Każdy blog technologiczny wydaje się obecnie zawierać przynajmniej jakąś analizę tematów związanych ze sztuczną inteligencją. Ten artykuł wynika z mojej osobistej potrzeby niezawodnego narzędzia, które pomoże mi w codziennych zadaniach przy biurku. Kryteria porównania opierają się na moich osobistych wymaganiach, a ocena jest całkowicie empiryczna i subiektywna. Oznacza to, że faktycznie pobrałem i zainstalowałem każde z wymienionych narzędzi i przetestowałem je pod kątem moich własnych potrzeb.

Możesz zauważyć, że Ollama nie jest uwzględniona. Podczas korzystania z Ollama priorytetowo traktowałem narzędzia z przyjaznym użytkownikowi UX, odpowiednie dla osób o średnim poziomie wiedzy technicznej, które wolą unikać zarządzania opartego na konsoli lub niskiego poziomu.

Oto kluczowe obszary, na których się skupiłem:

- łatwość instalacji i zarządzania
- praca z lokalnie pobranymi modelami
- możliwość przetwarzania dokumentów
- działanie jako centrum dla innych różnych aplikacji
- stabilność i ładny i łatwy interfejs użytkownika/doświadczenie użytkownika

Jeśli chcesz to inaczej ująć, powiedziałbyś, że chciałem znaleźć oprogramowanie, które wykorzystuje większość możliwości modelu bez wchodzenia w programowanie lub bardzo zaawansowaną konfigurację. Przede wszystkim opisuję funkcjonalności tych narzędzi, a następnie gdzieś pod koniec tego artykułu znajdziesz tabelę porównawczą.

Jeśli chcesz, abym dodał jakiś aspekt do tych narzędzi i porównał je lub wysłał Ci narzędzie, które chciałbyś zobaczyć w porównaniu z nimi, skontaktuj się ze mną za pomocą formularza kontaktowego pod tym artykułem.

Testowanie oprogramowania LLM do prowadzenia lokalnego LLM

Znalezienie odpowiednich kandydatów do porównania jest trudne. Wyobrażam sobie, że powodem jest to, że albo są oni znani programistom AI, więc nie muszą nikogo pytać, albo jeśli ktoś nie jest techniczny, to po prostu używa komercyjnych wersji online tych narzędzi – takich jak ChatGPT. Chciałem jednak zająć się tematem

z „półtechnicznego” punktu widzenia. Możliwe więc, że pominąłem jakieś dobre narzędzia – jeśli tak uważasz, skontaktuj się ze mną, a dodam je do porównania.

AnythingLLM – funkcjonalne rozwiązanie dla osób nietechnicznych

- Strona internetowa: https://anythingllm.com/
- Dokumentacja: https://docs.anythingllm.com/
- Kod: https://github.com/Mintplex-Labs/anything-llm
- Wersja testowana: 1.6.9
- Zobowiązania: 1 tys., gwiazdki: 27,4 tys., widełki: 2,8 tys.
- Organizacja/osoba/właściciel: Mintplex Labs Inc.

LM studio – przyjazne dla programistów środowisko AI

- Strona internetowa: https://lmstudio.ai/
- Dokumentacja: https://lmstudio.ai/docs
- Kod: https://github.com/lmstudio-ai
- Wersja testowana: 0.3.5
- Zmiany (tylko główne repozytorium): 0,1 tys., Gwiazdki: 1,7 tys., Forki: 0,1 tys.
- Organizacja/osoba/właściciel: Element Labs, Inc.

Jan – KISS to najlepsza zasada

- Strona internetowa: https://jan.ai/
- Dokumentacja: https://jan.ai/docs
- Kod: https://github.com/janhq/jan
- Wersja testowana: 0.5.8
- Zobowiązania: 3,8 tys., gwiazdki: 23,6 tys., rozwidlenia: 1,4 tys.
- Organizacja/osoba/właściciel: Homebrew Computer Company

GPT4All – proste, ale skuteczne rozwiązanie

- Witryna internetowa: https://GPT4All.io/
- Dokumenty: https://docs.GPT4All.io/
- Kod: https://github.com/nomic-ai/GPT4All
- Testowana wersja: 3.4.2
- Zatwierdzenia: 2,1 tys., gwiazdki: 70,7 tys., widelce: 7,7 tys.
- Organizacja/osoba/właściciel: Nomic, Inc

OpenWebUI – znany interfejs użytkownika, nieznane możliwości

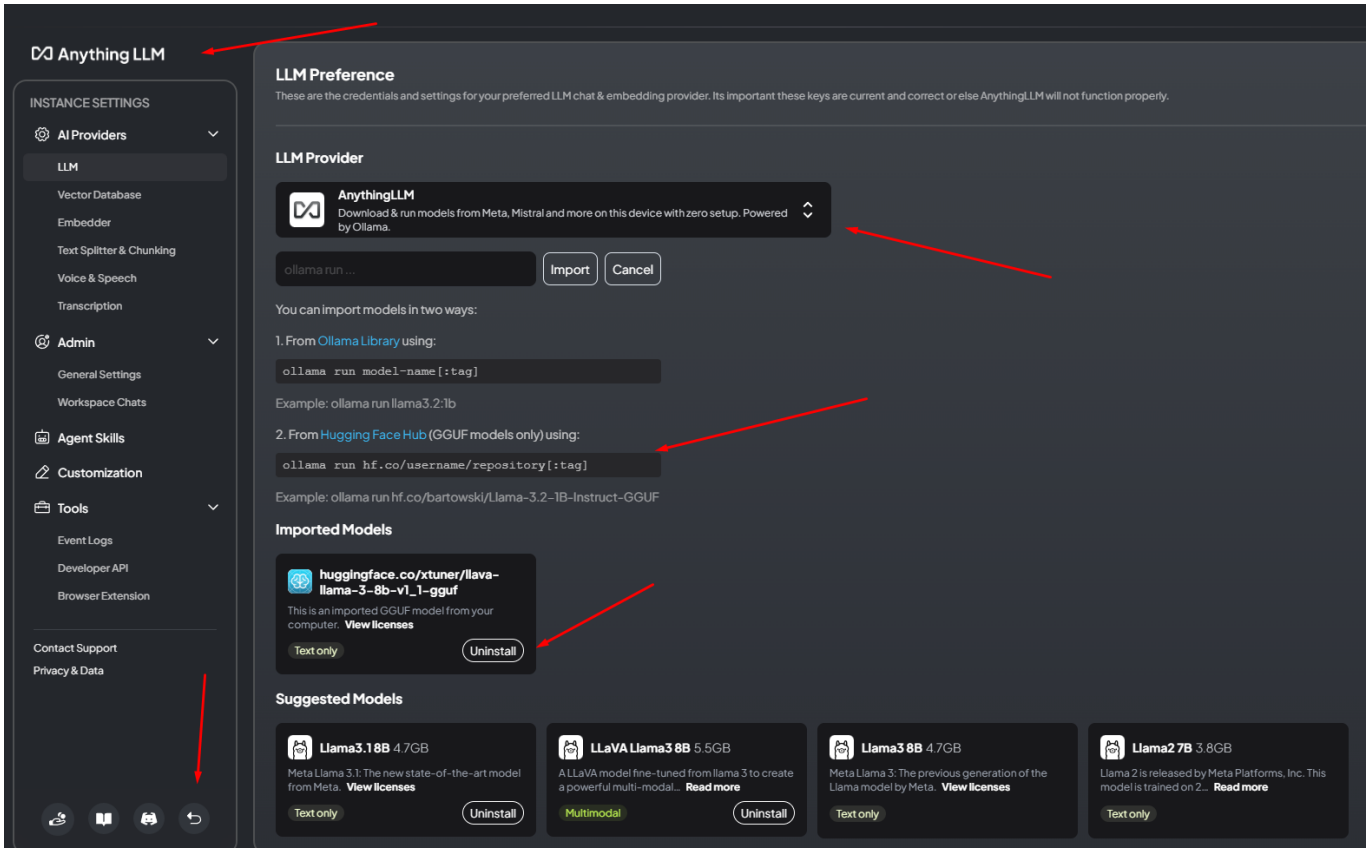
- Strona internetowa: https://openwebui.com/
- Dokumenty: https://docs.openwebui.com/
- Kod: https://github.com/open-webui/open-webui
- Wersja testowana: 0.3.35
- Zatwierdzenia: 7,2 tys., gwiazdki: 47,6 tys., rozwidlenia: 5,8 tys.
- Organizacja/osoba/właściciel: Tim J. Baek (zobacz https://docs.openwebui.com/team)

