Tworzenie z AI

OD POMYSLU DO PRODUKCJI

Kompletny przewodnik all-in-one do zbudowania wlasnej strony internetowej, aplikacji lub startupu od zera. Opanuj agentow AI i narzedzia, zintegruj platnosci, wdroz serwer i poznaj triki marketingowe na organiczny wzrost.

apifreellm.com

Wersja 1.0 — Luty 2026

Spis Tresci

1. Wprowadzenie

> *Czytasz to teraz, poniewaz gdzies, w jakis sposob, kombinacja strategii marketingowych, technik SEO i sprytnego pozycjonowania sprawila, ze ten kurs trafil na Twoj ekran. Juz samo to powinno Ci cos powiedziec: taktyki opisane w tym przewodniku naprawde dzialaja. I tak, nauczysz sie kazdej z nich.*

Swiat sie zmienil

Tworzenie oprogramowania nie jest juz takie, jak kiedys. Jeszcze kilka lat temu zbudowanie aplikacji internetowej wymagalo miesiecy pracy, zespolu programistow i znacznego budzetu. Dzisiaj jedna osoba z odpowiednimi narzedziami i odpowiednia wiedza moze zbudowac i uruchomic aplikacje gotowa do produkcji w kilka dni, a nie miesiecy.

Ten kurs zostal napisany przez profesjonalistow, ktorzy pracuja w branzy i codziennie uzywaja agentowych narzedzi AI do budowania prawdziwych produktow. Nie uczymy teorii z podrecznika. Uczymy tego, co naprawde dziala w praktyce, tu i teraz.

Dlaczego ten kurs powstal

AI i LLM sa teraz lepszymi i szybszymi wykonawcami niz ludzie. Potrafia pisac kod, debugowac go, refaktoryzowac i wdrazac z predkoscia, ktorej zaden czlowiek nie dorovna. Ale jest cos, czego im zasadniczo brakuje: **pomyslow**.

> Modele AI to nadzwyczajni wykonawcy, ale nie sa innowatorami. Nie widza luki na rynku i nie mysla "moglbym zbudowac cos, co to naprawi". To Twoja rola. Twoje zadanie to byc architektem pomyslow. AI jest Twoim budowniczym.

Kiedy dasz LLM slabe instrukcje, i tak cos wytworzy. Nie zatrzyma sie, zeby wyjasnic, co naprawde byloby lepsze. Jakosc tego, co budujesz, jest wprost proporcjonalna do jakosci Twojej wiedzy. Ten kurs wypelnia te luke.

To nie jest tylko kurs programowania. To kompletny plan przejscia od pomyslu do dzialajacego produktu generujacego przychody. Lacznie z niezwykle skutecznymi taktykami marketingowymi i SEO — tymi samymi technikami, ktore sprawily, ze trafiles na te strone.

2. Stos technologiczny AI-First

> *Narzedzia, ktore wybierzesz, zdecyduja o tym, jak szybko bedziesz sie poruszac. W tym rozdziale omawiamy dostepne dzisiaj agenty AI do kodowania, pomagamy wybrac odpowiedniego i uczymy strategii promptowania, ktore odrozniaja amatorow od profesjonalistow.*

> **Uwaga:** Ten przewodnik napisano z uzyciem Windowsa, ale wszystkie narzedzia i kroki dzialaja identycznie na macOS i Linuxie. Google Antigravity, Claude Code i kazda inna aplikacja omawiana w tym kursie sa w pelni wieloplatformowe. Jesli uzywasz Maca lub Linuxa, po prostu wykonuj te same kroki — interfejsy i procesy pracy sa takie same.

Krajobraz agentow AI

Rynek narzedzi AI do kodowania eksplodowal. Ale nie wszystkie narzedzia sa sobie rowne. Niektore to gloryfikowane silniki autouzupelniania. Inne to w pelni autonomiczni agenci, ktorzy potrafia przeczytac cala baze kodu, zaplanowac strategie, wykonac zmiany w wielu plikach, uruchomic testy i naprawic bledy samodzielnie. Zrozumienie tej roznicy jest kluczowe.

Istnieja dwie podstawowe kategorie narzedzi AI do kodowania:

- **Asystenci inline** — Siedza w Twoim edytorze i sugeruja kod podczas pisania. Pomysl o oryginalnym GitHub Copilot. Sa reaktywni: czekaja, az zaczniesz pisac, a nastepnie probuja zgadnac, co dalej. Przydatne, ale ograniczone.
- **Narzedzia agentowe** — To zupelnie inna liga. Dajesz im zadanie w jezyku naturalnym, a oni autonomicznie planuja, pisza, edytuja, debuguja i iteruja. Nie sugeruja jednej linijki kodu. Buduja funkcjonalnosci. Tu tkwi prawdziwa sila.

Oto narzedzia, ktore teraz sie licza:

Claude Code (od Anthropic)

Claude Code jest, z naszego doswiadczenia, najpotezniejszym agentem AI do kodowania dostepnym dzisiaj. To agent terminalowy, ktory dziala bezposrednio w katalogu Twojego projektu. Dajesz mu instrukcje w jezyku naturalnym, a on czyta Twoje pliki, pisze kod, wykonuje komendy, tworzy commity i naprawia bledy autonomicznie. Ma pelny dostep do Twojego systemu plikow i shella, co czyni go niesamowicie skutecznym w prawdziwej pracy programistycznej. Mozesz go zainstalowac przez npm (`npm install -g @anthropic-ai/claude-code`) lub uzywac jako rozszerzenia VS Code, co omowimy wkrotce.

Google Antigravity

Antigravity to srodowisko programistyczne od Google, ktore wprowadza czat i agentowe mozliwosci AI bezposrednio do Twojego procesu pracy. Pomysl o nim jak o inteligentnym workspace, w ktorym mozesz wchodzic w interakcje z agentami AI przez interfejsy czatowe, a z kazdej rozmowy z agentem mozesz otworzyc zintegrowany edytor VS Code. To kluczowe: Antigravity daje Ci warstwe konwersacyjnego AI, podczas gdy VS Code zapewnia moc edycji kodu. Polaczenie jest bezszwowe — omawiasz z agentem, co zbudowac, a nastepnie przeskakujesz prosto do kodu bez przelaczania okien.

Cursor

Cursor to fork VS Code z gleboko zintegrowanym AI. Ma zarowno sugestie inline, jak i agentowy tryb "Composer", w ktorym mozesz opisac zmiany w wielu plikach. Interfejs jest znajomy, jesli juz uzywasz VS Code, i obsluguje wiele modeli (Claude, GPT itp.). To solidne narzedzie, szczegolnie jesli preferujesz w pelni wizualny workflow. Jednak jego mozliwosci agentowe, choc dobre, nie sa tak autonomiczne jak Claude Code. Czesto bedziesz musial przegladac i zatwierdzac poszczegolne zmiany krok po kroku.

> Nasza rekomendowana konfiguracja: **Google Antigravity + Claude Code jako rozszerzenie VS Code**. Uzyj czatow agentowych Antigravity do planowania i omawiania pracy, a nastepnie otworz zintegrowany VS Code i pozwol Claude Code zająć sie ciezka realizacja. To daje Ci to, co najlepsze z obu swiatow: inteligentna rozmowe do planowania i autonomiczne wykonywanie kodu do budowania.

Wybor i konfiguracja agenta

Instalacja i uruchomienie narzedzia agentowego to latwa czesc. Wyciagniecie z niego maksimum wymaga zrozumienia, jak dziala, wybrania odpowiedniej subskrypcji i prawidlowej konfiguracji.

Subskrypcja Claude: zacznij od Pro

Claude Code wymaga konta Anthropic. Zalecamy zaczynanie od **subskrypcji Claude Pro**, czyli podstawowego platnego planu. Jest przystepny cenowo i daje dostep do Claude Code ze wszystkimi funkcjami. To idealny punkt wyjscia do eksperymentowania, nauki workflowu i zbudowania pierwszego prostego projektu.

Pamietaj jednak, ze plan Pro ma **ograniczone uzycie**. Jesli uzywasz Claude Code intensywnie, dojdziesz do limitu stosunkowo szybko. Do nauki i malych projektow jest wiecej niz wystarczajacy. Ale jesli juz wiesz, ze chcesz uzywac Claude Code jako glownego narzedzia programistycznego i planujesz pracowac nad nim godzinami kazdego dnia, rozwaz przejscie na jeden z wyzszych planow (np. Max) od samego poczatku. Wyzsze plany oferuja znacznie wieksze limity, co oznacza mniej przerw i plynniejszy workflow przy budowaniu wiekszych projektow.

Model: Claude Opus 4.6

Obecnie zalecamy uzywanie **Claude Opus 4.6** jako glownego modelu do programowania. Jest to w tej chwili najlepszy model do budowania stron internetowych i aplikacji. Rozumie zlozone architektury, pisze czysty kod gotowy do produkcji, obsluguje zmiany w wielu plikach z niezwykla dokladnoscia i rzadko wymaga poprawek przy pierwszym podejsciu. Pracujac z Claude Code, ustaw Opus 4.6 jako domyslny model. Roznica w jakosci wynikow w porownaniu z mniejszymi modelami jest natychmiast zauważalna.

Konfiguracja Antigravity

Od tego momentu ten przewodnik uzywa **Google Antigravity** jako glownego srodowiska programistycznego. Oto jak przygotowac wszystko.

Krok 1: Utworz folder projektu. Przed otwarciem Antigravity utworz folder na komputerze, w ktorym bedziesz przechowywac swoj pierwszy projekt. Na przyklad w Windowsie mozesz utworzyc `C:\Projects\my-first-app`, a na Mac/Linux `~/Projects/my-first-app`. To jest folder, ktory Antigravity otworzy jako workspace. Moze byc na razie pusty — wypelnimy go kodem pozniej w kursie.

Krok 2: Zainstaluj i uruchom Antigravity. Pobierz Google Antigravity, zainstaluj i otworz. Zaloguj sie kontem Google, gdy zostaniesz poproszony.

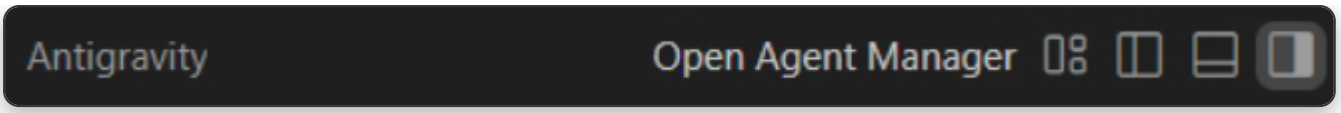
Podczas procesu instalacji Antigravity poprosi Cie o skonfigurowanie kilku zasad. Dla szybkiego, autonomicznego procesu pracy zalecamy wybranie **konfiguracji niestandardowej** i ustawienie tych opcji:

- **Terminal Execution Policy** → Always Proceed
- **Review Policy** → Always Proceed
- **JavaScript Execution** → Always Proceed

Z tymi ustawieniami agenci Antigravity beda mogli wykonywac komendy, zapisywac pliki i uruchamiac kod autonomicznie, bez pytania o potwierdzenie na kazdym kroku. To wlasnie umozliwia szybkie, plynne doswiadczenie programistyczne, do ktorego dazymy w tym kursie.

