

Tab 0

LLM Engineer (GenAI / Agentic AI)

1. Core AI & Engineering Domains

Primary Focus Areas

- Generative AI (GenAI)
 - Large Language Models (LLMs)
 - Small Language Models (SLMs)
 - Agentic AI
 - Natural Language Processing (NLP)
 - Enterprise AI systems
 - No-code / test automation AI platforms
-

2. Large Language Models & Core Concepts

LLM / GenAI Concepts

- Transformers
 - Embeddings
 - Generative AI architectures
 - Domain-specific language models
 - Token efficiency optimization
 - Latency & cost optimization
-

3. Model Training, Fine-Tuning & Adaptation

Fine-Tuning Techniques

- LoRA
- QLoRA
- PEFT

Model Types

- Large Language Models (LLMs)

- Small Language Models (SLMs)

Training Approaches

- Fine-tuning
 - Domain adaptation
 - Pre-training (mentioned in bonus context)
-

4. ML & GenAI Frameworks

Deep Learning Frameworks

- PyTorch
- JAX

GenAI Frameworks

- Hugging Face Transformers
-

5. Retrieval-Augmented Generation (RAG)

RAG Capabilities

- RAG architectures
 - Vector-search workflows
 - Intelligent document processing pipelines
 - Retrieval optimization
-

6. Vector Databases & Retrieval Systems

Vector Databases

- Pinecone
 - Weaviate
 - FAISS
 - Milvus
-

7. Programming & Backend Development

Languages

- **Python**

Backend Capabilities

- API development
 - Backend services for AI systems
 - Integration of LLMs into SaaS products
-

8. Cloud Platforms & Deployment

Cloud Platforms

- **Amazon Web Services**
- **RunPod**
- E2E cloud environments (explicitly mentioned)

Deployment Scope

- AI model deployment
 - Production-grade inference environments
-

9. MLOps & Production Engineering

MLOps Lifecycle

- Model packaging
- CI/CD pipelines
- Model versioning
- Monitoring
- Automated evaluation

Containerization

- **Docker**
-

10. Model Optimization & Inference Efficiency

Optimization Techniques

- Quantization
 - Distillation
 - Caching
 - Batching
 - Token efficiency optimization
-

11. Research & Innovation

Research Areas

- Emerging GenAI techniques
 - Prompt experimentation
 - Domain-specific adaptation
 - ERP / testing-focused AI innovation
-

12. Enterprise & Domain Context

Enterprise Domains

- ERP systems
- Test automation
- Enterprise SaaS platforms

Enterprise Applications Mentioned

- Oracle Fusion Cloud
 - SAP S/4HANA
 - Workday
 - Salesforce
-

13. Collaboration & Product Integration

Collaboration

- Product engineering teams
 - Enterprise SaaS platform integration
 - Cross-functional AI roadmap execution
-

14. Experience & Qualification Context (Technical)

Experience

- 2–4 years (Bachelor's) or 1–2 years (Master's) in AI/ML engineering
- Hands-on NLP / GenAI experience

Education

- Bachelor's or Master's degree (technical fields)
-

15. Bonus / Differentiating Skills

Additional Capabilities

- Training or fine-tuning SLMs
 - Inference optimization
 - Open-source contributions
 - Publications, blogs, or research in NLP / GenAI
 - ERP or automation domain exposure
-

16. Consolidated Technical Stack Summary

Languages

- Python

GenAI & LLMs

- LLMs & SLMs
- Transformers
- Embeddings
- Prompting & fine-tuning
- Domain adaptation

Frameworks

- PyTorch
- JAX
- Hugging Face Transformers

Fine-Tuning

- LoRA
- QLoRA
- PEFT

RAG & Retrieval

- RAG pipelines
- Vector search workflows
- Document processing pipelines

Vector Databases

- Pinecone
- Weaviate
- FAISS
- Milvus

Cloud & Deployment

- AWS
- RunPod
- Docker

MLOps

- CI/CD
- Model versioning
- Monitoring
- Automated evaluation

Optimization

- Quantization
- Distillation
- Caching
- Batching

Tab 1

1. Large Language Models (LLMs)

Models Mentioned

- **GPT-4**
- **Claude**
- **Mistral**

Use Cases

- Natural language query processing
 - Personalized travel recommendations
 - Multi-turn conversational agents
 - Booking assistance, modifications, and cancellations
-

2. LLM Integration & Agentic Systems

Core Capabilities

- LLM deployment and integration
- Multi-platform conversational agents
- Conversational flow orchestration
- Context handling across sessions

Architectural Concepts

- Multi-turn conversation management
 - Conversational state handling
 - Intelligent automation pipelines
-

3. Prompt Engineering & Retrieval-Augmented Generation (RAG)

Techniques

- Prompt optimization
- Context-aware prompt design
- Retrieval-Augmented Generation (RAG)
- Grounding LLM responses with external knowledge

RAG Components

- Retrieval pipelines
 - Context injection for LLM inference
 - Multi-step reasoning support
-

4. Backend Frameworks & API Development

Programming Language

- **Python**

Frameworks

- **FastAPI**
- **Django**

Backend Capabilities

- Asynchronous programming
 - Secure API design
 - Scalable backend services
 - High-performance REST APIs
-

5. Databases & ORM Technologies

Databases

- **PostgreSQL**
- **MongoDB**

ORM

- **SQLAlchemy**

Data Responsibilities

- Schema design
 - Performance optimization
 - Structured and unstructured data handling
-

6. Cloud, DevOps & Infrastructure

Cloud Platforms

- Amazon Web Services (AWS)
- Google Cloud Platform (GCP)
- Microsoft Azure

Serverless & Compute

- AWS Lambda
- Cloud Functions

Infrastructure Patterns

- Containerized deployments
 - Serverless architectures
 - Scalable AI services
-

7. LLM Tooling & Ecosystem Libraries

LLM Frameworks

- LangChain
- LlamaIndex

Model & API Platforms

- OpenAI APIs
 - Anthropic APIs
 - Hugging Face Transformers
-

8. Model Fine-Tuning & Optimization

Fine-Tuning Techniques

- LoRA (Low-Rank Adaptation)
- PEFT (Parameter-Efficient Fine-Tuning)

Model Types

- Open-source LLMs

- Proprietary LLMs
-

9. Model Evaluation Metrics

Evaluation Methods

- BLEU
- ROUGE

Purpose

- Output quality evaluation
 - Language generation benchmarking
-

10. NLP Pipelines & Language Understanding

NLP Components

- Named Entity Recognition (NER)
 - Sentiment Analysis
 - Dialogue State Tracking
 - Intent Detection
 - Text Classification
 - Dialogue Management
-

11. Performance, Scalability & Optimization

Optimization Goals

- Latency reduction
- Cost efficiency
- Throughput scalability
- Production-grade reliability

System Considerations

- Distributed environments
- Multi-cloud deployment readiness
- AI workload optimization

12. Collaboration & Development Practices

Cross-Functional Teams

- AI researchers
- Frontend developers
- Product teams

Engineering Practices

- Iterative development
 - Rapid feature delivery
 - AI-first system design
 - User-centric problem solving
-

13. Domain-Specific Context

Industry

- Travel technology
 - Conversational travel agents
 - Intelligent booking assistance
 - Travel recommendation systems
-

Summary View (Condensed Technology Stack)

Languages

- Python

Frameworks

- FastAPI, Django, LangChain, LlamaIndex

LLMs

- GPT-4, Claude, Mistral

Databases

- PostgreSQL, MongoDB

ORM

- SQLAlchemy

Cloud

- AWS, GCP, Azure

Serverless

- AWS Lambda, Cloud Functions

NLP

- NER, Sentiment Analysis, Dialogue State Tracking

Fine-Tuning

- LoRA, PEFT

Evaluation

- BLEU, ROUGE

Tab 2

Generative AI Developer Role

1. Core AI & Generative AI Domains

Primary Focus Areas

- Artificial Intelligence (AI)
- Generative Artificial Intelligence (GenAI)

Responsibilities

- Design, build, and deploy generative AI solutions
 - Develop AI-based custom solutions for service delivery teams
 - Production deployment and monitoring of generative AI systems
 - Performance and reliability management of AI systems
 - Adoption of best practices for AI system development and operations
-

2. Generative AI Models & Techniques

Model Types Explicitly Mentioned

- Generative Adversarial Networks (GANs)
- Diffusion Models
- Transformer-based Models

Generative AI Capabilities

- State-of-the-art generative modeling
 - Application of multiple generative AI techniques
 - Model optimization for production use
-

3. Machine Learning & Deep Learning

Machine Learning Scope

- Scalable machine learning model design
- Model deployment in production environments

- Machine learning optimization techniques

Deep Learning

- Deep learning model development
 - Training and deployment of advanced neural architectures
-

4. Prompt Engineering & Optimization

Techniques

- Prompt engineering
 - Model behavior optimization
 - Performance tuning of generative models
-

5. AI Subdomains

Natural Language Processing (NLP)

- Text understanding and generation
- Language-based AI systems

Computer Vision

- Image-based AI systems
- Visual data processing

Reinforcement Learning (RL)

- Learning from reward-based feedback
 - Decision-making models
-

6. Cloud Platforms & Deployment

Cloud Platforms

- Microsoft Azure
- Google Cloud Platform (GCP)

Cloud Capabilities

- Scalable model deployment
 - Cloud-based AI infrastructure
 - Production-grade AI services
-

7. Distributed Systems & Scalability

System Characteristics

- Distributed computing
 - Scalable systems architecture
 - High-performance AI workloads
-

8. Production AI & MLOps Practices

Operational Responsibilities

- Deployment of generative AI systems into production
- Monitoring AI systems in live environments
- Ensuring reliability and performance standards
- Managing lifecycle of AI models

Best Practices

- AI system governance
 - Deployment standards
 - Operational stability for AI solutions
-

9. Software Engineering & Development Experience

Experience Areas

- AI-based software development
 - End-to-end AI solution delivery
 - Integration of AI systems into existing services
-

10. Collaboration & Engineering Leadership

Team Interaction

- Collaboration with engineers and scientists
- Close coordination with product managers
- Engagement with business stakeholders

Leadership Responsibilities

- Leading and mentoring engineering teams
 - Driving innovation in AI solution development
-

11. Educational & Experience Requirements (Technical Context)

Educational Background

- Computer Science
- Artificial Intelligence
- Related technical fields

Experience

- Generative AI: 2+ years
 - AI-based software development: 3+ years
-

12. Condensed Technical Stack Summary

AI Domains

- Generative AI
- Machine Learning
- Deep Learning

Model Types

- GANs
- Diffusion Models
- Transformers

AI Techniques

- Prompt Engineering
- Model Optimization
- Reinforcement Learning

Subdomains

- Computer Vision
- Natural Language Processing

Cloud Platforms

- Azure
- GCP

System Design

- Distributed Computing
- Scalable Systems

Operational Focus

- Production Deployment
- Monitoring & Reliability
- AI Best Practices

Tab 3

Generative AI / AI-ML Role

1. Core Technical Domains

Primary Skill Areas

- Generative AI (GenAI)
- Artificial Intelligence (AI)
- Machine Learning (ML)

Engineering Scope

- Architecture design for GenAI-powered systems
 - Implementation of AI-driven solutions
 - Scalable AI system development
 - Technical leadership and mentoring
-

2. Programming Languages

- Python
-

3. Machine Learning & AI Frameworks

Core Frameworks

- TensorFlow
- PyTorch

LLM & GenAI Frameworks

- LangChain
 - Hugging Face
-

4. Generative AI Models & Techniques

Model Architectures

- Large Language Models (LLMs)
- Diffusion Models
- Transformer Models

Techniques

- Prompt Engineering
-

5. API Development & Integration

API Types

- RESTful APIs
- GraphQL APIs

Use Case

- AI-powered service exposure
 - Product and platform integration
-

6. ML Pipelines & Workflow Orchestration

Pipeline & MLOps Tools

- **MLflow**
- **Kubeflow**
- **Apache Airflow**
- **Vertex AI**

Capabilities

- ML pipeline construction
 - Performance optimization
 - Reliability engineering
-

7. Cloud Platforms & Infrastructure

Cloud Providers

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)

Cloud-Native Technologies

- Docker
 - Kubernetes
-

8. Vector Databases & RAG

Vector Databases

- Pinecone
- FAISS
- Weaviate

Retrieval Techniques

- Retrieval-Augmented Generation (RAG)
-

9. Data Engineering & Model Operations

Data Engineering Concepts

- Feature extraction
- Model serving
- Model monitoring

Operational Focus

- Performance optimization
 - Reliability of ML systems
-

10. MLOps & LLMOps Practices

Practices

- Continuous model training
- Continuous testing
- Continuous deployment of AI models

Operational Domains

- MLOps
 - LLMOps
 - Responsible AI practices
-

11. Software Engineering Best Practices

Engineering Standards

- Clean coding practices
 - Testing strategies
 - Cloud-based deployment workflows
-

12. Leadership & Collaboration (Technical Context)

Responsibilities

- Technical mentorship
 - Cross-functional collaboration
 - Architecture ownership
 - Engineering leadership
-

13. Consolidated Technology Stack Summary

Languages

- Python

ML / DL Frameworks

- TensorFlow
- PyTorch

LLM & GenAI

- LangChain
- Hugging Face
- LLMs
- Diffusion Models
- Transformers
- Prompt Engineering

APIs

- RESTful APIs
- GraphQL APIs

ML Pipelines

- MLflow
- Kubeflow
- Apache Airflow
- Vertex AI

Cloud & Infrastructure

- AWS
- Azure
- GCP
- Docker
- Kubernetes

Vector Databases

- Pinecone
- FAISS
- Weaviate

AI System Practices

- RAG
- MLOps
- LLMOps
- Model Monitoring
- Responsible AI

Tab 4

Essential Skills & Responsibilities (AI Role)

1. Programming Languages

- Python
 - Java
-

2. AI / ML Frameworks & Platforms

AI / ML Frameworks

- TensorFlow
- PyTorch

Cloud AI Platforms

- Oracle Cloud AI Services
-

3. Enterprise Software Engineering

Architectural Patterns

- REST APIs
- Microservices
- Cloud-native architectures

Application Scope

- Enterprise-grade AI applications
 - Scalable AI-driven systems
 - Business workflow automation
-

4. Large Language Models & Modern AI Tooling

Advanced AI Technologies

- Large Language Models (LLMs)
 - Retrieval Augmented Generation (RAG)
 - AI Agents
 - Model Context Protocol (MCP)
-

5. Classical AI & Machine Learning Techniques

Machine Learning Methods

- Clustering
- Classification
- Regression

Statistical & Probabilistic Techniques

- Statistical modeling
 - Monte Carlo simulations
 - Bayesian blending
-

6. Cloud Platforms & Infrastructure

Cloud Environment

- Cloud-based AI application deployment
- Enterprise cloud environments

Preferred Cloud Platform

- Oracle Cloud Infrastructure (OCI)
-

7. DevOps & CI/CD

Practices

- DevOps workflows

- Continuous Integration (CI)
 - Continuous Deployment / Delivery (CD)
-

8. AI System Design & Operations

System Responsibilities

- Design and development of AI-based enterprise applications
- Support and maintenance of AI systems
- Performance optimization
- Reliability and scalability assurance

Operational Focus

- Cloud scalability
 - High availability
 - Enterprise performance requirements
-

9. Product & Feature Development

Development Activities

- Building new AI-powered product features
 - Enhancing existing features using AI
 - Automating complex business flows
 - Supporting enterprise operations
-

10. Software Engineering Best Practices

Engineering Standards

- Development best practices
 - Testing methodologies
 - Deployment standards
 - Enterprise software lifecycle management
-

11. Collaboration & Delivery

Team Interaction

- Cross-functional collaboration
- Multi-disciplinary teams
- Global team environments

Delivery Context

- Requirements gathering
 - System planning and scheduling
 - Enterprise-scale implementation
-

12. Educational & Experience Context (Technical Relevance)

Academic Background

- Computer Science
- Engineering
- Mathematics
- Data Science
- Artificial Intelligence

Experience Areas

- AI
 - Software Engineering
 - Enterprise AI solution development
-

13. Consolidated Technical Stack Summary

Languages

- Python
- Java

AI / ML Frameworks

- TensorFlow

- PyTorch

AI Paradigms

- Large Language Models
- AI Agents
- Retrieval Augmented Generation (RAG)
- Model Context Protocol (MCP)

