OMAR ELOUI

BACK-END DEVELOPER

Ocario, Egypt 📞 +02 0111 944 7500 (a contact@omareloui.com

PROFILE

Extremely resourceful Back-End Developer with extensive experience in Node.JS and TypeScript, Good decision maker with ability to think critically and provide out of the box solutions; Team player with strong interpersonal skills and able to work to tight deadlines.

EDUCATION

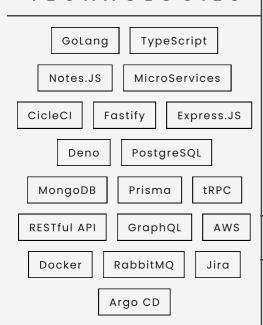
Advanced Full-Stack Web Development Nanodegree

Udacity May, 2022 - July, 2022

BA in Spanish Language

Alsun, University of Ain Shams 2016 - 2022

TECHNOLOGIES



LANGUAGES

Arabic, native speaker.

English, fluent.

Spanish, intermediate.

EMPLOYMENT HISTORY

BACK-END DEVELOPER

March 2023 - Present

Arab African International Bank

- Worked with distributed systems. Develop all MSs using NestJS.
- · Responsible for 2 main projects
- 1. Merchant Portal:
 - I'm one of three people responsible for over 8 services communicating with each other through RabbitMQ.
 - Implement the SAGA pattern over the services to increase data integrity.
 - Wrote a services utility to improve the DX. It's a CLI tool to interact
 with the running MSs from one place (start/stop/restart MS, push
 to remote, run a cli command in the MS, and over 5 more actions).
 - Created a Geo-location MS for the bank MSs to use.
 - Developed a way to interact with the Active Directory to retrieve the bank users information and sign them to the Merchant Portal.
 - Develop TOTP authentication system.
 - Developed over 8 emails using MJML to send complex responsive emails for over 3 parties.
- 2. CVM (Customer Value Management):
 - Design the architecture for the full project.
 - Decide the the stack (GoLang, MongoDB, RabbitMQ, and more...)
 and design patterns (Monorepo, Hexagonal Architecture).
 - Write a code generator for the Monorepo to make easier to create new service, generate Dockerfiles and way more.

PROJECTS

IMAGE PROCESSOR Repo https://github.com/omareloui/ImageProcessorAPI

API to manipulate images

- Create the API with Node.js, TypeScript and Sharp.
- Expose an endpoint to be able to manipulate the image and retrieve the wanted result.
- Cache the images on requesting a change.
- Build a simple frontend if needed to upload an image and manipulate. Rendered from EJS.
- Test the API with Jasmine.