

Anoop Pant

Senior Gen AI Engineer | Full Stack Python Developer

Email id: anooppant999@gmail.com | contact no.: 4698460983 | [LinkedIn](#) | [Portfolio](#)

PROFESSIONAL SUMMARY

Senior Generative AI and Python Engineer with 7+ years of experience building production-ready AI/ML, full-stack, and data-intensive applications across banking, financial services, healthcare, and market-data environments. Hands-on expertise in RAG, multi-agent workflows, LLM orchestration, semantic search, prompt engineering, model evaluation, and responsible AI using Python, Amazon Bedrock, Azure OpenAI, LangChain, LangGraph, and FastAPI. Experienced in developing secure microservices, document-ingestion pipelines, and cloud-native solutions on AWS and Azure using Docker, Kubernetes, CI/CD, and modern observability practices. Proven ability to translate complex business workflows into reliable, human-reviewed AI solutions while collaborating with product, data, platform, security, risk, and domain teams.

TECHNICAL SKILLS

- **Programming & Web Technologies:** Python 3.x, SQL, PySpark, JavaScript, TypeScript, HTML5, CSS3, Object-Oriented Programming
- **Generative and Agentic AI:** Large Language Models, Retrieval-Augmented Generation, LangChain, LangGraph, Multi-Agent Workflows, Tool Calling, Prompt Engineering, Prompt Chaining, Contextual Grounding, Amazon Bedrock, Azure OpenAI Service, Hugging Face Transformers
- **Machine Learning & Evaluation:** Amazon SageMaker, Azure Machine Learning, scikit-learn, XGBoost, Intent Classification, Query Routing, Cross-Validation, Precision, Recall, F1-Score, Groundedness Evaluation, Hallucination Testing, Human-in-the-Loop Review
- **Backend Development:** FastAPI, Django, Django REST Framework, Flask, Pydantic, SQLAlchemy, RESTful APIs, Python Microservices, Swagger/OpenAPI
- **Data Engineering:** Pandas, NumPy, AWS Glue, Azure Data Factory, Azure Databricks, Apache Kafka, Apache Airflow, AWS Lambda, AWS Step Functions, Azure Functions
- **Databases, Search & Storage:** PostgreSQL, Amazon RDS, Azure Database for PostgreSQL, Amazon OpenSearch Service, Azure AI Search, Elasticsearch, Amazon S3, Azure Blob Storage, ADLS Gen2
- **AWS:** Amazon Bedrock, SageMaker, Glue, Lambda, S3, OpenSearch Service, ECS, Fargate, ECR, RDS, API Gateway, Step Functions, CloudWatch, IAM, KMS, Secrets Manager, AWS CDK, CloudFormation
- **Microsoft Azure:** Azure OpenAI Service, Azure Machine Learning, Azure AI Search, Azure AI Document Intelligence, Azure Data Factory, Azure Databricks, Azure Functions, AKS, ACR, Key Vault, Entra ID, Azure RBAC, Azure Monitor, Application Insights, Log Analytics
- **DevOps, Testing & Observability:** Docker, Kubernetes, Jenkins, Azure DevOps, Git, CI/CD, PyTest, unittest, Jest, OpenTelemetry, Structured Logging
- **Security & Responsible AI:** OAuth 2.0, JWT, Role-Based Access Control, Encryption, Data Masking, Private Endpoints, Audit Logging, Amazon Bedrock Guardrails, Azure AI Content Safety, Prompt-Injection Testing, Model-Risk Controls, HIPAA and GDPR