Classical AI

- Clustering
- Classification
- Regression
- Monte Carlo simulations
- Bayesian blending
- Statistical modeling

Enterprise Engineering

- REST APIs
- Microservices
- Cloud-native architectures

Cloud

- Oracle Cloud AI Services
- Oracle Cloud Infrastructure (OCI)

DevOps

- CI/CD
- DevOps practices

Tab 5

AI / ML & Generative AI Engineer Role

1. Programming Language & Core Libraries

Programming Language

- Python

Python Libraries

- NumPy
 - Pandas
 - Scikit-learn
 - PyTorch
 - TensorFlow
-

2. AI / ML Model Types & Use Cases

Model Use Cases

- Classification
- Prediction
- Recommendation
- Optimization

Model Lifecycle

- Data preprocessing
 - Model training
 - Evaluation
 - Deployment
 - Versioning
 - Monitoring
 - Scaling
-

3. Generative AI Models & Architectures

Generative AI Models

- Large Language Models (LLMs)
- Diffusion Models
- Transformer Models

LLMs Explicitly Mentioned

- **GPT**
- **LLaMA**
- **Claude**

Techniques

- Prompt engineering
 - Model fine-tuning
 - Retrieval Augmented Generation (RAG)
-

4. Agentic AI & Autonomous Systems

Agentic Capabilities

- Reasoning
- Planning
- Autonomous task execution

Agentic AI Frameworks

- **LangChain**
 - **LlamaIndex**
 - **AutoGPT**
 - **CrewAI**
-

5. Natural Language Processing & Embeddings

NLP Concepts

- Natural Language Processing (NLP)
- Embeddings

6. Vector Databases

Vector Stores

- Pinecone
- Weaviate
- FAISS
- Milvus

7. API Development & Backend Frameworks

API Type

- REST APIs

Backend Frameworks

- FastAPI
- Flask

API Engineering Concepts

- Asynchronous programming
- High-performance API design
- API scaling in production

8. Security & Authentication

Authentication & Authorization

- JWT (JSON Web Tokens)
- OAuth

9. Containers & Cloud Deployment

Containerization

- **Docker**
- **Kubernetes**

Cloud Platforms

- **Amazon Web Services (AWS)**
 - **Google Cloud Platform (GCP)**
 - **Microsoft Azure**
-

10. Enterprise Integration & System Design

Integrations

- AI models with APIs
- AI models with databases
- AI models with enterprise systems

Workflow Scope

- End-to-end AI-driven workflows
 - Production-grade enterprise applications
-

11. Performance, Reliability & Optimization

Operational Concerns

- Model performance evaluation
 - API scalability
 - Reliability assurance
 - Cost and latency optimization
-

12. Research, Prototyping & Productionization

Engineering Activities

- Researching new AI techniques
- Rapid prototyping
- Production deployment of AI solutions

13. Collaboration & Delivery Context

Cross-Functional Collaboration

- Product managers
- Data scientists
- Software engineers

Delivery Goals

- Scalable AI-driven applications
 - Business-aligned AI systems
-

14. Consolidated Technical Stack Summary

Languages & Libraries

- Python
- NumPy, Pandas, Scikit-learn
- PyTorch, TensorFlow

Generative AI

- GPT, LLaMA, Claude
- LLMs, Diffusion Models, Transformers
- Prompt Engineering, RAG, Fine-tuning

Agentic AI

- LangChain
- LlamaIndex
- AutoGPT
- CrewAI

APIs & Backend

- REST APIs
- FastAPI
- Flask
- Asynchronous programming

Vector Databases

- Pinecone
- Weaviate
- FAISS
- Milvus

Security

- JWT
- OAuth

Containers & Cloud

- Docker
- Kubernetes
- AWS
- GCP
- Azure

Tab 6

AI Engineer Role

1. Core AI & ML Domains

Primary Areas

- Artificial Intelligence (AI)
- Machine Learning (ML)
- Generative AI
- Large Language Models (LLMs)

AI Capabilities

- ML model design and implementation
 - LLM training and fine-tuning for domain-specific tasks
 - Data-driven AI system development
-

2. Generative AI & Prompt Engineering

Generative AI Scope

- GPT-based systems
- Large Language Models (LLMs)
- Model fine-tuning

Techniques

- Prompt engineering
 - Prompt refinement
 - Functional analysis of AI systems
-

3. Programming Languages

- Python
 - Node.js (JavaScript runtime)
-

4. Backend Frameworks & API Development

Frameworks

- Node.js
- Express.js

API Development

- REST APIs
 - Backend service development
 - Enterprise system integration
-

5. Data Engineering & Pipelines

Data Pipeline Activities

- Data ingestion
 - Data processing
 - Data analysis
 - Pipeline optimization
-

6. Cloud Platforms & Deployment

Cloud Providers

- Amazon Web Services (AWS)
- Microsoft Azure
- Vercel

Deployment Models

- Cloud-based AI applications
 - Platform-as-a-Service (PaaS)
 - Runtime environment management
-

7. Monitoring, Performance & Operations

Operational Capabilities

- Application monitoring
 - Performance management
 - High availability systems
 - Systems monitoring
 - Runtime troubleshooting
 - Configuration management
-

8. Source Control & Configuration Management

Tools

- **Git**

Practices

- Version control
 - Code reviews
 - Configuration management
-

9. Agile & Delivery Methodologies

Agile Frameworks

- **SAFe (Scaled Agile Framework)**
- **FAST**

Agile Practices

- Sprint planning
 - Backlog grooming
 - Team reviews
 - Agile at Scale collaboration
-

10. Software Engineering Best Practices

Engineering Standards

- Code quality enforcement
- Peer code reviews

- Maintainable backend architecture
 - Scalable application design
-

11. Statistics & Data Science

Knowledge Areas

- Data science
 - Statistics
 - Machine learning theory
-

12. IT Operations & System Integration

Operational Responsibilities

- IT operations support
 - Enterprise system integration
 - Issue resolution related to:
 - Configuration
 - Runtime
 - Performance
-

13. Performance Monitoring Tools (Awareness)

Preferred Knowledge

- Application performance monitoring (APM) tools
(Specific tools are not named in the content.)
-

14. Domain Exposure (Non-Technical Context)

Optional Domain Experience

- Smart building automation
- Engineering domains

15. Consolidated Technology Stack Summary

Languages

- Python
- JavaScript (Node.js)

Backend Frameworks

- Node.js
- Express.js

AI & ML

- Machine Learning
- Large Language Models (LLMs)
- Generative AI
- GPT-based systems
- Prompt Engineering
- Model Fine-tuning

Data Engineering

- Data ingestion pipelines
- Data processing & analysis

Cloud & Platforms

- AWS
- Azure
- Vercel
- PaaS environments

DevOps & Operations

- Git
- Configuration management
- Application monitoring
- Performance management
- High availability systems

Agile

- SAFe

- FAST
- Sprint planning
- Backlog grooming

Tab 7

GenAI Software Engineer Role

1. Core AI & Generative AI Technologies

Primary Domains

- Generative AI (GenAI)
- AI-based application development

Mandatory Skill

- Generative AI Copilot

GenAI Techniques

- Prompt Engineering
 - Embedding Models
-

2. Programming Languages

- Python
 - Java
 - .NET
 - JavaScript (via React / Angular)
-

3. GenAI Frameworks & Libraries

- LangChain

Usage Context

- Building GenAI-based applications
 - Integrating GenAI workflows into existing systems
 - Orchestrating LLM interactions using prompts and embeddings
-

4. Software Engineering Frameworks & Technologies

Backend / Platform Technologies

- **.NET**
- **Java**

Frontend Frameworks

- **React**
 - **Angular**
-

5. Cloud Platforms

Cloud Environments

- **Amazon Web Services (AWS)**
- **Microsoft Azure**
- **Google Cloud**

Cloud Usage

- Deployment of GenAI solutions
 - Performance tuning in cloud environments
 - Production support of AI applications
-

6. Software Development & Engineering Practices

Development Activities

- Design, development, and testing of applications
- Writing clean, maintainable, and efficient code
- Adhering to coding standards and quality assurance parameters

Quality & Reliability

- Debugging
- Performance tuning
- Reliability and scalability of software solutions

7. DevOps & Deployment

DevOps

- Familiarity with DevOps tools and practices
(*Specific tools are not named in the content.*)

Deployment

- Cloud-based deployment of GenAI applications
 - Client-end software deployment support
-

8. Agile & Development Processes

Process Exposure

- Agile development processes
-

9. Operational Responsibilities

Production Support

- Testing and maintenance of software programs
 - Supporting deployment at client environments
 - Ensuring quality assurance compliance
 - Ongoing system optimization
-

10. Continuous Learning & Improvement

Technical Growth

- Staying updated with advancements in Generative AI
 - Continuous improvement of GenAI solutions
-

11. Certifications (Technical Context)

Relevant Certifications

- Python
 - AI / ML
 - Cloud technologies
-

12. Experience Requirements (Technical Relevance)

Experience

- 4-6 years in software development
 - 1-2 years in AI / Generative AI
 - Mandatory experience range: 3-5 years
-

13. Consolidated Technical Stack Summary

Languages

- Python
- Java
- .NET
- JavaScript

GenAI

- Generative AI
- Generative AI Copilot
- Prompt Engineering
- Embedding Models
- LangChain

Frontend

- React
- Angular

Cloud

- AWS

- Azure
- Google Cloud

Practices

- Agile development
- DevOps practices
- Debugging & performance tuning
- Quality assurance

Tab 8

AI / ML / GenAI Role

1. Programming & Data Skills

Languages

- **Python** (statistical programming)
 - **SQL** (database querying)
-

2. AI / ML & Generative AI

Core Areas

- Machine Learning algorithms and workflows
- Large Language Models (LLMs)
- Generative AI solutions

Techniques

- Prompt engineering (AI performance optimization)

Conversational AI

- **ChatGPT** (or similar conversational AI platforms)
-

3. Natural Language Processing (NLP) & Text Analytics

Experience

- 2+ years in NLP

NLP Tasks

- Text analytics
- Document AI
- Optical Character Recognition (OCR)

- Sentiment analysis
 - Entity recognition
 - Topic modeling
-

4. Statistical & Analytical Techniques

Experience

- 2+ years in statistical modeling

Methods

- Forecasting
 - Classification
 - Applied statistics
 - Advanced statistical techniques
-

5. Machine Learning Frameworks & Libraries

Deep Learning & ML

- TensorFlow
 - PyTorch
 - Keras
 - scikit-learn
 - spaCy
-

6. Cloud, Deployment & Orchestration

Cloud Platforms

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)

Deployment & Ops

- Model deployment
- Orchestration

- Containerization
 - **Kubernetes**
-

7. Applied AI Solution Areas

Application Types

- Prediction systems
 - Recommendation systems
 - Text analytics platforms
 - Computer vision applications
 - Bots
 - Content intelligence solutions
-

8. Automation & AI Technologies

AI Technologies

- NLP
- Virtual agents
- Computer vision
- Neural networks

Use Case

- Automation
-

9. Collaboration & Engineering Practices

Version Control

- **Git**

Skills

- Problem-solving
 - Technical communication
-

10. Consolidated Technical Stack Summary

Languages

- Python
- SQL

GenAI & NLP

- LLMs
- Generative AI
- Prompt Engineering
- ChatGPT
- NLP (Sentiment, NER, Topic Modeling, OCR, Document AI)

ML Frameworks

- TensorFlow
- PyTorch
- Keras
- scikit-learn
- spaCy

Statistics & ML

- Statistical modeling
- Forecasting
- Classification
- Advanced statistics

Cloud & Deployment

- AWS
- Azure
- GCP
- Kubernetes
- Model deployment & orchestration

Applied AI

- Prediction
- Recommendation
- Computer Vision
- Bots
- Content Intelligence

Dev Practices

- Git

Tab 9

AI / ML & NLP Engineer Role

1. Machine Learning & Model Development

Core Activities

- Design, training, and deployment of machine learning models
- Real-world ML application development

ML Algorithms

- Linear Regression
 - Logistic Regression
 - Decision Trees
 - Random Forest
 - K-Nearest Neighbors (KNN)
 - Support Vector Machines (SVM)
 - Bayesian Models
-

2. Natural Language Processing (NLP)

NLP Techniques

- Zero-Shot Classification
- Few-Shot Classification

Text Representation & Embeddings

- TF-IDF
- Word2Vec
- BERT

Use Cases

- Text-based processing tasks
 - NLP-driven solutions for classification and understanding
-

3. Generative AI & Large Language Models

Generative AI Models

- **OpenAI** (models)
- **LLaMA**

Capabilities

- Generative AI model development
 - Model fine-tuning for business use cases
-

4. Backend Development & APIs

API & Service Architecture

- RESTful APIs
- Microservices

Backend Frameworks

- **Flask**
 - **FastAPI**
-

5. Data Engineering & Databases

Data Pipelines

- Data pipeline management
- Integration of structured and unstructured data

Databases

- **MongoDB**
 - **MySQL**
-

6. Version Control & Collaboration Tools

Version Control Systems

- **Git**
 - **Bitbucket**
-

7. Project Management & Delivery Tools

Project Management Platforms

- **JIRA**
 - **Asana**
-

8. Collaboration & Business Integration

Cross-Functional Collaboration

- Product teams
- Business teams

Engineering Responsibilities

- Translating business requirements into scalable AI solutions
 - Supporting end-to-end AI solution delivery
-

9. Consolidated Technical Stack Summary

Machine Learning

- Linear Regression
- Logistic Regression
- Decision Trees
- Random Forest
- KNN
- SVM
- Bayesian Models

NLP

- Zero-Shot Classification
- Few-Shot Classification
- TF-IDF

- Word2Vec
- BERT

Generative AI

- OpenAI models
- LLaMA

Backend & APIs

- Flask
- FastAPI
- RESTful APIs
- Microservices

Databases

- MongoDB
- MySQL

Dev & Collaboration

- Git
- Bitbucket
- JIRA
- Asana

Tab 10

Data Scientist / Generative AI Role

1. Core Technical Domains

Primary Areas

- Data Science
- Artificial Intelligence (AI)
- Generative AI
- Machine Learning (ML)

Work Scope

- End-to-end execution of data science projects
 - Production deployment of generative AI models
 - Business-oriented analytical solution design
-

2. Programming Languages

- Python
-

3. Machine Learning & Generative AI

Machine Learning Capabilities

- Predictive modeling
- Forecasting solutions
- Recommendation systems

Generative AI

- Building generative AI models
 - Deploying generative AI models in production environments
 - Application of generative AI techniques
-

4. Data Processing & Analytics

Data Types

- Structured data
- Unstructured data

Techniques

- Data preprocessing
 - Statistical analysis
 - Analytical modeling
-

5. Statistical Methods

Approach

- Statistical techniques for data preprocessing and analysis
 - Applied statistics for modeling and forecasting
-

6. AI Frameworks & Tooling

Framework Experience

- AI frameworks (not individually named, but explicitly referenced as relevant AI frameworks)
-

7. Cloud Platforms & MLOps

Infrastructure

- Cloud platforms (explicitly mentioned, not named)

MLOps

- MLOps tools for:
 - Model deployment
 - Model lifecycle management
 - Production monitoring

8. Model Deployment & Production Systems

Production Responsibilities

- Deploying AI and generative AI models
 - Ensuring production readiness
 - Supporting scalable AI systems
-

9. Cross-Functional & Stakeholder Collaboration (Technical Context)

Collaboration With

- Engineering teams
- Product teams
- Business teams
- External stakeholders / clients

Activities

- Requirements gathering
 - Translating business problems into analytical approaches
 - Presenting technical findings
 - Guiding data-driven decision-making
-

10. Communication & Data Storytelling

Technical Communication

- Data storytelling
 - Presentation of analytical insights
 - Explaining AI and ML outcomes to non-technical audiences
-

11. Project & Delivery Management

Project Skills

- Project management in data science initiatives
 - End-to-end ownership of analytics projects
-

12. Professional Experience Context (Technical Relevance)

Experience Range

- 3–6 years total professional experience
 - 1–3 years hands-on experience in AI
 - Proven production experience with generative AI models
-

