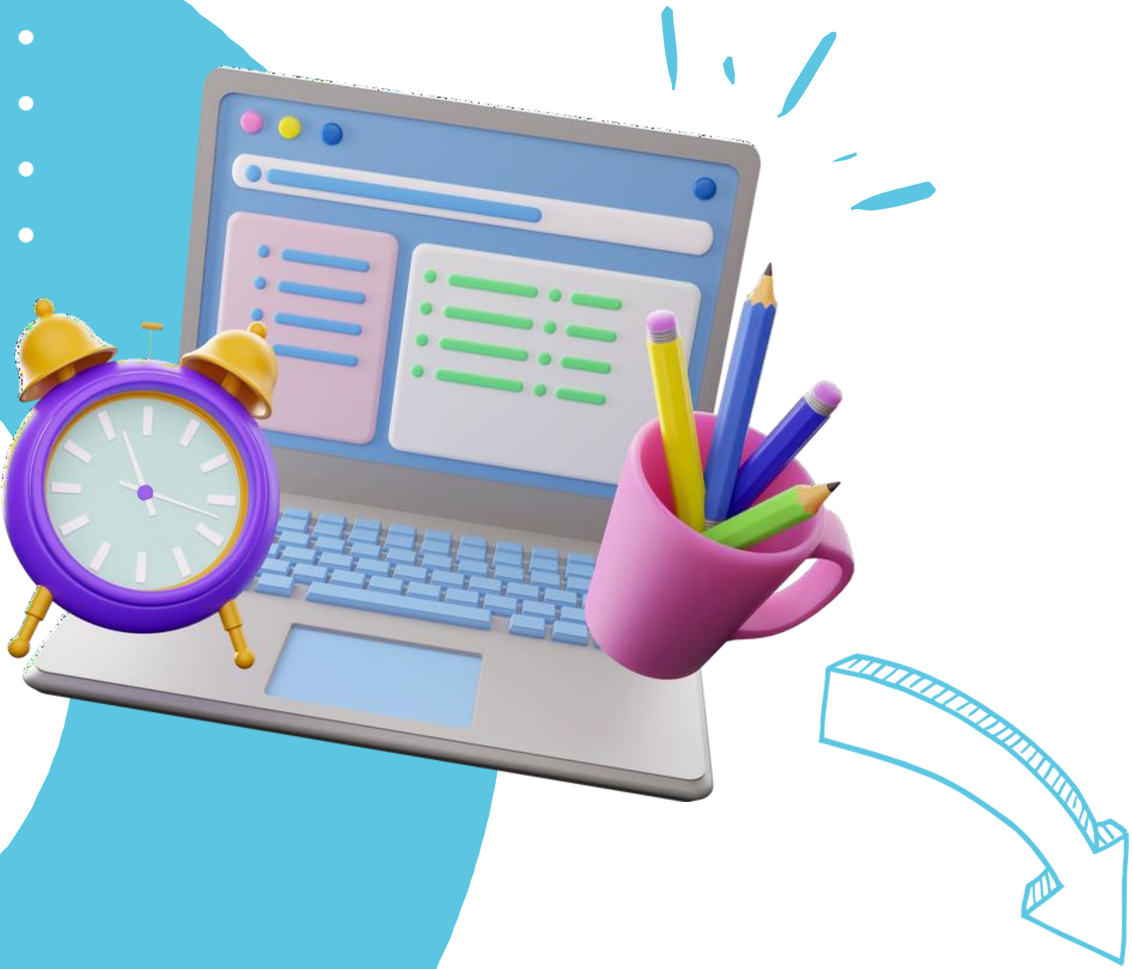
IT AS CODE
Information Technology as Code

TRAIN & HIRE!



>>Information Technology as Code <<

5000+ TRAINED PROFESSIONALS
75+ Batches Delivered

End to End - DevSecOps, Kubernetes & Cloud

IT AS CODE

Information Technology as Code

Upskill Your IT Workforce

Employers always consider ways to attract, outfit, and train their workforce

to meet the demands of this new skills revolution in light of the reality of a global talent shortage.

ITasCode is an end-to-end DevSecOps, Kubernetes cloud, Java Full Stack, Data Analytics, Data Science providing numerous comprehensive trainings.

We hear when there is a need for teams within your organisation with ability to perform to extreme efficiency. Fulfilling IT Training requirement is part of our code, prepared to meet the needs of your organisation.

Academy.ITasCode.in | +91 970 5535 444

IT AS CODE

Information Technology as Code

The Sprint

Recruitment Division

Our recruitment services focus on sourcing and deploying top IT talent for organizations.   We operate under a unique "learn and hire" model, where companies can train their employees and potentially recruit from a pool of pre-trained talent

Specialized IT Roles: We identify and recruit skilled professionals in DevOps, cloud architecture, Kubernetes administration, Software Development, Data Analytics and more.

Streamlined Hiring Process: We handle recruitment needs from screening to onboarding, ensuring we integrate the right talent into the team seamlessly.

Information Technology as Code

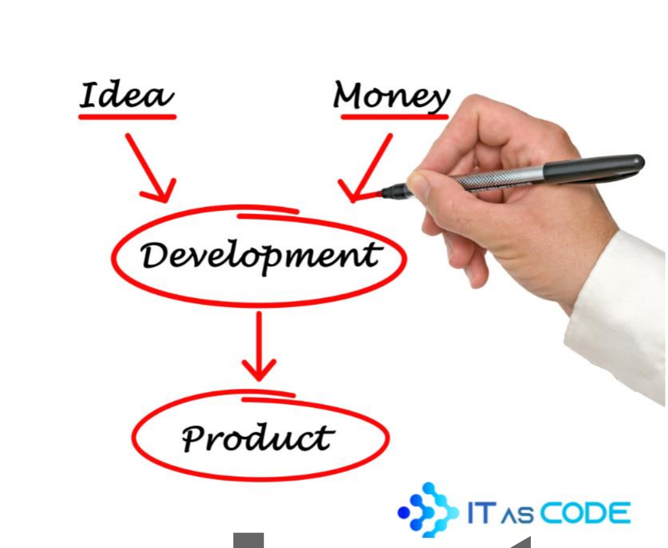
IT Consulting Services

The expert team provides consulting services

- DevOps, DevSecOps & SRE Consulting
- Cloud Modernization
- Cloud Optimization
- Technical Discovery
- Cloud Native Strategy

IT AS CODE

Information Technology as Code



Product Development

- We specialize in designing and developing cutting-edge products that are scalable, secure, and tailored to specific business objectives.
- Our expertise ensures seamless integration, superior performance, and a focus on user experience.
- From concept to launch, we guide clients through the entire lifecycle of product development, adapting to changing business needs and emerging technologies.

Information Technology as Code

Business Analytics Solutions & Consulting

Our expert team provides consulting services in:

- Data Visualization and Dashboarding
- Predictive and Prescriptive Analytics
- Data Modeling and Warehousing
- Business Intelligence Implementation
- Cloud Analytics Optimization
- Data Strategy and Governance

www.ITasCode.in | +91 970 5535 444

Hire from a pool of Trained Professionals!

Run through pre-screened, qualified candidates, increasing your interview-to-offer ratio.

Employ candidates already well trained on projects with hands-on experience. Candidates with expertise in DevOps and Cloud computing.

Make Highly Technical Hires

IT AS CODE
Pvt Ltd.,

For more details,

Visit: Academy.itascode.in
Email: contact@itascode.in
+91 970 5535 444
Google Play store App Name – ITasCode Academy

What makes us unique

Online & Classroom

Live online & Classroom classes enabling learners to attend classes from anywhere. Interact with the instructor and peers in real-time for effective learning experience

Practice Sessions

Daily practice sessions to reinforce learning. Practice sessions in real-world scenarios to prepare for the job.

Industry Experienced & Instructor

Access to industry experts for mentorship and guidance. Regular sessions for doubt resolution and personalized feedback. Entire course delivered solely by Murali Dulam; an experienced industry professional.

Q&A and Doubt Clearance

Weekly Q&A sessions for clarification of concepts and doubts. Special sessions for interview preparation and mock interviews

Hands-On Projects

Weekly Q&A sessions for clarification of concepts and doubts. Special sessions for interview preparation and mock interviews

Mock Interviews

Mock interviews with industry professionals to provide valuable feedback and build confidence. Opportunities to practice and improve interview skills in a safe and supportive environment.