EDUCATION

Bachelor of Computer Science, East Texas A&M University - May 2021

Associate of Computer Science, Dallas College - May 2018

WORK EXPERIENCE

Capital One, Plano, TX

Dec 2024 – Present

Senior Generative AI Engineer

- Designed and developed a production-grade, associate-facing Generative AI chatbot and agent-assist platform using Python, Amazon Bedrock, LangChain, and LangGraph, helping servicing teams respond to customer questions involving card transactions, account services, disputes, and internal banking policies.
- Built supervised multi-agent workflows using LangGraph and tool-calling frameworks to coordinate intent classification, document retrieval, approved API calls, policy validation, and response generation, reducing average case-preparation time by approximately 25% while retaining human confirmation for account-affecting actions.
- Developed Retrieval-Augmented Generation (RAG) pipelines using Amazon S3, Amazon Titan Text Embeddings, and Amazon OpenSearch Service, implementing hybrid keyword and vector search, metadata filtering, reranking, and source citations that improved retrieval relevance by approximately 20% across curated evaluation queries.
- Built intent-classification and query-routing models using Hugging Face Transformers and scikit-learn, achieving a macro F1-score of 0.83 across transaction, account-servicing, dispute, and policy-related inquiries while routing low-confidence requests for clarification or human review.
- Engineered reusable prompt-orchestration components using prompt chaining, few-shot examples, contextual grounding, structured outputs, and response-validation rules, improving conversational consistency while reducing unsupported or incomplete responses.
- Created incremental document-ingestion and vector-indexing pipelines using Python, AWS Glue, PySpark, Amazon S3, AWS Lambda, and AWS Step Functions, automating document validation, chunking, metadata enrichment, version tracking, embedding generation, and index refreshes.
- Developed asynchronous Python microservices and RESTful APIs using FastAPI, Pydantic, and OpenAPI, implementing request validation, authentication, retries, timeouts, idempotency, and structured error handling for integration with approved banking and servicing systems.
- Established automated evaluation frameworks using curated test datasets and human feedback to measure retrieval recall, groundedness, citation correctness, answer completeness, tool-selection accuracy, hallucination risk, and response latency before production releases.
- Implemented responsible-AI and security controls using Amazon Bedrock Guardrails, AWS IAM, AWS KMS, Secrets Manager, private endpoints, sensitive-data redaction, prompt-injection testing, allowlisted tools, human-review checkpoints, and audit logging to meet privacy, model-risk, and regulatory requirements.

- Containerized services using Docker and deployed workloads through Amazon ECR, Amazon ECS with AWS Fargate, Amazon SageMaker endpoints, and AWS Lambda; automated infrastructure, testing, and releases using AWS CDK, Jenkins, Git, and PyTest while monitoring production performance through Amazon CloudWatch and OpenTelemetry.

PNC Bank, Farmers Branch, TX

June 2023 – Nov 2024

Python AI/ML Engineer

- Developed internal AI/ML applications using Python 3.10, FastAPI, Azure Machine Learning, and Microsoft Azure; incorporated Azure OpenAI Service and LangChain beginning in late 2023 for controlled enterprise search, document summarization, knowledge retrieval, and workflow-automation use cases.
- Built Retrieval-Augmented Generation (RAG) pipelines using Azure OpenAI embeddings and Azure AI Search, implementing hybrid keyword and vector search, semantic ranking, metadata filtering, and source citations that improved retrieval relevance by approximately 20% across evaluated queries.
- Created document-processing pipelines using Python, Azure Blob Storage, Azure Functions, and Azure AI Document Intelligence to extract, validate, chunk, embed, and index banking policies, operating procedures, technical runbooks, and support documentation, reducing manual research effort by approximately 25%.
- Developed asynchronous RESTful microservices using FastAPI, Pydantic, SQLAlchemy, and Azure Database for PostgreSQL, and integrated them with React.js and TypeScript interfaces for enterprise search, AI-generated summaries, user feedback, and secure role-based workflows.
- Trained and evaluated classification models using Azure Machine Learning, scikit-learn, and XGBoost to categorize and prioritize operational support cases; applied precision, recall, F1-score, cross-validation, and error analysis, reducing manual triage time by approximately 20%.
- Built scalable data pipelines using Azure Data Factory, Azure Databricks, PySpark, ADLS Gen2, and SQL to process operational logs, support records, user feedback, and model-evaluation datasets used by AI applications and reporting workflows.
- Containerized Python services using Docker, stored versioned images in Azure Container Registry, and deployed workloads on Azure Kubernetes Service; automated builds, testing, and releases using Azure DevOps, Git, and PyTest-based CI/CD pipelines.
- Implemented security controls using Microsoft Entra ID, Azure RBAC, managed identities, Azure Key Vault, private endpoints, encryption, sensitive-data masking, Azure AI Content Safety, and human-review workflows to support secure and responsible AI usage.
- Monitored API performance, model inference, data pipelines, and RAG quality using Azure Monitor, Application Insights, and Log Analytics while collaborating with AI, data, platform, DevOps, cybersecurity, risk, and product teams to deliver reliable banking solutions.

