Prince Savsaviya

princemsavs@gmail.com | (951) 377-3380 | linkedin.com/in/prince-savsaviya | github.com/princesavsaviya | princesavsaviya.github.io/profile

EDUCATION

University of California, Riverside | *Master of Science in Computer Science, GPA: 3.59/4.0* March 2026

Adani Institute of Infrastructure Engineering | *B.E. Information & Communication Technology, GPA: 8.71/10* June 2024

EXPERIENCE

Author & Maintainer, TethrLink | Site | GitHub Mar 2026 – Present

Open-Source | Python, GStreamer, PipeWire, Wayland/D-Bus, GTK4, Jetpack Compose, MediaCodec

- Engineered a Linux-to-Android second-monitor system in Python using GStreamer and PipeWire, negotiating H.264 hardware encoders (NVENC, VA-API, QSV) at runtime and verifying each candidate against the live capture resolution instead of trusting driver metadata; benchmarked throughput up to 320 FPS at 720p and 196 FPS at 1080p on NVENC
- Architected Wayland-native screen capture via Mutter's ScreenCast D-Bus API, creating a true virtual display sized at connect time from the smaller of the two devices' heights scaled to the tablet's aspect ratio, keeping the shared pointer-crossing edge aligned
- Shipped touch and pointer input in TethrLink v2.0.0 by integrating GNOME's RemoteDesktop API, with gesture interpretation written in pure Kotlin and unit-tested independent of the Android runtime
- Hardened peer discovery in Python by replacing a hardcoded tether subnet with runtime detection of USB-attached interfaces over sysfs, fixing cross-device connection failures and scoping the server to physically-tethered peers by construction, not configuration

Graduate Student Researcher May 2025 – Mar 2026

University of California, Riverside

- Engineered a modular fluid-structure interaction (FSI) simulation pipeline coupling Navier-Stokes fluid, nonlinear solid mechanics, and ALE mesh-motion solvers using FEniCSx, PETSc, and MPI
- Validated 9 standalone benchmarks (CFD1-3, CSM1-3) achieving $<$0.5% error vs. published references on refined meshes across 6 refinement levels (L0-L5)
- Integrated strong Dirichlet-Neumann coupling with Aitken relaxation and adaptive time stepping, reproducing stable FSI1/FSI2 benchmark oscillatory behaviour
- Built a configuration-driven reproducibility framework supporting 6 mesh refinement levels and checkpoint-based restart, reducing re-run setup from manual scripting to a single config file

Research Intern May 2023 – Dec 2023

Adani Institute of Infrastructure Engineering

- Devised a hybrid PDE-ML model for Li-Ion battery degradation prediction, reducing MSE by 98% (10.36→0.21), RMSE by 86% (3.22→0.46), and MAE by 65% (0.74→0.26) by integrating physics-informed PDEs with Gaussian Process Regression
- Forecasted State of Charge, State of Health, and Remaining Useful Life of NCA batteries under varying temperature and C-rate conditions using an ensemble of algorithms (GPR, Random Forest, SVM, XGBoost, Neural Networks)
- Co-authored findings in *Journal of Power Sources* (Impact Factor ∼8.1) | Paper

PROJECTS

Wildfire-Twin | *Python, Kafka, Docker, Spark, Apache Sedona, Streamlit, DuckDB, H3* | GitHub Jan 2026 – Mar 2026

- Constructed a high-throughput Kafka-to-Spark geospatial pipeline processing live fire streams against 11.5 million building footprints, maintaining a sub-30-second processing SLA during bursts of 1,000 concurrent events
- Reduced P95 processing latency by 91% (215s down to 18.65s) by replacing inefficient Python UDFs with native Sedona Java expressions and executing broadcast spatial joins
- Stabilized Streamlit dashboard rendering to prevent browser crashes at statewide zoom levels using Uber's H3 (Resolution 7) indexing and a low-latency DuckDB analytical sink

Application Load Balancer | *Go, TypeScript, Docker, net/http* Apr 2025 – Jun 2025

- Programmed a Layer-7 load balancer in Go using net/http, implementing Round Robin, Weighted Round Robin, and Least-Connections routing for concurrency-safe traffic management
- Built a full-stack control plane with Go and TypeScript for real-time configuration of traffic weights and circuit-breaker thresholds
- Deployed a cooldown window mechanism for unstable servers before reintegration to prevent cascading outages; containerized the system with Docker for scalable deployment

RetriVeX | *Python, Django, Elasticsearch, FAISS, BERT, PyLucene* | GitHub Jan 2025 – Mar 2025

- Formulated a hybrid semantic and keyword search engine over a 150K+ record drug database, improving query relevance by combining dense vector embeddings (BERT/FAISS) with sparse retrieval (Elasticsearch BM25)
- Achieved sub-second semantic similarity search by generating context-aware PyTorch embeddings and constructing an optimized FAISS vector index
- Shipped a Django REST application with 3 plug-and-play search backends (Lucene, Elasticsearch, BERT/FAISS) for real-time drug information lookup

TECHNICAL SKILLS

Languages: Python, Go, TypeScript, Java, Kotlin, C++, SQL

Backend & Systems: Django, net/http, GStreamer, PipeWire, Wayland/D-Bus, Android SDK, Jetpack Compose, MediaCodec, MPI, FEniCSx, PETSc, Linux

Data & Infrastructure: Docker, Apache Kafka, Apache Spark (Sedona), Elasticsearch, DuckDB, Git

Machine Learning & AI: PyTorch, Scikit-learn, Transformers/BERT (HuggingFace), FAISS, Deep Learning, NLP, Physics-Informed ML, Model Evaluation