

ARUN JYOTI CHAKRABORTY

Applied AI Engineer · RAG & Agentic AI Developer · IIT Guwahati

West Bengal, India

github.com/Arun660248

linkedin.com/in/arun-jyoti-chakraborty

arunjyoticode.me

arunjyotichakraborty18@gmail.com

PROFESSIONAL SUMMARY

Applied AI Engineer with production experience deploying multi-agent systems, RAG pipelines, and quantitative research agents on AWS and Google Cloud. Specialized in low-latency LLM orchestration (MCP/FastMCP, Gemini, Bedrock), agentic workflow design, and robust guardrail implementation. Built 4 live AI systems with custom domains, automated safety testing, and real-time analytics. Pursuing B.Sc. Data Science & AI at IIT Guwahati (online, self-paced) — fully available for remote full-time or internship roles immediately. Certified by Microsoft (Azure) and AWS (Bedrock).

TECHNICAL SKILLS

AI & LLMs: LangChain · Google ADK · Gemini API · MCP/FastMCP · RAG Pipelines · FAISS · ARIMA Forecasting · Agentic Workflows · Prompt Chaining

Backend: FastAPI · Python (Advanced) · SQL · Bash · REST APIs

Databases: SQLite · FAISS Vector Store · Vector Embeddings · PostgreSQL

DevOps & Cloud: Docker · AWS (EC2, S3, Bedrock, PartyRock) · Google Cloud · Cloudflare · Linux/WSL · Git/GitHub

Tools: Streamlit · Google AI Studio · n8n · pytest · pandas · NumPy

ENGINEERING PROJECTS

Algorithmic Trading & Predictive Research Agent | Python · Google ADK · Gemini 1.5 Pro · FastMCP · Docker · AWS EC2

- Architected a decoupled multi-agent system: a **Gemini 1.5 Pro orchestrator** (Google ADK) coordinates research while a separate **FastMCP tool server** executes live market data retrieval and pmdarima ARIMA forecasting over HTTP/SSE — eliminating tight coupling between agent logic and analytical tools.
- Implemented persistent session memory via ADK's **SqliteSessionService** and an automated **pytest guardrail suite** verifying refusal of financial advice and tool-call blocking under adversarial prompt-injection attacks.
- Backtested ARIMA module against 30 days of live market data achieving **~5% MAPE** on 1-day-ahead price forecasts under normal volatility; documented degradation during news-driven shocks motivates the sentiment-aware tool layer in the architecture.
- Containerized with Docker Compose, deployed on AWS EC2 behind Cloudflare with SSL. GitHub: [Algorithmic-Trading-Predictive-Research-Agent](#). Capstone, Google-Kaggle 5-Day AI Agents Intensive.

AI Financial Analyst Agent | Python · LangChain · FastAPI · Docker · AWS EC2

- Built an autonomous financial research agent using LangChain **@tool decorators** to dynamically fetch real-time stock prices, historical OHLCV data, and income statements via yfinance API — replacing static training-weight reliance with live data tool-calling on every query.
- Engineered a **ReAct-pattern agent loop** that plans multi-step research sequences (price fetch → trend analysis → income comparison), generates time-series visualizations via headless matplotlib, and returns structured financial summaries with source attribution.
- Containerized with FastAPI backend + Streamlit UI, deployed on AWS EC2 behind Cloudflare with SSL — live at finai.arunjyoticode.me. GitHub: [ai_financial_analyst](#).

Enterprise Document RAG System | Python · LangChain · FAISS · Gemini 2.5-flash · FastAPI · AWS EC2

- Engineered a production RAG pipeline: PDFs are chunked via **RecursiveCharacterTextSplitter**, embedded into a **FAISS vector store**, and retrieved semantically at query time — with automatic source-and-page citations on every response to eliminate hallucinations and support auditability.
- Pre-loaded a sample financial report enabling **zero-friction recruiter demos** without requiring document upload; Gemini 2.5-flash selected for its 1M-token context window to handle large enterprise document corpora without chunking loss.
- Deployed on AWS EC2 behind Cloudflare with SSL and Streamlit interface — live at rag.arunjyoticode.me. GitHub: [Enterprise-RAG-System](#).

Cognitive Load Balancer | Amazon Bedrock · AWS PartyRock · Multi-Prompt Chaining

- Designed a **multi-step prompt-chaining** pipeline on Amazon Bedrock functioning as a ruthless prioritization engine — calculates task ROI scores, enforces execution limits, and eliminates decision paralysis through structured constraint logic. Live on AWS PartyRock; code: [cognitive-load-balancer-ai](#).

CERTIFICATIONS & PROGRAMS

Microsoft Azure — 25hr (Microsoft Elevate × AICTE)

Feb–Mar 2026 · 4-week internship, Emerging Tech

Cloud Admin & Engineering — 40hr (Microsoft Elevate × AICTE)

Feb–Mar 2026 · Same cohort

Google 5-Day AI Agents Intensive (Kaggle)

Multi-agent systems · MCP · Capstone submitted

AWS AI & ML Scholars Program (Udacity)

Challenge Phase selected · Bedrock · 2026

EDUCATION

B.Sc. Data Science & Artificial Intelligence · Indian Institute of Technology Guwahati (IITG) ·

Online

Coursework: Recommender Systems · Machine Learning · Time Series Analysis · Data Modeling & Visualization

2024 – Present (2nd Year)