LIVESH L

AI & XR Developer | Unity VR/XR Simulations | LLM Applications (RAG) & AI Security | Python, FastAPI, C#

liveshjaidj@gmail.com | 9345406612 | Tamil Nadu, India

linkedin.com/in/livesh-l-aa9821322 | github.com/Livesh-L-28 | myportfolio-kappa-plum.vercel.app

PROFESSIONAL SUMMARY

Final-year **Artificial Intelligence & Data Science** student (CGPA 8.53/10, no backlogs) building at the intersection of **AI** and **XR**: **Unity/C#** VR training simulations with a **16-step guided workflow** and **FastAPI** backend integration, plus an XR module-management desktop client (**React, TypeScript, Electron, Unity Addressables, AWS S3**), alongside production **LLM applications** spanning **RAG, CAG, MAG, guardrails**, and **evaluation/observability**, and **Machine Learning** and **Computer Vision** systems. Built **NexaRAG**, a multi-tenant RAG platform on PostgreSQL/pgvector (~24.5 ms average retrieval latency, 321 passing tests), and **LLMFirewall Core**, an open-source LLM security framework (v1.0.0, 608 passing tests). Published **aireliability** (LLM reliability & regression testing) and **aegis-resilience** (AI incident investigation and controlled remediation) to **PyPI**. Experienced with **Agile/Scrum, Git,** and **CI/CD** from internships. Seeking an **AI / XR Developer** role.

TECHNICAL SKILLS

XR / VR & Unity: Unity, C#, VR Interaction Framework, VR Training Simulations, Unity Addressables, Electron XR Desktop Clients, Backend Integration (FastAPI, REST APIs)
AI & Machine Learning: Machine Learning, Deep Learning, Computer Vision, OpenCV, YOLO, TensorFlow, PyTorch, Scikit-learn, NLP
AI Forward Engineering: LLMs, RAG, CAG, MAG, Context Engineering, Hybrid Retrieval (BM25 + Vector, RRF), Vector Search (pgvector, FAISS), Cross-Encoder Reranking, Observability (Prometheus, OpenTelemetry, Loki)
AI Security: LLM Guardrails (NVIDIA NeMo), Prompt-Injection Defense, Secret/PII Protection, Threat Modeling, Multi-Tenant Isolation, RBAC, Policy-Based Authorization
Programming Languages: Python, Java, SQL, C#, TypeScript
Backend & APIs: FastAPI, Flask, REST APIs, SSE Streaming, JWT, Microservices, Unit Testing, API Design
Frontend: React, TypeScript, Electron
Databases: PostgreSQL (pgvector), MySQL, SQLite
Cloud & Deployment: AWS S3, Docker, Docker Compose, Kubernetes, Firebase, Render, CI/CD, GitHub Actions
Developer Tools: Git, GitHub, Postman, VS Code, Jupyter Notebook, Linux
Core CS & Practices: Data Structures & Algorithms, OOP, DBMS, Operating Systems, Computer Networks, Agile, Scrum, System Design

EXPERIENCE

XR Developer — XARC *May 2026 – Aug 2026*

- Engineered the **XARC Nexus Aggregator Desktop Client**, an enterprise application for managing and launching XR training modules, using **React, TypeScript,** and **Electron**, with **Unity Addressables** for on-demand module loading.
- Configured **AWS S3** for remote asset hosting and version management; delivered module management, download tracking, and update-detection workflows via **Git** and **Agile/Scrum** practices.

AI Intern — Chargebee *Feb 2026 – Aug 2026*

- Delivered **Leviathan**, an AI-powered sustainability and **anti-greenwashing** audit platform, architecting a **scikit-learn ensemble** (Random Forest + Gradient Boosting + Ridge Regression) that predicts carbon footprint and durability with an **R2 of 0.88**.
- Developed a **Greenwashing Inspector** using **TF-IDF + Gemini LLM** reasoning to audit marketing claims against **ISO 14021** standards, and integrated real-time product search across **Open Food Facts** and **Google Shopping** APIs.

Machine Learning Intern — Femtosoft Technologies *Jul 2025 – Aug 2025*

- Applied **supervised and unsupervised Machine Learning** algorithms in Python across the full ML workflow (data collection, preprocessing, EDA, feature engineering, model training, and evaluation), achieving strong model accuracy across multiple datasets.
- Strengthened analytical and problem-solving skills while following industry-standard ML development practices.