13. Research & Open Source (Optional Technical Exposure)

Good-to-Have

- Publications in AI
 - Open-source contributions in AI-related projects
-

14. Organizational Context

Company

- Metyis
-

15. Consolidated Technical Stack Summary

Languages

- Python

AI & ML

- Machine Learning
- Generative AI
- Predictive Modeling
- Forecasting
- Recommendation Systems

Data Science

- Structured & Unstructured Data
- Statistical Analysis
- Data Preprocessing

Deployment & Ops

- Cloud Platforms
- MLOps Tools
- Production AI Systems

Collaboration

- Stakeholder engagement
- Data storytelling
- Client-facing analytics

Tab 11

Agentic AI / GenAI Engineer Role

1. Core AI & ML Domains

Primary Focus Areas

- Agentic AI systems
- Generative AI
- Large Language Models (LLMs)
- Applied Machine Learning Engineering

System Capabilities

- Autonomous workflows
 - Retrieval-Augmented Generation (RAG)
 - Model-driven services
 - Scalable, production-grade AI systems
-

2. Programming Languages & Foundations

Languages

- Python

Core Knowledge Areas

- Data handling
 - ML pipelines
 - Deep learning concepts
-

3. Generative & Agentic AI Frameworks

Agentic / GenAI Frameworks

- LangChain
- LangGraph
- CrewAI

- **DSPy**

Use Cases

- Intelligent system orchestration
 - Agent-based reasoning and planning
 - Fine-tuned and retrieval-augmented pipelines
-

4. Model Engineering & Prompt Design

LLM Engineering

- Integration of large language models
- Model customization
- Lightweight model tuning workflows

Prompt Engineering

- Structured prompt design
- Context optimization
- Cost and latency optimization

Model Evaluation

- Prompt and model evaluation logic
- Quantitative and qualitative assessment

Evaluation Frameworks

- **AgentEval**
 - **DeepEval**
-

5. API Development & Service Deployment

API Frameworks

- REST APIs
- **FastAPI**

Service Engineering

- Scalable AI services

- Cloud-ready application design
 - Service versioning and monitoring
-

6. Data Pipelines & Vector Databases

Embedding & Retrieval Pipelines

- Embedding generation pipelines
- Retrieval optimization
- Collaboration with data engineering teams

Vector Databases / Search

- **Milvus**
 - **Pinecone**
 - **ChromaDB**
 - **FAISS**
 - **Azure AI Search**
-

7. Cloud Platforms & Model Hosting

Managed LLM Platforms

- **Azure OpenAI**
 - **AWS Bedrock**
 - **Vertex AI**
-

8. Containerization & DevOps

Container & Orchestration

- **Docker**
- **Kubernetes**

CI/CD Platforms

- **GitHub**
- **Azure DevOps**

9. MLOps, Observability & Monitoring

MLOps Tools

- MLflow
- Arize Phoenix
- Weights & Biases

Capabilities

- Model versioning
 - Experiment tracking
 - Observability and monitoring
-

10. Evaluation, Governance & Responsible AI

Evaluation Practices

- Structured prompt testing
- Quantitative and qualitative evaluations

Governance & Ethics

- Responsible AI
 - Data privacy
 - Model governance
-

11. AI-Assisted Development Tools (Nice to Have)

Developer Tools

- Cursor
 - Claude Code
-

12. Application Domains & Experience (Nice to Have)

Experience Areas

- LLM-based assistants
 - Retrieval systems
 - Open-source AI/ML contributions
 - Applied AI research
-

13. Consolidated Technical Stack Summary

Languages

- Python

Agentic & GenAI

- LangChain
- LangGraph
- CrewAI
- DSPy
- LLMs
- RAG
- Prompt Engineering
- Lightweight Model Tuning

Vector Databases

- Milvus
- Pinecone
- ChromaDB
- FAISS
- Azure AI Search

APIs & Services

- REST APIs
- FastAPI

Cloud & Hosting

- Azure OpenAI
- AWS Bedrock
- Vertex AI

Containers & DevOps

- Docker
- Kubernetes
- GitHub
- Azure DevOps

MLOps & Evaluation

- MLflow
- Arize Phoenix
- Weights & Biases
- AgentEval
- DeepEval

Governance

- Responsible AI
- Data Privacy
- Model Governance

Tab 12

Sr AI Engineer & AI Engineer Jr

1. Core Domains & Responsibilities

Primary Technical Domains

- Artificial Intelligence (AI)
- Generative AI (GenAI)
- Machine Learning (ML)
- Data Engineering
- Applied AI
- Data Science
- Analytics Applications

Engineering Scope

- End-to-end AI application lifecycle (analysis → design → build → test → production)
 - Feasibility studies and solution design
 - High-performance, standards-compliant coding
 - Code reviews
 - Customer-facing AI solution delivery
 - Proof of Concepts (PoCs) and production deployments
-

2. Programming Languages & Frameworks

Languages

- Python
- PySpark
- JavaScript (Good to Have)
- .NET (Good to Have)

Backend / API Frameworks

- Flask
 - FastAPI
-

3. Generative AI & LLM Technologies

Core GenAI Skills

- Large Language Models (LLMs)
- Prompt Engineering
- Fine-tuning and customization of pre-trained models
- Deployment of generative models

GenAI Architectures

- Retrieval Augmented Generation (RAG)

Frameworks & SDKs

- Semantic Kernel
-

4. Azure AI & Cloud Services (Primary Platform)

Azure AI & ML Services

- Azure OpenAI
- Azure Machine Learning
- Azure Form Recognizer
- Azure Cognitive Search

Azure Application & Integration Services

- Azure App Service
 - Azure Bot Framework
 - Azure Functions
 - Azure SQL
 - Azure Blob Storage
-

5. Vector Databases & Search

Vector & Semantic Search

- Vector Databases
- Azure Cognitive Search (vector & semantic capabilities)

6. Data Engineering & Large-Scale Processing

Data Engineering

- Large data processing
- AI data engineering patterns
- Analytics application development

Technologies

- PySpark
 - Data pipelines
 - Data engineering workflows
-

7. ML / AI Platforms & MLOps

Platforms & APIs

- Azure Machine Learning
- MLflow
- Amazon SageMaker

Capabilities

- Model training
 - Model deployment
 - Experiment tracking
 - ML lifecycle management
-

8. Generative AI Project Experience

Experience Requirements

- 12+ GenAI customer projects / PoCs (Sr role)
 - 1+ GenAI customer project / PoC (Jr role)
 - Recent LLM model experience
-

9. Software Development Lifecycle

Lifecycle Coverage

- Requirements analysis
 - Architecture
 - Design
 - Build
 - Test
 - Production deployment
-

10. Certifications (Explicitly Mentioned)

Microsoft Certifications

- **AI-102 Microsoft Certified Azure AI Engineer Associate**
- **DP-100 Microsoft Certified Azure Data Scientist Associate**

Other Certifications

- **Databricks Professional Certificate in Large Language Models**
-

11. Soft Skills & Collaboration (Technical Context)

- Problem-solving
 - Customer communication
 - Team collaboration
 - Matrix organization experience
 - Independent solution ownership
-

12. Education (Technical Context)

Minimum

- BCA
- MCA
- BE
- BTech

- Equivalent technical degree

Preferred

- MTech
 - MS
-

13. Consolidated Technology Stack Summary

Languages

- Python
- PySpark
- JavaScript (optional)
- .NET (optional)

GenAI & LLM

- LLMs
- Prompt Engineering
- RAG
- Semantic Kernel
- Fine-tuning
- Vector Databases

Azure AI

- Azure OpenAI
- Azure ML
- Azure Form Recognizer
- Azure Cognitive Search

Azure Platform

- Azure App Service
- Azure Bot Framework
- Azure Functions
- Azure SQL
- Azure Blob Storage

APIs

- Flask
- FastAPI

MLOps

- MLflow
- Azure ML
- SageMaker

Certifications

- AI-102
- DP-100
- Databricks LLM Certificate

Tab 13

Generative AI / Agentic AI Engineer

1. Programming Languages & Core Engineering

Primary Language

- C#

Engineering Practices

- Asynchronous programming
 - Dependency Injection
 - Configuration management
 - Backend service development
 - Microservices architecture
 - REST API development
 - System design
 - Debugging and problem-solving
-

2. AI / ML & Generative AI Foundations

Core Domains

- Artificial Intelligence (AI)
- Machine Learning (ML)
- Generative AI (GenAI)

Model Concepts

- Large Language Models (LLMs)
- Embeddings
- Multi-step reasoning

Techniques

- Prompt engineering
 - Grounding techniques
-

3. LLM Integration & Agentic AI Frameworks

Agentic / LLM Frameworks

- **Semantic Kernel**
- **AutoGen**
- **MS Agentic Framework**
- **LangChain**
- **LangGraph**
- ****OpenAI (APIs / models)**

Capabilities

- Autonomous agent design
 - Multi-agent workflows
 - LLM-powered reasoning systems
-

4. Retrieval & Knowledge Augmentation

RAG & Retrieval

- Retrieval-Augmented Generation (RAG) pipelines
- Vector databases
- Embedding-based retrieval

Search & Grounding

- **Azure AI Search**
-

5. Cloud Platforms & Infrastructure

Cloud Provider

- **Microsoft Azure** (preferred)

Cloud-Native Architecture

- Cloud-native services
 - Scalable backend architectures
-

6. Containers & Orchestration

Containerization

- Docker

Orchestration

- Kubernetes
-

7. CI/CD & DevOps

CI/CD

- Azure DevOps

Practices

- Automated pipelines
 - Build, test, and deployment workflows
-

8. Observability, Safety & Guardrails

System Safety

- Content filters
- Safety policies
- Tool access constraints
- Output constraining

Operational Resilience

- Observability
 - Fallback mechanisms
-

9. Development Methodology

Ways of Working

- Agile development
 - Fast-paced experimentation
 - Iterative delivery
-

10. Domain Experience (Optional / Advantage)

Industry Exposure

- Healthcare
-

11. Educational Background (Technical Context)

- BE / B.Tech
 - MCA
 - ME / M.Tech
-

12. Consolidated Technical Stack Summary

Languages

- C#

GenAI & Agentic AI

- LLMs
- Prompt Engineering
- Multi-step Reasoning
- RAG Pipelines
- Embeddings
- Semantic Kernel
- AutoGen
- MS Agentic Framework
- LangChain
- LangGraph
- OpenAI APIs

Backend & APIs

- REST APIs

- Microservices
- Asynchronous Programming
- Dependency Injection

Cloud & DevOps

- Azure
- Azure AI Search
- Azure DevOps
- Docker
- Kubernetes

System Qualities

- Observability
- Safety Guardrails
- Fallback Mechanisms
- Cloud-Native Architecture

Tab 14

Agentic AI Engineer

1. Programming Languages & Core Engineering

Languages

- Python
- SQL
- Go (familiarity)
- Rust (familiarity)

Software Engineering Practices

- Microservices architecture
 - Test-driven development (TDD)
 - Concurrency processing
 - Low-Level Design (LLD)
 - High-performance backend systems
-

2. Agentic AI & Generative AI Frameworks

Agentic / LLM Frameworks

- LangGraph
- CrewAI
- LangChain

LLM Frameworks

- Hugging Face Transformers

Capabilities

- Autonomous agent workflows
- Prompt engineering
- Semantic retrieval
- Chatbot development
- Recommendation systems
- Translation services

3. API Development & Backend Frameworks

API Framework

- **FastAPI**

Backend Architecture

- REST APIs
 - Microservices
 - High-performance API design
-

4. Data Storage & Databases

Databases

- **MongoDB**
- Vector Databases (VectorDB)

Use Cases

- Data persistence
 - Semantic retrieval
 - Efficient embedding-based search
-

5. Machine Learning & Deep Learning Frameworks

ML / DL Frameworks

- **TensorFlow**
- **PyTorch**

ML Capabilities

- Model debugging and optimization
 - Model fine-tuning
 - Performance and security optimization
-

6. LLM Infrastructure & Inference

LLM Serving Platforms

- vLLM

Inference Optimization

- GPU memory optimization
 - Model quantization
 - Scalable LLM inference architectures
-

7. Big Data & Distributed Processing

Big Data Processing

- Apache Spark

Use Case

- Large-scale data analytics
 - ML data pipeline processing
-

8. MLOps, Experiment Tracking & Version Control

MLOps Tools

- MLflow

Version Control

- Git
-

9. Cloud Platforms & Deployment

Cloud Providers

- Amazon Web Services
- Google Cloud Platform

Deployment Models

- Cloud-based microservices
 - ML workload deployment
-

10. Containers, Orchestration & Service Mesh

Containerization & Orchestration

- **Kubernetes**

Advanced Deployment

- Service mesh implementations
 - Scalable ML workloads
 - Enterprise-grade orchestration
-

11. NLP & Applied AI Solutions

NLP Tasks

- Text classification
- Sentiment analysis
- Topic modeling

Applied AI Domains

- Chatbots
 - Recommendation systems
 - Translation services
 - Semantic search
 - Natural language understanding
-

12. Data Pipelines & ML Integration

Data Engineering

- Enterprise-grade data pipelines
- Data acquisition and preprocessing

- ML model integration
 - Structured and unstructured data handling
-

13. LLM Optimization & Prompt Engineering

Techniques

- Advanced prompt engineering
 - LLM fine-tuning
 - Semantic retrieval optimization
 - Performance and security tuning
-

14. Healthcare AI (Domain Application)

Domain-Specific AI

- Healthcare AI solutions
 - Task adaptation using domain knowledge
 - Specialized model deployment
-

15. Collaboration, Delivery & Leadership (Technical Context)

Collaboration

- Cross-functional teams (data scientists, product managers)
- Stakeholder engagement

Delivery Skills

- Translating business requirements into ML solutions
 - Coordinating ML experiments and deployments
 - Mentoring team members
-

16. Consolidated Technical Stack Summary

Languages

- Python
- SQL
- Go (basic)
- Rust (basic)

Agentic & GenAI

- LangGraph
- CrewAI
- LangChain
- Hugging Face Transformers
- Prompt Engineering
- LLM Fine-tuning

Backend & APIs

- FastAPI
- REST APIs
- Microservices

Databases

- MongoDB
- Vector Databases

LLM Infrastructure

- vLLM
- GPU Optimization
- Model Quantization

Big Data

- Apache Spark

MLOps & DevOps

- MLflow
- Git
- Kubernetes
- Service Mesh

Cloud

- AWS

- GCP

NLP

- Text Classification
- Sentiment Analysis
- Topic Modeling

Domain

- Healthcare AI

Tab 15

Generative AI Developer / GenAI Engineer

1. Core AI & Generative AI Domains

Primary Focus

- Generative AI (GenAI)
- Large Language Models (LLMs)
- Agentic AI

Architectural Concepts

- Generative AI architecture
 - End-to-end GenAI system design
 - LLM chaining
 - Agentic workflows
 - Production-grade GenAI systems
-

2. Programming Language

- Python
-

3. LLM Frameworks & Agentic AI Frameworks

LLM / Agentic Frameworks

- LangChain
- LangGraph

Agentic AI

- Agentic AI systems
 - Agent-based workflows
-

4. Retrieval-Augmented Generation (RAG)

RAG Capabilities

- RAG implementation
- RAG pipelines
- Retrieval-based LLM augmentation

Related Concepts

- Context grounding
 - Knowledge-augmented generation
-

5. Model Integration & Deployment

Model Engineering

- Model deployment
- LLM integration
- LLM application development

Production Concerns

- Scalability
 - Security
 - Performance
-

6. Generative AI Application Development

Development Scope

- Development of intelligent applications
 - Conversion of business requirements into technical designs
 - Delivery of production-ready GenAI solutions
-

7. LLM Platforms & Providers

LLM Platform

- **OpenAI** (LLM usage explicitly mentioned)
-

8. Software Engineering & Delivery

Engineering Practices

- End-to-end solution ownership (concept → deployment)
 - System scalability and reliability
 - Secure AI application design
-

9. Data Engineering (Preferred / Supporting Skill)

Data Engineering

- Data engineering principles (mentioned as preferred)
-

10. Collaboration & Communication (Technical Context)

Cross-Functional Work

- Collaboration with engineering and business stakeholders
 - Technical communication of GenAI designs and systems
-

