Personal Portfolio of Projects

Faith Olusegun

faitholusegun60@gmail.com | propenster.github.io | github.com/propenster

Abstract

A curated collection of professional and open-source software engineering projects demonstrating expertise in high-performance back-end systems, responsive front-end development, and robust cloud architecture. This portfolio highlights my contributions to financial technology, bioinformatics, and developer tooling.

Professional Experience

Interswitch — Senior Software Engineer — Dec 2023 – Aug 2024

High-Throughput Payment Gateway Engine

Architected and implemented a transaction processing engine designed for extreme low-latency and high availability. The system's core was built in `C++` for performance-critical operations, with a service layer in `C#/.NET 8` for business logic and API integration. It consistently processed over 5,000 transactions per second (TPS) for Interswitch's primary payment gateway, maintaining 99.99% uptime.

Technologies Used: `C++, C#, .NET 8, Redis, Azure Key Vault, OAuth2.0, DDD`

Real-Time Transaction Monitoring Dashboard

Designed and developed a reactive, single-page application for live monitoring of over 10 million daily payment transactions. The dashboard provided real-time analytics, alerts, and visualizations, enabling the operations team to reduce incident response time by 50%.

Technologies Used: `React, TypeScript, SignalR, .NET Core`

Parallex Bank — Software Engineer — Mar 2022 – Nov 2023

Core Banking API Integration

Led the development of a high-volume Core Banking API that served as the backbone for the bank's digital services, supporting over 500,000 customers. This project involved integrating with the Finacle core banking system using the FIXML protocol to ensure secure and standardized financial data exchange.

Technologies Used: `C#, .NET, OracleDB, FIXML, REST APIs`

Automated KYC Workflow System

Engineered an automated Know Your Customer (KYC) workflow that drastically reduced manual verification time by 70%. The system utilized Azure Form Recognizer to extract data from customer documents and stored the verified information in MongoDB.

Technologies Used: `C#, Azure Form Recognizer, MongoDB, Blazor WASM`

Fidelity Bank PLC — Software Engineer **Jan 2019 – Dec 2021**

Loan Management Portal

Developed a full-stack customer-facing portal for managing loan applications, which processed over $1 million in applications monthly. The system was optimized for performance, achieving sub-500ms API latency for a responsive user experience.

Technologies Used: `Angular, .NET Core, SQL Server, Microservices`

Internal DevOps Toolkit for Azure Kubernetes

Created a command-line toolkit to standardize CI/CD pipelines for Azure Kubernetes Service (AKS) deployments. This tool was adopted by over 15 development teams, improving deployment frequency by 3x and streamlining cloud operations.

Technologies Used: `C++, Python, Azure DevOps, Kubernetes (AKS), Docker`

Open Source Projects

A selection of personal projects focused on financial engineering, bioinformatics, and developer utilities.

TopScript

URL: https://github.com/propenster/TopScript

An interpreted, dynamically-typed scripting language written from scratch in C#. TopScript is designed to be simple, easy to learn, and embeddable within larger .NET applications.

Technologies Used: `C#, Compilers, Compiler Design, Parsing, Interpreters`

tinycsharp

URL: https://github.com/propenster/tinycsharp

An educational project building a tiny C# compiler from scratch in C++. It uses modern C++17 features and leverages the LLVM framework for code generation.

Technologies Used: `C++17, LLVM, googletest`

trade_sharpp

URL: https://github.com/propenster/trade_sharpp

A complete high-frequency trading (HFT) suite for the Nigerian Stock Exchange (NGX). This project provides a full-stack solution for market data consumption, order management, and strategy execution using the FIX protocol.

Technologies Used: `C++, C#, FIX Protocol`

NgXQuickFix

URL: https://github.com/propenster/NgXQuickFix

A C# wrapper for the popular C++ QuickFIX library, specifically tailored to provide a simple and robust API for creating Financial Information eXchange (FIX) applications for the Nigerian Stock Exchange (NGX).

Technologies Used: `C++, C#, QuickFIX/N`

VealFramework

URL: https://github.com/propenster/Veal

An experimental cross-platform C# web development framework designed for modern cloud application development on Windows, Mac, and Linux. It offers functionality similar to Microsoft's ASP.NET Core for creating fast and efficient APIs and web applications.

Technologies Used: `C#`

Unity-MCP

URL: https://github.com/propenster/unity-mcp-server

A server application for a multi-user creative platform in Unity. It facilitates real-time, collaborative 3D scene creation using natural language commands processed by Anthropic's Claude AI model.

Technologies Used: `TypeScript, C#, Unity, AI Integration`

wdlrunner

URL: https://github.com/propenster/wdlrunner

A high-performance compiler and runner for the Workflow Description Language (WDL) 1.0, written from scratch in C++. This tool is designed for executing bioinformatics workflows efficiently.

Technologies Used: `C++`

ERVCaller-rs

URL: https://github.com/propenster/ERVCaller-rs

A Rust port of ERVCaller, a clinical virology tool used to detect Endogenous Retroviruses (ERVs) from Next-Generation Sequencing (NGS) data. The Rust implementation focuses on improving performance and memory safety.

Technologies Used: `Rust, Bioinformatics`

FQL (File Query Language)

URL: https://github.com/propenster/FQL

A fast, SQL-like query language for structured text files (e.g., CSV, TSV). Written in Rust, FQL allows for efficient data analysis directly on the command line without needing to load data into a database.

Technologies Used: `Rust, Parsers, CLI Tools`

debigenic

URL: https://hub.docker.com/r/propenster/debigenic

A portable bioinformatics pipeline packaged in a Docker container. It includes over 50 pre-installed and configured tools to ensure reproducible genomic research across different computing environments.

Technologies Used: `Docker, Shell Scripting, Bioinformatics`

Faker.NET

URL: https://github.com/propenster/Faker.NET

A .NET library that generates a wide variety of realistic, fake data for testing and development purposes. It simplifies the process of populating databases and creating mock objects for unit tests.

Technologies Used: `C#, .NET`