TECHNOLOGY CAREER PROGRAM

All-in-One Full Stack, AI & Cloud Program

A structured learning journey across seven technology tracks. Build applications. Connect intelligence. Ship with confidence.

PROJECT-LED LEARNING
LABS + CODE REVIEWS
FINAL DEMONSTRATIONS

START WITH THE CORE.



7 TRACKS. ONE CONNECTED FUTURE.

INR 49,000

24W →

6-MONTH TECHNOLOGY CAREER PROGRAM / ALL 7 TRACKS INCLUDED

YOUR NEXT 24 WEEKS.

BUILD / CONNECT / DEMONSTRATE

Build Java and Python full-stack applications with React. Develop foundations in GenAI, cloud, microservices, blockchain and data science.

Practice testing, Git workflows, documentation, deployment thinking and technical presentation.

THE CONNECTED CURRICULUM / WEEKS

EXPLORE ON PAGE

01–05	Java Full Stack Development	03 ↗
06–10	Python Full Stack Development with React	04 ↗
11–14	Generative AI & Agentic AI	05 ↗
15–18	DevOps & Cloud Engineering	06 ↗
19–20	Microservices & Cloud-Native Architecture	07 ↗
21–22	Blockchain & Web3 Essentials	08 ↗
23–24	Data Science & Machine Learning Essentials	09 ↗

ALL-IN-ONE / ALL 7 TRACKS / 24 WEEKS

INR 49,000

Save INR 31,000 · 38.75%

Compared with the proposed separate-track total.

7 COURSES SEPARATELY

INR 80,000

Proposed prices. Individual price allocation is shown in each separate course brochure; it is not specified here.

24 weeks · project-led learning · labs and code reviews · final demonstrations. Complete track projects, then connect APIs, data, deployment and AI in an integrated capstone.

JAVA FULL STACK

01

5 WEEKS

Build the application core: typed logic, persistent data, REST APIs and a connected React interface.

WEEK 01

Java Foundations & OOP

JDK/JRE/JVM setup; primitive types, conversion, operators, flow and loops; arrays, strings and methods. Classes, objects, constructors, encapsulation and access; inheritance vs composition, polymorphism and overriding; abstract classes, interfaces, packages, exceptions; collections, generics and exercises.

WEEK 02

Advanced Java, SQL & Persistence

Collections, bounded generics, lambdas and functional interfaces; Streams/collectors, Optional, date/time, I/O/NIO; threads, executors and concurrency. JDBC, prepared statements and transactions; modelling/normalization, joins, subqueries, indexes/query plans, connection pooling and DAO/repository patterns.

WEEK 03

Spring Boot & REST APIs

Maven/dependencies; IoC/DI, bean lifecycle/configuration, starters/auto-config and layered controllers/services/repositories. REST/HTTP, DTO mapping, validation and exceptions; JPA/entities, pagination/sorting, transactions/isolation; OpenAPI/Swagger, Postman and integration exercises.

WEEK 04

React Frontend & Full-Stack Integration

JavaScript/ES modules, Node.js/npm, React/JSX; components/props, state/events, useState/useEffect, conditions/lists; forms/validation, Router and reusable components. Fetch/Axios, loading/error handling, CORS and backend integration; responsive CSS and end-to-end CRUD.

WEEK 05

Testing, Security, Deployment & Capstone

JUnit5/Mockito; integration, API contract and negative tests; logging/environment config. Spring Security authentication and role-based authorization concepts; secure inputs/secrets; Git branches/PRs, CI, deployable JAR, Docker, cloud and health checks; demo, review and documentation.

24W →

BUILD / APPLIED PROJECT

Employee Management Portal

React UI + Spring Boot REST API + PostgreSQL. Include validation, search, role-aware screens, tests, API documentation and a deployable build.

PYTHON + REACT FULL STACK

02

5 WEEKS

Move from expressive Python to production-minded services and responsive customer experiences.

WEEK 06

Python Programming & Engineering Practices

Installation, venv/pip, syntax/types and strings; lists, tuples, sets, dictionaries; control flow/functions, modules/packages, exceptions, files/context managers. Comprehensions, generators, OOP, dataclasses/type hints, iterators/decorators; pytest, Git and code quality.

WEEK 07

SQL, Data Access & Backend Design

Relational design/normalization, filtering/grouping/joins, constraints/indexes and ACID; PostgreSQL tooling/connectivity, SQLAlchemy/ORM and CRUD repositories. Environment configuration and secrets outside source control; Pydantic, application layers, migrations, tests and fixtures.

WEEK 08

FastAPI, REST & Backend Services

Project structure; path/query parameters, request/response models, DI and CRUD endpoints; HTTP/errors, validation/exception handlers, OpenAPI. Async fundamentals, middleware/logs, CORS, pagination/sorting; authentication/authorization overview, pytest API tests, versioning and maintainability.

WEEK 09

React Frontend & Integration

ES6+/JSX; components, props, state, hooks/lifecycle, lists/keys, forms/validation, Router and reusable UI. Fetch/Axios; loading, error and empty states; environment URLs, CORS troubleshooting, responsive design and accessibility; React CRUD connected to FastAPI.

WEEK 10

Testing, Deployment & Capstone

Pytest unit/integration tests, mocking/coverage and logging; dev/production configuration and secrets; Git/review and CI. Production servers, Dockerfiles/containers and Python deployment; database connectivity, health/readiness, troubleshooting, presentation and handover.

