

# TYLER CRANMER

1415 N. Washington St | Denver, Colorado 80203 | 760-644-0291 | [tylerc.developer@gmail.com](mailto:tylerc.developer@gmail.com)

## PROFILE

I am a software engineer who has a strong foundation in core computer science principles, data science, and machine learning. With hands-on experience in building full-stack web applications, developing machine learning models, and creating data visualizations, I am well-equipped to tackle a range of technical challenges.

## SKILLS

Python, TypeScript, JavaScript, PostgreSQL, MySQL, MongoDB, GraphQL, Solidity, Next.js, React, Node.js, Django, Flask, FastAPI, PyTorch, TensorFlow, LangChain, Keras, Web3.py, Ethers.js, AWS, GCP, Heroku, Vercel, Docker, Linux, Bash, Git, GitHub, Tailwind CSS, Mui

[Certified AWS Solutions Architect Associate](#)

[GitHub](#) | [LinkedIn](#) | [Website](#)

## EDUCATION

Professional Masters - Computer Science, Artificial Intelligence Aug 2023 - May 2025  
University of Colorado – Boulder, Colorado

Bachelor of Science – Applied Computer Science Aug 2018 - Dec 2022  
University of Colorado - Boulder, Colorado  
**Cum Laude**

Bachelor of Science – Exercise Science  
Northern Arizona University – Flagstaff, Arizona

## WORK EXPERIENCE

Software Engineer – MKT July 2023 – Present

MKT (Market) is a web3 / blockchain company that develops Dapps on the Canto blockchain ecosystem.

- Built a full-stack Web3 growth attribution and [referral platform](#) for the Canto ecosystem, enabling Dapps to gain insights into on-chain actions related to their products. Technologies used: Python, TypeScript, FastAPI, Web3.py, Next.js, Docker, PostgreSQL.
- Revamped the frontend code for the [Canto identity](#) suite, leading to enhanced design and user experience. Developed using TypeScript, Ethers.js, and Next.js.
- Enhanced and updated the frontend code for the [Canto namespace](#) product, improving design, functionality, and user experience. Implemented with TypeScript, Ethers.js, and Next.js.
- Constructed a NFT indexing platform to capture details about NFTs, their transfers, token types, and follower relationship structures, thereby enhancing the user experience for the Canto identity suite. Built with Python, Web3.py, and PostgreSQL.
- Coauthored comprehensive documentation for the [Canto identity](#) and [Canto namespace](#).
- Developed the [Canto concierge](#) marketing website to bolster the promotion of the Canto ecosystem. Created using TypeScript, Next.js, and Sanity.io.
- Created a [multi-chain asset bridge](#), enabling users to transfer crypto assets to and from the Canto ecosystem. Utilized TypeScript, Next.js, and the Synapse Protocol.

Machine Learning Engineer / Software Engineer – More Seconds

Dec 2022 – August 2023

More Seconds is a web development and digital agency that builds custom solutions for businesses. Currently working on SaaS developer and project manager platform.

- Collected and cleaned developer operation data to create a machine learning regression model that estimated time to completion for specific development tasks to be displayed on the More Second's client project management portal. This was built using Python, XGBoost, CloudFormation, ECS, Docker, S3, SAM CLI and AWS Lambda.
- Developed three different NLP classification models using NLTK, Naïve Bayes and TD-IF to predict the type of web developer tasks, clarification features and its overall complexity for More Seconds project manager portal. Built with Python, Scikit-learn, CloudFormation, ECS, Docker, S3, SAM CLI and AWS Lambda.
- Developed a customized language model using GPT-4 to extract structured information from unstructured client web development requests, enabling automation of developer task creation.
- Developing an AI application using LangChain and GPT-4 that will be used in place of technical project managers to decipher our client's technical requirements and generate clear objective tasks for our web dev team. Built with Python, LangChain and FastAPI.
- Developed, maintained, and updated Dashboard UI to efficiently manage project and client data, leveraging Typescript and React.
- Updated MongoDB schema to seamlessly integrate and support new features, enhancing the functionality and scalability of the platform.

Software Engineer - Index Coop

May 2021 – June 2022

The Index Coop is a web3 / blockchain company that creates and maintains the words leading crypto index products.

- Active member of the engineering, analytics, and finance pods.
- Built an application hosted on an AWS EC2 Linux instance that automated the collection, recording and calculations of monthly community contributions for the finance pod using Python, SQL, and Google Sheets. Utilized Bash scripts to deploy webserver to AWS.
- Was part of a two-man engineering group that was tasked to build an analytic tool called a subgraph, which collected and recorded all on chain data that pertained to the company's index products. This tool was built using GraphQL and TypeScript.
- Created technical documentation on Solidity, Web3.js and Hardhat.
- Contributed to the creation of the engineering on-boarding process for new developers.