> **Ostrzezenie dotyczace bezpieczenstwa:** Kiedy pozwalasz agentom dzialac autonomicznie, zarowno Antigravity, jak i Claude Code moga wykonywac akcje w Twoim systemie bez recznego zatwierdzania. To oznacza, ze musisz byc swiadomy tego, na co ich narazasz. Nie kieruj agentow na bazy kodu, ktore moga zawierac malware, i zachowaj ostroznosc wobec linkow lub zasobow z niezaufanych zrodel. W erze AI pojawiaja sie nowe wektory atakow — zlosliwi aktorzy moga tworzyc tresci specjalnie zaprojektowane, by oszukac LLM i sklonic je do wykonywania szkodliwych komend. Zawsze obserwuj, co Twoi agenci robia. Zalecamy konfiguracje "Always Proceed" dla szybkosci, ale badz czujny i regularnie sprawdzaj wyniki.

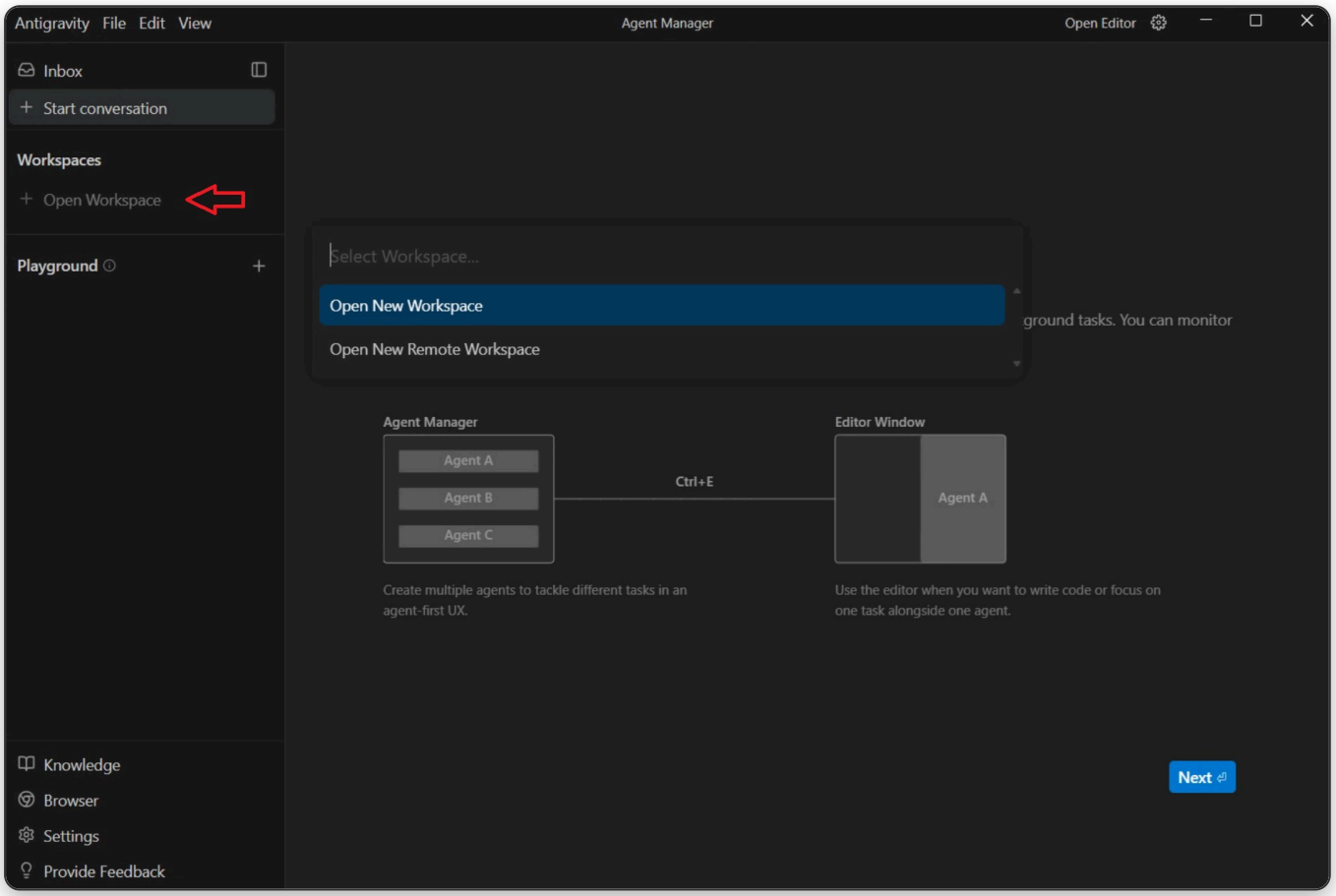
Krok 3: Otworz Agent Manager. Gdy Antigravity jest uruchomiony, kliknij **"Open Agent Manager"** na gornym pasku okna.



Przycisk "Open Agent Manager" na gornym pasku Antigravity.

Agent Manager to centralne centrum Antigravity. Tutaj zarzadzasz projektami, rozpoczynasz rozmowy z AI i otwierasz swoje przestrzenie robocze.

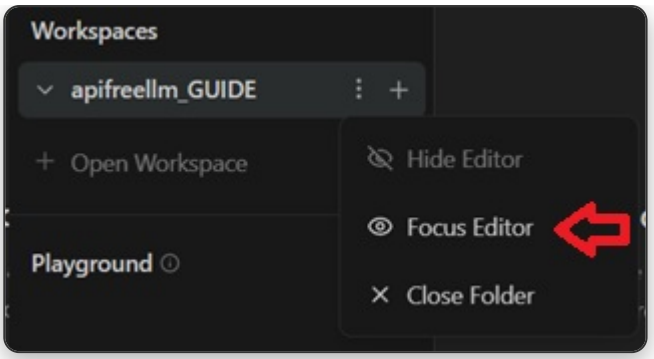
Krok 4: Utworz swoj pierwszy workspace. W Agent Managerze spójrz na lewy pasek boczny. W sekcji **"Workspaces"** kliknij **"+ Open Workspace"**. Pojawi sie menu rozwijane — wybierz **"Open New Workspace"**.



Kliknij "+ Open Workspace" w lewym pasku bocznym, a nastepnie wybierz "Open New Workspace".

Antigravity poprosi Cie o wybranie folderu. Przejdz do folderu projektu, ktory utworzyles w Kroku 1, i wybierz go. Twoj nowy workspace pojawi sie w lewym pasku bocznym pod "Workspaces" z nazwa wybranego folderu. Pomysl o kazdym workspace jako o indywidualnej sesji programistycznej — jedna na projekt. Mozesz tworzyc tyle workspace'ow, ile potrzebujesz, i przelaczac sie miedzy nimi z Agent Managera w dowolnym momencie.

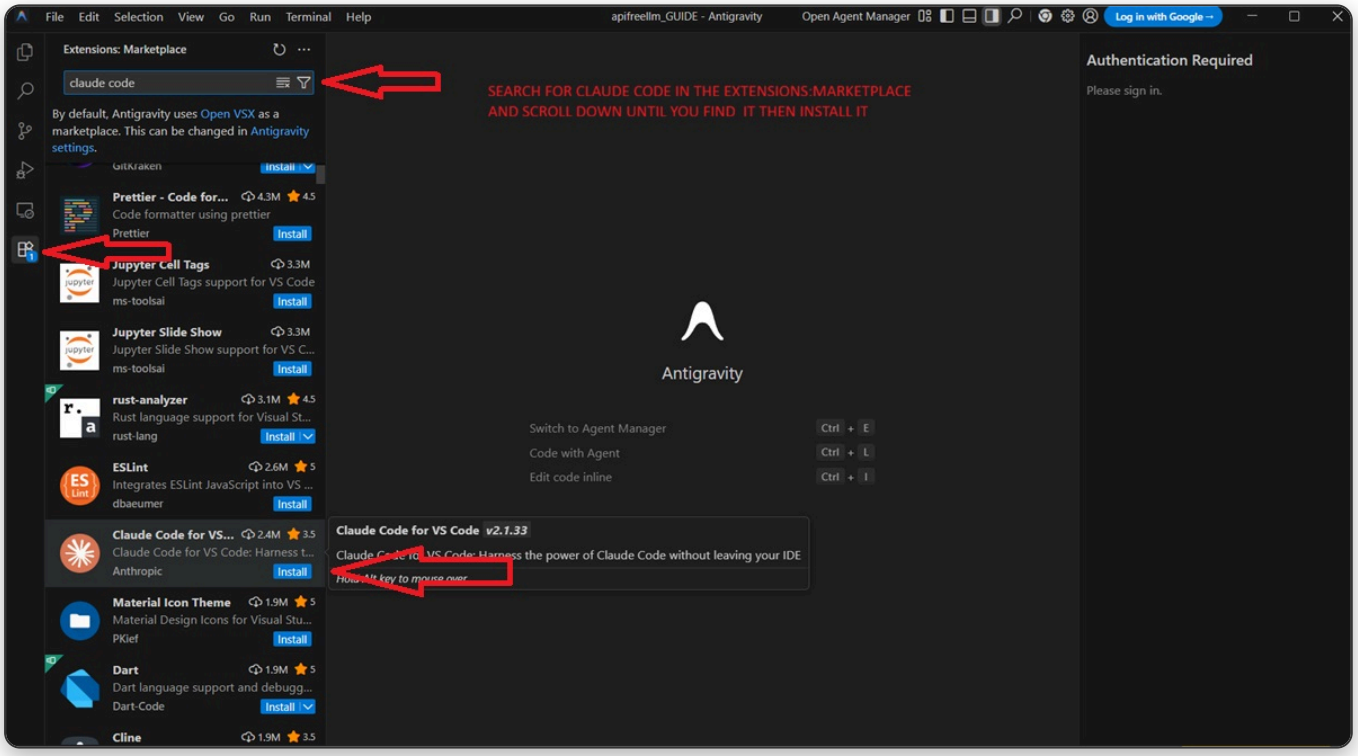
Krok 5: Otworz edytor. Gdy workspace jest utworzony, zobaczysz jego nazwe w pasku bocznym. Kliknij **trzy pionowe kropki** (:) obok nazwy workspace'a, a nastepnie wybierz **"Focus Editor"**. To otwiera pelne srodowisko VS Code dla tego workspace'a, w ktorym bedziesz pisac i edytowac kod.



Kliknij trzy kropki obok nazwy workspace'a i wybierz "Focus Editor".

