Igor Karkhalev

Back-end Developer

+79149014666 | karkhalev.iv@phystech.edu | Dolgoprudny
github.com/kxigor | gitlab.com/kxigor | 04.03.2005

Professional Summary

Back-end C++ developer who genuinely enjoys programming with expertise in system architecture, database design, and algorithm implementation. Strong background in low-level programming, performance optimization, and developing complex software systems from the ground up.

Education

Moscow Institute of Physics and Technology (MIPT) **Bachelor's Degree**

2024 – 2027 (Expected)
Computer Science and Engineering: System Programming and Applied Mathematics.

Irkutsk Alise Programming School **Advanced Programming Curriculum**

2017 – 2024
Comprehensive training in algorithms, data structures, and software development.

Technical Skills

- **Programming Languages:** C++ (advanced), C, Assembly, Go, Python;
- **Systems & Tools:** Git (GitHub, GitLab), CI/CD, Linux (advanced user), Bash, Make, CMake;
- **Databases:** Graph database design, low-level file structure implementation;
- **Areas of Expertise:** Reverse Engineering, Algorithm Design, Parser Development, System Architecture;

Professional Experience

Back-end Developer | Charting Project Project Repository

- Designed and developed the core class system architecture for a charting application
- Implemented a high-performance mathematical expression parser supporting 25+ functions and operators
- Created comprehensive internal APIs and interfaces for modular system components
- Built all core structures and non-graphical interface components

Back-end Developer | Graph Database Project Project Repository

- Architected the on-disk file system structure for a custom graph database, ensuring data integrity and efficient access
- Built a 4-layer abstraction hierarchy from ground-up

- Developed robust type system supporting complex data structures of arbitrary size

MIPT Algorithm and Programming Course

- Mastered fundamental algorithms and data structures including sorting, searching, graphs, and trees
- Implemented various advanced data structures: Fibonacci heaps, B-trees, hash tables, and suffix trees
- Solved complex algorithmic problems across multiple domains including graph algorithms, string algorithms, and computational geometry
- **Sample Projects:** Geometry, SuffixTree, ASM FibHeap

MIPT C/C++ Course

- Developed a custom assembler with lexer, parser, and code generation modules
- Implemented standard-compliant C++ containers with allocator support and exception safety guarantees, explored template metaprogramming techniques
- Worked with low-level features, experimented with reflection-like patterns using template magic and preprocessor macros for runtime type information