Obsidian & Notion: Complete Feature Specification for Web-Based Knowledge Platform Clone

Purpose: This document serves as a comprehensive specification for a coding agent to build a web-based knowledge platform combining the best features of Obsidian and Notion. The platform will serve as a knowledge base for an AI agent, with a web interface for human viewing. Future integrations with ChemELN and exPTube (proprietary SciSymbioAI platforms) are planned.

Date: February 2026

TABLE OF CONTENTS

---

1. OBSIDIAN — Complete Feature Deep Dive

Official Site: https://obsidian.md
Documentation: https://help.obsidian.md
GitHub (Releases/Plugins): https://github.com/obsidianmd/obsidian-releases
Wikipedia: https://en.wikipedia.org/wiki/Obsidian_(software)

1.1 Core Philosophy & Architecture

- **Local-first**: All notes stored as plain-text Markdown files on the user's device
- **Vault-based**: A "vault" is a folder on disk containing all notes, settings, and plugins

- **Privacy-focused**: No mandatory cloud; no account required for basic use
- **Plugin-driven architecture**: Most features (even core ones) are implemented as plugins that can be toggled on/off
- **Offline-first**: Fully functional without internet
- **Cross-platform**: Windows, macOS, Linux, iOS, Android (NOT available as a web app natively)
- **Free for personal use**; commercial license $50/user/year

1.2 Core Editor

Markdown Support

Obsidian uses extended Markdown with the following supported syntax:

Element	Syntax	Notes
Headings	`# H1` through `###### H6`	Standard ATX headings
Bold	`**text**` or `__text__`	
Italic	`*text*` or `_text_`	
Strikethrough	`~~text~~`	
Highlight	`==text==`	Non-standard extension
Inline code	`` `code` ``	
Code blocks	```` ```language ``` ````	Syntax highlighting for 100+ languages
Blockquotes	`> text`	Nestable
Unordered lists	`- item` or `* item`	
Ordered lists	`1. item`	
Task lists / Checkboxes	`- [ ] task` / `- [x] done`	
Tables	`	col1
Horizontal rules	`---` or `***`	

Links	`[text](url)`	Standard markdown links
Images	`![alt](path)`	Local and remote images
Footnotes	`[^1]` and `[^1]: text`	
Math/LaTeX	`$inline$` and `$$block$$`	MathJax/KaTeX rendering
Mermaid diagrams	`` ```mermaid ``` ``	Flowcharts, sequence diagrams, Gantt, etc.
Callouts	`> [!note]`, `> [!warning]`, etc.	Styled admonition blocks with icons
Comments	`%% comment %%`	Hidden from reading view
Embeds	`![[note]]` or `![[image.png]]`	Transclusion of other notes/media

Editor Modes

1. **Source Mode**: Raw markdown text editing (plain text)
2. **Live Preview Mode** (default): Hybrid — renders markdown inline while editing; clicking on rendered elements switches to raw markdown for that element
3. **Reading Mode**: Fully rendered, read-only view of the note

Editor Features

- Vim keybindings support (optional)
- Foldable headings and indented sections
- Line numbers (optional)
- Spell check
- Auto-pair brackets and quotes
- Smart indent for lists
- Tab/Shift-Tab for indent/outdent

1.3 Vault & File System

- **Vault** = a folder on disk with a hidden `.obsidian/` config directory
- **File formats**: `.md` (notes), `.canvas` (canvas files), plus any attachments (images, PDFs, audio, etc.)
- **Folder structure**: User-defined; no enforced hierarchy
- **`.obsidian/` folder contains**:
  - `app.json` — app settings
  - `appearance.json` — theme settings
  - `core-plugins.json` — which core plugins are enabled
  - `community-plugins.json` — installed community plugins
  - `hotkeys.json` — custom keyboard shortcuts
  - `plugins/` — community plugin code
  - `themes/` — installed themes
  - `snippets/` — custom CSS snippets
  - `workspace.json` — layout state (panes, tabs, sidebar)
- **Multiple vaults**: Users can create and switch between multiple independent vaults
- **Supported attachment types**: Images (png, jpg, gif, svg, bmp, webp), Audio (mp3, wav, m4a, ogg, 3gp, flac, webm), Video (mp4, webm, ogv, mov, mkv), PDF

1.4 Linking & Knowledge Graph

Internal Linking

- **Wikilinks**: `[[Note Name]]` — links to another note by name
- **Wikilinks with alias**: `[[Note Name|Display Text]]`
- **Heading links**: `[[Note Name#Heading]]`
- **Block links**: `[[Note Name#^block-id]]` — link to a specific paragraph/block
- **Markdown links**: `[Display Text](Note%20Name.md)` — standard markdown style (optional)
- **Automatic link updating**: When a note is renamed, all links to it are automatically updated

Backlinks

- **Backlinks pane**: Shows all notes that link TO the current note
- **Unlinked mentions**: Detects text that matches a note name but isn't linked — offers to convert to link
- **Outgoing links pane**: Shows all notes linked FROM the current note

Graph View

- **Global Graph**: Visualizes ALL notes in the vault as a network graph
  - Nodes = notes (sized by number of connections)
  - Edges = links between notes
  - Color coding by folders, tags, or search queries
  - Filtering: by tags, folders, search terms; show/hide orphans, attachments
  - Forces: repel strength, link force, link distance adjustable
  - Animation: real-time physics simulation
  - Zoom, pan, click to navigate to note
- **Local Graph**: Shows only notes connected to the currently open note (within N degrees of separation)
- **Graph settings**: Adjustable display for arrows, text fade threshold, node size, line thickness

1.5 Canvas

- **Infinite 2D canvas** for visually arranging content
- **File type**: `.canvas` (JSON-based)
- **Card types**:
  - **Note cards**: Embed existing vault notes (content renders inline)
  - **Text cards**: Standalone text blocks (not saved as separate notes)
  - **Media cards**: Images, videos, PDFs
  - **Web page cards**: Embed URLs
- **Connections**: Draw arrows/lines between cards

    - Line types: straight, curved
    - Labels on connections
    - Directional arrows
- **Color coding**: Cards and connections can be color-coded
- **Groups**: Select multiple cards and group them
- **Zoom levels**: Content renders differently at different zoom levels (text appears when zoomed in)
- **Use cases**: Mind maps, mood boards, project planning, dashboards

1.6 Search & Navigation

Search

- **Full-text search** across all notes in vault
- **Search operators**:
    - `file:name` — search by filename
    - `path:folder` — search within specific folder
    - `tag:#tagname` — search by tag
    - `line:text` — match entire line
    - `section:text` — match within a heading section
    - `content:text` — match within note content only
    - `match-case:text` — case-sensitive
    - `[property:value]` — search by property/frontmatter
    - Boolean: `OR`, grouping with `()`, negation with `-`
- **Regex search** supported
- **Search results**: Shows matched lines with context; click to navigate
- **Embedded search**: `query` code block embeds live search results inside a note