11. Experience Context (Technical Relevance)

Experience Range

- 4–8 years total experience
 - Proven delivery of GenAI projects
 - Hands-on GenAI development experience
-

12. Consolidated Technical Stack Summary

Languages

- Python

GenAI & LLM

- Generative AI
- Large Language Models (LLMs)
- Agentic AI
- LLM chaining
- GenAI architecture

Frameworks

- LangChain
- LangGraph

RAG

- RAG implementation
- RAG pipelines

Deployment

- Model deployment
- Production GenAI systems

Platforms

- OpenAI

Tab 16

Generative AI Professional (Legal Services Domain)

1. Core AI & Generative AI Domains

Primary Focus Areas

- Generative AI
- Machine Learning (ML)
- AI-powered application development

Key Activities

- Design, development, and deployment of Generative AI models
 - Training and fine-tuning of AI systems
 - Custom language model and generative model development
 - Optimization of AI-powered applications
-

2. Programming Languages

- **Python**
 - JavaScript (plus)
 - TypeScript (plus)
-

3. Generative AI Frameworks & Tools

GenAI Frameworks / Platforms

- **AI Foundry**
- **Copilot Agents**
- **Azure Assistant**
- **OpenAI** (models / APIs)

Large Language Models

- **Claude** (LLMs)
-

4. Machine Learning Frameworks & Libraries

ML / DL Libraries

- **PyTorch**
 - **TensorFlow**
 - **Scikit-learn**
-

5. Cloud Platforms & Azure AI Services

Cloud Platform

- **Microsoft Azure**

Azure AI Services

- **Azure OpenAI**
- **Azure Cognitive Services**

Cloud Capabilities

- Scalable AI deployment
 - Cloud-based AI solution management
-

6. Web & API Development

Web Technologies

- REST APIs
- HTTP
- Web services

Frontend / UI Frameworks (Plus)

- **React**
 - **Blazor**
-

7. Containerization & DevOps

Containerization

- Docker

Version Control / DevOps

- Azure DevOps
-

8. Model Engineering & Lifecycle Management

Model Engineering

- Fine-tuning large language models
- Custom generative model training
- Model optimization for business use cases

Lifecycle Activities

- Deployment
 - Maintenance
 - Optimization
 - Documentation of models and processes
-

9. Software Engineering Practices

Engineering Responsibilities

- Integration of AI functionalities into applications
 - Documentation of technical solutions
 - Code maintainability and optimization
-

10. Certifications (Preferred)

Microsoft Certifications

- Microsoft Azure AI Engineer

11. Collaboration & Delivery Context (Technical Relevance)

Team Interaction

- Cross-functional collaboration
- Global distributed teams

Operational Focus

- Enterprise-scale AI solutions
 - Technology-driven legal services
-

12. Education & Experience Context (Technical)

Education

- Computer Science
- Artificial Intelligence
- Data Science
- Related technical fields

Experience

- 4–9 years professional experience
 - Minimum 3 years with Generative AI frameworks and tools
-

13. Consolidated Technical Stack Summary

Languages

- Python
- JavaScript
- TypeScript

Generative AI

- AI Foundry
- Copilot Agents

- Azure Assistant
- OpenAI
- Claude LLMs

ML / DL

- PyTorch
- TensorFlow
- Scikit-learn

Cloud

- Azure
- Azure OpenAI
- Azure Cognitive Services

Web & APIs

- REST APIs
- HTTP
- Web Services

Frontend (Plus)

- React
- Blazor

DevOps

- Docker
- Azure DevOps

Tab 17

AI/ML Engineer (Generative AI, NLP, Legacy Modernization)

1. Core AI & ML Domains

Primary Focus Areas

- Artificial Intelligence (AI)
- Machine Learning (ML)
- Generative AI (GenAI)
- Agentic AI
- Natural Language Processing (NLP)

Engineering Scope

- AI-driven automation
 - Legacy system modernization
 - Migration optimization
 - Governance-compliant AI systems
-

2. Programming Languages & Data Skills

Languages

- Python
 - SQL (joins, CTEs, query optimization)
-

3. Machine Learning & Deep Learning Frameworks

ML / DL Frameworks

- PyTorch
 - TensorFlow
-

4. Generative AI & Agentic AI Tooling

GenAI / Agentic Frameworks

- LangChain
- LlamaIndex

LLM Techniques

- Embeddings
 - Prompt engineering
 - Recommendation engines
 - AI agents for automation and risk detection
-

5. NLP Applications & Use Cases

NLP Solutions

- SAS script classification
- Business relevance analysis
- Regulatory impact analysis
- Technical complexity assessment

Text Intelligence

- Metadata extraction
 - Ownership detection
 - Lineage and usage pattern analysis
-

6. Agentic AI Automation

Agent-Based Capabilities

- Detection of redundant scripts
 - Overlap identification
 - Migration risk analysis
 - Migration prioritization recommendations
-

7. Legacy Modernization & Migration

Legacy Technology

- SAS (scripts, master scripts)

Modernization Activities

- Script consolidation
 - Scope reduction
 - Migration workload optimization
 - Large-scale script analysis (70+ SAS master scripts)
-

8. Cloud Platforms & Azure AI Services

Cloud Platform

- Microsoft Azure

Azure AI Services

- Azure AI
- Azure OpenAI
- Azure Machine Learning Studio
- Azure Cognitive Services

Cloud Capabilities

- Scalable AI execution
 - Azure AI sandbox for PoC experimentation
-

9. Governance, Compliance & Risk Management

Governance Domains

- Data governance
- Metadata management
- Audit-ready lineage mapping

Regulatory Standards

- GDPR

- IFRS 17

Compliance Requirements

- Traceable outputs
 - Enterprise governance adherence
-

10. Proof of Concept (PoC) & Validation

PoC Activities

- Azure AI sandbox setup
 - Pilot dataset validation
 - Model validation on enterprise datasets
-

11. Industry Domain Knowledge

Industry Experience

- Insurance
 - Finance
-

12. Collaboration & Delivery Context

Team Collaboration

- Cross-functional engineering teams
 - Stakeholder collaboration
 - Agile, collaborative work culture
-

13. Experience Requirements (Technical Context)

Experience

- 5–10 years overall experience
- 5+ years in AI/ML engineering

- Hands-on GenAI, Agentic AI, and NLP experience
-

14. Organizational Context

Company

- **Claidroid Technologies**

Service Domains

- Enterprise Service Management
 - Enterprise Security
 - ServiceNow
 - IAM (Identity and Access Management)
 - Digital Archiving
-

15. Consolidated Technical Stack Summary

Languages

- Python
- SQL

AI / ML

- Generative AI
- Agentic AI
- NLP
- Recommendation systems
- Embeddings
- Prompt engineering

Frameworks

- PyTorch
- TensorFlow
- LangChain
- LlamaIndex

Cloud & AI Platforms

- Azure AI

- Azure OpenAI
- Azure ML Studio
- Azure Cognitive Services

Legacy Modernization

- SAS
- Script migration & consolidation

Governance & Compliance

- GDPR
- IFRS 17
- Data governance
- Metadata management
- Lineage tracking

Domains

- Insurance
- Finance

Tab 18

Senior Generative AI Engineer

1. Core Technical Domains

Primary Focus Areas

- Generative AI
 - Retrieval-Augmented Generation (RAG)
 - Agentic AI systems
 - Machine Learning Engineering
 - Backend & Microservices Engineering
-

2. Programming Language & Design

Languages

- Python

Engineering Concepts

- Object-Oriented Design (OOD)
 - Microservices architecture
 - Multi-step AI orchestration
 - Tool-calling and external API integration
-

3. Generative AI & Agentic Frameworks

Frameworks

- LangChain
- LlamaIndex

Capabilities

- Agentic planners
- Multi-step reasoning workflows
- Tool-based AI orchestration

- Domain-specific AI “skills” / models
-

4. Retrieval-Augmented Generation (RAG)

RAG Components

- RAG pipeline design
 - RAG optimization
 - Context retrieval and grounding
 - Vector-based retrieval systems
-

5. Vector Databases

Vector Stores

- Pinecone
 - Qdrant
 - pgvector
-

6. API Development & Backend Services

API Technologies

- REST APIs
- API Gateway-based architectures

Backend Patterns

- Scalable microservices
 - External service integration
-

7. Cloud Platform – AWS (Primary)

Core AWS Services

- Amazon Elastic Kubernetes Service (EKS)

- **Amazon Elastic Container Service (ECS)**
 - **AWS Lambda**
 - **Amazon API Gateway**
-

8. AWS AI / ML Stack (Preferred)

Managed AI Services

- **Amazon Bedrock**
 - **Amazon Kendra**
 - **Amazon Textract**
-

9. MLOps & Cloud Infrastructure

Infrastructure & DevOps

- **Docker**
- **Kubernetes**
- **CI/CD pipelines**

Security & Monitoring

- **AWS Secrets Manager**
 - **Amazon CloudWatch**
-

10. Enterprise Integration

Enterprise Platforms

- **NetSuite** (or similar)

Use Cases

- **Integration of AI systems with enterprise business platforms**
 - **Domain-specific data analysis**
-

11. Development Methodology

Ways of Working

- Agile development
 - Rapid iteration
 - High-impact product delivery
 - Production-grade engineering practices
-

12. Experience & Qualification Context (Technical)

Experience

- 6+ years in software or ML engineering
- 2+ years hands-on Generative AI experience
- Proven experience deploying RAG and agentic systems

Education

- Bachelor's degree in:
 - Computer Science
 - Machine Learning
 - Related technical field
-

13. Consolidated Technical Stack Summary

Languages

- Python

GenAI & Agentic AI

- LangChain
- LlamaIndex
- RAG pipelines
- Agentic planners
- Tool calling & orchestration

Vector Databases

- Pinecone

- Qdrant
- pgvector

APIs & Backend

- REST APIs
- Microservices
- API Gateway

AWS Cloud

- EKS
- ECS
- Lambda
- API Gateway
- Bedrock
- Kendra
- Textract

MLOps & DevOps

- Docker
- Kubernetes
- CI/CD
- AWS Secrets Manager
- Amazon CloudWatch

Enterprise Integration

- NetSuite (or similar platforms)

Tab 19

Generative AI Engineer (Multimodal & LLM Systems)

1. Core AI & ML Domains

Primary Focus Areas

- Generative AI
 - Deep Learning
 - Natural Language Processing (NLP)
 - Multimodal AI (text, image, video, audio)
 - Transformer-based architectures
 - Large Language Models (LLMs)
-

2. Programming Languages

- Python
-

3. Transformer Models & Foundation Models

Transformer Architectures

- GPT
- BERT
- T5
- LLaMA

Multimodal & Vision Models

- Stable Diffusion
 - CLIP
-

4. Generative Model Families

Model Types

- Diffusion Models
 - Generative Adversarial Networks (GANs)
 - Variational Autoencoders (VAEs)
-

5. Generative AI Frameworks & Libraries

LLM & GenAI Libraries

- Hugging Face
- OpenAI (APIs / models)
- Custom LLM architectures

Deep Learning Frameworks

- TensorFlow
 - PyTorch
-

6. Multimodal & Applied AI Systems

AI Applications

- AI-powered chatbots
 - Virtual assistants
 - Content generation tools
 - Text, image, video, and audio synthesis systems
-

7. Model Optimization & Alignment

Optimization Goals

- Inference speed
- Cost efficiency
- Performance optimization
- Edge and cloud deployment optimization

Alignment Techniques

- Reinforcement Learning with Human Feedback (RLHF)
-

8. Model Deployment & Serving

Deployment & Interoperability

- ONNX
- MLflow

Model Serving & APIs

- FastAPI
 - Triton Inference Server
 - Ray Serve
-

9. Vector Databases & Retrieval

Vector Databases

- FAISS
- Pinecone
- Weaviate

Retrieval Techniques

- Text embeddings
 - Embedding-based search
 - Retrieval-Augmented Generation (RAG)
-

10. GPU Acceleration & Performance Engineering

Acceleration Technologies

- CUDA
- TensorRT
- ONNX optimization

Training Techniques

- Distributed training
 - Large-scale dataset training
-

11. Cloud Platforms & Managed AI Services

Cloud Providers

- AWS Bedrock
 - Azure OpenAI
 - Google Vertex AI
-

12. Data Engineering & Model Evaluation

Data Handling

- Data preprocessing
- Data annotation
- Large-scale dataset management

Evaluation

- Model evaluation metrics
 - Performance benchmarking
-

13. Security, Ethics & Responsible AI

AI Governance

- Model security
 - Bias mitigation
 - Ethical AI practices
-

14. Development Practices & Collaboration

Ways of Working

- Agile environments
 - DevOps practices
 - Cross-functional collaboration (data scientists, ML engineers, product teams)
-

15. Experience & Qualification Context (Technical)

Experience

- 6+ years in AI, ML, and deep learning
- Hands-on experience with Generative AI and transformer models

Certifications (Preferred)

- Machine Learning certifications
 - AI certifications
 - Cloud AI service certifications
-

16. Consolidated Technical Stack Summary

Languages

- Python

Transformers & Foundation Models

- GPT
- BERT
- T5
- LLaMA
- CLIP
- Stable Diffusion

Generative Models

- Diffusion Models
- GANs
- VAEs

Frameworks & Libraries

- TensorFlow
- PyTorch
- Hugging Face
- OpenAI APIs

Deployment & Serving

- FastAPI
- Triton
- Ray Serve
- ONNX
- MLflow

Vector Databases

- FAISS
- Pinecone
- Weaviate

Acceleration

- CUDA
- TensorRT
- Distributed Training

Cloud AI

- AWS Bedrock
- Azure OpenAI
- Google Vertex AI

Advanced Techniques

- RLHF
- RAG
- Prompt Engineering
- Multimodal AI

Tab 20

Generative AI Developer

1. Core AI & Generative AI Domains

Primary Focus Areas

- Generative AI
- Large Language Models (LLMs)
- Machine Learning (ML)
- Multimodal AI (text, image, multimodal generation)

Engineering Scope

- Design, development, and deployment of AI-driven solutions
 - Scalable AI application development
 - Model optimization for production environments
-

2. Programming Language

- Python
-

3. Generative AI Models & LLMs

Large Language Models

- GPT
- LLaMA
- Falcon

Model Capabilities

- Text generation
 - Image generation
 - Multimodal generation
-

4. Generative AI Frameworks & APIs

Frameworks / Platforms

- **Hugging Face Transformers**
 - **LangChain**
 - **OpenAI (API)**
-

5. Machine Learning & Deep Learning Frameworks

ML / DL Libraries

- **PyTorch**
 - **TensorFlow**
-

6. Prompt Engineering & Model Optimization

Techniques

- Prompt engineering strategies
 - Model fine-tuning
 - Model evaluation
 - Accuracy optimization
 - Latency optimization
 - Cost optimization
-

7. API Integration & Microservices

Backend Integration

- API-based model integration
 - Microservices-based workflows
 - Application and workflow integration
-

8. Data Engineering & Embeddings

Data Handling

- Data preprocessing
- Embedding models

Vector Databases

- Pinecone
 - Weaviate
-

9. Cloud Platforms

Cloud Providers

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform

Cloud Capabilities

- AI model deployment
 - API integration in cloud environments
-

10. Collaboration & Delivery Context (Technical)

Cross-Functional Collaboration

- Data scientists
- Machine learning engineers
- Product teams

Delivery Focus

- AI-driven feature development
 - Production-ready AI systems
-

11. Ethics & Continuous Learning

Practices

- Awareness of AI ethics
 - Staying updated with GenAI and LLM advancements
-

12. Consolidated Technical Stack Summary

Languages

- Python

LLMs & GenAI

- GPT
- LLaMA
- Falcon
- Generative AI
- Multimodal AI

Frameworks & APIs

- Hugging Face Transformers
- LangChain
- OpenAI API

ML / DL

- PyTorch
- TensorFlow

Vector Databases

- Pinecone
- Weaviate

Cloud

- AWS
- Azure
- GCP

Practices

- Prompt Engineering
- Fine-tuning
- Model Evaluation
- API & Microservices Integration

Tab 21

Generative AI / Agentic AI Engineer

1. Core AI & Engineering Domains

Primary Focus Areas

- Generative AI
- Agentic AI
- Large Language Models (LLMs)
- Small Language Models (SLMs)
- Task-Specific Language Models (TLMs)
- Machine Learning (ML)
- AI-driven decision systems