Interfejs zarządzania modelem AI i pobierania

Pierwszą rzeczą, którą musisz zrobić, jest zainstalowanie interfejsu użytkownika – co jest proste dla wszystkich interfejsów z wyjątkiem OpenWebUI. Następną rzeczą, którą musisz zrobić, jest pobranie modelu AI na lokalny komputer, aby móc go uruchomić. Później oczywiście powinieneś móc z łatwością zarządzać pobranymi modelami.

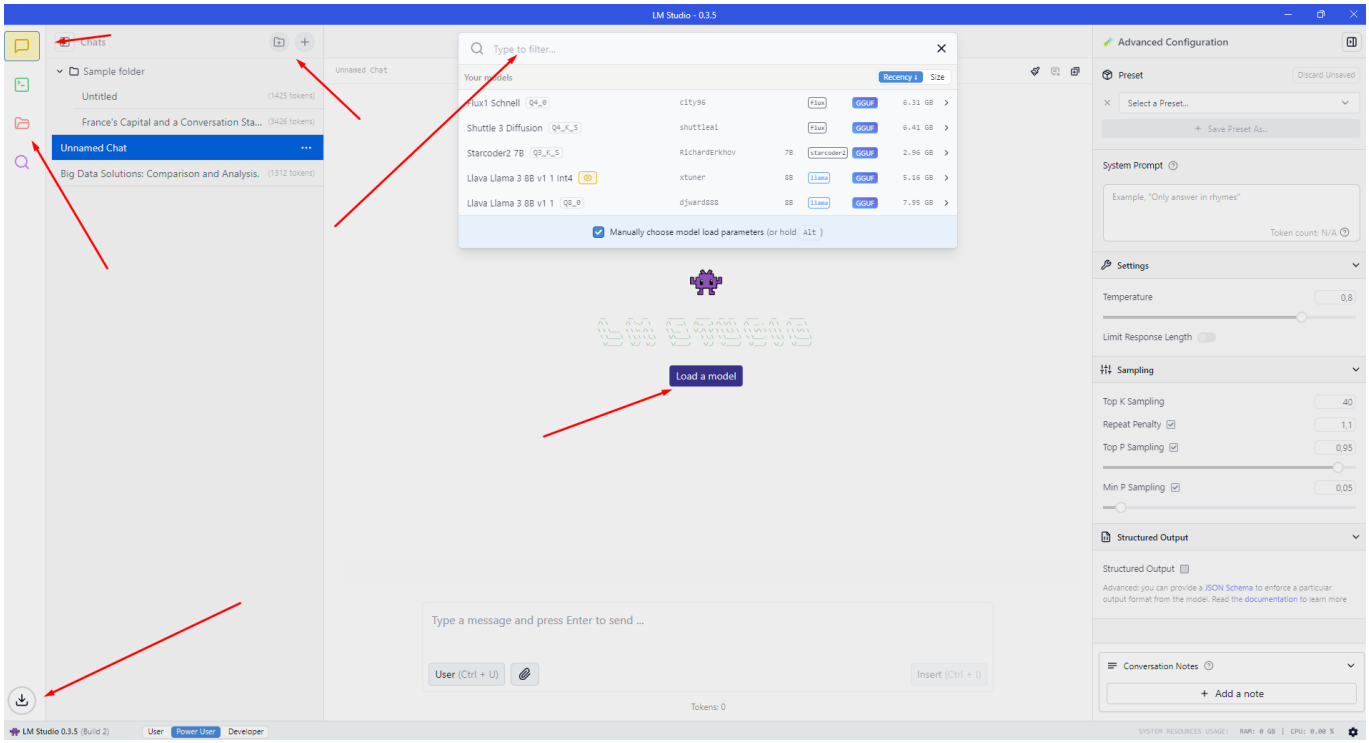
AnythingLLM

W przypadku **AnythingLLM** pierwszą rzeczą, którą musisz zrobić, jest znalezienie ustawień — znajdują się one w lewym dolnym rogu oprogramowania. Następnie przejdź do „Wszystkich dostawców” i wybierz jednego z menu rozwijanego lub pobierz sugerowany model za pomocą konsoli. Na liście rozwijanej zobaczysz wielu różnych dostawców, więc miło, że obsługują wielu z nich. Obejmuje to nie tylko HuggingFace, ale także lokalnie uruchomionego Ollama i różnych komercyjnych dostawców, takich jak OpenAI, Azure, Antropic itd. W programie nie ma opcji „Wyszukaj modele”, więc musisz wyszukać je online, a następnie uruchomić polecenie z interfejsu **AnythingLLM**, aby je pobrać. Interfejs nie pomoże Ci w wyborze różnych rozmiarów modeli ani ich filtrowaniu, ale po pobraniu zobaczysz, czy model jest tylko tekstowy, czy multimodalny. Następnie możesz po prostu wrócić i otworzyć nowy wątek, który można zorganizować w folderach, a każdy folder może mieć inne ustawienia dostawcy LLM. Taka organizacja jest bardzo przydatna, jeśli chcesz przetestować różne modele (lub ustawienia modelu) na tych samych zapytaniach.



