Ahmed Driouech

Student | AI/software developer

a.a.driouech@outlook.com https://www.linkedin.com/in/ahmed-driouech-02700a268 Leiden

SUMMARY

Motivated and technically skilled software developer with experience in backend development, API design, and fullstack project delivery. Developed production-level features as a student .NET developer and contributed to multiple academic projects involving AI, web development, and DevOps. Proficient in technologies such as .NET, SQL, Python, Flask, and React, with practical exposure to CI/CD, Docker, Kubernetes, and large language models. Eager to join a forward-thinking team where I can apply and expand my software engineering skills.

EXPERIENCE

.NET student developer

PANGAEA

02/2023 - 09/2023 The Hague, Netherlands

Worked on the OneLeapAhead platform:

- Contributed to the development of PANGAEA's new platform, OneLeapAhead, as part of the backend team.
- Designed and implemented endpoints to support various entities and business logic.
- Worked with SQL databases to create, query, and manage data structures and relationships.

EDUCATION

Master's Degree in Data Science & AI

TU Delft

09/2024 - Present Delft, Netherlands

Bachelor's Degree in Computer Science

Vrije Universiteit Amsterdam

09/2021 - 07/2024 Amsterdam, Netherlands

LANGUAGES

English
Proficient

Dutch
Native

Arabic
Proficient

SKILLS

Python JavaScript HTML/CSS React SQL C# Git Docker Kubernetes

PROJECTS

Fullstack AI app

https://github.com/remla25-team4

We made a web app where restaurant customers can leave a review and trained a model to predict the sentiment of the reviews.

- We deployed the app with kubernetes using Docker and VM's.
- We used React.js/node.js for the frontend and python/Flask for the backend.
- Set up CI/CD workflows using github Actions

Testing AI models for bias

https://github.com/TaoufikElkadi/SWE-Testing4AI

We used an anonymized version of the dataset that was used in the 'toeslagen affaire' to deliberately train a good and a bad model.

- We tested anonymized models of other groups to see which model is the biased one.
- We used black-box testing techniques such as metamorphic testing and equivilance partitioning to test the models for robustness
- We used standard metrics such as accuracy and F1 scores to test performance.

Fine-tuning LLM for code refactoring

https://gitlab.ewi.tudelft.nl/CS4570/2024-2025/teams/team-12

We fine-tuned GPT-4 for code refactoring based on design patterns

- We gathered code from github repositories where design patterns were used to use as examples
- We made some synthetic data with claude that contains before and after examples.
- We used few-shot prompting to finetune the LLM.