

Harshana Lakshara Fernando

📍 Panadura, Sri Lanka • 📞 +94 783 310 520 • ✉ 2020thlf@gmail.com • ✉ lakshara.21@cse.mrt.ac.lk
🌐 Portfolio • 🔗 LinkedIn • 🐙 GitHub

Summary

Backend-focused Computer Science Engineering undergraduate with a rigorous foundation in **System Design**, **Distributed Systems**, and **Cloud-Native Engineering**. Proven experience architecting high-performance, fault-tolerant applications using **Microservices**, **Serverless Architectures**, and **Event-Driven Design** with **Java (Spring Boot)**, **Go**, **AWS Lambda**, and **Kafka**. Validated through the delivery of high-throughput systems handling massive load, seeking to solve complex **concurrency** and **low-latency** challenges within high-velocity engineering teams.

Education

B.Sc. (Hons) in Engineering – Computer Science and Engineering

2021 – Present

University of Moratuwa, Sri Lanka

Relevant coursework: Programming Fundamentals, Program Construction (OOP), Data Structures and Algorithms, Advanced Algorithms and Advanced Data Structures, Parallel and Concurrent Programming, Operating Systems, Programming Languages, Theory of Computing, Advanced Software Engineering, Database Systems, Computer Networks, Computer Architecture, Calculus, Linear Algebra, Numerical Methods, Operational Research, Statistics, Introduction to AI, IoT Devices

Experience

Software Engineer Intern (Backend Focus)

Dec 2024 – May 2025

Information Systems Associates (Pvt) Ltd — Aviation Technology Company

- Engineered a distributed, high-throughput unique ID generator for Passenger Name Records (PNRs) using a Kafka-backed pool model, ensuring uniqueness and security across scaled instances.
- Optimized the PNR generation service to achieve ****25+ PNRs/sec throughput**** per node with ****<100ms latency****.
- Architected and implemented a hybrid integration strategy by embedding a gRPC client and Kafka consumer into a legacy Java 8 monolith.
- Investigated and refactored inefficient Kafka Streams processing logic, building a proof-of-concept to validate a more efficient Producer-Consumer model.
- Developed dynamic, database-driven feature toggles to control PNR routing logic.

Keywords: Java 18, Spring Boot, gRPC, Kafka, REST API, Couchbase, Oracle DB, PostgreSQL, Microservices, Distributed Systems.

Projects (All projects 📁)

1. Employee Management System (EMS) (2025 | Individual)

Architecture | Github 📄

Designed and built a comprehensive, cloud-native microservices application for employee and department management, featuring a full-stack architecture with a React/TypeScript frontend and Spring Boot backend.

- Architected a distributed system with independent microservices for employees, departments, and configuration, using **Spring Cloud** for service discovery (**Eureka**), centralized configuration, and an API gateway.
- Implemented robust fault tolerance using **Resilience4j**, incorporating **Circuit Breaker**, **Retry**, and **Rate Limiter** patterns to ensure system stability during partial failures.
- Enabled dynamic configuration management using **Spring Cloud Config Server** and **Spring Cloud Bus** with **RabbitMQ** for real-time property updates across services.
- Established a robust observability stack, integrating **Zipkin/Sleuth** for distributed tracing, the **ELK Stack** for centralized logging, and **Prometheus/Grafana** for real-time metrics and monitoring.
- Containerized all services using **Docker** and orchestrated local deployments with **Docker Compose**, while targeting **Kubernetes (OKE)** for production.
- Designed and managed a CI/CD pipeline using **GitHub Actions**, with **ArgoCD** for GitOps-based deployments to the Kubernetes cluster.

Keywords: Java 17, Spring Boot, Spring Cloud, React, TypeScript, Vite, Nginx, RabbitMQ, Resilience4j, MySQL, Docker, Kubernetes, GitHub Actions, ArgoCD, Prometheus, Grafana, ELK Stack, Nexus, GHCR

2. GoTogether – Intelligent Travel Companion for Sri Lanka (2024 | Group) [Architecture](#) | [Live](#) | [Github](#)

Developed and deployed a full-stack travel assistant platform tailored for Sri Lankan tourism, offering AI-based itinerary generation, real-time transport data, and social interaction features through a cloud-native microservices architecture.

