Omar Qazi *Senior Software Engineer*

me.omarqazi@gmail.com (845) 547-7771 omarqazidev omarqazi.net US Citizen

PROFILE

Senior Software Engineer with 4 years of industry experience skilled in full-stack development with expertise in Node.js, React, JavaScript, TypeScript, Go and AWS. Holds a BS in Software Engineering.

Currently building scalable backend solutions for a high-traffic e-commerce platform. Previously worked with payment, logistics, and authentication systems.

Experienced in backend and frontend technologies, distributed systems, cloud infrastructures, testing methodologies, and DevOps practices, leading feature delivery from concept to release.

PROFESSIONAL EXPERIENCE

Software Engineer 2023 – present

Boohoo Group PLC

A leading fast-fashion e-commerce platform with a strong online presence, serving 18M active customers and boasting $2.21B in sales.

- Led the design and development of multiple enterprise-level AWS Serverless solutions, managing millions of requests, supporting over 5 million active users, ensuring seamless performance even during peak usage periods. (Node.js, TypeScript, AWS Lambda, SQS, SNS, S3)
- Architected backend infrastructure solutions using IaC (Infrastructure as Code) with AWS CDK to provision and manage AWS resources, resulting in automated and accelerated deployments and enhanced scalability.
- Spearheaded the successful integration of PLT products into TikTok, enabling in-app products purchases, driving significant revenue growth through expanded market reach. (Lambda, S3, DynamoDB)
- Designed and implemented an automated product categorization solution to enhance targeted marketing strategies, resulting in improved customer engagement and increased conversion rates by accurately matching customers with relevant products. (System Design, S3, Caching)
- Led the refactor of the warehouse management service from a monolithic architecture to a microservices-based system, resulting in improved scalability and maintainability.
- Led the efforts in upgrading legacy dependencies to modern, native solutions, which improved performance by 50-100%. (Node.js, TypeScript)
- Conducted unit, integration, and infrastructure tests with 98% test coverage, and oversaw end-to-end testing by analyzing logs, monitoring queues, and validating system functionality. (Jest, Cypress, Docker, MongoDB, CloudWatch, SQS, DataDog)
- Enhanced CI/CD workflows to enable ticket tracking, accelerating release cycles and ensuring consistent delivery by providing visibility.
- Implemented automated monitoring and incident response workflows across services using AWS CDK and Squadcast, ensuring rapid detection and resolution of issues. (CDK, TypeScript)
- Monitored post-release activities to ensure seamless operations and promptly addressed any alarms or errors to maintain high system reliability and performance. (CloudWatch, Sentry, Grafana, DataDog)
- Led cross-functional collaboration, including conducting meetings, writing proposal documents, securing stakeholder buy-in, and aligning technical strategies with business goals to remove blockers and ensure efficient feature delivery.
- Conducted code reviews and mentored junior engineers, leading to 3x faster feature delivery.
- Collaborated with DevOps to streamline deployment pipelines and resource management, for continuous delivery.

- Documented and collaborated with the QA team to ensure thorough testing and high product reliability.

Software Engineer
2022 – 2023

Swvl

A unicorn providing tech-enabled mass transit solutions across intercity, intracity, B2B and B2G sectors.

Global Payment Integration and Payment Service Enhancement:
Played an integral role in the redesign and development of SWVL's payment system refactor.

- Designed and implemented a generic payment flow, decoupling existing implementations and restructuring the payment system (SQL database schema re-design, code refactor) to improve scalability and flexibility, resulting in reduction in new payment service integration development time by 90%.
- Integrated payment services (M-PESA and EBANX) for 500K active users, enhancing global payment capabilities.
- Developed backend features for monolithic APIs and microservices, including routes, controllers, validations, versioning, serialization and documentation.
- Conducted rigorous unit testing with Mocha and Chai, API testing using Docker and Postman, End-to-End testing using Android Dev Toolbox and Chrome DevTools, ensuring robust and reliable functionality.
- Ensured high code quality by achieving 95%+ code coverage (SonarQube), eliminating code smells, and resolving quality issues (CodeScene).
- Monitored deployments using GCloud and Kubernetes (Lens), including error resolution via pods logs and SSH access. (GCloud, Kubernetes, SSH)
- Collaborated with other engineers in code reviews and system design discussions for SWVL's monolithic API (430k LOC) and Payment Microservice.
- Enabled database browsing, pod testing, and endpoint verification through port forwarding.
- Facilitated communication and issue resolution for frontend and test engineers.
- Prioritized issues based on urgency, dependency, availability, requirements, and level of collaboration.

Seamless Message Broker Integration:
Enhanced SWVL's internal library to allow developers to switch message brokers without code changes.

- Enhanced SWVL's SQS wrapper library (WTSQS) to support RabbitMQ and Apache Kafka, allowing seamless broker switching via environment variables.
- Conducted research on messaging systems (ActiveMQ, RabbitMQ, Kafka) and communication protocols (REST, STOMP, AMQP), producing detailed comparisons including the differences in architectures, features and limitations.
- Authored integration documentation and developed components for WTKafka and WTRabbitMQ.
- Achieved 95%+ code test coverage, utilized CI/CD pipelines with Jenkins, and ensured high-quality linting adherence.

