation provides a migration guide and a troubleshooting guide [66] [67], and there's an active Discord for support. This is simply the nature of a project under heavy development – but it means an educator deploying Open Notebook should be prepared to update it periodically and manage those updates. The flip side is that it's improving rapidly.

In summary, Open Notebook already delivers on the core promise of the NotebookLM approach: it **empowers self-directed learners with an AI that can organize, analyze, and converse about their**

notes. Its privacy, flexibility, and rich feature set are strong advantages. However, users should be mindful of the inherent AI limitations (like hallucinations), the need for an active learning approach, and some features still catching up to their proprietary counterparts. With those caveats, Open Notebook stands out as a **powerful open-source tool that transforms note-taking into an interactive, intelligent process** – effectively bringing the NotebookLM vision to anyone willing to run it.

Comparative Analysis: Open Notebook and Alternative Tools

Several other tools and projects share a similar goal of integrating LLMs into notebook-based or document-grounded workflows for learning. Some are open-source like Open Notebook, while others are commercial or experimental. Below we compare Open Notebook with a few notable alternatives across key dimensions (model integration, document handling, user experience, openness, and extensibility):

Tool / Project	Model Integration	Document Handling	UX / Workflow	Openness	Extensibility
Open Notebook (open-source)	**Multi-model:** 16+ providers (OpenAI, Anthropic, Google Vertex, Ollama for local, etc.). Users can choose or mix models; supports embeddings, some speech-to-text and TTS via providers [15] [29].	**Broad multi-format:** Import PDFs, Word, PPT, TXT, URLs (web pages), YouTube transcripts, etc. [38]. Full-text and semantic search over all content [26]. Can apply transformations (summaries, extractions) to documents. No built-in image understanding (images treated as files).	**Notebook-style UI:** Web app with sections for Sources, Notes, and Chat [36]. Users manage a notebook per project. AI can be interacted with via chat or used to generate notes. Offers fine-grained context control for AI [46]. Emphasizes active note-taking (AI suggests connections, summaries) and even podcast generation from notes. Primarily single-user interface (one user's notebook at a time).	**Open Source:** Yes (MIT) – code is on GitHub [68]. Self-hosted (Docker or from source). No cloud account required; data stays local [13].	**Highly extensible:** Full REST API available [35]. Users can modify code or add custom "transformation" workflows. Plugin-like provider system for new models. Community contributions encouraged; no proprietary constraints.

Google NotebookLM (closed beta)	**Single-model (Google):** Uses Google's own PaLM/ Gemini models. No option for other providers – model improvements tied to Google's updates [50] . (Likely very powerful models, but closed).	**Google ecosystem:** Accepts Google Docs and possibly PDFs/ other file types via Google Drive. Integrates with Google Drive for document selection. Can handle text-based media; also has features like Video Overviews and Audio Overviews for YouTube content [69] [7] . Document limit may apply (was limited in beta).	**AI-augmented notebook in Google Labs:** Web interface similar to Google Docs with an AI sidebar. Users create "notebooks" by adding source documents. Strong focus on study tools: one-click generation of flashcards/ quizzes from sources [62] , auto-generated reports (briefings, study guides, blog drafts) [70] , and a "Learning Guide" mode for Q&A tutoring [71] . Polished UI with easy share (links to public notebooks). However, all AI context is essentially all-or-nothing (no granular privacy per note) [45] . Multi-user collaboration not emphasized (notebooks are individual, though you could share the Google Docs themselves).	**Proprietary:** Closed source, available to limited users via Google Labs. Runs on Google Cloud – your data (notes and prompts) are processed by Google's servers [52] . No self-host option.	**Limited extensibility:** No public API for NotebookLM as of 2025. Users can't modify the AI's behavior beyond what the interface allows. Integrations are basically whatever Google provides (since it's tied into Google Drive, but you can't plug in external models or extend it).