- Led development of the web frontend using **Next.js** with secure authentication, session handling via cookies, SSR/CSR hybrid rendering, and route protection for authenticated access.
- Designed and implemented the **User Management module** within the **Social Media Service (Spring Boot)** to manage profiles, preferences, and social connections, integrating securely with **Keycloak** for identity management.
- Built a dedicated **Auth Service using Keycloak**, managing OAuth2/OIDC flows, token issuance, and automated realm/client boot-time import scripts.
- Solved startup dependency issues between services by writing a custom wait script to delay the social media service until Keycloak initialization completed, improving reliability.
- Configured **HTTPS for production with NGINX + Certbot + DuckDNS**, ensuring compatibility with Vercel's secure API policies.
- Developed the **API Service (Go)** to provide AI-based itinerary generation (via Gemini API) and map services using **gRPC** for high-performance inter-service communication.
- Optimized container builds using multi-stage Docker builds (Go service) and manual JAR injection (Spring Boot service) to reduce image size and speed up builds.
- Created a robust **deploy.sh script** to support local builds, selective service orchestration, and GitHub Actions-based CI/CD.
- Migrated backend infrastructure from **AWS EC2 to Oracle Cloud Free Tier**, improving stability, memory headroom, and deployment speed.

Keywords: Next.js, React Native, Go, Spring Boot, Kong, Keycloak, PostgreSQL, Cloudflare, Vercel, NGINX, Certbot, DuckDNS, Gemini API

3. Food Delivery Application (2024 | Group) [Architecture](#) | [Github](#)

Built a scalable, event-driven food delivery platform inspired by systems like Uber Eats, using a modern microservices architecture. As part of the core team, I focused on designing and developing the Restaurant Service.

- Developed the Restaurant Module with secure sign-up/login using **Spring Security** and **JWT** (with extended claims).
- Built CRUD APIs for menu and item management; integrated **Redis caching** to improve performance and reduce database load, with appropriate cache invalidation strategies.
- Enabled order sync via **Kafka**: when an order is placed, an event is published to the 'order_created' topic, which the restaurant service consumes to update order details.
- Implemented real-time order state updates through Kafka's publish/subscribe model to keep restaurant and rider services in sync.
- Integrated **WebSockets** for real-time restaurant notifications (e.g., new orders, rider assignments).
- Built a secure "Forgot Password" flow using time-sensitive tokens for authentication recovery.
- Implemented robust exception handling and API protection through middleware guards and JWT verification.
- Dockerized all components (frontend, backend, Kafka, Redis, PostgreSQL) and managed them via **Docker Compose**.
- Worked in Agile sprints using **Jira** for task tracking and **Bitbucket** for version control.

Keywords: Java, Spring Boot, gRPC, Kafka, Redis, PostgreSQL, WebSocket, Docker, React.js, Ant Design

4. Serverless Auction Platform (2025 | Individual) [Architecture](#) | [Github](#)

A capstone project built using AWS, following a microservices architecture with an event-driven, serverless (FaaS) design.

- Architected an event-driven microservices system (Auctions, Auth, Notifications) using **AWS Lambda**, **API Gateway**, and **EventBridge** for scheduled tasks.
- Implemented asynchronous service communication using **SQS** and **SNS** to decouple components and handle email notifications via **SES**.
- Designed a high-performance data model in **DynamoDB** utilizing **Global Secondary Indexes (GSI)** for efficient scheduled querying and **On-Demand Capacity** for auto-scaling.
- Secured API endpoints with a custom **Lambda Authorizer** that validates **JSON Web Tokens (JWT)**, ensuring robust role-based access control.
- Enhanced code reliability and maintainability by integrating **Middy** middleware for input validation and error handling.

- Utilized **LocalStack** and **Docker** to create a comprehensive local development environment that emulates AWS cloud infrastructure.

Keywords: Node.js, Serverless Framework, AWS Lambda, DynamoDB, SQS, SNS, SES, JWT, Middy, LocalStack, Docker

5. Go gRPC GraphQL – Multi-Tenant Microservices (2025 | Individual)

Architected a **polyglot, multi-tenant microservices** platform with a focus on scalability, fault isolation, and unified data access.

- Developed backend services in **Go** using **gRPC** and **Protobuf** for efficient inter-service communication.
- Implemented a **GraphQL API Gateway** to provide a single, unified endpoint for client applications.
- Designed multi-tenant data storage with **PostgreSQL**, **Elasticsearch**, and optional **Cassandra** for high-volume workloads.
- Containerized and orchestrated services using **Docker Compose** for local and staging environments.

Keywords: Go, gRPC, GraphQL, Protobuf, PostgreSQL, Elasticsearch, Cassandra, Docker Compose

6. Fintech & Crypto Trading Automation Lab (Ongoing | Group)

Industry-Mentored Initiative (Ongoing)

Selected to participate in an exclusive, industry-mentored initiative focused on cutting-edge development in fintech, virtual assets, and derivative markets.

- Selected based on strong development skills to work on intelligent automation and production-level projects with real-world deadlines.
- Developing high-performance trading automation tools and fintech solutions with opportunities for mutual ownership.
- Gaining deep exposure to technical analysis (TA), algorithmic trading, and asset management strategies across crypto and forex markets.
- Receiving direct industrial mentorship aimed at long-term growth in personal finance, wealth management, and professional software engineering practices.