24W →

BUILD / APPLIED PROJECT

Service Booking Platform

React client + FastAPI backend + PostgreSQL. Build customer and booking CRUD, validation, filters, API documentation, tests and deployable configuration.

GENERATIVE AI & AGENTIC AI

03

4 WEEKS

Ground intelligence in evidence. Design useful assistants with tools, evaluation and responsible boundaries.

WEEK 11

AI Foundations, LLMs & Prompt Engineering

AI, ML, deep learning and GenAI; capabilities/limits, LLM architecture/use cases, tokens/context/embeddings and high-level transformers. Hosted/open models and APIs; system/developer/user instructions; zero/few-shot prompts, role/task framing, JSON, templates/chaining, injection awareness and prompt evaluation.

WEEK 12

APIs, Embeddings & RAG

API calls, keys/environment, temperature/parameters and streaming; embeddings/similarity, ingestion/extraction/chunking, metadata/sources and vector databases. Hybrid search, top-k/filtering, RAG prompts, grounded answers with citations, retrieval failure analysis and document Q&A.

WEEK 13

Agents, Tools, MCP & Guardrails

Agents vs deterministic workflows; function calling/schemas, planning/execution, state/memory and orchestration/frameworks. Model Context Protocol, tools/data and approvals; input/output guardrails, injection defenses, PII protection, timeouts/retries/budgets and tracing/debugging.

WEEK 14

Evaluation, LLMOps & Production Capstone

Acceptance tests/golden datasets, groundedness/relevance and human review; latency/token costs, caching/rate limits, observability/traces, model/prompt versions and fallbacks. Secrets, API wrapping, containers/cloud, monitoring/feedback; responsible AI and limitations; demo/report.

24W →

BUILD / APPLIED PROJECT

Enterprise Knowledge Assistant

Ingest business documents, retrieve vector-index evidence and answer with citations. Expose an API, log traces and evaluate response quality.

DEVOPS & CLOUD ENGINEERING

04

4 WEEKS

Make delivery repeatable. Turn source code into observable, secure and recoverable running systems.

WEEK 15

Linux, Git & CI/CD Foundations

DevOps/delivery; Linux filesystem/shell, users/permissions/processes, networking/DNS and scripting. Commits/branches, merge/rebase, PRs/review, artifacts/dependencies; CI stages, builds/tests, variables/secrets, versions/environments, failure diagnosis and rollback.

WEEK 16

Docker & Kubernetes

Containers vs VMs; layers/registries, Dockerfiles/build context/.dockerignore, networking/volumes and Compose. Kubernetes architecture; Pods, Deployments, Services/discovery, ConfigMaps/Secrets, probes/resources and Ingress; rolling updates/rollback, pod and image-pull troubleshooting.

WEEK 17

Infrastructure as Code & Cloud

Cloud service models/shared responsibility, VPC/subnets and IAM; IaC with Terraform resources/providers, state, plan/apply, reusable modules/variables and environments. Managed databases/networking, registries/publishing, deployment pipelines/configuration, backups/recovery, cost/tagging and runbooks.

WEEK 18

Observability, Security & Delivery Capstone

Application/infrastructure metrics, Prometheus/Grafana, centralized logs, alert noise and tracing. Scanning, least-privilege IAM/secrets, supply-chain hygiene, network/TLS, GitOps, blue-green/canary delivery, incident response and rollback; demonstrate the pipeline.



BUILD / APPLIED PROJECT

CI/CD Delivery Platform

Build and test, containerize, publish an image and deploy to Kubernetes. Add health checks, dashboards and a rollback runbook.

MICROSERVICES & CLOUD-NATIVE

05

2 WEEKS

Architecture in motion: define service boundaries, coordinate events and design for failure.

WEEK 19

Service Design, APIs & Communication

Monolith, SOA and microservices trade-offs; bounded contexts/business decomposition, service and data ownership. REST/versioning, gateway/discovery and synchronous risks; messaging/event-driven design, Kafka/brokers, schemas/ idempotency and database per service. Distributed transactions/Saga; architecture diagrams and decision records (ADRs).

WEEK 20

Resilience, Security & Operations

Timeouts/retries/circuit breakers, bulkheads/rate limits, idempotent consumers, outbox/eventual consistency and centralized configuration. Service authentication with OAuth2/JWT, secrets/TLS; logs, metrics, distributed traces/correlation IDs; health/readiness, contract/integration tests, deployment/rollback, review and runbook.

FIELD NOTES / CONNECT THE CONCEPTS

NODE / 01

API gateway

→

NODE / 02

Order + inventory

→

NODE / 03

Events + traces

24W →

BUILD / APPLIED PROJECT

Order Fulfilment Workflow

API gateway + order/inventory services + asynchronous events. Include idempotency, Saga-style compensation, health checks and traceable logs.

BLOCKCHAIN & WEB3 ESSENTIALS

06

2 WEEKS

Follow the chain of trust: understand ledgers, programmable contracts and real-world constraints.

WEEK 21

Ledger Fundamentals, Cryptography & Ethereum

History, centralization/decentralization, ledgers, blocks, transactions and chains; hashes/integrity, public/private keys/signatures, Merkle trees and consensus. Bitcoin UTXO vs account models; Ethereum accounts/transactions, gas/fees, wallets and key safety, testnets/explorers; practical uses and limitations.