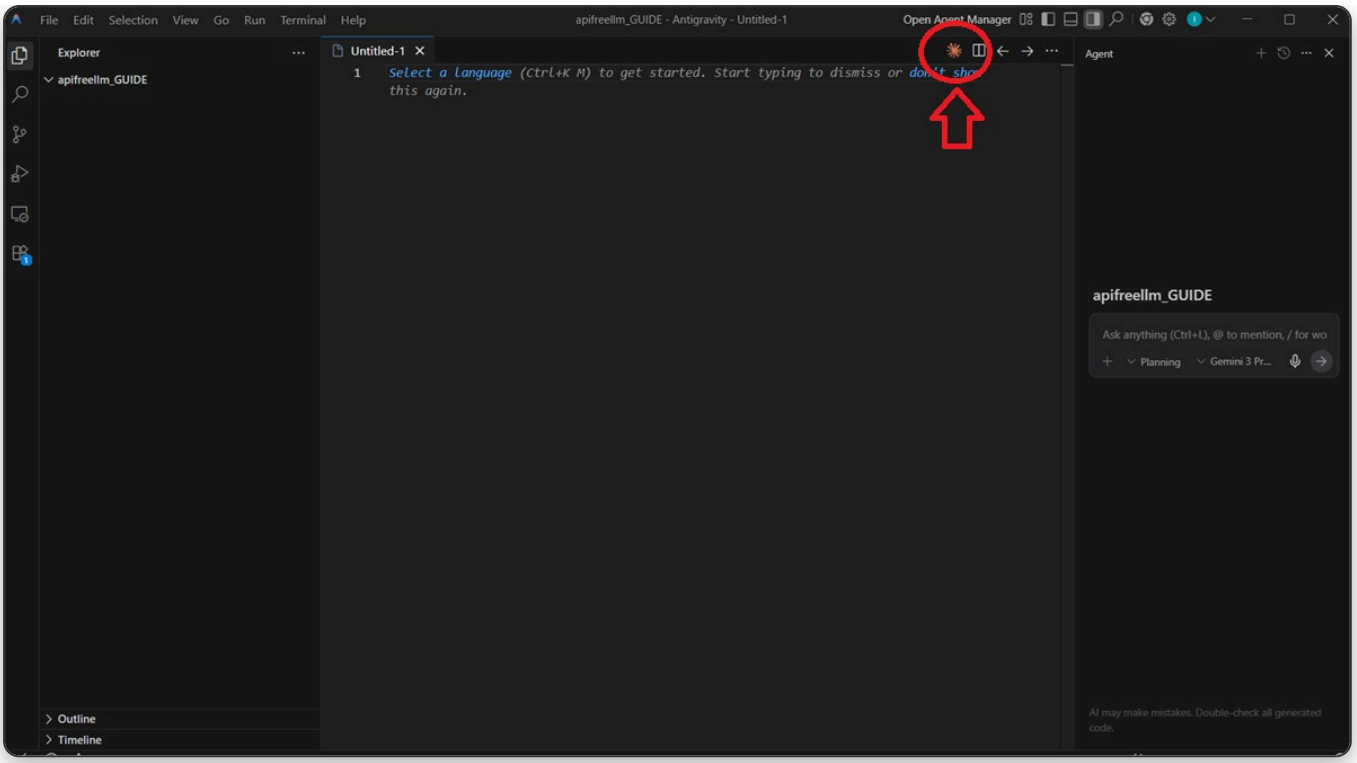
Instalacja Claude Code jako rozszerzenia VS Code

Teraz, gdy masz otwarty edytor, pora zainstalowac Claude Code. Poniewaz edytor jest oparty na VS Code, masz dostep do marketplace rozszerzen. Kliknij ikone **Extensions** w lewym pasku bocznym (lub nacisnij `Ctrl+Shift+X`). W pasku wyszukiwania wpisz **"claude code"**. Moze byc konieczne przewiniecie wynikow — szukaj **"Claude Code for VS Code"** od Anthropic. Gdy je znajdziesz, kliknij przycisk **Install**.



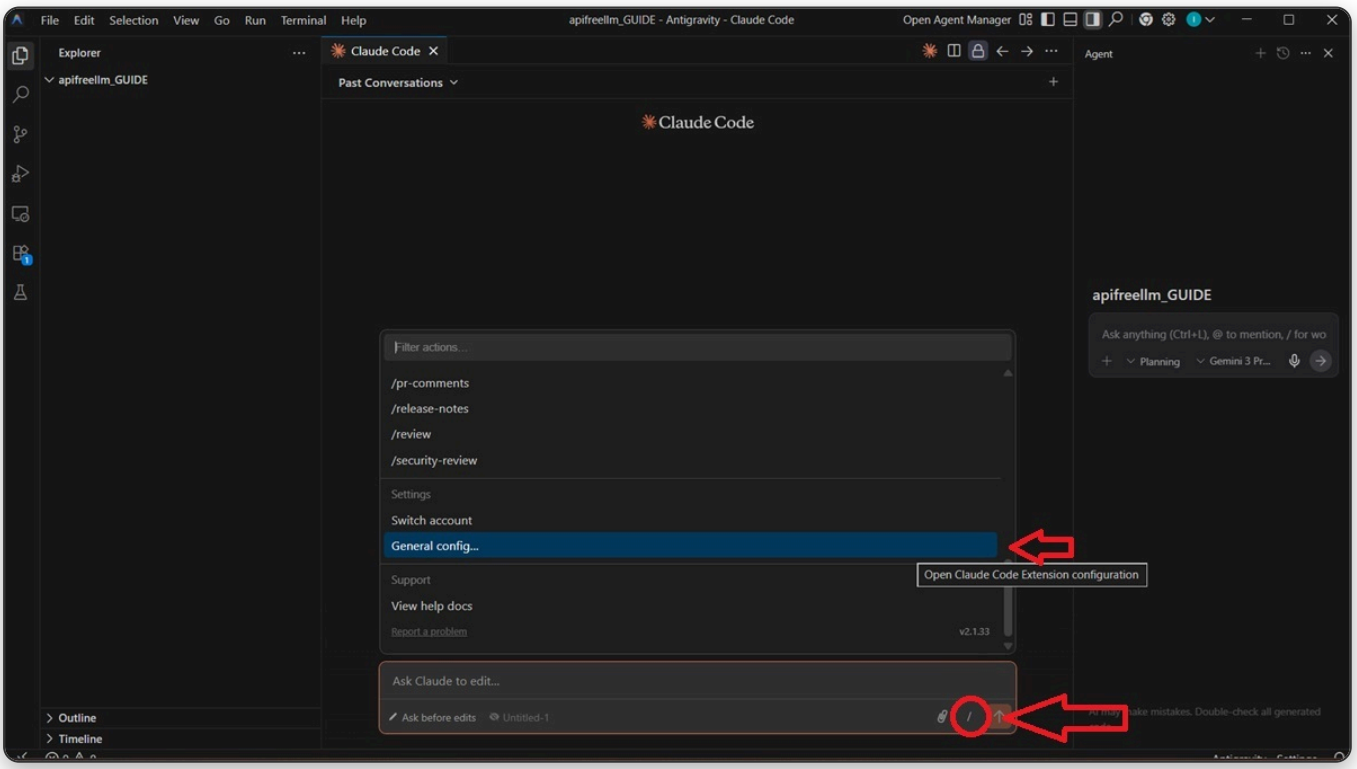
Wyszukaj "claude code" w marketplace rozszerzen, przewin w dol, aby go znalezc, i kliknij Install.

Po instalacji zamknij panel Extensions i otworz nowy plik (lub dowolny plik w projekcie). Zauważysz mala **ikone Claude Code** pojawiajaca sie w prawym gornym rogu edytora — wyglada jak maly pomaranczowy symbol. Kliknij ja, aby otworzyc panel czatu Claude Code.



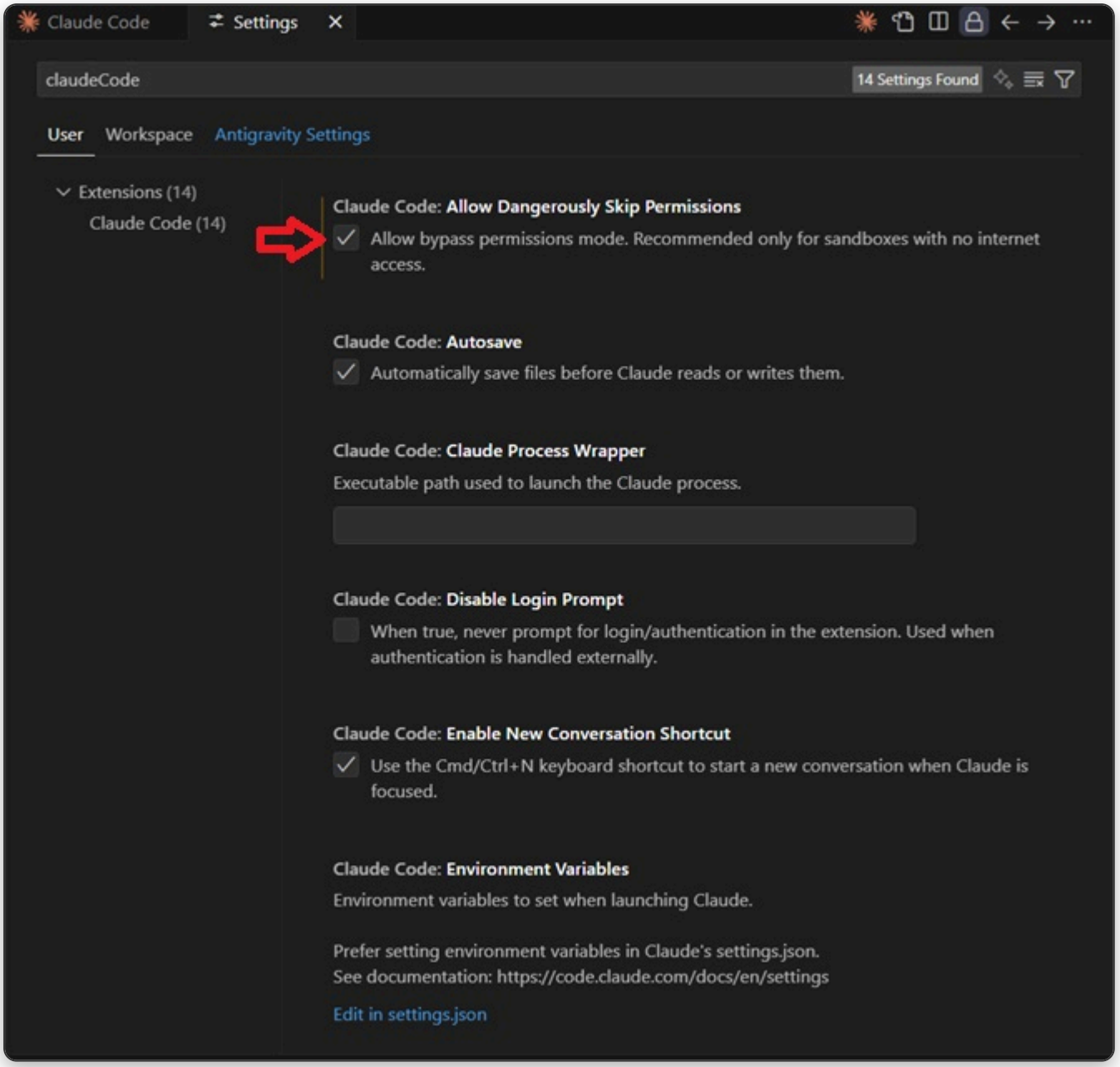
Przycisk Claude Code (zaznaczony) pojawia sie w prawym gornym rogu edytora. Kliknij go, aby otworzyc czat Claude Code.

