Rust Developer

Backend Startup SMB

[Company Name] is looking for a Rust Developer to design and build performant, reliable systems software. You will work on core infrastructure and services where safety, speed, and concurrency are critical. This role is ideal for engineers who are passionate about systems programming and writing code that is both fast and correct.

Key Responsibilities

- Design and implement backend services and libraries in Rust
- Optimize application performance, memory usage, and concurrency
- Write safe, idiomatic Rust code with comprehensive tests
- Contribute to architecture decisions for new systems and modules
- Integrate Rust services with existing infrastructure and APIs
- Participate in code reviews and help establish Rust best practices across the team
- Investigate and resolve production issues related to performance or reliability

Required Skills & Experience

- 2+ years of professional experience writing Rust
- Strong understanding of ownership, borrowing, lifetimes, and trait systems
- Experience with async Rust (Tokio or async-std)
- Familiarity with systems programming concepts (memory management, concurrency, FFI)
- Proficiency with Cargo, Clippy, and Rust testing frameworks
- Experience building and deploying production services
- Solid understanding of networking protocols (HTTP, gRPC, TCP)
- Comfortable with Linux/Unix environments

Nice-to-Have

- Contributions to open-source Rust projects or crates
- Experience with WebAssembly (Wasm) compilation targets
- Background in C or C++ systems programming
- Familiarity with embedded or real-time systems
- Experience with database internals or storage engines

Tech Stack

Rust Tokio Actix-web Serde gRPC / Tonic PostgreSQL Redis Docker Linux

What We Offer

- Competitive salary and equity package
- Flexible remote or hybrid work arrangement
- Health, dental, and vision insurance
- Annual learning and development budget
- Generous PTO policy

Interview Process

1. Recruiter phone screen (30 min)
2. Technical phone screen with a Rust-focused coding exercise (60 min)
3. System design interview focused on concurrency and performance (60 min)
4. Take-home project: build a small Rust service (3-4 hours)
5. Team culture and values interview (45 min)