Crehul Vlad

vladc12.github.io ← Explore my projects and contributions.

I am a software engineer with a passion for all things technology: software, hardware, electronics, and artificial intelligence. My main professional focus is web development. Especially in React and NextJS.

Phone: **0751141805**
GitHub: github.com/VladC12
Email: **vladcrehul12@gmail.com**
Linkedin: www.linkedin.com/in/crehulvlad

EXPERIENCE

Artificial Intelligence Visual Analytics —*Software Engineer*

09/2021 - PRESENT

- Built full-stack web applications using React, NodeJS and NextJS for both internal use and client products.
- Developed pipelines for synthetic data generation using Python and Blender to be used in AI model training.
- Created 3D models for realistic datasets.
- Deployed our solutions which used edge devices and cloud infrastructure.

Continental Automotive Systems — *Intern Software Developer*

07/2018 - 09/2018

- Tested electronic components, contributing to an ATV prototype.
- Developed software for said prototype.

SKILLS

- **Frontend:** React, Electron
- **Backend & Full-Stack:** NodeJS, Flask, MongoDB, NextJS
- **Programming Languages:** Python, Typescript
- **Other:** Blender, Kafka, ThreeJS

ACCOMPLISHMENTS

Semi-finals: Innovation Labs 2018,
Collaborated with a team of 4 to develop prototype software and electronics for automotive systems, contributing to the successful implementation of 2 device features.

LANGUAGES

- **Romanian**: *Native*
- **English**: *Fluent*

PROJECTS

Web Resume—React, *TypeScript, ThreeJS*

A website accessible via the link at the top of this document. It serves as my preferred way to share my resume, while also showcasing my skills and displaying other projects in a more interactive and visual format.

Caster— *NextJS, TypeScript, Flask*

An internal tool developed at AiVA using NextJS and Flask. Its main purpose was to simplify the process of visually saving image coordinates. Users could save RTSP video streams with thumbnails generated from screenshots. For each stream, they could create directional lines and polygons for regions of interest (ROI) using Canvas. These annotations were then used by AI to detect when people passed specific lines or entered an ROI. Flask handled RTSP streams and OpenCV-related APIs. Additional options, though less central, were also included to enhance functionality.

Event Manager— *NextJS, TypeScript, MongoDB*

A client-facing project built with NextJS, made for AiVA. It visualizes various checkout events, such as item-scanning fraud, by displaying a detailed, filterable table of data. The application allows user creation with different permission levels, ensuring flexible access control for administrators.

Event Watcher— *Electron, React, NodeJS, MongoDB*

A desktop app created using Electron and React, made for AiVA, designed to notify users in real time about checkout fraud. Users can review ~5-second videos of flagged events, along with information such as the SKU barcode, price, and timestamp. It's accompanied by Event Watcher Buddy, which helps manage user accounts, track app versions, and enable software updates. Buddy also supports user creation and handles API integration.

Cradle— *Python, OpenCV, Kafka, Tensorflow, Multiprocessing*

This Python-based application, made for AiVA. It processes RTSP streams from multiple sources, running real-time video inference with machine learning models. The results, which include people detection and other analytical insights, are packaged into payloads for use in scenarios like surveillance systems and traffic monitoring. It's designed for environments where real-time video analysis is critical.

Quartermaster— *NextJS, TypeScript, threeJS*

An internal tool for AiVA, built to help organize large datasets by enabling mass uploads into a structured directory. It supports special cases, like organizing 3D model files with texture dependencies, and allows videos, images, and 3D models to be previewed directly in the browser. Files are stored in a shared folder with a visible file structure generated from JSON files. Users can download entire folders or individual files as archives, with options for immediate download or time-limited links.

EDUCATION

UTCN, Faculty of Electronics Telecommunications and Technology Information — *Bachelor of Engineering*

06/2021

Telecommunication Technologies and System specialization.

BACHELOR THESIS

Tacotron 2 DNN Text-To-Speech Web App— *Flask, Tensorflow*

Implemented a Text-to-Speech system using Tacotron 2 architecture, training models in both Romanian and English. Developed a web interface using Flask to demonstrate the model capabilities. The Romanian model was trained on the Mara Corpus, while the English model utilized the LJSpeech Dataset.