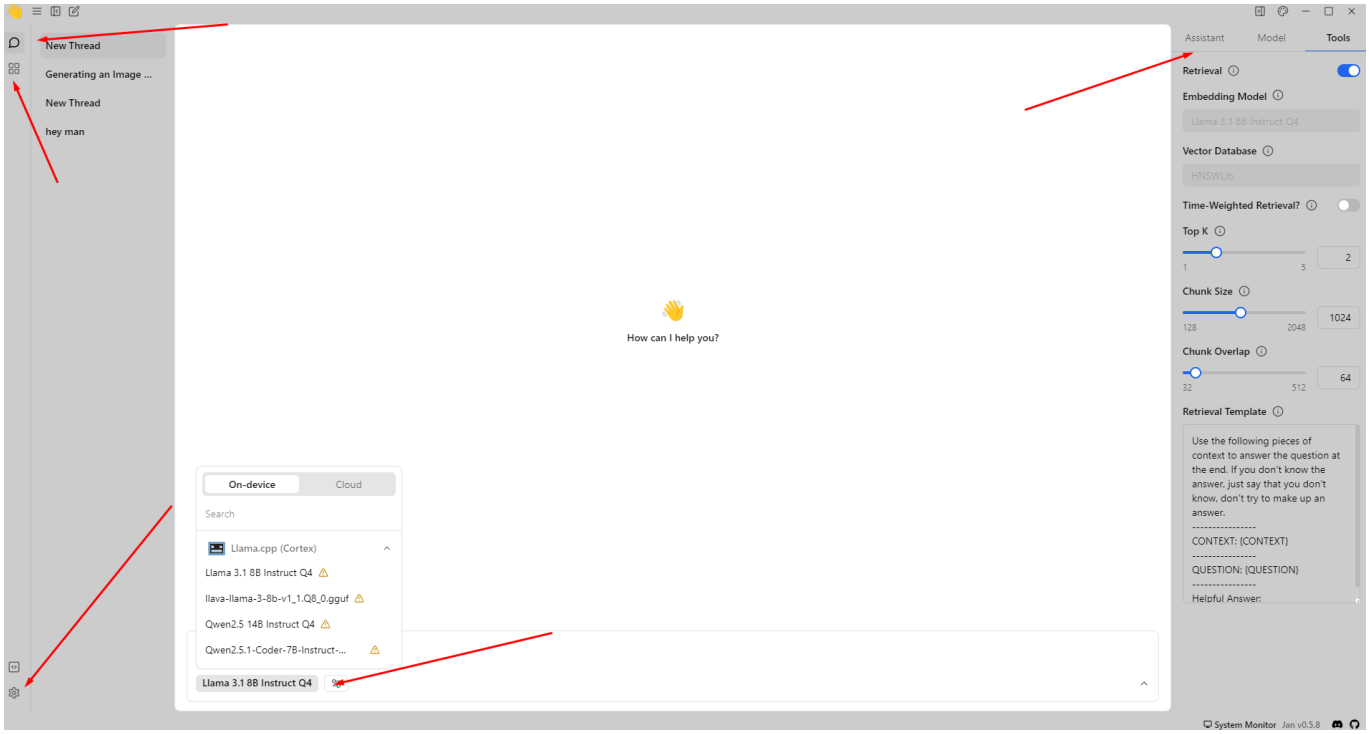
LM Studio

W **LM Studio** nie jest również oczywiste, gdzie pobrać modele. Po lewej stronie znajdziesz ikony, z których trzecia to „Moje modele”, ale nie znajdziesz jej tam. Musisz wpisać coś w polu wyszukiwania na górnym pasku wyszukiwania. Następnie zobaczysz, że otwiera się nowe okno, w którym możesz zobaczyć wyniki z HuggingFace i znaleźć różne modele. To okno pozwala wybrać wersję lub rozmiar modelu, co jest bardzo przydatne, jeśli repozytorium ma różne wersje tego samego LLM. Kiedy wrócisz do „Czatu”, zauważysz, że możesz organizować czaty w folderach, tak jak w AnythingLLM.