SurfSense (open-source)	**Multi-LLM via API hub:** Integrates with "100+ leading LLMs" on demand [72]. Likely leverages an orchestration backend to route queries to different models (e.g. OpenAI, Azure, local models via API). Aimed at flexibility similar to Open Notebook. Also connects to AI services for search and agents.	**Connected knowledge sources:** Designed to aggregate many sources: Notion pages, Slack chats, GitHub repos, Google Drive, emails (Gmail), Confluence, etc. [73]. Performs **hybrid search** across all connected sources (combines vector and keyword search) [74]. Also supports multimedia content: can generate podcasts from documents, similar to Open Notebook [74]. Essentially serves as a unified company or personal knowledge base.	**Workspace UI with chat and docs:** Web interface where you can chat with an AI about your connected data. Emphasizes real-time **collaboration**: multiple users can see live edits, presence, and comment within chats/ docs [74]. Good for teams – you might ask questions and get answers that reference, say, a company Slack thread plus a Notion page. It's less of a traditional "notebook" and more of an **integrated dashboard**. Offers contextual commenting (you can discuss a specific document via chat). Probably a steeper learning curve due to many integrations, but powerful for knowledge management.	**Open Source:** Yes (Apache-2.0 on GitHub [75]). Self-hostable. Data remains self-controlled when self-hosted, though it connects to many external APIs for data sources (so privacy depends on those integrations).	**Extensible/ Integrations:** Highly extensible by design – it has plugins/ connectors for many services, and you can add more. A "visual canvas" or similar might exist for building custom workflows (if integrated with NoodleFlow). Because it's open source, developers can add connectors or use its API. SurfSense aims to be *"highly customizable"* and already provides many third-party hooks (search engines, SaaS apps) [76]. This makes it a sort of platform for AI research assistants.

AnythingLLM (open-source) 📚	**Model agnostic (API-based):** By default uses OpenAI API for both embedding and chat (so GPT-3.5/4, etc.) [77] [78]. Encourages use of "off-the-shelf" tools like OpenAI + Pinecone for simplicity [79]. However, it supports local vector DB (Chroma) and plans to allow other LLMs beyond OpenAI [80]. Currently, users bring their own OpenAI key, but future versions may enable local model integration.	**Document Q&A focus:** Can ingest many document types – PDF, TXT, CSV, HTML, DOCX, PPTX, etc. – by dropping them into a watched folder [81]. It provides scripts to bulk import content from YouTube channels (transcripts), Substack blogs, websites, GitBook docs, etc. [82]. Automatically chunks and embeds documents; uses a vector store (Pinecone or Chroma) to enable semantic search [77]. Supports multiple **workspaces**, each a collection of docs with its own vector index [83] [84]. Content updates trigger re-embedding (and it has caching to avoid re-embedding unchanged docs) [85] [86].	**User-centric chat UI:** Web interface with a simple chat screen. Users create a "workspace" (like a notebook) and add documents to it; then ask questions in a chat box. The interface is clean but relatively basic – the AI just answers questions with cited references. There's no separate note-taking area or multi-step workflows in the UI (it's not an outline or notebook editor). The focus is on quickly "chatting with your documents" [88] [89]. It's designed to be very easy to set up and use, even for non-developers. No real multi-user features (each instance is used by one person typically), though you could share a workspace by sharing the web access.	**Open Source:** Yes (MIT). The code (full-stack JS/ Python) is on GitHub [90]. It's meant to run locally ("local by default") – uses local SQLite for data and can use a local Chroma DB for vectors [77]. All data stays on your machine unless you use external APIs (OpenAI/ Pinecone). This makes it a popular choice for privacy-conscious users who want a quick doc-QA solution.	**Extensibility:** Moderate. It has a well-defined architecture: a Node.js frontend and Python backend. It's open, so one could modify it, and it's built in a generic way (e.g. "workspaces" concept) that could integrate with other systems. It doesn't yet have a plugin system or API out-of-the-box for third-party integration – it's more an end-user application. However, its code can be adapted by developers. In future, the project hinted at adding more integrations (Google Drive, GitHub, etc.) [80], which would extend its utility. Currently, customization might involve editing the source (for example, to swap out the LLM or vector DB beyond

Tool / Project	Model Integration	Document Handling	UX / Workflow	Openness	Extensibility
		Citations are provided in answers, including links for external sources [87].			what's supported).

(Table Legend: = designed for personal note-taking; = proprietary/closed; = team/collaboration focus; 📚 = document chatbot focus.)

