

# HRISHIKESH SHINDE

Madison, WI 53703 | (608) 690-1162 | hshinde@wisc.edu  
linkedin.com/in/hrishikeshh-shinde | github.com/hrishikeshh-shinde

## Education

<b>University of Wisconsin, Madison</b> <i>Master of Science in Computer Science</i> <b>GPA: 3.81/4</b>	<b>May 2026</b> <i>Madison, WI</i>
<b>Coursework:</b> Distributed Systems, Advanced Operating Systems, Advanced Database Systems, High Performance Computing, Big Data Systems, Computer Networks, Machine Learning, Artificial Intelligence	

  

<b>Pune Institute of Computer Technology</b> <i>Bachelor of Engineering in Electronics and Telecommunications</i> <b>CGPA: 3.83/4</b>	<b>Jun. 2022</b> <i>Pune, India</i>
<b>Coursework:</b> Data Structures and Algorithms, Object Oriented Programming	

## Technical Skills

**Languages:** C, C++, Python, Java, Go, Rust  
**Systems & Backend:** Linux, Multithreading, OpenMP, CUDA, gRPC, Kafka  
**Databases:** SQL, NoSQL, Cassandra  
**Cloud & Tools:** AWS, Docker, Git

## Experience

<b>Software Development Engineer</b> <i>Mastercard</i>	<b>Jul. 2022 – Jul. 2024</b> <i>Pune, India</i>
• Developed scalable backend services in a distributed transaction processing system building APIs for transaction switch, ETL, scheduler and ISO converter services used by internal teams and partner systems.	
• Identified, debugged, and fixed backend data consistency and performance issues by analyzing database queries, transaction paths, and concurrent request flows across production services.	
• Implemented server monitoring and structured logging to track service health, request failures, and latency issues, enabling faster identification of production bottlenecks.	
• Ensured 95%+ uptime for dev and staging environments supporting 50+ partner banks by resolving infrastructure issues, certificate expirations, and deployment failures.	
• Reduced production issue resolution time by performing root cause analysis through request tracing and log based debugging during on-call support.	
• Improved system correctness by creating and maintaining automated test suites, increasing test coverage from 65% to 90% and significantly reducing manual testing effort.	

## Projects

<b>F2FS Data Temperature Prediction</b>   <i>Linux FS internals</i>	<b>Nov 2025 - Dec 2025</b>
• Improved data placement in Linux F2FS by predicting page lifetimes using a weighted moving average of overwrite behavior, enabling lifetime aware data placement in the write path.	
<b>External Sort Engine</b>   <i>C++, Database Internals</i>	<b>Oct 2024 - Dec 2024</b>
• Built an external merge sort engine to handle datasets larger than memory by creating cache-sized runs and spilling data from RAM to disk during execution.	
<b>Custom Filesystem with RAID Support</b>   <i>C, FUSE</i>	<b>Nov 2024 - Dec 2024</b>
• Built a user-level block-based filesystem using FUSE with support for core POSIX operations like read, write, mkdir, unlink, and metadata persistence.	
<b>Concurrent HashMap</b>   <i>C++, OpenMP</i>	<b>Mar 2025 - May 2025</b>
• Built a concurrent hashmap in C++ supporting parallel inserts, lookups, and deletes on workloads of up to 1M keys using fine-grained locking and sharding.	
• Implemented dynamic resizing with parallel rebalancing using OpenMP, achieving up to 90% CPU utilization and significantly reducing resize latency under multi-threaded workloads.	