WEEK 22

Solidity, Smart Contracts & Enterprise Use Cases

Contract structure, types/state, functions/visibility, mappings/arrays, events/logs, modifiers/access control and require/revert/errors; Remix. ERC20, NFT/ERC721 concepts; reentrancy/vulnerabilities, OpenZeppelin, testing/review. Hyperledger Fabric peers/channels/chaincode overview; a traceability mini-project.

FIELD NOTES / CONNECT THE CONCEPTS

NODE / 01
Batch identity

→

NODE / 02
Ownership event

→

NODE / 03
Verifiable history

24W →

BUILD / APPLIED PROJECT

Product Traceability Ledger

Model batches and ownership events. Implement a basic smart-contract or permissioned-ledger workflow and document integrity and access-control decisions.

DATA SCIENCE & MACHINE LEARNING

07

2 WEEKS

From raw records to reasoned decisions. Prepare data, evaluate models and communicate what the evidence supports.

WEEK 23

Python Analytics, Statistics & Data Preparation

Data lifecycle/problem framing, notebooks, NumPy/vectorization, Pandas and CSV/tabular data. Types/missing values, duplicates/outliers, filtering/grouping/aggregation, joins/reshape; descriptive statistics, probability/distributions, sampling/bias, Matplotlib/EDA; features and leakage awareness.

WEEK 24

Machine Learning, Evaluation & Capstone

Supervised/unsupervised learning; train/validation/test splits, baselines, linear/logistic regression, trees/random forests and clustering. Scaling/encoding, cross-validation, classification/regression metrics, confusion matrix and precision/recall; overfitting/regularization, model comparison/reproducibility, predictions API concept, findings and limitations.

FIELD NOTES / CONNECT THE CONCEPTS

NODE / 01
Prepare + explore

NODE / 02
Train + evaluate

NODE / 03
Explain + report

24W →

BUILD / APPLIED PROJECT

Customer Churn Analysis

Clean data, explore patterns, build a baseline classifier and compare metrics. Explain churn drivers and present a reproducible notebook.

CONNECT IT ALL. BUILD SOMETHING REAL.

Smart Service Operations Platform with AI Assistance

Build a small full-stack service platform with React and a Java or Python REST backend. Store operational records in a relational database; add an AI knowledge assistant over curated documentation. Package and deploy through CI/CD, documenting service boundaries, data analysis and security decisions.

01 ——→
REACT Interface

02 ——→
REST + SQL Operations

03 ——→
RAG + AI Knowledge

04 ——→
CI/CD Deployment

Include source control, tests, API documentation, logs and health checks, deployment notes, a working demo and clearly stated limitations.

ASSESSMENT & DELIVERABLES

01 **Practice every week**
Weekly coding labs and short knowledge checks.

02 **Make the work inspectable**
Track project source code, README, architecture sketch and API examples; tests and sample data appropriate to each project.

03 **Demonstrate and explain**
Final demo, technical explanation and an improvement backlog.

YOUR NEXT CHAPTER STARTS WITH A CONVERSATION.

Ask the program team for confirmed batch dates, class schedule, delivery format, eligibility and enrollment details.

IMPORTANT NOTE / CONFIRM BEFORE PUBLICATION

This brochure describes a proposed curriculum and schedule. Actual class hours, batch dates, lab access, placement services, certification and taxes (if applicable) should be confirmed by the training provider before publication.

RKCP Information Technologies

One stop solution! for all your IT needs.

rkcpinformationtechnology.info

Contact info

135, 5th cross kv layout opposite LVS Excellency Apartment, Anandpura Circle tc palya, kr puram, Bangalore - 560036

enquiry@rkcpinfo.com

+91 8904046146
+91 7676403437

Quick Links

Employee Login

Terms of Service

newsletter

Be the first to find out about our latest news, updates, and trends.

Your Email Address →



DEVOPS COURSE OUTLINE (BEGINNER → ADVANCED)

Beginner Level (40 Hours)

Module 1: Introduction to DevOps (3 hrs)

- Evolution of Software Development (Waterfall → Agile → DevOps)
- DevOps principles and lifecycle
- Key tools overview (Git, Jenkins, Docker, Kubernetes, Ansible, Terraform, Monitoring)

Module 2: Linux Fundamentals for DevOps (5 hrs)

- Basic Linux commands (file system, permissions, process management)
- Shell scripting basics
- User & group management
- Package management (apt, yum, dnf)

Module 3: Version Control with Git & GitHub (6 hrs)

- Git architecture, Git workflow (clone, commit, push, pull, merge)
- Branching strategies (Git Flow, trunk-based development)
- GitHub features: Issues, Pull Requests, Actions basics
- Secret scanning with **Gitleaks**

Module 4: Build Automation with Maven/Gradle (4 hrs)

- Build lifecycle & dependency management
- Creating and packaging applications
- Integration with CI/CD

Module 5: Containerization with Docker (7 hrs)

- Docker architecture (images, containers, volumes, networks)
- Dockerfile creation & optimization
- Pushing/pulling from Docker Hub
- Secure image best practices

Module 6: Continuous Integration with Jenkins (5 hrs)

- Jenkins installation & pipeline basics
- Jenkins master-agent setup
- Job configuration & parameterized builds
- Integration with Git

Module 7: Automated Testing with Selenium & JUnit (5 hrs)