Navigation

- **Quick Switcher** (`Ctrl/Cmd+O`): Fuzzy search to jump to any note instantly
- **Command Palette** (`Ctrl/Cmd+P`): Search and execute any command
- **File Explorer**: Tree view of vault folders in left sidebar
- **Breadcrumbs**: Path navigation showing current note location
- **Recent files**: List of recently opened notes
- **Starred/Bookmarked**: Pin frequently used notes
- **Tab system**: Multiple notes open as tabs (like a browser)
- **Split panes**: Horizontal and vertical splits for side-by-side editing
- **Pop-out windows**: Detach panes into separate OS windows
- **Hover previews**: Hovering over a link shows a preview popup of the linked note

1.7 Tags & Metadata

Tags

- **Inline tags**: `#tagname` anywhere in note body
- **Nested tags**: `#parent/child/grandchild`
- **Frontmatter tags**: `tags: [tag1, tag2]` in YAML
- **Tag pane**: Lists all tags with counts; click to filter

Properties (Frontmatter / YAML)

- Structured metadata at the top of notes using YAML frontmatter:

```
---
title: My Note
date: 2025-01-15
tags: [project, research]
status: draft
author: Martin
aliases: [alt-name]
cssclasses: [custom-class]
---
```

- **Built-in property types**: Text, List, Number, Checkbox, Date, Date & Time

- **Custom properties**: Any key-value pair
- **Aliases**: Alternative names for a note (can link using any alias)
- **CSS classes**: Apply custom CSS to specific notes
- **Properties view**: Dedicated pane showing all properties across vault
- **Property type enforcement**: Set expected types for properties vault-wide

1.8 Bases (Core Plugin — New in 2025)

- **Database-like views** of vault notes (similar to Notion databases but built on local markdown files)
- Creates `.base` files
- **View types**:
  - **Table view**: Spreadsheet-style view of notes with properties as columns
  - **Card/Gallery view**: Visual card layout
  - **List view**: Hierarchical bulleted list
- **Features**:
  - Filter notes by properties, tags, folders
  - Sort by any property
  - Group By: Categorize notes by folder, tags, creation date, file type, or any property
  - Table Summaries: Calculate averages, max, min, median, ranges across property values
  - Customizable columns showing note properties
  - Inline editing of properties directly in the table
- **Data source**: Pulls from existing vault notes — no separate database; everything is still markdown files with YAML frontmatter

1.9 Templates

Core Templates Plugin

- Designate a templates folder

- Insert template content into new notes
- Template variables:
  - `{{title}}` — note title
  - `{{date}}` — current date
  - `{{time}}` — current time
  - Date/time format tokens (Moment.js format strings)

Templater (Community Plugin — very popular)

- JavaScript execution in templates
- System commands
- User-defined functions
- Cursor placement with `<% tp.file.cursor() %>`
- Dynamic dates, file manipulation, and more
- Auto-apply templates on note creation based on folder

1.10 Core Plugins (Complete List)

All built-in plugins that ship with Obsidian (can be toggled on/off):

Plugin	Description
Audio Recorder	Record audio using system microphone; embeds into current note
Backlinks	Show backlinks and unlinked mentions for the active note
Bases	Database-like views of vault notes (table, card, list views)
Bookmarks	Bookmark notes, headings, searches, and graph views
Canvas	Infinite canvas for visual note arrangement
Command Palette	Search and execute commands via `Cmd/Ctrl+P`
Daily Notes	Create a new note for each day; template support
File Explorer	Browse vault files and folders in sidebar
File Recovery	Snapshot-based backup; roll back recent changes

Format Converter	Convert files from various formats to Obsidian markdown
Global Search	Full-text search across the entire vault
Graph View	Interactive network graph of note connections
Importer	Import from Evernote, Notion, Apple Notes, Google Keep, etc.
Note Composer	Merge two notes or split one note into two
Outgoing Links	Show all links from the active note
Outline	Table of contents for the active note (heading tree)
Page Preview	Hover preview of linked notes
Properties View	List and manage all properties in vault
Publish	Publish notes as a website (paid service)
Quick Switcher	Fuzzy search to open any note instantly
Random Note	Open a random note
Search	Find files in vault
Slash Commands	`/` menu for inserting content types
Slides	Create presentations from notes (horizontal rules = slide breaks)
Sync	End-to-end encrypted sync across devices (paid)
Tags View	List all tags in vault with counts
Templates	Insert pre-defined content templates
Unique Note Creator	Create note with timestamp-based unique name
Web Viewer	Open external links as tabs within Obsidian
Word Count	Display word and character count in status bar

1.11 Community Plugin Ecosystem

- **2,000+ community plugins** available
- **Plugin API**: JavaScript/TypeScript-based; plugins can access vault files, editor, UI,

settings

- **Plugin architecture**: Each plugin is a folder with `manifest.json`, `main.js`, and optional `styles.css`
- **Installation**: Via built-in Community Plugins browser or manually from GitHub
- **BRAT**: Beta Reviewers Auto-update Tool for testing unreleased plugins

Most Popular/Important Community Plugins

Plugin	Downloads	Description
Dataview	1M+	Query notes as a database using SQL-like syntax (DQL) or JavaScript; create dynamic tables, lists, and task views
Templater	800K+	Advanced template engine with JavaScript execution
Calendar	700K+	Calendar view for daily notes
Kanban	500K+	Kanban boards stored as markdown
Excalidraw	400K+	Whiteboard/drawing tool integrated into Obsidian
Advanced Tables	400K+	Improved table editing with tab navigation and formatting
Tasks	350K+	Advanced task management with due dates, recurrence, filtering
Periodic Notes	300K+	Weekly, monthly, quarterly, yearly notes
Obsidian Git	300K+	Version control using Git
Style Settings	250K+	GUI for adjusting theme CSS variables
Linter	200K+	Auto-format notes based on configurable rules
Smart Connections	150K+	AI-powered note recommendations using embeddings
Mind Map	150K+	Visualize notes as mind maps
Database Folder	100K+	Notion-like database views (predecessor to Bases)

Advanced Canvas	100K+	Extra canvas features (shapes, connection types)
Tag Wrangler	100K+	Rename, merge, and manage tags
Callout Manager	80K+	Create custom callout types with icons and colors
Web Clipper	New	Browser extension to save web content to vault

1.12 Themes & Appearance

- **Built-in themes**: Default light and dark themes
- **Community themes**: 200+ custom themes installable via Settings
- **CSS customization**:
  - **Themes**: Complete CSS overrides in `themes/` folder
  - **CSS Snippets**: Small CSS files in `snippets/` folder for targeted customizations
  - CSS variables for colors, fonts, spacing
- **Settings**:
  - Font family, font size
  - Line width (max width of content area)
  - Light/Dark mode toggle (+ per-theme toggle)
  - Translucent window (macOS)
  - Native menus
  - Accent color picker