Material and Recordings Access

Learn at your own pace with access to course materials and video recordings. The Right materials will be available whenever you require them at any point in the future

Internship Program

Dive into DevSecOps, Kubernetes, and cloud with hands-on projects using Agile Scrum methodology in this internship program.

Meet Your Coach

Explore transformative DevOps solutions, Kubernetes expertise, and AWS & Azure Cloud innovation guided by Murali Dulam, your dedicated coach at ITasCode. With extensive industry experience.

Founder @ ITasCode Pvt Ltd |
Solution Architecture | DevSecOps,
Kubernetes & Cloud Consultant |

22

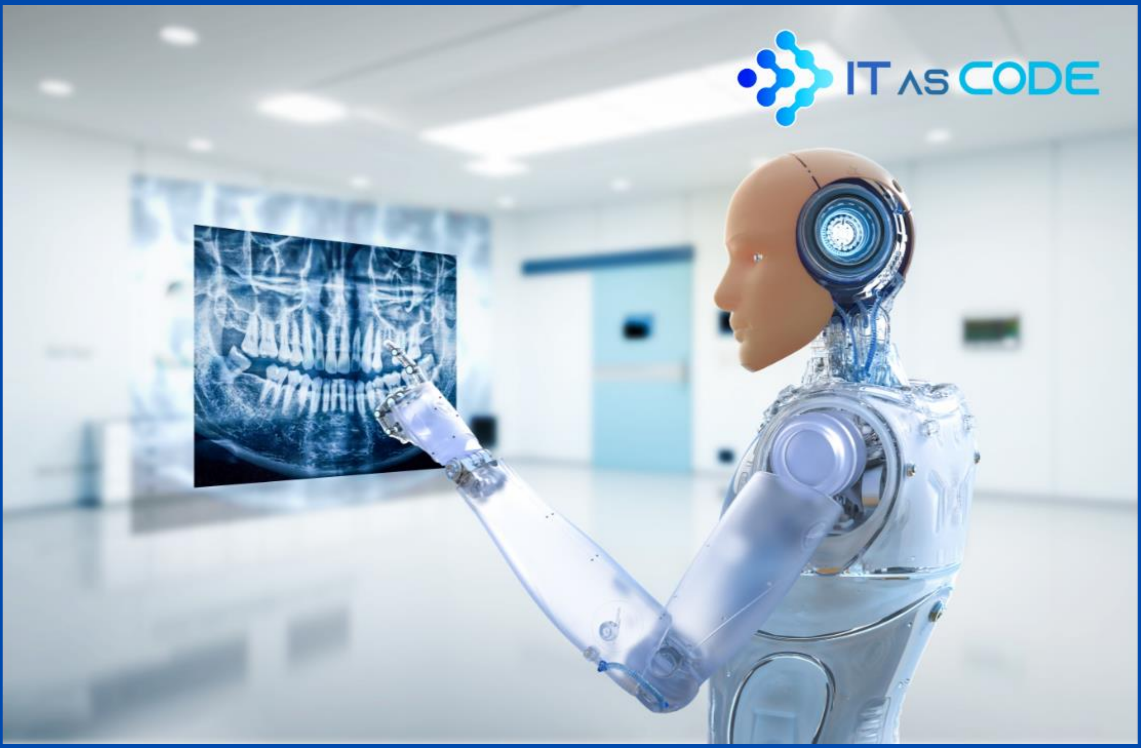
Years of Total IT Experience

75+

Batches Completed

5000+

Learners from 11 Counties



DevSecOps Training With AI ML & Realtime Projects

Duration : 120 Days

Online & Classroom

The Structure of the Training Program

- **Step 1**
- Training
- Receive structured training that covers fundamental to advanced topics in DevSecOps, Kubernetes, AWS, and Azure with topic-focused assignments..

- **Step 2**
- Internship Stage1
- Hands-on Projects with Basic Tasks, Group Discussions and Presentations & Agile Scrum & Engage in hands-on projects that introduce basic tasks, incorporating Agile Scrum methodologies.

- **Step 3**
- Internship Stage2
- Hands-on Projects with End-to-End Solutions, Group Discussions and Presentations & Agile Scrum & Advance to complex projects providing end-to-end solutions, further integrating Agile Scrum practices.

- **Step 4**
- Resume preparation
- Enhance your job prospects with a professionally crafted resume that showcases your skills and achievements effectively.

- **Step 5**
- Mock Interviews – Every 15 Days
- Prepare for success with personalized mock interviews, gaining confidence and valuable feedback to excel in real job interviews.

- **Step 6**
- Tech Connect
- Students will engage in structured sessions and informal interactions with seasoned tech experts and industry leaders, gaining invaluable insights into real-world challenges and streamlined solutions that will significantly simplify their future work

- **Step 7**
- Community Group
- Students will participate in regular community group meetings, both virtual and in-person, where they can engage in in-depth discussions on cutting-edge technologies, share their project experiences, and leverage the alumni network to gain access to unadvertised job openings and professional referrals.

- **Step 8**
- Award Internship Certificate
- Upon successful completion of all internship requirements and a comprehensive performance evaluation, interns will be awarded a prestigious internship certificate

- **Step 9**
- Forward Resumes With Clients/Internal Projects
- Increase your job opportunities by having your resume forwarded to top employers

DevSecOps - AWS & Azure Training With AI ML & Realtime Projects

section 1

- **Software Development Overview**
- Software Development Lifecycle Stages
- Challenges in Traditional Software Development Lifecycle Stages
- Streamline software delivery with Security & Importance of DevSecOps strategies
- Explore DevOps , DevSecOps & Cloud Job Roles
- Future DevSecOps, Cloud and Kubernetes Trends
- DevSecOps, Cloud and Kubernetes Role in AI, Machine Learning (ML) Edge Computing and IOT
- DevSecOps, Cloud and Kubernetes role in Low-Code and No-Code Platforms
- Importance of the Agile in the Software Development Process

section 2

- **Cloud Computing Overview**
- What is Cloud Computing? Definitions, history, and evolution.
- Traditional IT vs. Cloud Computing: A comparative analysis with Realtime use case
- Key Characteristics of Cloud Computing: On-demand self-service, broad network access, resource pooling, rapid elasticity, measured service.
- Benefits1 and Advantages of Cloud Computing: Cost savings, scalability, agility
- Cloud Computing Service and Deployment Models

section 3

- **Certification Roadmap**
- Why get certified? Career benefits.
- Choosing the right certification path.
- Certification Study tips.
- Certification exam Practice tests.
- Certification Resources
- Azure Cloud Certification
- AWS Cloud Certification
- Kubernetes Certification
- Terraform Certification
- Docker Certification
- Azure DevOps Certification
- Ansible Certification

DevSecOps - AWS & Azure Training With Realtime Projects

section 4

- **Virtualization**
- What is virtualization? Different ways to host your apps.
- On-premises, VMs, Cloud, and Containers explained.
- Pros and cons of each hosting type.
- Which one is best for you?
- Hands-on practice with the key technologies.

section 5

- **Linux Basics**
- What is Linux? Why use it?
- Linux Operating System overview
- Essential Linux commands
- Working with the command line
- Users, permissions, and package management
- Text editors
- Networking commands
- system administration Commands

section 6

- **Azure & AWS Cloud Overview**
- IAAS & PAAS Service in Azure & AWS
- Azure & AWS High availability
- Azure & AWS Storage
- Azure & AWS CLI

section 7

- **Git Version Control**
- What is Git? Why use it? (Version control, collaboration, real-world project examples)
- Install & configure Git (username, email, setting up SSH keys).
- Staging & committing changes (add, commit, status, writing good commit messages).
- Ignoring files and viewing differences (gitignore, diff, handling sensitive data).
- Branching: Creating, merging, and deleting (branch, merge, feature branches, hotfixes).
- Working with remote repositories (clone, push, pull, fetch, collaborating with a team).
- Pull requests and code reviews (PRs, giving and receiving feedback).
- Resolving merge conflicts (merge conflict resolution, common conflict scenarios).
- Basic Git workflow (feature branching, GitFlow - optional, choosing a workflow for your team).
- Using Git with a code editor (VSCode, etc., integrating Git into your development environment).
- Resetting and reverting changes (reset, revert, undoing mistakes).
- Hands-on exercises and assignments (simulating real-world development tasks).