- Introduction to automated testing
- Writing test cases with Selenium
- Maven integration with testing
- Headless browser testing

🟡 Intermediate Level (45 Hours)

Module 8: Container Orchestration with Kubernetes (8 hrs)

- Kubernetes architecture (nodes, pods, deployments, services)
- ConfigMaps, Secrets, StatefulSets, DaemonSets
- Autoscaling (HPA, VPA)
- Ingress & LoadBalancer setup

Module 9: Configuration Management with Ansible & Puppet (6 hrs)

- Ansible: Playbooks, roles, dynamic inventory
- Puppet: Manifests, modules, agents
- Configuring cloud servers (AWS EC2 examples)

Module 10: Infrastructure as Code with Terraform (6 hrs)

- Terraform basics (providers, resources, state, variables, outputs)
- AWS provisioning: VPC, EC2, S3

- Modules, workspaces & security best practices

Module 11: Continuous Delivery with GitLab CI/CD (5 hrs)

- GitLab runners (shared vs self-hosted)
- Stages, jobs, caching, artifacts
- Docker integration in GitLab pipelines
- Deployments to AWS (ECR, ECS, EC2)

Module 12: Continuous Monitoring (6 hrs)

- Prometheus: setup, metrics, PromQL queries
- Grafana: dashboard creation, alerts
- ELK Stack (Elasticsearch, Logstash, Kibana) basics

Module 13: Generative AI in DevOps (2 hrs)

- GitHub Copilot for code suggestions
- Automating scripts and error handling
- AI-assisted documentation

Module 14: Security Basics in DevOps (4 hrs)

- DevSecOps introduction
- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Dependency scanning

🔴 Advanced Level (55 Hours)

Module 15: Advanced Kubernetes & Security (8 hrs)

- RBAC & Network Policies
- Pod Security Standards
- Image scanning & runtime protection
- Kubernetes auditing & monitoring

Module 16: AWS EKS & Cloud Native Workloads (6 hrs)

- Provisioning AWS EKS with Terraform
- Cluster autoscaler setup
- Add-ons (ALB Ingress Controller, ExternalDNS)

- Compliance checks

Module 17: Service Mesh with Istio (6 hrs)

- Istio architecture
- mTLS, ingress & egress
- Traffic management (blue-green, canary)
- Observability with Istio dashboards

Module 18: Policy-as-Code with OPA Gatekeeper (4 hrs)

- OPA fundamentals
- Constraint templates & policies
- Integrating OPA into CI/CD pipelines

Module 19: Secrets Management (4 hrs)

- Vault basics (KV secrets, policies, tokens)
- AWS Secrets Manager
- Kubernetes External Secrets Operator

Module 20: GitOps with ArgoCD & Kustomize (6 hrs)

- GitOps fundamentals
- Declarative deployments with ArgoCD
- Application rollback & sync waves
- Using Kustomize for environment overlays

Module 21: DevSecOps & Compliance (6 hrs)

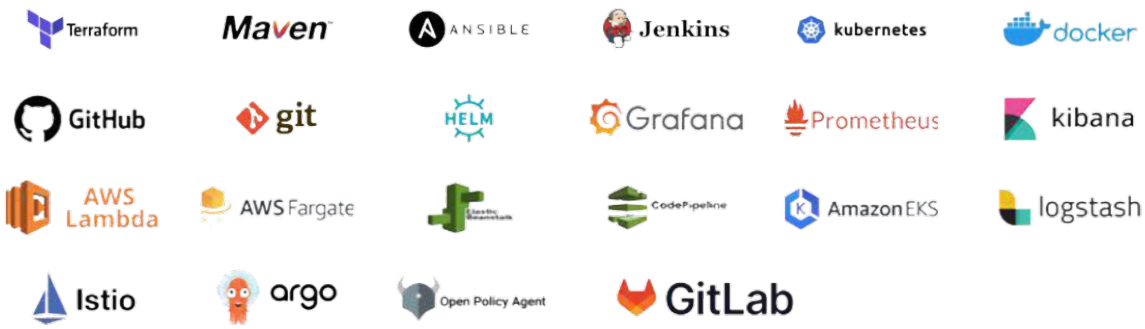
- End-to-end security integration in pipelines
- DefectDojo for vulnerability management
- Compliance frameworks (CIS, NIST, PCI-DSS)

🧪 Real-World Projects:-

1. **Banking:** Secure CI/CD pipeline with Jenkins & Vault integration
2. **Healthcare:** Deploy EHR application on Kubernetes with monitoring
3. **E-Commerce:** Scalable microservices architecture with Docker + Kubernetes
4. **Streaming App:** Deploy Netflix-like platform with autoscaling & CDN integration
5. **Retail:** Infrastructure automation with Terraform on AWS (multi-tier setup)

6. **Insurance:** Real-time monitoring dashboards with Prometheus & Grafana
7. **AI-Powered DevOps:** GitHub Copilot-assisted CI/CD pipeline
8. **Media:** Logging & centralized monitoring with ELK stack
9. **Supply Chain:** Secure Kubernetes workloads with Istio & OPA policies
10. **Capstone Mega Project:** Full-fledged DevOps pipeline (Git → Jenkins/GitLab → Docker → Kubernetes → ArgoCD → Monitoring → Security Scanning)

Tools to Master





Beginner Level – Blockchain Foundations (40 Hours)