1.13 Obsidian Publish (Paid: $8/month/site)

- Publish selected notes as a public or password-protected website
- Custom domain support
- Navigation: sidebar with folder tree
- Graph view on published site

- Search on published site
- Theme customization via `publish.css`
- Selective publishing (choose which notes to publish)
- Password protection for entire site or specific pages

1.14 Obsidian Sync (Paid: $4-8/month)

- End-to-end encrypted sync across devices
- Version history (up to 12 months)
- Up to 10 remote vaults
- Up to 10GB per vault (Plus plan)
- Selective sync (choose folders)
- Activity log showing changes by collaborators
- Shared vaults for team collaboration

1.15 UI/UX Layout (Desktop)

```
┌──────────────────────────────────────────────────────────────┐
│ [Ribbon]  │  [Left Sidebar]  │    [Main Editor Area]    │ [Right Sidebar] │
│           │                  │                          │                 │
│ (icons)   │  File Explorer   │  ┌─Tab 1─┬─Tab 2─┬─...─┐│  Backlinks      │
│ - New     │  ─or─            │  │                      ││  Outgoing Links │
│ - Search  │  Search Results  │  │   Note Content       ││  Tags           │
│ - Graph   │  ─or─            │  │   (Markdown)         ││  Outline (TOC)  │
│ - Canvas  │  Tag View        │  │                      ││  Properties     │
│ - etc.    │  ─or─            │  │                      ││  Graph (local)  │
│           │  Bookmarks       │  │                      ││                 │
│           │                  │  └──────────────────────┘│                 │
│           │                  │                          │                 │
├───────────┴──────────────────┴──────────────────────────┴─────────────────┤
│ [Status Bar] — word count, line count, encoding, reading time            │
└──────────────────────────────────────────────────────────────────────────┘
```

- **Ribbon** (leftmost): Vertical icon strip for quick actions
- **Left Sidebar**: Collapsible; houses File Explorer, Search, Bookmarks, Tag View
- **Main Editor Area**: Tabbed interface; supports split panes (horizontal/vertical)

- **Right Sidebar**: Collapsible; houses Backlinks, Outline, Properties, Local Graph
- **Status Bar**: Bottom bar with word count, character count, etc.
- **Command Palette**: Floating modal activated by `Cmd/Ctrl+P`
- **Quick Switcher**: Floating modal activated by `Cmd/Ctrl+O`
- **Settings Modal**: Full-screen modal with sidebar navigation

1.16 Keyboard Shortcuts (Key Examples)

Action	Shortcut
Open Command Palette	`Cmd/Ctrl + P`
Quick Switcher	`Cmd/Ctrl + O`
Search	`Cmd/Ctrl + Shift + F`
New note	`Cmd/Ctrl + N`
Toggle edit/preview	`Cmd/Ctrl + E`
Bold	`Cmd/Ctrl + B`
Italic	`Cmd/Ctrl + I`
Insert link	`Cmd/Ctrl + K`
Toggle left sidebar	`Cmd/Ctrl + Shift + L`
Toggle right sidebar	`Cmd/Ctrl + Shift + R`
Close current tab	`Cmd/Ctrl + W`
Graph view	`Cmd/Ctrl + G`
All shortcuts fully customizable	via Settings → Hotkeys

---

2. NOTION — Complete Feature Deep Dive

Official Site: https://www.notion.com
Documentation: https://www.notion.com/help

API Documentation: https://developers.notion.com
Community: https://www.notion.com/community

2.1 Core Philosophy & Architecture

- **Cloud-first**: All data stored on Notion's servers
- **Block-based editor**: Everything is a "block" — paragraphs, headings, images, databases, embeds
- **All-in-one workspace**: Notes + databases + wikis + project management
- **Real-time collaboration**: Multiple users can edit simultaneously
- **Cross-platform**: Web app, desktop (Windows, macOS), mobile (iOS, Android)
- **Freemium model**: Free tier with limitations; paid plans (Plus $10/month, Business $18/month, Enterprise custom)

2.2 Block-Based Editor — Complete Block Type List

Every piece of content in Notion is a "block." Blocks can be nested, reordered via drag-and-drop, and converted between types.

Basic Blocks

Block Type	Description	Details
Text (Paragraph)	Default block; plain text	Rich text formatting: bold, italic, underline, strikethrough, code, color, highlight, link
Heading 1	Large heading	Toggleable (can collapse content beneath)
Heading 2	Medium heading	Toggleable
Heading 3	Small heading	Toggleable
Bulleted List	Unordered list	Nestable via Tab indent
Numbered List	Ordered list	Auto-increments; nestable
To-do List	Checkbox items	Click to toggle complete/incomplete
Toggle List	Collapsible	Click arrow to expand/collapse; can contain any blocks

	section	inside
Quote	Blockquote	Styled with left border
Divider	Horizontal rule	Visual separator
Callout	Highlighted box with icon	Customizable icon (emoji) and background color
Link to Page	Internal page link	Creates navigable reference to another page

Media Blocks

Block Type	Description	Details
Image	Embedded image	Upload, URL, or Unsplash integration; resize, caption, link
Video	Embedded video	Upload or embed from YouTube, Vimeo, etc.
Audio	Audio player	Upload audio file or embed
File	File attachment	Upload any file type; download link
Web Bookmark	Link preview card	Shows title, description, favicon, thumbnail
Code	Code block	Syntax highlighting for 80+ languages; copy button; line wrapping
PDF	Embedded PDF viewer	Scrollable inline viewer

Database Blocks

Block Type	Description	Details
Table (Full Page)	Full-page database as table	
Table (Inline)	Database embedded within a page	
Board	Kanban board view	

Gallery	Card/grid view	
List	Minimal list view	
Calendar	Calendar view	
Timeline	Gantt-chart view	
Chart	Data visualization	Bar (vertical/horizontal), Line, Donut
Simple Table	Static table (not a database)	Rows and columns without database features

Embed Blocks

Block Type	Description
Embed	Generic iframe embed for any URL
Google Drive	Embed Google Docs, Sheets, Slides
Google Maps	Embed map view
Figma	Embed Figma designs
GitHub Gist	Embed code snippets
Twitter/X	Embed tweets
CodePen	Embed pens
Loom	Embed recordings
Miro	Embed whiteboards
Abstract	Embed designs
Framer	Embed prototypes
Whimsical	Embed diagrams
500+ more	Via generic embed or integrations

