



Claude Code dla piekarza – 28 pojęć bez ściemy

Myślisz, że AI jest dla brodatych programistów w kapciach, a nie dla kogoś, kto o czwartej rano wyrabia zakwas? Błąd. I zaraz ci to udowodnię – zaczynając od czegoś, co masz w kieszeni.

Zacznijmy od ChatGPT, bo to znasz

ChatGPT masz pewnie na telefonie. Wpisujesz pytanie, on odpisuje. „Napisz mi post na Facebooka o świeżych rogalikach na maśle” – i masz gotowy tekst. To jest chat: ty piszesz, on gada. I na tym koniec. Nie upiecze, nie zadzwoni do dostawcy, nie zbuduje ci strony. Tylko odpowie słowami.

Takich chatów zrobiło się kilka. Najpierw ChatGPT od firmy OpenAI, potem Gemini od Google, a później Claude od firmy Anthropic. Wszystkie działają tak samo: pytasz, dostajesz odpowiedź w formie tekstu. Claude to po prostu kolejny taki chat – trochę jak ChatGPT, tylko od innej firmy. Nic więcej na razie o nim nie musisz wiedzieć.

A teraz uwaga, bo tu robi się ciekawie.

Ta sama firma, Anthropic, zrobiła coś więcej niż zwykły chat – Claude Code. I to już nie jest chat. To znaczy: rozmawiasz z nim tak samo, po ludzku, jak z ChatGPT. Ale on nie kończy na gadaniu. On realnie siada do komputera i robi

robotę – tworzy pliki, buduje stronę z zamówieniami, liczy koszt receptury, porządkuje grafik pieców. Mówisz mu po polsku, czego chcesz, a on to buduje.

Różnica jest banalnie prosta: ChatGPT albo Claude powie ci, jak zbudować stronę. Claude Code ją zbuduje. I w przeciwieństwie do zwykłego chatu w przeglądarce – Claude Code instaluje się jak normalny program na komputerze.

Jedna rzecz, którą musisz załapać od razu: VPS

Powiedzmy, że Claude Code zbuduje ci stronę, na której klienci składają zamówienia. Świetnie. Ale jest haczyk. Jeśli ta strona siedzi na twoim laptopie, to działa tylko wtedy, gdy laptop jest włączony. Zamykasz go na noc? Strona pada. Klient chce zamówić tort o dwudziestej trzeciej – nie ma jak. Bez sensu.

Dlatego aplikacja, która ma działać zawsze i wszędzie, musi stać na VPS-ie. VPS to po prostu komputer, który wynajmujesz przez internet i który jest włączony całą dobę, siedem dni w tygodniu. Nie gaśnie, nie idzie spać, nie ma weekendu.

Pomyśl o tym tak: piec w domu wyłączasz na noc. Ale wynajęta piekarnia z piecem chodzącym non stop piecze także wtedy, gdy ty śpisz. VPS to ten wynajęty piec dla twojej aplikacji. Postawisz na nim stronę i klient zamówi tort o trzeciej w nocy, a rano masz gotowe zgłoszenie na ekranie. Ty spałeś, a interes działał.

I jeszcze jedno, żebyś nie pomyślał, że to gadżet jednej firmy

Claude Code to nie jakiś wynalazek Anthropic, który jutro zniknie. To cały nowy sposób pracy z AI. Przystałeś tylko rozmawiać z komputerem



– zacząłeś mu delegować pracę. I każdy duży gracz ma już swoją wersję tego samego. Anthropic monopolu nie ma.

Świat ChatGPT – Codex. OpenAI, ta sama firma od ChatGPT, ma swój odpowiednik: nazywa się Codex. Ta sama zasada – mówisz po ludzku, on buduje. Działa w terminalu, w edytorze kodu i przez twoje konto ChatGPT. Płacisz już za ChatGPT? Masz go w pakiecie.

Świat VS Code – Copilot i Cursor. VS Code to najpopularniejszy na świecie darmowy program do pisania kodu, od Microsoftu. Ma wbudowanego pomocnika, GitHub Copilot, który dorobił się „trybu agenta”, czyli sam pisze i poprawia, a nie tylko podpowiada. Obok wyrosły Cursor i Windsurf: podrasowane wersje VS Code z agentem w środku. Inna nazwa, ta sama bajka.

