ADITI SINGLA

Delhi, India | +91 98998 59193 | aditisingla.as11@gmail.com | aditi21372@iiitd.ac.in
LinkedIn | GitHub | Publication (ICLR 2024 Tiny Paper)
Integrated Degree (B.Tech + M.Tech), Computer Science and Engineering, Indraprastha Institute of Information Technology, Delhi

PROFESSIONAL SUMMARY

Computer Science graduate (Integrated B.Tech + M.Tech) currently working as a Research Engineer at TCS Research, building Linage, an LLM-assisted legacy-code modernization framework that verifies every LLM-proposed rule against executed tests before it reaches production code. Thesis built a 331-task LLM coding-agent benchmark (**129,770+ trajectories, 11 models**) with a from-scratch ReAct agent, a hybrid BM25 + dense-embedding retrieval pipeline, and a process reward model trained with LoRA and ranking-calibration losses. Combines this with hands-on experience building LLM-powered agentic systems, RAG pipelines, and backend services: multi-agent orchestration (**CrewAI, LangChain**) with planner/orchestrator and specialist agents, prompt-optimization techniques (DSPy, GEPA), and integration of OpenAI, NVIDIA, and Ollama APIs. Backed by production backend engineering using Python/Django and REST APIs, a strong foundation in data structures and algorithms, and published research (ICLR 2024 Tiny Paper) on cost-quality optimization for language model ensembles. Comfortable working across the full AI development lifecycle, from agent design and evaluation methodology to backend integration, in fast-paced, evolving environments.

TECHNICAL SKILLS

Programming Languages: Python, C++, Java, C, R, Kotlin

Agentic AI & LLM Tooling: LangChain, CrewAI, DSPy, GEPA-based prompt optimization, ReAct agent design (built from scratch), multi-agent orchestration (planner/orchestrator + specialist agents), Retrieval-Augmented Generation (RAG), FAISS (vector search/embeddings), hybrid retrieval (BM25 + dense embeddings + RRF fusion), OpenAI API, NVIDIA API, Ollama, Vision-Language Models (VLMs), process reward models (PRM), LoRA fine-tuning, ListNet ranking loss, prompt engineering & prompt design

AI / ML & Model Training: Small Language Model (SLM) development, CodeT5 pre-training & fine-tuning, PyTorch, DeepSpeed (distributed/multi-node GPU training), Transformers, model benchmarking & profiling, algorithmic/memory optimization, NLP fundamentals, tokenization strategies

Backend & APIs: Django, RESTful APIs, ORM, Retrofit, PostgreSQL, MySQL, SQLite, Flask

Software Engineering: Data Structures & Algorithms, Git/GitHub, automated testing pipelines, Linux/UNIX, Shell Scripting, AWS, Docker

Front-End & Tools: React, HTML, CSS, Figma, Android Studio, Jetpack Compose, LaTeX, NumPy

WORK EXPERIENCE

Research Engineer, TCS ResearchJul 2026 – Present
Python, LLM Orchestration (NVIDIA NIM/Gemini/OpenAI/Ollama), AST Parsing, OMG KDM/XMI, JVM Test Harnesses, SQLite, Flask

- Built Linage, a semantic migration pipeline that parses legacy COBOL/C/Java into OMG KDM models and mines provenance-grounded evidence (file + line citations) for every business-rule candidate.
- Designed a "propose-then-verify" trust boundary: LLM-proposed contracts are validated by auto-generated tests executed against the real legacy system, raising rule precision from **71% (raw LLM) to 97%**.
- Implemented differential testing (legacy vs. generated target, byte-identical outputs) achieving 2,712/2,712 passing cases at ~10K LOC scale, with parsing of a **56K-LOC codebase** in under 2s (warm cache).
- Engineered **SQLite-backed semantic memory** for incremental re-migration, reusing 1,252 verified contracts on code changes and running ~**5.8× faster** than full re-migration.
- Added polyglot rule reconciliation across COBOL/Java twins, a human-in-the-loop review queue with audit trail, and prompt-injection/schema-validation hardening against LLM hallucination.
- Built a Flask web dashboard to launch migrations, browse extracted contracts, and drive human review.

Backend Developer, Graduation Checklist PortalJan 2025 – Jun 2025
Python (Django) · Guide: Dr. Debajyoti Bera

- Built and maintained the backend of a **Django-based graduation checklist portal**, coding graduation requirement logic into a configurable rules engine that could be updated as academic policy evolved.
- Cut graduation-eligibility processing time from three months to one week by replacing manual transcript checks with an automated, ORM-driven data pipeline for retrieving and validating student grades and course records.
- **Reduced the graduation verification workload from a 3–4 person manual process to a single-person,** system-assisted workflow, improving both speed and accuracy of eligibility assessment.
- Integrated automated testing pipelines into the development workflow to catch regressions early and keep the rules engine robust as requirements changed.

AGENTIC AI, LLM & SOFTWARE PROJECTS

M.Tech Thesis: T3-CPP Bench: Protocol-Sensitive Evaluation of LLM Coding AgentsMay 2025 – May 2026
C++, Python, Docker, ReAct Agents, Hybrid RAG (BM25 + Dense), Process Reward Modeling · Advisor: Dr. Bapi Chatterjee · Collaborative Thesis (Team of 2)

- Built a **331-task, 24-repository C++ benchmark** (129,770+ agent trajectories across 11 LLMs, 14B–~1T parameters) to test whether coding-agent capability rankings hold stable across evaluation protocols.
- Built a **ReAct-style coding agent from scratch** with 5 tools (code search, file read, Docker-based compile/test, directory browsing, file search) under strict iteration/token/time budgets, and a hybrid retrieval pipeline combining BM25 sparse search with Qwen3-Embedding-8B dense retrieval via Reciprocal Rank Fusion.

