Nihar K T

niharkt336@gmail.com — +91 88678 00257 — GitHub: NiharKT12 — LinkedIn: Nihar K T

About Me

Information Science undergraduate with strong foundations in C++, algorithms, and systems programming. Interested in building performance-oriented software, including game engines, emulators, and backend systems.

Education

Bachelor of Technology — Information Science and Engineering
NMAM Institute of Technology, Nitte — CGPA: 8.56 — Expected 2027

Pre-University College (PUC)
Udupi Jnana Sudha PU College — 93.16%

Internship Experience

Research Intern — Speech-to-Text Transcript System
Manipal Institute of Technology (MIT), 2025

Benchmarked open-source Automatic Speech Recognition (ASR) systems including OpenAI Whisper and SpeechBrain across diverse multi-speaker and noisy audio datasets.

- Developed Python-based evaluation pipelines for preprocessing, transcription, and metric computation.
- Integrated DeepSeek API for enhanced transcription workflows and comparative performance analysis.
- Evaluated model performance using Word Error Rate (WER), Character Error Rate (CER), Accuracy, and F1-Score.
- Conducted systematic experimentation and hyperparameter tuning to analyze model robustness.

Gained hands-on experience in NLP, deep learning-based speech recognition, API integration, and research-driven model evaluation using Python.

Projects

1. **Chess Engine (C++)**
   - Engineered a chess engine utilizing 64-bit bitboards and Zobrist hashing, optimizing memory footprint and reducing redundant evaluations.
   - Implemented Negamax search with Alpha–Beta pruning, quiescence search, and 3+ advanced heuristics to improve AI decision-making depth.
   - Generated 100% compliant pseudo-legal moves for 6 piece types and built an interactive Raylib GUI for real-time game state visualization.
2. **AI Code Review Agent (Python, tree-sitter, Groq API)**
   - Built an agentic pipeline that clones any GitHub repo, extracts every function via `ast` (Python) or tree-sitter (7 other languages), and exposes the inventory to an LLM agent as three lookup tools rather than pasting source into the prompt.
   - Designed a verification layer that checks every reported bug's cited evidence against the real source, cutting false positives from 8 issues to 2 on a live run; enforced `temperature=0` after measuring run-to-run variance (0–8 issues at temp 0.2 across 5 runs on identical code).
   - Engineered around Groq's free-tier rate limits with server-directed retry timing and hard per-request token budgets; validated with 97 offline unit tests including a scripted-client harness for the agent loop.
3. **Pixel Link — URL Shortener (Node.js, Express, MongoDB, Redis)**
   - Developed a full-stack URL shortener providing RESTful APIs using Node.js and MongoDB, utilizing Base62 encoding for collision-free links.
   - Incorporated a Redis Cloud caching layer with a 1-hour TTL, accelerating redirection latency and reducing overall database load.
   - Deployed the pixel-art themed frontend on Vercel and backend on Render, featuring a real-time click analytics dashboard.

Technical Skills

Languages: C, C++, Java, Python, JavaScript
Frontend: HTML, CSS, raylib (GUI)
Backend: Node.js, Express.js
Databases: MySQL, MongoDB, Redis
Tools / Build: Git, GitHub, VS Code, CMake

Certificates

Google AI Essentials — Coursera
Frontend Development — Scrimba