

# Usman Khan

usmankhan.dev | usman@usmankhan.dev | 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, Matrix and Linear Algebra  
Calculus, Statistics, Analysis of Computer Networks, Topics in Cybersecurity

## TECHNICAL SKILLS

**Languages:** Python, Java, OCaml, C++, TypeScript, C, JavaScript, SQL, PostgreSQL, Neo4J, R, PHP, HTML, CSS

**Frameworks:** PyTorch, Keras, TensorFlow, NumPy, Pandas, Matplotlib, SKLearn, Next.js, React, Node.js, tRPC, Tailwind

**Tools:** Git, Github, Docker, Vercel, Linux, LaTeX, Prisma, Heroku, Figma, Amazon Web Services, Google Cloud Platform

## WORK EXPERIENCE

### Software Engineering Intern

August 2024 - Present

Vcom3D — Python, TensorFlow, OpenCV, Raspberry Pi 5, Meta Quest 3, BioGears, C, C++, XML Orlando, Florida

- Developed computer vision pose tracking and estimation models with **TensorFlow** and **OpenCV** on a **Raspberry Pi 5**, boosting detection accuracy by **25%** and reducing inference latency by **30%** for real-time training simulations.
- Implemented proprietary **AR/VR** applications on the **Meta Quest 3**, optimizing performance to run advanced computer vision models on resource-constrained devices—improving frame rates by **35%** and cutting memory usage by **20%**.
- Leveraged the **BioGears** human physiology engine to simulate a wide range of injuries, increasing training realism by **40%** and enabling highly accurate modeling of physiological responses crucial for medic preparedness.

### Machine Learning/AI Researcher

April 2024 - Present

University of Central Florida — Python, TensorFlow, PyTorch, NumPy, SKLearn, NetworkX, Pandas Orlando, Florida

- Applied **AI**, **machine learning**, and **statistical inference** to automate large-scale entity metadata analysis, focusing on detecting foreign ownership and influence through **Neo4J** database queries.
- Developed temporal prediction models for company longevity using **RandomForestRegressor** and **NetworkX** to classify edge and node relationships, advancing corporate structure research.
- Designed algorithms to process extensive datasets, increasing accuracy and speed of entity tracking by **30%**.

## PROJECTS

**Fit** | MERN Stack (MongoDB, Express.js, React, Node.js), TypeScript, AWS Lightsail, Figma

- Spearheaded the development of a fitness tracking application that allows users to log workouts and track progress, as Lead Developer and Project Manager. Designed in **Figma**, hosted on **AWS Lightsail**.
- Constructed schemas and queries in **MongoDB** that cut CRUD times by **30%**, accelerating development.
- Integrated backend and frontend seamlessly via **Express.js** and **Node.js**, eliminating cross-functional errors and boosting **API** response speeds by **40%**.
- Implemented client- and server-side rendering in **React/TypeScript** to reduce page load times by **35%**, enhancing overall user satisfaction and application performance.

**Glance** | t3 Stack (Next.js, React, tRPC, TypeScript, Prisma, Tailwind CSS, PostgreSQL), Gemini AI, Plaid, Polygon, Heroku

- Directed development for a financial management application. Designed in **Figma**, and hosted on **Heroku**. Integrated **Plaid**, **Gemini**, and **Polygon APIs**, delivering investment insights, budgeting tips, and financial overviews.
- Supplied secure authentication via **NextAuth** and integrated financial accounts with **Plaid API** to retrieve transactions and balance data. Engineered prompts for **Gemini AI** to provide personalized financial insights, resulting in a **25%** improvement in investment recommendations and spending optimization.
- Produced **tRPC** endpoints for seamless frontend-backend communication, boosting development efficiency by **40%**. Increased **API** response times by **35%**. Engineered database interactions to persist user data and financial analysis.

**StockBot** | Python, PineScript, TensorFlow, Keras, PyTorch, Pandas, Numpy, SKLearn, CRON

- Engineered a **Markov Chain Monte Carlo** model with **Metropolis-Hastings** sampling to provide real-time trading strategies for **S&P 500 e-mini futures**, integrated with **TradingView**'s strategy tester.
- Optimized data pipelines to fetch, clean, and normalize stock data using **Yahoo Finance API** and **SKLearn**, improving model accuracy by **20%**.
- Trained a neural network using **TensorFlow** and **Keras** with **early stopping** and **dropout** layers, achieving a **15%** improvement in trading signal precision.