Świat Google – Antigravity. Google odpowiedziało własną platformą, Antigravity. Agent sam planuje, pisze kod, odpala go i testuje w przeglądarce, a na koniec pokazuje ci zrzuty ekranu z tym, co zrobił. W wersji próbnej jest za darmo. Co ciekawe – sam wybierasz, który „mózg” ma pracować: Gemini od Google, Claude od Anthropic albo model od OpenAI. Wszystkie w jednym oknie.

Koniec z klepaniem wszystkiego ręcznie

Widzisz schemat? Cztery różne firmy, jedna idea: koniec z klepaniem wszystkiego ręcznie, czas delegować. To jest ten nowy poziom współpracy z maszyną. I nie jest dla wybranych geniuszy z Doliny Krzemowej – jest dla każdego, kto ma komputer i nie przespał ostatniego roku pod kamieniem. Piekarz w to wchodzi tak samo jak każdy inny.

Którego wybrać? Na start, bez znaczenia. Zasady z tego przewodnika działają wszędzie tak samo – nauczysz się jednego narzędzia, ogarniesz każde. Ja pokazuję na przykładzie Claude Code, bo w pracy z kodem wciąż trzyma najwyższą jakość, ale wybór należy do ciebie.

Tyle teorii. Poniżej 28 pojęć, które usłyszysz, gdy zaczniesz pracować z Claude Code. Bez żargonu. Każde z przykładem z piekarni, żeby było jasne, po co ci to w ogóle jest.

Claude Code

Chatboty typu ChatGPT czy Gemini tylko z tobą rozmawiają. Claude Code idzie dalej – tworzy pliki, stawia strony, instaluje narzędzia, automatyzuje nudne, powtarzalne zadania. Piszysz do niego po ludzku, a nie w kodzie. Programować nie musisz. Musisz umieć dobrze opisać, czego chcesz.



Piekarnia: „Zbuduj mi stronę z formularzem na zamówienia tortów na komunie”. On tworzy pliki, ustawia strukturę, stawia stronę. Ty nie klikasz w nic.

Cennik

Nie ma nic za darmo. Claude Pro to 20 \$ miesięcznie – dobre na start, ale limity małe. Claude Max to 100 lub 200 \$ – na tym pracuje cała firma i wystarcza z zapasem. Jest jeszcze API, płatne za zużycie: tanie, gdy używasz od święta, ale przy codziennej pracy koszty potrafią eksplodować. Wyjaśniam: API to takie połączenie internetowe. Jak w każdym połączeniu e-mail coś wysyłamy i coś dostajemy. Tutaj wysyłamy rozkazy i w zamian dostajemy wykonaną pracę. Każdy rozkaz kosztuje, to koszt rzędu 50 groszy za obszerny wysłany rozkaz i kilkunastu złotych za zrobioną robotę. Jaką robotę? Wczoraj na darmowej wersji Antigraviti (czyli tego od Googla) zrobiłem system monitorowania zbiorników hydraulicznych. Czas wykonania 20 minut, koszt zero złotych. Pracuje non stop, możesz zobaczyć, wpisz do przeglądarki adres: <http://62.72.20.95:8303/>.

Prompt

Prompt to twoje polecenie. Ta sama zasada, co przy uczniu: co dasz, to dostaniesz. Śmieci na wejściu, śmieci na wyjściu. „Zrób stronę” to polecenie dla nikogo. „Zbuduj jednostronicową stronę dla piekarni, akcent kolorystyczny brązowy, formularz z polami: imię, telefon, rodzaj pieczywa, data odbioru” – to już konkret.

Piekarnia: Powiedz uczniowi „upiecz coś” i zobacz, co dostaniesz. Daj mu recepturę co do grama – dostaniesz chleb. Z Claude jest identycznie.

Terminal

Terminal to czarne okno, w którym piszesz komendy zamiast klikać. Brzmi strasznie, jest banalne. Musisz znać dokładnie trzy rzeczy: `claude` – otwiera, `Ctrl+C` dwa razy – zamyka, `clear` – czyści. Reszta to zwykła rozmowa.

Piekarnia: To po prostu łada, przy której gadasz z Claude. Nic więcej.

Permissions

Claude ma realną władzę nad twoim komputerem, więc domyślnie pyta o zgodę przed każdą ważną akcją. To dobre na start – uczysz się, co robi. Ale klikanie „tak” co dziesięć sekund zabija pracę. Więc ustawiasz sobie listę bezpiecznych akcji (czytanie plików, testy), a ryzykowne rzeczy (kasowanie, instalacja) dalej wymagają twojej zgody. Jest opcja „pozwól na wszystko” – nawet twórca każe jej nie ruszać.

