

SUMMARY

Software engineer with backend experience building RESTful APIs, optimizing databases, and improving application performance. Hands-on experience developing AI projects involving LLMs, retrieval-augmented generation, voice interfaces, and agent workflows. Skilled in Python, TypeScript, Node.js, and SQL, with a focus on reliable systems and practical AI integration.

EDUCATION

UNIVERSITY OF TEXAS AT ARLINGTON | ARLINGTON, TX

Aug 2022-May 2024

Master of Science, Computer science

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY | Hyderabad, India

Jun 2018-May 2022

Bachelor of Technology, Electronics and Communications Engineering

TECHNICAL SKILLS

Frameworks & Libraries: TensorFlow, Keras, PyTorch, Sklearn, Pandas, NumPy, D3.js, Spring Boot, React.js, Angular, Node.js, Hadoop, Spark, NLTK, spaCy, Gensim, Transformers, SQLite.

IDE & Development tools: VS Code, Jupyter Notebook, Google Colab, Tableau, RStudio, KNIME, Unix, Git, Docker, Jenkins, AWS CodePipeline.

Programming Languages: Python, Java, SQL, JavaScript, TypeScript, C, C++, HTML, Scala.

PROFESSIONAL EXPERIENCE

Software Engineer | KeelWorks

Dallas, Texas | Jul 2024 – Jul 2025

- Developed 15+ scalable RESTful APIs using Node.js, Express.js, TypeScript, MySQL, and Sequelize ORM, enabling secure and efficient communication across multiple application modules.
 - Designed and optimized 20+ relational database schemas and SQL queries, improving API response times by 30% through indexing, query tuning, and efficient data modeling.
 - Built reusable middleware for authentication, authorization, request validation, logging, and centralized error handling, reducing duplicate code by 35% and improving maintainability.
 - Enhanced backend performance by implementing asynchronous processing, caching strategies, and optimized database interactions, increasing overall application throughput by 25%.
 - Conducted 50+ code reviews and mentored junior developers on backend architecture, REST API design, and coding best practices, improving overall code quality and consistency.
 - Collaborated with 8+ cross-functional developers in an Agile environment to deliver new features, resolve integration issues, and support bi-weekly production releases.
 - Streamlined development and deployment workflows using Git, GitHub, Docker, and CI/CD pipelines, reducing deployment time by 40% and ensuring reliable application releases.
-

Projects:

Job Application Automation Pipeline | OpenAI Response API's, Microsoft Playwright, Gmail MCP, OAuth 2.0, Node.js, TypeScript [GitHub](#)

- Built a pipeline that ingests and deduplicates ATS and job-board postings, ranks them against candidate profiles using semantic search, and generates job-tailored LaTeX resumes compiled to PDF.
- Developed a Playwright-driven form runner with DOM/ARIA extraction, reusable widget handlers, profile-grounded answers, and committed-value verification across multiple job portals.
- Orchestrated multi-step applications through Responses API tool calls and previous_response_id, with bounded model assistance for unfamiliar controls and approval gates for ambiguous answers and submission.
- Maintained an application ledger and integrated Gmail through MCP and OAuth 2.0 to link confirmations, assessments, interviews, and rejections to their applications.

Visa Advisor | Python, FastAPI, Claude, Tavily, Supabase(PostgreSQL), DeepEval, Langfuse, Docker

[Live Website](#)

- Built an async FastAPI service that parses traveler profiles with Claude Haiku, searches visa and re-entry rules in parallel through Tavily, and validates structured guidance with Pydantic; reduced the search phase from over 30 seconds to under 5 seconds.
- Designed a two-stage LLM pipeline using Claude Haiku for analysis and optional Sonnet review, with SSE streaming and a versioned Supabase cache that returns repeat results without rerunning the analysis.
- Developed pytest and DeepEval evaluations for accuracy, completeness, clarity, and consistency; used Langfuse to trace search results and model responses, and deployed the Dockerized service to Fly.io through GitHub Actions.

Conversational AI Agent | Swift, TypeScript, OpenAI Responses API, Apple Vision, NLEmbedding, LangGraph, Playwright

[GitHub](#)

- Engineered a Swift voice client with streamed speech capture and ElevenLabs synthesis, coordinating voice activity detection, barge-in, and playback cancellation for natural conversation.
 - Built a TypeScript agent runtime around Responses API function/tool calls, with validated tool schemas, call-ID deduplication, previous_response_id chaining, and LangGraph checkpoints that suspend and resume tasks for human approval.
 - Designed a self-contained, access-controlled RAG pipeline using Apple Vision OCR, document chunking, and Apple NLEmbedding sentence vectors; retrieved relevant and adjacent passages with file/page citations, and evaluated retrieval relevance through LangSmith datasets and traces.
 - Connected Gmail search, thread reading, reply drafting, and mail monitoring through Gmail MCP tools, alongside Playwright browser research and scheduling; required executor-verified previews and approval before sending messages or committing external actions.
-

PUBLICATIONS AND PATENTS:

RESEARCH PAPER (2021): "Automatic Speed Control System in Speed Zone Areas using OpenCV" - International Conference on Electronic Circuits and Signaling Technologies. <https://iopscience.iop.org/article/10.1088/1742-6596/2325/1/012017>