

Abhirami Thiagarajan

abhi.thiagaraja@utexas.edu • 972-859-9048 • abhithiagarajan.dev • github.com/abhithiagarajan • linkedin.com/in/AbhiThiagarajan

EDUCATION

The University of Texas at Austin

Bachelor of Science, Computer Science, Minor in Math | GPA: 3.68

May 2029

Austin, TX

Relevant Coursework: Data Structures & Algorithms, Discrete Mathematics, Linear Algebra & Matrix Computations, Differential Equations, Probability & Statistics, Introduction to Computing

EXPERIENCE

AllWay

May 2026 – August 2026

Software Engineering Intern

Dallas, TX

- Built agentic AI RAG pipeline in Python integrating PostgreSQL, HuggingFace Transformers, and Anthropic Claude API, exposing functionality via RESTful API endpoints to resolve natural language queries across 5 enterprise system classes with full source citation
- Designed query pipeline architecture: NL query → intent classification → parallel retrieval (SQL / vector search / knowledge graph) → LLM synthesis → cited response, following agile sprint cycles
- Architected three-layer retrieval system (structured SQL querying, vector similarity search, domain knowledge graph), enabling cross-system operational intelligence across PLM, ERP, MES, QMS, and LIMS platforms
- Wrote automated unit and regression tests integrated into CI/CD pipeline, catching retrieval accuracy regressions pre-deploy
- Optimized multi-step LLM pipeline using GPU acceleration, asyncio-based concurrency, and embedding caching, cutting end-to-end query latency from 120s to under 30s

University of Texas

August 2025 – May 2026

Student Researcher

Austin, TX

- Designed and shipped modular FastAPI microservice architecture handling multi-modal medical imaging requests, with middleware-based validation, async request queuing, and pluggable preprocessing supporting 3 scan types
- Implemented distributed processing pipeline: ingestion → validation → preprocessing queue → worker pool → batched inference → result aggregation, profiling and resolving concurrency and thread contention bottlenecks to increase throughput 40% across concurrent queues
- Built automated CLI tooling and test scripts to replace manual pipeline execution, improving reliability and reducing regression risk across distributed compute environments

Tiger Analytics

May 2025 – August 2025

Software Engineering Intern

Dallas, TX

- Engineered scalable Python/SQL ETL pipelines (raw ingestion → schema validation → transformation/cleaning → incremental load → analytics-ready tables), ingesting 50K+ daily records and cutting data latency to near real-time
- Built and deployed production Flask REST API with versioned prediction endpoints, input validation, and automated regression/integration testing, serving live traffic
- Developed self-serve internal dashboard platform, improving cross-team efficiency and eliminating 15+ hours/week of manual analyst requests

Wolfram

May 2024 – August 2024

Research Intern

Boston, MA

- Engineered distributed, scalable backend across 250K+ nodes, refactoring sequential jobs into parallelized, multi-threaded pipelines, cutting runtime 60%
- Built parallelized job pipeline: task queue → job partitioning → distributed worker dispatch → node-level execution → result reduction
- Debugged and resolved memory management and thread contention issues in GPU-accelerated molecular simulation pipelines, reducing peak memory usage 35% on high-node workloads

PROJECTS

CardioPulse | Python, FastAPI, Plotly/Dash, AWS Lambda, API Gateway

2026

- Architected and deployed full-stack cardiovascular diagnostics platform with FastAPI backend serving real-time ECG signal analysis and cardiac risk prediction models via REST API
- Built serverless inference architecture on AWS Lambda + API Gateway, achieving sub-second ECG response latency for real-time clinical feedback
- Validated with 5 clinical users, iterating on model accuracy and UI based on direct feedback

CarbonAI | Python, Dash, AWS Lambda, LLM APIs, Type Script

2026

- Architected and deployed full-stack sustainability platform with serverless Python backend and Dash frontend delivering real-time AI-generated carbon reduction recommendations from user-submitted emissions data
- Designed stateless microservice APIs for LLM inference, carbon scoring, and recommendation generation, supporting concurrent multi-user sessions

ACCOMPLISHMENTS

WWT Hackathon - 1st Place, built CNN-based diagnostic imaging classifier

2026

EnviHacks – Sustainability Innovation Award + Girls Recode Hacks (2nd Place)

2026

Wolfram Emerging Leaders/Scholarship Award – Top 10% of research papers

2025

Siemens Tech for Sustainability Hackathon - 3rd Place

2025

TECHNICAL SKILLS

Languages: Python, Java, TypeScript, JavaScript, SQL, Bash, HTML/CSS

Frameworks: FastAPI, React, Next.js, Node.js, Express.js, PyTorch, Tailwind CSS

Databases & Infrastructure: PostgreSQL, Redis, Supabase, ChromaDB, Docker, AWS, Celery

Libraries & Testing: SQLAlchemy, Pydantic, pandas, NumPy, sentence-transformers, pytest, SWR

Developer Tools: Git, GitHub Actions, Linux/Unix, REST APIs, Alembic, OAuth