Power BI Intern — Cognifyz Technologies *Sep 2024 – Oct 2024*

- Built interactive **Power BI** dashboards and reports to visualize business data for stakeholder reporting, significantly cutting manual reporting time; performed data cleaning, transformation, and preprocessing with **Power Query** across multiple data sources.
- Created **DAX** measures and calculated columns for KPI analysis; configured dashboards with charts, slicers, filters, and drill-through features.

PROJECTS

VR-Based Venipuncture Training System — Unity

Unity, C#, VR Interaction Framework, Python, FastAPI, REST APIs

Demo Video

- Developed an immersive **VR clinical training environment** in **Unity** where learners practise venipuncture through a structured **16-step guided workflow** that walks them through each stage of the procedure in the correct sequence.
- Built interactive virtual medical equipment and object manipulation using **C#**, giving learners a repeatable, hands-on practice experience without needing physical training materials for every session.
- Structured the training experience around individual procedural stages, helping learners understand the sequence of actions involved in venipuncture and navigate the simulation systematically.

AI-Enhanced VR Venipuncture Training & Medical AI System

Unity, C#, Python, PyTorch, FastAPI, PostgreSQL, pgvector, BM25, Whisper, REST APIs, LLMs, RAG, PDF Generation

Demo

- **VR application:** Built an immersive medical training environment in **Unity/C#** with a structured **16-step venipuncture workflow**, interactive medical equipment, procedural guidance, and separate **Training and Test modes**.
- **Custom medical LLM:** Built a **110M-parameter MedicalTransformerLM** from scratch in **PyTorch**, integrated into a multi-model architecture with **OpenBioLLM-8B** and **Llama 3.2 3B** for medical question answering and educational assistance.
- **Medical RAG pipeline:** Developed document ingestion, text extraction, chunking, **BM25** keyword retrieval, embeddings, and **PostgreSQL/pgvector** similarity search.
- **Voice assistant & backend:** Integrated Unity with a **FastAPI** (Uvicorn) backend for voice-based queries using **Whisper** speech-to-text, intent routing, LLM inference, and text-to-speech; REST endpoints cover medical Q&A, speech processing, document ingestion, AI inference, and health monitoring.
- **Grounding & validation:** Implemented medical-answer grounding checks and deterministic procedural validation to reduce unsupported AI responses and enforce valid progression through the workflow.

- **Assessment & integration:** Connected training data to an analysis pipeline producing performance summaries, AI-assisted educational feedback, and **downloadable PDF reports**, unifying the VR app, model-routing layer, retrieval system, voice services, and reporting into one learning platform.

NexaRAG — Enterprise Document Intelligence & RAG Platform

Python, FastAPI, PostgreSQL, pgvector, React, TypeScript, Docker, BM25, Cross-Encoder, NeMo Guardrails
github.com/Livesh-L-28/Nexa_RAG | *Demo Video*

- Architected a production-oriented **RAG platform** with **PostgreSQL/pgvector, BM25 hybrid retrieval, Reciprocal Rank Fusion,** and MS-MARCO **cross-encoder reranking**, achieving **~24.5 ms** average retrieval latency.
- Engineered unified **RAG**, **Cache-Augmented Generation (CAG)**, and **Memory-Augmented Generation (MAG)** with **6-route dynamic query routing, 7-tier context fusion,** and a 6,000-token dynamic context budget.
- Built a full-stack **React 18 + TypeScript + FastAPI** application with **SSE streaming**, document ingestion for PDF/DOCX/TXT, page-accurate citations, retrieval inspection, memory/cache management, analytics, and audit logs.
- Implemented **multi-tenant security** and **NVIDIA NeMo Guardrails** with JWT authentication, RBAC, rate limiting, prompt-injection defenses, secret/PII protection, and **16 verified security isolation invariants**.
- Containerized the complete application using **Docker Compose**, separating React/Nginx, FastAPI, and PostgreSQL/pgvector services for reproducible development and deployment; validated with **321 automated tests (100% pass rate)** and production frontend builds.

MedicalTransformerLM — Domain-Specific Medical LLM & RAG System

PyTorch, Transformers, OpenBioLLM, FAISS, BM25, RAG, Apple MPS
github.com/Livesh-L-28/MEDICAL_LLM | *Demo Video*

- Built a domain-specific medical AI system combining a lightweight **~110M-parameter MedicalTransformerLM**, **OpenBioLLM**, hybrid **BM25 + FAISS** retrieval, medical acronym expansion, and LLM synthesis grounded in retrieved medical knowledge.

