Vihaan Sarin

608-282-5559 | vihaansarin@gmail.com | linkedin.com/in/vihaan-sarin

Education

University of Wisconsin–Madison Madison, WI
B.S. Computer Science and Data Science ***GPA: 3.74*** *Sep 2022 – May 2026*

Technical Skills

Languages : Python, Java, JavaScript, TypeScript, SQL, C++, C, Bash, R, HTML, CSS
Frameworks & Tools: React.js, React Native, Node.js, Express, Flask, FastAPI, Apache Spark, JUnit, JavaFX, Docker, Git, GCP, BigQuery, VS Code, PyCharm, IntelliJ, Eclipse, WSL
Databases & Libraries: PostgreSQL, MongoDB, Cassandra, pandas, NumPy, SDV, Bootstrap, dplyr, Tidyverse, ggplot2
Systems & Platforms: Unix/Linux, HDFS, Distributed Systems, POSIX Threads, Concurrency, Systems Programming, REST APIs, CI/CD Pipelines
Concepts & Coursework: Algorithms, Data Structures, Operating Systems, Database Management, Object-Oriented Design, Machine Learning, Data Architecture, Data Modeling, Data Visualization, Big Data Systems

Experience

Software Engineer Intern (Data & AI) May 2025 – Nov 2025
Innover Digital *Remote / Atlanta, GA*

- Designed a full-stack React + FastAPI platform for synthetic data generation (single-table and multi-table ZIP workflows), and built an LLM-based generation mode that produces schema-validated sample rows using OpenAI-SDK prompts, guardrails, and preset templates.
- Integrated five SDV models (CTGAN, TVAE, GaussianCopula, CopulaGAN, HMASynthesizer) alongside a PII masking engine (spaCy NER + Faker), enabling secure synthesis of relational datasets while preserving schema constraints.
- Created tabbed previews, post-generation dashboards, and export pipelines in React, which **streamlined dataset exploration and reduced setup time by 30%**, enhancing efficiency for analytics workflows.
- Enhanced metadata inference and constraint-handling pipelines, **reducing runtime by 25–40% and cutting training errors**, leading to an internship extension of **3 months** after successful deployment.

Web Development Intern Jul 2024 – Aug 2024
Thales Group *Noida, India*

- Developed a full-stack **animal adoption platform** using HTML, CSS, and JavaScript, implementing interactive filtering, secure adoption workflows, and responsive layouts to enhance user engagement, accessibility, and cross-device compatibility.
- Architected modular components and optimized client-side rendering with efficient DOM updates, **reducing page load times by 25%** while improving maintainability, scalability, and long-term integration flexibility for future extensions.
- Integrated and validated backend routes for seamless data exchange, applying Thales' engineering standards to ensure **secure, fault-tolerant, and production-ready web solutions** aligned with enterprise-grade deployment requirements.

Projects

Aura Farmer | *React Native, Supabase* May 2025 – Nov 2025

- Created a social app where users share content and earn "aura" points, exploring engagement models without likes or follows.
- **Integrated RESTful Supabase APIs** for real-time post, comment, and media handling within React Native profile and vault interfaces, ensuring secure storage policies, consistent performance, and scalable data management.
- **Optimized Supabase queries**, cutting redundant API calls by 40% and reducing profile fetch time from 2.0s to 1.5s, which **significantly enhanced app responsiveness and refined usability during feature testing**.
- Applied an agile **SCRUM-based development cycle** to plan sprints, track progress, and coordinate across roles, ensuring on-time delivery of the prototype and first full-featured release.
- **Recognized as 1 of 3 finalists** from 15 teams in Voodoo's Studio incubator after presenting the project, highlighted for technical execution, rapid prototyping, scalable product vision, and strong teamwork in development.

MiniSpark | *Parallel data framework in C* Apr 2025

- Constructed a Spark-inspired parallel data framework in C with Resilient Distributed Datasets (RDDs), supporting map, filter, join, and partitionBy; enabled lineage recomputation that **recovered lost partitions with under 5% overhead**.
- Architected a Directed Acyclic Graph (DAG) scheduler using POSIX threads and a worker pool, which **accelerated data processing throughput by 4x on datasets up to 10 GB** in multicore environments.
- Implemented thread-safe queues, mutexes, and custom memory allocators to **eliminate race conditions, cut memory fragmentation by 20%, and improve stability under heavy workloads**.

WSH | *Unix Shell in C* Apr 2025

- Authored a Unix-like shell from scratch with interactive and batch modes, supporting fork/exec, **128-argument parsing**, environment variables, built-in commands, piping, and background jobs—covering **over 80% of Bash functionality** used in daily workflows.
- Verified reliability by running **500+ diverse commands**, including nested pipelines and background jobs, achieving **zero memory leaks under Valgrind** and execution speeds within 5% of Bash on standard workloads.