LANGUAGES TO USE

Python – blockchain basics & prototyping • JavaScript – frontend + Web3.js integration • Solidity (Intro) – simple smart contracts

Module 1: Blockchain Fundamentals (6 hrs)

- What is Blockchain? History & evolution (Bitcoin → Ethereum → Web3)
- Centralized vs Decentralized
- Distributed Ledger Technology (DLT) basics
- Consensus mechanisms overview

Module 2: Cryptography & Security Basics (6 hrs)

- Public/Private Keys
- Hashing (SHA-256, Keccak)
- Merkle Trees
- Digital Signatures

Module 3: Bitcoin & Ethereum Basics (6 hrs)

- Bitcoin UTXO model
- Ethereum Accounts, Gas & Transactions
- Testnets (Sepolia, Holesky)
- Using Etherscan & Blockchain explorers

Module 4: Smart Contract Introduction (8 hrs)

- Solidity basics: variables, functions, modifiers
- Remix IDE for contract deployment
- Events & logs
- Simple contract deployment

Module 5: Tools & Wallets (6 hrs)

- MetaMask setup & transactions
- Ganache for local blockchain
- Web3.js / Ethers.js introduction
- DApp structure

BEGINNER PROJECTS (4)

1. Create a mini blockchain ledger in Python.
2. Deploy a “Hello Blockchain” smart contract (Solidity + Remix).
3. Create a crypto wallet & send ETH on a testnet (MetaMask).
4. Build a basic DApp that stores and retrieves user data (JavaScript + Web3.js).

Intermediate Level – Smart Contracts & DApps (50 Hours)

LANGUAGES TO USE

Solidity – advanced smart contracts • TypeScript/JavaScript – DApp integration & testing • Go (Golang) – Hyperledger basics

Module 6: Advanced Smart Contracts (10 hrs)

- Solidity: Inheritance, Libraries, Structs
- Security patterns (Ownable, Pausable)
- Hardhat/Truffle for testing & deployment
- Upgradeable contracts

Module 7: Token Standards (8 hrs)

- ERC-20 Fungible Tokens
- ERC-721 NFTs
- ERC-1155 Multi-tokens
- Gas optimization in tokens

Module 8: DeFi Basics (10 hrs)

- What is DeFi? Lending, borrowing, yield farming
- Automated Market Makers (Uniswap model)
- Liquidity Pools & Staking
- Governance tokens

Module 9: Oracles & Off-Chain Data (7 hrs)

- What are Oracles?
- Chainlink basics
- Connecting contracts to external APIs
- Real-world use cases

Module 10: Scaling Solutions (7 hrs)

- Layer 2: Polygon, Arbitrum, Optimism
- Rollups, Plasma, State Channels
- Bridges between chains

INTERMEDIATE PROJECTS (4)

5. Create & deploy your own ERC-20 token (Solidity).
6. Build an NFT collection with metadata (Solidity + IPFS).
7. Develop a DeFi staking smart contract (Solidity + JS).
8. DApp with live ETH price feed using Chainlink (Solidity + JS).

Advanced Level – Enterprise & Cross-Chain (60 Hours)

LANGUAGES TO USE

Solidity – production-grade contracts • Rust – Solana, Polkadot, Cosmos SDK • Go – Hyperledger Fabric chaincode • Java – Corda enterprise blockchain

Module 11: Blockchain Architecture & Consensus (10 hrs)

- Proof of Work vs Proof of Stake vs Delegated Proof of Stake
- Byzantine Fault Tolerance (BFT)
- Sharding & scalability models
- Blockchain governance

Module 12: Enterprise Blockchain (10 hrs)

- Hyperledger Fabric (Peers, Orderers, Channels)
- Writing chaincode in Go
- R3 Corda basics (Java/Kotlin)
- Identity & permissioned blockchains

Module 13: Interoperability & Cross-Chain (8 hrs)

- Blockchain bridges
- Polkadot & Substrate (Rust)
- Cosmos SDK & IBC protocol
- Cross-chain asset swaps

Module 14: Security & Auditing (8 hrs)

- Vulnerabilities (Reentrancy, Overflow, DoS)
- Audit tools: MythX, Slither
- Using OpenZeppelin contracts
- Best practices & bug bounties

Module 15: Real-World Applications & Future Trends (8 hrs)

- Blockchain in Supply Chain, Healthcare, Voting
- Stablecoins & CBDCs
- Decentralized Identity (DID)
- Zero-Knowledge Proofs (ZK-SNARKs, ZK-Rollups)

ADVANCED PROJECTS (4)

9. Build a Decentralized Exchange (DEX) (Solidity + JS frontend).
10. Create a DAO with voting system (Solidity + TypeScript).
11. Develop a Hyperledger Fabric network for supply chain (Go).
12. Build a cross-chain asset bridge prototype (Rust + Solidity).