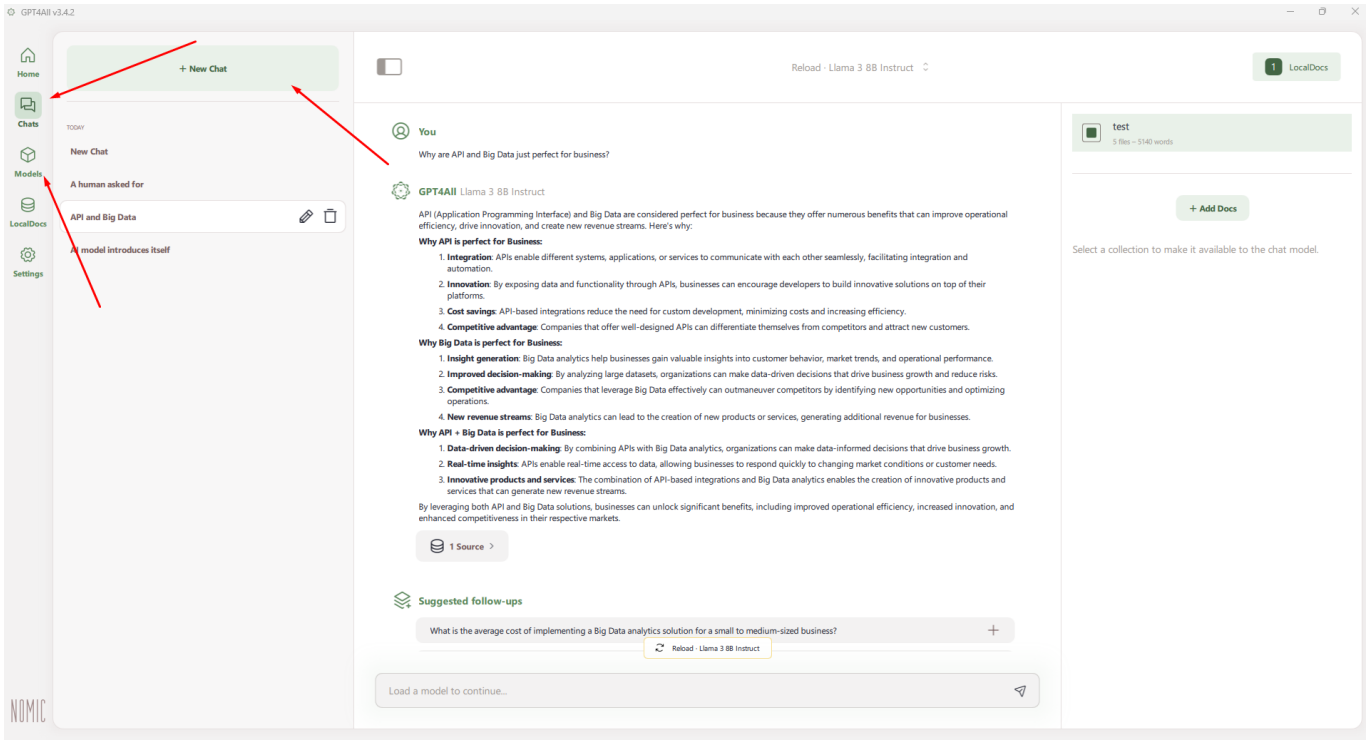
Sty

Podczas korzystania z **Sty**, pierwszą czynnością powinno być przejście do Hub. W Hubie znajdziesz różne modele, które są oznaczone, jeśli mogą być wolne na Twoim urządzeniu lub niedziałające, lub jeśli możesz ich używać. Jest to bardzo fajna i przydatna funkcja, szczególnie jeśli jesteś nowicjuszem. Pomaga Ci ona określić, które modele chcesz wypróbować lokalnie. W ustawieniach na samym dole możesz zobaczyć, że możesz połączyć się z wieloma różnymi dostawcami komercyjnymi, takimi jak: Anthropic, MistralAI, OpenAI i tak dalej. Po pobraniu modelu, jeśli wrócisz do czatów, zauważysz, że nie możesz organizować czatów w folderach. Chociaż interfejs ma być bardzo prosty, więc rozumiem tę decyzję dotyczącą interfejsu użytkownika.



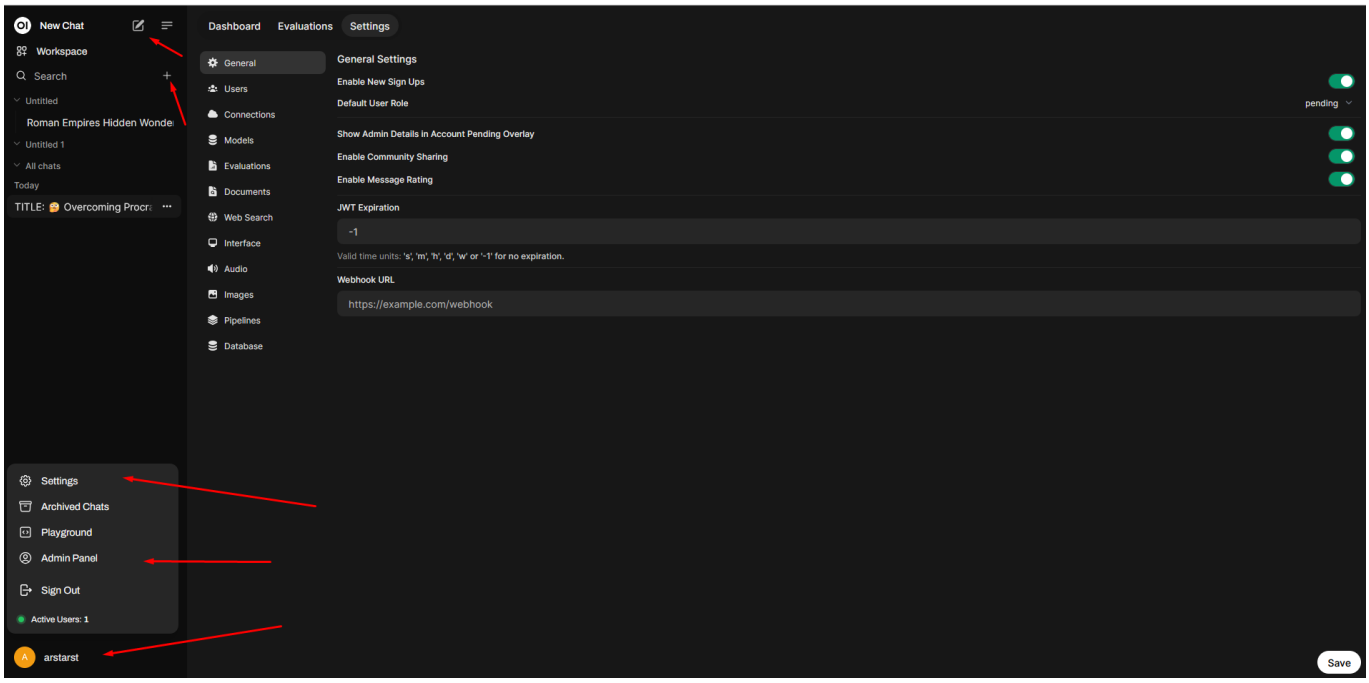
GPT4All

Dzięki **GPT4All** pierwszą rzeczą, którą chcesz zrobić, jest przejście do modeli po lewej stronie, gdzie możesz znaleźć model, który chcesz pobrać. Ma również bardzo przydatny interfejs, ponieważ otrzymujesz parametry, takie jak rozmiar pliku, wymagana pamięć RAM, parametry, kwantyzacja i typ. Na początku otrzymasz propozycje modeli, ale możesz użyć funkcji wyszukiwania, aby znaleźć cokolwiek z HuggingFace. Po pobraniu modelu oczywiście wracasz do czatów, gdzie możesz rozpocząć rozmowę. Tak jak w styczniu, nie otrzymasz folderów, tylko wątki, w których możesz użyć konkretnych modeli.



OpenWebUI

W przypadku wszystkich powyższych opcji **OpenWebUI** jest jedynym wyjątkiem, w którym nie otrzymasz instalatora. Zamiast tego musisz zainstalować Pythona, a następnie obsługiwać aplikację. Jest to poważna wada, jeśli nie jesteś osobą korzystającą z konsoli. Chociaż zdecydowałem się na uwzględnienie go ze względu na dostępne funkcjonalności i ponieważ instalacja polega w rzeczywistości na uruchomieniu dwóch poleceń z konsoli. Domyślnie **OpenWebUI** używa Ollama jako źródła modeli i nie otrzymujesz żadnego interfejsu do wyszukiwania i pobierania modeli z panelu administracyjnego. Aby połączyć usługi komercyjne, musisz wprowadzić adres URL i klucz API zgodny z OpenAI. Na początku zostaniesz poproszony o utworzenie konta — ponieważ jest to konto lokalne, możesz bez obaw używać fikcyjnych danych (jestem świadomy, że możesz to wyłączyć podczas instalacji Docker). Możesz organizować konwersacje w folderach i czatach, a interfejs użytkownika jest bardzo podobny do ChatGPT, więc taka kombinacja byłaby najłatwiejszym wyborem, jeśli chcesz uruchomić **OpenWebUI** online.