Piekarnia: Pracownik, który sam sięgnie po mąkę i przeczyta recepturę, ale zanim wyrzuci partię do kosza albo zamówi towar za tysiąc złotych – pyta. Czyli jeszcze raz, jak Claude robi robotę na twoim komputerze to pyta czy może skasować, czy może otworzyć. Warto czytać co on tam wyczynia.

Tools

Claude ma wbudowane narzędzia: `read` – czyta pliki, `write` – tworzy i zmienia, `bash` – odpala komendy w systemie. Nie mówisz mu, którego użyć. Opisujesz cel, on sam dobiera narzędzie. Dlatego to coś więcej niż chatbot – ma ręce. Zwykły chatGPT nic nie robi poza pogadaniem, może coś sprawdzić w internecie. Claude Code po prostu pracuje na twoim kompie tak jak ty.

Piekarnia: Mówisz: „zamów drożdże i wydrukuj listę na jutro”. Dobry pracownik sam wie, że do jednego bierze telefon, a do drugiego drukarkę. Nie musisz mu tego tłumaczyć.

Context Window

Context Window to wszystko, co Claude widzi i pamięta w danej rozmowie. Wyobraź sobie blat z ograniczonym miejscem. Każda wiadomość i każdy plik to kartka na blacie. Gdy się zapełni, najstarsze kartki znikają pod stosem – i o nich zapomina. To „context rot”: im dłuższa rozmowa, tym gorsze odpowiedzi. Czyli oszczędzaj na gadaniu, bo to kosztuje, ale też Codex się gubi. To jak z uczniem, jak go zagadasz to się pogubi.

Piekarnia: Tablica z zamówieniami. Rano jest miejsce, wszystko widać. Po południu zapisana po brzegi – poranne zamówienie na weselny tort gdzieś przepadło pod nowymi. Tak samo gubi się Claude.

Tokeny

Token to około trzy czwarte słowa. Każdy prompt, każda odpowiedź, każdy przeczytany plik zżera tokeny.

To waluta, którą płacisz. Są 3 modele wewnątrz Claude: Haiku najtańszy, Sonnet średni, Opus najdroższy.

Piekarnia: Tokeny to twoje paliwo albo mąka. Nie liczysz każdego grama przy jednym bochenku, ale przy dwustu dziennie nagle widzisz, gdzie ucieka budżet. Tu jest identycznie.

Historia konwersacji

Claude sam zapisuje wszystkie rozmowy. Zamykasz laptop, wracasz następnego dnia i ciągniesz dokładnie od miejsca, w którym skończyłeś. W terminalu robi to komenda `claude resume`. W aplikacji masz historię z boku.

Piekarnia: Zakładka w książce. Nie zaczynasz recepty od nowa tylko dlatego, że wyszedłeś do łazienki.

Claude.md

Plik, w którym raz zapisujesz swoje zasady, preferencje i strukturę projektu. Claude czyta go na starcie każdej rozmowy. Prawdopodobnie najważniejszy plik, jaki stworzysz – bez niego Claude zgaduje, a zgadywanie daje słabe wyniki. Co lepsze: może sam go ulepszać, gdy trafi na powtarzalny błąd.

Piekarnia: Twój regulamin produkcji na ścianie. „Nigdy nie zamieniamy masła na margarynę. Zamówienia liczymy zawsze na sztuki, nie na wagę. Każdy nowy pracownik czyta to najpierw”. Napisz raz, działa zawsze.

Memory, czyli pamięć

Poza tym, co świadomie zapiszesz w `Claude.md`, Claude ma też pamięć automatyczną. Sam wyłapuje wzorce z waszej pracy – twój styl, twoje wymagania – i pamięta między sesjami. Możesz te wspomnienia przeglądać i kasować.

Piekarnia: Pracownik z długim stażem, który po miesiącu sam wie, że w piątki robisz podwójną partię chałek i że nie znosisz przypalanej skórki. Nie musisz mu tego powtarzać.