Claude Code poprosi Cie o zalogowanie sie kontem Anthropic (tym z subskrypcja Pro). Po zalogowaniu musisz go skonfigurowac. W panelu czatu Claude Code kliknij przycisk **/** na dole panelu. Pojawi sie menu z roznymi komendami. Przewin w dol do sekcji **Settings** i kliknij **"General config..."**, aby otworzyc konfiguracje rozszerzenia Claude Code.



Kliknij przycisk "/" na dole, a nastepnie przewin do Settings i wybierz "General config...", aby otworzyc konfiguracje.

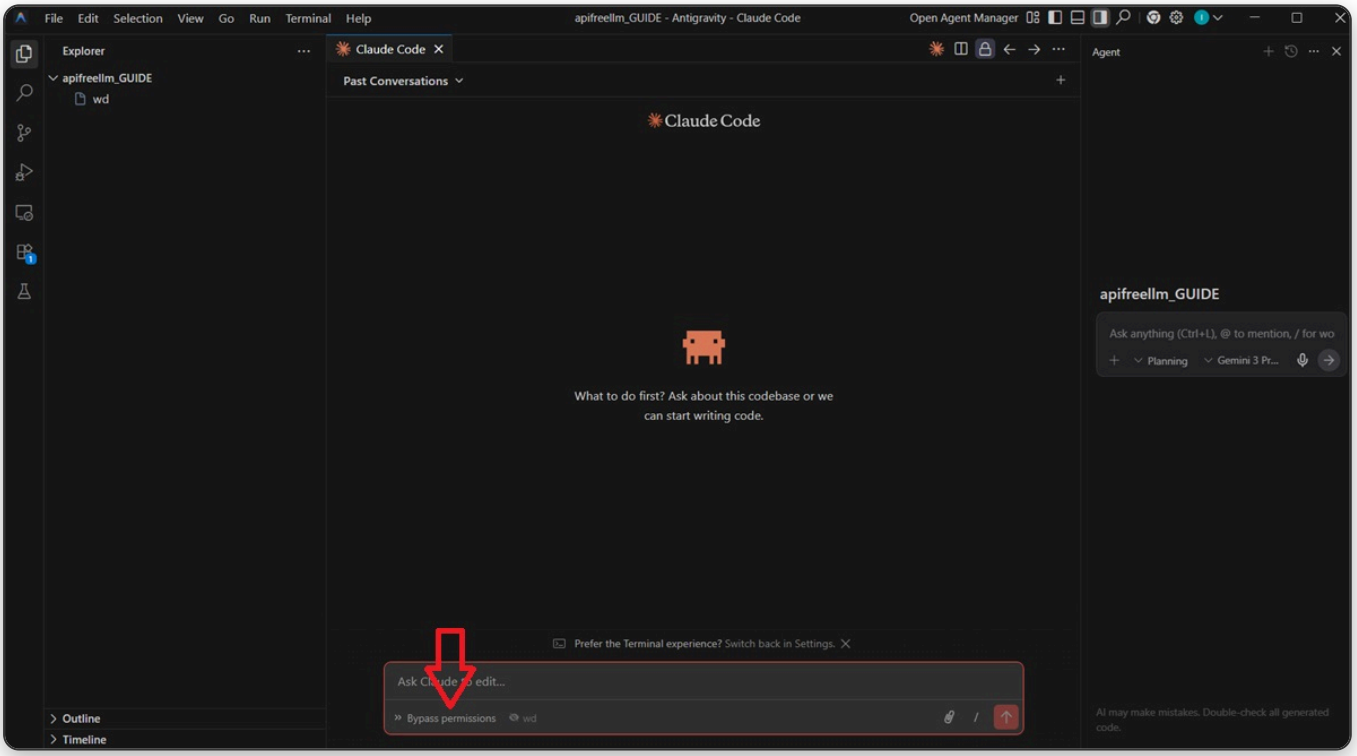
W panelu konfiguracji, ktory sie otworzy, poszukaj opcji **"Allow Dangerously Skip Permissions"**. Wlacz ten przelacznik. Po aktywacji bedziesz miec mozliwosc wybrania **"Bypass permissions"** jako trybu uprawnien, co pozwala Claude Code dzialac w pelni autonomicznie — czytac pliki, pisac kod, uruchamiac komendy terminalowe i wprowadzac zmiany bez pytania o potwierdzenie na kazdym kroku.



Wlacz przelacznik "Allow Dangerously Skip Permissions" w ustawieniach Claude Code, a nastepnie wybierz "Bypass permissions".

> **Wazne:** Z wlaczonym bypass permissions Claude Code bedzie wykonywac akcje w Twoim systemie bez recznego zatwierdzania. To niezbedne dla plynnego procesu pracy, ale wiaze sie z odpowiedzialnoscia. Zachowaj ostroznosc wobec kodu, ktory prosisz go uruchomic, i nigdy nie kieruj go na niezaufane repozytoria ani podejrzane adresy URL. Agenci AI moga byc wykorzystywani przez prompt injection — zlosliwe tresci ukryte w plikach lub na stronach, ktore oszukuja agenta i sklaniaja go do wykonywania szkodliwych komend. Zawsze sprawdzaj, co robi Claude Code, szczegolnie gdy pracujesz z zewnetrznymi zasobami.

Zalecamy rowniez wlaczenie ustawienia, ktore sprawia, ze **"Bypass permissions" jest domyslnym wyborem** dla nowych czatow, wiec nie musisz go wybierac recznie za kazdym razem, gdy rozpoczynasz nowa rozmowe. Teraz zamknij Claude Code i otworz go ponownie (klikajac ponownie przycisk ikony Claude Code). Od tego momentu zobaczysz opcje **"Bypass permissions"** dostepna w selektorze uprawnien na dole panelu czatu Claude Code. Kliknij ja, aby ja aktywowac.



Wybierz "Bypass permissions" z przycisku wskazanego na zrzucie ekranu.

Z aktywnym Bypass permissions Claude Code bedzie wykonywal komendy, tworzyc pliki, edytowac kod i uruchamiac operacje terminalowe calkowicie samodzielnie — bez zatrzymywania sie, by prosic o zatwierdzenie na kazdym kroku. To wlasnie pozwala Ci **zautomatyzowac 100% procesu programistycznego**: opisujesz, co chcesz zbudowac, a Claude Code buduje to autonomicznie od poczatku do konca. Koniec z klikaniem "Accept" na kazdej pojedynczej zmianie pliku czy wykonaniu komendy. Ty dajesz instrukcje, a agent zajmuje sie reszta.

Madre zarzadzanie uzyciem

Jedna rzecz, ktora powinienes wiedziec od poczatku: kazda interakcja z Claude Code zuzywa **tokeny**, a Twoja subskrypcja ma limit uzycia. Dobra wiadomosc jest taka, ze mozesz dokladnie

monitorowac, ile zuzyłes. W panelu czatu Claude Code kliknij przycisk **/** — wsrod dostepnych komend znajdziesz swoje **aktualne zuzycie**, pokazujace ile tokenow zuzyłes i ile pojemnosci Ci pozostalo.

Nie wszystkie modele zuzywaja tokeny w tym samym tempie. **Claude Opus 4.6** to najbardziej inteligentny i zdolny model, ale zużywa tez najwiecej tokenow na interakcje. Mniejsze modele jak **Sonnet** czy **Haiku** sa mniej potezne, ale maja wyzsze limity uzycia i zuzywaja znacznie mniej tokenow. Nasza rekomendacja: uzywaj **Opus do zlozonych zadan**, ktore wymagaja glebbokiego rozumowania, zmian w wielu plikach lub decyzji architektonicznych — tutaj jego inteligencja naprawde robi roznice. Do prostszych zadan jak szybkie poprawki, male edycje czy proste pytania, przelacz sie na lzejszy model, aby zachowac tokeny Opus na momenty, gdy naprawde maja znaczenie.

Jest jeszcze inna strategia oszczedzania tokenow Claude: **uzyj wbudowanego agenta Antigravity** do pytan, ktore nie wymagaja poziomu inteligencji Claude. Antigravity obsluguje wiele modeli, ale zalecamy uzywanie **Gemini** do tych szybkich pytan — poniewaz Antigravity jest produktem Google, Gemini ma najwyzszy limit uzycia i jest tez najszybszym modelem dostepnym na platformie. Potrzebujesz szybkiego przypomnienia o CSS? Chcesz poznac skladnie komendy Git? Ciekawi Cie, jak dziala biblioteka? Zapytaj Antigravity zamiast Claude Code. W ten sposob zachowujesz tokeny Claude na ciezka prace programistyczna, gdzie maja najwiekszy wplyw.

Jak widac na wczesniejszym zrzucie ekranu, zalecamy trzymanie **czatu Claude Code po lewej stronie** i **czatu agenta Antigravity po prawej**. Ten uklad obok siebie daje Ci natychmiastowy dostep do obu agentow w kazdej chwili.

> Madry workflow: **Opus do budowania, lzejsze modele do szybkich zadan, Antigravity do ogolnych pytan.** W ten sposob maksymalizujesz wartosc subskrypcji Claude. Jesli konczy Ci sie limit uzycia Claude, pamietaj, ze zawsze masz agenta Gemini Antigravity tuz obok do prostszych pytan — uzyj go, aby zachowac tokeny Claude na zadania programistyczne, ktore naprawde ich potrzebuja.

Niezbedna konfiguracja

Niezaleznie od tego, jak zainstalujesz Claude Code, te kroki konfiguracji dramatycznie poprawia Twoje wyniki:

- **Uzyj pliku CLAUDE.md** — Umiesc plik `CLAUDE.md` w katalogu glownym projektu. Plik ten jest automatycznie czytany przez Claude Code na poczatku kazdej sesji. Uzywaj go do opisania struktury projektu, konwencji kodowania, stosu technologicznego i wszelkich zasad, ktorych agent powinien przestrzegac. Pomysl o nim jak o dokumentacji onboardingowej dla Twojego programisty AI.
- **Utrzymuj projekt w porzadku** — Agenci AI dzialaja dramatycznie lepiej z czystymi, dobrze zorganizowanymi bazami kodu. Jesli Twoj kod to balagan, agent wyprodukuje balaganiarskie wyniki. Dobra struktura folderow, jasne konwencje nazewnictwa i spojne wzorce robia ogromna roznice.