- Generated the multi-model trajectory corpus via the OpenAI API and NVIDIA-hosted model endpoints (Meta, Qwen, and other open-weight models), and used Ollama for local model inference during pipeline development and testing.
- Found that agentic evaluation **(58.5% Pass@1)** diverges sharply from single-pass evaluation (31.1%) and produces large rank reordering across models, demonstrating that coding-agent capability is protocol-sensitive rather than a fixed model property.
- Designed a three-layer contamination-mitigation pipeline (function-body masking, pre-training corpus audit via infini-gram, repository-disjoint train/test splits), achieving 90%+ verified-clean task descriptions and zero train/test leakage.
- Co-developed a **Disagreement-Aware Credit Assignment (LDCA)** method for process reward models: combined Monte Carlo rollout labels with LLM-judge labels, fine-tuned a Qwen2.5-Coder-1.5B reward model with LoRA and a ListNet-ranking + MSE-calibration hybrid loss, and showed signed disagreement between the two supervision signals improves step-level ranking accuracy on 235,000+ held-out evaluation steps.

Agentic Math Reasoning & Proof Verification (IneqMath)Apr 2026 (2-week course project)
Agentic AI Systems Course · Python, Gemini, DSPy, GEPA, Lean 4 · Team of 2 · GitHub

- Built an LLM-agent pipeline with a solver-judge architecture and a process reward model to generate and granularly verify mathematical inequality proofs (e.g., Cauchy-Schwarz, Bernoulli's Inequality) step-by-step using Gemini, with Lean 4 formal proof verification; results and proof artifacts included in the repo.
- Applied GEPA-based prompt optimization via DSPy and benchmarked optimized vs. unoptimized prompts as the core evaluation metric for the project.

NeuroSync, Empathic AI for Inclusive Co-DesignJan 2025 – May 2025 (Team of 4)
Python, PyTorch, Flask, React 19 + TypeScript, Hume EVI · GitHub

- Built a PyTorch multi-modal fusion network fusing 5 facial and 5 vocal emotion scores with intensity-weighted embeddings and an adapter bottleneck to output real-time stress/engagement/fatigue scores, reaching MSE 0.0032 and cosine similarity 0.9877.
- Developed the React 19 + TypeScript client (Hume EVI voice capture, camera-based emotion capture, emotion-trend dashboards) and a Flask inference API powering chatbot and VR co-design modalities.

Research-Paper-to-Presentation AgentJan 2026 (1-week course project)
Agentic AI Systems Course · Python, LLM Agents, LaTeX · Individual · GitHub

- Built an agent that parses the LaTeX source of a research paper and automatically generates a corresponding PowerPoint/Beamer presentation summarizing its content, with a generated deck included in the repo (docs/demo_output.pdf).

Fashion Trend Assistant, Myntra Hackathon (RAG)Jun 2025 (2-week hackathon)
Python, Retrieval-Augmented Generation (RAG), FAISS · Team of 2

- Built a RAG system using FAISS for vector-based semantic search that ingested and organized fashion blogs by category and specification, then retrieved the most relevant, up-to-date style and trend information in response to user queries.

A Bi-Objective ε-Constrained Framework for Quality-Cost Optimization in LLM EnsemblesJan 2024 – May 2024
Independent Research · ICLR 2024 Tiny Paper, presented in Vienna, Austria · Team of 3 · Paper

- Developed a language-model ensembling meta-model, framed as a 0/1 knapsack bi-objective optimization, that cut inference floating-point-operation cost by 80% while preserving output quality.
- Integrated DeBERTa-based regression models for quality prediction and used GEN-FUSER for multi-model response aggregation.

ADDITIONAL PROJECTS

- CodeT5 Pre-Training & Optimization with DeepSpeed (Jul–Dec 2025, Team of 3): Pre-trained and optimized CodeT5 for code summarization on a national HPC cluster using data/model parallelism, mixed-precision training, and gradient checkpointing. (Code & checkpoints)
- Multi-Agent Website Builder (Feb 2026, 1-week course project): Designed a multi-agent orchestration system using CrewAI with a planner/orchestrator agent that decomposed a website-building task and delegated work to backend-developer, frontend-developer, and prompt-optimizer agents, reducing token consumption in agent-to-agent communication while preserving task completion quality.
- LINUX/UNIX Shell (Jan–May 2024, Team of 4): Built a LINUX shell supporting internal/external commands via process forking and the exec system-call family, with simulated OS-level scheduling and memory optimization. (GitHub)
- Grad-Guru: College Decision Support System (Aug–Dec 2023, Team of 4): Built a decision-support system integrating 10 scraped datasets into a 100,000+ record PostgreSQL database, automating ETL and comparative-analysis workflows and cutting manual research effort by 90%. (GitHub)

POSITIONS OF RESPONSIBILITY

- Teaching Assistant, IIITD (Jan 2025 – May 2025)
- Coordinator, Meraki IIITD, Art Society (Aug 2023 – May 2024)
- Organising Committee, E-Summit 2024 (Feb 2024 – Apr 2024)

EDUCATION

Integrated Degree (B.Tech + M.Tech), Computer Science and Engineering, CGPA 9.29/102021 – 2026
Relevant coursework: Data Structures & Algorithms, Advanced Programming, Algorithm Design and Analysis, Large Deep Learning Systems, Information Integration and Application, Fundamentals of Database Management Systems, Interactive Systems, Convex Optimization, Ethical Hacking Essentials, Network Science.
CBSE Class XII, 91.6%2020 – 2021
CBSE Class X, 91.6%2018 – 2019