Discussion of Alternatives

Google NotebookLM: This is the benchmark vision from which others draw inspiration. Its strengths are in its tailored educational features (flashcards, learning guides, etc.) and deep integration with Google's ecosystem and advanced models. However, it's not open or flexible – users must trust Google with their data and are limited to Google's AI capabilities. In contrast, Open Notebook trades off some of that polish and plug-and-play ease for complete control and extensibility [52] [91]. For an individual learner who already lives in Google Docs and needs quick AI help summarizing those, NotebookLM (if available to them) might be convenient. But many others will prefer Open Notebook's self-hosted privacy and the ability to choose cheaper or offline models.

SurfSense: Compared to Open Notebook, SurfSense is like an enterprise-grade knowledge hub. Both support multimedia and even podcast generation, but SurfSense shines in collaborative and integrative features – it's meant to tie together a team's disparate knowledge stores and let an AI agent help navigate them [73]. It has a broader scope (e.g., connecting to Slack, Jira, etc.), essentially blending NotebookLM-like Q&A with something like an internal company search engine. SurfSense's UI revolves around chat and real-time content sync, whereas Open Notebook's UI is more static in terms of user-created notes and one-user focus. If you're an individual or student, Open Notebook's structured approach might feel more coherent for learning (you curate a notebook of relevant sources). If you're in a company or research lab and want a **shared AI assistant** that knows all project documents, SurfSense is very appealing. Both are open-source and self-hostable, indicating a general movement toward private AI research assistants. Interestingly, they both independently implemented podcast creation from text – indicating educational use cases like listening to content are valued in this space [74] [48].

AnythingLLM: This project is more specialized: it focuses on being the easiest way to "chat with your documents." It overlaps with part of Open Notebook's functionality (Q&A on a set of documents with citations). AnythingLLM is praised for being lightweight and quick to set up, even on modest hardware [79] [92]. It also introduced useful features like caching embeddings to save costs [85]. However, it lacks the richer note-taking environment Open Notebook provides. There's no concept of AI-generated *notes* or a section to write your own commentary – it's essentially an AI document assistant in a chatbox. For someone who only wants a personal chatbot that knows their PDFs, AnythingLLM is a great choice (and indeed it brands itself as "no code or infra headache" solution) [93]. But for a student who wants to both ask questions *and* maintain a structured notebook with highlights, outlines, and mixed AI/human notes, Open Notebook is more appropriate. AnythingLLM could be seen as a subset: you could replicate a lot of its

function within Open Notebook's chat, but Open Notebook offers more on top (organization, editing, transformations). Both are open and local-first, addressing the privacy concern.

Other Tools: There are other noteworthy entrants in this space. **AFFiNE**, for example, is an open-source knowledge base (a "KnowledgeOS") that has integrated some AI features for content creation. While AFFiNE allows writing and graph-style organization and even boasts an edgeless whiteboard, its AI usage is more about generating content or completing text, not specifically Q&A on user-provided documents as NotebookLM does [94] [95]. **Obsidian** (with community plugins) and **Logseq** are popular local note-taking apps that now have GPT-3 powered plugins to query your vault or summarize notes. These can be seen as lightweight alternatives: they keep data local (notes in Markdown) and you can use GPT to get insights, but they require manual setup and are not as feature-complete as Open Notebook's integrated system [96] [40]. On the commercial side, **Notion AI** and **Mem** offer AI assistants that work on your notes, but they are cloud-based and not focused on academic uses (and certainly not open-source). **Microsoft's Copilot** is also entering tools like OneNote, which will allow Q&A and summarization of your OneNote content – another parallel to NotebookLM, but tied to Microsoft's ecosystem and proprietary.

In the table above, we highlighted how each compares to Open Notebook. In summary, Open Notebook strikes a middle ground: it's personal (not multi-user like SurfSense), but far more feature-rich for note-taking than a bare-bones doc chatbot like AnythingLLM. Its open-source nature sets it apart from big tech offerings. No other open tool currently matches its combination of **multi-modal support, multi-model AI integration, and notebook-style workflow**. As the space evolves, we may see these projects converge in capabilities, but for now each has its niche. A student or researcher concerned with privacy and wanting a deep learning companion will find Open Notebook a compelling choice. Others who need team collaboration might lean towards SurfSense, or those wanting extreme simplicity might try a tool like AnythingLLM or smaller plugins.

