Pranit Reddy

925-549-2300 | preddy4@wisc.edu | linkedin.com/in/pranit | github.com/jollymancan

Education

University of Wisconsin-Madison Madison, WI
B.S. Computer Science, B.S. Mathematics, Minor in Statistics *Sept. 2024 – May 2027*

Relevant Coursework: Algorithms, Data Structures, Database Management Systems, Matrix Methods, Game Theory & Learning, Optimization, Probability Theory, Linear Algebra

Experience

Undergraduate Summer Software Intern June 2025 – August 2025
Essert Inc. *Santa Clara, CA*

- Designed and deployed a system to semantically map security controls across multiple compliance frameworks (SOC 2, NIST 800, ISO) using Sentence Transformers embeddings and cosine similarity scoring
- Built a matching engine that automatically linked customer evidence documents to the appropriate security controls, eliminating manual effort in compliance tracking
- Delivered Angular UI modules for customers to review, approve, or reject automated control and evidence mappings

Summer Intern June 2024 – August 2024
Create Circles (nonprofit) *Remote*

- Implemented Twilio SMS API workflows to support automated notifications and two-way messaging with skilled-nursing facilities (SNF)
- Assisted with general logistics and administrative tasks, including organizing volunteers and communicating with SNFs

Projects

Chess Engine | *C/C++, Git*

- Represented chess positions using 64-bit bitboards (one bit per square) to enable fast, low-level bitwise operations for move generation
- Implemented and validated a chess engine move generator using perft tests up to depth 7 (3.2B positions), achieving close to 160M nodes/second using multithreading while matching standard perft reference counts
- Used magic bitboards (precomputed hash-based attack tables) to compute sliding piece moves (rook/bishop/queen) in effectively constant time
- Implemented the engine's search with a negamax-style minimax algorithm using alpha-beta pruning and move ordering to efficiently find the best move

Interactive Gridworld RL Sandbox | *Python, NumPy, Streamlit, Git*

- Built an interactive web-based RL sandbox where users draw custom gridworld environments and watch a Q-learning (model-free) agent learn in real time
- Implemented tabular Q-learning from scratch with epsilon-greedy exploration, adjustable hyperparameters (alpha, upsilon, epsilon), and live visualization of value-function heatmaps and learned policies.
- Designed an intuitive UI with Streamlit so users can edit maps, reset training, and compare learning behavior across different environments and settings
- Logged and plotted training dynamics to demonstrate key RL concepts such as exploration vs exploitation, sparse rewards, and the impact of environment design on sample efficiency

Technical Skills

Languages: Java, Python, C/C++, SQL, JavaScript, HTML/CSS, R
Frameworks: React, Streamlit, Node.js, Angular, Flask, JUnit, WordPress, Material-UI, FastAPI
Developer Tools: Git, Google Cloud Platform, Twilio, VS Code, Visual Studio, PyCharm, IntelliJ, Eclipse
Libraries: TensorFlow, PyTorch, Hugging Face, Keras, Scikit-learn, OpenCV, pandas, NumPy, Matplotlib