Section 8

- **Build Tools (Maven & .NET)**
- What are build tools? Why do we need them?
- Problems without build tools (manual work, errors).
- Maven overview (POM, dependencies, lifecycle).
- Understanding Maven project structure.
- .NET build overview (MSBuild, NuGet).
- Understanding .NET project structure.
- Maven vs .NET builds.
- identifying key areas in the developer workflow for automation
- Hands-on: Building projects with Maven and .NET.

section 9

- **Azure DevOps**
- What is Azure DevOps? Why use it? (CI/CD, collaboration, project management)
- CI/CD basics (what it is, why it's important, benefits).
- Azure DevOps pipelines (building, testing, and releasing, stages, jobs).
- YAML pipelines (defining pipelines with code, syntax, templates).
- Working with code (Azure Repos, branching strategies, pull requests).
- Testing and code quality (SonarQube, static/dynamic analysis).
- Deployments (blue/green, deployment groups, environments).
- Package management (Azure Artifacts, publishing, consuming).
- Integrating with other tools (IaC - Terraform/Bicep, Kubernetes, configuration management).
- Agents (hosted, self-hosted, managing agents).
- Build types (different build configurations).
- Scheduled and triggered builds.
- Security and permissions in Azure DevOps.
- Monitoring and reporting.
- Hands-on exercises and assignments (real-world scenarios)

section 10

- **Docker**
- Why containers? (Problems with old ways, benefits of Docker: portability, efficiency)
- Docker basics (images, containers, volumes, networks, registries).
- Installing Docker (for your OS).
- Working with images (build, run, push, pull, inspect, layers).
- Working with containers (start, stop, manage, logs, attach, exec).
- Dockerfile (creating images, instructions: FROM, RUN, COPY, CMD, ENTRYPOINT, ENV, ARG).
- Docker Compose (multi-container apps, defining services, networking).
- Docker Hub (sharing images, public vs. private registries).
- Security (image scanning, vulnerability management).
- Container networking (bridge, host, overlay networks, port mapping).
- Docker volumes (persistent storage, data management).
- Multi-stage builds (optimizing image size).
- Advanced Dockerfile tricks (env vars, health checks).
- Docker security best practices.
- Docker monitoring & logging.
- Docker & CI/CD.
- Hands-on exercises (containerizing applications, using Docker Compose).

section 11

- **Kubernetes – Part1**
- Introduction to Kubernetes: What is Kubernetes? (Container orchestration platform). Why use it? (Scalability, portability, resilience). Benefits (Automated deployments, resource efficiency). Use cases (Microservices, web apps, data processing).
- Kubernetes Architecture: Master/control plane components: API Server (Interface for interacting with the cluster). etcd (Distributed key-value store for cluster data). Scheduler (Assigns pods to nodes). Controller Manager (Manages controllers for various resources).
- Kubernetes Architecture: Worker node components: kubelet (Node agent, manages pods). kube-proxy (Network proxy, handles service routing). Container runtime (Docker, containerd, CRI-O, runs containers).
- Kubernetes Architecture: Data flow and component interaction: How requests are processed, how deployments and updates are orchestrated, understanding the communication between components.
- Kubernetes Setup: Minikube for local development: Setting up a single-node Kubernetes cluster for learning and testing.
- Kubernetes Setup: Cloud-managed Kubernetes (AKS, EKS, GKE - comparison): Setting up and managing Kubernetes clusters on Azure, AWS, and Google Cloud, comparing features and pricing.
- Kubernetes Setup: Kubeadm for on-premises deployments: Setting up a production-ready Kubernetes cluster on your own infrastructure.
- Kubectl: Managing resources: Command-line tool for interacting with the Kubernetes API, creating, getting, describing, deleting, editing, applying, executing commands in containers, viewing logs.
- Pods: Basic pod concepts, pod lifecycle: The smallest deployable unit in Kubernetes, containing one or more containers, understanding the pod lifecycle (pending, running, succeeded, failed).
- Pods: Multi-container pods, use cases: Running multiple containers within a single pod, common use cases (sidecars, logging agents).
- Pods: Init containers, purpose and usage: Containers that run before the main application containers, used for initialization tasks.
- ConfigMaps: Managing application configuration: Storing configuration data as key-value pairs, injecting configuration into pods.
- Secrets: Handling sensitive data: Storing sensitive data like passwords and API keys, securely accessing secrets from pods.
- Namespaces: Logical isolation of resources: Creating isolated environments within a Kubernetes cluster.
- Services: Exposing applications (NodePort): Exposing a service on a static port on each node, accessible from outside the cluster.
- Services: Exposing applications (LoadBalancer): Exposing a service using a cloud provider's load balancer, making it accessible externally.

section 11

- **Kubernetes – Part2**
- Services: Exposing applications (ClusterIP): Exposing a service on an internal IP address, accessible only within the cluster.
- Services: Ingress controllers and external access: Managing external access to services using Ingress controllers, providing routing and TLS termination.
- Storage: Volumes (hostPath, emptyDir): Providing storage to containers, hostPath mounts a directory from the host node, emptyDir provides temporary storage within a pod.
- Storage: Persistent Volumes (PVs and PVCs): Abstracting storage from the underlying infrastructure, PersistentVolumes are provisioned by administrators, PersistentVolumeClaims are requested by users.
- Storage: Storage Classes and dynamic provisioning: Defining different types of storage using StorageClasses, dynamically provisioning PersistentVolumes on demand.
- Deployments: Rolling updates and rollbacks: Updating applications with zero downtime, rolling out new versions gradually, rolling back to previous versions if necessary.
- Deployments: Blue/green and canary deployments: Advanced deployment strategies for minimizing downtime and risk.
- Deployments: Horizontal Pod Autoscaler (HPA): Automatically scaling the number of pod replicas based on resource utilization.
- Deployments: ReplicaSets and managing replicas: Ensuring the desired number of pod replicas are running.
- Helm: Package manager, charts, repositories: Managing Kubernetes applications using Helm charts, packaging and deploying applications.
- Helm: Templating and hooks: Using Go templates for dynamic configuration of Helm charts, using hooks to perform actions during the Helm release lifecycle.
- Security: RBAC (Role-Based Access Control): Controlling access to Kubernetes resources based on roles and permissions.
- Security: Network Policies for pod isolation: Restricting network traffic between pods.
- Security: Pod Security Policies (PSPs) or Pod Security Admission (PSA): Defining security constraints for pods.
- Security: Vulnerability scanning of images and containers: Identifying and mitigating security vulnerabilities in container images and running containers.

section 11

- **Kubernetes – Part3**
- DaemonSets: Running a pod on each node: Ensuring a specific pod runs on every node in the cluster.
- Jobs and CronJobs: Batch and scheduled tasks: Running one-time tasks and scheduled tasks.
- StatefulSets: Managing stateful applications: Managing applications that require persistent storage and stable network identities.
- Networking: CNI (Container Network Interface): Interface for plugging in different networking solutions into Kubernetes.
- Networking: Service Mesh (Istio - traffic management, security, observability): Adding a layer of infrastructure for managing communication between microservices.
- Monitoring & Logging: Prometheus and Grafana: Collecting and visualizing metrics from Kubernetes and applications.
- Monitoring & Logging: ELK stack (Elasticsearch, Logstash, Kibana): Centralized logging and analysis.
- GitOps: Argo CD and Flux: Using Git as the source of truth for deployments and infrastructure management.
- Operators: Managing complex applications with custom controllers: Extending Kubernetes functionality by creating custom controllers for managing complex applications.
- Troubleshooting: Common Kubernetes issues and debugging techniques: Diagnosing and resolving common Kubernetes problems.
- Design Patterns: Best practices for Kubernetes application development: Designing and building scalable and resilient applications for Kubernetes.
- Resource Management: Limits and requests for CPU and memory: Allocating resources to containers and pods.
- Liveness and Readiness Probes: Health checks for pods: Ensuring that pods are healthy and ready to serve traffic.
- Advanced Scheduling: Node selectors, taints, and tolerations: Controlling where pods are scheduled based on node labels and conditions.
- Kubernetes Federation: Managing multiple clusters: Managing and coordinating multiple Kubernetes clusters.
- Serverless on Kubernetes: Running serverless functions on Kubernetes.
- Hands-on Projects: Real-world application deployments: Practical exercises and projects to reinforce learning.

section 12

- **Generative AI, ML & IOT - Cloud & DevSecOps**

- I. GenAI & ML Basics
- What is GenAI? - Defining Generative AI and its capabilities.
- Why it's important. - Exploring the impact and potential of GenAI.
- ML refresh. - Brief overview of key machine learning concepts.
- Cloud & DevSecOps for GenAI. - Why cloud platforms and secure development are essential.