Przetwarzanie plików w lokalnym czacie AI

Możliwość analizowania plików, które dostarczę do modelu, jest jedną z najważniejszych rzeczy, których wymagam od interfejsu. Bez niej mógłbym po prostu użyć wiersza poleceń z Ollamą zamiast instalować pełny interfejs. Chciałem sprawdzić, czy interfejs może przetwarzać przesłane dokumenty, przekazywać je do modelu i zwracać znaczące informacje. Podczas gdy w przypadku prostych plików tekstowych nie stanowiło to GŁÓWNIE problemu, warto zauważyć, że szukałem również obrazów (na przykład do celów OCR) i transkrypcji głosu. Żadne z narzędzi nie było w stanie faktycznie przepisać dźwięku od razu po wyjęciu z pudełka: AnythingLLM nie mogło pobrać modelu Whisper, Jan i GPT4All nie mogły przesłać takich plików, LM Studio zawsze interpretowało dane wejściowe jako tekst. Dla OpenWebUI istnieje narzędzie Whisper, ale nie byłem w stanie sprawić, by działało – chociaż istnieje otwarty problem z natywnym wsparciem: https://github.com/open-webui/open-webui/issues/

Powinieneś wiedzieć, że testowane przeze mnie oprogramowanie nie jest przeznaczone do tworzenia grafiki, ale głównie do LLM przy użyciu formatu GUFF (tworzenie grafiki LLM najczęściej wykorzystuje format “safetensors”) – jeśli szukasz interfejsu użytkownika, który jest w stanie tworzyć obrazy, powinieneś przyjrzeć się

na przykład ComfyUI, który nie jest analizowany w tym artykule, ponieważ nie jest to interfejs użytkownika do prostego użytku LLM. Wiem, że można połączyć OpenWebUI z ComfyUI, ale to znowu wykracza poza zakres tego porównania.

Jedną z rzeczy, które są reklamowane w **AnythingLLM** jest to, że pozwala przetwarzać wiele różnych typów dokumentów. Problem polega na tym, że – przynajmniej w moim przypadku – że **AnythingLLM** lubi się zawieszać. Przestaje przetwarzać dokument, jeśli napotka jakiś błąd i nie podaje informacji o tym, co się stało, ale zamiast tego musisz ponownie uruchomić całą aplikację, aby znów zaczęła działać. Ponadto pamiętaj, że każdy przesłany dokument jest dodawany do kolekcji, do której **AnythingLLM** może uzyskać dostęp w celu przyszłych czatów. Czasami wydawało mi się to sprzeczne z intuicją, ale rozumiem logikę RAG w tym przypadku. Po wybraniu odpowiedniego modelu był w stanie prawidłowo opisać również przesłane obrazy.

W przypadku **LM Studio** nie miałem żadnych takich problemów. Wszystko, co dostarczyłem do **LM Studio**, zostało przetworzone prawidłowo (lub wyświetlony został zrozumiały błąd). Po przesłaniu otrzymasz powiadomienie, że dokument zostanie dodany do RAG — ale RAG to temat, który omówię później. To oprogramowanie również przetworzyło przesłane przeze mnie obrazy bez żadnych problemów i pozwoliło LLM opisać, co się na nich znajduje.

W przypadku **Jan** miałem problemy z przesyłaniem dokumentów, ale na stronie internetowej wspomniano, że ta funkcjonalność jest eksperymentalna. Zasadniczo istnieje problem w GitHub w toku (podejrzewam, że jest on związany z problemami, które miałem — https://github.com/janhq/jan/issues/4023), więc spodziewałbym się, że zostanie to naprawione w następnej wersji. Jednak w chwili pisania artykułu interpretacja dokumentu w ogóle nie działa. Oczywiście z tego powodu nie mogłem przetestować przesyłania i interpretacji obrazów.

GPT4All był szczerze mówiąc najdziwniejszym przypadkiem funkcjonalności przesyłania dokumentów ze wszystkich. Pozwala tylko na przesyłanie plików XLSX. Nie mam pojęcia, dlaczego tak jest i dlaczego nie mogę przesyłać plików TXT, DOCX ani niczego innego. To poważna wada tego oprogramowania. W rezultacie nie mogłem przetestować jego zdolności do przesyłania obrazów do LLM i otrzymywania właściwej odpowiedzi.

Nie wiem dlaczego, ale w przypadku **OpenWebUI** przesyłanie dokumentów było najwolniejsze, ale dokumenty były interpretowane poprawnie. Ponieważ

uruchamiając LLM lokalnie, zgadzam się, że czasami może działać wolno, więc jest to dla mnie akceptowalny przypadek. Przetwarzał obrazy bez żadnych problemów ani zbędnego czekania.

Używanie RAG, wiedzy, baz danych i informacji zewnętrznych

Porównywanie dokumentów z lokalnie hostowanymi LLM

Jedną z rzeczy, którą chciałem wypróbować, było zmuszenie AI do porównania pięciu do dziesięciu różnych dokumentów, które im dostarczyłem. Ale zasadniczo uniemożliwiło to we wszystkich testowanych przeze mnie przypadkach, prawdopodobnie dlatego, że wszystkie dokumenty były traktowane jako kontekst lub dodawane do RAG. W przypadku traktowania kontekstowego po prostu osiągnąłem limity kontekstowe LLM, a w przypadku RAG nie były brane pod uwagę całe dokumenty, ale tylko ich części. Rozumiem, że jest to możliwe w przypadku niektórych LLM, zwłaszcza komercyjnych, ale na potrzeby tego artykułu pominę ten aspekt w porównaniu, ponieważ nie byłem w stanie osiągnąć tego celu za pomocą rozważanych narzędzi. Chociaż otwiera to temat: w jaki sposób testowane interfejsy wykorzystują możliwości RAG?

Aby lepiej wyjaśnić ten aspekt, muszę najpierw opisać, czym jest RAG w kontekście Large Language Models (LLM), takich jak GPT (Generative Pre-trained Transformer). „RAG” oznacza Retrieval-Augmented Generation (generowanie rozszerzone o wyszukiwanie). RAG to metodologia, która łączy możliwości wstępnie wytrenowanych modeli językowych z mechanizmami wyszukiwania w celu zwiększenia generowania odpowiedzi. Ta architektura modelu pomaga poprawić trafność i dokładność faktograficzną wyników generowanych przez LLM.

Jak działa RAG

- **Mechanizm pobierania:** Gdy wyświetlane jest zapytanie lub monit, system RAG pobiera odpowiednie dokumenty lub informacje z dużego korpusu lub bazy danych.
- **Rozszerzanie:** Pobrane dokumenty są następnie używane do rozszerzania monitu, zapewniając dodatkowy kontekst lub bezpośrednio istotne informacje dla modelu języka.
- **Generowanie:** Rozszerzony monit jest wprowadzany do opartego na

transformatorze modelu języka, który generuje odpowiedź na podstawie zarówno oryginalnego zapytania, jak i nowo pobranych informacji.