Advanced Blocks

Block Type	Description	

		Details
Math Equation	LaTeX/KaTeX math rendering	Inline and block-level
Button	Trigger automations	Create pages, edit properties, insert blocks, send notifications
Table of Contents	Auto-generated from headings	Links to each heading on the page
Breadcrumb	Shows page hierarchy	Clickable path navigation
Synced Block	Content synced across pages	Edit once, reflects everywhere
Column Layout	Multi-column layout	2-5 columns; drag blocks between columns
Template Button (legacy)	Insert predefined block structure	Replaced by Button block

Block Operations (applicable to all blocks)

- **Drag and drop**: Reorder any block by dragging the `⋮⋮` handle
- **Turn into**: Convert any block to another type (e.g., text → heading, bulleted list → to-do)
- **Duplicate**: Copy a block
- **Delete**: Remove a block
- **Move to**: Move block to another page
- **Comment**: Add a comment on any block
- **Color/Background**: Change text color or background color (10 colors each)
- **Copy link**: Get a direct link to a specific block
- **Slash command** (`/`): Type `/` to see all available block types to insert

2.3 Page System

- **Pages are blocks**: Every page is itself a block that can be nested inside other pages
- **Page hierarchy**: Unlimited nesting depth (pages within pages)
- **Page icon**: Emoji or uploaded image icon for each page

- **Page cover**: Banner image at top of page (upload, Unsplash, gradient, solid color)
- **Page properties**: When inside a database, pages have typed properties (see Databases)
- **Full-width toggle**: Expand content to fill screen width
- **Small text toggle**: Reduce text size
- **Font options**: Default, Serif, Mono
- **Lock page**: Prevent accidental edits
- **Page history**: Version history with timestamps; restore previous versions
  - Free plan: 7 days
  - Plus plan: 30 days
  - Business plan: 90 days
- **Favorites**: Pin pages to sidebar favorites section
- **Breadcrumb navigation**: Shows full page path at top

2.4 Databases — FULL DEEP DIVE

Databases are Notion's most powerful feature. Every row in a database is a full Notion page.

Database Views

View Type	Description	Best For
Table	Spreadsheet-style rows and columns	Data-heavy content, bulk editing
Board (Kanban)	Cards organized in columns by a property	Project management, workflows, pipelines
Calendar	Month/week view based on date property	Scheduling, content calendars, deadlines
Gallery	Grid of cards with cover images	Visual content, directories, portfolios
List	Minimal text-based list	Quick task lists, simple overviews
Timeline	Horizontal timeline/Gantt chart	Project timelines, roadmaps

(Gantt)		
Chart	Data visualization (bar, line, donut)	Analytics, reporting, dashboards

- **Multiple views per database**: Create unlimited views of the same data
- **Linked database views**: Reference the same database from different pages with independent views, filters, and sorts
- **View-specific settings**: Each view has its own filters, sorts, groups, visible properties, and layout options

Database Property Types (Complete List)

Basic Properties:

Property Type	Data Type	Description
Title	String	Required; primary identifier; every database has exactly one
Text	String	Plain text field; supports rich text
Number	Number	Integers or decimals; format as number, percentage, currency, progress bar
Select	Enum	Single choice from predefined options with colors
Multi-select	Enum[]	Multiple choices from predefined options with colors
Status	Enum (special)	Three groups: To Do, In Progress, Complete; customizable option names
Date	Date/DateTime	Date picker with optional end date (range) and time
Person	User reference	Assign workspace members
Files & Media	File[]	Upload files or add links to media
Checkbox	Boolean	True/false toggle
URL	String (URL)	Clickable URL field

Email	String (email)	Clickable email field
Phone	String (phone)	Clickable phone number

Advanced Properties:

Property Type	**Description**
Formula	Computed value using Notion’s formula language (supports if/else, math, string ops, date functions, lookups across relations)
Relation	Links to pages in another database (or self-referencing); one-way or two-way (bidirectional)
Rollup	Aggregates data from related database entries via a Relation property; functions: Count, Sum, Average, Min, Max, Range, Show original, etc.
Created time	Auto-recorded creation timestamp
Created by	Auto-recorded creating user
Last edited time	Auto-updated modification timestamp
Last edited by	Auto-updated last editor
Unique ID	Auto-incrementing ID with optional prefix (e.g., TASK-001)
Button	Triggers an automation action per row

AI-Powered Properties (Notion AI):

- **AI Summary**: Auto-generate summary of page content
- **AI Custom Autofill**: Use custom prompts to auto-fill properties based on page content
- **AI Key Info Extraction**: Extract entities, dates, etc. from page content

Database Operations

- **Filter**: Filter rows by any property using operators (is, is not, contains, does not contain, is empty, is not empty, before/after for dates, etc.); multiple filters with AND/OR logic
- **Sort**: Sort by any property ascending/descending; multiple sort levels
- **Group**: Group rows by any property value; collapsible groups
- **Sub-items**: Hierarchical parent-child relationships between database pages
- **Dependencies**: Link tasks with predecessor relationships (for Timeline view)
- **Calculations**: Column summary calculations (Count, Sum, Average, Min, Max, Percent empty, etc.) shown at bottom of table columns
- **Database templates**: Pre-configured page layouts that auto-apply when creating new database entries
- **Database lock**: Prevent structural changes while allowing data entry

2.5 Formulas (Formula Language)

Notion has a comprehensive formula language, sometimes called "Formula 2.0":

- **Data types**: String, Number, Boolean, Date, List, Person, Page
- **Operators**: `+`, `-`, `*`, `/`, `%`, `==`, `!=`, `>`, `<`, `>=`, `<=`, `and`, `or`, `not`
- **Conditional**: `if(condition, true_value, false_value)`, ternary `X ? Y : Z`, `ifs()` for multi-condition
- **Math functions**: `abs`, `ceil`, `floor`, `round`, `min`, `max`, `sqrt`, `pow`, `ln`, `log10`, `log2`, `exp`, `sign`, `pi`, `e`
- **String functions**: `length`, `contains`, `replace`, `replaceAll`, `lower`, `upper`, `trim`, `test` (regex), `match` (regex), `format`, `substring`, `split`, `join`, `repeat`, `padStart`, `padEnd`
- **Date functions**: `now`, `today`, `dateAdd`, `dateSubtract`, `dateBetween`, `dateStart`, `dateEnd`, `dateRange`, `timestamp`, `fromTimestamp`, `formatDate`, `parseDate`, `minute`, `hour`, `day`, `month`, `year`
- **List functions**: `map`, `filter`, `find`, `findIndex`, `some`, `every`, `flat`, `reverse`, `sort`, `unique`, `slice`, `concat`, `at`, `first`, `last`, `sum`, `count`, `min`, `max`, `average`, `median`, `includes`, `indexOf`
- **Property access**: Reference any property by name; dot notation for relation properties
- **Style functions**: `style()` for conditional formatting (colors, bold, italic, strikethrough,

code, underline)