- II. Building GenAI Apps
- App basics. - Understanding the structure of a GenAI application.
- Code setup. - Configuring a development environment.
- Example apps. - Building simple GenAI applications.

- III. GenAI & Copilot
- GenAI platforms. - Exploring cloud platforms for GenAI development.
- Model deployment. - Deploying trained models for use.
- Using Copilots. - Integrating AI assistants into workflows.

- IV. Small Language Models (SLMs)
- What are SLMs? - Defining and explaining smaller language models.
- How they work. - Simplified explanation of SLM operation.
- SLM uses. - Practical applications of SLMs.

- V. Large Language Models (LLMs) & RAG
- What are LLMs? - Understanding the power of large language models.
- RAG explained. - How Retrieval Augmented Generation works.
- Using LLMs. - Prompt engineering and fine-tuning.

- VI. Agentic AI
- What is it? - Defining agentic AI and autonomous agents.
- How it's used. - Applications of agentic AI.

- VII. Cloud for GenAI
- Azure . – Azure OpenAI services for GenAI and ML.
- AWS . - AWS Bedrock services for GenAI and ML.

- VIII. DevSecOps
- Security basics. - Fundamental security concepts for GenAI.
- DevSecOps for GenAI. - Secure development practices for GenAI.

- IX. GenAI & IoT
- IoT basics. - Introduction to the Internet of Things.
- GenAI for IoT. - Combining GenAI with IoT data.

- X. Final Project
- Build a GenAI app. - Developing a practical GenAI application.
- Use cloud & DevSecOps. - Applying cloud services and secure practices.

section 13

- **Ansible Automation**

- I. Introduction to Ansible
- What is Ansible? Benefits of automation, use cases.
- Ansible Architecture: Control node, managed nodes, modules, playbooks.
- Idempotency and Declarative Language.

- II. Ansible Installation
- Installing Ansible on the control node (Linux, macOS, Windows).
- Setting up the environment.
- Verifying the installation.

- III. Ansible Ad-hoc Commands
- Basic ad-hoc commands: Running simple tasks on managed nodes.
- Using -m for modules, -a for arguments.
- Examples: pinging hosts, running commands, copying files.

- IV. Ansible Inventory
- Understanding the Ansible inventory file.
- Defining hosts and groups.
- Using variables in the inventory.
- Dynamic inventory (brief overview).

- V. Ansible Ad-hoc Commands & Inventory
- Combining ad-hoc commands with inventory groups.
- Targeting specific hosts or groups.
- Running commands on multiple hosts.

- VI. Ansible Configuration & Playbook
- Introduction to Ansible Playbooks: YAML-based automation scripts.
- Playbook structure: plays, tasks, modules.
- Basic playbook example: running a command on a group of hosts.

- VII. Ansible Configuration File
- Understanding the ansible.cfg file.
- Configuring Ansible settings: inventory path, remote user, etc.

- VIII. YAML Script
- Introduction to YAML: Syntax and data structures.
- YAML for Ansible: Writing playbooks and configuration files.

section 13

- **Ansible Automation**

- IX. YAML & Ansible PlayBook
- Combining YAML and Ansible: Writing more complex playbooks.
- Using multiple tasks, variables, and modules.

- X. Setup additional Ansible Managed nodes
- Preparing managed nodes for Ansible control.
- SSH key-based authentication.
- Connectivity testing.

- XI. Modules - Playbooks
- Exploring Ansible modules: Core, extras, and community modules.
- Using modules in playbooks: Managing packages, services, files, etc.

- XII. Playbooks with Multi tasks
- Creating playbooks with multiple tasks.
- Controlling task execution order.

- XIII. Notify and Handlers
- Using handlers for event-driven automation.
- notify and listen: Triggering handlers based on task results.

- XIV. Gatherfacts-Material
- Understanding Ansible facts: Gathering information about managed nodes.
- Using facts in playbooks: Conditional logic, variable substitution.

- XV. Conditions-Packages using conditions
- Conditional execution: Using when clauses to control task execution.
- Managing packages based on conditions.

- XVI. Notify and Handlers - Gatherfacts-Conditions-Packages using conditions
- Combining handlers, facts, and conditions in playbooks.
- Complex automation scenarios.

- XVII. Variables
- Defining and using variables in Ansible: Inventory variables, playbook variables, extra vars.
- Variable precedence.

section 13

- **Ansible Automation**

- IX. YAML & Ansible PlayBook
- Combining YAML and Ansible: Writing more complex playbooks.
- Using multiple tasks, variables, and modules.

- X. Setup additional Ansible Managed nodes
- Preparing managed nodes for Ansible control.
- SSH key-based authentication.
- Connectivity testing.

- XI. Modules - Playbooks
- Exploring Ansible modules: Core, extras, and community modules.
- Using modules in playbooks: Managing packages, services, files, etc.

- XII. Playbooks with Multi tasks
- Creating playbooks with multiple tasks.
- Controlling task execution order.

- XIII. Notify and Handlers
- Using handlers for event-driven automation.
- notify and listen: Triggering handlers based on task results.

- XIV. Gatherfacts-Material
- Understanding Ansible facts: Gathering information about managed nodes.
- Using facts in playbooks: Conditional logic, variable substitution.

- XV. Conditions-Packages using conditions
- Conditional execution: Using when clauses to control task execution.
- Managing packages based on conditions.

- XVI. Notify and Handlers - Gatherfacts-Conditions-Packages using conditions
- Combining handlers, facts, and conditions in playbooks.
- Complex automation scenarios.

- XVII. Variables
- Defining and using variables in Ansible: Inventory variables, playbook variables, extra vars.
- Variable precedence.

section 13

- **Ansible Automation**

- XVIII. Tomcat Setup
- Automating Tomcat installation and configuration using Ansible.
- Example playbook for Tomcat deployment.

- XIX. Ansible Tags Error handling
- Using tags to organize and run specific parts of a playbook.
- Error handling in playbooks: block, rescue, always.

- XX. Ansible Roles
- Introduction to Ansible Roles: Reusable units of automation.
- Role structure: tasks, vars, templates, handlers, etc.

- XXI. Ansible Datadog Roles Realtime use case
- Creating and using Ansible roles for Datadog integration.
- Real-time monitoring use case.

- XXII. Ansible Real-time Use Case Project
- Capstone project: Building a real-world automation solution using Ansible.
- Example projects: Web server deployment, database configuration, cloud infrastructure management.

section 14

- **Jenkins**

- I. Introduction to Jenkins
- What is Jenkins? (Automation server, CI/CD tool)
- Why Jenkins? (Benefits, use cases, integration with other tools)
- Jenkins Architecture (Master-slave architecture, agents)

- II. Jenkins Installation
- Installing Jenkins (Different methods, platform-specific instructions)
- Post-installation Setup (Initial setup wizard, creating admin user)

- III. Jenkins Configuration Settings
- Global Configuration (Managing plugins, security settings, system settings)
- Managing Plugins (Installing, updating, and uninstalling plugins)
- Security Configuration (User authentication, authorization, access control)
- System Configuration (Setting up paths, tools, and other configurations)

- IV. Jenkins Maven Build
- Maven Integration (Configuring Maven in Jenkins)
- Creating a Maven Job (Building a Maven project in Jenkins)
- Build Triggers (Triggering builds automatically)
- Build Artifacts (Archiving and managing build artifacts)

- V. Setting up Tomcat Server
- Tomcat Installation (Installing and configuring Tomcat)
- Tomcat Deployment (Deploying web applications to Tomcat)