Capstone Projects (20+ Hours)

REAL-WORLD CAPSTONE PROJECTS

1. NFT Marketplace (ERC-721 contracts, Solidity + JS frontend)
2. DeFi Lending Platform (Collateralized lending, Solidity + TypeScript)
3. DAO Governance System (On-chain voting + treasury, Solidity + React)
4. Healthcare Blockchain System (Patient records on Hyperledger, Go)
5. Cross-chain Wallet (Ethereum + Polygon + Cosmos using Rust + JS)

BLOCKCHAIN ECOSYSTEM



Best Blockchain Languages (Summary)

- **Solidity** → Smart contracts (Ethereum, Polygon, BSC) – Must-learn
- **Rust** → Solana, Polkadot, Cosmos – High-performance chains
- **Go (Golang)** → Hyperledger, Cosmos SDK – Enterprise apps
- **Python** → Prototyping & learning blockchain basics
- **JavaScript/TypeScript** → DApp frontends, Web3.js, Hardhat
- **Java/Kotlin** → R3 Corda enterprise blockchain



PROGRAM CURRICULUM

Advanced Certification in Data Science and AI

Syllabus overview

CORE CURRICULUM — Modules 01–09

MODULE 01 Preparatory Sessions – Python & Linux

- Python: Introduction to Python and IDEs
- Python Basics
- Object Oriented Programming
- Hands-on Sessions And Assignments for Practice
- Linux: Introduction to Linux
- Linux Basics
- Hands-on Sessions And Assignments for Practice

MODULE 02 Data Wrangling with SQL

- SQL Basics
- Advanced SQL
- Deep Dive into User Defined Functions
- SQL Optimization and Performance

MODULE 03 Python With Data Science

- Data Handling with NumPy
- Data Manipulation Using Pandas
- Data Preprocessing
- Data Visualization

MODULE 04 Linear Algebra and Advanced Statistics

- Descriptive Statistics
- Probability
- Inferential Statistics
- Linear Algebra

MODULE 05 Machine Learning

- Introduction to Machine learning
- Regression
- Classification
- Clustering

MODULE 06 Supervised Learning

- Linear Regression
- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine
- K-Nearest Neighbors
- Time Series Forecasting
- Performance Metrics
- Classification reports
- Confusion Matrix
- Evaluation Matrix

MODULE 07 Unsupervised Learning

- K-means
- Dimensionality reduction
- Linear Discriminant Analysis
- Principal Component Analysis

MODULE 08 Deep Learning Using TensorFlow

- Artificial Intelligence Basics
- Neural Networks
- Deep Learning

MODULE 09 **Data Science Capstone Project** • Extracting, loading and transforming data into usable format to gather insights • Data manipulation and handling to pre-process the data • Feature engineering and scaling the data for various problem statements • Model selection and model building on various classification, regression problems using supervised/unsupervised machine learning algorithms • Assessment and monitoring of the model created using the machine learning models	

Program Curriculum

ADVANCED TOPICS & ELECTIVES — Modules 10–15

MODULE 10 Business Case Studies

- Recommendation Engine
- Rating Predictions
- Census
- Housing
- Object Detection
- Stock Market Analysis
- Banking Problem
- AI Chatbot

MODULE 11 Generative AI

- LSTM
- Transformers
- BERT
- GPT
- LLM

MODULE 12 Power BI · Elective

- Power BI Basics
- Data Visualization with Analytics

MODULE 13 Deploying Machine Learning Models With Cloud · Elective

- Introduction to MLOps
- Deploying Machine Learning Models

MODULE 14 GIT · Elective

- Version Control
- GIT

MODULE 15 Data Analysis With MS-Excel · Self-paced course

- Excel Fundamentals
- Excel For Data Analytics
- Data Visualization with Excel
- Excel Power Tools
- Classification Problems using Excel
- Information Measure in Excel
- Regression Problems Using Excel
- Hands-on Exercise

JOB READINESS — Career preparation

- Job Search Strategy
- Resume Building
- LinkedIn Profile Creation

SKILLS TO MASTER — Program outcomes

Python	**Data Science**	**Data Analysis**
Artificial Intelligence	**GIT**	**MLOps**
Data Wrangling	**SQL**	**Story Telling**
Machine Learning	**Prediction Algorithms**	**NLP**

PySpark	Model	Data Visualization

TOOLS TO MASTER

Technology toolkit

Matplotlib	Python	Jupyter	SciPy
NumPy	Pandas	SQL	Excel
Power BI	Git	Spark SQL	PySpark



GENERATIVE AI

COURSE CURRICULUM | BEGINNER TO ADVANCED

LEARN • BUILD • INNOVATE

- Generative AI (GenAI) Course Curriculum

🟢 Beginner Level (35–40 Hours)

Objective: Build solid foundations in AI, understand GenAI basics, and learn prompt engineering & hands-on APIs.

◆ Module 1: Foundations of AI & GenAI (3 hrs)

- AI vs ML vs DL vs GenAI
- History & breakthroughs in Generative AI
- Key models: GPT, Stable Diffusion, MidJourney
- Real-world applications & industry use cases

◆ Module 2: Introduction to LLMs (5 hrs)

- How LLMs work: tokens, embeddings, transformers
- Pretraining vs fine-tuning
- Popular open-source LLMs (LLaMA, Mistral, Falcon)
- Hosted APIs (OpenAI, Anthropic, Cohere)

◆ Module 3: Prompt Engineering (6 hrs)

- Zero-shot, few-shot prompting
- Instruction-tuning
- Role-based prompts & system instructions
- Structured prompting with templates
- Prompt chaining

◆ Module 4: GenAI Tools & APIs (6 hrs)

- OpenAI API basics (chat completions, embeddings)
- Hugging Face Spaces & Models
- Google Gemini, Anthropic Claude
- VS Code + GitHub Copilot

◆ Module 5: Hands-on with Text, Image, Audio (8 hrs)

- Text: Q&A, summarization, rewriting
- Images: DALL·E, Stable Diffusion, MidJourney basics
- Audio: TTS (text-to-speech), STT (speech-to-text)
- Safety & ethics in GenAI

✓ Beginner Projects

- Build a personal AI note summarizer.
- AI-powered email writing assistant.
- Generate an infographic/poster with Stable Diffusion.
- Create a voice-based chatbot with STT + TTS.

