MOUAD EL AMMARI

Full-Stack Engineer

mouad.elammari.dev@gmail.com +212643954762

https://mouadjs.dev https://github.com/Poider

Casablanca, Morocco (Temporary)

English - French

Willing to relocate Q3 2025 (NL) · Requires visa sponsorship.

WHO'S MOUAD

Versatile Full-Stack engineer blending robust backend APIs, intuitive frontends, automated data pipelines, and DevOps practices. I bridge complex tech with clear communication to deliver solutions that scale.

SKILLS

Programming languages and frameworks:

NestJs/ NodeJs	React
JavaScript (ES6)	Python
TypeScript	C++

Tools and technologies:

Postgresql	MongoDb
Git	Rest Api
Docker	GraphQl
Linux	Socket.IO
Css	

EXPERIENCE

August 2023 —present

Full Stack Engineer OCP Africa

- **Built scalable backend APIs** on top of Postgres and intuitive React frontends, while handling the deployments.
- **Developed real-time notifications** OOP based, using WebSockets, enhancing user engagement.
- Crafted front-end components with React and Tailwind, responsive and accessible.
- **Built REST and GraphQL APIs** with clean architectural patterns.
- **Engineered ETL pipelines** in Node.js and Python to automate data ingestion, cleaning, and transformation.

February 2021 —August 2023

Contract-based Projects

Full Stack Engineer OCP GROUP

- **Delivered front-end and back-end solutions,** Building React interfaces and NestJS.
- **Developed high-performance C++ applications** tailored to specific client needs.

EDUCATION

Oct 2017 — Jun 2021

Bachelor's degree

Computer Science, Casablanca

Databases

University Hassan II Casablanca

Nov 2021 — Jun 2023

42 Curriculum of Architect

Digital Technologies

42 Network, 1337 Coding School -

Mohammed VI Polytechnic University

PROJECTS

More projects → mouadjs.dev

- Real-Time PingPong Web App – matchmaking, live chat & gameplay with NestJS, Socket.IO & TypeORM.
- C++ Web Server (Nginx-Style) – custom memory management, SSL/TLS, non-blocking I/O for large file uploads.
- Minishell (Bash-Inspired) – commands, piping, redirection & env-vars in a two-stage C interpreter.
- Tree-Based Data-Structures Library – AVL & Red-Black tree implementations with templated C++ APIs.