- **Uzyj kontroli wersji** — Zawsze pracuj z zainicjalizowanym Gitem. To daje Ci siatke bezpieczenstwa. Jesli agent popelni blad, mozesz natychmiast cofnac zmiany. Pozwala to rowniez agentowi tworzyc commity za Ciebie, co jest zaskakujaco przydatne do sledzenia, co i dlaczego sie zmienilo.

Niezbedne narzedzia: Git i GitHub CLI

Zanim zaczniemy cokolwiek budowac, sa dwa narzedzia, ktore musza byc zainstalowane w Twoim systemie: **Git** i **GitHub CLI (gh)**. Sa fundamentalne dla kazdego nowoczesnego procesu programistycznego i to one pozwalaja Twoim agentom AI autonomicznie zarzadzac repozytoriami kodu.

Dlaczego Git i GitHub CLI sa wazne

Git to system kontroli wersji, ktory sledzi kazda zmiane w Twoim projekcie. To Twoja siatka bezpieczenstwa: jesli Claude Code popelni blad, mozesz natychmiast cofnac zmiany. Pozwala tez agentowi tworzyc commity, zarzadzac branchami i utrzymywac czysta historie ewolucji projektu — wszystko automatycznie.

GitHub CLI (gh) to narzedzie wiersza polecen, ktore daje bezposredni dostep do GitHub z terminala. To klucz do pelnej automatyzacji: gdy `gh` jest zainstalowany i uwierzytelniony, Claude Code moze tworzyc repozytoria, pushowac kod, zarzadzac pull requestami, konfigurowac ustawienia repozytoriow, ustawiac GitHub Actions do deploymentu, dodawac sekrety i wiele wiecej — wszystko ze swojego terminala, bez otwierania GitHub w przegladarce.

Instalacja Git i GitHub CLI

Najprostszym sposobem na zainstalowanie tych narzedzi jest **poproszenie Claude Code lub Antigravity, zeby to zrobily za Ciebie**. Po prostu powiedz agentowi: *"Zainstaluj Git i GitHub CLI na moim systemie."* Agent wykryje Twoj system operacyjny i uruchomi odpowiednie komendy instalacyjne. Na Windowsie zazwyczaj uzyje `winget` lub pobierze instalatory; na macOS uzyje `brew`; na Linuxie `apt` lub menedzera pakietow Twojego systemu.

Jesli wolisz zainstalowac je recznie, mozesz pobrac Git z oficjalnej strony, a GitHub CLI ze strony releasow na GitHub. Ale pozwolenie AI to zalatwic jest szybsze i pozwala uniknac typowych bledow instalacji.

Uwierzytelnianie GitHub CLI

Po instalacji musisz sie zalogowac, aby `gh` moglo uzyskac dostep do Twojego konta GitHub. Ponownie mozesz po prostu poprosic agenta: *"Zaloguj mnie do GitHub CLI."* Agent uruchomi `gh auth login` i przeprowadzi Cie przez proces uwierzytelniania, ktory zazwyczaj polega na otwarciu linku w przegladarce i wpisaniu kodu. Po uwierzytelnieniu agent ma pelny dostep do Twoich repozytoriow GitHub.

To zmienia zasady gry. Z uwierzytelnionym `gh` mozesz powiedziec Claude Code takie rzeczy jak: *"Utworz nowe prywatne repozytorium o nazwie my-app, zainicjalizuj projekt i wypchnij kod."* Lub pozniej: *"Skonfiguruj GitHub Action, ktory deployuje na moj serwer przy kazdym pushu do main."* Agent zajmuje sie wszystkim — tworzeniem plikow, konfigurowaniem sekretow, ustawianiem workflow — bez opuszczania edytora. Tak wyglada prawdziwa automatyzacja programowania.

3. Twoj pierwszy projekt: od zera do produkcji

> *Twoje srodowisko jest gotowe. Claude Code jest otwarty, Antigravity dziala, Git i GitHub CLI sa skonfigurowane. Czas zbudowac cos prawdziwego. Od tego momentu kurs przechodzi od konfiguracji do strategii — praktycznych technik i spostrzezen, ktore daja Ci realna przewage przy budowaniu z agentami AI.*

Nigdy nie zaczynaj od zera

To najwazniejsza rada w calym tym kursie: **nigdy nie buduj od zera.**

Choc Claude Opus jest niesamowicie zaawansowanym i inteligentnym modelem, bedzie popelniac bledy. Kazdy model AI to robi. Moze zle zrozumiec strukture projektu, uzyc przestarzalej biblioteki, stworzyc niespojny uklad plikow lub wygenerowac kod, ktory nie do konca pasuje, gdy projekt rosnie. Zaczynanie od pustego folderu oznacza, ze AI musi podjac setki decyzji bez punktu odniesienia — a niektore z tych decyzji nieuchronnie beda bledne.

Oto rzeczywistosc: **99,9% tego, co chcesz zbudowac, juz istnieje** w jakiejs formie. Czy to sklep e-commerce, dashboard SaaS, strona portfolio, aplikacja social media czy platforma rezerwacji — ktos juz zbudowal cos podobnego. A wiele z tych projektow jest open-source, dostepnych jako szablony na GitHub, gotowych do sklonowania i dostosowania.

Twoj workflow powinien zawsze zaczynac sie tak samo: **najpierw poszukaj szablonu.** Wejdz na GitHub i poszukaj projektow open-source, ktore pasuja do tego, co chcesz zbudowac. Znajdz taki, ktory jest bliski Twojej wizji, sklonuj go do folderu projektu, a nastepnie skieruj na niego Claude Code. Teraz zamiast budowac od zera, agent modyfikuje, ulepsza i dostosowuje istniejaca, dzialajaca baze kodu. Roznica w jakosci i szybkosci jest ogromna.

Szablony to nie oszustwo — to strategia. Profesjonalni programisci uzywaja boilerplate'ow i starter kitow caly czas. Nie kopiujesz czyiegos produktu. Uzywasz strukturalnego fundamentu i budujesz na nim swoj wlasny, unikalny produkt. AI dziala dramatycznie lepiej, gdy ma istniejace wzorce do nasladowania, zamiast wymyslac wszystko od nowa.

Wybor odpowiedniego stosu technologicznego

Jesli wyszukiwanie szablonow nie przyniesie rezultatow i naprawde musisz zaczac nowy projekt, wybrana technologia moze miec znaczenie — szczegolnie przy pracy z agentami AI. Niemniej nie ma jednego obowiazkowego wyboru. Najlepszy stos to ten, ktory najszybciej doprowadzi Cie do dzialajacego produktu.

Claude Code, jak wszystkie LLM, jest zasadniczo lepszy w pisaniu **kodu webowego**: HTML, CSS, JavaScript i TypeScript. To jezyk internetu i to na nim modele te byly trenowane najbardziej. Im blizej technologii webowych pozostaje Twoj projekt, tym lepiej AI bedzie dzialac.

Dla **aplikacji webowych Next.js** to doskonaly wybor, jesli uda Ci sie go znalezc. To nowoczesny framework React z wbudowanym renderowaniem po stronie serwera, co jest krytyczne dla SEO. Claude Code dziala wyjatkowo dobrze z projektami Next.js: rozumie routing oparty na plikach, trasy API, komponenty serwerowe i caly ekosystem. Na GitHub znajdziesz ogromna liczbe szablonow Next.js do praktycznie kazdego typu aplikacji.

Dla **aplikacji desktopowych** (pliki wykonywalne na Windows, macOS lub Linux), **Electron** to swietna opcja. Electron pozwala budowac aplikacje desktopowe uzywajac HTML, CSS i JavaScript — tych samych technologii webowych, w ktorych Claude sie wyroznia. Poniewaz interfejs jest w zasadzie strona internetowa renderowana wewnatrz natywnego okna, AI moze budowac piekne, funkcjonalne aplikacje desktopowe z taka sama latwoscia jak budowanie strony internetowej.

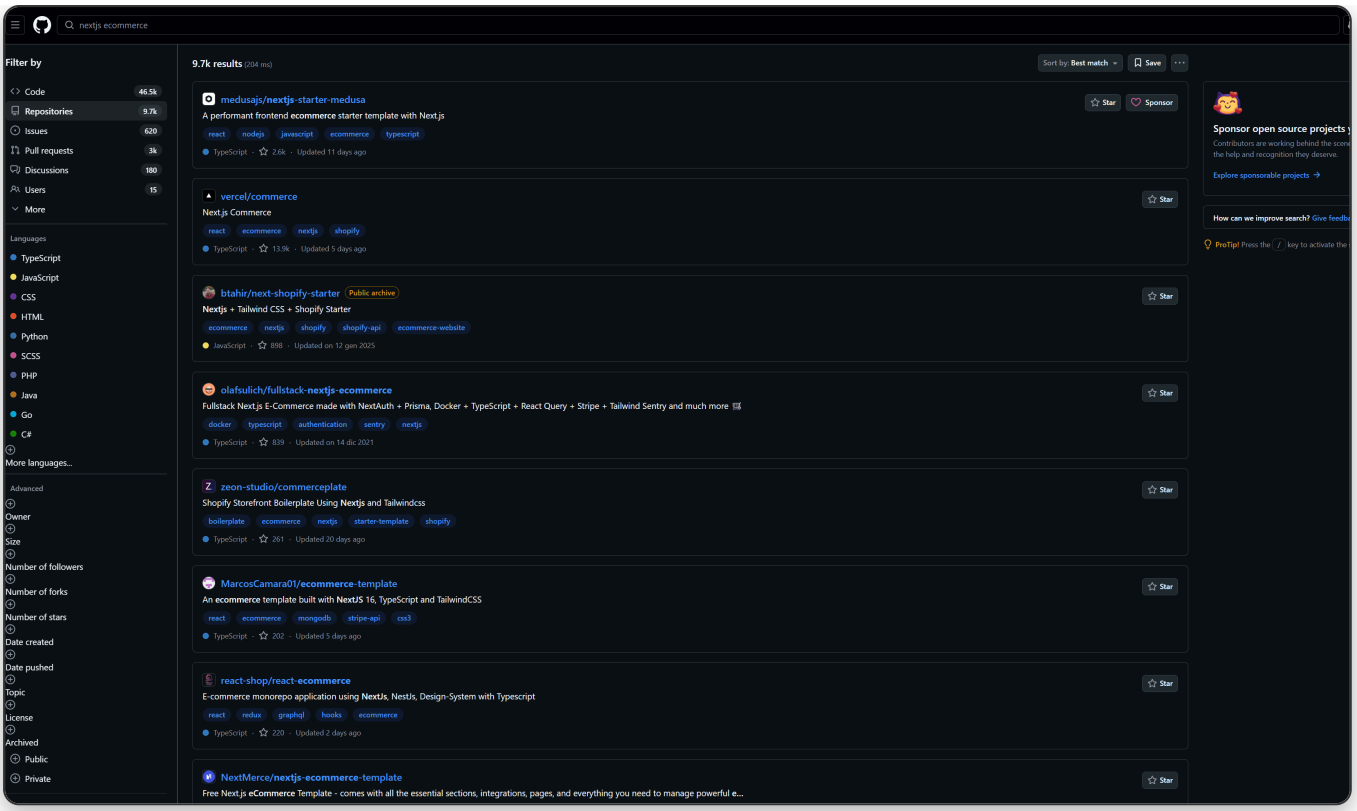
Ale oto wazny niuans: **dobrze zbudowany szablon w starszym stosie technologicznym jest prawie zawsze lepszy niz zaczynanie od zera z nowoczesnym.** Mozesz znalezc projekt w PHP, jQuery lub Laravel, ktory jest dokladnie tym, czego potrzebujesz — kompletny, dobrze ustrukturyzowany, sprawdzony w boju, ze wszystkimi funkcjami juz zaimplementowanymi. W takim przypadku uzyj go. Agenci AI moga pracowac z kazda technologia, a czas zaoszczedzony dzieki solidnemu, gotowemu fundamentowi znacznie przewyzsza teoretyczne zalety nowszego frameworka. Claude Code potrafi zrozumiec i modyfikowac kod PHP, Python, Ruby czy jakikolwiek inny bez problemu.

