

Hriday Adani

803-357-9673 | hriday1136@gmail.com | linkedin.com/in/hridayadani | github.com/hriday1136

EDUCATION

Rutgers University - New Brunswick

Bachelor of Science; Computer Science and Economics

Relevant Coursework : Data Structures, Artificial Intelligence, Algorithms

New Brunswick, NJ

Expected Graduation - May 2027

PROJECTS

StorageKV | *Rust (zero runtime dependencies)*

- Engineered an embedded RocksDB/LevelDB-inspired key-value storage engine from scratch on an LSM-tree architecture with a WAL, immutable sorted SSTables with sparse indexing, and size-tiered compaction
- Implemented crash recovery via manifest checkpoint plus WAL replay, enforcing fsync ordering and atomic file installation so any acknowledged write survives crashes in the write path, similar to PostgreSQL's architecture
- Verified correctness through property testing against a reference model across **300+** randomized operation sequences with interleaved crash recovery, plus crash window tests for every durable operation
- Measured **sub-12 μ s p99** reads and quantified the cost of durability at approximately **1000x** over an in-memory baseline by isolating fsync overhead in same-machine benchmarks

ClauseWatch | *Next.js, FastAPI, PostgreSQL, OpenAI*

- Extracted structured fields from PDF and DOCX contracts at **97.9%** field-level accuracy (100% on effective and renewal dates) across a validated 30-document test set, using confidence-based OpenAI structured outputs
- Engineered 5 independently-secured HMAC token systems including webhooks, downloads, password reset, email verification, and cron auth, each with a dedicated secret and constant-time comparison to prevent timing attacks
- Designed a multi-tenant data isolation architecture by enforcing tenant-scoped queries at the ORM layer across every API endpoint, verified via 6 automated integration tests confirming **zero cross-tenant data leakage**
- Evolved a production PostgreSQL schema through 9 zero-downtime Alembic migrations, including safe handling of Postgres enum-type constraints unsupported by migration autogeneration, applied in an isolated transaction

AccessForge | *FastAPI, LangGraph, PostgreSQL, Redis/RQ, Docker, Playwright, Axe-core*

- Developed an end-to-end accessibility remediation platform that scans React apps, generates and independently verifies WCAG fixes, and opens GitHub PRs, designed as a deterministic state machine using LangGraph
- Architected a crash-recoverable async pipeline using PostgreSQL-backed state and Redis workers, verified via a real worker crash captured during development, resuming task with **0 duplicate GitHub branches** or lost progress
- Validated LLM-generated accessibility fixes across 2 independent React/Vite codebases, including an independently built, real production app, using multimodal visual-regression comparison of rendered before/after screenshots
- Designed a confidence-gated localization check that deferred **68%** of detected violations to human review instead of guessing when no signal matched a source file, resolving **100%** of attempted accessible-name fixes on first pass

WORK EXPERIENCES

Scientific Software Research Assistant

RCSB Protein Data Bank

Jul. 2026 - Present

New Brunswick, NJ

- Orchestrated a new MCP server tool exposing PDB search API to LLM agents enabling multi-service AND/OR queries in a single call, by composing the filter nodes from the 7 extended tools through a custom validation layer
- Rectified **2 silent data corruption bugs** in MCP server's result-grouping logic by tracing parameter flow against the tool's own docstrings, preventing the formation of invalid API parameters and user configurations

Summer Research Fellow (CSI SURF)

Rutgers Artificial Intelligence and Data Science Collaboratory

May. 2026 - Jul. 2026

New Brunswick, NJ

- Redesigned a dictionary-driven column-typing classifier in py-mmcif, an open-source python package, adding automatic data type detection to cut molecular structure file sizes by **27%**, submitted upstream as a pull request
- Optimized floating-point binary encoding in py-mmcif's compression pipeline, cutting compressed file sizes by up to **76%** across 37 varied PDB structures, working with a team of 4 researchers on the encoding chain design

TECHNICAL SKILLS

Languages: Python, Java, C, PHP, JavaScript (React, Next), SQL, HTML/CSS, Rust

Libraries/Frameworks: Tailwind CSS, Tauri, Flask, FastAPI, LangGraph, Playwright, Axe-core

Developer Tools: VS Code, IntelliJ, Git, Docker, WSL, Alembic, Neon, Cloudflare, MCP, Redis