Aditya Kuniyil Kattil

(765) 409-0422 | akuniyil@purdue.edu | LinkedIn | GitHub | Website

Education

Purdue University West Lafayette, IN
BS in Computer Science (Machine Intelligence, Security), Minor: Economics *May 2027*
- Relevant Coursework: Data Structures and Algorithms, Systems Programming, Data Mining, Machine Learning, Computer Architecture, Databases, Artificial Intelligence, Discrete Math, Linear Algebra, Programming in C

Experience

BoilerBlockchain | ***Solidity, Hardhat, ethers.js, React, EVM*** West Lafayette, IN
Technical Course Lead / Instructor *Jan 2025 - Present*
- Designed and delivered a **10-week blockchain engineering curriculum** covering **EVM execution model, Solidity, ERC standards, DAOs, DEX mechanics, and L2 scaling**.
- Developed hands-on Solidity labs using **Hardhat + ethers.js**, teaching **contract deployment, testing, debugging, and security best practices**.
- Taught and certified **100+ students**, improving course completion rate by **15%** and expanding the club's active developer team by **3**×.
- Promoted to **Course Lead** in Jan 2026 after scaling course delivery and improving student onboarding into the developer pipeline.

MOI Labs x BoilerBlockchain | ***TypeScript, React, Node.js, MOI Chain, A2A Protocol*** West Lafayette, IN
Software Engineer / Project Lead *Sep - Dec 2025*
- Architected and built **Sageo**, a **trust & discovery layer** for AI agents on the **MOI chain**, enabling **verifiable agent identity**, interaction tracking, and on-chain proofs atop Google's A2A protocol.
- Implemented a modular stack including a **REST API explorer**, **agent identity logic**, and **hashed interaction proofs** to support transparent client SDK integrations and agent discoverability.
- Designed interaction logging primitives that balance **privacy (off-chain payloads)** and **on-chain provability (hash commitment)**, facilitating cryptographically sound auditability for decentralized multi-agent ecosystems.

Solsten.in | ***Python, Vertex AI, DSPy, pytest*** Bengaluru, KA
Software Engineering Intern *Jul - Aug 2025*
- Developed an automated **medical receipt/invoice parsing pipeline** to extract structured **billing and inventory fields** from handwritten and printed hospital documents.
- Implemented **regex-based extraction** and post-processing with **Gemini-2.5-Flash** fallback for missing/low-confidence fields, improving robustness across edge-case receipts while reducing **inference costs** by ≈ **65%**.
- Iteratively refined **regex rule sets** using **DSPy** to evaluate prompt variants and analyze LLM outputs, improving field-level accuracy to ≈ **98.2%**.
- Partnered with QA to build a validation and unit testing suite using **pytest**, reducing manual correction workload by ≈ **80%** and improving extraction reliability.

Projects

Memora | *Next.js, TypeScript, Node.js, Hedera HCS, EVM, IPFS, Supabase* Spring 2026
- Built a **distributed system for verifiable agent execution**, enabling autonomous agents to execute tasks and log **replayable, tamper-resistant workflows** using Hedera HCS for ordered commits and IPFS for storage.
- Designed and implemented **event-driven backend APIs** (task dispatch, runtime ingestion, mission replay) supporting asynchronous execution pipelines and lineage-aware memory tracking.
- Developed a **task marketplace for autonomous agents** with bidding, scoring, and execution flows, integrating secure memory storage (**AES-256-GCM**) and on-chain permission checks via a Hedera EVM registry.
- Engineered a **full-stack interface in Next.js** for task input, agent discovery, and cinematic mission replay, enabling users to inspect execution steps and verify outputs in real time.

CosmosPool | *Solidity, Hardhat, Uniswap V3, Unichain, Python* ETHDenver Hackathon 2025
- Built a DeFi protocol enabling **single-token deposits into CLMM liquidity pools** by matching users with complementary liquidity providers to remove the dual-asset requirement.
- Implemented a **Solidity liquidity manager contract** to handle deposits, matching, and withdrawals, integrating Uniswap V3-style pool interactions for liquidity provisioning.
- Applied **Uniswap concentrated liquidity formulas** to compute optimal matched liquidity within a defined price range and automated end-to-end testing via Python scripts.

Technical Skills

Languages: Python, TypeScript, JavaScript, Solidity, SQL, Java, C, C++, HTML/CSS
Web3: Hardhat, Foundry, ethers.js, wagmi, RainbowKit, Uniswap V3, EVM, Base, Unichain
Frameworks/Libraries: React.js, Next.js, Node.js, PyTorch, scikit-learn, Pandas, NumPy
Tools: Linux, Git, Google Cloud, Firebase, Supabase, Vercel, HuggingFace, pytest, Cursor, GitHub Copilot, ClaudeCode