

Jad Akil

AI Engineer & Backend Systems

Tripoli, Lebanon | jadakeel05@gmail.com | +961 81531154
github.com/jadAkeel | linkedin.com/in/jad-akil

PROFESSIONAL SUMMARY

AI-focused Computer Science graduate with hands-on experience building machine learning, deep learning, NLP, and full-stack AI systems.

Proficient in Python, TensorFlow, PyTorch, scikit-learn, FastAPI, Spring Boot, and data pipelines, with practical experience in model training, evaluation, and deployment-ready backend services.

Skilled in backend development, distributed systems, databases, Docker, and cloud fundamentals, with a strong interest in scalable AI products and real-world problem solving.

EDUCATION

Lebanese University (LU) — Bachelor's Degree in Computer Science

Lebanon | **Graduated: 2025**

Specialized in Python programming, Machine Learning, and Software Development

Strong foundation in algorithms, data structures, system design, and practical AI applications

Focused on data analysis, problem-solving, and building applied software systems

TECHNICAL SKILLS

Programming Languages: Python (AI/ML/Deep Learning), Java (OOP, Design Patterns, Android), C/C++ (Data Structures & Algorithms), JavaScript (MERN Stack).

AI & Machine Learning: scikit-learn, TensorFlow, PyTorch, Neural Networks, Data Preprocessing, NLP, RAG Systems, LangChain, Embeddings, Sentence Transformers, Decision Trees, Ensemble Methods, Hyperparameter Tuning (GridSearchCV), Class Imbalance Handling (SMOTE), Model Evaluation & Cross-Validation.

Backend & APIs: FastAPI, Node.js, Express.js, RESTful APIs, JWT, OAuth Authentication, Non-blocking I/O (asyncio).

Databases: MongoDB, Oracle, PostgreSQL, PostGIS, Vector Databases (Indexing, Query Optimization, Performance Tuning), Redis.

Distributed Systems & Big Data: Parallel Programming (OpenMP, MPI), Distributed Systems Fundamentals, Hadoop, Spark, ELT.

Cloud & Certifications: AWS Cloud Practitioner Certified (CLF-C02), Docker.

Tools: AWS, Git, GitHub, VS Code, Docker, Virtual Environments (venv).

PROJECT EXPERIENCE

Chess Engine Hybrid AI Platform github.com/jadAkeel/chessEngine

Tech Stack: Python, PyTorch, FastAPI, React, Neural Networks (Policy + Value Network)

- Architected a full-stack chess AI system combining a React frontend, FastAPI backend, and deep learning training pipeline for learning chess strategy from large game datasets.
- Designed a shard-based data pipeline to process millions of chess positions efficiently, enabling large-scale model training without exceeding memory constraints.
- Built a PyTorch policy + value neural network, inspired by AlphaZero-style architecture, to predict legal move probabilities and evaluate board positions.
- Engineered a streaming data loader that avoids full dataset materialization and improves training throughput and scalability.
- Designed a modular training framework with replay buffer, iterative training loops, validation tracking, and model checkpointing for reproducible experiments.
- Structured the codebase with clear data, model, and training modules, with extensibility toward self-play reinforcement learning and MCTS integration.

AI Recruiter Assistant github.com/jadAkeel/AI-recruiter

Tech Stack: Python, FastAPI, SQLAlchemy, PostgreSQL, NLP Embeddings, Docker, Redis, OpenAI API, Ollama

- Built a production-grade FastAPI backend with async endpoints, SQLAlchemy ORM, JWT authentication, and documented REST APIs for recruitment automation.

- Developed a CV parsing engine to extract structured skills, experience, and education data from raw PDF resumes using NLP techniques.
- Implemented AI-powered candidate-to-job matching with sentence embeddings and cosine similarity to rank candidates against job requirements.
- Created a bilingual Arabic/English AI interview agent that conducts text-based interviews and produces structured candidate evaluation scores.
- Integrated Ollama for local LLM inference and the OpenAI API with automatic fallback handling to improve reliability under model unavailability.
- Containerized development and production configurations with Docker and used Redis for caching and query performance optimization.

