Usman Khan

usmankhan.dev | usman@usmankhan.dev | **US Citizen** | linkedin.com/in/khanu | github.com/ukhan1219

Education

University of Central Florida — Orlando, Florida
B.S. in Computer Science 3.8/4.0 GPA — *Expected Graduation: December 2025*

Relevant Coursework: Algorithms in Machine Learning, Artificial Intelligence/Machine Learning, Robot Vision, Computer Vision

Technical Skills

Languages: Python, C++, SwiftUI, Go, TypeScript, JavaScript, C, Java, SQL, NoSQL, R
Frameworks: PyTorch, Keras, TensorFlow, NumPy, Pandas, SKLearn, Next.js, Node.js, Express.js, React, Tailwind
Tools: Git, Github, Docker, Linux, LaTeX, Prisma, Neo4J, Figma, Amazon Web Services, Google Cloud Platform

Work Experience

Software Engineering Intern — Aug 2024 – Jun 2025
Vcom3D *— Python, TensorFlow, OpenCV, Raspberry Pi 5, Meta Quest 3, BioGears (UW), C++, XML* — *Orlando, Florida*

- Built pose tracking models using TensorFlow on Raspberry Pi, boosting accuracy & reducing latency by **30%**
- Merged BioGears (University of Washington) for injury simulation, boosting training realism by **98%** across modules
- Created AR/VR apps on Meta Quest to support simulations ran by BioGears in a distributed system architecture
- Refined system integration across multiple components via cross-functional collaboration, slashing errors & streamlining updates

Machine Learning/AI Undergraduate Research Assistant — Apr 2024 – Apr 2025
University of Central Florida *— Python, TensorFlow, Neo4J, NumPy, SKLearn, NetworkX, Pandas* — *Orlando, Florida*

- Enforced automated distributed data mining algorithms using AI/ML via Neo4J for enhanced predictive analytics
- Generated data mining methods using RandomForestRegressor on a DARPA dataset (**6.8M+ nodes**) to detect illicit activity
- Devised scalable distributed data pipelines boosting entity tracking accuracy and speed by **30%** across datasets
- Deployed statistical methods for performance optimization, reducing processing time by **40%** for high-volume pipelines

Projects

PyChess | *Python, PyTorch, Hugging Face Transformers/TRL, Accelerate, python-chess, Stockfish, TensorBoard, DistilGPT-2*

- Engineered an end-to-end chess AI post-training pipeline that automates data generation and training with robust tracking
- Processed 90M positions; curated 1M supervised samples and 500k preference pairs using strict quality filters
- Reduced data generation time by **97%** via multithreading and optimized I/O; eliminated memory-related training failures

Mantle | *SwiftUI, Python, PyTorch, Core ML, Transformers, Hugging Face, Metal (MPS), Amazon Web Services EC2*

- Converted Transformer models (Mistral, Llama) from PyTorch to Core ML utilizing AWS EC2 instances
- Applied Core ML compression (quantization, pruning, palettization) shrinking models by **75%** while retaining accuracy
- Accelerated inference **25%** leveraging Metal Performance Shaders (MPS) optimization on for On-Device inference
- Developed privacy-first SwiftUI app (**iOS 18+**) for On-Device ML inference, enabling offline AI chatbot functionality

Glance | *SwiftUI, Go, Firestore, Firebase Auth, Plaid API, Google Cloud Platform, Figma, XCTest*

- Architected a budgeting app using SwiftUI and a Go backend, achieving seamless Plaid API integration
- Implemented secure authentication via Firebase Auth & managed sessions, supporting **100+** concurrent users reliably
- Designed responsive UI/UX flows in Figma & built with SwiftUI, boosting user engagement metrics and retention
- Enhanced data retrieval speeds by **40%** through strategic caching & optimized Firestore queries in the Go backend

DUI | *Rust, Crates.io (Cargo), Homebrew, clap, rustyline, crossterm, tui-rs, serde*

- Built a Docker CLI with **100%** command parity, interactive mode, and real-time charts/dashboard
- Published on Cargo as dui-cli and Homebrew tap; reached **150+** downloads
- Enhanced UX with tab completion, contextual help, smart suggestions; optimized release builds

Fit | *MERN: MongoDB, Express.js, React, Node.js, TypeScript, AWS Lightsail, Figma*

- Led Agile software development lifecycle of workout tracking app; deployed scalable application on Amazon Web Services
- Evolved distributed storage solutions using MongoDB with optimized query interfaces, cutting CRUD times by **30%**
- Unified Express.js/Node.js backend with a React frontend, resulting in a **40%** improvement in API response speed