Partner Portal:
A portal for vendors to manage their SWVL assets.

- Researched on optimizing database query performance.

Software Engineer
2021 – 2022

Tintash

Auth Service:
An authentication and authorization service with role-based access control written in Go.

- Designed and architected an authentication and authorization service using Go and Gorilla Mux.
- Implemented role-based access control (RBAC) for granular resource permissions management.
- Implemented OAuth 2.0, session storage, and JWT token authentication using Google OAuth and Firestore.
- Reduced login session processing time by implementing session caching with Redis.
- Deployed and tested the AuthService backend on GCP and Heroku, and the frontend on Firebase, ensuring seamless integration and reliability across all service layers.

Live-Sessions Platform:
Led the development of a robust, real-time, multi-provider live-sessions platform using the Vonage video API, serving as the primary front-end engineer for the project.

- Designed and developed a live-conferencing platform that allows for real-time collaboration. (ReactJS, NodeJS)
- Developed the front-end client application. (React, TypeScript)
- Implemented global state management. (React Context API)
- Utilized REST APIs for communication between the frontend and backend. (Axios, ExpressJS)
- Employed Tailwind CSS to craft a visually appealing and responsive design.
- Collaborated with backend engineers to to establish API endpoints and request-response structure.

PROJECTS

Thread-Tasks (npm)

Boost performance by executing CPU-intensive tasks in parallel threads. (Node.js)

Switch Git Account

A CLI tool to switch between multiple Git accounts on the same computer. (Go)

Additionally, implemented automated releases using GitHub Actions.

ReactPaint

ReactPaint is a drag-n-drop style page builder that generates ReactJS code. (React, Redux, TypeScript)

Overview:
ReactPaint is a web-based page builder that empowers users to design dynamic web pages. This application utilizes a drag-and-drop interface, enabling users to create and customize web components. ReactPaint generates HTML/CSS and React.js code based on the user's designed layout.

Features:
Interface: Drag-and-drop various components onto the workspace for easy arrangement and layout structuring.
Customization: Edit component contents using the double-click feature and its properties the properties sidebar.
Code Generation: Generate React.js code from the designed layout.
Project Management: Save and load project files, and also export as HTML/CSS, JSON or PNG.

Usage:
The platform boasts an array of components such as containers, text elements, buttons, text fields, images, videos, audio, iframes, and carousels. Users can drag-n-drop those components onto the design area, and extensively customize these components through properties like layout, height, width, margin, padding, background settings, font properties, display preferences, and even add custom CSS or classes.

Technologies:
React, Redux, TypeScript, Immer for Redux, ReactDnD for drag-and-drop functionalities, Bootstrap, UIKit, and custom CSS for styling, among other supporting technologies.

Pokemon API

API for accessing and managing Pokemon data and sorting datasets. (TypeScript, MongoDB)

- Developed an API system focused on secure data access and management.
- Implemented an authentication middleware.
- Created distinct (public and protected) routes within the API.
- Enhanced API functionality by integrating query filters and pagination.
- Developed a solution to extract and store individual datasets into a chosen database system.

- Designed and implemented data sorting algorithms utilizing datasets.
- Enabled cross-dataset search functionality.

Data Warehouse Tools
DWT is a suite of database and data warehouse tools including Faker, ETL and OLAP. (C#, SQL)

Overview:
Developed a comprehensive suite of tools for data management, analysis, and manipulation. This suite includes:

Data Faker and Fabrication Tool: Facilitates data fabrication.
DB Manager Tool: Creates and populates database and data warehouse tables.
ETL (Extract, Transform, Load) Tool: Streamlines data transfer from the database to warehouse.
Data Warehouse Analysis Tool (OLAP): Enables data analysis, querying and export.

EDUCATION

BS Software Engineering
Comsats University

SKILLS

Programming Languages: JavaScript, TypeScript, Go, Java, C#, C, Bash

Backend Development: Node.js, Express.js, Nest.js, Gorilla Mux

Frontend Development: React.js, Redux, Zustand, HTML, CSS, Tailwind, Bootstrap, MUI

Database: SQL (PostgreSQL, MySQL, MSSQL), NoSQL (MongoDB, Firestore, Redis), ORM (Sequelize, Mongoose)

Testing: Jest, Cypress, Mocha, Chai, Sinon, Unit Testing, Integration Testing, E2E Testing

Cloud Services & Platforms: AWS (Lambda, SNS, SQS, S3, DocumentDB, DynamoDB, RDS, EC2, ECS, CloudWatch, CDK, Step Functions, EventBridge, CodePipeline), GCP, Firebase, Heroku

DevOps & Infrastructure: Containerization (Docker, Kubernetes), CI/CD (GitHub Actions, Jenkins, Sonor), Serverless, Cloud Computing

Architecture & Design: OOP (Object-Oriented Programming), Microservices, Distributed Systems, Serverless Architecture, RESTful APIs

Full-Stack Development: Backend, Frontend, Full-Stack

Collaboration: Jira, Confluence, Agile, Scrum, Sprint

Version Control: Git, GitHub, BitBucket