- VI. CI/CD with Jenkins and Tomcat Server
- CI/CD Pipeline (Setting up a basic CI/CD pipeline)
- Automated Build and Deployment (Building and deploying to Tomcat automatically)
- Testing Integration (Integrating automated tests into the pipeline)

section 14

- **Jenkins**

- VII. Setting up Jenkins Agents/Slaves
- Agent/Slave Configuration (Connecting agents to the Jenkins master)
- Agent Launch Methods (SSH, JNLP, other methods)
- Distributed Builds (Running builds on different agents)

- VIII. Pipeline as Code (Jenkinsfile)
- Introduction to Pipelines (What are pipelines, benefits)
- Jenkinsfile Syntax (Declarative and scripted pipelines)
- Pipeline Stages (Defining stages in a pipeline)
- Pipeline Examples (Creating different types of pipelines)

- IX. CI/CD with Jenkins and Kubernetes
- Blue Ocean (Enhanced Jenkins UI)
- Multibranch Pipelines (Managing builds for multiple branches)
- Docker Integration (Building and deploying Docker images)
- Kubernetes Integration (Configuring Kubernetes in Jenkins)
- Deploying to Kubernetes (Deploying applications to Kubernetes using Jenkins)
- Kubernetes Pipeline Example (Building and deploying to Kubernetes in a pipeline)

- X. Jenkins Real-time Project
- Real-time Project (Applying Jenkins to a real-world CI/CD scenario, integrating all learned concepts).

section 15

- **Datadog Observability**

- I. Intro to Observability
- Monitoring vs. Observability: Understanding the difference between traditional monitoring and the more comprehensive approach of observability.
- Why it matters: Exploring the benefits of observability for understanding complex systems and improving performance.

- II. Datadog Basics
- What is Datadog?: Overview of the Datadog platform and its capabilities for monitoring and observability.
- Key features: Highlighting the core features of Datadog, such as metrics, traces, logs, and dashboards.

- III. Datadog Setup
- Account setup: Steps to create a Datadog account and get started.
- Agent install: Instructions for installing the Datadog agent on different

- IV. Docker Monitoring
- Datadog & Docker: How Datadog integrates with Docker to monitor containers.
- Container metrics & logs: Collecting and analyzing metrics and logs from Docker containers.

- V. Kubernetes Monitoring
- Datadog & Kubernetes: Integrating Datadog with Kubernetes for cluster and application monitoring.
- Pod & service monitoring: Monitoring the health and performance of Kubernetes pods and services.

- VI. Azure DevOps Monitoring
- Datadog & Azure DevOps: Connecting Datadog to Azure DevOps for CI/CD pipeline monitoring.
- Pipeline monitoring: Tracking the performance and health of Azure DevOps pipelines.

- VII. Project
- Real-time monitoring project: Hands-on project applying Datadog to monitor a real-world application or infrastructure.

section 16

- **Jira - Planning & Tracking**

- I. Introduction to Jira
- What is Jira? (Agile project management tool, issue tracking)
- Why Jira? (Benefits, use cases, team collaboration)
- Jira's Core Concepts (Projects, issues, workflows, boards)

- II. Jira Configuration & Project Setup
- Jira Administration (User management, permissions, global settings)
- Creating a New Project (Choosing a project template, configuring settings)
- Project Roles and Permissions (Defining roles, assigning permissions)
- Customizing Workflows (Creating and modifying workflows, transitions)
- Setting up Issue Types (Defining issue types, custom fields)

- III. Jira for Planning & Tracking
- Creating and Managing Issues (Creating issues, adding descriptions, attachments)
- Prioritizing Issues (Using priority levels, ranking issues)
- Assigning Issues (Assigning issues to team members)
- Tracking Progress (Updating issue status, adding comments)
- Using Boards (Scrum boards, Kanban boards, visualizing workflow)
- Creating Sprints (Planning sprints, adding issues to sprints)
- Reporting and Analytics (Generating reports, tracking progress, burndown charts)

- IV. Jira Integrations
- Integrating Jira with Datadog (Connecting Jira with Datadog for monitoring and alerting)
- Integrating Jira with Azure DevOps (Connecting Jira with Azure DevOps for CI/CD integration)
- Other Integrations (Exploring other Jira integrations)

- V. Jira Real-time Project
- Real-time Project: Applying Jira to a real-world project scenario, utilizing all the learned concepts.

section 17

- **GitHub Actions Automation**

- I. Introduction to GitHub Actions
- GitHub Actions Overview: What are GitHub Actions? (CI/CD platform, automation)
- Why GitHub Actions? (Benefits, use cases, integration with GitHub)
- Key Concepts: Workflows, jobs, steps, actions, runners.

- II. GitHub Actions Build Workflow & Syntax
- Workflow Files: Creating and configuring workflow files (.github/workflows/*.yml).
- YAML Syntax: Understanding the YAML syntax used in workflow files.
- Workflow Triggers: Events that trigger workflows (push, pull request, schedule, etc.).
- Defining Jobs: Specifying jobs within a workflow, dependencies between jobs.
- Defining Steps: Defining individual steps within a job, using actions and commands.
- Building and Testing: Automating build and test processes.
- Releases and Tags: Automating release creation and tagging based on events.

- III. GitHub Action Runners
- What are Runners?: Understanding GitHub-hosted and self-hosted runners.
- GitHub-hosted Runners: Using GitHub's infrastructure for running workflows.
- Self-hosted Runners: Setting up and managing your own runners.
- Choosing the Right Runner: Considerations for selecting a runner type.

- IV. GitHub Actions with Kubernetes
- Kubernetes Integration: Connecting GitHub Actions to Kubernetes.
- Deploying to Kubernetes: Automating deployments to Kubernetes clusters.
- Kubernetes Actions: Using pre-built actions for Kubernetes interactions.
- Example Workflow: Building and deploying a containerized application to Kubernetes.

- V. GitHub Actions Real-time Project
- Real-time Project: Applying GitHub Actions to a real-world CI/CD scenario, integrating all the learned concepts. This could include building, testing, containerizing, and deploying an application.

section 18

- **HashiCorp Vault Secret Management**

- I. Introduction to HashiCorp Vault
- What is HashiCorp Vault? (Secret management, data protection, access control)
- Why Vault? (Benefits, use cases, security best practices)
- Vault's Core Concepts: Secrets, authentication methods, policies, leases.

- II. HashiCorp Vault Overview
- Vault Architecture: Server, storage backend, authentication plugins, secrets engines.
- Key Features: Secret storage, dynamic secrets, access control, auditing.
- Dynamic Secrets: Generating secrets on demand (e.g., database credentials, AWS credentials).
- Secrets Engines: Using different secrets engines (e.g., KV, database, AWS).
- Use Cases: Managing API keys, database credentials, certificates, and other sensitive data.

- III. HashiCorp Vault with Docker Server
- Setting up Vault in Docker: Running Vault in a Docker container.
- Initializing and Unsealing Vault: Understanding the initialization and unsealing process.
- Basic Vault Operations: Storing, retrieving, and deleting secrets.
- Using the Vault CLI: Interacting with Vault using the command-line interface.

- IV. HashiCorp Vault with GitHub Authorization
- GitHub Authentication Method: Configuring Vault to authenticate using GitHub.
- Access Control with Policies: Defining policies to control access to secrets based on GitHub identities.
- Example Scenario: Granting access to secrets based on GitHub team membership.

- V. HashiCorp Vault with Kubernetes
- Kubernetes Integration: Setting up Vault to run in a Kubernetes cluster.
- Authentication with Kubernetes: Using Kubernetes service accounts to authenticate with Vault.
- Secret Injection: Automatically injecting secrets into Kubernetes pods.
- Vault Agent: Using the Vault Agent for secret management in Kubernetes.
- Example Scenario: Deploying an application to Kubernetes that retrieves secrets from Vault.

