ARNAV PURUSHOTAM

📱 720-351-1267 — @ arnavpsusa@gmail.com — LinkedIn — GitHub — 📍 San Jose, CA, USA

Education

University of Colorado, Boulder Boulder, USA | August 2024 – May 2026
MS, Computer Science *GPA: 3.9/4.0*

BMS College of Engineering Bengaluru, India | July 2020 – July 2024
B.E., Computer Science *GPA: 4.0/4.0*

Skills

Programming Languages: Python, C/C++, Java
Web Development: Flask, REST APIs, Agentic AI LLMs, Langchain, SpringBoot
Tools & Technologies: CMake, Git, Docker, CI/CD, Kubernetes, AWS
Databases: PostgreSQL, MongoDB

Open Source Contributions

curl/libcurl Nov 2025 – Present
Open Source Contributor (C / Portability / Toolchains)

- Contributed upstream fixes to **curl/libcurl** (large-scale, production, public-facing C codebase), hardening the `curl.h` public API bitmask macros against LP64 type-width pitfalls by constraining "ANY/ALL" masks to a **32-bit flag domain**, ensuring consistent behavior across 32/64-bit platforms. (PR #20416)
- Eliminated **GCC 15.2 `-Woverflow/-Wconversion`** failures (often promoted to `-Werror` in CI) by preventing unintended high-bit propagation from `~mask` / `~0L` patterns, making auth/protocol/SSH option masks **warning-free, portable, and stable** when stored in 32-bit contexts.
- Proposed and validated a cross-compiler logging strategy that enabled **CURL_DISABLE_VERBOSE_STRINGS** to truly compile out verbose trace format strings on **MSVC/Windows** via C99-style variadic trace macros; collaborated in public review with maintainers and was **credited in the final upstream solution**. (PR #20387)

Projects

Modular C++ Database Engine August 2025 – September 2025
https://github.com/Arnav-Purushotam-CUBoulder/cpp-db-engine

- Built a modular database engine in modern C++ (C++17/20): an in-memory hash map and a file-backed key–value store delivering $O(1)$ lookups with durable writes; benchmarked against Redis to sanity-check throughput and latency.
- Designed a pluggable storage layer (PIMPL) to swap memory, disk, and cached-disk backends without changing call sites—setting the stage for sharding and future distribution.
- Added secondary indexes, bucketed namespaces, and templated key/value types (C++20 concepts) to enable fast searches over STL containers with zero-copy paths where possible.
- Automated cross-platform builds and tests with CMake and Catch2, using GitHub CLI; verified on Linux (GCC/Clang), macOS (Clang), and Windows (MSVC).
- Used complexity analysis to guide performance trade-offs.

Experience

CU-BioFrontiers October 2024 – Present
Software Engineer - Part-time

- Built an LLM AI Agent with RAG, LangChain and python integrating many databases and docs, enabling natural language queries over multiple data sources and saving researchers 30 minutes per complex request.
- Automated shift scheduling for **200 CU Stores employees** on **AWS** by ingesting Excel availability/requirements from S3 and generating assignments via a 3-stage Airflow pipeline, replacing manual scheduling and delivering schedules end-to-end in ~5 minutes/run (EC2/Flask trigger, Lambda, RDS, SQS, DynamoDB, SES).
- Worked on the BioBit internal platform using Java Spring Boot, Maven, HTTP RESTful APIs, PostgreSQL, Docker for 10+ labs and over 100 researchers, significantly accelerating research productivity with a scalable, distributed microservices architecture.
- Implemented OAuth2-based authentication and multi-tenancy with Keycloak, enforcing secure role based access control (RBAC) for 10+ labs and 100 researchers-boosting usability, data privacy, security.