

• POZYCJONOWANIE

Dlaczego SumizAI różni się od reszty edytorów notatek?

Większość aplikacji do notatek konkuruje doświadczeniem pisania: ładniejsze płótno, szybszy outliner, czystszy panel boczny. **SumizAI nie konkuruje tam w ogóle.** Zakłada, że pisanie już się odbyło — w rozmowie z modelem AI — i przejmuje pałeczkę w momencie, który normalnie przepada: kiedy zamykasz kartę.

Aktualizacja: 2026-08-14 9 pytań 5 min czytania

Q. Od czego właściwie zaczyna SumizAI, jeśli nie od pustej strony?

Od odpowiedzi. Rozmawiasz z modelem — Claude, GPT, tym, którego podłączyłeś — i gdzieś w tej rozmowie pada odpowiedź, której szkoda byłoby stracić. W zwykłym edytorze to jest miejsce, w którym zaczyna się Twoja praca: kopiujesz, wklejasz, wymyślasz tytuł, decydujesz, gdzie to położyć, i obiecujesz sobie, że uporządkujesz to później.

W SumizAI to jest miejsce, w którym Twoja praca się **kończy**. Odpowiedź sama staje się notatką: tytuł, krótkie podsumowanie, rozwinięcie i pełny zapis pytania oraz odpowiedzi, z których powstała. Nie otwierałeś edytora. Nie nazywałeś pliku. Rozmawiałeś, a notatka się pojawiła.

Cała różnica w jednym zdaniu: edytor notatek to miejsce, do którego wkłada się tekst, a SumizAI to miejsce, do którego tekst przychodzi.

Q. Czyli to czat z przyciskiem „zapisz”?

Nie, i ta różnica ma znaczenie. Przycisk „zapisz” daje transkrypcję — tę samą ścianę tekstu, którą już miałeś, tylko przechowywaną gdzie indziej. SumizAI produkuje **notatkę o strukturze**, i to struktura jest tu sednem.

Każda notatka ma cztery wyraźne części. **Tytuł** mówi, o czym jest. **Podsumowanie** to wersja, którą czytasz, przeglądając listę. **Rozwinięcie** to wersja, po którą sięgasz, kiedy naprawdę potrzebujesz szczegółu. A **materiał źródłowy** to kompletne oryginalne pytanie i kompletna oryginalna odpowiedź, trzymane pod spodem jako dowód, skąd wzięła się notatka.

Ta ostatnia część jest celowo tylko dopisywalna. Notatkę można scalić, przemianować i przelinkować, ale oryginalne pytanie i odpowiedź nigdy nie są przepisywane. Pół roku później, kiedy podsumowanie brzmi odrobinę zbyt pewnie, możesz sprawdzić, co faktycznie padło.

Q. Co dzieje się z notatką po jej napisaniu?

Trafia na półkę, choć nikt jej tam nie odkłada. Każdy vault ma spis treści — prawdziwe drzewo rozdziałów, do sześciu poziomów, numerowane `1.` → `1.1` → `a)` → `a.1)` i dalej. Kiedy powstaje notatka, model dostaje ponumerowane rozdziały, które już istnieją, i ma **wskazać jeden** albo poprosić o podrozdział pod nim.

Wskazuje numer, a nie wymyśla nazwę. Ten szczegół sprawia, że drzewo pozostaje zdrowe: gdyby model za każdym razem układał nazwę rozdziału od nowa, dostałbyś *Warzywa*, *uprawa warzyw* i *Uprawa roślin warzywnych* jako trzy różne gałęzie. Nowa gałąź powstaje wyłącznie wtedy, gdy nic istniejącego nie pasuje.

Notatka staje się numerowanym liściem tego drzewa i nigdy nie zostaje poza nim. Usuniesz rozdział — jego notatki przesuwają się do rodzica, zamiast wpaść w próżnię.

Q. Co powstrzymuje bibliotekę przed zamienieniem się w stos prawie-duplikatów?

Sprawdzenie, które biegnie przed zapisem. Wyszukiwanie pełnotekstowe i podobieństwo trigramowe wybierają z vaulta krótką listę notatek, które wyglądają na opisujące to samo. Potem **drugi model czyta kandydata i nowy materiał, po czym ocenia dopasowanie w skali od 0 do 1** — z pisemnym uzasadnieniem oceny.

Jeśli ocena przekracza próg ustawiony dla tego vaulta, dostajesz wybór: scalić z istniejącą notatką albo zostawić obie. Vault można też ustawić na „zawsze scalaj” lub „zawsze twórz nową” — wtedy próg nadal decyduje, co w ogóle liczy się jako ten sam temat.

