

Степан Коток (Stepan Kotok)

AI Engineer, Full Stack Developer, Solution Architect

React · Next.js · Node.js · TypeScript · PostgreSQL/Supabase · Claude API, OpenAI API, MCP, RAG · Playwright, Vitest · GitHub Actions

Русский, украинский – родные; английский – B2, разговорный.

Кишинёв, Молдова · [+373 78 722 382](tel:+37378722382) · wa.me/37378722382 · stkotok@gmail.com ·

linkedin.com/in/stkotok · github.com/StKotok · stepankotok.com



Профиль

В разработке с 2014 года, с 2020-го вёл проекты как техлид в своей компании [Neatapps](#), заказчики из США и Европы. С 2024 года системно усиливаю весь инженерный цикл AI-разработкой.

Роль в команде

- Основная роль – Senior-разработчик, отвечающий за техническую надёжность: выбираю технологии под задачи, проектирую системную архитектуру, пишу код и контролирую качество на всех этапах.
- При необходимости беру роль тимлида: в [Neatapps](#) координировал взаимодействие с заказчиками и командой, проводил ежедневные планёрки с отчётами о выполненной работе и разбором сложностей.

Как я пишу код с ИИ

Инструменты

Claude Code – основной инструмент, каждый день, на старшем тарифе Max. Субагенты, навыки, хуки, собственные MCP-серверы, параллельная работа в изолированных git worktree. Модель подбираю под задачу: сильная планирует и принимает работу, более быстрая пишет код по заданию с критериями приёмки.

Cursor – уверенно владею, работал на платной подписке; сейчас основной инструмент – Claude Code.

ChatGPT/Codex, Gemini, Grok – на платных тарифах Plus, Pro и Super: независимое ревью кода и планов. Несколько моделей по очереди, с исправлениями между раундами; второе мнение – перед необратимыми шагами: миграцией, выкладкой, изменением прав доступа.

Проверка в браузере и на телефоне (Playwright, Mobile-MCP) – агенты сами проверяют результат в настоящем браузере и на мобильном телефоне: сквозные сценарии, доступность, снимки экрана.

Как это ускоряет работу, не снижая качества

- Код быстрее.** За четыре дня собрал с ИИ-агентами платформу онлайн-курсов с офлайн-режимом, входом и правами доступа, панелью администратора, покрыл сквозными автотестами.
- Поиск ошибок быстрее.** Разбор логов и ошибок отдаю агентам: они находят место сбоя и предлагают исправление, а я проверяю диагноз и принимаю результат.
- Тестирование быстрее и полнее.** Агенты пишут автотесты в коде и сценарии для ручного тестирования. На проекте [ИИ-ассистента для туристического сервиса](#) (ниже) за семь дней написал 1 310 тестов в шести слоях. Также ИИ сам выполняет тестирование как живой тестирующий: проходит сценарии в настоящем браузере и на мобильном телефоне, делает снимки экрана и сообщает, что исправить.
- Запуск быстрее и безопаснее.** Блокирующая проверка перед основной веткой, каждое изменение связано с задачей, CI в GitHub Actions.
- Рутинная работа – агентам.** Однотипные правки по десяткам файлов, разбор задач, поиск в документации делают агенты. За мной – архитектура, сложные места и проверка результата.
- Проверяю, а не верю на слово.** Если ИИ ошибся, фиксирую для него правило, чтобы ошибка не повторилась. Любой сгенерированный код сначала верифицирую типизацией, тестами и архитектурным ревью и только потом внедряю.

Избранные проекты

ИИ-ассистент для туристического сервиса: отели, маршруты, бронирование

2026 · Next.js, TypeScript, Supabase/PostgreSQL, Claude API и OpenAI API, Vitest, Playwright, GitHub Actions

Проект Willy Agency, Саас-Фе, Швейцария · Туристический сервис

- Владелец собрал продукт с ИИ-агентами за 71 день; продукт работал и был близок к первым продажам. За неделю в августе 2026 года, управляя ИИ-агентами, я:
- Организовал надёжную работу диалогового ассистента с базой знаний туристического сервиса (отели, маршруты, сервисные вопросы гостей, бронирование слотов).
 - Построил тесты с нуля: 1 310 тестов в шести слоях, без сети, базы и модели, – полный прогон за 3,64 секунды без секретов.
 - Провёл аудит всей поверхности: авторизация маршрутов, изоляция данных на уровне строк (RLS в Supabase для защиты приватных данных от prompt injection), входящие вебхуки, секреты, разделение окружений; критичное закрыл и закрепил тестами, которые валят сборку при возврате проблемы.

Приложение для оценки состояния устройств

Neatapps · проект под NDA для международной компании потребительской электроники · Flutter (iOS, Android), нативный iOS, Node.js, компьютерное зрение

Под ключ: дизайн интерфейса, мобильное приложение на Flutter с нативными модулями для iOS и бэкенд на Node.js. Компьютерное зрение в реальном времени определяло положение устройства перед камерой и само делало снимки, а оценку состояния по ним возвращал API.

Опыт работы

- Willy Agency – технический партнёр по контракту**
Август 2026 – сейчас · частичная занятость · Саас-Фе, Швейцария, удалённо
- Full Stack / AI-разработчик по контрактам**
Апрель 2026 – сейчас · Кишинёв
 - Довожу прототипы, собранные с помощью ИИ, до продакшена: архитектура, данные, авторизация, платежи, тесты критичных сценариев, релиз.
- Neatapps – сооснователь, технический руководитель проектов**
Июль 2020 – март 2026 · заказчики из США и Европы
 - Превращал цели заказчиков в технический объём работ: исследование, требования, оценки, план поставки, общение с заказчиком.
 - Координировал разработчиков и дизайнеров на всём цикле: архитектура, разработка, тестирование, релиз, поддержка после релиза.
- Neatapps – сооснователь, мобильный разработчик**
Июль 2015 – июль 2020
 - Нативный Android (Java, Kotlin), затем iOS (Swift) и кроссплатформенная разработка на Flutter.

Технологии

Frontend:	React, Next.js, TypeScript, Vite, Tailwind CSS, PWA (Service Worker, IndexedDB), TanStack Query.
Backend и данные:	Node.js, NestJS, RESTful API, PostgreSQL, Supabase (RLS, RPC, миграции, pgvector), pgTAP.
AI и LLM:	Claude API (Anthropic SDK), OpenAI API, Gemini API, MCP (собственные серверы), RAG, вызов инструментов (Function Calling), структурированные ответы.
Архитектура:	Solution Architecture, System Design, безопасность данных.
Cloud и DevOps:	Docker, Cloudflare Workers/Pages, Vercel, Netlify, GitHub Actions (CI/CD), Git, Linux.
Мобильная разработка:	Flutter, Android (Java, Kotlin), iOS (Swift), BLE/IoT.
Качество и надёжность:	Vitest, Testing Library, Playwright, pgTAP, axe, Sentry.
Работал раньше / знаком:	Java (Core, Spring), Python (FastAPI, SQLAlchemy), Redis, MongoDB, AWS, GCP, Nginx.