Compact Context

Pamiętasz blat z ograniczonym miejscem? Gdy rozmowa się wydłuża, Claude przy około 85% zapełnienia robi porządek: bierze stos notatek, wyciąga to, co ważne, streszcza i wyklada na czysty blat. Dzieje się to samo, ale możesz też wymusić ręcznie komendą: `compact`. Uwaga: robione za często, potrafi zjeść budżet tokenów.

Piekarnia: Sprzątasz zagracony blat w środku dnia. Zostawiasz tylko aktualne zamówienia, resztę chowasz. Pracuje się szybciej i bez błędów.

Modele AI

Model to cyfrowy mózg wytrenowany na ogromnych zbiorach danych. Pod Claude Code kryje się rodzina modeli od firmy Anthropic: Haiku – najsłabszy, szybki i tani, do banalnych rzeczy. Sonnet – uniwersalny, na większość zadań. Opus – najmądrzejszy i najdroższy, do złożonych problemów. Zmieniasz je komendą: `model`. Jest jeszcze Fable, ale to inna bajka (po ang. *fable* to bajka). To taki

model, który przed wymówieniem jego nazwy trzeba się przeżegnać. Oczywiście żartuje, rząd Stanów Zjednoczonych zawiesił jego używanie na 2 tygodnie, bo okazał się zbyt mądry. Tam w Claude też go znajdziesz, tylko uważaj na to ile zapłacisz za pracę.

Piekarnia: Uczeń, czeladnik i mistrz. Do zamykania nie wołasz mistrza. Do weselnego tortu nie posyłasz ucznia. Dobierasz asystenta AI do zadania.

Deny Access

Nie każdy plik ma trafiać do Claude. Klucze do systemów, hasła, dane poufne – tego nie ma czytać, a tym bardziej wklejać do rozmowy. W ustawieniach robisz „deny list”: czego nie wolno tknąć. Po jej ustawieniu Claude tych plików nawet nie pokaże w wynikach.

Piekarnia: Twoja tajna receptura na chleb, który zbudował ci markę. Leży w sejfie i tam zostaje. Pracownik do niej nie zagląda, kropka.

Flagi

Flagi to dopiski do komendy startowej. Zamiast zmieniać model w trakcie, odpalasz od razu z konkretnym ustawieniem, np. `claude – model opus`. Całkowicie opcjonalne – sam używam rzadko, bo rzadko siedzę w terminalu.

Piekarnia: Ustawiasz piec na konkretną temperaturę, zanim jeszcze włożysz blachę. Ramy z góry, żeby nie kombinować później.

Extended Thinking

Dedykowany budżet „myślenia”, który Claude zużywa na analizę problemu, zanim zacznie działać. Kiedyś trzeba było prosić: „myśl intensywnie”. Teraz jest to domyślnie włączone – sam decyduje, ile analizy potrzebuje.

Piekarnia: Zmiana nazwy blachy nie wymaga myślenia. Rozplanowanie trzypiętowego tortu weselnego – bardzo. Claude sam wyczuwa, kiedy przystanąć i pomyśleć.

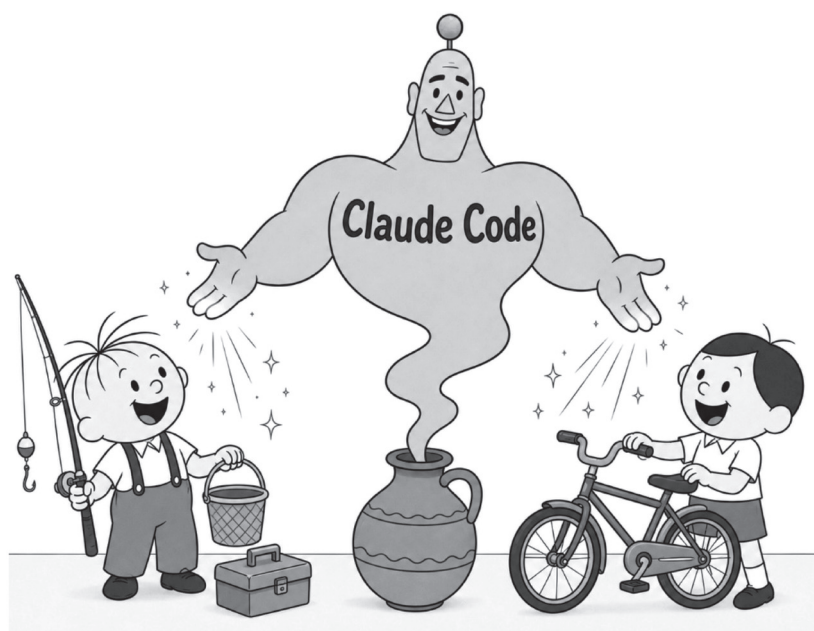
Slash Commands

Komendy po ukośniku to wbudowane skróty do częstych akcji. Wpisz `/help`, a zobaczysz listę wszystkich. Najlepsze: możesz tworzyć własne. Robisz codziennie to samo? Zrób z tego komendę.