Disease Risk Prediction github.com/jadAkeel/humanRisk_MLPredection

Tech Stack: Python, scikit-learn, pandas, NumPy, Decision Tree, GridSearchCV, SMOTE

- Developed a supervised learning pipeline to predict 10-year coronary heart disease (CHD) risk from the Framingham Heart Study dataset using a Decision Tree classifier.
- Applied data preprocessing including missing-value imputation, feature scaling, and feature importance analysis; identified blood pressure, age, and smoking as top predictors.
- Addressed class imbalance by evaluating SMOTE, class weighting, and downsampling, selecting the approach that maximized recall for the at-risk class.
- Optimized model performance with GridSearchCV, 5-fold cross-validation, and post-pruning, achieving ~70% held-out accuracy and 80% synthetic patient testing accuracy.

SmartOrder — Event-Driven Order & Delivery System github.com/jadAkeel/smartOrder-microservices

Tech Stack: Java 17, Spring Boot, Kafka, Spring Cloud Gateway, Eureka, PostgreSQL, MongoDB, Redis, Docker

- Developed 10 Spring Boot microservices using Java 17 to support order, inventory, payment, delivery.
- Designed Kafka-based event streaming and a choreography-style Saga pattern for distributed order processing and compensating transactions.
- Integrated Spring Cloud Gateway and Eureka for API routing, and cleaner service-to-service communication.
- Integrated PostgreSQL, MongoDB, and Redis to support polyglot persistence, caching concepts, and scalable .
- Containerized the platform with Docker Compose, including Kafka, PostgreSQL, MongoDB, Redis, and backend .
- Improved service resilience using Resilience4j patterns such as circuit breaker, retry, and timeout handling.

Multi Agent system MCP Bridge github.com/jadAkeel/Ai_Agent

Tech Stack: Node.js, JavaScript, MCP Protocol, OpenCode CLI, SQLite, Git Worktrees

- Built an MCP bridge server in Node.js that orchestrates OpenCode CLI agents (planner, builder, reviewer, debugger, architect) through a Model Context Protocol interface, with Codex as the primary orchestrator.
- Designed a SQLite-backed temporary write-lock registry with WAL mode and token-based ownership to safely coordinate parallel write agents across non-overlapping path scopes.
- Implemented Git worktree integration for isolated branch-level execution, with rollback on changed-file validation failures to protect the main working tree from unintended edits.
- Engineered a job queue (memory and SQLite modes) supporting parallel read-only and serialized write jobs, with configurable write-conflict policies, retry logic, and status tracking via dedicated MCP tools.
- Developed a compact delegation packet format and preflight validation tool to safely plan parallel OpenCode jobs, including overlap detection, serial-only path guards, and per-agent timeout enforcement.

CityMind Tripoli — AI Urban Digital Twin github.com/jadAkeel/mind_city

Tech Stack: Next.js, TypeScript, Tailwind, FastAPI, Python, PostgreSQL/PostGIS, Redis, Docker, MapLibre

- Architected a full-stack monorepo (Next.js frontend, FastAPI backend, shared packages) serving as a municipal intelligence and urban digital twin prototype for Tripoli, Lebanon, built with Turborepo and pnpm workspaces.
- Integrated MapLibre with OpenStreetMap and PostGIS-backed geospatial models to render seed-populated report, asset, and neighborhood layers on a live interactive Tripoli city map.
- Implemented Green Score and Walkability formula endpoints with explainability breakdowns, City Pulse dashboard aggregation, rule-based Smart Mobility and Urban Scenario Simulator predictions, and a grounded Ask Your City AI assistant.
- Set up SQLAlchemy ORM with Alembic migrations on PostgreSQL/PostGIS, Tripoli-specific demo seed data loaded via a seed pipeline, and full Docker Compose orchestration covering the database, API, and Redis layers.
- Delivered readiness interfaces for 3D digital twin (Cesium-ready), microservices extraction, and advanced SUMO/MATSim simulation adapters, with a Makefile validation bundle covering lint, typecheck, pytest, and frontend smoke tests.