

Григорий Захаров

Email: zakharov9933@gmail.com | GitHub: github.com/GrigoryZakharov | LinkedIn: [grigory-zakharov](https://www.linkedin.com/in/grigory-zakharov)

Английский язык: продвинутый уровень (C1)

Backend / Fullstack Developer | Python, FastAPI, React, PostgreSQL |

Образование

Ульяновский государственный технический университет — Ульяновск, Россия

Бакалавр по направлению *Математическое обеспечение и администрирование информационных систем*

Ожидаемое окончание: май 2028 г.

Мотивированный backend / fullstack разработчик с опытом создания RESTful API на Python (FastAPI) и фронтенда на React. Опыт разработки масштабируемых приложений с PostgreSQL, Docker, JWT-аутентификацией и миграциями базы данных. Акцент на чистой архитектуре, производительности и тестируемом коде.

Languages & Frameworks: Python, FastAPI, SQLAlchemy, Django, C++

Databases: PostgreSQL, MySQL, SQLite

Tools & Technologies: Git, Docker, Redis, Celery, AsyncIO, Unit Testing

Other: REST API Design, JWT Authentication, CI/CD (basic), SPA, React Router

Soft Skills

- Аналитическое мышление • Работа в команде • Быстрая обучаемость • Внимание к деталям

Опыт работы

- **Python Developer Стажировка**

Ulyanovsk State Technical University (on site)

June 2025- July 2025

- Разрабатывал и тестировал REST API с использованием FastAPI и PostgreSQL.
- Работал с системой контроля версий Git, участвовал в код-ревью.
- Улучшил понимание архитектуры API и моделирования баз данных.

Проекты

- **Blog Platform** – Python, FastAPI, React, PostgreSQL, SQLAlchemy, JWT, Docker

- Реализовал полноценный fullstack блог с аутентификацией и авторизацией через JWT.
- CRUD операции для постов и комментариев, функционал лайков/дизлайков.
- Настроил Alembic для миграций базы данных и Docker для backend, frontend и БД.
- SPA на React с React Router, интеграция с FastAPI через REST.
- Интерактивная документация API (/docs) и тесты backend с pytest.
- [[GitHub Repository](#)]

- **WeatherHub API** – Python, FastAPI, PostgreSQL, SQLAlchemy, Redis, Celery

- Создал масштабируемый API для получения и синхронизации погодных данных из внешних источников.
- Настроил Celery и Redis для фоновых задач и улучшения производительности запросов.
- Реализовал эндпоинты для запросов по городам с сохранением данных в PostgreSQL.
- Подготовил Docker и .env-конфигурации для удобного деплоя и тестирования.
- [[GitHub Repository](#)]