

Смирнов Никита Денисович

Backend/Fullstack разработчик

✉ nikita.d.smirnov@gmail.com

📞 +7 965 365-97-93

📠 @abkubyb

👤 Nikita-Smirnov-idk

О себе

Backend разработчик с опытом создания масштабируемых микросервисных приложений. Специализируясь на Go и Python, применяю современные архитектурные паттерны (Clean Architecture, DDD) и принципы проектирования (SOLID, GRASP, GoF). Имею опыт полного цикла разработки: от проектирования архитектуры до развертывания в production с использованием Docker, CI/CD, gRPC и REST API.

Опыт работы

Backend/Frontend Developer — Gpugo.ru

2025-2026

- Разработка и поддержка сервиса-хостинга GPU мощностей для ML обучения, инференса и вычислительных задач
- Работа над frontend и backend частями проекта
- Оптимизация производительности и масштабируемости сервиса
- Создание международного веб-сайта

Freelance Web Developer

2024-2025

- Разработка сайтов на заказ для кондитерских кафе и ресторанов
- Портфолио: [cg78261.tw1.ru](#)
- Прямая коммуникация с заказчиками, сбор требований, техническая поддержка
- Получение первой коммерческой прибыли от собственного "бизнеса"

Образование

НИУ ВШЭ, Факультет компьютерных наук

2026

Программная инженерия, 2 курс

Пройденные курсы:

- Яндекс Лицей: Основы программирования на Python (2021)
- Основы промышленного программирования (2022)
- ML | Специализации Яндекс Лицей (2023)
- Django | Специализации Яндекс Лицей (2023)
- Go: шаг за шагом

Технические навыки

- **Языки программирования:** Go (2+ года), Python (7+ лет), C++ (1 год), C# (1 год), TypeScript, JavaScript, SQL
- **Backend фреймворки:** Gin, Chi, Django, FastAPI, Flask, .NET
- **Frontend:** React, Next.js (1+ год опыта)
- **Базы данных:** PostgreSQL, SQLite, Redis, MinIO/S3
- **Архитектура:** Clean Architecture, DDD, микросервисная архитектура
- **Принципы и паттерны:** SOLID, DRY, KISS, YAGNI, ООП, GoF patterns, GRASP
- **API:** REST, gRPC, JWT, OAuth 2.0
- **DevOps:** Docker, Docker Compose, CI/CD, Git, GitHub, GitLab
- **Инструменты:** Linux, Bash/Zsh, Celery, Kafka
- **Интеграции:** OpenAI API, Reddit API, Google OAuth, SMTP
- **Английский язык:** B2

Ключевые проекты

Система проверки на плагиат (Go)

- Микросервисная архитектура с API Gateway, Storage Service и Plagiarism Service
- gRPC коммуникация между сервисами, REST API для клиентов
- PostgreSQL для метаданных, MinIO/S3 для хранения файлов
- Алгоритм определения плагиата на основе n-грамм и метрики Jaccard
- Docker Compose для оркестрации, кэширование результатов анализа

SubRMatch API (Python + FastAPI)

- AI-powered поиск подходящего сабреддита для любого поста юзера и форматирование этого поста согласно правилам найденного сабреддита на Reddit
- Интеграция с OpenAI/OpenRouter API (DeepSeek Chat) со streaming ответами
- JWT аутентификация, Google OAuth 2.0, email верификация
- Celery + Redis для асинхронной обработки задач
- Rate limiting, middleware для логирования

SubRMatch Frontend (React)

- SPA для работы с Reddit постами
- Интеграция с SubRMatch API, обработка streaming ответов
- Современный UI/UX с использованием React

TrackHub (Python + Django, Swift)

- Мобильное приложение для тренеров и клиентов на SwiftUI
- Backend на Django с JWT аутентификацией
- Docker контейнеризация, PostgreSQL, S3 Cloud Storage
- Email уведомления через Yandex SMTP

Модуль учета финансов (Go)

- Clean Architecture с применением паттернов GoF и GRASP
- Реализация паттернов: Factory, Facade, Proxy, Decorator, Command, Visitor, Strategy, Template Method

- Импорт/экспорт данных в JSON, YAML, CSV форматах
- SQLite база данных, полное покрытие принципами SOLID

Консольное приложение Зоопарк (Go)

- Clean Architecture, DI-контейнер для управления зависимостями
- Применение принципов SOLID, интерфейсно-ориентированное проектирование
- Фабрики для создания объектов, юнит-тесты

Профессиональные качества

- Опыт работы в команде и самостоятельной разработки
- Умение проектировать масштабируемую архитектуру
- Внимание к качеству кода и best practices
- Быстрое освоение новых технологий
- Опыт коммуникации с заказчиками