System Characteristics

- Production-grade AI systems
 - Secure, scalable, business-oriented architectures
 - Structured and unstructured data processing
-

2. Programming Languages & Core Libraries

Programming Language

- Python

ML / AI Libraries

- **scikit-learn**
 - **TensorFlow**
 - **PyTorch**
 - **Hugging Face**
-

3. Generative AI & LLM Technologies

Model Types

- Large Language Models (LLMs)
- Small Language Models (SLMs)
- Task-Specific Language Models (TLMs)

GenAI Capabilities

- Online and offline LLM deployment
 - Domain-specific model usage
 - Prompt engineering
 - Model fine-tuning
 - Parameter-efficient tuning methods
-

4. Agentic AI Frameworks & Orchestration

Agentic / Orchestration Frameworks

- **LangGraph**
- **CrewAI**
- **LangChain** (preferred / differentiating)

Agent Capabilities

- Multi-step reasoning
 - Tool-enabled agents
 - Agent orchestration
 - Workflow automation
-

5. Retrieval-Augmented Generation (RAG) & Data Pipelines

RAG Components

- Vector databases
- Retrieval pipelines
- Enterprise knowledge base integration
- External data source integration

Data Pipelines

- Ingestion pipelines
- Preprocessing pipelines

- Indexing pipelines
- Retrieval pipelines

Data Types

- Text
 - Tables
 - Documents
 - Images
-

6. Vector Databases & Embeddings

Vector Databases

- Vector databases (explicitly mentioned, vendor-agnostic)

Embedding Techniques

- Embedding-based retrieval
 - Multimodal embeddings (preferred)
-

7. Model Evaluation, Optimization & Experimentation

Evaluation Techniques

- Evaluation metrics
- A/B testing
- Iterative model improvement cycles

Optimization Areas

- Performance profiling
 - Inference cost optimization
 - Observability
-

8. LLM Safety, Security & Red-Teaming

Red-Teaming & Vulnerability Assessment

- **Garak** (Generative AI Red-teaming & Assessment Kit)

Safety & Governance

- Content filtering systems
 - Moderation systems
 - Bias mitigation
 - Access controls
 - Data privacy compliance
 - LLM vulnerability assessment
-

9. MLOps, Deployment & Platform Engineering

MLOps Practices

- CI/CD
- Model registry
- Model versioning
- Monitoring & observability
- Model scaling

Containerization & Orchestration

- **Docker**
 - **Kubernetes**
-

10. APIs, Microservices & System Integration

Backend Engineering

- API development
 - Microservices architecture
 - Model serving services
-

11. Cloud Platforms & GenAI Services

Cloud Platforms

- **Azure AI Services** (preferred)

- **Amazon Web Services**
 - **Google Cloud Platform**
-

12. Security, Data Governance & Compliance

Governance Domains

- AI security
 - Data governance
 - Privacy best practices
 - Access control policies
 - Compliance in AI deployments
-

13. Multimodal & Advanced AI (Preferred)

Advanced Capabilities

- Multimodal models
 - Retrieval-augmented multimodal pipelines
 - Human-in-the-loop systems
 - Agentic safety
 - Tool-use constraints
 - LLM application firewalls
-

14. Software Engineering Lifecycle

Lifecycle Coverage

- Architecture
 - Development
 - Testing
 - Deployment
 - Monitoring
 - Iterative improvement
-

15. Experience Context (Technical)

Experience

- 4–6 years in software engineering / AI roles
 - 2–3 years hands-on with Generative AI & LLMs
 - Production deployment experience
 - 0–30 days joiner preference
-

16. Consolidated Technical Stack Summary

Languages

- Python

GenAI & LLM

- LLMs
- SLMs
- TLMs
- Prompt Engineering
- Fine-tuning
- Parameter-efficient tuning
- RAG

Agentic AI

- LangGraph
- CrewAI
- LangChain
- Tool-enabled agents
- Multi-step workflows

ML Frameworks

- scikit-learn
- TensorFlow
- PyTorch
- Hugging Face ecosystem

Data & Retrieval

- Vector databases
- Structured & unstructured data
- Multimodal data pipelines

Safety & Security

- Garak
- Content filtering
- Moderation
- Bias mitigation
- Privacy & access control

MLOps & Deployment

- Docker
- Kubernetes
- CI/CD
- Model registry
- Observability

Cloud

- Azure AI Services
- AWS
- GCP

Tab 22

Senior Computer Vision Engineer

1. Core AI & Computer Vision Domains

Primary Focus Areas

- Computer Vision
- Deep Learning
- Video Analytics
- Real-time inference systems
- Large-scale production pipelines

Use Cases

- Detection
 - Classification
 - Segmentation
 - Live video stream processing (RTSP)
-

2. Deep Learning Frameworks

Training & Inference Frameworks

- TensorFlow
- PyTorch

Model Types

- CNN-based architectures
 - Detection models
 - Classification models
 - Segmentation models
-

3. Model Serving & Inference Infrastructure

Model Servers

- **NVIDIA Triton Inference Server** (preferred)
- **TensorFlow Serving**
- **TorchServe**

Inference Characteristics

- Live video feed inference
 - High-throughput, low-latency processing
 - Scalable inference architectures
-

4. Model Optimization & NVIDIA Stack

Optimization Techniques

- Model optimization for performance and throughput
- TensorRT conversion for:
 - TensorFlow models
 - PyTorch models

Precision & Quantization

- FP16 / Mixed Precision inference
- INT8 optimized inference
- INT8 Quantization Aware Training (QAT) (*good to have*)
- TensorFlow32 (*good to have*)
- BFloat16 (*good to have*)

NVIDIA Ecosystem

- **NVIDIA** hardware
 - **TensorRT**
 - **NVIDIA Docker**
 - **NVIDIA NGC**
-

5. Data Engineering & Dataset Techniques

Dataset Techniques

- Dataset augmentation
- Synthetic dataset generation
- State-of-the-art (SOTA) augmentation techniques

Training Pipelines

- Re-training pipelines
 - Continuous model improvement with incoming data
 - Iterative accuracy improvement loops
-

6. End-to-End Pipeline Engineering

Pipeline Scope

- End-to-end CV pipelines
- Scalable production pipelines
- Training → inference → retraining workflows

Deployment Environments

- Cloud deployment
 - Edge deployment
 - Hybrid cloud-edge architectures
 - Large-scale deployments
-

7. Containerization & Deployment

Containers

- Docker
- NVIDIA Docker
- NVIDIA NGC

Scalable Architectures (Good to Have)

- Kubernetes
 - Apache Kafka
-

8. Algorithms & Theoretical Foundations

Core Concepts

- Convolutional Neural Networks (CNNs)

- Regularization techniques
- Deep learning fundamentals

Computer Science Foundations

- Data Structures
 - Algorithms
-

9. Programming & Version Control

Version Control Systems

- **Git**
 - **Mercurial**
 - **Subversion**
-

10. Front-End & UI Tooling (Supporting Skills)

Frontend Tooling

- **Babel**
- **Webpack**
- **npm**

State Management

- **Redux**
-

11. Collaboration & Development Practices

Engineering Practices

- Code reviews
- Design discussions
- Documentation
- Data-driven decision making
- Ownership-driven development

Agile Practices

- Daily standups
 - Team meetings
-

12. Communication & Collaboration Tools

Internal Tools

- **Microsoft Teams**
-

13. Work Environment & Operating Model

Work Model

- 100% remote
 - Distributed global teams
-

14. Consolidated Technical Stack Summary

Computer Vision & DL

- Detection, Classification, Segmentation
- CNNs
- TensorFlow
- PyTorch

Inference & Optimization

- Triton Inference Server
- TensorFlow Serving
- TorchServe
- TensorRT
- FP16 / Mixed Precision
- INT8 / QAT
- TF32, BFloat16

Deployment

- RTSP video inference
- Cloud & Edge deployment

- Docker
- NVIDIA Docker
- NVIDIA NGC
- Kubernetes (*good to have*)
- Kafka (*good to have*)

Data & Pipelines

- Dataset augmentation
- Synthetic data generation
- Retraining pipelines
- End-to-end CV pipelines

Engineering Foundations

- Data Structures & Algorithms
- Version control (Git, Mercurial, SVN)

Frontend (Supporting)

- Babel
- Webpack
- npm
- Redux

Tab 23

Lead Agentic AI Engineer (Enterprise)

1. Core AI & Engineering Domains

Primary Focus Areas

- Agentic AI systems
- Artificial Intelligence (AI)
- Machine Learning (ML)
- Predictive analytics
- Anomaly detection
- Recommendation systems
- Real-time data analysis
- Enterprise decision automation

System Characteristics

- Production-grade, scalable agentic systems
 - Autonomous workflow orchestration
 - Event-driven and serverless architectures
 - Enterprise integration and governance
-

2. Programming Languages & Low-Code Platforms

Languages

- Python

Low-Code / Automation Platforms

- Power Apps
 - Power Automate
-

3. Agentic AI Platforms & Tooling

AI Platforms

- **Microsoft AI Foundry** (highly preferred)
- **Copilot Studio** (optional)

Agent Capabilities

- Autonomous workflow orchestration
 - Decision-making agents across enterprise domains (finance, sales, operations)
-

4. Cloud, Serverless & Workflow Orchestration (Azure)

Serverless & Integration

- **Azure Functions**
 - **Azure Logic Apps**
 - **Power Automate**
-

5. Machine Learning & Data Platforms

ML Platforms

- **Azure Machine Learning**
- **Databricks**

Data Platforms

- **Snowflake**

ML Use Cases

- Predictive analytics
 - Anomaly detection
 - Recommendations
-

6. Enterprise Systems & Integration

Enterprise Platforms

- **SAP HANA**
- **Salesforce**
- **Kafka**
- **UiPath**

Integration Scope

- Real-time and batch integrations
 - Event-driven enterprise workflows
-

7. Observability, Monitoring & Reliability

Observability Tools

- **Azure Monitor**

Operational Capabilities

- Logging
 - Alerting
 - Reliability engineering
 - Production observability
-

8. DevOps, CI/CD & Source Control

DevOps & CI/CD

- **Azure DevOps**
- **GitHub**

Practices

- CI/CD pipelines
 - Model lifecycle management
 - Reusable architecture components
 - Governance standards
-

9. Data & Retrieval Infrastructure

Data Technologies

- Vector Databases (explicitly mentioned, vendor-agnostic)
-

10. Vision, Edge AI & IoT

IoT Platform

- Azure IoT Hub

Computer Vision & Edge AI

- OpenCV
- TensorFlow
- PyTorch

Use Cases

- Vision-based agents
 - Edge AI deployments
 - IoT-driven intelligence
-

11. Architecture, Governance & Engineering Leadership

Architecture Responsibilities

- Architecture patterns
- Reusable components
- Enterprise governance standards

Leadership Activities

- Code reviews
 - Technical design sessions
 - Agile ceremonies
 - Delivery ownership
 - Quality benchmarks
-

12. Cross-Functional Collaboration (Technical Context)

Collaboration With

- Data Scientists & ML Engineers
 - Product Managers & UX Designers
 - Enterprise Architects & DevOps Engineers
 - Security & Compliance Teams
 - Business Stakeholders & Subject Matter Experts
 - Change Management & Enablement Teams
-

13. Strategic & Emerging Technology Awareness

Emerging Domains

- Machine vision
 - Internet of Things (IoT)
 - Enterprise AI strategy
-

14. Consolidated Technical Stack Summary

Languages & Platforms

- Python
- Power Apps
- Power Automate

Agentic AI

- Microsoft AI Foundry
- Copilot Studio

Serverless & Workflow

- Azure Functions
- Azure Logic Apps

ML & Data

- Azure Machine Learning

- Databricks
- Snowflake

Enterprise Integration

- SAP HANA
- Salesforce
- Kafka
- UiPath

DevOps & Observability

- Azure DevOps
- GitHub
- Azure Monitor

Data & Retrieval

- Vector Databases

Vision & IoT

- Azure IoT Hub
- OpenCV
- TensorFlow
- PyTorch

Tab 24

Lead Agentic AI Engineer (Enterprise)

1. Core AI & Engineering Domains

Primary Focus Areas

- Agentic AI systems
- Artificial Intelligence (AI)
- Machine Learning (ML)
- Predictive analytics
- Anomaly detection
- Recommendation systems
- Real-time data analysis
- Enterprise decision automation

System Characteristics

- Scalable, production-grade agentic AI systems
 - Autonomous workflow orchestration
 - Event-driven and serverless architectures
 - Enterprise-grade governance and reliability
-

2. Programming Languages & Platforms

Languages

- Python

Low-Code / Automation Platforms

- Power Apps
 - Power Automate
-

3. Agentic AI Platforms

Agent Development Platforms

- **Microsoft AI Foundry** (key platform)
- **Copilot Studio** (optional)

Agent Capabilities

- Autonomous orchestration of workflows
 - Embedded decision-making across finance, sales, and operations
-

4. Cloud, Serverless & Workflow Orchestration (Azure)

Serverless & Integration Services

- **Azure Functions**
 - **Azure Logic Apps**
 - **Power Automate**
-

5. Machine Learning & Data Platforms

ML & Data Platforms

- **Azure Machine Learning**
- **Databricks**
- **Snowflake**

ML Use Cases

- Predictive analytics
 - Anomaly detection
 - Recommendation systems
-

6. Enterprise Systems & Integration

Enterprise Platforms

- **SAP HANA**
- **Salesforce**
- **Kafka**

- **UiPath**

Integration Scope

- Real-time and event-driven enterprise workflows
-

7. Observability, Monitoring & Reliability

Observability Tools

- **Azure Monitor**

Operational Capabilities

- Logging
 - Alerting
 - Reliability engineering
 - Production observability
-

8. DevOps, CI/CD & Source Control

DevOps & CI/CD Tools

- **Azure DevOps**
- **GitHub**

Practices

- CI/CD pipelines
 - Model lifecycle management
 - Reusable architecture components
 - Governance standards
-

9. Data & Retrieval Infrastructure

Data Technologies

- Vector Databases (explicitly mentioned, vendor-agnostic)
-

10. Vision, Edge AI & IoT

IoT Platform

- Azure IoT Hub

Computer Vision & Edge AI

- OpenCV
- TensorFlow
- PyTorch

Use Cases

- Vision-based agents
 - Edge AI deployments
 - IoT-enabled intelligence
-

11. Architecture, Governance & Engineering Leadership

Architecture Responsibilities

- Definition of architecture patterns
- Reusable component design
- Enterprise governance standards

Leadership Activities

- Code reviews
 - Technical design sessions
 - Agile ceremonies
 - Delivery ownership
 - Quality benchmarks
-

12. Cross-Functional Collaboration (Technical Context)

Collaboration With

- Data Scientists & ML Engineers
 - Product Managers & UX Designers
 - Enterprise Architects & DevOps Engineers
 - Security & Compliance Teams
 - Business Stakeholders & Subject Matter Experts
 - Change Management & Enablement Teams
-

13. Strategic & Emerging Technology Awareness

Emerging Domains

- Artificial Intelligence advancements
 - Machine vision
 - Internet of Things (IoT)
-

14. Consolidated Technical Stack Summary

Languages & Platforms

- Python
- Power Apps
- Power Automate

Agentic AI

- Microsoft AI Foundry
- Copilot Studio

Serverless & Workflow

- Azure Functions
- Azure Logic Apps

ML & Data

- Azure Machine Learning
- Databricks
- Snowflake

Enterprise Integration

- SAP HANA

- Salesforce
- Kafka
- UiPath

DevOps & Observability

- Azure DevOps
- GitHub
- Azure Monitor

Data & Retrieval

- Vector Databases

Vision & IoT

- Azure IoT Hub
- OpenCV
- TensorFlow
- PyTorch

Tab 25

AI Software Engineer (Agentic AI)

1. Core AI & Engineering Domains

Primary Focus Areas

- Agentic AI
- Generative AI
- Large Language Models (LLMs)

Application Scope

- Agentic AI application development
 - AI-driven user experience and business solutions
 - Secure, scalable, and sustainable AI systems
-

2. Programming Language

- Python
-

3. Generative AI & Agentic AI Frameworks

Frameworks & Toolkits

- **LangChain**
- Agentic AI frameworks (general category, explicitly mentioned)