Zalety RAG

- **Poprawiona dokładność:** Pobierając odpowiednie informacje zewnętrzne, RAG może generować dokładniejsze i kontekstowo odpowiednie odpowiedzi.
- **Odpowiedzi oparte na wiedzy:** Pozwala modelowi na udzielanie odpowiedzi w oparciu o szerszą bazę wiedzy wykraczającą poza to, na czym został początkowo wyszkolony.
- **Elastyczność:** Połączenie pobierania i generowania oferuje zrównoważone podejście, które może mieć kluczowe znaczenie dla aplikacji wymagających wysokiej jakości, niezawodnych wyników.

Wnioskiem byłoby, że jeśli funkcjonalność RAG jest dostępna, równie dobrze mógłbym spróbować jej użyć. Poniżej znajdziesz porównanie tego, co każde z rozważanych narzędzi jest w stanie zrobić w kontekście korzystania z RAG-ów, baz wiedzy, wyszukiwania informacji online, które mogłyby rozszerzyć wiedzę, łączenia się z zewnętrznymi źródłami danych, takimi jak MySQL/MariaDB, GitHub/GitLab, Postgres itd.

Oczywiście korzystanie z systemów RAG i agentów AI to idealny przypadek integracji z Twoją firmą SaaS. Zobacz kilka opcji poniżej:

Narzędzia LLM w kontekście korzystania z informacji zewnętrznych

AnythingLLM wykorzystuje funkcję o nazwie „Umiejętności agenta” do łączenia się z różnymi źródłami danych. Zasadniczo na czacie dzwonisz na „@agent” i podajesz polecenie, które agent powinien wykonać (umiejętności te nie są domyślnie używane na czacie, jeśli po prostu dzwonisz do LLM, ale jeśli dzwonisz bezpośrednio do agenta, działa to jak marzenie). Można to uznać za zaletę lub wadę, w zależności od tego, jak wolisz korzystać z tej funkcji. W przypadku umiejętności agenta masz dostępne funkcje, takie jak generowanie i zapisywanie plików, generowanie wykresów, wyszukiwanie w sieci za pomocą DuckDuckGo lub wyszukiwarki Google, Bing lub Serply (i więcej) oraz łącznik SQL, który umożliwia łączenie się z MySQL, Postgres lub SQL Server. Muszę powiedzieć, że nie byłem w stanie prawidłowo uruchomić generowania wykresów, ale wyszukiwanie w sieci działało bez zarzutu.

Miałem trochę problemów z łącznikiem SQL, ale udało mi się go uruchomić. Jednak interpretowanie danych z bazy danych nie działało zgodnie z oczekiwaniami, prawdopodobnie dlatego, że baza danych była traktowana jako dane kontekstowe, z których można wybierać elementy, a nie jako „monolit", który może zwracać dokładne wyniki. Możliwe, że użycie innego modelu (lub innych ustawień temperatury) dałoby lepsze wyniki, ale zasadniczo wniosek jest taki, że **AnythingLLM** połączył się z bazą danych i wykorzystał ją jako źródło wiedzy. Jeszcze jedna rzecz na temat dokumentów: gdy przesyłasz dokument na czacie, jest on dodawany do systemu zarządzania dokumentami, gdzie możesz go przeciągnąć w prawo lub w lewo, w zależności od tego, czy chcesz użyć tego dokumentu w bieżącym kontekście. Jeśli chcesz, aby dokument był interpretowany precyzyjnie na czacie, powinieneś również użyć pinezki. Ale nawet deweloperzy twierdzą, że to może, ale nie musi dawać oczekiwanych rezultatów (więcej informacji znajdziesz na https://docs.useanything.com/llm-not-using-my-docs). Poza tym jest to dość łatwy i intuicyjny system zarządzania dokumentami. Warto również zauważyć, że jest to jedyne rozważane narzędzie, które zapewnia natywnie (gdy przejdziesz do zakładki „Dokumenty" i „Łączniki danych" w RAG) możliwość połączenia z repozytorium GitHub, repozytorium GitLab, transkryptami YouTube, link scrapperem i Confluence.

W przypadku **LM Studio** ta funkcjonalność jest, powiedziałbym, częściowo dostępna — w każdym czacie możesz przesyłać dokumenty, które będą używane w tym czacie, ale nie ma centralnego punktu, w którym mógłbyś zarządzać wszystkimi dokumentami lub wszystkimi połączeniami. Nie znalazłem również żadnych innych możliwości łączenia się z zewnętrznymi źródłami danych, takimi jak bazy danych SQL lub wyszukiwarki. Prawdopodobnie można by to dodać, używając zakładki dewelopera, ale ponieważ chciałbym używać narzędzia jako użytkownik półtechniczny, pisanie niestandardowego skryptu nie byłoby w tej chwili akceptowalnym rozwiązaniem.

Nie znalazłem również żadnego systemu zarządzania dokumentami w **Jan**. Być może dlatego, że chcą to później zaimplementować za pomocą systemu wtyczek, chociaż w tej chwili nie istnieje. Nie ma również możliwości połączenia się z wyszukiwarkami lub bazami danych.

GPT4All dobrze obejmuje funkcjonalność RAG — mają całkiem niezły system przesyłania i zarządzania dokumentami w bazie wiedzy, a w czacie możesz wybrać, których zestawów danych chcesz użyć. Nie ma jednak możliwości łączenia się z zewnętrznymi bazami danych ani wyszukiwarkami, więc możesz pracować tylko na dokumentach tekstowych.

Dzięki **OpenWebUI** masz w rzeczywistości dość zaawansowane ustawienia przetwarzania dokumentów w kontekście konkretnego czatu. Myślę, że zapewnia on większość ustawień przetwarzania dokumentów ze wszystkich porównywanych interfejsów. Dostępne jest również wyszukiwanie w sieci — musisz zaznaczyć pole wyboru w panelu administracyjnym, wybrać „wyszukiwanie w sieci” i wybrać wyszukiwarkę. Następnie użyj „+” podczas tworzenia nowego monitu, aby to zadziałało. Miałem pewne problemy z DuckDuckGo domyślnie, ale inne usługi działały prawidłowo. Jeśli chodzi o korzystanie z innych łączników danych, którymi byłem zainteresowany, **OpenWebUI** ich nie zapewnia. Ale kiedy przejdziesz do obszaru roboczego na górze, masz zakładki, modele, wiedzę, monity, narzędzia i funkcje. A kiedy klikniesz odkryj narzędzie lub jakikolwiek inny monit, zostaniesz przekierowany na stronę, na której możesz pobrać wtyczki stworzone przez społeczność. Chociaż nie znalazłem wtyczek, których potrzebowałem (Git i SQL), więc traktowałbym ten przypadek tak samo jak w LM Studio — jeśli potrzebne jest kodowanie, to nie jest ono dostępne. Ale są tam inne wtyczki, które mogą okazać się przydatne — i dla łatwej instalacji polecam ten link https://docs.openwebui.com/features/plugin/

Przydatne integracje i API zgodne z OpenAI

Tutaj chciałbym porównać, jakie inne możliwości zapewnia każdy interfejs użytkownika. Nie ma tu wiele do dodania, ale jest kilka zaskakujących elementów, które dla niektórych osób mogą być przełomowe. Próbowałem skupić się na integracji z przeglądarką internetową, pakietem Visual Studio Code i Office oraz Thunderbird.