- **Notion AI assistance**: AI can help write, debug, and explain formulas

2.6 Relations & Rollups

Relations:

- Link entries between two databases (or self-referencing within one database)
- **One-way relation**: Only visible in the source database
- **Two-way (bidirectional) relation**: Creates a matching relation property in the target database automatically
- Click to see related pages; select existing or create new related pages

Rollups:

- Aggregate data from a related database through a Relation property
- Select which relation to roll up and which property from the related database
- Aggregation functions: Count all, Count values, Count unique, Count empty, Count not empty, Sum, Average, Median, Min, Max, Range, Show original, Earliest date, Latest date, Date range, Show unique, Percent empty, Percent not empty, Percent checked, Percent unchecked

2.7 Automations

Database automations with trigger → action pattern:

Triggers:

- Page added to database
- Property value changes (any property type)
- Trigger scope: all views or specific view

Actions:

- Edit property values on the trigger page
- Edit pages in another database
- Create new page in any database
- Send Notion notification

- Send Slack message (requires Slack integration)
- Send email via Gmail
- Send webhook (HTTP POST)
- Define custom variables (using formulas and mentions)
- Insert blocks into a page

Advanced:

- Multiple actions per automation
- Formula-based variables for dynamic values
- Chain actions together
- Conditional logic via formula variables

2.8 Notion AI

- **AI writing assistant**: Summarize, explain, translate, improve writing, fix grammar, make longer/shorter, change tone, continue writing
- **AI Q&A**: Ask questions about workspace content; AI searches across all pages
- **AI autofill**: Auto-fill database properties using AI based on page content
- **Custom AI prompts**: Define custom AI actions
- **AI in formulas**: AI assistance for writing and debugging formulas
- **Pricing**: Included in paid plans (was previously separate add-on)

2.9 Comments & Collaboration

Comments:

- **Inline comments**: Highlight text and add a comment
- **Page-level comments**: Comment on the page itself (shown in discussions)
- **Resolved comments**: Mark comments as resolved (can view resolved)
- **@mentions**: Mention users in comments and page content → triggers notification

Collaboration:

- **Real-time co-editing**: See cursors and changes of other users live

- **Presence indicators**: Avatars of active viewers/editors
- **Sharing permissions**:
  - Full access (edit structure + content)
  - Can edit (edit content only)
  - Can comment
  - Can view
- **Share to web**: Make any page publicly accessible via URL
- **Guest access**: Invite external users (limited in free plan)
- **Teamspaces**: Organized team sections within a workspace
- **Groups**: User groups for bulk permission management

2.10 Notion Sites (Web Publishing)

- Publish pages as websites with custom domains
- Custom navbar, footer, and logo
- SEO settings (meta title, description, social preview image)
- Google Analytics integration
- Custom fonts and themes
- Sitemap generation
- Responsive design
- Password protection
- Search engine indexing control

2.11 Templates

- **Page templates**: Pre-built page structures from Notion's template gallery (thousands available)
- **Database templates**: Default page layouts for new database entries; multiple templates per database
- **Button block templates**: Insert predefined block structures on click
- **Template gallery**: Community-contributed templates at notion.com/templates

2.12 Integrations & API

Official Integrations:

- Slack, Google Drive, GitHub, Figma, Jira, Linear, Trello, Asana, Zapier, Make, GitLab, Typeform, and many more

Notion API (Public REST API):

- **Base URL**: `https://api.notion.com/v1/`
- **Authentication**: Bearer token (internal integrations) or OAuth (public integrations)
- **Endpoints**:
    - Pages: Create, retrieve, update, archive
    - Databases: Create, query (with filters/sorts), update schema
    - Blocks: Retrieve, append, update, delete children
    - Users: List, retrieve
    - Search: Full-text search across workspace
    - Comments: Create, retrieve
- **Webhooks**: Available for automation triggers
- **Rate limits**: 3 requests/second per integration
- **SDKs**: Official JavaScript SDK (`@notionhq/client`); community SDKs for Python, Go, Ruby, etc.
- **MCP (Model Context Protocol)**: Connect workspace to AI tools like ChatGPT, Claude, Cursor

Import/Export:

- **Import from**: Evernote, Trello, Asana, Google Docs, Word, Dropbox Paper, Quip, Confluence, HTML, CSV, Markdown
- **Export as**: Markdown, CSV, HTML, PDF

2.13 Search

- **Quick Search** (`Cmd/Ctrl+K`): Fuzzy search across all pages, databases, and content
- **Filters**: By date (last week, last month, created/edited by me), by teamspace, by page type

- **Recently visited**: Quick access to recent pages
- **On-page search** (`Cmd/Ctrl+F`): Find text within current page, including inside toggles and database properties
- **AI-powered Q&A**: Ask natural language questions about workspace content

2.14 Web Clipper

- Browser extension for Chrome, Firefox, Safari
- Clip web pages as Notion pages
- Highlights capture
- Choose destination page/database
- Auto-extract metadata (title, URL, description, tags)

2.15 UI/UX Layout

```
┌─────────────────────────────┬─────────────────────────────────────────────┐
│  [Sidebar]                  │  [Main Content Area]                        │
│                             │                                             │
│  🔍 Search                  │  ← → Breadcrumbs: Page > Sub > Current      │
│  🏠 Home                    │  ┌────────────────────────────────────────┐ │
│  📥 Inbox                   │  │  [Page Icon] [Page Title]              │ │
│                             │  │  [Page Cover Image]                    │ │
│  ⭐ Favorites               │  │                                        │ │
│  ├─ Page A                  │  │  Property 1: value                     │ │
│  └─ Page B                  │  │  Property 2: value                     │ │
│                             │  │                                        │ │
│  🏢 Teamspace 1             │  │  --- Page Content ---                  │ │
│  ├─ Wiki                    │  │  Text blocks, databases, media,        │ │
│  ├─ Projects DB             │  │  embeds, etc.                          │ │
│  └─ Meeting Notes           │  │                                        │ │
│                             │  │                                        │ │
│  👤 Private                 │  │                                        │ │
│  ├─ Personal Notes          │  │                                        │ │
│  └─ Journal                 │  └────────────────────────────────────────┘ │
│                             │                                             │
│  📄 Shared                  │  [Side Peek Panel — opens linked pages      │
│  ├─ Team Docs               │   in a side panel instead of navigating]    │
│                             │                                             │
│  🗑️ Trash                   │                                             │
│  ⚙️ Settings                │                                             │
└─────────────────────────────┴─────────────────────────────────────────────┘
```

Sidebar:

- Collapsible (hover to reveal in collapsed state)
- Sections: Search, Home (feed), Inbox (notifications), Favorites, Teamspaces, Private, Shared, Trash
- Drag-and-drop reorder pages
- Inline page creation (`+` button)
- Expand/collapse page trees

