

# Kaleab Alemayehu

+251-942-323-094 | [kaleabalemayehu04@gmail.com](mailto:kaleabalemayehu04@gmail.com) |

<https://www.linkedin.com/in/kaleab-alemayehu-a3218422a/> | [github.com/kaleabAlemayehu](https://github.com/kaleabAlemayehu)

## EDUCATION

---

### Bahir Dar University

*Bachelor of Engineering in Software Engineering*

Bahir Dar, Ethiopia

*Jun. 2021 – Jun 2025*

## EXPERIENCE

---

### Mid-Level Backend Developer

*Awura Technologies*

Nov 2025 – Present

*Addis Ababa, Ethiopia*

- Engineered a high-performance gRPC API using ConnectRPC and Go, implementing a modular, event-driven architecture to orchestrate real-time P2P payment flows and ledger synchronization.
- Architected an asynchronous notification and state-management system to handle high-concurrency contributions, triggering low-latency payout events upon successful epoch completion.
- Developed a cryptographically secure selection engine utilizing CSPRNG primitives to ensure verifiable fairness and transparency in automated rotating resource allocation cycles.
- Designed a scalable modular monolith for peer-to-peer financial rotation, integrating identity verification and OTP-secured transactions within a strictly typed, event-sourced backend.
- Implemented a RESTful API using state machines and event-driven architecture, publishing messages to RabbitMQ for user notifications.
- Implemented a Payment Service that is idempotent and uses eventually consistency to ensure data integrity.
- Designed a reliable and configurable cron job system with dynamic worker pools to process background tasks and perform batched database updates.
- Applied optimistic locking to maintain data consistency and prevent race conditions in concurrent operations.

### Fullstack Developer

*Alet-Systems ( MSA with LakiPAY )*

Nov 2025 – Feb 2026

*Addis Ababa, Ethiopia*

- Implemented a robust RESTful API for a gifter mini app on LakiPAY, enabling users to send gifts to influencers.
- Designed dynamic worker pools to consume webhook requests from Redis queues and batch database operations, preventing database overload.
- Developed a Golang SDK for seamless LakiPAY integration.
- Built a real-time overlay system for OBS and TikTok Studio using WebSockets to display animated overlays when gifts are sent.
- Implemented an admin panel for the gifter mini app within the LakiPAY Admin Panel using Next.js, ensuring a smooth administrative experience.

### Fullstack Developer

*Minab IT Solutions*

Jul 2024 – Nov 2024

*Addis Ababa, Ethiopia*

- Design and implement a robust web-based administration panel that Hahujobs CMS uses.
- Develop an SDK for ArifPay Integration using Golang.
- Integrate Arifpay into the Internal Product.

### Backend Intern

*Kuraz Technologies*

Jun 2024 - Sep 2024

*Addis Ababa, Ethiopia*

- Develop a Payroll Management System
- Led a team of 6 interns in developing assigned projects under the guidance of an advisor, ensuring successful execution and collaboration throughout the process.

## PROJECTS

---

### Mini CDC (Change Data Capture) Engine | *Golang, PostgreSQL, pgx*

May 2026 – Present

- Built a lightweight Change Data Capture (CDC) engine in Golang by streaming database changes directly from PostgreSQL WAL using logical replication.
- Implemented a replication client using `pgx` and `pglogrepl` to consume and process WAL events in real time.
- Parsed logical decoding output (`wal2json`) into structured events for INSERT, UPDATE, and DELETE operations.

- Designed an event processing pipeline to normalize and stream database changes as JSON for downstream consumers.
- Tracked Log Sequence Numbers (LSN) to ensure reliable event processing and enable resume-from-failure semantics.
- Handled connection failures and reconnections for long-running streaming processes.