- VI. HashiCorp Vault Real-time Project
- Real-time Project: Applying HashiCorp Vault to a real-world scenario, integrating all the learned concepts. This could involve securing a microservices application deployed in Kubernetes, using GitHub authentication for developers, and dynamically generating database credentials.

section 19 & 20

- **DevSecOps Tools: Securing the Development Lifecycle**

- **Module 1: Introduction to DevSecOps and Core Concepts**
- The importance of security in modern software development
- Understanding the DevSecOps lifecycle
- Shift-left security principles
- Key DevSecOps practices and benefits

- **Module 2: Secrets Management with HashiCorp Vault**
- Introduction to secrets management and its challenges
- Overview of HashiCorp Vault architecture and features
- Vault installation and configuration
- Authentication methods in Vault (e.g., AppRole, Kubernetes)
- Secret engines and their usage (e.g., KV, database, PKI)
- Dynamic secrets and their advantages
- Access control policies and best practices
- Vault integration with CI/CD pipelines
- Hands-on labs: Setting up Vault, managing secrets, integrating with applications

- **Module 3: Static Code Analysis with SonarQube**
- Introduction to static code analysis and its benefits
- Overview of SonarQube architecture and features
- SonarQube installation and configuration
- Understanding SonarQube rules and quality profiles
- Analyzing code quality and security vulnerabilities
- Customizing SonarQube rules and reports
- Integrating SonarQube with CI/CD pipelines
- Hands-on labs: Analyzing code, setting up quality gates, customizing rules.

- **Module 4: Dynamic Application Security Testing (DAST) with OWASP ZAP**
- Introduction to dynamic application security testing (DAST)
- Overview of OWASP ZAP architecture and features
- ZAP installation and configuration
- Using ZAP for automated and manual vulnerability scanning
- Generating and interpreting ZAP reports
- Integrating ZAP with CI/CD pipelines
- Hands-on labs: Performing scans, analyzing reports, configuring ZAP.

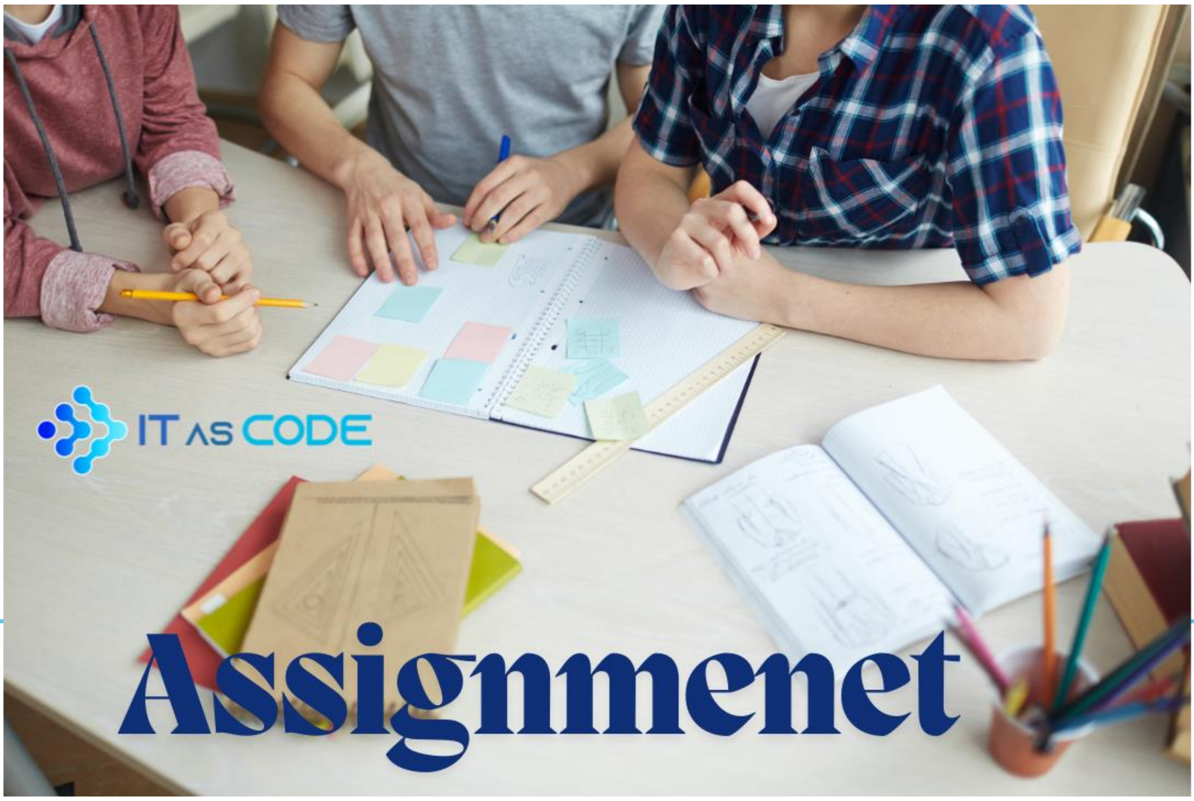
section 21 & 22

- **DevSecOps Tools: Securing the Development Lifecycle**

- Module 5: Container and Artifact Vulnerability Scanning with JFrog Xray
- Introduction to software composition analysis (SCA)
- Overview of JFrog Xray architecture and features
- Xray installation and configuration
- Integrating Xray with JFrog Artifactory
- Scanning container images and other artifacts for vulnerabilities
- Understanding Xray's vulnerability database and policies
- Generating and interpreting Xray reports
- Integrating Xray with CI/CD pipelines
- Hands-on labs: Scanning containers, creating policies, analyzing reports.

- Module 6: Kubernetes Security with KubeLinter
- Introduction to Kubernetes security best practices
- Overview of KubeLinter architecture and features
- KubeLinter installation and configuration
- Using KubeLinter for static analysis of Kubernetes manifests
- Understanding KubeLinter checks and rules
- Generating and interpreting KubeLinter reports
- Integrating KubeLinter with CI/CD pipelines
- Hands-on labs: Linting Kubernetes manifests, customizing rules, integrating with pipelines.

- Module 7: Integrating DevSecOps Tools into CI/CD Pipelines
- Designing secure CI/CD pipelines
- Infrastructure as Code (IaC) security
- Integrating Vault, SonarQube, ZAP, Xray, and KubeLinter into pipelines
- Automating security testing and vulnerability scanning
- Implementing security gates and approvals
- Monitoring and reporting on security metrics
- Best practices for DevSecOps pipeline automation
- Hands-on labs: Building a secure CI/CD pipeline using the covered tools.



Topic Based Assignments

•**Assignment Distribution:** Regular distribution of assignments tied to training modules.
•**Specific Topic Focus:** Assignments target specific learning objectives within DevSecOps, Kubernetes, and cloud.
•**Practical Application Exercises:** Hands-on exercises reinforce theoretical concepts.
•**Feedback & Review:** Thorough feedback on assignments to ensure understanding.
•**Reinforced Learning:** Assignments solidify knowledge before moving to practical projects.

Internship with Realtime project setup

- **Internship Stage1**
- Hands-on Projects with Basic Tasks, Group Discussions and Presentations & Agile Scrum & Engage in hands-on projects that introduce basic tasks, incorporating Agile Scrum methodologies.

- **Internship Stage2**
- Hands-on Projects with End-to-End Solutions, Group Discussions and Presentations & Agile Scrum & Advance to complex projects providing end-to-end solutions, further integrating Agile Scrum practices.

Academy.ITasCode.in | +91 970 5535 444

Course Duration: 120 Days

Resume Preparation

- Personalized Guidance:
- One-on-one sessions with ITasCode professionals.
- Tailored advice based on individual skills and experience.

- Resume Optimization:
- Crafting a professional and visually appealing resume.
- Ensuring compatibility with Applicant Tracking Systems (ATS).

- Skill Highlighting:
- Strategically showcasing relevant technical skills (DevSecOps, Kubernetes, cloud).
- Emphasizing practical project experience and contributions.

- Industry Alignment:
- Tailoring the resume to meet current IT industry standards.
- Highlighting knowledge of DevSecOps, Kubernetes, Cloud and Agile principles.

- Interview Readiness:
- Positioning the resume to increase the likelihood of securing interviews.
- Creating a resume that captures the attention of hiring managers.

