

# Ali Shoumar

SOFTWARE DEVELOPER

Beirut - Lebanon

☎ (+961) 76 033 179 | ✉ aleshoumar@gmail.com | 🏠 alishoumar-portfolio.vercel.app/ | 📷 shoumarali | 🌐 ali-shoumar

## Summary

Software Developer with a strong passion for building scalable, and maintainable systems. Particularly interested in distributed systems, cloud platforms, and AI driven solutions. Committed to writing clean code, applying solid design principles, and continuously improving technical expertise.

## Work Experience

### FOO SAL

Sin el Fi, Lebanon

SOFTWARE DEVELOPER INTERN (ANDROID/JAVA)

Jun. 2024 - Sep. 2024

- Developed an Android application following the clean MVVM architecture, consuming third-party REST APIs.
- Conducted research on Spring's Aspect Oriented Programming (AOP) for Android implementation and its potential benefits.
- Learned and applied unit testing fundamentals, including test doubles, Mockito, and JUnit.
- Created custom, reusable views for an in-house framework.

## Education & Certificates

B.S. in Computer Science - Al Maaref University

Sep. 2021 - Feb. 2025

AWS Certified Cloud Practitioner – **Verify Credential**

Feb. 2026

## Skills

**Languages** C#, JavaScript, TypeScript, Java

**Backend** Linux, DotNet Core, EF Core, Identity, Spring Boot, Hibernate, Spring Cloud, MySQL, Docker

**Frontend** HTML, CSS, Angular, Android(XML & Compose)

## Projects

### Linkly | GitHub Repository

ASP.NET 8, C#, SQL SERVER, CLEAN ARCHITECTURE, CQRS, MEDIATR, JWT, BACKGROUND SERVICES

- Developed a scalable Web API using ASP.NET 8 and SQL Server following Clean Architecture and CQRS with MediatR.
- Implemented Repository Pattern and Unit of Work to ensure data consistency, maintainability, and testability.
- Built secure authentication with JWT, refresh tokens, email OTP 2FA, user verification, and protected OTP resend tokens.
- Designed background services for queued email processing and automated cleanup of expired or used OTP records.
- Implemented centralized global exception handling middleware with custom exception types and standardized API responses.

### Leave Management System | GitHub Repository

ASP.NET 8, C#, SQL SERVER, MVC, LAYERED ARCHITECTURE, ROLE-BASED ACCESS CONTROL

- Built a full-stack MVC web application using ASP.NET 8 and SQL Server following a layered architecture approach.
- Implemented role-based access control (Admin, Supervisor, Employee) to manage leave workflows and approvals.
- Automated leave assignment upon user registration based on predefined leave periods to streamline leave tracking.

### Banking Microservices | GitHub Repository

JAVA, SPRING BOOT, SPRING CLOUD, KEYCLOAK, DOCKER, DOCKER COMPOSE, PROMETHEUS, GRAFANA

- Refactored legacy banking functionality into Spring Boot microservices to eliminate monolithic scaling limitations.
- Centralized configuration using Spring Cloud Config and secured services via Keycloak-authenticated Spring Cloud Gateway.
- Containerized services using Jib and Docker, orchestrated with Docker Compose for local distributed deployment.
- Enhanced resilience and observability using Eureka, Resilience4j, Micrometer, Prometheus, Loki, and Grafana.

### Personal Portfolio | GitHub Repository | Demo

TYPESCRIPT, ANGULAR, RXJS, HTTPCLIENT, SCSS

- Developed a dynamic personal portfolio using Angular with maintainable SCSS styles structured according to the BEM methodology.
- Developed a custom HTTP interceptor to mock backend responses, supporting configurable delays, errors, and persistence.
- Implemented CI/CD pipeline using GitHub Actions to automate Docker containerization, image publishing to Docker Hub.

### VaultPic: A Privacy-Focused Photo Management Solution | GitHub Repository

ANDROID, KOTLIN, DAGGER-HILT, CAMERA-X, COIL, NAVIGATION COMPONENT

- Built biometric-protected gallery app using Clean MVVM supporting secure photo storage (internal/external) across Android 9- and 10+.
- Integrated CameraX, AES encryption, and biometric authentication.
- Developed a secure Coil fetcher with LRU-cached decryption (1/8th heap), 344px downsampling, and RGB\_565 decoding, reducing memory usage by 42.8% at peak (350MB → 200MB) and 34.7% on average (245MB → 160MB) for encrypted media grids.
- Developed security checks in C++ (root, emulator, system integrity) using JNI bindings and integrated with RootBeer (Kotlin).