Priorytet jest prosty: **znajdz najlepszy dostepny szablon.** Jesli znajdziesz dwa szablony o podobnej jakosci i jeden uzywa Next.js, a drugi PHP, wybierz Next.js. Ale jesli szablon PHP jest bardziej kompletny, bogatszy w funkcje i lepiej utrzymywany — wybierz go bez wahania. Jakosc i kompletnosc punktu wyjscia sa wazniejsze niz nowoczesnosc stosu.

> **Zasada:** uzyj tego, co znajdziesz, co przyblizy Cie najbardziej do celu. Celuj w nowoczesne technologie webowe (Next.js, Electron, React Native), gdy sa dostepne, ale nigdy nie odrzucaj swietnego szablonu tylko dlatego, ze uzywa starszego stosu. Czas zaoszczedzony dzieki solidnemu fundamentowi jest zawsze cenniejszy niz czystosc stosu technologicznego.

Praktyczny przyklad: szukanie szablonu

Zalozmy, ze chcesz zbudowac sklep e-commerce. Zamiast mowic Claude Code *"Zbuduj mi sklep e-commerce od zera"*, otworz GitHub i wyszukaj cos w stylu **"nextjs ecommerce"**. Znajdziesz dziesiatki gotowych projektow z listami produktow, koszykami zakupowymi, procesami zamowien i juz zbudowana integracja platnosci.



Szybkie wyszukiwanie na GitHub "nextjs ecommerce" juz pokazuje kilka obiecujacych szablonow.

Jedna wazna rzecz do zapamietania: wiele szablonow na GitHub to **projekty freemium** — rdzen jest open-source i darmowy, ale niektore funkcje lub wersje premium wymagaja platnosci. Zawsze sprawdz README i licencje repozytorium przed wybraniem szablonu. Unikaj wszystkiego, co wymaga platnych subskrypcji lub ukrytych kosztow, ktorych nie potrzebujesz.

Patrzac na wyniki wyszukiwania, mozemy zauwazyc **fullstack-nextjs-ecommerce** — i to doskonaly punkt wyjscia. Przeanalizujmy dlaczego. Projekt uzywa Next.js z TypeScript, co jest dokladnie tym, czego chcemy do programowania wspomaganego AI. Ale to, co naprawde wyroznia, to co jest juz zintegrowane: **Stripe do platnosci**, **PostgreSQL z Prisma** jako baza danych i **NextAuth do uwierzytelniania**. To trzy najwazniejsze — i najbardziej podatne na bledy — czesci kazdej aplikacji webowej.

Posiadanie **juz zintegrowanego Stripe** jest szczegolnie cenne. Przetwarzanie platnosci obejmuje webhooki, zarzadzanie sesjami, obsluge bledow i przypadki brzegowe, ktore moga byc zaskakujaco trudne do prawidlowego zaimplementowania. Gdy cos pojdzie nie tak z platnosciami, to Twoi klienci na tym cierpia — nieudane obciazenia, podwojne platnosci lub zepsute procesy zamowien to rodzaj bledow, ktore niszcza zaufanie i kosztuja prawdziwe pieniadze. Zaczynanie od szablonu, ktory juz ma dzialajaca integracje ze Stripe, oszczedza Ci tych problemow.

Projekt uzywa rowniez **PostgreSQL** jako bazy danych, co jest solidnym wyborem. Postgres jest niezawodny, dobrze udokumentowany, oferuje nieco lepsze domyslne ustawienia bezpieczenstwa w porownaniu z alternatywami jak MySQL i moze byc latwo hostowany na Amazon AWS, Hetzner lub u podobnych dostawcow — konfiguracje serwera i deployment omowimy w nastepnym rozdziale. Fakt, ze **NextAuth** jest juz podlaczony, oznacza, ze uwierzytelnianie uzytkownikow jest obslugiwane od razu, kolejny zlozony element, ktorego nie musisz budowac od zera.

To jest sila zaczynania od odpowiedniego szablonu: zamiast spedzac dni (lub tygodnie) na konfigurowaniu platnosci, bazy danych i uwierzytelniania — wszystkiego, co latwo zrobic zle — zaczynasz od projektu, w ktorym te krytyczne systemy juz dzialaja. Mozesz wtedy skupic sie wylacznie na dostosowaniu produktu do swojej wizji: zmianie designu, dodawaniu produktow, modyfikowaniu logiki biznesowej i budowaniu funkcji, ktore czynia Twoj sklep unikalnym.

Gdy znajdziesz swoj szablon, nastepny krok jest prosty: **pobierz go i umiesc w folderze workspace.** Otworz ten folder w Claude Code (lub Antigravity) i zacznij pracowac. Mozesz powiedziec Claude Code, zeby bezposrednio modyfikowal szablon — zmienil branding, dodal nowe strony, przerobi uklad, zintegrowal nowe funkcje — lub mozesz uzyc szablonu jako **referencji** do wlasnego projektu. Nawet jesli budujesz cos nieco innego, posiadanie szablonu w tym samym workspace oznacza, ze Claude Code moze na niego spojrzec, przestudiowac wzorce kodu i uzyc ich jako fundamentu do tego, co pisze.

To kluczowy punkt: **zawsze daj Claude Code cos jako referencje.** Gdy agent ma solidna, dzialajaca baze kodu do ogladania, pisze znacznie lepszy kod. Nasleduje te same wzorce, uzywa tych samych konwencji i produkuje wyniki, ktore sa spojne i niezawodne. Gdy nie ma nic do referencji i musi generowac wszystko od zera, wtedy pojawiaja sie bledy — niespojne struktury plikow, zle wersje bibliotek, kod, ktory nie pasuje do siebie. Szablon dziala jak kotwica, ktora utrzymuje AI przy ziemi i produkuje wysokiej jakosci, koherentne wyniki.

> **A jesli szablon nie ma platnosci ani bazy danych?** To tez nie problem. Ten kurs zawiera gotowe pliki integracji ze Stripe, ktore mozesz dac bezposrednio Claude Code, aby zintegrowal platnosci w dowolnym projekcie. Dla bazy danych zalecamy PostgreSQL lub jakakolwiek baze danych, ktora szablon juz uzywa — nie walcz z wyborami szablonu, chyba ze jest ku temu mocny powod. To samo dotyczy uwierzytelniania: jesli szablon uzywa NextAuth, Clerk lub innego systemu auth, pracuj z nim. Celem jest wykorzystanie tego, co juz zbudowano, a nie zastepowanie wszystkiego.

4. Od kodu do produkcji

> *Twoj produkt jest zbudowany. Teraz musi przyjmowac platnosci, dzialac na serwerze i byc szybki i bezpieczny dla uzytkownikow na calym swiecie. Ten rozdzial obejmuje niezbedne uslugi, ktore zamieniaja Twoj projekt z lokalnego kodu w dzialajacy, gotowy do produkcji biznes.*

> **Uwaga o tym rozdziale.** Trzymamy sie zwiezlosci i konkretow. Naszym celem jest powiedziec Ci **co** musisz zrobic i **dlaczego** — nie pisac dlugich tutoriali, ktore marnuja Twoj czas. Kazdy LLM (Claude, Gemini, ChatGPT) moze wyjasnic szczegoly, przeprowadzic Cie przez kazdy krok i odpowiedziec na Twoje pytania znacznie lepiej niz statyczna strona. Kiedykolwiek cos nie jest jasne, po prostu zapytaj agenta: *"Sledze kurs i mowi, ze musze [zrobic X]. Mozesz wyjasnic, co to znaczy i przeprowadzic mnie przez to?"* To szybsze, bardziej spersonalizowane i zawsze aktualne.

Platnosci ze Stripe

Jesli Twoj produkt cokolwiek sprzedaje, potrzebujesz procesora platnosci. **Stripe** to standard branzowy: niezawodny, dobrze udokumentowany i obslugiwany przez praktycznie kazdego agenta AI ze wzgledu na to, jak szeroko jest uzywany.

Oto co musisz zrobic:

- **Zaloz konto Stripe** na stripe.com. Otrzymasz **klucze API** (klucz publiczny dla frontendu, klucz tajny dla backendu) i dostep zarowno do **trybu testowego** (fałszywe platnosci do rozwoju), jak i **trybu produkcyjnego** (prawdziwe pieniadze).
- **Skonfiguruj webhooki** w panelu Stripe. Webhooki to adresy URL na Twoim serwerze, ktore Stripe wywoluje, gdy cos sie wydarzy — platnosc sie udaje, subskrypcja sie odnawia, obciazenie nie udaje sie. W ten sposob Twoja aplikacja wie, kiedy aktywowac subskrypcje, potwierdzic zamowienie lub obsluzyc blad. Potrzebujesz webhookow zarowno do **testowania lokalnego** (Stripe ma narzedzie CLI, ktore przekierowuje zdarzenia na localhost), jak i **produkcji** (wskazujace na serwer produkcyjny). Zawsze doprowadz wszystko do dzialania lokalnie przed uruchomieniem na produkcji.
- **Zintegruj Stripe ze swoim projektem.** Jesli Twoj szablon juz ma Stripe, po prostu podlacz swoje klucze API. Jesli nie, ten kurs zawiera gotowe pliki integracji ze Stripe — wrzuc je do workspace i powiedz Claude Code, zeby je zintegrowal. Stripe obsluguje platnosci jednorazowe i subskrypcje cykliczne, oba za pomoca hostowanych stron checkout, ktore obsluguja walidacje karty, 3D Secure i zgodnosc z PCI za Ciebie.

Jedna kluczowa zasada: **zawsze weryfikuj platnosci po stronie serwera przez webhooki**, nigdy nie ufaj frontendowi. Webhook to Stripe informujacy Twoj serwer bezposrednio, ze pieniadze faktycznie zmienily wlasciciela — to jedyne wiarygodne zrodlo prawdy.