## FEATURED ENGINEERING PAPERS AND PROJECTS

Links to each project's codebase is located at [www.teewhy.xyz](http://www.teewhy.xyz)

### Prometheus

As the lead backend engineer, I co-authored a paper titled "The Future of News Delivery: Integrating AI, Big Data, and Cloud Services to Revolutionize Journalism," in collaboration with graduate students from the University of Colorado Boulder and guided by a Google Advisor. Our project, Forethinker, leverages advanced AI technologies, big data, and cloud services to create a tool that synthesizes and summarizes news trends for C-level executives. I designed and implemented the backend architecture, focusing on building robust AI agents using LangChain and OpenAI's GPT-4 model for tasks such as summarization, sentiment analysis, article classification, and Named Entity Recognition (NER). The AI agents stored processed data in BigQuery and Pinecone, optimizing retrieval and similarity searches. By employing separate AI chains for specific subroutines and utilizing Pydantic for reliable LLM outputs, I ensured high efficiency and reliability. This innovative backend architecture made Forethinker a powerful tool for delivering personalized, timely, and relevant news insights to high-level decision-makers.

Technologies: Python, Big Query, Pinecone, Fast API, GCP, LangChain, GPT-4

### Text to SQL

For the NAACL Clinical NLP 2024 Shared Task, I co-authored a paper titled "Will the real SQL translator, please stand up?". Our research focused on the transformation of natural language queries into SQL commands to leverage Electronic Health Records (EHRs) efficiently. We proposed a novel approach by fine-tuning Small Language Models (SLMs) for text-to-SQL conversion within the healthcare domain. Unlike traditional methods that utilize large language models with in-context learning, our approach modifies a subset of the SLM's parameters to tailor it to specific EHR text-to-SQL retrieval tasks. By incorporating

fine-tuning and in-context learning techniques, we enhanced the model's accuracy and efficiency, making it viable for real-world applications where computational resources are limited. This research contributes to the field by providing a scalable, cost-effective solution for data querying in EHR systems, potentially transforming how healthcare professionals interact with and utilize patient data.

Technologies: PyTorch, Llama 2, Llama 3, Hugging Face, Pandas, NumPy

#### [Multigenerator, Multidomain and Multilingual Black-Box Machine-Generated Text Detection](#)

For the SemEval-2023 Task 8 on Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection, I contributed to the innovative approach developed by the Rambunctious Raccoons team from the University of Colorado Boulder. Our method utilized a transformer-based ensemble incorporating models such as RoBERTa, BERT, XLNet, and ELECTRA. By tailoring preprocessing for each model, we significantly enhanced the ensemble's performance, achieving an accuracy of 87.6% on the Test set. This experience allowed me to contribute to the cutting-edge research in natural language processing, demonstrating the effectiveness of ensemble methods in improving classification accuracy across diverse domains and languages.

Technologies: PyTorch, LLMs, Hugging Face, Pandas, NumPy

#### [Osce Medical Training Platform](#)

Developed an innovative web-based simulator, "AI Virtual Doctor's Office," specifically tailored for Physician Assistant students. This platform leverages artificial intelligence to simulate real-life clinical scenarios, allowing students to engage in structured clinical examinations across various medical specialties. Through interactive, voice-guided sessions with an AI patient, users can refine their diagnostic skills and clinical approach, enhancing their practical training and preparedness for real-world medical environments.

Technologies: Python, Typescript, FastAPI, Next.js, PostgreSQL, LangChain, GPT-4

#### [Machine Learning Recommendation System](#)

Engineered an advanced content-based movie recommendation system by leveraging diverse machine learning and natural language processing techniques. This system analyzes a vast dataset comprising over 8,000 Netflix movies and TV shows, creating a detailed feature matrix. I computed various similarity metrics to refine recommendation accuracy. Additionally, I developed and integrated two proprietary machine learning models, conducting comparative analysis with existing algorithms to enhance recommendation precision and user experience.

Technologies: Python, NumPy, Matplotlib, Pandas, Scikit-learn, NLTK

#### [CNN Image Classifier](#)

Developed a Convolutional Neural Network (CNN) to categorize various NFT images into their respective collections. The project harnessed 6,000 images from six distinct NFT series for training and evaluating multiple deep learning models. My efforts culminated in a model boasting an impressive 99% accuracy rate, significantly advancing the precision of NFT classification.

Technologies: Python, TensorFlow, Keras, Matplotlib, NumPy

#### [Medical Treatment Costs](#)

In this project, I engaged in an in-depth analysis of medical treatment costs, utilizing a dataset that included variables such as age, sex, BMI, number of children, smoking status, and geographical region. Through exploratory data analysis, I identified key factors that drive higher healthcare expenses. I developed a regression model to accurately predict individual patient costs based on these determinants, thereby enhancing my expertise in data analysis, feature extraction, and regression modeling.

Technologies: Python, Pandas, NumPy, Scikit-learn, Seaborn