🟡 Intermediate Level (45–50 Hours)

Objective: Learn fine-tuning, RAG, multimodal models, and integrating GenAI into apps.

◆ Module 6: Model Fine-Tuning (8 hrs)

- Fine-tuning vs adapter methods (LoRA, PEFT)
- Hugging Face Trainer basics
- Training domain-specific chatbots
- Evaluating model performance

◆ Module 7: Retrieval-Augmented Generation (RAG) (7 hrs)

- Why RAG is needed
- Embeddings for semantic search
- Vector databases: Pinecone, FAISS, Weaviate
- Implementing a RAG-based chatbot

◆ Module 8: Multi-Modal Generative AI (6 hrs)

- Text-to-image (Stable Diffusion, DALL·E)
- Text-to-audio (ElevenLabs, Bark)
- Text-to-video (Pika Labs, RunwayML)
- Vision+Language (CLIP, BLIP-2)

◆ Module 9: GenAI in Software Development (6 hrs)

- Copilot for code generation
- AI-powered debugging
- Test case generation
- AI-assisted documentation

◆ Module 10: Application Development with LangChain (8 hrs)

- LangChain basics (chains, memory, tools, agents)
- Connecting APIs & databases
- Advanced prompting with LangChain
- Deploying apps with Streamlit/Gradio

✓ Intermediate Projects

5. Fine-tune GPT on company FAQs.
6. Build a RAG-based legal/educational Q&A chatbot.
7. AI-powered SQL query generator.
8. Build a multimodal app: text → image + audio + explanation.
9. Code assistant that generates unit tests with Copilot + LangChain.

◆ Module 11: Advanced Transformer Architectures (8 hrs)

- Deep dive into attention mechanisms
- Scaling laws, efficiency (quantization, pruning, distillation)
- LoRA & PEFT methods

◆ Module 12: Scaling & Deployment (7 hrs)

- Serving LLMs with Docker + FastAPI

- Cloud deployment (AWS, GCP, Azure)
- Cost optimization & monitoring

◆ Module 13: AI Safety & Responsible AI (6 hrs)

- Bias in LLMs
- Prompt injection & jailbreaks
- Guardrails (Azure Safety, Llama Guard)
- Responsible AI policies

◆ Module 14: Autonomous AI & Agentic Systems (8 hrs)

- What is Agentic AI (self-directed AI agents)
- LangChain Agents (tool-use, reasoning, planning)
- AutoGPT, BabyAGI, CrewAI
- Multi-agent collaboration (team of agents with roles)
- Use cases: research assistants, workflow automation, orchestration

◆ Module 15: Anthropic Model Context Protocol (MCP) (8 hrs)

- Introduction to MCP and context-sharing
- How MCP allows AI models/tools to communicate
- Structured context passing across agents
- MCP + Agentic AI integration
- Enterprise-grade use cases: knowledge workers, co-pilots, secure tool integration

◆ Module 16: Enterprise GenAI Applications (8 hrs)

- Healthcare: diagnostic copilots
- Finance: fraud/risk copilots
- Retail: AI-powered assistants
- Media: generative pipelines (text → video → marketing)

✦ Advanced + Agentic Projects

10. Deploy a fine-tuned model with Docker + FastAPI on AWS.
11. Build a multi-agent research assistant (Planner + Researcher + Writer).
12. AI-powered workflow automation system with LangChain Agents.
13. RAG + Guardrails chatbot with secure context injection.
14. Real-time voice-to-voice translation app with multimodal input.
15. AI-powered contract compliance checker (legal domain).
16. Anthropic MCP-based Knowledge Assistant (agents sharing context between finance & HR data).
17. Agentic AI + MCP Integration: Build an enterprise helpdesk with multi-agent orchestration (e.g., one agent handles FAQs, one checks HR policies, one generates email replies).

🧪 Capstone Projects (Choose 2–3)

- Healthcare: AI-powered EHR copilot with Agentic AI + MCP.
- Finance: Risk analysis system with MCP-based secure data exchange.
- Education: Autonomous tutor agent with memory & context-sharing.
- Entertainment: Text-to-video content generator using Agentic AI pipelines.
- Retail: Virtual AI shopping assistant that uses multi-agent + MCP for recommendations + order processing.

✦ BUILD THE FUTURE WITH GENERATIVE AI ✦



BEGINNER LEVEL | 40–45 HOURS

OBJECTIVE Learn microservices fundamentals, REST APIs, and containerization with Java (Spring Boot) and Python (Flask/FastAPI).