Top Bar:

- Breadcrumb navigation
- Share button
- Comments toggle
- Favorite toggle (star)
- More menu (●●●): page history, export, move, delete, lock, etc.
- Clock icon: updates/activity
- Tab previews (hover to see page preview)

Editor Area:

- Max width content (default) or full-width toggle
- Page icon area (click to change)
- Cover image area (hover "Add cover")
- Properties section (collapsible)
- Content body (block-based)
- `+` button on hover left of blocks for adding new blocks
- `⋮⋮` handle on hover left of blocks for drag-and-drop
- `/` slash command for inserting blocks

Side Peek:

- Open linked pages in a panel on the right without leaving current page
- Useful for viewing/editing related database entries

2.16 Keyboard Shortcuts (Key Examples)

Action	Shortcut
Search	`Cmd/Ctrl + K`
New page	`Cmd/Ctrl + N`
Bold	`Cmd/Ctrl + B`
Italic	`Cmd/Ctrl + I`
Underline	`Cmd/Ctrl + U`
Strikethrough	`Cmd/Ctrl + Shift + S`
Code inline	`Cmd/Ctrl + E`
Link	`Cmd/Ctrl + K` (with text selected)
Heading 1	`/h1` or `Cmd/Ctrl + Shift + 1`
Toggle block	`/toggle`
To-do	`/todo` or `Cmd/Ctrl + Shift + 4`
Bulleted list	`/bullet` or `Cmd/Ctrl + Shift + 5`
Numbered list	`/number` or `Cmd/Ctrl + Shift + 6`
Mention	`@`
Page search	`Cmd/Ctrl + P`
Toggle sidebar	`Cmd/Ctrl + \`
Indent	`Tab`
Outdent	`Shift + Tab`
Move block up/down	`Cmd/Ctrl + Shift + ↑/↓`
Find on page	`Cmd/Ctrl + F`
Duplicate block	`Cmd/Ctrl + D`
Turn into (block conversion)	`/turn` or select block → Turn into

Color text	`/color` or `Cmd/Ctrl + Shift + H`

3. Comparison & Architectural Patterns

3.1 Feature Comparison Matrix

Feature	Obsidian	Notion
Data Storage	Local files (Markdown)	Cloud (proprietary)
Editor Type	Markdown-based	Block-based WYSIWYG
Offline Support	Full	Limited (cached)
Real-time Collaboration	Limited (via Sync)	Full real-time co-editing
Databases	Bases (new, basic) + Dataview (plugin)	Native, powerful, mature
Relations/Rollups	Via Dataview plugin	Native
Graph View	Native, interactive	Not available
Backlinks	Native, robust	Limited (via @ mentions)
Wiki-style Links	`[[wikilinks]]` native	Link to Page block
Canvas/Whiteboard	Native Canvas	Not native (use embeds)
Templates	Core + Templater plugin	Native + template gallery
API	Plugin API (local)	REST API (cloud)
Plugin Ecosystem	2000+ community plugins	Limited integrations
Formulas	Via Dataview (JS queries)	Native formula language
Automation	Via plugins (e.g., Templater)	Native database automations
Web Publishing	Obsidian Publish (paid)	Notion Sites (included in plans)

Custom Themes	CSS themes + snippets	Limited (3 fonts, colors)
Math/LaTeX	Native	Native
Mermaid Diagrams	Native	Not native
Charts	Via plugins	Native (chart view)
Version History	Sync only (or Git plugin)	Native (all plans)
Pricing	Free + paid add-ons	Freemium subscription
Web App	Not available	Full web app

3.2 Data Model Comparison

Obsidian:

- **File-based**: Each note = one `.md` file on disk
- **Metadata**: YAML frontmatter at top of file
- **Structure**: Folder hierarchy + bidirectional links
- **Relationships**: Implicit via `[[wikilinks]]` and tags
- **Querying**: Dataview plugin (DQL or JS) for database-like queries across files
- **Schema**: Unstructured (any frontmatter key-value pairs)

Notion:

- **Block-based**: Content tree of typed blocks (each with UUID)
- **Database model**: Structured tables with typed columns (properties)
- **Structure**: Page hierarchy + database relations
- **Relationships**: Explicit Relation properties between databases
- **Querying**: Database filters, sorts, and groups (UI-based); formula property for computed values
- **Schema**: Structured (defined property types per database)

3.3 Key Architectural Decisions for the Clone

For a web-based clone serving as an AI knowledge platform, consider this hybrid approach:

1. **Block-based editor** (Notion-style) for rich content — easier for web rendering
2. **Markdown import/export** (Obsidian compatibility) for interoperability
3. **Graph view** (Obsidian-style) for knowledge visualization
4. **Databases with typed properties** (Notion-style) for structured data
5. `[[wikilink]]` **support** (Obsidian-style) for quick knowledge linking
6. **Backlinks panel** (Obsidian-style) for discoverability
7. **API-first design** for AI agent integration
8. **Local-first with sync** (optional cloud) for data ownership

---

4. UI/UX Design Reference

4.1 Visual Design Language

Obsidian:

- Dark theme default (light available)
- Minimalist, code-editor aesthetic
- Monospace-friendly typography
- Dense information layout
- Color palette: Dark grays (#1e1e1e, #2b2b2b), purple accent (#7f6df2), muted text
- Icons: Lucide icon set (outlined, minimal)

Notion:

- Light theme default (dark available)
- Clean, spacious, print-like aesthetic
- Serif/sans-serif reading-optimized typography
- Generous whitespace and padding
- Color palette: White (#ffffff), light gray (#f7f6f3), black text, subtle borders
- Icons: Custom icon set (outlined, rounded)
- Emoji-heavy for page icons and expression

- Consistent 8px grid spacing system

4.2 Interaction Patterns

Pattern	Obsidian	Notion
Block insertion	Markdown syntax or `/` commands	Slash commands (`/`) or `+` button
Block manipulation	N/A (text-based)	Drag handle (`⋮⋮`), turn into, duplicate
Navigation	Quick Switcher, File Explorer	Sidebar tree, search, breadcrumbs
Commands	Command Palette (`Cmd+P`)	Slash commands + `Cmd+K` search
Keyboard shortcuts	Fully customizable	Mostly fixed
Context menus	Right-click on files/tabs	Right-click on blocks or `⋮⋮` handle
Hover states	Link previews, toolbar	Block handles, property editors
Drag and drop	Files in explorer, canvas cards	Blocks, pages in sidebar, database rows

4.3 Typography

Obsidian defaults:

- UI font: System (Inter on most platforms)
- Editor font: System default (monospace available)
- Font sizes: 14-16px body, heading scale
- Line height: 1.5-1.6

Notion defaults:

