# Grigory Zakharov

**Email:** zakharov9933@gmail.com | **GitHub**: github.com/GrigoryZakharov | **LinkedIn**: grigory-zakharov
Languages: English – Advanced (C1)
Backend Developer | Python, FastAPI, PostgreSQL | Open to Remote Opportunities (U.S.).

## Education

**Ulyanovsk State Technical University – Ulyanovsk, Russia**
Bachelor of Science in Mathematical Software and Information Systems Administration
Expected Graduation: May 2028

Passionate backend developer with experience building RESTful APIs using FastAPI and PostgreSQL. Strong focus on clean architecture, performance, and scalable backend systems. Looking for an opportunity to contribute to a collaborative engineering team while continuing to grow technical depth in Python backend development.

**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 Auth, CI/CD (basic)

## Soft Skills

- Problem-Solving, Team Collaboration, Fast Learning, Attention to Detail

## Work Experience

- **Python Developer Intern**
  **Ulyanovsk State Technical University** (on site) *June 2025- July 2025*
  - Developed and tested REST APIs using FastAPI and PostgreSQL.
  - Collaborated with team members using Git and code reviews.
  - Improved understanding of database modeling and API architecture.

## Projects

- **Blog API** – Python, FastAPI, PostgreSQL, SQLAlchemy, JWT
  - Designed a complete REST API for a blog platform with secure authentication and authorization.
  - Implemented CRUD operations for posts and comments with relational models and pagination.
  - Integrated JWT-based authentication with password hashing (using `passlib`) and token validation middleware.
  - Focused on clean architecture and error handling for production-readiness.
  - [GitHub Repository]

- **WeatherHub API –** Python, FastAPI, PostgreSQL, SQLAlchemy, Redis, Celery
  - Built a scalable weather data API capable of background synchronization with external sources.
  - Integrated Celery and Redis to schedule asynchronous data updates and improve response performance.
  - Implemented endpoints for dynamic city-based weather queries and stored results using optimized SQLAlchemy models.
  - Added Docker and `.env` configurations for smooth deployment and environment consistency.
  - [GitHub Repository]