Capabilities

- Agent-based workflows
 - AI orchestration
 - LLM-based application design
-

4. Prompt Engineering

Techniques

- Prompt design
 - Prompt optimization
 - Prompt performance tuning for AI applications
-

5. Machine Learning & AI Libraries

AI / ML Libraries

- TensorFlow
 - PyTorch
 - OpenAI (API)
-

6. Cloud Platforms & Deployment

Cloud Providers

- Microsoft Azure
- Amazon Web Services
- Google Cloud Platform

Cloud Responsibilities

- Deployment of AI models
 - Cloud-based AI model management
 - Integration of AI applications with cloud services
-

7. Software Engineering Practices

Engineering Activities

- Code reviews
- Testing
- Debugging
- Performance optimization
- Secure coding practices

Development Methodology

- Software Development Life Cycle (SDLC)
-

8. AI Lifecycle Management

Lifecycle Stages

- Model development
 - Model deployment
 - Model maintenance
 - Continuous improvement of AI systems
-

9. Quality, Security & Reliability

Quality Assurance

- High-quality code standards
- Sustainable AI solution design

Security

- Secure coding practices
 - Reliable AI system operation
-

10. Education & Experience Context (Technical)

Education

- Bachelor's or Master's degree in:
 - Computer Science
 - Artificial Intelligence
 - Related technical fields

Experience

- Minimum 3 years of experience with:
 - Generative AI
 - Large Language Models
 - LangChain

11. Consolidated Technical Stack Summary

Languages

- Python

GenAI & Agentic AI

- Large Language Models (LLMs)
- Prompt Engineering
- LangChain
- Agentic AI frameworks

AI / ML Libraries

- TensorFlow
- PyTorch
- OpenAI API

Cloud

- Azure
- AWS
- Google Cloud Platform

Practices

- SDLC
- Secure coding
- Code reviews
- Testing & debugging
- AI lifecycle management

Tab 26

RAG + Agentic AI Engineer

1. Core AI & System Domains

Primary Focus Areas

- Retrieval-Augmented Generation (RAG)
- Agentic AI
- Large Language Model (LLM) systems
- Enterprise AI applications
- Governed & secure AI systems

System Characteristics

- Production-grade AI systems
 - Multi-agent orchestration
 - Enterprise-ready, multi-tenant architectures
 - Governed, auditable, and secure AI pipelines
-

2. Programming Language & Backend Foundations

Languages

- Python

Backend Frameworks

- **FastAPI**

Backend Capabilities

- Asynchronous programming
 - AI microservices
 - Caching strategies
 - Async execution models
-

3. RAG Engineering (Core)

RAG Pipeline Components

- Text / component-aware chunking
- Hybrid retrieval
- Reranking
- Query rewriting
- Multi-hop retrieval
- Graph-RAG (basic concepts)

Retrieval Techniques

- BM25 (sparse retrieval)
 - Dense vector retrieval
 - Hybrid (BM25 + dense)
-

4. Agentic AI & Orchestration

Agentic Frameworks

- **LangGraph**

Agent Capabilities

- Multi-agent orchestration
- Planning-based agents
- Tool-calling workflows
- Self-healing agent behaviors

Agent Tooling

- MCP tools
-

5. Vector Databases & Search Systems

Vector Databases

- **Qdrant**
- **Weaviate**

Search Engines

- **OpenSearch**

- Elasticsearch

Indexing & Optimization

- HNSW tuning
-

6. LLM Operations & Evaluation

LLM Ops Capabilities

- Prompt engineering
 - Hallucination reduction
 - Automated evaluation frameworks
 - RAG and agent evaluation
 - Synthetic data generation
-

7. Document & Data Engineering

Document Ingestion Pipelines

- PDF
- DOCX
- OCR
- Images

Data Engineering Concepts

- Document parsing
 - ETL pipelines
 - Batching
 - Streaming
 - Caching
-

8. Knowledge Modeling & Enterprise Data

Knowledge Layer Design

- Ontology-driven knowledge layers
- Multi-tenant knowledge architectures

- Enterprise data modeling
-

9. Enterprise Integration

Enterprise Systems

- SharePoint
- CRM systems
- Databases
- MCP tools

Integration Scope

- LLM system integration with enterprise applications
-

10. Security, Governance & Compliance

Security & Governance

- PII handling
 - Access control
 - Audit trails
 - Governance requirements
 - Secure AI system design
-

11. Infrastructure, DevOps & Deployment

Containerization & DevOps

- **Docker**
- CI/CD pipelines

Infrastructure

- Cloud fundamentals (explicitly mentioned as “cloud basics”)
-

12. Optional / Nice-to-Have Technologies

RAG & LLM Frameworks

- Haystack
- LlamaIndex

Advanced Capabilities

- Retriever fine-tuning
 - Small LLM fine-tuning
 - Ontology modeling
 - Enterprise search deployments
-

13. Application Domains (Nice-to-Have)

Agent Deployment Domains

- Customer support
 - Operations
 - Legal
 - Engineering
-

14. Collaboration & Delivery Context

Collaboration

- Solution architects
- Domain subject-matter experts (SMEs)

Delivery Focus

- Translating enterprise use cases into production AI systems
 - Ownership of AI system design and implementation
-

15. Experience Context

Experience Requirement

- 5+ years overall experience
-

16. Consolidated Technical Stack Summary

Languages & Backend

- Python
- FastAPI
- Async programming

RAG

- Chunking
- Hybrid retrieval (BM25 + dense)
- Reranking
- Query rewriting
- Graph-RAG
- Multi-hop retrieval

Agentic AI

- LangGraph
- MCP tools
- Multi-agent orchestration
- Tool-calling workflows
- Self-healing agents

Vector & Search

- Qdrant
- Weaviate
- OpenSearch
- Elasticsearch
- HNSW tuning

LLM Ops

- Prompt engineering
- Hallucination control
- Evaluation frameworks
- Synthetic data generation

Data Engineering

- Document ingestion (PDF, DOCX, OCR, images)
- ETL pipelines
- Streaming & batching
- Caching

Security & Governance

- PII handling
- Access control
- Audit trails
- Enterprise governance

DevOps

- Docker
- CI/CD
- Cloud basics

Tab 27

Agentic AI Engineer (2–3 Years)

1. Core AI & System Domains

Primary Focus Areas

- Agentic AI
- Large Language Models (LLMs)
- Multi-agent systems
- Production-grade AI infrastructure

System Characteristics

- Scalable, reliable agentic systems
 - Concurrency-aware execution
 - Robust error handling
 - End-to-end ownership (architecture → production)
-

2. Programming & Engineering Scope

Engineering Focus

- Full-stack agentic AI engineering
 - Core agent orchestration infrastructure
 - Execution, concurrency, and reliability challenges
 - Production deployment at scale
-

3. Agentic AI Frameworks & SDKs (Production Experience Required)

Agentic Frameworks / SDKs

- Lyrz
- LangChain
- LangGraph
- CrewAI

- **Google ADK**
- **Claude Agent SDK**

Capabilities

- Agent orchestration
 - Multi-agent workflows
 - Tool-enabled agents
 - Production-grade agent execution
-

4. Large Language Models (LLMs)

LLM Usage

- Hands-on LLM-based system development
 - Production use of LLMs within agent frameworks
 - Dedicated LLM experience (1.5+ years)
-

5. Infrastructure & Reliability Engineering

Infrastructure Challenges

- Agent execution management
- Concurrency control
- Error handling and recovery
- Reliability at scale

Ownership Model

- End-to-end responsibility for system design and delivery
 - Direct collaboration with leadership on undefined problem spaces
-

6. Production & Delivery Experience

Production Expectations

- Shipped agentic AI systems
- Real-world usage by end users
- Systems that operate reliably at scale

7. Experience Requirements (Technical Context)

Experience

- 2–3 years overall experience
- 1.5+ years hands-on experience with:
 - LLMs
 - Agent frameworks
- Demonstrated production deployments

8. Optional / Differentiating Experience

Nice-to-Have

- Startup or founding experience
- Open-source contributions to AI or agent frameworks
- Experience in AI-native or Y Combinator-backed environments

9. Collaboration & Work Style (Technical Context)

Working Model

- High ownership
- High agency
- Lean, fast-moving team
- Solving problems without predefined playbooks

10. Consolidated Technical Stack Summary

Agentic AI

- Lyzr
- LangChain
- LangGraph
- CrewAI
- Google ADK

- Claude Agent SDK

Core Capabilities

- Agent orchestration
- Multi-agent workflows
- LLM-powered systems
- Production reliability
- Concurrency & error handling

Engineering Focus

- Full-stack agentic AI systems
- Infrastructure-level problem solving
- End-to-end production ownership

Tab 28

Agentic AI Developer (Capgemini)

1. Core AI & System Domains

Primary Focus Areas

- Agentic AI
- Generative AI
- Large Language Models (LLMs)
- Autonomous AI agents
- Multi-agent systems

Core Capabilities

- Planning, reasoning, and acting agents
 - Goal-driven and adaptive behavior
 - Iterative execution and feedback loops
 - Real-time strategy adaptation
-

2. Programming Languages

Mandatory

- Python

Optional

- JavaScript
 - C#
-

3. Agentic AI & Orchestration Frameworks

Agentic Frameworks

- LangChain
- LangGraph
- AutoGen

- **CrewAI**

Agent Capabilities

- Task planning and execution
 - Agent collaboration and delegation
 - Multi-agent coordination
 - Strategy adaptation based on feedback
-

4. Large Language Models (LLMs)

LLMs

- **GPT-4**
- **GPT-4o**
- **Claude**
- **LLaMA**
- **Mistral**
- **Gemini**

LLM Techniques

- Prompt engineering
 - Fine-tuning
 - Retrieval-Augmented Generation (RAG)
-

5. Generative AI & ML Frameworks

Frameworks

- **Hugging Face Transformers**
-

6. Retrieval & Vector Databases

Vector Databases

- **FAISS**
- **Pinecone**
- **ChromaDB**

- **Weaviate**
-

7. API, UI & Application Development

Backend / API Frameworks

- **FastAPI**
- **Flask**

UI / App Framework

- **Streamlit**

Integration Capabilities

- External APIs
 - Databases
 - Web applications
 - Autonomous system interaction
-

8. Cloud Platforms & Managed AI Services

Cloud Providers

- **Microsoft Azure**
- **Amazon Web Services**
- **Google Cloud Platform**

Cloud AI Services

- **Azure OpenAI**
 - **Azure ML Studio**
 - **AWS Bedrock**
 - **Amazon SageMaker**
 - **Google Vertex AI**
-

9. Monitoring, Observability & Reliability

Monitoring Tools

- **LangSmith**
- **Grafana**
- **Splunk**

Monitoring Scope

- Agent behavior tracking
 - Performance monitoring
 - Reliability analysis
-

10. DevOps, CI/CD & Infrastructure

DevOps Tools

- **Git**
- **Azure DevOps**

Containerization & Orchestration

- **Docker**
 - **Kubernetes**
-

11. Data Privacy, Security & Governance

Governance Requirements

- Data privacy compliance
 - Security standards adherence
 - Documentation of AI systems
 - Secure autonomous execution
-

12. AI System Design & Algorithms

Core Concepts

- Autonomous systems
- Planning algorithms
- Multi-agent coordination
- Tool-enabled AI agents

13. Collaboration & Delivery Context

Cross-Functional Collaboration

- Data scientists
- Architects
- Product teams

Delivery Focus

- Business-aligned agentic solutions
 - Well-documented, production-ready systems
-

14. Experience & Education Context (Technical)

Experience

- 2–13 years total experience
- 1–2+ years in Generative AI or Agentic AI

Education

- Bachelor's or Master's degree in:
 - Computer Science
 - Artificial Intelligence
 - Data Science
 - Related technical field
-

15. Consolidated Technical Stack Summary

Languages

- Python
- JavaScript
- C#

Agentic AI

- LangChain

- LangGraph
- AutoGen
- CrewAI

LLMs

- GPT-4
- GPT-4o
- Claude
- LLaMA
- Mistral
- Gemini

RAG & Retrieval

- FAISS
- Pinecone
- ChromaDB
- Weaviate

Frameworks

- Hugging Face Transformers

APIs & UI

- FastAPI
- Flask
- Streamlit

Cloud

- Azure (OpenAI, ML Studio)
- AWS (Bedrock, SageMaker)
- GCP (Vertex AI)

DevOps & Infra

- Git
- Azure DevOps
- Docker
- Kubernetes

Monitoring

- LangSmith
- Grafana

- Splunk

Tab 29

AI / ML Engineer

1. Core AI & ML Domains

Primary Focus Areas

- Artificial Intelligence (AI)
- Machine Learning (ML)
- Deep Learning
- Natural Language Processing (NLP)

Engineering Scope

- Design, development, and deployment of AI/ML models
 - AI-driven problem solving aligned with business needs
 - Research and adoption of emerging AI technologies
-

2. Programming Languages

Languages

- Python
 - Java
 - C++
-

3. Data Engineering & Pipelines

Data Engineering Responsibilities

- Development of large-scale data pipelines
 - Data architecture design
 - Maintenance of data processing systems
-

4. Cloud Platforms

Cloud Providers

- Amazon Web Services
- Google Cloud Platform

Cloud Capabilities

- Cloud-based AI model deployment
 - Scalable data and compute infrastructure
-

5. Containerization & Deployment

Containerization

- Docker

Use Cases

- Packaging AI applications
 - Consistent deployment environments
-

6. Software Engineering Practices

Engineering Activities

- Code reviews
- High-quality code standards
- Debugging and troubleshooting complex AI systems

Development Practices

- Collaborative development
 - Cross-functional teamwork
-

7. Research & Continuous Learning

Research Focus

- Staying current with AI, ML, and NLP advancements
 - Evaluating new tools, frameworks, and techniques
-

8. Collaboration & Delivery Context

Team Interaction

- Cross-functional teams
- Business stakeholders
- Engineering collaboration

Problem-Solving

- Translating business problems into AI solutions
 - Resolving technical issues in AI applications
-

9. Consolidated Technical Stack Summary

Languages

- Python
- Java
- C++

AI & ML

- Machine Learning
- Deep Learning
- Natural Language Processing

Cloud

- AWS
- Google Cloud Platform

Deployment

- Docker

Engineering Practices

- Data pipeline architecture

- Code reviews
- Troubleshooting
- Collaborative development

Tab 30

Agent & Prompt Engineering Subject Matter Expert (SME)

1. Core AI & Engineering Domains

Primary Focus Areas

- Agentic AI
- Prompt Engineering
- Generative AI
- Natural Language Processing (NLP)
- Machine Learning (ML)

Business Context

- B2B enterprise AI adoption
 - Marketing technology (MarTech)
 - AI-driven automation and decision support
 - Change management and user enablement
-

2. Agent Engineering & Agentic AI

Agent Capabilities

- Goal definition and capability design
- Decision-making logic
- Agent input/output design
- Agent interaction mechanisms
- Agent learning and adaptation
- Agent performance monitoring
- Production deployment of agents

Agent Architectures

- Single agents
- Common reusable agents
- Multi-agent workflows

- Agent orchestration for complex multi-tasking

Agent Lifecycle

- Agent configuration (onboarding & offboarding)
- Agent evaluation datasets
- Agent testing
- Continuous improvement based on data insights

Knowledge Representation

- Knowledge bases
 - Ontologies for AI agents
-

3. Prompt Engineering & Linguistic AI

Prompt Engineering Use Cases

- Text generation
- Code generation
- Data analysis
- Chatbots
- Virtual assistants
- Marketing content (articles, scripts, social media posts)

Prompt Engineering Practices

- Prompt optimization
- Prompt repositories
- Prompt best practices (beginner → expert)
- Prompt performance analysis
- Prompt-driven insight extraction

Linguistic Foundations

- Syntax
 - Semantics
 - Language understanding
-

4. AI / ML Frameworks & Platforms

Agent Frameworks & Platforms

- **OpenAI Gym**
 - **TensorFlow Agents**
 - **Rasa**
-

5. Programming Languages

Languages

- **Python**
 - **Java**
 - **C++**
-

6. AI & ML Foundations

Concepts & Techniques

- AI algorithms
 - Machine learning techniques
 - Natural language processing techniques
 - Knowledge representation
 - Agent learning mechanisms
 - Data-driven optimization
-