- Three font options: Default (system sans-serif), Serif (Georgia/Charter), Mono (iA Writer Mono/Menlo)
- Body text: 16px
- H1: 30px bold

- H2: 24px bold
- H3: 20px bold
- Small text toggle: reduces to 14px
- Line height: ~1.5

4.4 Color System (Notion)

Text colors: Default (black), Gray, Brown, Orange, Yellow, Green, Blue, Purple, Pink, Red **Background colors:** Same 10 colors as light highlights **Database tag colors:** Same palette for Select/Multi-select options

---

5. Implementation Notes for Coding Agent

5.1 Recommended Tech Stack (Web-Based)

Frontend:

- **Framework**: React or Next.js (SSR for public pages)
- **Editor**:
    - **TipTap** (https://tiptap.dev) — ProseMirror-based; highly extensible block editor; used by many Notion clones
    - **BlockNote** (https://blocknotejs.org) — Block-based editor built on TipTap specifically designed as a Notion-like editor
    - **Lexical** (https://lexical.dev) — Meta's editor framework; good for custom block editors
    - **ProseMirror** (https://prosemirror.net) — Low-level; maximum control
    - **CodeMirror 6** (https://codemirror.net) — For code blocks and markdown source editing
- **Graph visualization**: D3.js, Sigma.js, or Cytoscape.js for knowledge graph
- **Canvas/Whiteboard**: Excalidraw (open source), tldraw, or custom Canvas API implementation
- **Charts**: Recharts, Chart.js, or D3.js
- **Drag and drop**: dnd-kit (https://dndkit.com)

- **State management**: Zustand or Jotai (lightweight) or Redux Toolkit
- **Styling**: Tailwind CSS + CSS custom properties for theming

Backend:

- **Runtime**: Node.js (Express/Fastify) or Python (FastAPI)
- **Database**: PostgreSQL (structured data, properties, relations) + JSON/JSONB for block content
- **Search**: PostgreSQL full-text search or Meilisearch/Typesense for fast fuzzy search
- **File storage**: Local filesystem or S3-compatible object storage for attachments
- **Real-time**: WebSockets (Socket.io or native WS) for live updates
- **Authentication**: NextAuth.js, Passport.js, or custom JWT
- **API**: REST + optional GraphQL; OpenAPI/Swagger for documentation

AI Integration Layer:

- RESTful API for AI agent to read/write/query knowledge base
- Webhook system for event-driven updates
- Vector embeddings store (pgvector in PostgreSQL or Pinecone/Weaviate) for semantic search
- Structured API endpoints: `/api/pages` , `/api/databases` , `/api/blocks` , `/api/search` , `/api/graph`

5.2 Key Open Source References

Project	URL	Relevance
AppFlowy	https://github.com/AppFlowy-IO/AppFlowy	Rust/Flutter Notion alternative; open source
AFFiNE	https://github.com/toeverything/AFFiNE	Notion + Miro + Figma hybrid; block editor + canvas
Outline	https://github.com/outline/outline	Team wiki/knowledge base; markdown-based; React + Node
		Block editor framework

BlockSuite	https://github.com/toeverything/blocksuite	(powers AFFiNE)
TipTap	https://github.com/ueberdosis/tiptap	ProseMirror-based editor framework
BlockNote	https://github.com/TypeCellOS/BlockNote	Notion-like block editor for React
Lexical	https://github.com/facebook/lexical	Meta's extensible editor framework
Plate	https://github.com/udecode/plate	Rich text editor framework built on Slate
Huly	https://github.com/hcengineering/huly	All-in-one project management platform
Trilium Notes	https://github.com/zadam/trilium	Personal knowledge base with tree structure + links
Logseq	https://github.com/logseq/logseq	Obsidian-like outliner with graph; open source
Silverbullet	https://github.com/silverbulletmd/silverbullet	Markdown-based knowledge platform (web-native)
Obsidian-dataview	https://github.com/blacksmithgu/obsidian-dataview	Reference for query language over markdown metadata
react-force-graph	https://github.com/vasturiano/react-force-graph	React component for force-directed graphs

5.3 Data Model Recommendation

```
Database Schema (PostgreSQL):

── workspaces
   ├── id (UUID)
   ├── name
   ├── settings (JSONB)
   └── created_at

── pages
```

```
   ├── id (UUID)
   ├── workspace_id → workspaces.id
   ├── parent_page_id → pages.id (nullable, for hierarchy)
   ├── database_id → databases.id (nullable, if page is a database entry)
   ├── title
   ├── icon (emoji or image URL)
   ├── cover_image_url
   ├── content (JSONB — tree of blocks)
   ├── is_template (boolean)
   ├── is_archived (boolean)
   ├── is_public (boolean)
   ├── created_at
   ├── updated_at
   └── full_text_search (tsvector — for PostgreSQL FTS)

── blocks
   ├── id (UUID)
   ├── page_id → pages.id
   ├── parent_block_id → blocks.id (nullable)
   ├── type (enum: paragraph, heading1, heading2, heading3, bulleted_list,
   │         numbered_list, todo, toggle, quote, divider, callout, code,
   │         image, video, audio, file, bookmark, embed, equation,
   │         table, table_row, table_of_contents, breadcrumb,
   │         synced_block, column_layout, button, database_view,
   │         link_to_page)
   ├── content (JSONB — type-specific data, rich text, properties)
   ├── position (integer — ordering among siblings)
   ├── created_at
   └── updated_at

── databases
   ├── id (UUID)
   ├── page_id → pages.id (database is associated with a page)
   ├── workspace_id → workspaces.id
   ├── title
   ├── schema (JSONB — array of property definitions)
   │   [{ name, type, options, ... }]
   └── created_at

── database_entries (pages that belong to a database)
   ├── id (UUID)
   ├── database_id → databases.id
   ├── page_id → pages.id
   └── properties (JSONB — { property_name: value, ... })

── database_views
   ├── id (UUID)
   ├── database_id → databases.id
   ├── type (enum: table, board, calendar, gallery, list, timeline, chart)
   ├── name
   ├── filters (JSONB)
   ├── sorts (JSONB)
   ├── groups (JSONB)
   ├── visible_properties (JSONB — array of property names)
   └── settings (JSONB — view-specific settings)

── links (for knowledge graph)
   ├── id (UUID)
   ├── source_page_id → pages.id
   ├── target_page_id → pages.id
   ├── source_block_id → blocks.id (optional — where the link appears)
   ├── link_text
   └── created_at

── tags
   ├── id (UUID)
   ├── workspace_id → workspaces.id
   ├── name (with support for nested: "parent/child")
   └── color

── page_tags
   ├── page_id → pages.id
   └── tag_id → tags.id

── comments
   ├── id (UUID)
   ├── page_id → pages.id
   ├── block_id → blocks.id (nullable — for inline comments)
   ├── parent_comment_id → comments.id (nullable — for replies)
   ├── author_id → users.id
   ├── content (text)
   ├── resolved (boolean)
   └── created_at

── automations
   ├── id (UUID)
   ├── database_id → databases.id
   ├── trigger (JSONB — event type + conditions)
   ├── actions (JSONB — array of action definitions)
   ├── is_active (boolean)
   └── created_at

── page_versions
   ├── id (UUID)
   ├── page_id → pages.id
   ├── content_snapshot (JSONB)
   ├── properties_snapshot (JSONB)
   ├── edited_by → users.id
   └── created_at

── integrations (for ChemELN, exPTube, etc.)
   ├── id (UUID)
   ├── workspace_id → workspaces.id
   ├── type (string — "chemeln", "exptube", "webhook", etc.)
   ├── config (JSONB — API keys, endpoints, mappings)
   ├── is_active (boolean)
   └── created_at
```

