Subhash Yadav

(431)-588-3184 | yadavs2@myumanitoba.ca | Winnipeg, Canada | Website | LinkedIn | GitHub

Summary

Computer Science student with hands-on experience building scalable systems, AI/ML pipelines, and full-stack applications. Developed projects involving distributed systems, cloud-based deployments, and data-driven architectures. Currently working on real-time systems with a focus on backend reliability and performance.

Education

University of Manitoba Winnipeg, Canada

Bachelor of Computer Science, Minor in Mathematics *Expected April 2027*

- **Relevant Courses:** Software Engineering 2, Human Computer Interaction, Databases, Distributed Systems, Operating Systems, Data Structures and Algorithms 3, Object-Oriented Programming, Artificial Intelligence 1, Machine Learning 1, etc.

Technical Skills

Frontend: React.js, Next.js, JavaScript, HTML5, CSS3
Backend: C#, ASP.NET Core, Java, Python, FastAPI, C, C++
Databases: MongoDB, PostgreSQL, SQL, NoSQL, Vector Databases
AI Concepts & Tools: LLMs, NLP, RAG, AI Agents, LangChain, LangGraph
ML Concepts & Tools: PyTorch, TensorFlow, NumPy, Pandas, Deep Learning, CNNs, RNNs, LSTM, Transformers
Infrastructure & Cloud: Docker, Kubernetes, AWS, Azure, GCP, Terraform, CI/CD, REST APIs
Testing: Unit Testing, Integration Testing, End-to-End (E2E) Testing, Acceptance Testing, REST API Testing, JUnit, Mockito, xUnit, MSTest, Moq, PyTest, Postman
Operating Systems: Linux/Unix, Windows
Developer Tools: Git, GitHub, GitLab, VS Code, Cursor, Android Studio

Experience

Industry Project

Applied AI Systems Engineer January 2026 – April 2026

BIL Security Service Canada Ltd, Industry Project (University) *Winnipeg, Canada*

- Implemented a 10-second rolling ring-buffer architecture that selectively extracts and analyzes critical video segments from live RTSP streams before central server ingestion.
- Validated system efficiency across multiple models on Intel i7 hardware, achieving a peak real-time rate of ∼21 FPS with a minimal detection latency of ∼49ms using YOLOv8 architectures.
- Justified the adoption of YOLOv8-Small, which delivered a 5.5X gain in detection density and a +446% accuracy breakthrough compared to legacy Nano-scale models.
- Developed a multi-stage verification funnel utilizing silhouette aspect-ratio checks (< 0.85), center-frame positioning bias, and threat-priority policies to eliminate environmental false positives.
- Engineered a modular model registry that allows for "Live Swaps" of object detection models in system memory without interrupting the active RTSP video stream.

Projects

CarDex | Card Trading Platform | Group Project (SWE-2) + Personal Project September 2025 – March 2026

Docker, Terraform, GitHub Actions, ASP.NET Core, C#, React, TypeScript, PostgreSQL | GitHub | Medium | Live

- Led backend and DevOps using ASP.NET Core, React, PostgreSQL, and Azure to build a containerized full-stack trading platform for a 6-person team, resulting in a production-ready app with a fully automated CI/CD pipeline.
- Architected a three-layer system (API → Services → Repository) using interface-driven contracts to enable parallel feature development, resulting in decoupled Garage, Trade, and Pack-opening modules built against mock services before database logic was finalized.
- Eliminated a critical race condition in the Trade Service using per-trade `SemaphoreSlim` locks in a `ConcurrentDictionary` to prevent simultaneous trade acceptance, resulting in exactly-once execution without blocking unrelated trades.

- Built a custom sliding-window rate limiter using a concurrent queue to isolate aggressive clients, resulting in dynamic `429 Too Many Requests` responses with `Retry-After` headers that protected platform performance for all users.
- Diagnosed and resolved a linear payload growth bottleneck discovered during load testing using pagination and selective fetching to bound response sizes, resulting in constant latency at scale for card collection endpoints.
- Automated containerized deployments using GitHub Actions, Terraform, and Docker Compose to orchestrate the multi-container Azure App Service environment, resulting in consistent builds and zero-downtime releases across all team members.

UStep App | Grade Analytics Platform | Group Project (SWE-1) + Personal Project January 2025 – July 2025

Java, Android Studio, HSQLDB, Python, JUnit, Espresso, GitHub Actions | GitHub | Medium | APK

- Engineered backend grade calculation logic using an N-Tier architecture in Java to decouple the UI from data layers, resulting in seamless swapping between in-memory testing stubs and a live HSQLDB with no UI changes required.
- Built a Python web scraper to crawl the university's public course catalogue and parse HTML descriptions and prerequisites, resulting in auto-generated SQL INSERT statements that populated over 3,100 courses without manual data entry.
- Diagnosed and resolved a critical main-thread crash caused by loading 3,114 courses using infinite-scroll pagination and optimized SQL text-search queries, resulting in smooth 60fps scrolling and elimination of memory exhaustion.
- Stabilized a flaky Espresso UI test suite using `ActivityScenario` and standardized intent lifecycles to resolve async rendering timing issues, resulting in consistent, reliable automated test runs in the GitHub Actions CI/CD pipeline.

Voice-Driven Q&A Chatbot | Personal Project September 2025

LangChain, Python, FastAPI, RAG | GitHub

- Developed a web application using LangChain, FastAPI, and RAG to enable document uploads and multi-modal (text + voice) querying, resulting in instant and accurate answers grounded in personal document content.
- Integrated Pinecone vector databases with LLMs using LangGraph to orchestrate complex AI retrieval workflows, resulting in optimized response quality through advanced prompt engineering.

Automated ETL Data Pipeline | Personal Project March 2026 – April 2026

Apache Airflow, Python, PostgreSQL, AWS S3, Relational Data Modeling | GitHub

- Engineered a fault-tolerant ETL pipeline using Apache Airflow, writing custom PythonOperators to orchestrate daily DAGs that extract semi-structured JSON payloads from external REST APIs.
- Architected a dual-storage data ecosystem, utilizing AWS S3 as a data lake for raw payload preservation and a PostgreSQL instance as a staging warehouse for serving analytical queries.
- Developed a scalable Python transformation layer leveraging Airflow XComs for data lineage, utilizing NLP techniques to parse unstructured job descriptions into a strict relational schema.
- Integrated automated Data Quality (DQ) checks as strict Airflow task dependencies, safeguarding database integrity by enforcing schema validation and executing conditional UPSERT operations.

HCI To-Do App | Group Project (HCI Class) September 2025 - December 2025

HTML, CSS, JavaScript | GitHub | Live

- Built a To-Do web app using HTML, CSS, and JavaScript to demonstrate key HCI principles through iterative user testing, resulting in a simple, intuitive, and user-centered interface evaluated for accessibility and usability.

IMDB Sentiment Analysis | Personal Project June 2025

Python, NLP, scikit-learn | GitHub | Container

- Built a sentiment analysis web app using TF-IDF and Logistic Regression to classify IMDB movie reviews in real time, resulting in a live interface that predicts Positive or Negative sentiment from user-submitted text.

Other Experience

Customer Service Representative May 2023 – Present

Sobeys, Part-time, Casual *Winnipeg, Canada*

- Provided friendly and efficient customer service in a fast-paced retail grocery environment, assisting customers with inquiries, product location, and checkout processes.
- Collaborated with team members to maintain store standards and ensure a positive, welcoming shopping experience for all customers.