PROGRAM SZKOLENIA SZTUCZNA INTELIGENCJA

Szkolenie LLM w Twojej aplikacji

Dodasz do aplikacji funkcje AI, które działają przewidywalnie i mieszczą się w budżecie.

CZAS TRWANIA	POZIOM	FORMA
3 dni	**średniozaawansowany**	**warsztat dla zespołu, zdalnie lub stacjonarnie**

Pierwsze demo z modelem językowym powstaje w godzinę. Schody zaczynają się później: model odpowiada raz tak, raz inaczej, zmyśla dane, gubi się w długiej rozmowie, a rachunek za tokeny rośnie szybciej niż liczba użytkowników.

Na szkoleniu dokładasz funkcje AI do prawdziwej aplikacji w TypeScript (backend w Node.js, prosty interfejs w przeglądarce): czat nad własnymi dokumentami, ekstrakcję danych i agenta, który woła Twoje API. Wszystkie ćwiczenia są wyłącznie na API Claude, bez wariantu na innych dostawców ani Copilota, ale wzorce przeniesiesz na każde API modelu. Uczysz się też, jak sprawdzić, czy to działa, zanim zobaczy to klient.

PROGRAM

3 dni

- Model językowy od strony programisty
  - Dobór modelu: jakość, szybkość i koszt
  - API Claude i SDK w TypeScript, wzorce przenośne na innych dostawców
  - Modele z rozumowaniem i parametr `effort`: kiedy się opłacają
  - Wejście multimodalne: PDF i obrazy
- Prompty i streaming
  - Prompt systemowy, szablony i przykłady w prompcie (few-shot)
  - Wersjonowanie promptów razem z kodem
  - Odpowiedź strumieniowana z serwera do przeglądarki
  - Przerywanie generowania i obsługa błędów w UI
- Ustrukturyzowane odpowiedzi
  - Natywne structured outputs i tryb strict w narzędziach
  - JSON Schema i typy z Zod
  - Kiedy walidacja nadal jest potrzebna: odmowy i ucięte odpowiedzi

- Narzędzia (tool use)
  - Model, który woła funkcje Twojej aplikacji
  - Pętla agenta: plan, wywołanie, wynik
  - Granice: co agent wykona bez Twojego potwierdzenia
  - Długie zadania: kompaktowanie rozmowy i czyszczenie starych wyników narzędzi
- RAG: model i Twoje dane
  - Embeddingi z Voyage AI albo lokalnego modelu w Ollama
  - Podział dokumentów na fragmenty
  - Baza wektorowa w PostgreSQL (pgvector)
  - Wyszukiwanie hybrydowe: pełnotekstowe i wektorowe
  - Cytowanie źródeł w odpowiedzi
- Model Context Protocol
  - Własny serwer MCP według specyfikacji 2026-07-28
  - Protokół bezstanowy: serwer MCP skaluje się jak zwykłe API HTTP
  - Narzędzia, zasoby i autoryzacja OAuth
  - MCP Apps: interfejs serwera wyświetlany w aplikacji hosta
- Ewaluacja
  - Zestaw testowy pytań i oczekiwanych odpowiedzi
  - Model jako sędzia (LLM-as-a-judge)
  - Testy regresji po zmianie promptu lub modelu
- Bezpieczeństwo
  - Prompt injection, także przez dokumenty i strony WWW
  - Dane osobowe i sekrety w kontekście
  - Walidacja tego, co model zwraca, zanim trafi dalej
- Koszty i działanie na produkcji
  - Liczenie tokenów, prompt caching i mniejsze modele do prostych zadań
  - Przetwarzanie wsadowe (Batch API) dla zadań, które mogą poczekać
  - Logowanie, śledzenie wywołań (OpenTelemetry) i monitoring
  - Obsługa awarii dostawcy

DLA KOGO?

- Dla programistów, którzy mają dodać funkcję AI do istniejącego produktu w Node.js.
- Dla zespołów, które mają działające demo i chcą przenieść je na produkcję.
- Dla tech leadów, którzy oceniają koszty i ryzyka funkcji opartych na LLM.

CO MUSISZ UMIEĆ?

- Programować w JavaScript lub TypeScript na poziomie komercyjnym.
- Pracować z REST API i Node.js.
- Znać podstawy pracy z modelem językowym: tokeny, prompt, okno kontekstu.
- Czytać proste zapytanie SQL `SELECT` (zapytania do pgvector dostaniesz gotowe do uzupełnienia).

- Uruchomić w Dockerze gotowy kontener z bazą danych.
- Mieć klucz API Anthropic, a do embeddingów klucz Voyage AI albo Ollama na swoim komputerze.

CZEGO SIĘ NAUCZYSZ?

- Wywołasz model z kodu i dostaniesz odpowiedź zgodną ze schematem.
- Prześlesz odpowiedź strumieniowo do przeglądarki i obsłużysz jej przerwanie.
- Zbudujesz czat, który odpowiada na podstawie Twoich dokumentów i podaje źródła.
- Napiszesz narzędzia, które model wywołuje w Twojej aplikacji.
- Wystawisz funkcje swojego produktu jako serwer MCP.
- Przygotujesz zestaw testów, który wyłapie regresję po zmianie promptu.
- Rozpoznasz próbę prompt injection i ograniczysz jej skutki.
- Oszacujesz koszt funkcji AI i obniżysz go bez utraty jakości.

TRENER

Piotr Kowalski

W branży IT od 2008 roku. Od 2014 jest mentorem dla wielu developerów. Jako trener spędził na sali szkoleniowej ponad 1000 godzin, a na LinkedIn ma 52 rekomendacje. Każde jego szkolenie to dawka solidnej wiedzy oraz humoru.

Więcej: piecioshka.pl/piotr-kowalski

ZORGANIZUJMY TO SZKOLENIE W TWOIM ZESPOLE

Program dopasuję do Waszego projektu i poziomu grupy. Napisz, z czym pracujecie, a odpowiem z propozycją terminu i zakresu.

KONTAKT **piecioshka.pl/kontakt**
E-MAIL **piecioshka+blog [małpka] gmail.com**
SZKOLENIE **piecioshka.pl/szkolenia/llm-w-aplikacji/**