5.4 Block Content JSON Structure (Example)

```
{
  "type": "paragraph",
  "content": {
    "richText": [
      { "type": "text", "text": "Hello ", "style": {} },
      { "type": "text", "text": "world", "style": { "bold": true } },
      { "type": "text", "text": " and ", "style": {} },
      { "type": "mention", "pageId": "uuid-here", "style": {} },
      { "type": "text", "text": " link", "style": { "link": "https://example.com"
    ]
  }
}
```

```
{
  "type": "callout",
  "content": {
    "icon": "💡",
    "color": "yellow_background",
    "richText": [
      { "type": "text", "text": "Important note here", "style": {} }
    ]
  },
  "children": []
}
```

```
{
  "type": "code",
  "content": {
    "language": "python",
    "code": "print('hello world')",
    "caption": "Example code"
  }
}
```

5.5 API Design for AI Agent

The AI agent needs these core endpoints:

```
# Pages
GET    /api/pages                       — List all pages (with pagination, filters)
GET    /api/pages/:id                   — Get page content + properties
POST   /api/pages                       — Create new page
PUT    /api/pages/:id                   — Update page content
DELETE /api/pages/:id                   — Archive/delete page

# Blocks
GET    /api/pages/:id/blocks            — Get all blocks for a page
POST   /api/pages/:id/blocks            — Append blocks to page
PUT    /api/blocks/:id                  — Update a specific block
DELETE /api/blocks/:id                  — Delete a block

# Databases
GET    /api/databases                   — List all databases
GET    /api/databases/:id               — Get database schema
POST   /api/databases/:id/query         — Query database (filters, sorts, pagination
POST   /api/databases/:id/entries       — Create new entry
PUT    /api/databases/:id/entries/:eid — Update entry properties

# Search
GET    /api/search?q=term               — Full-text search
POST   /api/search/semantic             — Semantic/vector search (for AI)

# Graph
GET    /api/graph                       — Get full knowledge graph (nodes + edges)
GET    /api/graph/:pageId               — Get local graph for a page
GET    /api/pages/:id/backlinks         — Get all pages linking to this page
GET    /api/pages/:id/outlinks          — Get all pages linked from this page

# Tags
GET    /api/tags                        — List all tags with counts
GET    /api/tags/:name/pages            — Get all pages with a tag

# Webhooks (for integrations)
POST   /api/webhooks                    — Register webhook
```

```
GET    /api/webhooks                  — List registered webhooks
DELETE /api/webhooks/:id              — Remove webhook

# Events (webhook payload)
POST   {webhook_url}                  — Page created, updated, deleted, etc.
```

---

6. Integration Context

6.1 ChemELN Integration (Future)

- **What**: Chemical Electronic Lab Notebook — proprietary platform built by SciSymbioAI
- **Integration approach**:
  - Bidirectional sync of experiment records
  - Link ChemELN experiments as pages/database entries in the knowledge platform
  - Embed experiment data views
  - Webhook-based event sync (new experiment → create linked page)
- **Architecture**: Build a generic integration/connector framework with:
  - Configurable API endpoint
  - Field mapping between ChemELN data and knowledge platform properties
  - OAuth or API key authentication
  - Sync status tracking

6.2 exPTube Integration (Future)

- **What**: YouTube-for-experiments platform — proprietary, built for SciSymbioAI
- **Integration approach**:
  - Embed experiment videos within knowledge pages
  - Link video content to related experiment notes and database entries
  - Auto-create pages from new experiment video uploads
  - Metadata sync (experiment tags, descriptions, timestamps)
- **Architecture**: Same connector framework as ChemELN with video-specific features:

  - Video embed block type
  - Timestamp linking (link to specific moments in videos)
  - Transcript integration (if available)

6.3 Generic Integration Architecture

Build a **connector/plugin system** that supports:

1. **Webhook receiver**: Accept events from external systems → trigger page creation/update
2. **Webhook sender**: Push events to external systems on page changes
3. **API connector**: Pull data from external APIs on demand or on schedule
4. **Field mapping**: Map external data fields to knowledge platform properties
5. **Authentication**: Support API keys, OAuth 2.0, bearer tokens
6. **Sync engine**: Track sync state, handle conflicts, support incremental sync
7. **Integration registry**: UI for configuring and managing active integrations

---

7. References

Official Documentation

- Obsidian Help: https://help.obsidian.md
- Obsidian API (Plugin Development): https://docs.obsidian.md/Home
- Obsidian Roadmap: https://obsidian.md/roadmap
- Notion Help Center: https://www.notion.com/help
- Notion API Reference: https://developers.notion.com/reference
- Notion Block Types Guide: https://www.notion.com/help/guides/types-of-content-blocks
- Notion Database Properties: https://www.notion.com/help/database-properties
- Notion Formulas: https://www.notion.com/help/formulas
- Notion Automations: https://www.notion.com/help/database-automations

Key Third-Party References

- Obsidian Community Plugins: https://obsidian.md/plugins
- Obsidian Wikipedia: https://en.wikipedia.org/wiki/Obsidian_(software)
- Thomas Frank Notion Guides: https://thomasjfrank.com
- Notion Blocks Deep Dive: https://carlosrayala.com/deep-dive-into-notion-blocks/
- Notion Property Types: https://www.notionapps.com/blog/notion-database-properties-explained
- Practical PKM (Obsidian): https://practicalpkm.com

Open Source Projects for Reference

- AppFlowy: https://github.com/AppFlowy-IO/AppFlowy
- AFFiNE: https://github.com/toeverything/AFFiNE
- Outline: https://github.com/outline/outline
- BlockNote: https://github.com/TypeCellOS/BlockNote
- TipTap: https://github.com/ueberdosis/tiptap
- Silverbullet: https://github.com/silverbulletmd/silverbullet

---

Document prepared for SciSymbioAI coding agent. This specification should be used alongside the actual ChemELN and exPTube codebases for integration implementation.