Islam Walid

✉ islamwaled33@gmail.com ☎ +201021927023 https://github.com/IslamWalid

https://www.linkedin.com/in/islam-walid-3300/

PROFILE

I am a self-motivated, collaborative, and proactive software engineer, I am an advocate for customer's experience, therefore, I am always interested in building user oriented software products.

I take writing well-designed, high-quality, testable, and well-thought-out software solutions very seriously, I'm always committed to continuously learning and improving my technical skills, I also sought out feedback from my peers and colleagues and actively worked to incorporate feedback into my work to improve the quality.

PROFESSIONAL EXPERIENCE

Software Engineer

Nov 2022 – present

Azzrk

- Implemented microservices within an Agile environment utilizing Scrum methodology.
- Leveraged Golang and Node.js extensively in project development.
- Demonstrated expertise in designing and deploying microservices architecture, employing HTTP and gRPC protocols.
- Established robust CI/CD pipelines for seamless deployment of production services.
- Distinguished in crafting comprehensive unit and integration tests leveraging leading industry tools.
- Spearheaded performance optimization efforts through benchmarking initiatives.
- Produced documentation for code, APIs, and technical procedures to facilitate comprehension and future maintenance.
- Proficiently managed both relational (RDBMS) and NoSQL databases, ensuring optimal data management solutions.
- Exhibited a profound understanding of containerization principles and demonstrated mastery of Linux environments.
- Actively contributed to the advancement of existing and new projects, enriching functionalities and features.
- Engineered high-quality, reusable codebases, fostering efficiency and sustainability across project lifecycles.

Software Engineer Intern

Jul 2022 – Sep 2022

Codescalers

- Implemented a Terraform module to dynamically create VMs and configure them automatically via environment variables to create Kuberentes cluster.
- Implemented Filesystem in Userspace (FUSE) software interface for linux to let non-privileged users create their own Filesystem using Loadable Kernel Modules (LKMs) without editing kernel code.
- Implemented priority-based process manager that runs services specified in YAML file by sorting them topologically according to their dependencies.
- Worked with the team on deploying several internal services using Kubernetes ,Terraform, Docker and Docker Compose.
- Wrote unit tests for all services I have worked on and used CI/CD tools (Github Actions) to apply Test Driven Development (TDD).

SKILLS

Programming
JavaScript, TypeScript, Go, Shell scripting.

Backend
Node.js, Express.js, Nest.js, Gin.

Testing
Test Driven Development, Jest.js, Go test.

Database & ORM
PostgreSQL, Mongo, Redis, Sequelize.js, Gorm.

APIs
gRPC, REST, OpenAPI(Swagger).

Infrastructure
Linux, Docker, Docker-Compose, Kubernetes, Terraform.

PROJECTS

Wildcart

Node.js, PostgreSQL, Express.js, Sequelize.

E-commerce REST-API built with Express.js and uses stripe as a payment gateway.

Codersquare

Node.js, PostgreSQL.

REST-API for a social web app for sharing learning resources in hacker new style by making posts, likes, and comments.

Todo

Go, Gin, gRPC, Mongo, PostgreSQL.

Todo is a service used to create todo applications. Although being simple idea, the project is used to learn and apply all software engineering principles to achieve high levels of scalability, availability and maintainability.

Bitcask

Go.

Bitcask is a log-structured hash table for fast key/value data.

Foreman

Go.

Foreman is a manager for Procfile-based applications, aims to abstract away the details of the Procfile format, and allow you to run your services directly.

Struct to file mapper

Go.

A mapper to an in-memory user data to a filesystem representation using FUSE filesystem.

TContainer

Go.

A Simple container implementation in go using linux namespaces and cgroups.

Posix Shell

C Programming Language, Linux, Shell.

A small Linux shell program that parses commands and executes them just like a normal shell.

Proxy Server

C Programming Language, Socket Programming, HTTP, Multi-Threading.

A simple concurrent proxy that handles HTTP/1.0 GET requests using multi-threading programming that caches recently requested web objects.

Web Server

C Programming Language, Socket Programming, HTTP, Multi-Threading.

Simple, multi-threaded HTTP/1.0 Web server that uses GET method to serve static and dynamic content.

AWARDS

Finalist in ECPC

Aug 2021

ICPC

Egyptian Collegiate Programming Contest

EDUCATION

Bachelor Computer Engineering

Sep 2018 – Aug 2023

Mansoura University