

Ali Rıza Aynacı

Last updated in July 2025

🏠 Kocaeli/Turkey | ✉ aynacialiriza@gmail.com | ☎ +90 539 232 56 82

🌐 linkedin.com/in/alirizaaynaci | 🐙 github.com/AliRizaAynaci

Summary

I'm an undergraduate student majoring in Computer Engineering, with a focus on backend development. I'm currently looking for opportunities where I can contribute to backend or infrastructure teams, learn from experienced engineers, and continue growing by building real systems used by real people.

Education

Erciyes University, BS in Computer Engineering

Expected Graduation: May 2027

- English Preparatory Class GPA: 82/100
- Transitioned from Electrical & Electronics Engineering to Computer Engineering in 2023

Experience

Software Engineer, Momentum Rocket Team (TEKNOFEST)

2023 – 2025

- Built an Arduino-based telemetry system for rocket flights, collecting real-time data at **5 Hz** from 8+ onboard sensors
- Applied a Kalman filter to smooth altitude readings, improving noise resilience and enabling accurate parachute deployment
- Developed a C# desktop interface to visualize and store telemetry data received via serial communication
- **Led the 5-person** embedded software team, overseeing development, integration, and flight testing

Projects

RLaaS – Rate Limiting as a Service | <https://rlaas.tech> | github.com/AliRizaAynaci/rlaas

Tech Stack: Go, Redis, PostgreSQL, Fiber, Docker, Next.js, OAuth2, Docusaurus

- Developed a production-ready API rate limiting platform with **multi-tenant** project isolation and per-endpoint rule configuration
- Implemented **Redis sharding** with pluggable selection strategies (hash_mod, consistent_hash) and dynamic config caching
- Built a full-featured dashboard (Next.js) with **Google OAuth2** login and JWT-based authentication for secure project and rule management
- **Deployed** via Vercel (frontend/docs) and Heroku (backend/db/cache), enabling integration through the "/check" endpoint
- **Authored** user-facing documentation using **Docusaurus**, covering usage examples, error handling, and integration patterns

GoRL – High-Performance Rate Limiting Library | github.com/AliRizaAynaci/gorl

Tech Stack: Go, Redis, GitHub Actions

- Designed and implemented a modular, **production-grade** rate limiter in Go, supporting Token Bucket, Sliding Window, Fixed Window, and Leaky Bucket algorithms → **Benchmarked at 89–504 ns/op** with low memory usage and allocations (AMD Ryzen 7 4800H)
- Architected clean, interface-based storage for Redis and in-memory (extensible to NATS/SQL)
- Delivered **95%+ test coverage** and CI/CD via GitHub Actions
- Reached **84+ GitHub stars** and published detailed performance benchmarks

Other Projects (archived)

- Java Spring Boot apps: e-commerce backend, flight reservation, and fleet management – included JWT auth, REST APIs, Docker and CI/CD with GitHub Actions

Skills

Programming Languages: Go, Java, C++, SQL

Technologies: Docker, GitHub Actions, Redis, PostgreSQL, Spring Boot, Next.js, Fiber, OAuth2, Docusaurus, Git

Concepts: Clean Architecture, OOP, Design Patterns, Rate Limiting, Sharding, JWT Auth, CI/CD

Technical Writings

on Medium: <https://medium.com/@aynacialiriza>

- Google OAuth2 Login in Go - Complete Guide
- Rate Limiting – Deep Dive
- Concurrency in Go
- CQRS Pattern
- Strategy Pattern
- Onion Architecture