Hosting: AWS, Hetzner i inne

Twoja aplikacja musi gdzies dzialac. Dobra wiadomosc: **AWS daje Ci darmowy plan** przy rejestracji — przez **caly rok** dostajesz **serwer t2.micro** i **mikro baze danych** calkowicie za darmo. To wystarczy do hostowania pierwszego projektu, gdy weryfikujesz pomysl i zaczynasz pozyskiwac uzytkownikow.

Nie wiesz, jak zalozyc konto AWS lub uzyc darmowego planu? Po prostu zapytaj Claude lub Antigravity — przeprowadza Cie przez kazdy krok.

Gdy przerośniesz darmowy plan, oto co musisz wiedziec o doborze serwera:

- **t3.small** to optymalny punkt dla wiekszosci aplikacji — dobre CPU, wystarczajaco RAM i rozsadna cena (~16$/miesiac na AWS). Litera "t" odnosi sie do generacji instancji; starsze generacje (t2 itp.) moga czasem kosztowac wiecej za mniejsza wydajnosc, wiec trzymaj sie najnowszej dostepnej.
- **Hetzner** to europejska alternatywa, ktora jest *znacznie* tansza — mozesz uzyskac wydajnosc porownywalna z t3.small za okolo **4$/miesiac**. To mniej wiecej 4 razy taniej niz AWS za podobna specyfikacje.
- **Nie kazda aplikacja potrzebuje serwera.** Jesli Twoj projekt to strona statyczna lub aplikacja JAMstack, moze w ogole nie potrzebowac dedykowanego serwera. Platformy takie jak Vercel, Netlify czy nawet Cloudflare Pages moga ja hostowac za darmo lub prawie za darmo. Zawsze zapytaj LLM: *"Jaki hosting jest najlepszy dla mojej konkretnej aplikacji?"*

Nasza rekomendacja: **zacznij od darmowego planu AWS** na pierwszy rok, a potem ocen, czy Hetzner lub inny dostawca ma wiekszy sens dla Twojego budzetu i lokalizacji odbiorcow. Jesli wiekszosc Twoich uzytkownikow jest w Europie, niemieckie serwery Hetzner dadza Ci nizsze opoznienia. Jesli Twoja publicznosc jest globalna lub z USA, regiony AWS moga byc lepszym dopasowaniem.

> **Klucze SSH i zarzadzanie serwerem przez AI.** Aby pozwolic Claude Code laczyc sie z Twoim serwerem i zarzadzac nim zdalnie, musisz skonfigurowac **klucz SSH**. Popros Claude, zeby wygenerował jeden i skonfigurowal go na Twoim serwerze. Po polaczeniu Claude moze deployowac kod, zarzadzac uslugami, rozwiazywac problemy — wszystko z Twojego terminala. Do zadan zarzadzania serwerem zalecamy uzywanie **Opus** dla najlepszych rezultatow, choc Sonnet tez sie sprawdzi, jesli chcesz oszczedzic tokeny (z nieco wyższym ryzykiem bledow).

Cloudflare: wydajnosc i ochrona

Gdy Twoja aplikacja jest na serwerze, potrzebujesz czegos, co stanie przed nia, zeby ja chronic i przyspieszyc. To **Cloudflare**. Dobra wiadomosc: **darmowy plan jest wiecej niz wystarczajacy** dla ogromnej wiekszosci stron. Nie potrzebujesz platnego planu.

Ale najpierw bedziesz potrzebowac **nazwy domeny**. Zalecamy kupno domeny bezposrednio od **Cloudflare** lub **Namecheap** — oba sa niezawodne i uczciwie wycenione. Gdy masz domene, musisz skonfigurowac **DNS**, aby wskazywal na adres IP Twojego serwera AWS lub Hetzner. Po prostu zapytaj LLM: *"Kupilem domene na [Cloudflare/Namecheap]. Jak skonfigurowac DNS, zeby wskazywal na moj serwer pod [Twoje IP]?"* Przeprowadzi Cie przez to.

Dlaczego Cloudflare jest tak wazny? Chroni Twoja strone przed **atakami DDoS**, roznymi **exploitami bezpieczenstwa** i typowymi podatnosciami webowymi. Zapewnia tez **globalny cache**, co

oznacza, ze Twoje tresci sa serwowane z serwerow blisko Twoich uzytkownikow, przyspieszajac strone na calym swiecie. Bez uslugi takiej jak Cloudflare narazasz strone na powazne ryzyka — szczegolnie teraz, w erze AI, gdy kazdy moze uzyc narzedzi AI do znajdowania podatnosci, wykorzystywania bledow konfiguracji, a nawet kradziezy danych ze zle chronionych stron. Nie pomijaj tego.

> **Szybka wskazowka: tryb Flexible SSL.** Podczas konfiguracji Cloudflare mozesz wybrac tryb **Flexible** SSL. To oznacza, ze polaczenie miedzy Cloudflare a Twoim serwerem uzywa zwyklego HTTP, ale polaczenie miedzy Cloudflare a uzytkownikami konczowymi to HTTPS — wiec odwiedzajacy widza bezpieczna klodke. To oszczedza Ci koniecznosci konfigurowania certyfikatow SSL na serwerze, co jest swietne na poczatek. Dla aplikacji produkcyjnych, ktore obsluguja wrazliwe dane, powinienes w koncu przejsc na tryb **Full** (szyfrowanie end-to-end). Ale na pierwsze testy i uruchomienia Flexible jest calkowicie wystarczajacy i oszczedza duzo czasu na konfiguracje. Zapytaj LLM o wyjasnienie roznic, jesli nie jestes pewien, ktory tryb pasuje do Twojego projektu.

Cloudflare oferuje znacznie wiecej niz sama ochrona. Darmowy plan zawiera potezne funkcje, o ktorych wielu deweloperow nawet nie wie:

- **Cloudflare Pages** — darmowy statyczny hosting dla aplikacji frontendowych (React, Next.js static export, Vue, itd.). Pushsz na GitHub, Cloudflare automatycznie buduje i wdraza. Zero konfiguracji, zero kosztow. Idealny na landing page, portfolio i aplikacje JAMstack.
- **Edge Functions (Workers)** — uruchamiaj bezserwerowy kod na edge, blisko Twoich uzytkownikow, w darmowym planie. Swietne do tras API, przekierowan, testow A/B i lekkiej logiki backendowej bez potrzeby dedykowanego serwera.
- **CDN & Caching** — Twoje statyczne zasoby (obrazy, CSS, JS) sa cachowane globalnie, co sprawia, ze Twoja strona jest blyskawicznie szybka z kazdego miejsca na swiecie.

Kluczowe pytanie brzmi: **czy Twoja aplikacja naprawde potrzebuje dedykowanego serwera, czy Cloudflare moze ja obsluzyc za darmo?** Zapytaj Claude'a: *"Buduje [opis Twojej aplikacji]. Czy potrzebuje dedykowanego serwera na AWS/Hetzner, czy moge uzyc Cloudflare Pages i Workers za darmo?"* Claude przeanalizuje Twoj konkretny przypadek i powie Ci najlepsza opcje. Mozesz byc zaskoczony, ile projektow moze dzialac calkowicie na darmowym planie Cloudflare.

Klucze API: Automatyzuj Wszystko

Oto wskazowka zmieniajaca zasady gry, ktora wiekszosc tutoriali pomija: **stworz klucze API dla swoich dostawcow hostingu** i przekaz je Claude'owi. To pozwala Claude'owi zarzadzac Twoja infrastruktura bezposrednio z terminala — bez klikania po dashboardach, bez kopiowania i wklejania, bez recznej pracy.

- **Klucz API Cloudflare** — przejdz do dashboardu Cloudflare → My Profile → API Tokens → stworz token. Dzieki temu Claude moze automatycznie konfigurowac rekordy DNS, tworzyc projekty Pages, zarzadzac Workers, aktualizowac ustawienia SSL i wiele wiecej. Powiedz Claude'owi:

"Oto moj token API Cloudflare. Skonfiguruj DNS dla mojej domeny wskazujac na IP mojego serwera." Gotowe w sekundy.

- **AWS Access Keys** — przejdz do AWS IAM → stworz access key. Dzieki temu Claude moze zarzadzac instancjami EC2, konfigurowac security groups, tworzyc bazy danych RDS, zarzadzac bucketami S3 i programistycznie obslugiwac cala Twoja infrastrukture AWS. Powiedz Claude'owi: *"Oto moje dane dostepu AWS. Uruchom instancje EC2 t3.small w us-east-1 i skonfiguruj security groups dla aplikacji webowej."*
- **Hetzner API Token** — przejdz do Hetzner Cloud Console → projekt → Security → API Tokens → wygeneruj jeden. Claude moze wtedy tworzyc serwery, konfigurowac firewalle, zarzadzac snapshotami i obslugiwac cala Twoja infrastrukture Hetzner. Powiedz Claude'owi: *"Oto moj token API Hetzner. Stworz serwer CX22 w Norymberdze z Ubuntu."*

> **Uwaga dotyczaca bezpieczenstwa kluczy API.** Te klucze API sa potezne — traktuj je jak hasla. **Nigdy nie commituj ich do Git**, nie udostepniaj publicznie i nie wklejaj w interfejsach czatu, ktorym nie ufasz. Przechowuj je w pliku `.env` lub przekazuj bezposrednio Claude'owi w sesji terminala. Jesli klucz zostanie skompromitowany, natychmiast go odwolaj z dashboardu dostawcy i wygeneruj nowy. Ponadto, tworzac tokeny API, **zawsze stosuj zasade minimalnych uprawnien**: przyznawaj tylko te uprawnienia, ktore sa faktycznie potrzebne, nie pelny dostep administratora.

Kombinacja tych kluczy API z Claude'em jest niesamowicie potezna. Mozesz dosłownie powiedziec: *"Mam aplikacje Next.js. Wdroz ja na Cloudflare Pages, skonfiguruj wlasna domene i skonfiguruj DNS — oto moje klucze API."* A Claude zrobi wszystko automatycznie. Lub: *"Stworz instancje EC2 na AWS, zainstaluj Node.js i PM2, skonfiguruj nginx, skonfiguruj DNS Cloudflare i wdroz moja aplikacje."* Pelna konfiguracja infrastruktury w minuty, nie godziny.

CI/CD z GitHub Actions

Pamietasz, gdy konfigurowalismy Git i GitHub CLI w Rozdziale 2? To tutaj wszystko sie opłaca. Z uwierzytelnionym `gh` mozesz powiedziec Claude Code, zeby **wypchnął Twoj projekt do prywatnego repozytorium GitHub**, skonigurowal **GitHub Actions**, ustawil wszystkie niezbedne **sekrety** (klucz SSH serwera, zmienne srodowiskowe itp.) i stworzyl automatyczny pipeline deploymentu — wszystko z terminala, bez dotykania interfejsu GitHub.