Academy.ITasCode.in | +91 970 5535 444

Course Duration: 120 Days

Mock Interviews

- **Custom Interviews:**
- Questions tailored to your job goals.
- Realistic practice for tech and behavior questions.
- **Expert Interviewers:**
- Mock interviews by ITasCode staff or industry pros.
- Real-feel interview experience.
- **Detailed Feedback:**
- Evaluation of tech skills, communication, and body language.
- Helpful tips for improvement.
- **Build Confidence:**
- Less interview anxiety.
- More confident self-presentation.
- **Improve Skills:**
- Find and fix weak points.
- Practice answering tough questions.
- **Prepare for Reality:**
- Learn employer expectations.
- Practice showing off your skills.
- Experience interview pressure.
- **Focus on Key Areas:**
- Practice cloud, DevSecOps, and Kubernetes questions.
- Practice agile and scrum questions.
- **Review and Analyze:**
- Review of the mock interview.
- Analyze your strengths and weaknesses.

Academy.ITasCode.in | +91 970 5535 444

Course Duration: 120 Days

ITasCode Tech Connect

- **Structured Learning:** Expert-led workshops and interactive Q&A on industry trends.
- **Informal Networking:** Casual chats and mentorship for building professional connections.
- **Expert Insights**: Real-world case studies and practical demonstrations from experienced pros.
- **Leadership Perspectives**: Discussions on tech strategy and decision-making from industry leaders.
- **Practical Application**: Learning Cloud, AI/ML, security, scalability, and CI/CD solutions for real-world scenarios.
- **Problem-Solving Skills**: Efficient troubleshooting and optimization techniques for tech challenges.
- **Career Advancement**: Developing in-demand skills and building a valuable professional network.

Academy.ITasCode.in | +91 970 5535 444

Course Duration: 120 Days

ITasCode Student Community Group

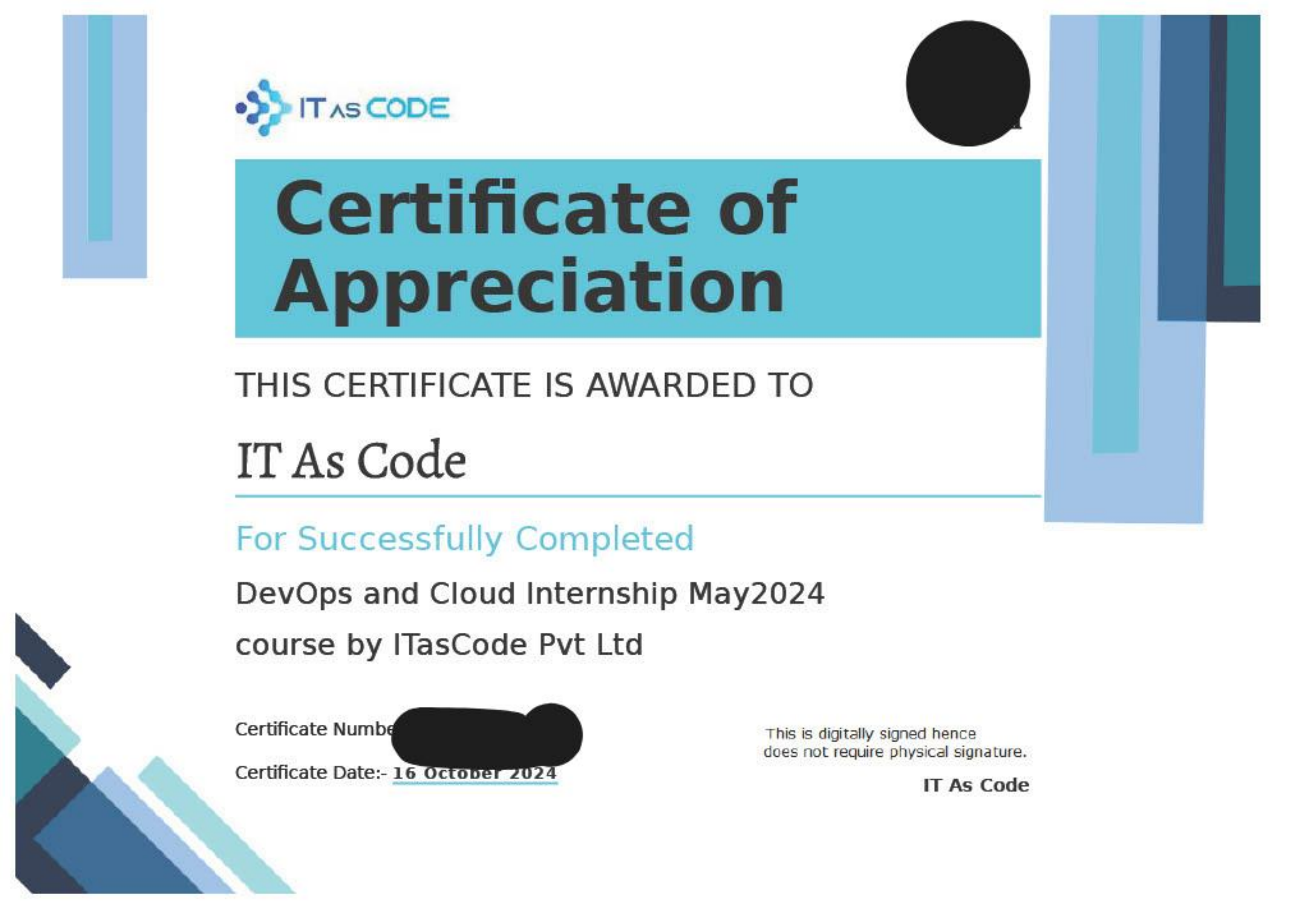
•**Regular Meetings:**
•Scheduled gatherings, both online and offline.
•Consistent opportunities for interaction and learning.
•**In-Depth Tech Discussions:**
•Exploration of cutting-edge technologies and trends.
•Sharing of knowledge and insights among participants.
•**Project Experience Sharing:**
•Students present and discuss their project work.
•Peer learning and feedback on practical applications.
•**Alumni Network Access:**
•Leveraging connections with program alumni.
•Building professional relationships with experienced individuals.
•**Unadvertised Job Opportunities:**
•Access to job openings not publicly listed.
•Gaining insider information on potential career paths.
•**Professional Referrals:**
•Obtaining referrals from alumni for job applications.
•Increasing the likelihood of securing interviews.
•**Community Building:**
•Creating a network of like minded individuals.
•Support from peers and those who have completed the program.

Academy.ITasCode.in | +91 970 5535 444

Course Duration: 120 Days

Award Internship Certificate

- **Successful Completion:**
- Interns must fulfill all program requirements.
- This includes completing assigned tasks and projects.
- **Comprehensive Performance Evaluation:**
- Interns will be assessed based on their overall performance.
- Evaluation will cover technical skills, participation, and work ethic.
- **Prestigious Internship Certificate:**
- A formal document will be awarded.
- The certificate signifies successful completion and achievement.
- The certificate adds value to the intern's professional portfolio.



IT AS CODE

[redacted]

Certificate of Appreciation

THIS CERTIFICATE IS AWARDED TO

IT As Code

For Successfully Completed

DevOps and Cloud Internship May2024

course by ITasCode Pvt Ltd

Certificate Numbe[redacted]

Certificate Date:-

This is digitally signed hence
does not require physical signature.

IT As Code

Academy.ITasCode.in | +91 970 5535 444

Course Duration: 120 Days

Course Completion Certificate

•Successful Completion:

•I Students must fulfill all program requirements.

•This includes completing assigned tasks and projects.

•Comprehensive Performance Evaluation:

•Students will be assessed based on their overall performance.

