Harshana Lakshara Fernando

✉ 2020thlf@gmail.com | ✉ lakshara.21@cse.mrt.ac.lk | ☎ +94 783 310 520

103, Gimpatha Road, Werawaththa, Panadura

GitHub Profile | Hugging Face Profile | Replit Profile | LinkedIn Profile

Portfolio Website

Summary

Dynamic and results-driven Computer Science undergraduate at the University of Moratuwa, specializing in building scalable, cloud-native applications with expertise in full-stack development, microservices architecture, and event-driven systems using Java (Spring Boot), Go, and modern tools. Passionate about Generative AI, with hands-on experience in RAG pipelines, LLM integration, Web3, and DevOps. Proven track record in distributed systems design, system performance optimization, and innovative projects like AI-driven travel assistants and blockchain governance platforms. Currently pursuing CKA and AWS Solutions Architect certifications to advance cloud-native proficiency.

Education

B.Sc. (Hons) in Engineering – Computer Science and Engineering *2021 – Present*

University of Moratuwa, Sri Lanka

Relevant Coursework: Object-Oriented Programming, Databases, Statistics, Data Structures and Algorithms, Operating Systems, Computer Networks, Programming Languages, Introduction to AI, IoT Devices, Software Engineering, Image Processing, Computer Architecture

High School *2006 – 2019*

St. John's College, Panadura, Sri Lanka

Experience

Software Engineer Intern *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 while adhering to aviation domain standards.
- Optimized the PNR generation service to achieve **25+ PNRs/sec throughput** per node with **<100ms latency**, validating performance and resilience through extensive load, stress, and spike testing with Apache JMeter.
- Resolved critical production issues, including Kafka consumer lag and offset resetting, by implementing robust shutdown hooks and enhanced retry mechanisms to improve fault tolerance and system recoverability.
- Architected and implemented a hybrid integration strategy by embedding a gRPC client and Kafka consumer into a legacy Java 8 monolith (AeroMart), enabling seamless, backward-compatible communication with the new AeroOrder microservices platform.
- Developed dynamic, database-driven feature toggles to control PNR routing logic, minimizing risk during the phased production rollout by allowing for granular control and immediate fallback.
- Led Research and Development for asynchronous load testing, establishing a monitoring stack with Prometheus and Grafana to collect and visualize key performance metrics like queue time and latency.
- Investigated and refactored inefficient Kafka Streams processing logic, building a proof-of-concept to validate a more efficient Producer-Consumer model that significantly reduced consumer lag and improved system stability.

- Modernized the logging infrastructure by implementing structured JSON logging, which removed legacy dependencies and improved log parsing efficiency across all microservices.
- Authored comprehensive design documents, sequence diagrams, and integration plans on Confluence; facilitated knowledge transfer sessions to ensure seamless handover to the core development team.
- Actively contributed to the agile development lifecycle, participating in all ceremonies, performing code reviews, and collaborating with cross-functional teams to deliver production-ready features using Jira and Bitbucket.

Keywords: Java 18, Java 8, Spring Boot, Spring Security, Ant, gRPC, Kafka, React, Protobuf, REST API, Couchbase, Oracle DB, Oracle SQL, PostgreSQL, JMeter, Gatling, Ant Design, Docker, Docker Compose, Gradle, Prometheus, Grafana, Kibana, Confluence, Jira, Bitbucket, Microservices, Kafka Streams, JSON

Significant Projects

Employee Management System (EMS)

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 both synchronous inter-service communication with **Feign Clients** and asynchronous messaging with **RabbitMQ**.
- 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**.
- Designed and managed a CI/CD pipeline using **Jenkins**, with **ArgoCD** for GitOps-based deployments to a production-like **Kubernetes (OKE)** cluster.

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

GoTogether – Intelligent Travel Companion for Sri Lanka

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 HTTP-only cookies, SSR/CSR hybrid rendering, and route protection for role-based access.
- Designed and implemented the **User 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 user service until Keycloak initialization completed, improving reliability.
- Configured **HTTPS for production with NGINX + Certbot + DuckDNS**, ensuring compatibility with Vercel's secure API policies.

- Refactored the Social Media Service, improving modularity, adding pagination, and enhancing error handling for REST APIs.
- Optimized container builds using .dockerignore and multi-stage Docker builds; manually built JARs outside Docker 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, PostgreSQL, Cloudflare, Vercel, NGINX, Certbot, DuckDNS, Gemini API

Food Delivery Application

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 ”Forget 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

Serverless Auction Platform

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

- Architected an event-driven system with independently deployable microservices for auctions, authentication, and notifications.
- Leveraged a full suite of AWS services, including **Lambda**, **API Gateway**, **DynamoDB**, **SQS**, **SNS**, and **SES**.
- Implemented secure, token-based authentication using **JWT** and a custom **Lambda Authorizer**.

- Utilized the **Serverless Framework** for streamlined deployment and infrastructure management on AWS.

Keywords: Node.js, Serverless Framework, AWS Lambda, API Gateway, DynamoDB, SQS, SNS, SES, JWT

Automated AI Shorts Video Creator

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.
- Integrated various AI services for script generation, text-to-speech, and content preparation, including **Together AI**, **OpenRouter**, and **ElevenLabs**.
- Containerized the entire stack using **Docker Compose** for one-click local deployment.
- Implemented notifications via **Telegram** to monitor the automation workflow.

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

Other Projects

E-Commerce Platform – Admin

Cloud-native admin system for managing products, orders, and inventory.

- Developed a Next.js frontend with ShadCN UI, and a backend with Prisma and Supabase.
- Integrated Sanity as a Headless CMS and used architectural patterns like CQRS and the Repository Pattern.

Keywords: Next.js, Sanity CMS, Supabase, Prisma, CQRS

Virtual Me – 3D Animated AI Chatbot

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

Governance DAO – Token-Weighted Voting DApp

Developed a decentralized governance platform with token-weighted voting using Solidity smart contracts.

- Implemented an ERC-20 token with a faucet system and a VotingDAO contract for proposals and voting.
- Deployed and tested contracts on the Ethereum network using Hardhat.

Keywords: Ethereum, Solidity, Web3.js, Hardhat, OpenZeppelin

Go gRPC GraphQL – Multi-Tenant Microservices

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

- Developed backend services in Go using gRPC and Protobuf for inter-service communication.
- Implemented a GraphQL API Gateway and designed multi-tenant data storage with PostgreSQL and Elasticsearch.

Keywords: Go, gRPC, GraphQL, Protobuf, PostgreSQL, Elasticsearch

PaperMind – AI Research Assistant

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

RPAL Interpreter – Functional Language Compiler

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

Content-Aware Meme Generator

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

AI Blog Post Generator

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

Automated AI Shorts Video Creator

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