

Fedor Abakumets

Telegram Bot Developer (vibe coding)

[GitHub](#)

[Telegram \(preferred\)](#)

[Email](#)

EDUCATION

St. Petersburg College "Primorsky"

2022 — 2025

Digital Information Processing Specialist

Self-education

2023 — present

Python, Telegram bots, automation

SKILLS

Frameworks & libraries

[aiogram](#) · [pyTelegramBotAPI](#) · [Jinja2](#)

Architecture

[FSM](#) · [Dependency Injection](#) · [ServiceFactory](#) · [middleware](#)

Databases

[SQLite](#) · [aiosqlite](#) · [PostgreSQL](#) · [Drizzle ORM](#)

Testing

[pytest](#) · [Pydantic](#)

Tools

[Git](#) · [GitHub](#) · [Railway](#) · [Kiro](#) · [OpenAI Codex](#) · [Qwen](#)

Languages (spoken)

Russian — Native · English — Basic

LANGUAGES

Russian

Native

English

Basic

About me

Telegram bot developer from Saint Petersburg, 21 years old. I build bots on commission and for myself — from simple commands to complex systems with FSM, DI, tests and deployment. Working via vibe coding with Kiro, Codex and Qwen — fast and lean.

Key Achievements

- Telegram Bot Builder — visual no-code Telegram bot builder with drag-and-drop and Python code generation (13 GitHub stars)
- game-bot — bot with Minesweeper, dice, user transfers, promo codes and admin panel; 454 tests (pytest), 100% handler coverage
- fluffy-computing-machine — support bot for a VPN service with ticket system and admin rating

Experience

Telegram Bot Builder (personal project) — Telegram Bot Developer (vibe coding) 2024 — present

- Visual no-code Telegram bot builder — user drags nodes onto a canvas, TypeScript + Jinja2 generate ready-to-run Python code (aiogram)
- Code can be copied or the bot launched directly in the browser tab — no local setup needed
- Supports media, keyboards, conditional logic, JSON config, analytics and Railway deployment

Python, aiogram, Jinja2, TypeScript, React, PostgreSQL, Drizzle, Railway

Freelance projects — Telegram Bot Developer (vibe coding) 2024 — present

- game-bot — bot with Minesweeper, dice, user-to-user transfers, promo codes and admin panel
- Full architecture with DI (ServiceFactory), SQLite, FSM, middleware — 454 tests (pytest), 100% handler coverage
- fluffy-computing-machine — support bot for a VPN service with ticket system, request categories and admin rating
- Both projects built from scratch for specific client requirements

Python, aiogram, SQLite, aiosqlite, Pydantic, pytest, FSM, Dependency Injection