LLMFirewall Core — Open-Source LLM Security Framework

Python, AI Security, FastAPI, RAG, Security Engineering
github.com/Livesh-L-28/LLMFirewall | *PyPI*

- Developed an open-source **LLM security framework** to detect and mitigate **prompt injection, secrets leakage, PII exposure,** and **unsafe tool calls**, via a modular pipeline of detection, risk assessment, policy evaluation, redaction, runtime enforcement, audit logging, and incident response.
- Implemented security controls for LLM/RAG applications, AI agents, tool calling, memory, and document ingestion; built extensible components for asset inventory, attack graphs, threat modeling, compliance, supply-chain validation, and security evaluation.
- Developed an automated validation suite with **608 tests**, regression and integration testing, security benchmarks, dependency auditing, and clean-install verification.
- Packaged as a Python distribution with CLI tooling, FastAPI/ASGI integration, documentation, and API reference; released **v1.0.0** with repository governance, release documentation, a security audit, and PyPI-ready packaging.

AI Reliability Engine (aireliability) — Open-Source LLM Reliability & Regression Testing Framework

Python, Pydantic, Pytest, Ruff, GitHub Actions, PyPI
pip install aireliability | *github.com/Livesh-L-28/aireliability_testing* | *PyPI*

- Built a deterministic **LLM/AI-agent reliability testing framework** with execution tracing, tool-call assertions, failure taxonomy, and regression detection.
- Implemented validation for tool ordering, arguments, outputs, schemas, latency, and cost, enabling reproducible AI workflow testing.
- Developed the **airel** CLI, automated CI/CD with **362 passing tests**, and published to **PyPI**; independently validated on another machine with reproducible package and CLI execution.

AEGIS Framework — AI Reliability & Controlled Autonomous Remediation

Python, FastAPI, Pydantic, Prometheus, OpenTelemetry, Loki, Kubernetes, Docker, GitHub Actions
github.com/Livesh-L-28/AEGIS-Framework | *PyPI*

- Designed and developed a modular, production-oriented Python framework for **AI-assisted incident investigation**, evidence correlation, root-cause reasoning, and **controlled infrastructure remediation**.
- Integrated observability and infrastructure providers for **Prometheus** metrics, **Loki** logs, **OpenTelemetry** traces, GitHub deployments, and **Kubernetes** workload management.
- Implemented **fail-closed security controls**, policy-based action authorization, bounded autonomy, approval gates, dry-run remediation, and cryptographic audit trails.
- Built a developer CLI with initialization, environment diagnostics, policy validation, and runnable demonstrations; added configuration-driven setup and AI-coding-agent integration documentation.
- Validated the end-to-end incident lifecycle using real infrastructure integrations, including evidence collection, controlled Kubernetes scaling, and recovery verification.
- Published the reusable framework on GitHub and PyPI as **aegis-resilience**, with automated quality checks, packaging validation, and developer documentation.

Stock Price Prediction — Machine Learning Project *(Apr 2025 – May 2025)*

Python, Pandas, NumPy, Scikit-learn, TensorFlow/Keras, Matplotlib
github.com/Livesh-L-28/Stock_price_prediction | *Live Demo*

- Trained time-series forecasting models on historical market data with **Scikit-learn** and **TensorFlow/Keras**; optimized a **2-layer Stacked LSTM** with dropout, maintaining a RMSE/MAE ratio of 1.25 and **under 4% MAPE** across the top 100 blue-chip stocks.

AI Visual Memory Navigation System — Personal Project

OpenCV, YOLOv11, ByteTrack, CLIP, FAISS, FastAPI, React, Docker
github.com/Livesh-L-28/AI-Visual-Memory-System | *Live Demo*

- Constructed an AI-powered **Computer Vision** system for real-time object detection, tracking, and re-identification using **OpenCV, YOLOv11, CLIP** embeddings, and **FAISS** (~30–50 ms per frame), with FastAPI services, a React dashboard for live monitoring and semantic search, WebSockets, SQLite, and Docker.

EDUCATION

B.Tech, Artificial Intelligence and Data Science *08/2023 – Present (Expected 2027)*
Prathyusha Engineering College, Thiruvallur, Tamil Nadu | Aggregate CGPA: 8.53/10 | No active backlogs

CERTIFICATIONS

- OpenCV Bootcamp — OpenCV University (Grade: 100%) | Apr 2026
- Joy of Computing Using Python — NPTEL, IIT Madras (Elite 77%) | 2024
- Cloud Computing — NPTEL | 2025
- Databases and SQL for Data Science — IBM (Coursera) | Sep 2024
- Intro to Machine Learning — NPTEL | 2025
- Power BI Micro Course — Learn More Pro | 2024