

# Marlon Munoz

New York, NY · 646-873-0251

[Github](#) | [Blog](#) | [LinkedIn](#) | [marlon@marlonshade.com](mailto:marlon@marlonshade.com)

## AI ENGINEER | FULL-STACK DEVELOPER

Results-driven AI Engineer with a strong foundation in full-stack development and a focus on data-driven, AI-powered solutions. Experienced in building end-to-end machine learning pipelines, retrieval-augmented generation systems, and NLP applications. Proficient in transforming data into actionable insights and integrating intelligent features into production-ready applications using modern AI frameworks and cloud infrastructure.

## TECHNICAL SKILLS

**AI & Machine Learning:** TensorFlow, Keras, Deep Learning, NLP, Transfer Learning, Transformer Models (RoBERTa), Prompt Engineering, RAG Architecture, Vector Embeddings, LangChain, LangGraph

**Data Engineering:** Python, Pandas, NumPy, PostgreSQL, Data Cleaning & Transformation, Data Validation, ETL Pipelines, Data Modeling, Dimensional Modeling

**Languages & Frameworks:** Python, JavaScript (ES6+), React, Flask, FastAPI, SQL, HTML/CSS3, Tailwind, Bootstrap

**AI Infrastructure & Tools:** AWS Bedrock, Claude LLM, ChromaDB (Vector Databases), Hugging Face Transformers, boto3

**Databases & Storage:** PostgreSQL, SQLite, SQLAlchemy, ChromaDB, AWS S3

**Developer Tools & DevOps:** Git, GitHub, Docker, CI/CD, GitHub Actions, Postman, VSCode, Chrome DevTools, Pytest

**Concepts:** RESTful APIs, OOP, Agile/Scrum, MVC Architecture, Microservices, Clean Architecture, Model Training & Evaluation

## PROFESSIONAL EXPERIENCE

### Revature — AI Engineering / Data Engineering

Nov 2024 – Jan 2026

#### Data Ingestion Sub System - [Github](#)

*Python-based ETL orchestration system with automated workflows for multi-source data ingestion (CSV, JSON, REST APIs) into PostgreSQL staging tables with comprehensive validation, deduplication, and audit logging.*

- Built ETL pipeline extracting living wage data via web scraping and US Census Bureau API
- Designed PostgreSQL schema with staging tables, reject tracking, and Flyway migrations
- Implemented caching, retry logic with exponential backoff, and rate limiting for reliable extraction
- Transformed wide-format HTML tables to normalized records using Pandas and Pydantic validation
- Containerized PostgreSQL with Docker and set up CI/CD pipeline with GitHub Actions

**Environment:** Python, PostgreSQL, Pandas, Docker, Git, Agile-Scrum, Pytest, Data Modeling

#### Sentiment Classification API for Customer Reviews - [Github](#)

*AI-powered NLP system performing multi-class sentiment classification using transformer-based architectures with end-to-end ML workflows including preprocessing, fine-tuning, evaluation, and production API deployment.*

- Fine-tuned transformer models (RoBERTa-based) for sentiment classification on e-commerce review data
- Implemented text preprocessing, tokenization, and label mapping pipelines for supervised learning
- Evaluated model performance using accuracy and loss metrics, iterating on training strategies
- Exposed trained models through FastAPI service for real-time sentiment inference
- Containerized application using Docker for consistent deployment workflows

**Environment:** Python, TensorFlow, Hugging Face Transformers, FastAPI, Docker, Git, NLP

#### Cloud-Based Retrieval-Augmented Generation (RAG) System - [Github](#)

*Comprehensive RAG pipeline leveraging AWS Bedrock and vector database technology for semantic search and contextually-grounded LLM responses with cloud-native integration patterns.*

