



BORIS STEPANENKO

FRONTEND / FULLSTACK
DEVELOPER

Формат работы: удалённо, полная
занятость

Часовой пояс: UTC+3
(Europe/Moscow)

ОБО МНЕ

Frontend-разработчик с опытом fullstack-проектов на Vue / Nuxt. Проектирую масштабируемую архитектуру приложений, работаю с глобальным состоянием, REST API и базами данных (MongoDB, SQLite), реализую тёмную/светлую тему и мультиязычность (до трёх языков в одном проекте). Умею вести продукт целиком: от структуры данных и API до финального интерфейса. Сделал 2 коммерческих проекта в 2026 году, включая международный сайт на 3 языках.

КОНТАКТЫ

+7 993 676-10-75

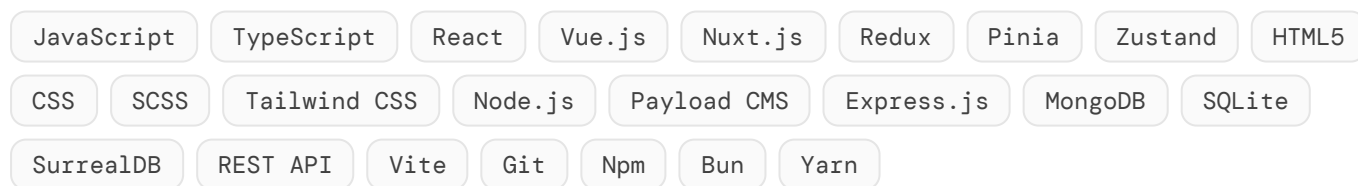
Telegram: @mervik104

GitHub: mervik104

mervik104@gmail.com

mervik.ru — сайт-портфолио

СТЕК ТЕХНОЛОГИЙ



ЯЗЫКИ

Русский

родной

Английский

A2-B1

ОБРАЗОВАНИЕ

Самообучение

Путь в разработку

МАЙ 2025 — ЯНВАРЬ 2026

● ОПЫТ РАБОТЫ

| FULLSTACK DEVELOPER

ИЮНЬ 2026 – АВГУСТ 2026

ГК «Некрасовский»  nekras.ru

Коммерческий сайт для группы компаний Некрасовский на заказ. Ключевой фокус – презентация бизнеса для международной аудитории: реализовал поддержку трёх языков (русский, английский, китайский) и переключение тёмной/светлой темы. Спроектировал backend на Payload CMS с админ-панелью для самостоятельного управления контентом заказчиком, структуру MongoDB и API. Frontend – Nuxt + Tailwind, ISR-рендеринг, SEO-оптимизация под коммерческий трафик.

| FULLSTACK DEVELOPER

ИЮЛЬ 2026 – ИЮЛЬ 2026

Internetlab (тестовое задание)  GitHub

Тестовое задание: форма обратной связи с анализом тональности сообщений через Google Gemini AI. Backend на FastAPI с слоистой архитектурой (API / Service / Repository), Pydantic-валидация, rate limiting и логирование запросов. Frontend на Nuxt 4 с Pinia и Tailwind CSS 4. Реализован graceful fallback при недоступности AI-модели, автогенерация Swagger-документации.

| FULLSTACK DEVELOPER

МАЙ 2026 – ИЮНЬ 2026

Freshcheck  freshcheckastra.ru  GitHub

Сайт для гражданской инициативы жителей Астрахани – независимый проект с open-source кодом. Здесь в приоритете была скорость запуска и открытость: backend на Payload CMS + MongoDB. Frontend на Nuxt + Tailwind с тёмной/светлой темой, ISR и SEO – чтобы контент по проверкам находился в поиске и быстро обновлялся.

| FRONTEND DEVELOPER

ИЮНЬ 2026 – ИЮНЬ 2026

Личный проект  mervik.ru  GitHub

Разработал сайт-визитку на Nuxt и Tailwind CSS v4. Задеплоил на GitHub Pages.

Nuxtgram - Личный проект  nuxtgram.mervik.ru  GitHub

Social/photo-блог в духе Instagram: посты с картинками, лайки/реакции, комментарии, подписки, профили. Архитектура полностью serverless — без собственного монолитного бэкенда, только сервисы-провайдеры с чётким разделением зон ответственности. Frontend — Nuxt 4 SPA (Vue 3.5, TypeScript 5.9): ssr отключён, hash-роутинг, сборка в один статический HTML через nuxt-single-html — весь проект деплоится как статика на GitHub Pages, без CI на данный момент. Аутентификация полностью на Clerk (@clerk/nuxt) — регистрация и вход по email/OAuth, управление сессией; для доступа к данным Clerk выдаёт JWT по кастомному шаблону с issuer и audience, привязанными к проекту. Этот же JWT служит сессионным токеном для SurrealDB и проверяется Cloudflare Worker (issuer/audience/наличие sub) перед любой операцией — фронтенд не является границей доверия, все секреты и права живут на сервере. Данные — SurrealDB (документно-графовая БД) через типизированный ORM surqlize: таблицы users, media, posts, comments и рёбра follows, post_reactions, comment_reactions, доступ построен на server-side permissions по схеме (миграция 001-infrastructure.surql), плюс real-time LIVE-подписки на изменения с инкрементальным обновлением стортов. Медиа и файлы — Cloudflare Worker поверх Backblaze B2: presigned URL на загрузку/скачивание с TTL, проверка владельца объекта, ограничение размера, публичных прямых загрузок в bucket нет. State-менеджмент — Pinia (auth/post/comment/follows-сторты), формы — vee-validate + Zod-схемы, UI — @nuxt/ui, Tailwind CSS 4, tailwind-variants, color-mode, toast-уведомления, анимации через @formkit/auto-animate. Есть юнит-тесты на bun:test (мапперы, схема, форматирование, логгер, редиректы) и собственный браузерный логгер, пишущий пачками в локальный лог-эндпоинт. Отдельная деталь эволюции проекта: раньше Nuxtgram работал на self-hosted Payload CMS (Next.js + MongoDB) как монолитном бэкенде — этот код вынесен и сохранён отдельно как legacy-репозиторий (nuxtgram-backend) исключительно для истории, а текущая архитектура полностью переехала на serverless-стек.