Keywords: Fintech, Crypto Trading, Forex, Algorithmic Trading, Technical Analysis, Intelligent Automation, Industrial Mentorship, Personal Finance & Wealth Management

7. Improving Blockchain Scalability: Throughput Enhancement in ZK-Rollups (Ongoing | FYP)

Final Year Research Project (Ongoing) — BSc (Hons) in Computer Science & Engineering, University of Moratuwa

- Designed and implementing a modular ZK-Rollup prototype to empirically study the performance and economic trade-offs of batching, sequencing, and data availability strategies on Ethereum Layer 2.
- Architecting a runtime-swappable ZK-Rollup framework with clearly separated modules for batching, sequencing, state execution, proving, and data availability, to support controlled, reproducible experimentation.
- Implementing core rollup components including an off-chain sequencer, a Sparse Merkle Tree-based state manager, zk-SNARK circuits in Circom, and Solidity bridge + verifier contracts deployed on Ethereum testnets.
- Building a dedicated benchmarking harness to drive synthetic workloads and measure throughput (TPS), latency, gas per transaction, and proof time under different configurations.
- Evaluating the impact of batch size and batch frequency, sequencer scheduling policies (e.g., FIFO vs fee/priority-based), and data availability modes (L1 calldata vs EIP-4844 blob transactions) on end-to-end performance and cost.
- Generating empirical trade-off curves (e.g., throughput vs latency vs gas/tx) and configuration insights to guide the economic optimization of ZK-Rollup systems in the post-EIP-4844 environment.
- Planning to release the prototype, benchmarking scripts, and datasets as open source to support future research on rollup scalability and protocol tuning.

Keywords: Ethereum, Layer 2, ZK-Rollups, Zero-Knowledge Proofs, zk-SNARKs, Circom, Groth16/Plonk, Solidity, TypeScript, Node.js, Sparse Merkle Trees, EIP-4844, Blockchain Scalability, Performance Benchmarking, Docker

8. Blockchain Supply Chain Management DApp (2025 | Individual)

Architecture | Live 

| **Github** 

Built a full-stack decentralized application (DApp) enabling end-to-end shipment tracking and secure escrow payments on the Ethereum blockchain (Polygon Amoy Testnet).

- Developed a production-grade Solidity smart contract (**Tracking.sol**) using **structs**, **mappings**, and a strict 3-state **enum** lifecycle (PENDING → IN_TRANSIT → DELIVERED), complete with guarded state transitions, timestamped event logs, and full on-chain auditability.

- Implemented a trustless escrow mechanism using `payable` functions—validating `msg.value`, locking funds on creation, and releasing payments only upon verified delivery via `completeShipment`.
- Built a responsive Next.js/React frontend powered by a global `TrackingContext`, providing real-time state management, shipment inspection, and lifecycle actions (create, start, complete).
- Integrated `Ethers.js` and `Web3Modal` for secure wallet connection, dynamic provider/signer selection, MetaMask interaction, and seamless network handling (Localhost Polygon Amoy).
- Added data-formatting utilities for user-friendly display, including UNIX→date conversion and Ether/Wei transformations using `ethers.utils.formatEther`.
- Implemented polished UI/UX behaviors—loading indicators, masked modals (`maskClosable=false`), and utility helpers (`shortenAddress`, `copyToClipboard`) for high-quality user interaction.
- Designed a manual CI/CD pipeline using GitHub Actions to automate contract deployment (with pre-flight balance checks) and frontend deployment to Vercel, fully parameterized via GitHub Secrets and environment isolation.
- Utilized Hardhat for development, including local blockchain simulation (`npx hardhat node`), compilation, contract verification, and controlled deployment workflows.

Keywords: Solidity, Hardhat, Ethers.js, Web3Modal, Next.js, React Context API, Tailwind CSS, Polygon Amoy, Smart Contracts, Escrow Payments, CI/CD

Other Relevant Projects

9. Automated AI Shorts Video Creator (2025 | Individual)

An open-source project to automate the creation of short-form videos using a powerful combination of AI tools and workflow automation.

- Orchestrated a video creation pipeline using `n8n` as the central workflow engine.
- Containerized the entire stack using **Docker Compose** for one-click local deployment.

Keywords: n8n, Docker, AI, Workflow Automation, Video Generation

10. RPAL Interpreter – Functional Language Compiler (2024 | Group)

Engineered a complete compiler and interpreter for the RPAL language from scratch.

- Developed a Lexical Analyzer, AST/ST generator, and a CSE Machine for evaluation.
- Built a robust testing framework with Pytest covering over 168 test cases.

Keywords: Python, Pytest, GitHub Actions, Makefile

11. AI Blog Post Generator (2025 | Individual)


A multi-agent system that uses specialized AI agents to generate professional, well-structured, and well-researched blog posts.