W przypadku każdej z tych integracji musisz połączyć się za pośrednictwem zgodnego z OpenAI interfejsu API udostępnianego przez interfejs lub bezpośrednio połączyć się z usługą Ollama. I chociaż każde testowane przeze mnie narzędzie zapewnia zgodne z OpenAI API, istnieje problem po drugiej stronie tego równania — niewiele wtyczek próbuje wykorzystać to API, a jeśli już próbowały, to przeważnie nie działały (na przykład w logach ze stycznia widziałem, że wywołania były wykonywane, ale były nieprawidłowe i niezgodne z OpenAI API). Tak więc, chociaż implementacje OpenAI API nie działają (co było rozwiązaniem, którego chciałem), nadal możemy używać połączenia Ollama — na razie jedyne rozwiązanie, które działa bez dodatkowych komplikacji.

Jak już wspomniałem, każdy testowany przeze mnie interfejs zapewnia zgodne z

OpenAI API. W tym aspekcie AnythingLLM zapewnia niestandardowy punkt końcowy (warto jednak zauważyć, że ich API zapewnia dodatkowe punkty końcowe, których można użyć do połączenia z usługą AnythingLLM), LM Studio wykonuje po prostu dobrą robotę, Jan ma przeglądarkę dzienników, która najbardziej mi się podobała, a GPT4All i OpenWebUI nie zapewniają łatwo dostępnych dzienników za pośrednictwem interfejsu użytkownika.

W przypadku połączenia z przeglądarką internetową tylko AnythingLLM zapewnia coś, co jest gotowe od razu. Jest to wtyczka do przeglądarek opartych na Chromium. Żaden inny interfejs nie zapewnia czegoś takiego, ale możesz połączyć się z dowolnym z nich, używając wtyczki Page Assist, która jest dostępna zarówno dla przeglądarek opartych na Chromium, jak i dla przeglądarek Firefox https://github.com/n4ze3m/page-assist – powiedziałbym, że to moje wielkie odkrycie w trakcie tych badań, bardzo mi się podobało.

W przypadku Visual Studio Code jedyną natywną integrację, jaką znalazłem, była integracja z LM Studio za pomocą wtyczki CodeGPT. Wszystkie inne wtyczki, które znalazłem, albo nie łączyły się z API, albo miały jakieś inne problemy, ale niektóre z nich działały prawidłowo z Ollamą. Nie będę polecał żadnej z wtyczek VSC, ponieważ jest ich wiele i możesz znaleźć taką, która najlepiej odpowiada Twoim potrzebom. Chociaż możesz również przetestować niektóre z wtyczek, które przetestowałem: CodeGPT, Privy, vscode-openai, llm-vscode, Local AI Pilot i kilka z „Ollama” w nazwie. Więc moim zdaniem dla VSC – nie zawracaj sobie głowy zgodnością API OpenAI, ale połącz się z Ollamą.

Do pakietu Office znalazłem dodatek TextCraft (https://github.com/suncloudsmoon/TextCraft). Ale poza tym każda inna wtyczka, którą wypróbowałem, jest albo płatna, albo wymaga rejestracji, albo obu, nie działają lokalnie, więc nie są zgodne ze standardem konfiguracji, który chciałem osiągnąć. A problem z TextCraft jest taki, że po prostu w ogóle nie działa z moją konfiguracją – nie jestem pewien, dlaczego – więc może zadziała z twoją konfiguracją.

Ostatni dodatek, który mogę polecić to ThunderAI – https://addons.thunderbird.net/en-US/thunderbird/addon/thunderai/ – jest to dodatek do Thunderbirda, który może łączyć się bezpośrednio z Ollama i standardowymi operacjami serwera, takimi jak: przepisywanie, podsumowywanie lub tłumaczenie. Co jest w nim świetne, to to, że możesz dodawać nowe monity w swoim języku, więc nawet jeśli masz skrzynki pocztowe uruchomione w różnych

językach (a ja tak mam), możesz łatwo sprawić, aby działał dla ciebie. Aby to działało, pamiętaj tylko o ustawieniu OLLAMA_ORIGINS zgodnie z opisem w sekcji “Ważne informacje” bez niego nie mogłem połączyć się z Ollamą.

UX, stabilność i licencjonowanie interfejsów LLM

AnythingLLM wydaje się być przyjazny dla użytkownika, ale jakoś zawodzi. Na początku wydaje się, że jest prosty w użyciu. Ale kiedy próbujesz zaawansowanych konfiguracji, mają one tendencję do bycia opisywanymi jako sprzeczne z intuicją – i to samo dotyczy rozmieszczenia przycisków i zapisywania. Wyobrażam sobie, że to kwestia przyzwyczajenia się, ale jeśli dodasz do tego, że AnythingLLM jest jedynym programem, który dla mnie regularnie rozwala (szczególnie podczas przesyłania plików), staje się to naprawdę denerwujące. Dla programisty powinno być zrozumiałe, chociaż jeśli coś zawiedzie, pojawiają się małe logi – czasami ten program zawodzi bez pozostawiania śladów. Licencja wydaje się być licencją MIT, co jest bardzo dobre.

LM Studio może wydawać się przytłaczające na początku, ale możesz również przełączyć się na widok klienta, który jest nieco prostszy. Tworzy to takie poczucie, że powinno być używane przez bardziej zaawansowanych programistów i myślę, że to właśnie chcą osiągnąć w produkcie końcowym. Możesz uzyskać dostęp do większej liczby logów, ale trochę niepokojącą rzeczą jest to, że masz wiele mieszanych licencji na korzystanie z tego oprogramowania i warunki korzystania z usługi, które mogą się zmienić w dowolnym momencie. Z tego powodu nie powiedziałbym, że jest to rozwiązanie w 100 procentach przyjazne dla open source, mimo że masz biblioteki dostępne w GitHub.

Następnie masz Jan, który moim zdaniem jest najłatwiejszy w użyciu i bardzo stabilny, ale nie jest zbyt zorientowany na programistów. Powiedziałbym, że nawet jeśli masz czytelne logi serwera, to ma być najprostszy w użyciu dla użytkowników nietechnicznych. Licencja jest również bardzo liberalna, ponieważ jest to AGPL.

GPT4All ma bardzo ładną estetykę i jest łatwy w użyciu – nie tak łatwy jak Jan, ale nadal bardzo łatwy. Dla przyjazności dla programistów nie ma tak łatwo dostępnych logów, jak mógłby. Poza tym używają licencji MIT, co jest bardzo dobre i nie ogranicza cię w niczym.

