

# Rahul Rakesh

437-983-4874 | rahul.rakesh1012@gmail.com | linkedin.com/in/rahulrakesh10 | rahulrakesh.dev

## Education

### University of Western Ontario

2023 – 2028

*B.Sc. Honours Specialization in Computer Science (Co-op)*

*London, Ontario*

- Relevant Coursework: Data Structures & Algorithms, Databases, Computer Architecture, Operating Systems

## Experience

### Machine Learning Engineer

June 2026 – Aug. 2026

*Fly Rank*

*Remote*

- Built a Python data pipeline processing a 28,795-row search dataset (90-day trailing aggregates), engineering features across 4 signals with leakage-safe, client-grouped transformations
- Designed the evaluation methodology and benchmarked a Random Forest model against a baseline on a held-out 5,759-row test split; validated feature reliability using permutation importance
- Shipped an automated decision-support API output (“Action Playbook”) that replaced manual review with data-driven prioritization for downstream teams

### Freelance Web Developer

Sept. 2024 – Present

*Self-Employed*

*Remote*

- Built and maintained backend services and REST APIs for 5+ clients using Node.js, Express.js, TypeScript, and PostgreSQL, deployed with automated CI/CD (GitHub Actions)
- Improved application performance with 50% faster load times through backend optimization and query tuning
- Owned projects end-to-end: requirements gathering, API design, containerized deployment, and production releases

## Projects

### Anomalock | *Python, scikit-learn, FastAPI, PostgreSQL, Docker*

Live Demo

- Trained and evaluated Isolation Forest and Random Forest for rare-event anomaly detection on a chronological, leakage-safe split of 137,164 login events (33M+ event dataset); Isolation Forest reached 60.5% recall vs. 21.1% for Random Forest at a fixed 20% review budget
- Engineered a 5-feature pipeline for real-time risk scoring (impossible travel, brute-force patterns, device/location novelty), using identical feature logic in training and production to prevent train/serve skew
- Deployed a low-latency FastAPI inference service scoring events in real time against PostgreSQL-backed history and routing high-risk cases to a review queue; containerized with Docker, deployed via CI/CD

### VitaLink | *Django REST Framework, PostgreSQL, Gemini API (LLM tool-calling)*

TerraHacks 2025

- Built a Django REST Framework backend integrating the Gemini API with function/tool calling to triage patient-reported symptoms and recommend a matching hospital via a multi-factor scoring engine (distance, wait time, specialty)
- Implemented role-based access control across 3 tiers (patient, staff, admin) to restrict access to patient records and enforce data-privacy boundaries
- Designed the PostgreSQL schema for patient records and hospital metadata behind the recommendation engine

### Fake Out | *Node.js, TypeScript, Socket.IO, Docker*

- Built the real-time backend for a multiplayer party game, handling game state via Socket.IO with authoritative server-side logic to prevent cheating and desync across concurrent clients
- Sustained sub-100ms latency and 99.5% uptime under concurrent load across 1,000+ production sessions, using room-based event handling and heartbeat monitoring to recover dropped connections
- Containerized services with Docker and deployed via an automated CI/CD pipeline

## Technical Skills

**ML & Data:** scikit-learn, model evaluation & benchmarking, feature engineering, permutation importance, pandas, ETL/data pipelines

**Languages:** Python, SQL, TypeScript, JavaScript, Java, C/C++

**Backend & Serving:** FastAPI, Django REST Framework, Express.js, Node.js, REST API design, real-time/low-latency services

**Cloud & DevOps:** AWS (EC2, S3, Lambda), Docker, CI/CD (GitHub Actions), Git, PostgreSQL, MongoDB, SQLite

**Certifications:** Anthropic (Claude API, Claude on Vertex AI, Claude on Amazon Bedrock, MCP: Advanced Topics) | Udemy (Intro to Cloud Computing on AWS, The Complete Cyber Security Course)