Kiedy dochodzi do scalenia, ono **dopisuje**. Istniejąca notatka zachowuje swoje oryginalne pytanie i odpowiedź nietknięte i zyskuje nowe obok nich. Nic, co złapałeś, nie zostaje po cichu nadpisane późniejszą, krótszą wersją tej samej myśli.

Q. Jak notatki są ze sobą połączone?

Przez sam tekst. Powiązane notatki są przywoływane jako zwykłe linki Markdown w treści notatki — `[Strategie dzielenia tekstu](strategie-dzielenia-tekstu.md)` — wplecione w zdanie, a nie wrzucone do sekcji *Powiązane* na dole.

Jest tu zabezpieczenie, o którym warto wiedzieć. Model dostaje mapę prawdziwych tytułów i prawdziwych nazw plików i może linkować wyłącznie do czegoś z tej mapy. Wszystko inne, co wyprodukuje, jest wycinane przy odbiorze. Model poproszony o odnośnik ochoczo wymyśli

prawdopodobnie wyglądającą nazwę pliku, a **martwy odnośnik jest gorszy niż jego brak**.

Pod każdą notatką masz też drugi kierunek: listę notatek, które prowadzą *do* niej. Razem zamieniają katalog plików w coś, po czym da się chodzić.

Q. Czy AI widzi moje pozostałe notatki, kiedy zadaję pytanie?

Część z nich, i to celowo. Twoje pytanie jest wysyłane razem ze spisem treści vaulta i **pełną treścią maksymalnie pięciu notatek**, które aplikacja uznała za najtrafniejsze, w budżecie 24 000 znaków.

To nie jest „AI czyta cały Twój vault” — byłoby to wolne, drogie i gorsze, bo model, który dostaje wszystko, odpowiada ze średniej wszystkiego. Pięć trafnych notatek plus mapa reszty to ostrzejsze narzędzie.

A kiedy odpowiedź przychodzi, aplikacja mówi, które notatki do niej weszły. Możesz sprawdzić rozumowanie w źródle, zamiast ufać pewnie brzmiącemu akapitowi.

Q. Gdzie leżą pliki?

W katalogu, który wskazałeś, jako zwykłe pliki `.md`. Układ jest prosty: katalog na vault, plik `base.md` ze spisem treści i katalog `notes/` z jednym plikiem na notatkę. Nazwa pliku to slug tytułu, więc katalog da się czytać, zanim cokolwiek otworzysz.

Baza danych jest wyłącznie indeksem — zna tytuły, podsumowania i to, gdzie leżą pliki. Źródłem prawdy jest plik, a przeindeksowanie odbudowuje indeks z plików, gdyby oba się rozjechały. **W sporze wygrywa plik.**

Praktyczna konsekwencja: wskaż ten katalog Obsidianowi, VS Code albo repozytorium Git i wszystko działa, bo odnośniki między notatkami to względne ścieżki do prawdziwych plików.

Q. Czym SumizAI świadomie nie jest?

Nie jest narzędziem do pisania. Nie ma płótna, tablicy, widoków bazodanowych ani kanbana. Jeśli chcesz redagować dokument, użyj czegoś zbudowanego do redagowania dokumentów.

Nie jest klientem czatu z lepszym stylowaniem. Rozmowa jest surowcem, nie produktem.

Nie jest miejscem, w którym Twoje notatki są zamknięte. Każdy vault pobiera się jako ZIP z `base.md` i katalogiem `notes/`, a odnośniki w środku są względne — archiwum otwiera się w Obsidianie, VS Code albo na GitHubie bez instalowania czegokolwiek.

I nie jest odsprzedawcą AI. Podłączasz **własny klucz API** do jednego z siedmiu dostawców i płacisz temu dostawcy bezpośrednio. SumizAI bierze dolara miesięcznie za aplikację i nie ma udziału w tym, co kosztuje model.

Q. Dla kogo to zły wybór?

Dla kogoś, kto chce wspólnej przestrzeni zespołowej. Nie ma vaultów wielouczestnikowych ani synchronizacji między urządzeniami — laptop i telefon trzymają dwa osobne zbiory notatek. Model danych został przygotowany pod przyszłą synchronizację, ale uczciwie trzeba powiedzieć, że jej dziś nie ma.

Dla kogoś, kto chce, żeby aplikacja dostarczyła AI. Nie dostarczy: bez Twojego klucza nie ma z czym rozmawiać, a lista modeli jest pobierana od dostawcy Twoim kluczem, a nie zgadywana z wbudowanego katalogu.

Dla kogoś, kto po prostu nie myśli na głos z modelem. SumizAI jest bardzo dobry w łapaniu rozmów, które i tak prowadzisz, i bezużyteczny, jeśli ich nie prowadzisz.