**Payment Gateway Service** | *Golang, PostgreSQL, Docker, RabbitMQ, Echo* Dec 2025 – Present

- Built an idempotent payment service using Golang, PostgreSQL, and RabbitMQ.
- Developed RESTful APIs using Go's Echo framework.
- Implemented the **Transactional Outbox Pattern** to handle dual-write consistency and ensure idempotent message publishing.
- Designed dynamic worker pools to process payments asynchronously and deliver results to client webhook endpoints.
- Applied row-level locking to ensure data consistency and prevent race conditions during concurrent transactions.
- Implemented retry queues and dead-letter queues to reliably handle failed payment events.
- Containerized the full development environment using Docker and Docker Compose for consistent and reproducible builds.

**Drivee** | *Golang, Vue.js, PostgreSQL, Docker, Memcached, TailwindCSS* May 2025 – Present

- Building a peer-to-peer car rental platform where users can list and rent vehicles.
- Developed the backend using Go's **net/http** for REST APIs.
- Created the frontend with Vue.js, using TailwindCSS, without any external UI libraries.
- Added Google and Twitter (X) SSO for login and user data access.
- Used Memcached to speed up response times by caching frequent API data.

**Eagle Commerce** | *Golang, NATS, MongoDB, Docker, Microservices* Jul 2025 – Jul 2025

- Designed and developed a distributed e-commerce backend using Hexagonal Architecture. The system consisted of **order-ms**, **payment-ms**, **product-ms**, **user-ms**, and an **api-gateway**.
- Implemented RESTful APIs using **Go-Chi** across all services.
- Utilized NATS for asynchronous, event-driven communication between services.
- Built a robust API Gateway to proxy requests to services and enforce rate limiting.
- Leveraged MongoDB for efficient document-based data storage across services.

**Werewolf Game Bot** | *Go, PostgreSQL, Docker, Event-Driven Architecture, Pub/Sub* Aug 2025 – Present

- Engineered a feature-rich Telegram bot in Go, used by multiple active groups to play the game of Werewolf. The bot leverages a sophisticated state machine to manage the entire game lifecycle, including complex phases, roles, and player actions.
- Architected the application using Clean Architecture principles and a custom Pub/Sub event broker to create a decoupled, concurrent, and maintainable codebase.
- Authored a public blog post detailing the design and implementation of the event-driven model. [blog](#).
- Drove a 4x reduction in hosting costs by analyzing performance and optimizing server resource allocation.
- Integrated PostgreSQL for persistent data storage, using **goose** for schema management and **sqlc** to generate type-safe Go code from raw SQL, improving reliability and developer productivity.

**Foodopia** | *Nuxt.js, Golang, PostgreSQL, Hasura, Apollo GraphQL, Docker, Flowbite* Jun 2024 – Jul 2024

- Built a recipe-sharing platform with user authentication, recipe creation, editing, filtering, and bookmarking.
- Used Hasura as the GraphQL engine and Gin (Go) for custom webhook handlers with a PostgreSQL database.
- Developed the frontend using Nuxt.js with Flowbite as the UI component library.

**Pong** | *Golang, Go-Raylib* Mar 2025 – Mar 2025

- Built a multiplayer Pong game over LAN.
- Developed the client side using Go-Raylib, a Go binding for the Raylib game library.
- Created the server side with Golang and handled game state synchronization using UDP.

**Audiobook Downloader Bot** | *Grammy.js, TypeScript, JSDOM* Aug 2024

- Built a Telegram bot that downloads audiobooks by scraping search results from the web.
- Implemented inline keyboard interactions using Grammy.js and TypeScript.
- Built a web scraper using JSDOM and the Fetch API to extract audiobook links from search results.

## TECHNICAL SKILLS

---

**Languages:** Go, Typescript, Javascript, SQL (Postgres), Python, Lua, C/C++, HTML/CSS

**Frameworks:** Vue, React, Nuxt, Next, Gin, Echo, Hugo, Node.js, TailwindCSS, headless-UI, Shadcn/ui, flowbite

**Developer Tools:** Git, Docker, Neovim, Linux, Nix, VS Code

**Technologies:** Kafka, NATS, RabbitMQ, GraphQL, PostgreSQL, SQLite, MongoDB, Redis, Memcached