Piekarnia: Twoje własne skróty. Jeden guzik zamiast pięciu ruchów. „Policz zamówienia na jutro” – jedna komenda, którą sam sobie ustawisz.

Skille

Claude to generalista – potrafi dużo, ale przeciętnie. Skill to plik z instrukcjami, jak ma zrobić jedną konkretną



rzecz naprawdę dobrze. Gdy prosisz o coś, co pasuje do skilla, Claude sam go ładuje. Zżera więcej kontekstu, ale jakość skacze – uczciwy deal.

Piekarnia: Karta receptury na twój sztandarowy sernik. Nie: „upiecz sernik”, tylko dokładnie twoja metoda, twój czas, twoja temperatura. Zamiast przeciętnego – twój najlepszy.

Effort

Claude nie musi zawsze zasuwać na maksa. Opus 4.8 wprowadził kontrolę wysiłku – dosłownie mówisz mu, ile energii ma włożyć. Chcesz, żeby coś wyszło perfekcyjnie? Dajesz maksymalny tryb. To wpływa i na jakość, i na koszt.

Piekarnia: Jednemu pracownikowi powiesz: „zrób po łebkach i tak idzie do kruszonki”. Drugiemu: „to na wesele, sprawdź trzy razy”. Ty ustawiasz poprzeczkę.

Hooks

Hooks to skrypty, które odpalają się same, gdy zajdzie dane zdarzenie: jeśli stanie się X, wykonaj Y. Kluczowe: nie używają AI, więc nie zżerają tokenów i są w stu procentach przewidywalne. Zawsze robią to samo.

Piekarnia: Minutnik przy piecu. Nie myśli, nie ma humorów, nie zżera paliwa. Dzwoni zawsze po tym samym czasie. Ustawiasz raz i masz spokój.

MCP

MCP (*Model Context Protocol*) to standard od Anthropic, który łączy Claude z zewnętrznymi narzędziami: Gmailem, Dyskiem Google, innymi aplikacjami. To mosty. Po podłączeniu Claude czyta z nich dane i sam w nich działa, wszystko z jednego okna. Tu Claude przestaje być narzędziem lokalnym, a staje się centrum dowodzenia.

Piekarnia: Podłączasz Claude do skrzynki, na którą przychodzą zamówienia, do arkusza z magazynem i do

maili od dostawcy mąki. Jedno okno, a on czyta zamówienia, aktualizuje stan magazynu i szykuje odpowiedź do dostawcy.

Sub-agents

Sub-agent to osobna instancja Claude z jednym konkretnym zadaniem. Pracuje w izolacji, z czystym oknem kontekstowym, i odsyła wynik do agenta głównego. Możesz odpalić kilka naraz. Zamiast pchać wszystko w jedną zapchaną rozmowę, główny agent rozdaje zadania specjalistom.

Piekarnia: Rozstawiasz stanowiska. Jeden robi tylko ciasto, drugi tylko dekoracje, trzeci tylko pakowanie. Każdy ma swój czysty blat i swoje zadanie. Szybciej i bez bałaganu.

Agent Teams

W odróżnieniu od sub-agentów, Agent Teams gadają między sobą i dzielą wspólną listę zadań. Pracują równolegle i widzą nawzajem, co robią. Sub-agenty raportują tylko do szefa, co przy większej liczbie staje się wąskim gardłem – tu agenci dogadują się bezpośrednio.

Piekarnia: Zgrana brygada zamiast luźnych stanowisk. Ciastownik woła do dekoratora: „skończone, bierz”, a pakowacz już wie, że za pięć minut ma robotę. Wszyscy widzą tę samą listę zamówień.

Image Support

Możesz wkleić Claude zdjęcie albo zrzut ekranu. Pokazać błąd, wrzucić projekt do odwzorowania. Claude widzi obraz i na niego reaguje. Obrazy zżerają dużo więcej tokenów niż tekst, ale czasem jedno zdjęcie mówi więcej niż akapit tłumaczenia.