IQVIA, Durham, NC

Apr 2021 – May 2023

Full Stack Python Developer

- Developed full-stack healthcare applications using Python 3.x, Django, Django REST Framework, Flask, and object-oriented design principles, supporting real-world evidence, clinical research, and healthcare-data analytics.
- Built independently deployable Python microservices and RESTful APIs that allowed researchers to discover healthcare datasets, define patient cohorts, run study-feasibility queries, and retrieve de-identified analytical results; documented endpoints using Swagger/OpenAPI.
- Created responsive user interfaces using React.js, Redux, TypeScript, JavaScript, HTML5, and CSS3, enabling healthcare analysts to explore patient journeys, apply clinical and time-based filters, compare cohorts, and review data-quality results, reducing research setup time by approximately 25%.
- Developed data-validation and transformation services using Python, Pandas, NumPy, and SQL for medical claims, EHR, prescription, and laboratory data; standardized ICD-10, CPT, HCPCS, and NDC codes into OMOP-aligned structures, reducing manual corrections by approximately 20%.
- Created and maintained PostgreSQL data models using Django ORM and database migrations, and optimized SQL queries, Elasticsearch indexes, pagination, and caching, improving response times for frequently used APIs and dashboards by approximately 30%.
- Containerized applications using Docker and deployed Linux-based services on AWS using ECS, Lambda, S3, RDS, API Gateway, and CloudWatch, supporting asynchronous processing, centralized logging, application monitoring, and reliable production operations.
- Implemented OAuth 2.0, JWT authentication, role-based access control, encryption, data masking, and audit logging to protect de-identified healthcare information and support HIPAA, GDPR, and organizational privacy requirements.
- Wrote unit, integration, API, and frontend tests using PyTest, unittest, and Jest; supported Jenkins and Git-based CI/CD pipelines, participated in code reviews, and collaborated with product managers, QA engineers, data scientists, epidemiologists, and healthcare researchers in an Agile environment.

Bloomberg, New York, NY

May 2019 – Mar 2021

Python Developer

- Developed and supported backend applications using Python 3.x, Django, Flask, object-oriented programming, and reusable modules for financial-market data processing, analytics, and internal reporting.
- Built data-processing workflows using Python, Pandas, NumPy, and SQL to validate and transform more than 500,000 daily pricing and reference-data records across equities, fixed-income products, and foreign exchange, along with corporate-action data.
- Developed RESTful APIs that provided instrument information, historical prices, market indicators, and validation results to internal financial-data and analytical applications, with input validation, authentication, and structured error handling.
- Developed and maintained Python microservices using Flask, REST APIs, and Apache Kafka to support market-data validation, security mapping, reference-data updates, and asynchronous communication between distributed applications.
- Improved Python scripts, PostgreSQL queries, and database operations through reusable functions, batch processing, indexing, and query optimization, reducing average processing time by approximately 18%.
- Contributed to Apache Airflow workflows that scheduled daily market-data jobs, monitored processing dependencies, handled job failures, and published validated datasets to downstream reporting and analytics systems.
- Implemented automated data-quality checks, exception logging, and reconciliation rules for missing prices, duplicate securities, invalid identifiers, and stale market data, reducing manual investigation effort by approximately 20%.
- Wrote unit, integration, and API tests using PyTest and unittest, and used Git, Jenkins, Docker, and Linux to support code reviews, CI/CD deployments, production monitoring, troubleshooting, and Agile software delivery.