Yisak Tolla

US Citizen | Sterling, VA | 571-839-XXXX| ytolla@gmu.edu | www.linkedin.com/in/yisaktolla | github.com/YisakTolla

EDUCATION

George Mason University Fairfax, VA

Bachelor of Science, Computer Science *Expected December 2025*

- GPA: **3.82/4.0** — *Dean's List x 8 times, Tech Talent, The Hanover Insurance Group Foundation's Emp. Scholarship*
- Coursework: Software Engineering, Databases, Networking, Operating Systems, Artificial Intelligence, Data Mining, Computer Systems and Programming, Data Structures & Algorithms

TECHNICAL SKILLS

Programming Languages: Python, Java, C, JavaScript, SQL, Rust, HTML/CSS

Frameworks: React, Node.js, Express.js, JUnit, JDBC, Flask

Developer Tools: Git, Linux, MongoDB, SQLDeveloper, SQLLite, Kaggle, Vercel, Render

Certifications: AWS Developers Associate(Ongoing), IBM Machine Learning (2024), AT&T Technology Academy (2025)

EXPERIENCE

Purgeon *Fairfax, VA*

Software Engineer *October 2024 – May 2025*

- Contributed to end-to-end project development of an AI-powered disk optimization tool, integrating Rust, Python, and React to boost performance and reclaim up to 20GB of space.
- Followed the software development life cycle (SDLC) to manage feature releases and maintain agile workflows.
- Conducted code reviews with a team of 6 engineers to maintain high standards of code quality and consistency.
- Applied performance optimization techniques to improve file scanning and categorization speed across 1k+ entries.

George Mason University - College of Computing *Fairfax, VA*

Undergraduate Teaching Assistant - Java, C, & Databases *January 2024 - Present*

- Mentored 300+ students in object-oriented programming, low-level programming, process execution, and database topics resulting in over 85% pass rate.
- Led sessions including unit testing and debugging strategies, improving assignment success rates.
- Facilitated software testing labs and explained web development fundamentals through student projects.

PROJECTS

DermoAI | Python, TypeScript, JavaScript, React, Express.js, Flask, Vercel, Render

- Developed AI-powered skin disease classification system using React 19, TypeScript, and Express.js backend, achieving 85.5% confidence accuracy across 24 different skin conditions through deep learning integration
- Built end-to-end machine learning pipeline with PyTorch ResNet152/DenseNet121 models and Flask API, processing real-time image analysis to deliver instant predictions with severity assessments and treatment recommendations
- Implemented professional healthcare UI/UX with responsive React components and Google Maps API integration, providing seamless dermatologist finder feature and modern interface for medical professionals
- Engineered automated ML model service with Python 3.8+, PyTorch framework, and 24-class classification system, processing base64 image uploads and returning JSON-formatted medical analysis with confidence scoring

Heart Disease Predictor | Python, JavaScript, HTML/CSS, Flask

- Delivered a clean and consistent dataset for heart risk prediction by preprocessing a public health dataset from Kaggle with 1,319 entries, resolving 100% of missing values, and correcting over 50 outliers and inconsistencies.
- Achieved 92% accuracy by implementing and evaluating 5 supervised machine learning algorithms (e.g., logistic regression, random forest, KNN) in Python, identifying key health patterns and risk factors.
- Enhanced usability by designing and developing a responsive web interface using HTML, CSS, and JavaScript, enabling real-time input of 10+ health metrics for over 100 test users during evaluation.

Snake Game | Rust, Kernel

- Built a fully functional Snake game in Rust by leveraging opengl_graphics, Glutin_Window, and Piston crates to create a responsive GUI and real-time event handling, resulting in smooth frame updates and intuitive controls.
- Optimized game performance by implementing a 125ms update interval, enabling consistent frame timing and reducing input lag by over 30% during testing.
- Designed a rendering pipeline with O(n) efficiency for visualizing snake segments and maintained O(1) operations for core game logic (e.g., movement and collision detection), supporting stable performance at runtime.