Piekarnia: Robisz zdjęcie tortu konkurencji z gabloty i mówisz: „chcę coś w tym stylu na stronie”. Albo fotka rozjechanej etykiety z ceną – Claude widzi, co jest nie tak.

Checkpoints/Undo/Rewind

Claude sam robi kopię zapasową przed każdą edycją. Gdy pójdziesz w złą stronę i coś się zepsuje, wpisujesz/rewind, widzisz listę wszystkich kroków i cofasz się do momentu sprzed błędu. Cofasz rozmowę, kod albo jedno i drugie.

Piekarnia: Cofnięcie partii, która wyszła nie tak. Wracasz do momentu sprzed nieudanego pomysłu i próbujesz inaczej, bez zaczynania wszystkiego od zera.

Git/GitHub

Git to system kontroli wersji – śledzi każdą zmianę w projekcie: kto, co, kiedy i dlaczego zmienił. GitHub to platforma, gdzie ta historia leży online, jak Dysk Google, tylko dla kodu. Claude używa Gita pod spodem do kopii zapasowych i cofania zmian. Ekspertem być nie musisz – poradzi sobie za ciebie.

Piekarnia: Historia zmian w recepturze. Widzisz, że pół roku temu ktoś podbił ilość drożdży o dwa gramy – i wtedy chleb zaczął wychodzić lepiej. Nic nie ginie, wszystko da się prześledzić.

Headless Mode

Normalnie Claude jest interaktywny: pisesz, on odpowiada, zatwierdzasz i tak w kółko. Headless mode to tryb pełnej automatyzacji – dodajesz flagę -p, dajesz zadanie, a on robi je od A do Z. Żadnych pytań, żadnych zatwierdzeń. Wraca z gotowym. Jest nawet „Ralph loop”, który każe mu kręcić się w pętli, aż skończy.

Piekarnia: Nocna zmiana bez ludzi. Zostawiasz zadanie na wieczór, idziesz spać, rano jest zrobione. To ta magia, gdzie ludzie budują całą aplikację przez noc, kiedy śpisz. Tyle że tu pilnujesz, co mu wolno – bo działa sam.

Worktrees

Przy większym projekcie pracujesz nad kilkoma rzeczami naraz: nowa funkcja, poprawka błędu, eksperyment. Worktrees tworzą osobne kopie robocze – każda ma swój folder i swoją gałąź. Odpalasz kilka niezależnych Claude, które pracują równolegle i nie nadpisują sobie roboty. Na koniec „mergeujesz”, czyli scalasz wszystko z powrotem w jedną całość.

Piekarnia: Trzy stoły, na każdym inny produkt: chleb, croissanty, eksperymentalna bułka z dynią. Żaden nie miesza się z drugim. Gdy każdy gotowy – wjeżdżają razem na wspólną ekspedycję.

Na koniec – bez owijania

Sporo tych pojęć może brzmieć teraz jak chińszczyzna. Normalne. Nikt nie nauczył się piec, czytając o piekarniku – nauczył się, wsadzając ręce w ciasto.

Więc rada jest jedna: buduj. Odpal Claude Code i zrób sobie pierwszą, najprostszą rzecz – stronę na zamówienia, kalkulator kosztu receptury, listę zakupów na tydzień. A gdy trafisz na pojęcie, którego nie rozumiesz, wróć tutaj i sobie przypomnij.

Acha, wciąż nie jasne? Jak to wygląda w praktyce? Mówisz mu co ma zrobić, on robi i wystawia Ci plik html do pobrania. Wstawiasz to na pulpit, klikasz i otwiera się. Masz swoją stronę, kalkulator, licznik słów czy co tam sobie chciałeś stworzyć.

Tyle. Do roboty.

Wojciech Moszczyński

Wojciech Moszczyński – absolwent Katedry Ekonometrii i Statystyki Uniwersytetu Mikołaja Kopernika w Toruniu, specjalista z zakresu ekonometrii, finansów, *data science* oraz rachunkowości zarządczej. Specjalizuje się w optymalizacji procesów produkcyjnych i logistycznych. Prowadzi badania w obszarze rozwoju i zastosowania sztucznej inteligencji. Od lat zaangażowany w popularyzację *machine learning* oraz *data science* w środowiskach biznesowych.