Darshan Golchha

Madison, Wisconsin, USA | +1 608-405-9563 | dgolchha@wisc.edu | darshangolchha.com

EDUCATION

University of Wisconsin–Madison
B.S. in Computer Science and Data Science, Expected May 2026
GPA: 3.8/4 (Dean's List)

Relevant Coursework: Object-Oriented Programming, Database Management (SQL), Big Data and Distributed Systems, Operating Systems, Data Structures & Algorithms, Machine Learning, Data Science Modeling (R), Data Science Programming (Python), Web Development

SKILLS

Languages: Java, Python, C, C#, C++, SQL, R, JavaScript
Frameworks/Tools: Spring Boot, Flask, Django, React, React Native, scikit-learn, Docker, REST API, JPA, Redux Toolkit, Web Sockets, Git, Unix/Linux, Tailwind CSS, JWT, Spring Security, Google OAuth, Firebase, Three.js, GSAP.
Databases & Cloud: MySQL, MongoDB, AWS, GCP, AWS S3, AWS EC2
Specialized: Full Stack Development, Software Development Lifecycle (SDLC), Operating Systems, Database Design, Distributed Systems, Concurrency (threads, synchronization), Anomaly Detection, ML Pipelines, Agile Methodologies (Scrum).

EXPERIENCE

Software Engineering & ML Intern — Opstree Solutions *(May 2024 – Aug 2024)*

- Built an **anomaly detection pipeline (Isolation Forest, scikit-learn)** to flag infrastructure anomalies in real time, improving monitoring accuracy.
- Engineered an **LLM-integrated code quality system** (SonarQube + SonarCloud), cutting critical bugs by 40% and automating reviews.
- Designed a **multi-VCS pull request analyzer** and a **dynamic infra design tool (React + Django + Google Gemini LLM)** that generated real-time high-level system architectures.

Software Developer Intern — R Systems *(May 2025 – Aug 2025)*

- Developed a **large-scale chat system** (1,000+ users) with React Native + Spring Boot, featuring one-to-one/group messaging, media sharing, and real-time notifications.
- Automated **CI/CD pipelines** for Android/iOS with GitHub Actions, achieving 70% faster release cycles.
- Deployed backend APIs on **Google Cloud Platform** with robust scaling and security.

Software Engineer Intern — Nucleus Software Exports *(Jun 2023 – Aug 2023)*

- Built **Nucleopedia**, a React + Spring Boot webapp knowledge platform adopted by 1,500+ employees, reducing onboarding time by 70%.
- Optimized **MySQL queries (–75% latency)** and integrated **Spring Security + JWT** to cut unauthorized access by 98%.

PROJECTS (More projects and info at https://www.darshangolchha.com/)

MiniSpark — Distributed Data Processing Framework *(CS537, Spring 2025)*

- Implemented a **mini version of Spark** supporting map, filter, join, and partition operations over RDD-like data structures.
- Built a **thread pool + work queue system** for efficient DAG scheduling, ensuring **parallel materialization of RDDs** across cores.
- Added runtime metrics collection for profiling execution, simulating real Spark behavior.

AI Commit Risk Analyzer — Opstree Solutions *(Jun 2024)*

- Automated bug prediction using **LLMs + ML Models + CI/CD integration**, reducing critical production issues by 40%.
- **LLMs** gave a comprehensive code review, **ML Models** used historical data associated with files and similar changes to predict severity of such a change. **SonarQube** metrics help provide further insights.
- Developed a **Master Severity Index** with dashboards, improving prioritization by 30%.

Super Tic Tac Toe AI — Personal Project (Oct 2024)

- Designed an **AI-powered multi-grid Tic Tac Toe** with **minimax algorithm + alpha-beta pruning** for optimal decision-making.
- Built backend in Flask (Python) for **state management & AI computations**, and a React frontend for interactive gameplay.
- Optimized algorithmic performance, achieving faster and more accurate predictions in real-time play.

Unix Shell (wsh) — CS537 Operating Systems Project (Spring 2025)

- Built a **Unix-style shell** in C supporting interactive & batch modes, environment/shell variables, and error handling.
- Implemented **pipes, command substitution, and variable substitution**, enabling concurrent process execution with fork(), execv(), wait(), pipe(), and dup2().
- Added support for built-in commands (exit, export, local, vars, ls, ps) with correct parsing and precedence rules.

ACTIVITIES

- Member, **Wisconsin Robotics – Software Team**
- Goalkeeper, Men's 7v7 Soccer Team