- Designed multi-phase RAG pipeline covering ingestion, embedding, retrieval, and generation
- Performed semantic chunking on source text to improve embedding quality and retrieval relevance
- Generated embeddings and stored them in ChromaDB vector store configured for cloud usage
- Integrated AWS Bedrock using Claude model for response generation grounded in retrieved context
- Implemented retrieval logic combining vector search with prompt construction for LLM inference
- Managed cloud configuration and authentication using AWS SDK (boto3)

**Environment:** Python, AWS Bedrock, Claude LLM, ChromaDB, LangChain, Vector Embeddings, boto3, Git

#### Vector-Based RAG API Platform

*Production-grade backend AI infrastructure implementing RAG API for natural language querying over domain-specific knowledge bases through semantic search and generative AI with clean architecture principles.*

- Designed modular RAG backend using FastAPI with separated ingestion, retrieval, and query layers
- Integrated ChromaDB as vector store for embedding persistence and semantic search
- Implemented document ingestion pipelines with configurable chunking and metadata handling
- Built query workflows combining vector similarity search with prompt construction for grounded LLM responses
- Integrated AWS Bedrock as LLM provider with client abstraction for model interaction
- Applied structured configuration management using environment-based settings and validation
- Containerized application with Docker for consistent deployment

**Environment:** Python, FastAPI, ChromaDB, AWS Bedrock, LangChain, Docker, Git, RAG Architecture

## TECHNICAL PROJECTS

### University of Richmond Chatbot — RAG-Powered Policy Q&A System - [Github](#)

- Production-grade RAG chatbot for University of Richmond policy queries with ChromaDB cloud vector storage and OpenAI embeddings
- Implemented persistent conversation memory using LangGraph with PostgreSQL for stateful multi-turn dialogues
- Built modular FastAPI backend with React/Tailwind frontend featuring dark/light mode and citation-grounded responses
- Tech Stack: FastAPI, LangGraph, ChromaDB Cloud, OpenAI API, PostgreSQL, React, Tailwind CSS

### X-PENSE — AI-Powered Budget Tracking App - [Github](#) | [Demo](#) | [Open](#)

- Full-stack financial tracking tool with interactive AI Assistant using NLP for natural language financial data queries
- Built with React and Flask; enhanced cross-device usability with Bootstrap (50% mobile performance improvement)
- Employed RESTful architecture and clean component design for maintainability

### IRYNK-App — Educational Toy Recommendation Platform - [Github](#) | [Demo](#)

- Personalized toy suggestion tool with custom Flask API and SQLite persistent storage
- Utilized React hooks and conditional rendering for intuitive UX; resolved CORS issues for robust API communication

### Fox Runner — JavaScript-Based Retro Platform Game - [Github](#) | [Demo](#) | [Play](#)

- Interactive 2D game using JavaScript OOP and game physics with collision detection, scrolling, and level design

## OTHER EXPERIENCE

### Motion Graphic Designer / Video Editor — Tommy Boy Entertainment, New York, NY

Jun 2015 – May 2017 · [tommyboy.com](http://tommyboy.com)

- Created compelling visual content using Adobe After Effects and Premier Pro
- Collaborated cross-functionally with artists and marketing teams to meet branding goals
- Produced videos generating 200K+ views across social and music platforms

### Caseworker (Temporary Role) — HANAC, New York, NY

Jun 2018 – Aug 2018 · [hanac.org](http://hanac.org)

- Provided direct support for senior citizens on housing applications and public service access
- Improved intake efficiency by 20% by streamlining internal documentation workflows
- 

## EDUCATION & CERTIFICATIONS

### Flatiron School, New York, NY

Certificate in *Full Stack Software Engineer (Python & JavaScript)* —

Jul 2024 – Oct 2024

### Baruch College (CUNY), New York, NY

*Bachelor of Arts in Management of Music Enterprises, Minor: Photography* —

Aug 2010 – Oct 2015

### LaGuardia Community College, New York, NY

*Associate of Science in Fine Arts* —

Sep 2005 – Jun 2008

## REFERENCES

Available upon request