7. Data Analysis & Knowledge Systems

Data Capabilities

- Data analysis
- Data mining
- Knowledge representation
- Large dataset analysis
- Insight extraction

Measurement & Analytics

- Agent effectiveness measurement
- Operational efficiency metrics
- Data-led performance insights

8. Deployment & Production Systems

Deployment Activities

- Production deployment of AI agents
 - Monitoring agent performance
 - Integration with existing systems
 - SaaS platform integration
-

9. Change Management & Enablement (Technical Context)

Enablement Assets

- Playbooks
- Videos
- Reference guides
- Learning materials
- Prompt engineering documentation

Change Management

- User adoption strategies
 - Product backlog feedback
 - Continuous improvement loops
-

10. Collaboration & Delivery

Cross-Functional Collaboration

- Product development teams
- Customer success teams
- Sales teams
- Enablement teams
- Enterprise B2B clients

Communication

- Stakeholder influence
 - Technical-to-non-technical translation
 - End-user training and education
-

11. SaaS & Enterprise Systems

Platforms

- B2B SaaS platforms
 - Enterprise AI solutions
 - Marketing technology platforms
-

12. Project & Program Management

Management Skills

- Complex project management
 - Change management initiatives
 - Delivery tracking
 - Success measurement
 - Multi-priority handling
-

13. Education & Experience Context (Technical)

Education

- Computer Science
- Linguistics
- Data Science
- Related technical fields

Experience

- AI
- Machine Learning
- Natural Language Processing
- Prompt engineering
- AI agent development

14. Consolidated Technical Stack Summary

Agentic AI

- Agent architecture design
- Multi-agent orchestration
- Agent evaluation & testing
- Knowledge bases & ontologies

Prompt Engineering

- Prompt optimization
- Prompt repositories
- Prompt best practices
- Linguistic modeling

Frameworks & Platforms

- OpenAI Gym
- TensorFlow Agents
- Rasa

Languages

- Python
- Java
- C++

AI / ML

- Machine Learning
- NLP
- AI algorithms

Data

- Data analysis
- Data mining
- Knowledge representation

Enterprise & SaaS

- B2B SaaS platforms
- AI-driven marketing systems

Tab 31

Agentic AI Engineer

1. Core AI & Engineering Domains

Primary Focus Areas

- Agentic AI
- Generative AI (GenAI)
- Large Language Models (LLMs)
- Applied Machine Learning (ML)
- Retrieval-Augmented Generation (RAG)

System Characteristics

- Autonomous workflows
 - Model-driven services
 - Scalable, production-grade AI systems
 - Full GenAI lifecycle ownership
-

2. Programming & ML Foundations

Languages

- Python

Foundational Knowledge

- Data handling
 - Machine learning pipelines
 - Deep-learning concepts
-

3. Generative & Agentic AI Frameworks

Agentic / LLM Frameworks

- LangChain
- LangGraph

- **CrewAI**
- **DSPy**

Capabilities

- Intelligent system orchestration
 - Autonomous agent workflows
 - Fine-tuned and structured model pipelines
-

4. Model Engineering & Prompt Design

Model Engineering

- LLM integration and customization
- Lightweight model tuning workflows
- Contextual accuracy, speed, and cost optimization

Prompt Engineering

- Structured prompt design
 - Prompt templates
 - Prompt evaluation logic
-

5. Retrieval-Augmented Generation (RAG) & Vector Pipelines

Embedding & Retrieval Pipelines

- Embedding pipeline development
- Retrieval optimization
- Collaboration with data engineering teams

Vector Databases & Search

- **Milvus**
 - **Pinecone**
 - **ChromaDB**
 - **FAISS**
 - **Azure AI Search**
-

6. API Development & Service Deployment

API & Backend Frameworks

- REST APIs
- FastAPI

Deployment Practices

- Containerization
 - Cloud deployment
 - Service scalability
-

7. Cloud Platforms & Managed GenAI Services

Cloud AI Platforms

- Azure OpenAI
 - AWS Bedrock
 - Google Vertex AI
-

8. Containers, Orchestration & CI/CD

Containerization & Orchestration

- Docker
- Kubernetes

CI/CD & DevOps

- GitHub
 - Azure DevOps
-

9. MLOps, Observability & Monitoring

MLOps & Tracking

- MLflow
- Arize Phoenix

- **Weights & Biases**

Capabilities

- Model versioning
 - Monitoring
 - Observability
-

10. Prompt & Model Evaluation Frameworks

Evaluation Frameworks

- **AgentEval**
- **DeepEval**

Evaluation Scope

- Quantitative evaluation
 - Qualitative evaluation
 - Prompt template testing
-

11. AI-Assisted Development Tools (Nice to Have)

Developer Tools

- **Cursor**
 - **Claude Code**
-

12. Responsible AI & Governance (Nice to Have)

Governance Concepts

- Responsible AI
 - Data privacy
 - Model governance
-

13. Collaboration & Delivery Context

Collaboration

- Data engineers
- ML engineers
- Cross-functional AI teams

Delivery Focus

- Production-grade AI systems
 - Enterprise-ready retrieval and agentic solutions
-

14. Experience & Qualification Context (Technical)

Education

- Bachelor's degree in:
 - Computer Science
 - Artificial Intelligence
 - Machine Learning
 - Related technical fields

Experience

- Hands-on GenAI and Agentic AI system development
 - LLM-based assistants or retrieval systems (preferred)
-

15. Consolidated Technical Stack Summary

Languages

- Python

Agentic & GenAI

- LangChain
- LangGraph
- CrewAI
- DSPy
- LLM integration & tuning

- RAG pipelines

Vector & Retrieval

- Milvus
- Pinecone
- ChromaDB
- FAISS
- Azure AI Search

Backend & APIs

- FastAPI
- REST services

Cloud & Infra

- Azure OpenAI
- AWS Bedrock
- Vertex AI
- Docker
- Kubernetes

MLOps

- MLflow
- Arize Phoenix
- Weights & Biases

Evaluation

- AgentEval
- DeepEval

Tab 32

Artificial Intelligence Engineer (LLM / Agentic AI)

1. Core AI & Engineering Domains

Primary Focus Areas

- Generative AI
- Large Language Models (LLMs)
- Agentic AI
- Retrieval-Augmented Generation (RAG)
- AI-powered backend services

System Characteristics

- LLM-powered workflows
 - Multi-step, stateful agents
 - Tool/API calling agents
 - Production-grade, observable AI systems
-

2. Programming Languages

- Python
-

3. LLMs & Providers

Large Language Models

- OpenAI (LLMs)
 - Claude
 - LLaMA
-

4. LLM Orchestration & RAG Frameworks

Frameworks

- **LangChain**
- **LlamaIndex**
- Custom orchestration frameworks (explicitly mentioned)

Capabilities

- Retrieval
 - Routing
 - Tool use
 - Function calling
 - Multi-step agent workflows
-

5. Agentic AI Patterns

Agent Capabilities

- Planning & execution
 - Tool calling
 - Memory/state management
 - Workflow engines for agent execution
-

6. Retrieval-Augmented Generation (RAG)

RAG Pipeline Components

- Data ingestion
 - Chunking
 - Embeddings
 - Indexing
 - Latency-optimized retrieval
-

7. Vector Databases & Embeddings

Vector Databases

- **Pinecone**
- **Weaviate**
- **FAISS**
- **Qdrant**

Embedding Techniques

- Text embeddings for retrieval
-

8. Prompt Engineering & Evaluation

Prompt Engineering

- Prompt design
- Prompt optimization
- Prompt guardrails

Evaluation

- A/B testing
 - Quality metrics
 - Latency metrics
 - Cost metrics
 - Safety metrics
-

9. Observability, Monitoring & Reliability

Observability

- Traces
- Token usage tracking
- Failure monitoring

Tools

- **LangSmith** (or similar tools)

Reliability

- Cost controls
- Fallback mechanisms

10. Backend Services & APIs

Backend Development

- RESTful APIs
- Backend service development

Frameworks

- FastAPI
-

11. Cloud, DevOps & Deployment

Cloud Platforms

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure

DevOps & Tooling

- Docker
 - Git
 - CI/CD pipelines
-

12. Security, Safety & Guardrails

AI Safety

- Prompt safety
 - Guardrails for AI outputs
 - Safe deployment practices
-

13. Optional / Differentiating Skills

Additional Tooling

- **Model Context Protocol** (MCP – familiarity is a plus)
-

14. Collaboration & Delivery Context

Collaboration

- Product teams
- Leadership teams

Delivery Focus

- Translating business requirements into user-facing AI features
 - Rapid experimentation and iteration
 - Startup-style, high-ownership environment
-

15. Experience & Education Context (Technical)

Experience

- 1–3 years professional software development
- Hands-on LLM and RAG system development

Education

- B.Tech in Computer Science or related fields
-

16. Consolidated Technical Stack Summary

Languages

- Python

LLMs & GenAI

- OpenAI models
- Claude
- LLaMA
- Agentic workflows
- Function calling

- Tool use
- Memory-enabled agents

Frameworks

- LangChain
- LlamaIndex
- Custom orchestration

RAG

- Chunking
- Embeddings
- Indexing
- Latency-optimized retrieval

Vector Databases

- Pinecone
- Weaviate
- FAISS
- Qdrant

Backend & APIs

- FastAPI
- RESTful services

Observability

- LangSmith
- Tracing & token metrics
- Cost & failure monitoring

Cloud & DevOps

- AWS / GCP / Azure
- Docker
- Git
- CI/CD

Optional

- MCP (Model Context Protocol)

Tab 34

AI/ML Engineer (Agentic AI for HR Systems)

1. Core AI & System Domains

Primary Focus Areas

- Agentic AI
- Large Language Models (LLMs)
- Retrieval-Augmented Generation (RAG)
- Enterprise AI assistants
- HR self-service automation

Use Case Context

- Employee self-service AI agent embedded in HR applications
 - Queries related to payroll, policies, leave balances, and HR data
-

2. Agentic AI Architecture

Agent System Characteristics

- Tool-orchestrated agentic system
- Specialized tools:
 - Payslip Retrieval
 - Payroll Handler
 - Policy RAG
- Graph-based orchestration
- Structured outputs
- Extensible agent design

Agent Orchestration Framework

- **LangGraph**
-

3. LLMs & Generative AI

Large Language Models

- **Gemini**

LLM Capabilities

- Conversational AI
- Structured outputs
- Prompt design
- Function calling
- Streaming responses
- Safety mechanisms

SDK

- **Google Generative AI SDK**
-

4. LLM Frameworks & Data Models

Frameworks

- **LangChain**
- **LangGraph**

Data Modeling

- **Pydantic** for structured outputs
-

5. Retrieval-Augmented Generation (RAG)

RAG Components

- Text preprocessing
- Chunking
- Embeddings
- Vector search
- Metadata filtering

Vector Databases

- **Milvus**
 - **Zilliz**
 - Zilliz Cloud (management experience preferred)
-

6. Model Evaluation & Observability

Observability

- **LangSmith**

Evaluation Frameworks (Preferred)

- **RAGAS**
- **TruLens**

Evaluation Focus

- Quality tracking
 - Experiments & runs
 - Guardrails
 - Prompt injection awareness
-

7. Backend & API Development

Backend Framework

- **FastAPI**

API Capabilities

- REST APIs
 - JSON schemas
 - Pagination
 - Error handling
-

8. Frontend Integration

Frontend

- **React**

Frontend Concepts

- Hooks
 - Fetch / streaming handlers
 - Rendering tables
 - PDF links
-

9. Programming Language & Core Engineering

Language

- **Python** (Python 3.10+)

Engineering Fundamentals

- Type annotations
- Packaging
- Testing

Testing

- **pytest**
-

10. Data Ingestion & Processing

Document Ingestion

- **PyPDF**
- **Unstructured**

Data Types

- PDF documents
- DOC documents
- Tabular data

Data Engineering

- Basic ETL pipelines
- Policy corpus ingestion

11. Datastores & Caching

Databases

- MongoDB
- PostgreSQL

Caching & Sessions

- Redis
-

12. Search & Retrieval Optimization (Preferred)

Search Techniques

- HNSW indexes
 - IVF indexes
 - Hybrid search (BM25 + vector search)
 - Rerankers
-

13. DevOps, CI/CD & Deployment

Version Control

- Git
- GitHub / GitLab

Containerization

- Docker

DevOps Practices

- CI pipelines
 - Environment management
 - Secrets handling
-

14. Model Context Protocol (MCP)

Protocol

- Model Context Protocol

Concepts

- Servers
 - Tools / capabilities
 - Permissions
 - Extensible agent design
-

15. Internationalization & UX (Preferred)

i18n

- Multilingual UX
 - English
 - Arabic
 - RTL (Right-to-Left) awareness
-

16. Domain Knowledge (Preferred)

HR Domain

- Payroll cycles
 - Payslip structure
 - Leave & attendance
 - Reimbursements
 - HR policy management
 - Regional labor rules
-

17. Experience Context

Experience

- 2-3 years AI/ML or software engineering experience

- Hands-on agentic AI & RAG development
-

18. Consolidated Technical Stack Summary

Languages

- Python 3.10+

Agentic AI

- LangChain
- LangGraph
- Tool-based agents
- MCP (Model Context Protocol)

LLMs

- Gemini (Google Generative AI)

RAG & Vector Search

- Milvus
- Zilliz / Zilliz Cloud
- Embeddings
- Metadata filtering
- Hybrid search (BM25 + vectors)

Backend

- FastAPI
- REST APIs
- JSON schemas

Frontend

- React

Evaluation & Observability

- LangSmith
- RAGAS
- TruLens

Data & Storage

- PyPDF
- Unstructured
- MongoDB
- PostgreSQL
- Redis

DevOps

- Git
- GitHub / GitLab
- Docker
- CI/CD

Search Optimization

- HNSW
- IVF
- Rerankers

i18n

- English / Arabic
- RTL support

Tab 35

Generative AI Engineer

1. Core AI & Engineering Domains

Primary Focus Areas

- Generative AI
- Artificial Intelligence (AI)
- Machine Learning (ML)
- Deep Learning (DL)
- Natural Language Processing (NLP)

Application Scope

- Enterprise-grade AI/ML/DL/NLP platforms
 - Scalable, distributed ML systems
 - Business-critical generative AI solutions
-

2. Generative AI Models & Techniques

Generative Model Types

- Large Language Models (LLMs)
- Small Language Models (SLMs)
- Generative Adversarial Networks (GANs)
- Variational Autoencoders (VAEs)
- Autoregressive models
- Novel generative architectures

Reinforcement Learning

- Deep Reinforcement Learning (explicitly mentioned)
-

3. Programming Languages & Core Frameworks

Programming Languages

- **Python**

Deep Learning Frameworks

- **TensorFlow**
 - **PyTorch**
-

4. LLM & NLP Frameworks

Language Model Frameworks

- **LlamaIndex**
- **LangChain**
- **Haystack**

Use Cases

- Language model development
 - Chatbots
 - Q&A bots
-

5. Machine Learning & AI Specializations

ML / AI Areas

- Recommendation systems
 - Pattern recognition
 - Natural Language Processing (NLP)
 - Data mining
 - Text mining
 - Artificial intelligence systems
-

6. Data Engineering & Architecture

Data Capabilities

- Data preprocessing
- Dataset curation
- Data mining

- Text mining
 - Designing data architectures for AI workloads
-

7. API Development & AI Services

Service Development

- Deployment of generative AI models as scalable services
 - REST API-based AI services
 - Serverless AI functions
-

8. Cloud Platforms & Infrastructure

Cloud Platform

- **Microsoft Azure**

Cloud Services

- Platform as a Service (PaaS)
- Software as a Service (SaaS)
- Serverless functions

Compute Infrastructure

- Compute engines
 - Virtual Machines (VMs)
 - Containers
 - GPU servers
-

9. DevOps, CI/CD & Containerization

DevOps & Deployment

- CI/CD pipelines
- DevOps practices

Containerization & Orchestration

- Docker
 - Kubernetes
-

10. Software Engineering Practices

Engineering Practices

- Code reviews
- Code optimization
- Creation of reusable libraries and components
- Model-based development
- Distributed system architecture

Development Methodology

- Agile-based principles and tools
-

11. Mathematical & Analytical Foundations

Mathematics

- Statistics
 - Algebra
 - Probability
-

12. Additional / Differentiating Technical Knowledge

Knowledge Areas

- Ontology (explicitly mentioned as an asset)
 - Computer Vision (explicitly mentioned as a plus)
-

13. Collaboration, Leadership & Delivery (Technical Context)

Team & Leadership Responsibilities

- Managing a team of AI/ML engineers
 - Technical mentoring and guidance
 - Architecture ownership
 - Cross-functional collaboration with:
 - Data scientists
 - Software engineers
 - Product teams
 - Project managers
-

14. Experience Context (Technical)

Experience

- 6+ years in one or more of:
 - Machine learning
 - NLP
 - Recommendation systems
 - Pattern recognition
 - Data mining
 - Artificial intelligence
-

15. Education Context (Technical)

Degrees

- B.Tech / B.E.
- M.E. / M.Tech
- M.S.