Idea jest prosta: gdy GitHub Actions jest skonfigurowany, **za kazdym razem, gdy Claude Code wypycha kod do repozytorium, automatycznie deployuje sie na serwer.** Zadnego recznego SSH, zadnego kopiowania plikow, zadnego uruchamiania komend na serwerze samodzielnie. Claude pushuje, GitHub Actions to przechwytuje, laczy sie z serwerem AWS lub Hetzner przez SSH i deployuje wszystko. W pelni zautomatyzowane.

Po prostu powiedz Claude: *"Wypchnij ten projekt do nowego prywatnego repozytorium na GitHub, skonfiguruj GitHub Action, ktory deployuje na moj serwer przy kazdym pushu do main. Oto IP mojego serwera i klucz SSH."* Claude juz wie, jak to zrobic — stworzy plik workflow, doda klucz

SSH i IP serwera jako sekrety GitHub i skonfiguruje caly pipeline. Dlatego wlasnie zainstalowalismy wczesniej GitHub CLI.

> **Pulapka dynamicznego IP.** Statyczne adresy IP zazwyczaj kosztuja dodatkowo zarowno na AWS, jak i Hetzner, wiec prawdopodobnie bedziesz uzywac **dynamicznego IP**, zeby zaoszczedzic. To oznacza, ze za kazdym razem, gdy zatrzymasz i uruchomisz ponownie serwer, IP sie zmienia. Gdy to sie stanie, musisz zaktualizowac go w **dwoch miejscach**: rekordzie DNS Cloudflare i sekrecie `SERVER_IP` w GitHub. Powiedz Claude, zeby przechowywal IP serwera jako sekret `SERVER_IP` w GitHub Actions, wiec gdy sie zmieni, musisz zaktualizowac tylko jedna zmienna. Powiedz tez Claude, zeby przypominal Ci o sprawdzeniu i aktualizacji IP, jesli deployment sie nie uda — zmienione IP to prawie zawsze przyczyna.

5. Marketing i taktyki SEO

> *Twoj produkt jest na zywo. Teraz swiat musi sie o nim dowiedziec. Najskuteczniejszy marketing dla produktow indie jest organiczny — darmowy, kreatywny i zaskakujaco potezny, gdy jest dobrze zrobiony. Ten rozdzial omawia dokladnie te taktyki, ktorych uzywamy.*

Strategie organicznego wzrostu

Badzmy szczerzy: **najlepszy marketing nie wyglada jak marketing.** Internet jest przesycony reklamami, a ludzie wyrobili sobie odruch ignorowania wszystkiego, co wyglada na promocje. To, co naprawde dziala, to **marketing ukryty** — tresci, ktore wygladaja jak autentyczna informacja, pytanie lub rekomendacja, a nie reklama. Ludzie sa z natury ciekawi i uwielbiaja pomagac sugestiami. Wykorzystaj to.

Reddit to jedna z najpotezniejszych platform do tego celu. Jest ogromny, wysoko indeksowany przez Google i pelen niszowych spolecznosci, w ktorych Twoja grupa docelowa juz przebywa. Ale oto klucz: **nigdy nie publikuj agresywnego marketingu.** Nie pisz *"Sprawdzcie moja niesamowita nowa strone!"* — to zostanie zminusowane, usuniete lub zbanowane. Zamiast tego napisz cos w stylu:

- *"Czy ktos moze polecic strony podobne do [znany konkurent] lub [Twoja strona]? Szukam alternatyw."*
- *"Czy ktos probowal [kategoria Twojego produktu]? Znalazlem [Twoja strona], ale jestem ciekaw innych opcji."*
- Odpowiadaj na pytania w odpowiednich subredditach i naturalnie wspominaj o swoim produkcie tam, gdzie to naprawde pomaga.

Tak dziala wiekszosc udanego marketingu indie na Reddit. Nie klamiesz — prezentujesz swoj produkt jako odkrycie w ramach autentycznej rozmowy. Ludzie klikaja, bo sa ciekawi, a nie dlatego, ze czuja sie zagadywani. I jest bonus: **posty na Reddit sa indeksowane przez Google**, wiec dobrze umieszczony watek moze przynosic Ci organiczny ruch z wyszukiwarek przez miesiace, a nawet lata.

Mozesz tez **sponsorowac post na Reddit** za maly budzet, aby tymczasowo go wypromowac. To podnosi go wyzej w wynikach wyszukiwania, pozwala Google szybciej go zaindeksowac i daje poczatkowa widocznosc. Zapytaj LLM, jak dziala promocja postow na Reddit i jaki budzet ma sens.

> **Trik z lejkiem.** Na swojej stronie glownej lub landing page umiesc cos, co **przyciaga ludzi niezaleznie od Twojego glownego produktu** — darmowe narzedzie, trendujacy temat, przydatne informacje lub cos, co jest aktualnie popularne. To dziala jak **lejek**: ludzie przychodza po darmowa/ciekawa tresc, odkrywaja Twoj produkt, a pewien procent z nich konwertuje. Pomysl o tym jak o przynecie, ktora dostarcza realna wartosc, jednoczesnie prowadzac do tego, co sprzedajesz.

SEO, tresci i dystrybucja

Zanim zaczniesz jakikolwiek marketing, skonfiguruj **analityke i sledzenie**. Potrzebujesz dwoch rzeczy:

- **Google Analytics** — dodaj kod sledzenia do swojej strony (popros Claude o integracje). Jest tez **aplikacja mobilna**, dzieki ktorej mozesz sprawdzac ruch z telefonu w dowolnym momencie. Pokazuje Ci calkowite odwiedziny, zachowanie uzytkownikow, zrodla ruchu i dane konwersji.
- **Google Search Console** — skonfiguruj to i polacz z Google Analytics. Search Console szczegolowo sledzi Twoj **organiczny ruch z Google**: jakie zapytania wyszukiwania prowadza ludzi na Twoja strone, jak czesto pojawiasz sie w wynikach wyszukiwania i jakie sa Twoje wskazniki klikalnosci. Zapytaj LLM, jak skonfigurowac i polaczyc oba narzedzia.

Jesli masz dostepny budzet, **Google Ads** moze dac Ci poczatkowy boost. Uruchomienie reklam na krotko pomaga zbudowac zaufanie w algorytmie Google i napedza wczesny ruch, podczas gdy Twoje organiczne SEO rosnie. Ale koszty szybko sie sumuja, wiec traktuj to jako krotkoterminowy akcelerator, a nie dlugoterminowa strategie. Twoj LLM moze szczegolowo wyjasnic konfiguracje i budzetowanie Google Ads.

Jesli chodzi o samo SEO, zacznij od podstaw. Otworz **DevTools** w przegladarce (F12), przejdz do **Lighthouse** i wygeneruj raport. Daje Ci oceny Wydajnosci, Dostepnosci, Najlepszych Praktyk i **SEO**. Napraw to, co Ci kaze naprawic — i tak, mozesz poprosic Claude Code o obsluge poprawek.

Kluczowe dzialania SEO do podjecia:

- **Dodaj tlumaczenia** do swoich stron. Obsluga wielu jezykow dramatycznie zwieksza Twoj zasieg. Popros Claude Code o skonfigurowanie internacjonalizacji dla Twojego projektu.
- **Dodaj odpowiednie meta tagi** — tytul, opis, tagi Open Graph do udostepniania w mediach spolecznosciowych, dane strukturalne. Maja bezposredni wplyw na to, jak Twoja strona

wyglada w wynikach wyszukiwania Google.

- **Utworz i zglos sitemap.** Wygeneruj aktualna mape strony i przeslij ja do Google Search Console. To informuje Google, jakie dokladnie strony istnieja na Twojej witrynie i pomaga im byc szybciej zaindeksowanymi.

> **SEO wymaga cierpliwosci.** Nie oczekuj organicznego ruchu z dnia na dzien — zajmuje tygodnie lub miesiace, zanim Google prawidlowo zaindeksuje i uplasuje Twoje strony. Ale kiedy ruszy, to **darmowy ruch, ktory nieustannie napływa** bez wydawania ani grosza. Klikniecia, za ktore w innym przypadku zaplacilbys setki euro przez Google Ads, beda przychodzic za darmo z organicznego wyszukiwania. W miedzyczasie wloz prace na platformach spolecznosciowych, takich jak Reddit, aby zbudowac poczatkowa widocznosc i backlinki. SEO to gra dlugoterminowa, ale to najbardziej wartosciowa inwestycja marketingowa, jaka mozesz zrobic.

Wschodzacy trend: ruch napedzany przez AI

Jest nowe i szybko rosnace zrodlo ruchu, na ktore wiekszosci ludzi jeszcze nie zwraca uwagi: **LLM rekomendujace Twoja strone.** ChatGPT, Gemini, Claude i inni asystenci AI sa coraz czesciej uzywani jako wyszukiwarki. Gdy ktos pyta *"Jaka jest dobra strona z [kategoria Twojego produktu]?"*, te modele moga i rekomenduja konkretne strony — a to generuje rzeczywisty ruch. Znaczna czesc naszych wlasnych odwiedzin pochodzi z ChatGPT i innych LLM.

Aby to wykorzystac, wejdz do **panelu Cloudflare** i upewnij sie, ze **wylaczyles opcje blokujaca crawlery AI**. Wiele stron domyslnie blokuje boty AI, co uniemozliwia LLM poznanie Twoich tresci. Pozwalajac crawlerom AI na dostep do strony, dajesz tym modelom mozliwosc indeksowania Twoich stron, zrozumienia tego, co oferujesz, i potencjalnego rekomendowania Cie uzytkownikom w przyszlosci. To w zasadzie **darmowe SEO na ere AI** — a boost moze byc ogromny.

> **Mysl szerzej niz Google.** Tradycyjne SEO celuje w wyszukiwarke Google. Ale rekomendacje napedzane przez AI staja sie glownym zrodlem ruchu — i ten trend tylko przyspiesza. Upewnij sie, ze Twoja strona jest dostepna dla crawlerow AI, ma jasne i opisowe tresci oraz jest dobrze ustrukturyzowana, aby LLM mogly latwo zrozumiec, co oferujesz. Strony, ktore pozycjonuja sie teraz pod katem odkrywania przez AI, beda mialy ogromna przewage w miare wzrostu tego kanalu.

Dziekujemy!

Dziekujemy za zakup tego kursu i za zaufanie, ktorym nas obdarzyles, powierzajac nam swoj czas i pieniadze. Szczerze mamy nadzieje, ze okazal sie wartosciowy i pomoze Ci zbudowac cos prawdziwego.

Chetnie uslyszymy Twoja opinie. **Dolacz do naszej spolecznosci Discord** na apifreellm.com — tam sie spotykamy, dzielimy aktualnosciami i pomagamy sobie nawzajem.

Jesli masz chwile, **napisz nam recenzje** i powiedz, co myslisz. Co bylo najbardziej przydatne? Czego brakowalo? Co mozna poprawic? Naprawde interesuje nas Twoja opinia — ten kurs to zywy projekt i chcemy go ciagle ulepszac w oparciu o to, czego **Ty** naprawde potrzebujesz.

Do zobaczenia na Discord. A teraz idz zbudowac cos niesamowitego.