

Adarsh B Shankar

Passionate about programming and solving problems in the real world

USA

Email: shankar1@uci.edu

LinkedIn:

[linkedin.com/in/adarsh-shankar-uci/](https://www.linkedin.com/in/adarsh-shankar-uci/)

Github: github.com/adarsh-hub

EXPERIENCE

Katana Graph, Austin, TX — Software Engineer I

MARCH 2022 - NOV 2022

- Analyzing performance metrics with Python scripts. Used Jenkins to improve CI/CD pipelines.
- Engineered quality automation tests from github using Bash and Python with code coverage reports.

Mu Sigma, Bangalore, KA India — Software Engineer Intern

JUNE 2021 - OCT 2021

- Improved chatbot services with RASA for all MuSigma product questions to improve customer experience and increased user retention by 15%.
- Improved Query search times on Neo4j by restructuring databases with a combination of Python and R scripts .

MoneyTap, Bangalore, KA India — Data Analyst Intern

SEPT 2020 - MARCH 2021

- Developed scripts in Python to implement a name and address matching feature on the platform to improve the customer verification process.
- Using FastAPI, integrated the above feature into the backend for all different platforms of the product to use. Increased customer verification process speed by 5%

EDUCATION

University of California, Irvine — B.S Computer Science DEC 2021

Relevant Coursework:

- Algorithms → Data Structures → Optimization Intro
- Machine Learning → Project in AI → Graphical Models

SKILLS

Strong: Python, Java, C++, Bash, Git

Intermediate: R, MySQL, Julia, Jenkins, GCloud

Familiar: HTML, CSS, C, ReactJS, FastAPI, AWS, PyTorch

PROJECTS

Checkers AI: Implemented Monte-Carlo-Tree-Search in an agent to play Checkers.

Minecraft Agent: Used reinforcement learning to create an agent that farms different resources based on user input through the chat.

Crux Compiler: amet Consectetur adipiscing elit, Sed diam nonummy

ML Classification: Tested different types of classifier models for Kaggle challenge to predict rainfall at a particular location. Final prediction score: 0.79

Graph Models: Used C++ to implement different types of random graph generators and analyze different algorithm performances.