Saruar Nurlybek

Astana, Kazakhstan | saruar.nb@gmail.com | linkedin.com/in/saruar-nurlybek | github.com/saruar-ryskaliyev

Education

Nazarbayev University **Aug. 2021 – Jun. 2025**
Bachelor of Science in Computer Science *Astana, Kazakhstan*

Experience

AI Engineer **Nace.ai** **Dec. 2025 – Present**
AI research startup automating financial audit and accounting workflows for Big4.

- Engineered an autonomous SQL-generating agent that processed **1M+** financial records, leveraging DuckDB and fuzzy matching for data categorization.
- Built an LLM compliance checker that turns regulation PDFs into structured rules and flags violations with cited evidence, reaching ∼**90%** rule-extraction consistency via overlapping-window batching and similarity-based dedup.
- Cut an audit agent's tool calls from **20 failed retries** to a **single successful** run by making the tools reject incomplete calls up front and rewriting the prompt to match what they actually expect.
- Eliminated **Temporal** workflow crashes on size and history limits by storing large payloads once and passing references between steps, cutting inter-step messages from **4MB+** to **a few KB**.
- Built a caching and persistence layer for agentic logs on **Redis** and **AWS S3**, cutting log retrieval latency by **70%**.

Software Engineer **The Fashion People Ltd.** **Jul. 2024 – Jul. 2025**
A fast-growing startup transforming the fashion industry with a scalable multi-tenant e-commerce for European clothing brands.

- Developed a Next.js e-commerce website serving **35,000** monthly active users, building the product catalog, shopping cart, and checkout flow.
- Moved the highest-traffic pages from client-side to server-side rendering, cutting page load times by **3 times** and server load by **10%** through pre-rendering and caching of repeat requests.
- Rewrote a slow aggregation pipeline over **415,000** conversations while porting the backend router from Node.js to **Golang**, cutting response times by **50%**.

Software Engineer Intern **nFactorial Incubator** **Jun. 2024 – Jul. 2024**
10-week incubator focused on building AI startups with mentorship in development, design, and strategy.

- Implemented a real-time car search and alert system parsing over **50,000** marketplace listings, notifying users the moment a new listing matched their criteria.
- Integrated GPT-3.5 with a vector database using **Python**, improving answer relevance on user queries by **40%** over the existing keyword search baseline.
- Secured a **$1,500** Chevron Corporation scholarship, ranking in the **top 5%** of **2000+** applicants.

Projects

bidflow | *Java, gRPC, Protobuf, Gradle, JMH*

- Built the auction itself in **Java**: picks the winning ads out of **64** candidates and computes what each pays in **275 ns**, allocating no memory per request so garbage collection never interrupts a live request.
- Let every server spend from one shared daily ad budget without a network call per request, by handing out short-lived spending allowances instead of a central counter — a **1.91 ns** local check on the request path.
- Built a simulator replaying server crashes, network splits, and out-of-sync clocks from a single seed; proved across **120** scenarios that advertisers are never charged over budget, and caught two protocol bugs.

Ricoh SP 210SU for macOS | *C, Swift, libusb, CUPS*

- Built an open-source macOS driver and app for a printer Ricoh never supported on Mac, reverse-engineering the scanner's USB protocol from the vendor's Windows driver.
- Wrote the print pipeline in **C** that compresses each page into the printer's own binary format, shipped as one installer covering Apple Silicon and Intel.
- Made the scanner appear in Image Capture and other Mac apps through a local-only service, with a SwiftUI app to scan, reorder, and export pages as PNG or PDF.

Rank5 | *Go, Kotlin, Jetpack Compose, WebSockets, PostgreSQL*

- Built a real-time multiplayer game with a **Go** backend and a **Kotlin** Android app, supporting up to **8** players per room over WebSockets.
- Kept games running when players lost connection by adding a **45-second** reconnect window and automatic host handover, instead of ending the game.
- Prevented scoring errors from late answers by tagging each submission with its round and question ID so the server drops outdated ones.

Technical Skills

Languages: Python, Go, Java, C/C++, Kotlin, Swift, TypeScript, JavaScript, SQL
Frameworks: FastAPI, React, Next.js, Express.js, Jetpack Compose, SwiftUI, Temporal, gRPC/Protobuf
Infrastructure/Data: PostgreSQL, MongoDB, Redis, DuckDB, AWS S3, WebSockets, libusb, Git