OpenWebUI jest najtrudniejszy w użyciu ze wszystkich, które testowałem – to znaczy do skonfigurowania i ustawienia. Ale gdy jest w pełni skonfigurowany, jest

dość łatwy w użyciu i wydajny. Nie masz dostępu do logów za pośrednictwem interfejsu użytkownika, ale jest zdecydowanie zorientowany na programistów, co jest szczególnie widoczne w funkcjonalnościach wtyczki. Korzysta z licencji MIT, więc jeszcze raz bardzo dobrze.

Którego lokalnego oprogramowania LLM powinienem użyć?

	AnythingLLM	LM Studio	Jan	GPT4all	OpenWebUI
ZARZĄDZANIE MODELAMI					
Dostępny instalator	Tak	Tak	Tak	Tak	Nie
Integracje z dostawcami modeli	Wielu dostawców, ale brak faktycznego interfejsu użytkownika wyszukiwania	Tylko HuggingFace, ale otrzymujesz ładny widok wyników	Wielu dostawców i bardzo przyjazne dla użytkownika wyszukiwanie	Łatwe do znalezienia i ładnie przedstawione parametry modelu	Nie, tylko Ollama
Integracje z dostawcami komercyjnymi	Tak, wielu	Nie	Tak, wielu	Nie	Zgodny z OpenAI
Łatwe zarządzanie folderami i czatami	Tak	Tak	Czaty bez grup	Czaty bez grup	Tak
PLIKI NA CZACIE					
Wyjście obrazów	Nie – nie znaleziono	Nie – nie znaleziono	Nie – nie znaleziono	Nie – nie znaleziono	Tak, przez ComfyUI lub komercyjny DALL-E
Akceptowanie dokumentów tekstowych	Czasami się zawiesza i trzeba ponownie uruchomić	Nie napotkano żadnych problemów	Przesyłanie dokumentów jest obecnie całkowicie przerwane	Tylko pliki XLSX, nie wiem dlaczego	Czasami wolno, ale działa
Interpretowanie obrazów i możliwe OCR	Tak	Tak	Nie można załadować do czatu	Nie można załadować do czatu	Tak

	AnythingLLM	LM Studio	Jan	GPT4all	OpenWebUI
Transkrypcja audio	Nie	Nie	Nie można załadować do czatu	Nie można załadować do czatu	Powinno być możliwe za pomocą narzędzi
INFORMACJE ZEWNĘTRZNE					
Analiza porównawcza różnych dokumentów	Nie, ograniczone przez możliwości LLM	Nie, ograniczone przez możliwości LLM	Nie, ograniczone przez możliwości LLM capabilities	Nie, ograniczone przez możliwości LLM	Nie, ograniczone przez możliwości LLM
Wiele dokumentów, które można traktować jako RAG	Tak, niezłe zarządzanie dokumentami	Nie	Nie	Tak, bardzo przyjazny dla użytkownika i z kontekstem czatu	Tak
Można wyszukiwać informacje online	Tak	Nie	Nie	Nie	Tak
Dostępne inne połączenia	Tak, natywnie	Nie	Nie	Nie	Tak, niektóre, z wtyczkami
INTEGRACJE I API					
Możliwość obsługi API zgodnego z OpenAI	Tak, niestandardowy punkt końcowy	Tak	Tak, z ładnym logowaniem	Tak, ale ukryte logi	Tak, ale ukryte logi
Integracja z przeglądarką internetową	Wtyczka dedykowana Chromium	Nie, ale Page Assist istnieje	Nie, ale Page Assist istnieje	Nie, ale Page Assist istnieje	Nie, ale Page Assist istnieje
Visual Studio Code	Nie przez API OpenAI, ale do Ollama	Natywny przy użyciu CodeGPT	Nie przez API OpenAI, ale do Ollama	Nie przez API OpenAI, ale do Ollama	Nie przez API OpenAI, ale do Ollama
Integracja z Office	Nie	Nie	Nie	Nie	Nie
INNE ELEMENTY					

	AnythingLLM	LM Studio	Jan	GPT4all	OpenWebUI
UX i stabilność	Próbuje być przyjazny dla użytkownika, ale taki nie jest, pełen błędów, niezbyt stabilny	Może wydawać się przytłaczający na początku, ale jest „widok klienta” aswell	Najłatwiejszy w użyciu, stabilny	Łatwy w użyciu i ładny	Niełatwy w konfiguracji, ale łatwy w użyciu, gdy już jest gotowy
Przyjazny dla programistów	Niektóre funkcjonalności są sprzeczne z intuicją, ale istnieje rozbudowana dokumentacja. Małe logi	Więcej logów i na pewno rozwiązanie zorientowane na deweloperów	Niezbyt zorientowane na deweloperów, ale czytelne logi serwera	Brak łatwo dostępnych logów	Brak logów za pośrednictwem interfejsu użytkownika, ale zdecydowanie zorientowane na deweloperów
Licencja	MIT	Mieszane + ToS	AGPL	MIT	MIT

Moje jednozdaniowe podsumowanie każdego testowanego oprogramowania brzmiałoby następująco:

- Najbardziej niepokojącą rzeczą w LM Studio jest ich licencja i dlatego bym go nie używał – poza tym, że wygląda na bardzo dobre oprogramowanie.
- Jeśli szukasz czegoś, co jest tak łatwe w użyciu, jak to tylko możliwe – poleciłbym Jan. Po prostu zadziała.
- Jeśli z jakiegoś powodu Jan ci się nie podoba – wybierz GPT4All – wydaje się, że po naprawieniu kilku rzeczy może być dość silnym konkurentem.
- AnythingLLM jest tym, który jest pełen funkcji, ale jest poważny problem ze stabilnością, a pobieranie modeli można by ulepszyć – więc jest to dobry wybór, jeśli czujesz się odważny lub jeśli działa stabilnie w Twoim środowisku.
- Jeśli nic z powyższych Cię nie satysfakcjonuje i nie boisz się od czasu do czasu trochę kodować (lub szukasz rozwoju LLM) – wybierz OpenWebUI, a odwdzięczy Ci się rozszerzalnością.

Mam nadzieję, że spodobało Ci się to porównanie i że trochę rozjaśniłem skomplikowane pojęcie korzystania z asystentów LLM offline (jakby sam świat LLM nie był wystarczająco skomplikowany). Być może pomoże Ci to nie tylko lepiej zrozumieć, jak działają LLM i AI, ale także lepiej je wykorzystać, nawet offline – zarówno do użytku komercyjnego, jak i do usprawnienia własnego przepływu pracy.

A jeśli chcesz wdrożyć rozwiązanie AI do swojego oprogramowania SaaS – to właśnie znalazłeś firmę, która może Ci w tym pomóc – po prostu użyj poniższego formularza kontaktowego i skontaktuj się z nami!