- Built with the Agno Framework for its beginner-friendly approach to multi-agent workflows.
- Designed for provider flexibility, with easy switching between OpenAI, Gemini, Claude, and Groq models.

Keywords: Streamlit, Agno Framework, Pydantic

12. Maison Vella – E-Commerce Admin Panel (2024 | Individual)

Architecture | Live 

| Github 

Comprehensive e-commerce admin interface built with Next.js 14, Server Actions, and tRPC.

- Implemented CQRS and layered architecture with Supabase, Prisma, and Sanity CMS.
- Optimized assets with Cloudflare R2 and secured with Clerk Auth.

Keywords: Next.js 14, Server Actions, tRPC, Supabase, Prisma, Sanity, Cloudflare R2

13. Content-Aware Meme Generator (2025 | Individual)

An automated system for generating contextually relevant memes from any URL, leveraging LLMs and the Memegen.link API.

- Orchestrated a multi-step meme generation workflow using LangGraph.
- Utilized Llama 3.3 (via Groq API) for content analysis and text creation.

Keywords: Streamlit, LangGraph, Llama 3.3, Groq

14. PaperMind – AI Research Assistant (2025 | Individual)

Developed a lightweight, RAG-powered platform for intelligent querying of academic papers.

- Enabled PDF uploads and arXiv integration for paper retrieval.
- Designed a LangChain pipeline with ChromaDB and Sentence Transformers for fast, context-aware lookups.

Keywords: Streamlit, LangChain, ChromaDB, Sentence Transformers, Gemini API

15. Virtual Me – 3D Animated AI Chatbot (2024 | Individual)



Created an interactive 3D virtual assistant with lifelike animations and AI-driven conversations.

- Integrated Hugging Face (Mixtral) for dialogue and Eleven Labs for TTS.
- Built an immersive 3D interface with React and Three.js.

Keywords: React, Three.js, Node.js, Hugging Face, Eleven Labs

Professional Training, Coursework & Certificates (All credentials)

- **Ultimate AWS Certified Solutions Architect Associate SAA-C03 (Exam Prep Course)** *Aug 2025*
Udemy — Instructor: Stephane Maarek [View certificate](#)
- **Certified Kubernetes Administrator (CKA) with Practice Tests (Exam Prep Course)** *Apr 2025*
Udemy / KodeKloud — Instructor: Mumshad Mannambeth [View certificate](#)
- **Apache Kafka Series (v3)** *Mar 2025*
Udemy — Instructor: Stephane Maarek [Part 1](#) | [Part 2](#) | [Part 3](#)
- **100 Days of DevOps** *Nov 2025*
KodeKloud — Trainer: Mumshad Mannambeth [View certificate](#)
- **Implementing Software Architecture for Large-Scale Systems** *Apr 2025*
Udemy [View certificate](#)
- **Advanced OAuth Security** *Mar 2024*
Udemy [View certificate](#)
- **Next.js: Build Scalable React Apps with Page & App Routers** *Nov 2025*
Udemy — Instructors: Anton Voroniuk, Denys Kohut, Anton Voroniuk Support [View certificate](#)
- **Fundamentals of Deep Learning** *Dec 2025*
NVIDIA Deep Learning Institute (DLI) [View certificate](#)
- **Large Language Models (Level 1–3)** *Nov 2025*
Udemy — H2O.ai University (Instructor: Andreea Turcu) [Level 1](#) | [Level 2](#) | [Level 3](#)
- **Mastering Solidity, the Ethereum Programming Language** *In progress*
Udemy
- **Cardano Blockchain Certified Associate (CBCA)** *In progress*
Cardano Academy

Skills

Languages: Java, Go, Python, TypeScript, Solidity

Backend: Spring Boot, Spring Cloud, Spring Security, Express.js, Gin, Resilience4j, Node.js

Frontend: React, Next.js, React Native, Ant Design, ShadCN UI, Three.js, Vite

APIs , Messaging & Protocols: REST, gRPC, tRPC, WebSockets, Protobuf, Kafka, RabbitMQ JWT, OAuth 2.0, Kong

Databases & Caching: PostgreSQL, MySQL, Oracle DB, Oracle SQL, DynamoDB, Redis, Elasticsearch, Cassandra, Couchbase, Supabase, Hibernate, Prisma

Cloud & DevOps: AWS , OCI, Docker, Docker Compose, Kubernetes, Helm, Terraform, ArgoCD, GitHub Actions, Jenkins, Serverless Framework, Nginx, Nexus, GHCR, Certbot, DuckDNS

Blockchain: Ethereum, Solidity, Hardhat, Web3.js

Tools & Concepts: Git, System Design, Distributed Systems, Microservices, Event-Driven Architecture, Jira, JMeter, Gatling, Gradle, Ant, Confluence, Bitbucket, Pytest, Makefile, Postman,

References available upon request.