Fields

- Computer Science
 - Information Technology
 - Electrical Engineering (EE / EEE)
-

16. Consolidated Technical Stack Summary

Languages

- Python

Deep Learning & GenAI

- LLMs
- SLMs
- GANs
- VAEs
- Autoregressive models
- Deep Reinforcement Learning

Frameworks

- TensorFlow
- PyTorch
- LangChain
- LlamaIndex
- Haystack

AI Applications

- Chatbots
- Q&A bots
- Recommendation systems
- NLP platforms

Data & Architecture

- Data mining
- Text mining
- Data architecture design

Cloud & Infra

- Microsoft Azure
- PaaS / SaaS
- Serverless functions
- VMs
- Containers
- GPU servers

DevOps

- CI/CD
- Docker

- Kubernetes

Methodologies

- Agile development

Additional

- Ontology knowledge
- Computer Vision exposure

Tab 36

AI Engineer (RAG & Agentic AI Specialist)

1. Core AI & Engineering Domains

Primary Focus Areas

- Generative AI (GenAI)
 - Retrieval-Augmented Generation (RAG)
 - Agentic AI
 - Large Language Models (LLMs)
 - Enterprise AI systems
 - AI-driven process automation
-

2. Agentic AI Frameworks & Platforms

Agentic / Orchestration Frameworks

- LangChain
- CrewAI
- AutoGen
- Copilot Studio
- Azure AI Foundry

Agent Capabilities

- Agent-based architectures
 - Reasoning-based automation
 - Agent workflow orchestration
 - Prompt-driven decision systems
-

3. LLMs & GenAI Ecosystem

LLMs & APIs

- **OpenAI**
- **GPT-4**
- **Hugging Face**

Model Types

- Large Language Models (LLMs)
 - Small Language Models (SLMs)
-

4. Retrieval-Augmented Generation (RAG)

RAG Components

- Prompt design
 - Retrieval pipelines
 - Vector databases
 - Context augmentation
 - Agent + retrieval integration
-

5. Protocols & Automation Technologies

AI Protocols & Automation

- **Model Context Protocol (MCP)**
 - **A2A**
 - **Microsoft Power Automate**
 - **Microsoft 365 Copilot**
-

6. Backend Development & APIs

Backend Frameworks

- **FastAPI**

API Technologies

- REST APIs
- gRPC APIs
- Microservices-based backend systems

7. Model Serving & Inference Optimization

Model Deployment Frameworks

- **TensorRT**
- **ONNX Runtime**
- **NVIDIA Triton Inference Server**
- **TorchServe**

Inference Characteristics

- Low-latency inference
 - High-throughput serving
 - Production-grade model deployment
-

8. Cloud Platforms & Data Ecosystem

Cloud Platforms

- **Microsoft Azure** (preferred)
- **Google Cloud Platform**

Data & Analytics Platforms

- **Databricks**
 - **Azure Synapse**
-

9. MLOps, DevOps & Observability

MLOps Practices

- Model versioning
- Model monitoring
- Automated deployments
- Production ML pipelines

CI/CD & DevOps

- CI/CD pipelines

- Microservices deployment
- Containerized environments

Monitoring & Observability

- **Prometheus**
 - **Grafana**
-

10. Containers, Security & Infrastructure

Containerization

- **Docker**

System Concerns

- Security
 - Authentication
 - Fault tolerance
 - Performance optimization
 - Scalability
-

11. Programming & Core Engineering

Programming Language

- **Python**

Engineering Concepts

- Backend systems design
 - Distributed systems
 - Scalable architectures
-

12. Databases & Storage

Databases

- SQL databases

- NoSQL databases

Storage

- Scalable data storage solutions
-

13. Software Engineering Practices

Practices

- Microservices architecture
 - Agile development methodologies
 - Version control with **Git**
 - Architecture reviews
 - Performance tuning
-

14. Collaboration & Delivery Context

Cross-Functional Collaboration

- Business teams
- Cloud infrastructure teams
- AI researchers
- Data scientists
- Product managers

Delivery Responsibilities

- End-to-end AI product ownership
 - Enterprise AI adoption & enablement
 - Mentoring junior engineers
 - Best practices in AI software engineering
-

15. Experience & Qualification Context (Technical)

Experience

- 2-8 years overall AI/ML experience
- 2-3 years hands-on RAG & Agentic AI experience

Education

- B.E./B.Tech
 - M.Sc./B.Sc.
 - Computer Science / Data Science or related fields
-

16. Consolidated Technical Stack Summary

Languages

- Python

Agentic AI

- LangChain
- CrewAI
- AutoGen
- Copilot Studio
- Azure AI Foundry

LLMs & APIs

- OpenAI APIs
- GPT-4
- Hugging Face

RAG

- Retrieval pipelines
- Prompt engineering
- Vector databases

Automation & Protocols

- MCP
- A2A
- Power Automate
- M365 Copilot

Backend & APIs

- FastAPI
- REST
- gRPC

- Microservices

Model Serving

- Triton Inference Server
- TensorRT
- ONNX Runtime
- TorchServe

Cloud & Data

- Azure
- GCP
- Databricks
- Azure Synapse

DevOps & Monitoring

- Docker
- CI/CD
- Prometheus
- Grafana

Engineering Practices

- Agile
- Git
- Security & performance optimization

Tab 37

Principal AI/ML Engineer (Healthcare AI)

1. Core AI & Engineering Domains

Primary Focus Areas

- Artificial Intelligence (AI)
- Machine Learning (ML)
- Natural Language Processing (NLP)
- Large Language Models (LLMs)
- Retrieval-Augmented Generation (RAG)
- Transformer-based models
- Healthcare AI & Clinical Decision Support

Healthcare Use Cases

- Patient-clinical trial matching
 - Clinical summarisation
 - Eligibility scoring
 - Real-time clinical data integration
 - Decision-support systems
-

2. Large Language Models & NLP

LLM & NLP Capabilities

- LLM-based reasoning
- Transformers
- Text embeddings
- Retrieval-Augmented Generation (RAG)
- Explainability benchmarking
- Speed & accuracy benchmarking

Model Types

- Proprietary LLMs

- Transformer-based NLP models
-

3. Programming Languages & ML Frameworks

Programming Language

- Python

Deep Learning Frameworks

- TensorFlow
 - PyTorch
-

4. Backend & API Development

Backend Framework

- FastAPI

Backend Responsibilities

- API-based AI services
 - Scalable ML-powered backend systems
 - End-to-end AI component ownership
-

5. Vector Databases & Retrieval Infrastructure

Vector Databases

- Milvus
- Weaviate

Retrieval Components

- Embeddings
 - Vector search
 - RAG pipelines
-

6. ML Infrastructure & Cloud Platforms

Cloud Platforms

- Microsoft Azure
- Amazon Web Services
- Google Cloud Platform

Infrastructure Capabilities

- Cloud ML pipelines
 - Model orchestration
 - Scalable AI infrastructure
 - Real-time data ingestion pipelines
-

7. Data Engineering & Pipelines

Data Capabilities

- Real-time data ingestion
 - Data pipelines
 - Clinical data processing
 - Embedding pipelines
 - Vector database integration
-

8. Model Experimentation & Evaluation

Experimentation

- LLM experimentation
- RAG pipeline benchmarking
- Transformer model evaluation

Evaluation Metrics

- Speed
 - Accuracy
 - Explainability
-

9. Responsible AI & Compliance

Responsible AI Practices

- Bias mitigation
- Model auditing
- Explainability
- Governance readiness

Regulatory Standards

- HIPAA
 - GDPR
-

10. Healthcare Data Standards & Ontologies (Nice-to-Have)

Clinical Standards

- FHIR

Clinical Ontologies

- SNOMED-CT
- ICD-10

Data Sources

- Electronic Health Records (EHR)
-

11. Open-Source & Advanced Model Work (Nice-to-Have)

Open-Source / OSS Models

- Mistral
- LLaMA
- GPT-NeoX

Advanced Techniques

- Fine-tuning LLMs
 - Distillation of LLMs
-

12. Research & Advanced AI Areas

Research Domains

- NLP
- Graph learning
- Multimodal AI

Activities

- Research evaluation
 - Integration of academic and industry breakthroughs
-

13. Leadership & Engineering Ownership (Technical Context)

Leadership Responsibilities

- Technical architecture ownership
 - Mentoring AI/ML engineers and researchers
 - Leading AI/ML Ops and data science functions
 - Driving end-to-end AI product delivery
-

14. Experience & Qualification Context (Technical)

Experience

- 6+ years hands-on AI/ML engineering
- Production-grade ML deployments
- Ownership of complex AI systems

Domain Exposure

- Healthcare
- Clinical trials

- Biomedical data (or willingness to learn domain constraints)
-

15. Consolidated Technical Stack Summary

Languages

- Python

AI & ML

- LLMs
- Transformers
- NLP
- RAG
- Embeddings
- Graph learning
- Multimodal AI

Frameworks

- TensorFlow
- PyTorch
- FastAPI

Vector Databases

- Milvus
- Weaviate

Cloud & Infra

- Azure
- AWS
- GCP
- Cloud ML pipelines
- Model orchestration

Healthcare Standards

- HIPAA
- GDPR
- FHIR
- SNOMED-CT
- ICD-10

- EHR systems

Open-Source Models

- Mistral
- LLaMA
- GPT-NeoX

Responsible AI

- Bias mitigation
- Model auditing
- Regulatory readiness

Tab 38

Machine Learning Engineer (AI / GIS / Defense)

1. Core AI & Engineering Domains

Primary Focus Areas

- Machine Learning (ML)
- Artificial Intelligence (AI)
- Data Analysis
- Algorithm Design
- Real-time ML systems

Application Domains

- AI & GIS technologies
 - Space applications
 - Military ISR (Intelligence, Surveillance, Reconnaissance)
 - Electronic Warfare systems
-

2. Machine Learning Capabilities

ML Engineering Responsibilities

- Design and deployment of ML models
- Training and inference pipelines
- Performance and scalability optimization
- Low-latency inference systems
- Model benchmarking and validation

ML Workflow

- Data preprocessing
- Feature engineering
- Model training
- Inference
- Testing and validation (lab and field environments)

3. Programming Languages

Languages

- Python
- R

4. ML Frameworks & Libraries

Machine Learning Frameworks

- TensorFlow
- PyTorch
- Scikit-learn

5. Data Visualization Tools

Visualization Libraries

- Matplotlib
- Seaborn
- Tableau

6. Data Engineering & Feature Development

Data Capabilities

- Data preprocessing
 - Feature engineering
 - Handling large datasets
 - Custom dataset creation
 - Dataset-driven model development
-

7. ML Pipelines & Systems Integration

Pipeline Engineering

- Scalable ML pipelines
- Efficient training pipelines
- Inference pipelines
- Integration into larger systems

System Integration

- Cross-functional integration
 - Deployment into production systems
-

8. Model Optimization & Performance

Optimization Focus

- Model performance
 - Scalability
 - Low-latency inference
 - Real-time ML system constraints
-

9. Version Control & Collaboration

Version Control

- Git

Git Practices

- Branching
 - Merging
 - Repository management
-

10. Cloud Platforms (Preferred)

Cloud Providers

- **Amazon Web Services**
- **Microsoft Azure**
- **Google Cloud Platform**

Use Cases

- ML model deployment
 - Scalable ML infrastructure
-

11. Advanced / Specialized ML Topics (Preferred)

Advanced ML Areas

- Federated learning
 - Edge AI
 - Real-time ML systems
 - Adversarial machine learning
 - AI-based countermeasures
-

12. Defense & Aerospace Context (Preferred)

Domain Exposure

- Defense projects
 - Aerospace systems
 - Military standards
 - Electronic warfare environments
-

13. Testing, Validation & Operations

Testing Environments

- Laboratory testing
- Field testing

Operational Tasks

- System validation
- Troubleshooting deployed ML systems

14. Research & Continuous Learning

Research Activities

- State-of-the-art ML algorithms
 - Emerging AI/ML techniques
 - Application of ML in electronic warfare
-

15. Experience & Education Context (Technical)

Experience

- 3+ years ML model development & deployment

Education

- Bachelor's degree in:
 - Computer Science
 - Data Science
 - Machine Learning
 - Related technical fields
-

16. Consolidated Technical Stack Summary

Languages

- Python
- R

ML Frameworks

- TensorFlow
- PyTorch
- Scikit-learn

Data Visualization

- Matplotlib
- Seaborn

- Tableau

ML Systems

- Training & inference pipelines
- Real-time ML systems
- Edge AI (preferred)
- Federated learning (preferred)

Cloud

- AWS
- Azure
- GCP

Version Control

- Git

Advanced Topics

- Adversarial ML
- AI countermeasures
- Defense & aerospace ML applications

Tab 39

Python Developer (AI / ML)

1. Core Engineering & AI Domains

Primary Focus Areas

- Artificial Intelligence (AI)
- Machine Learning (ML)
- Backend Development
- Frontend Development
- Full-Stack Development

Application Scope

- Intelligent systems
 - Scalable ML models
 - Real-time and batch AI services
 - Production-grade ML systems
-

2. Programming Languages

Languages

- Python
 - SQL
-

3. Python Libraries (Data & Visualization)

Core Python Libraries

- NumPy
 - Pandas
 - Matplotlib
 - Seaborn
-

4. Machine Learning Frameworks & Libraries

ML / DL Frameworks

- **scikit-learn**
 - **TensorFlow**
 - **PyTorch**
 - **Keras**
 - **XGBoost**
-

5. Data Engineering & Processing

Data Capabilities

- Data acquisition
- Data cleaning
- Data transformation
- Feature engineering
- ETL pipelines

Big Data

- **PySpark**

Databases

- SQL databases
 - NoSQL databases
-

6. Machine Learning Techniques

ML Algorithms

- Regression
 - Classification
 - Clustering
 - Natural Language Processing (NLP)
 - Deep Learning
-

7. API & Service Development

Backend APIs

- Real-time model serving
- Batch inference services

Frameworks

- **Flask**
 - **FastAPI**
-

8. Deployment, MLOps & Monitoring

Deployment Technologies

- **Docker**
- **Kubernetes**

Operational ML

- Model monitoring
- Model drift detection
- Retraining pipelines
- MLOps practices

CI/CD

- CI/CD pipelines
-

9. Cloud Platforms (Plus / Optional)

Cloud Providers

- **Amazon Web Services**
 - **Google Cloud Platform**
 - **Microsoft Azure**
-

10. Software Engineering Practices

Engineering Practices

- Code reviews
 - Agile development
 - Sprint planning
 - Collaboration with data scientists & product teams
 - Reusable and maintainable code design
-

11. Experience & Education Context (Technical)

Education

- Bachelor's or Master's degree in:
 - Computer Science
 - Engineering
 - Data Science
 - Related technical fields
-

12. Consolidated Technical Stack Summary

Languages

- Python
- SQL

ML & AI

- Machine Learning
- Deep Learning
- NLP
- Regression / Classification / Clustering

Frameworks & Libraries

- scikit-learn
- TensorFlow
- PyTorch
- Keras
- XGBoost
- NumPy
- Pandas

- Matplotlib
- Seaborn

Data Engineering

- PySpark
- ETL pipelines
- SQL / NoSQL databases

APIs & Backend

- Flask
- FastAPI

Deployment & MLOps

- Docker
- Kubernetes
- CI/CD
- Model monitoring & retraining

Cloud

- AWS
- GCP
- Azure