Module 1: Microservices Fundamentals (4 hrs)

- Monolith vs SOA vs Microservices
- Microservices advantages & challenges
- 12-Factor app principles
- Industry use cases (Netflix, Amazon, Uber)

Module 2: Java Microservices with Spring Boot (8 hrs)

- Spring Boot project setup (Maven/Gradle)
- REST controllers, DTOs, exception handling
- Data access with Spring Data JPA
- Testing with JUnit + Postman
- Swagger/OpenAPI docs

Module 3: Python Microservices with Flask & FastAPI (8 hrs)

- Flask basics: routes, request handling
- FastAPI async APIs with Pydantic models
- CRUD APIs with SQLAlchemy
- Dependency injection in FastAPI
- Auto-generated API docs (Swagger, Redoc)

Module 4: Database Integration (6 hrs)

- SQL databases: PostgreSQL/MySQL
- NoSQL databases: MongoDB basics
- ORM: Hibernate (Java), SQLAlchemy (Python)
- Database migrations (Flyway, Alembic)

Module 5: Containerization with Docker (7 hrs)

- Docker concepts: images, containers, volumes
- Dockerizing Java Spring Boot service
- Dockerizing Python FastAPI service
- Networking in Docker
- Multi-service setup with Docker Compose

BEGINNER PROJECTS (5)

1. Java Spring Boot CRUD service for Students.
2. Python FastAPI CRUD service for Inventory.
3. Dockerize both services & connect with Compose.
4. Build an API with Postgres backend + REST interface.
5. Create API documentation with Swagger/OpenAPI.



INTERMEDIATE LEVEL | 50 HOURS

OBJECTIVE Learn inter-service communication, messaging, discovery, monitoring, and API security.

Module 6: Inter-Service Communication (7 hrs)

- REST communication
- gRPC for microservices
- API Gateway: Kong, Nginx, Spring Cloud Gateway

Module 7: Messaging & Event-Driven Architecture (7 hrs)

- Synchronous vs asynchronous communication
- RabbitMQ producers/consumers in Java + Python
- Kafka basics: topics, partitions, producers, consumers
- Event sourcing vs CQRS

Module 8: Service Discovery & Configuration (7 hrs)

- Eureka service registry (Spring Cloud)
- Consul for Python microservices
- Centralized config with Spring Cloud Config Server
- Environment variables, secrets management

Module 9: Security in Microservices (7 hrs)

- Authentication vs Authorization
- JWT authentication flow
- Spring Security with OAuth2
- FastAPI JWT + OAuth2 integration
- Role-based access control (RBAC)

Module 10: Observability (7 hrs)

- Centralized logging with ELK stack
- Distributed tracing with Zipkin/Jaeger
- Metrics collection with Prometheus
- Visualization with Grafana

INTERMEDIATE PROJECTS (5)

6. Order + Payment microservices with API Gateway.
7. Event-driven email notification system (RabbitMQ/Kafka).
8. Secure microservices with JWT + OAuth2.
9. Centralized logging & monitoring dashboard (Grafana).
10. Service registry + config server for multi-service setup.



ADVANCED LEVEL | 60 HOURS

OBJECTIVE Master design patterns, CI/CD, Kubernetes, service mesh, and production best practices.

Module 11: Microservices Design Patterns (8 hrs)

- Saga pattern for distributed transactions
- Circuit breaker (Resilience4j, Hystrix)
- Strangler Fig & API Composition
- Bulkhead & Retry patterns

Module 12: CI/CD Pipelines for Microservices (8 hrs)

- CI/CD with Jenkins pipelines
- GitLab CI/CD for multi-service apps
- Docker build & push to registry
- Automated integration tests

Module 13: Cloud-Native Microservices (10 hrs)

- Kubernetes fundamentals (pods, services, deployments)
- Ingress controller setup
- ConfigMaps & Secrets
- Horizontal Pod Autoscaling (HPA)
- Deploying microservices on EKS/GKE/AKS

Module 14: Service Mesh & Advanced Security (8 hrs)

- Istio basics: sidecar proxy, ingress/egress
- Mutual TLS between microservices
- Policy enforcement with OPA Gatekeeper
- Zero-trust networking in microservices

Module 15: Resiliency & Production Best Practices (8 hrs)

- Canary & blue-green deployments
- API versioning strategies
- Observability with OpenTelemetry
- Handling failures gracefully
- Cost optimization & scaling strategies

ADVANCED PROJECTS (5)

11. E-commerce platform with Java + Python microservices (Products, Orders, Payments, Users).
12. Deploy complete microservices architecture on Kubernetes.
13. Implement Saga pattern with Java + Kafka for Order → Payment workflow.
14. Secure microservices with Istio mTLS + OPA policies.
15. CI/CD pipeline for microservices with Jenkins/GitLab.



CAPSTONE PROJECTS

Choose 2–3 projects | Approximately 20–30 hours

Banking	Multi-service system (Accounts, Transactions, Fraud Detection).
Healthcare	Patient record & appointment booking with event-driven architecture.
Retail	Inventory + order management with Kafka + real-time updates.
Education	Online course platform with Python recommendation engine + Java backend.
Streaming/Media	Netflix-style video platform with microservices + monitoring.

TOTAL PROJECTS | 15+

OBJECTIVE Beginner: 5 • Intermediate: 5 • Advanced: 5 • Capstone projects: 2–3

TOOLS & TECHNOLOGIES

- Java (Spring Boot)
- Python (Flask/FastAPI)
- Docker
- Kubernetes
- PostgreSQL / MySQL
- MongoDB
- RabbitMQ
- Kafka
- Jenkins
- GitLab CI/CD
- Prometheus
- Grafana
- ELK Stack
- Zipkin / Jaeger
- Eureka
- Consul
- Spring Cloud
- OAuth2 / JWT
- Istio
- OpenTelemetry
- OPA Gatekeeper
- AWS EKS / GKE / AKS