

Abd Alrhman Deyab

Dallas, Texas

[LinkedIn](#) | abd.k.deyab@gmail.com | [GitHub](#) | [Portfolio](#)

SUMMARY

Computer Engineering student focused on applied machine learning and full-stack development. Treats correctness as a measurement problem — benchmarking retrieval quality, profiling inference cost, and tracing failures to root cause rather than to the nearest plausible explanation. Seeking junior engineering roles where applied machine learning meets production systems.

EDUCATION

Istanbul Medipol University | *Istanbul, Turkey* | *September 2024 – July 2028*

Bachelor's Degree, Computer Engineering

WORK EXPERIENCE

MAG7 Solutions — ML Dev Intern | *Giza, Egypt (Hybrid)* | *July 2026 – August 2026*

- Contributed to the end-to-end development of a web application using machine learning and image-based product recognition, spanning the embedding pipeline, model integration, and API delivery.
- Designed, built, and deployed three full-stack AI systems (detailed below) covering multimodal semantic search, vision-language chat, and retrieval-augmented generation over user-uploaded documents.
- Containerized multi-service stacks with Docker Compose and Docker Swarm — multi-stage builds, service networking, and persistent volumes — for reproducible development-to-production environments.
- Profiled and optimized locally hosted model inference (GPU offloading, OCR concurrency, model selection), cutting cold model load ~7.5× and scanned-document ingestion from ~11.5 minutes to ~23 seconds.

PROJECTS

Study LLM — Notebook RAG Chat | *Spring Boot, React, PostgreSQL/pgvector, Ollama, Docker*

- Built a document-grounded RAG platform where users organize course materials into notebooks and receive answers cited to specific source chunks, with an explicit "not found in sources" response to suppress hallucination.
- Engineered an asynchronous ingestion pipeline (extract → chunk → embed → persist) with layered OCR — Tesseract first, vision-model fallback — and a symbol-repair pass restoring set-theory notation that OCR silently corrupts.
- Raised retrieval recall by widening each vector hit to adjacent chunks, after measuring that content spanning a page break ranked #18 rather than #2 and left answers covering only half a section.
- Verified with JUnit and Testcontainers integration tests; shipped with full architectural, technical, and user documentation.

Flower-VLM — Vision-Language Identification Chat | *Node.js/Express, React, Qdrant, CLIP, Ollama, Docker*

- Built a chat application answering flower queries from text or uploaded photos by embedding both into a shared CLIP vector space and searching a Qdrant collection of 7,390 indexed images.
- Grounded model output by reordering the pipeline to run vector search before generation, so descriptions explain the matched species rather than guessing blind; de-duplicated results by species to prevent crowding.
- Root-caused a relevance failure returning an identical result for every query to all-zero embedding vectors making cosine similarity degenerate, then rebuilt the index to restore ranking.

Flower Semantic Search Engine | *Python, FastAPI, React, CLIP, Qdrant/FAISS, Docker*

- Indexed ~7,370 Oxford-102 flower images as 512-dimensional L2-normalized CLIP embeddings, searchable by natural-language description or by example image.
- Improved short-query accuracy through taxonomy-aware query expansion and blended-score reranking; built a pluggable vector-store interface supporting either Qdrant (HNSW) or local FAISS.
- Secured the API with JWT bearer tokens and bcrypt password hashing over a SQLAlchemy/SQLite user store.

Social Media Platform for Medical Articles | *Python, Django, SQLite*

- Built a server-rendered platform with session-based authentication, Django ORM models, and a normalized SQLite schema.

SKILLS

Languages: Python, Java, JavaScript, SQL, HTML/CSS, MATLAB

Backend & Frontend: FastAPI, Spring Boot, Django, Node.js/Express, React, Vite, REST API design, JWT & bcrypt authentication

Machine Learning & AI: CLIP, embeddings, semantic/vector search, RAG, LLM & VLM integration (Ollama), Hugging Face Transformers, OCR (Tesseract), data chunking strategies, inference optimization (quantization, distillation, batching, caching)

Data & Storage: PostgreSQL/pgvector, Qdrant, FAISS, SQLite, SQLAlchemy, Excel analysis

DevOps & Tools: Docker, Docker Compose, Docker Swarm, Git/GitHub, Linux/SSH, Maven, JUnit, Testcontainers