

Sean Erfan

seanerfan@gmail.com | +1 (516)-467-9432 | seanerfan.us | GitHub: Shameed4 | LinkedIn: seanerfan

Education

Stony Brook University

May 2025

B.S. in Computer Science and Applied Mathematics & Statistics

GPA: 3.96 (Summa Cum Laude)

Awards: 3x Hackathon Winner (HackHarvard, SBUHacks, HopperHacks), 6x Dean's List

Technical Skills

Languages: Go, TypeScript, Python, C#, C, Java, SQL, Bash

Backend & Cloud: NestJS, Node.js, ASP.NET Core, PostgreSQL/PostGIS, Prisma, Redis, WebSockets, AWS (ECS, S3, CloudFront), Terraform, Docker, Jenkins

Frontend & Mobile: React, Next.js, Angular, Tailwind CSS, React Native/Expo, Android (Java)

Experience

Software Engineer

Babylon, NY

Lotus MD

Mar. 2026 – Present

- Sole engineer rebuilding a pre-release healthcare product off legacy ColdFusion/AngularJS onto **NestJS** and **Angular** over Postgres/**PostGIS**, cutting query latency ~60% through added indices and N+1 elimination.
- Drove one **OpenAPI** spec from NestJS Swagger through **Orval** to generate typed clients for the Angular web app and a **React Native/Expo** mobile app, so contracts cannot drift between them.
- Closed a **WebSocket** auth gap where a socket authenticated at connect time outlived its own JWT: tokens are verified on the **Socket.IO** handshake against a **JWKS** endpoint, and each connection arms a disconnect timer at token expiry so clients reconnect with a fresh token instead of holding a stale session.
- Made storage writes fail safe by committing uploads to object storage before the database rows and deleting the orphaned object if the DB write fails, so an interrupted upload cannot leave a dangling file.

Software Engineer

Bohemia, NY

North Atlantic Industries

May 2025 – Mar. 2026

- Built a mission-critical **Android** warehouse app (Java, MVVM, Hilt) with offline-first sync, queuing failed API requests locally to guarantee zero data loss during network partitions.
- Designed concurrency control in **ASP.NET Core** using **Serializable** transaction isolation to eliminate race conditions during concurrent warehouse picking, with collision detection that forks active sessions to preserve user state.
- Built an **Angular** production planning dashboard aggregating real-time BOM shortages and batch schedules, winning honorable mention at the company hackathon and saving the planning team ~6 hours weekly.

President and Co-Founder

ai-community-sbu.github.io

Artificial Intelligence Community

Nov. 2022 – May 2025

- Founded and scaled a university AI club to **750+ members**, leading a 10-person team to organize hackathons, a Kaggle computer vision competition, and weekly workshops on PyTorch, LLMs, and data analysis.

Projects

Kittens (Distributed Multiplayer Card Game)

github.com/Shameed4/Discord-Kittens

Mar. 2026 – Present

- Built a real-time multiplayer server in **Go** with a channel-driven event loop that serializes all state mutations without mutexes, driving a 9-state turn FSM with strict transition guards.
- Scaled the server to N nodes by leasing each lobby to exactly one owner in **Redis**, using monotonic epochs as **fencing tokens** against zombie owners and proxying **WebSockets** from non-owner nodes to the owner.
- Made lease acquisition return the dead owner's state snapshot so adopting an in-flight game is atomic, with client-assigned sequence numbers making moves **exactly-once** across reconnects. Killing a node mid-game in a 3-node, 60-bot chaos run left **16/16 games surviving** at 4.5ms p99 action latency.

Nexus (Research Discovery Platform)

nexus.sbu.online

Aug. 2024 – May 2025

- Built a full-stack platform (React → Next.js/Prisma/Tailwind) centralizing 130+ faculty research opportunities, reaching **300+ active users** across 6 departments within two weeks of launch, with **GPT-4o** auto-summarizing dense academic abstracts.

USG Chatbot (Agentic RAG over Governance Documents)

github.com/Shameed4/USG_Chatbot

Oct. 2025 – Dec. 2025

- Built a research assistant over student-government bylaws and minutes, giving **GPT-4o** four retrieval tools (**FAISS** vector search, regex, recursive file search, listing) in a bounded tool-calling loop, with **SHA-256** change detection re-embedding only modified documents.