Conclusion

The "notebook LM" approach is redefining how we interact with knowledge. By embedding LLM capabilities into our notebooks and study documents, it offers a path to transform learning from a passive consumption of information into an active, dialogue-driven process. Open Notebook exemplifies this paradigm: it takes the promise of Google's NotebookLM – an AI that can *"help you make connections and understand complex material faster"* [1] – and delivers it in an open, extensible platform that users can control. Technically, Open Notebook's architecture shows how such a system can be built with modern web frameworks, a vector-aware database, and integration layers for various AI models. Educationally, it stays true to the vision of a personal cognitive partner, whether by summarizing a dense article, prompting you with questions on your notes, or generating a podcast so you can review concepts on a walk.

In evaluating Open Notebook, we find a tool that is **powerful but still evolving**. Its strengths in privacy, flexibility, and feature set make it a frontrunner among open-source educational AI assistants. It empowers learners to grapple with information in new ways – highlighting that the true potential of note-taking is unlocked when notes are not just static records, but interactive springboards for insight [56]. At the same time, users must remain mindful of the AI's limitations and be willing to guide the tool with their own judgment and curiosity. As the comparisons illustrate, Open Notebook is part of a broader landscape of solutions aiming to bring AI into our personal knowledge workflows. Each solution balances convenience, openness, and educational focus differently.

Ultimately, the emergence of projects like Open Notebook signals a future where **learners have AI copilots** that adapt to their needs: open-source alternatives ensuring this capability is not confined to big companies. For anyone excited by the idea of a smarter notebook – one that can answer your questions, quiz you, and keep your information organized – Open Notebook is an exciting development. It stands as a testament to how quickly the community can advance the notebook LM vision, turning what was recently an experimental Google product into a practical reality that you can run on your own PC. The journey to truly personalized AI learning assistants has only just begun, but with Open Notebook and similar projects, the path is wide open and being paved by educators and developers together. 9 8

---

1 2 3 4 5 6 7 62 63 69 70 71 6 NotebookLM features to help students learn
https://blog.google/technology/google-labs/notebooklm-student-features/

8 15 22 23 24 25 26 27 28 29 30 31 32 33 34 35 44 45 48 49 50 51 52 53 65 66 67 91 lfnovo/open-notebook | zdoc.app
https://www.zdoc.app/en/lfnovo/open-notebook

9 16 17 18 19 20 21 47 What is Open Notebook? | Open Notebook
https://www.open-notebook.ai/

10 11 12 36 37 39 41 42 43 46 54 60 Home · lfnovo/open-notebook Wiki · GitHub
https://github.com/lfnovo/open-notebook/wiki

13 14 Open Notebook: An Open-Source, Privacy-First Notebook AI for Research & Note-Taking | by Dr. Shouke Wei | Oct, 2025 | Medium
https://medium.com/@shouke.wei/open-notebook-an-open-source-privacy-first-notebook-ai-for-research-note-taking-5cee01b1b9c0

38 40 56 57 61 Open Notebook Review: AI-Powered, Privacy-First Note-Taking
https://thecodebeast.com/open-notebook-the-ai-powered-privacy-first-tool-that-transforms-note-taking/

55 58 64 Open Notebook – Open source NotebookLM alternative
https://www.noodleflow.ai/notebook

59 GitHub - lfnovo/open-notebook: An Open Source implementation of Notebook LM with more flexibility and features
https://github.com/lfnovo/open-notebook

68 Open Notebook - Open Source Alternative to Google NotebookLM - Software - DAIHUM: Digital & AI Humanities
https://ai-humanities.com/t/open-notebook-open-source-alternative-to-google-notebooklm/1813

72 73 74 2 Best Open Source NotebookLM Alternatives (2025)
https://openalternative.co/alternatives/notebooklm

75 76 MODSetter/SurfSense: Open Source Alternative to ... - GitHub
https://github.com/MODSetter/SurfSense

77 78 79 80 81 82 83 84 85 86 87 88 89 90 92 Introducing AnythingLLM - an OSS for document chatting | Mintplex
https://medium.com/rampp/anythingllm-the-easiest-way-to-chat-with-documents-in-seconds-337ea08d4b7e

93 What is AnythingLLM
https://docs.anythingllm.com/introduction

94 95 96 2025 Top NotebookLM Alternatives to Elevate Your Research | AFFiNE
https://affine.pro/blog/notebooklm-alternatives