

Advit Singh

951-512-2003 | advits234@gmail.com | linkedin.com/in/advit-singh | advitsingh-project.com

Education

University of California, Riverside | Bachelor of Science in Computer Science

Expected Graduation: 2027

Coursework: Data Structures & Algorithms, Systems Architecture & Design, Artificial Intelligence, Compiler Design & Optimization

UC Riverside Competitive Coding Club, Board Member – Lead weekly algorithmic sessions for 50+ members, curate LeetCode and Codeforces contest problems, and coordinate intercollegiate programming competitions.

Technical Skills

C++, C, C#, Java, Python, Rust, TypeScript, Go, JavaScript, Assembly, SQL, HTML/CSS, React.js, React Native, Node.js, Next.js, FastAPI, Spring Boot, AngularJS, GraphQL, PyTorch, TensorFlow, LangChain, HuggingFace, OpenAI API, .NET, Kafka, RabbitMQ, Git, Linux, Vagrant

Experience

Embedded Systems Intern | Computer Science Department, UC Riverside

May 2026 – Present

Advisor: Professor Jay Lim – jlim@ucr.edu

- Developing low-level embedded software in Rust and C for a brain-computer interface, building real-time firmware for neural signal monitoring, safety controls, and hardware integration.
- Implemented signal-processing routines and optimized interrupt-driven sampling, reducing signal response latency by 9% across 20,000+ simulation and benchmark runs.
- Built AI-assisted agent-loop engineering workflows using ClaudeAI, GitHub Copilot, and Obsidian, accelerating debugging and design iteration while improving reliability across the embedded system.

Software Engineering Intern | Alien Attorney

Nov 2025 – Mar 2026

- Built an AI-powered semantic search system using Python, RAG, HuggingFace, Pinecone, and OpenAI APIs, architected as scalable backend microservices across 10,000+ client legal documents.
- Designed a service-oriented document ingestion and retrieval pipeline, improving search accuracy from 46% to 85%, cutting processing time from 14s to 4.3s, and reducing estimated attorney research time by 60%.
- Deployed Dockerized services with CI/CD, adding AI guardrails, anomaly detection, and token-usage optimization to reduce LLM costs and improve reliability while collaborating directly with attorneys and engineers.

Computer Science Researcher | Hoodle Labs, UC Riverside

November 2024 – Mar 2025

- Developed data science and scientific-computing pipelines using Python, Pandas, NumPy, and C++ to analyze large entomology datasets and extract quantitative insights from experimental research data.
- Optimized C++ data-processing workflows using SIMD, parallel computing, and memory profiling, reducing runtime by 60% across 50GB+ datasets and achieving 3x throughput across 1M+ simulation runs.
- Deployed scalable data-processing workloads on Azure and collaborated with entomology researchers to translate experimental objectives into reproducible analysis and data-driven results.

Projects

Concurrent Key-Value Storage Engine | C++, Multithreading, Distributed Systems | [Live](#) | [GitHub](#)

- Built a C++ storage system that works like a shared digital dictionary, allowing multiple applications to save and retrieve information at the same time without overwriting or corrupting each other's data. Exposed the system through REST APIs and applied multithreading, SOLID principles, and object-oriented design patterns to support safe concurrent access.
- Engineered a 64-shard hashmap with fine-grained reader-writer locks and an LRU caching layer that cut read latency from 45ms to 6ms, backed by a multi-client TCP/IP and UDP server sustaining 159,000+ operations per second across 200 concurrent connections with zero data corruption; instrumented with high performance computing profiling tools.

Distributed E-Commerce Platform | Next.js, Node.js, FastAPI, Kafka, Kubernetes, AWS | [Live](#) | [GitHub](#)

- Built a full-stack commerce platform using Next.js, Node.js, FastAPI, GraphQL, and PostgreSQL to manage product catalogs, pricing, inventory, carts, and transactional order workflows.
- Engineered 4 Dockerized microservices using Kafka, Redis, and WebSockets for event-driven order processing and real-time updates, supporting scalable communication across catalog and transaction services.
- Deployed services on Kubernetes using AWS EKS and Terraform with GitHub Actions CI/CD, adding caching and asynchronous processing to improve backend performance, reliability, and service consistency.

Open-Source Contributions

LangChain | Python, LLMs, RAG, Vector Databases, Knowledge Retrieval

- Contributed a fix to the document loader module that reduced peak memory usage by 43.5% (11MB to 6.2MB) for large file ingestion pipelines used in RAG-based knowledge retrieval systems, making it significantly more efficient for production use. ([Issue #632](#))
- Resolved 3 critical bugs that were causing agentic AI pipelines to crash mid-run when processing large documents through vision-language model integrations, improving stability for developers building RAG and reinforcement learning applications on top of LangChain.