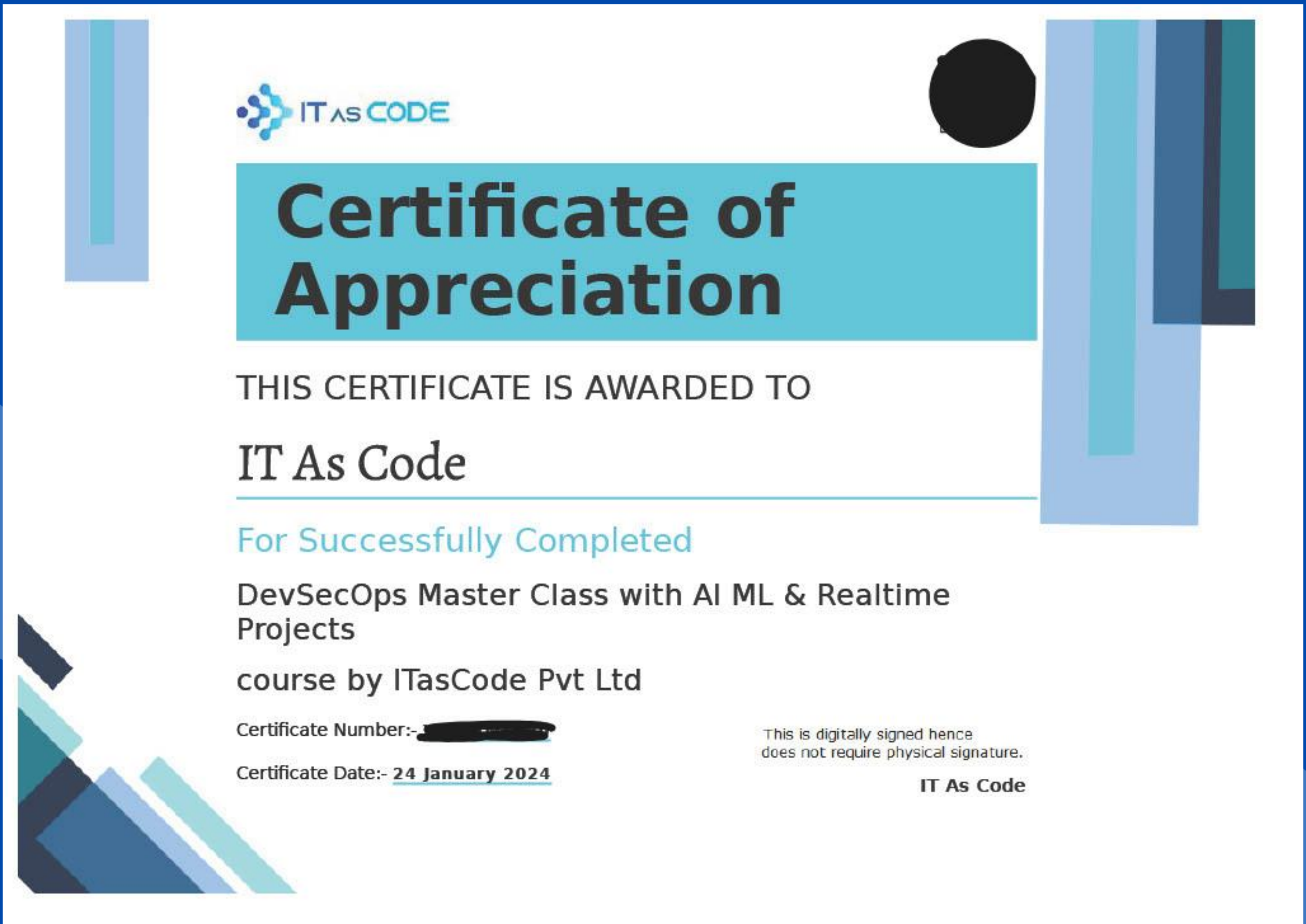
•Evaluation will cover technical skills, participation, and work ethic.

•Prestigious Course Completion Certificate:

•A formal document will be awarded.

•The certificate signifies successful completion and achievement.

•The certificate adds value to the student's professional portfolio.



IT AS CODE

[redacted]

Certificate of Appreciation

THIS CERTIFICATE IS AWARDED TO

IT As Code

For Successfully Completed

DevSecOps Master Class with AI ML & Realtime Projects

course by ITasCode Pvt Ltd

Certificate Number:-[redacted]

Certificate Date:- 24 January 2024

This is digitally signed hence does not require physical signature.

IT As Code

Academy.ITasCode.in | +91 970 5535 444

Course Duration: 120 Days

Azure Cloud Architecture & Administration Training With AI ML & Realtime Projects

Duration : 75 Days

Online & Classroom

The Structure of the Training Program

- **Step 1**
- Training
- Receive structured training that covers fundamental to advanced topics in Azure with AI ML & IOT topic-focused assignments..

- **Step 2**
- Internship Stage1
- Hands-on Projects with Basic Tasks, Group Discussions and Presentations & Agile Scrum & Engage in hands-on projects that introduce basic tasks, incorporating Agile Scrum methodologies.

- **Step 3**
- Internship Stage2
- Hands-on Projects with End-to-End Solutions, Group Discussions and Presentations & Agile Scrum & Advance to complex projects providing end-to-end solutions, further integrating Agile Scrum practices.

- **Step 4**
- Resume preparation
- Enhance your job prospects with a professionally crafted resume that showcases your skills and achievements effectively.

- **Step 5**
- Mock Interviews – Every 15 Days
- Prepare for success with personalized mock interviews, gaining confidence and valuable feedback to excel in real job interviews.

- **Step 6**
- Tech Connect
- Students will engage in structured sessions and informal interactions with seasoned tech experts and industry leaders, gaining invaluable insights into real-world challenges and streamlined solutions that will significantly simplify their future work

- **Step 7**
- Community Group
- Students will participate in regular community group meetings, both virtual and in-person, where they can engage in in-depth discussions on cutting-edge technologies, share their project experiences, and leverage the alumni network to gain access to unadvertised job openings and professional referrals.

- **Step 8**
- Award Internship Certificate
- Upon successful completion of all internship requirements and a comprehensive performance evaluation, interns will be awarded a prestigious internship certificate

- **Step 9**
- Forward Resumes With Clients/Internal Projects
- Increase your job opportunities by having your resume forwarded to top employers

Azure Cloud Architecture & Administration Course

Introduction to Cloud Computing

section 1

- What is Microsoft Azure?
- Microsoft Azure Services
- Creating a Microsoft Azure Account
- Azure CLI and Azure PowerShell
- Managing Azure resources and subscriptions
- Azure Resource Manager
- Datacenters, Regions, Region Pairs, Availability Zones and Geographies
- Resource Groups, Account, Subscription, Management Groups And Tenant

Realtime project Application Architecture

Virtual Machine Basics

section 2

- VM Instance Types, States & Disks)
- Creating a Linux VM in Azure
- Install lamp-stack and IIS WebServer on Azure VMs
- Creating a DB Resource & Understanding a VM Components
- Creation of Linux VM and Generating Pub&Pvt Keys in Azure
- VM Sizes, Families, Changing Size and Reserved & Spot Instances in Azure
- Fault & Update Domains and Availability Options in Azure
- Usage of Extensions & Features option in VM of Azure
- Azure CLI Introduction
- Creating Resource Group, Port, IP Address and VM using Azure-CLI
- Creating Custom Images for Linux and Windows VM and Shared Image Gallery in Azure
- Understanding VMSS Virtual Machine Scale Set and Load Balancer Concepts
- VMSS Configuring using Application LoadBalancer and Application Gateway in Azure
- Overview of Guest & Host Update Management and Creating Bastion Host in Azure

Azure Storage

section 3

- Storage needs of Organisations and Core Storage Services:
  Blobs Files Queues Tables & Disks in Azure
- StorageAccount with Containers Blobs and Creating StorageExplorer in Azure
- Storage Performance Tiers, Replication Options and Access Tiers in Azure
- Storage Account: Types, Endpoints and Understand StorageAccount Pricing
- Lock on ResourceManager, Enable/Manage SoftDelete on Containers, Blob – SavingPeriodically, PointInTime Restore and Recovering Deleted StorageAccount
- Azure AccessTier Lifecycle Concepts and Install and Configure AZ-CLI
- Creating StorageAccount Containers BlobStorage and Files by Using Azure-CLI Commands
- Authorization to AzStorageAccounts by SAS Token, AZCOPY CLI and Data: Disk Box Heavy to Transfer from Local, AWS-S3 and Google-StorageBucket (Offline/On-premise))
- Content Delivery Networks (CDN) feature and Mounting the NFS by AzureFileShare feature to Windows/Linux VMs
- Create and Attach a Disk to WinVM LinuxVM and Partition Disk; DataDisk ManagedDisk UnmanagedDisk and Snapshots for Backup & Recovery in Azure
- Creation of Azure SQL Database and Understand differences between Azure SQL Databases and Azure SQL Managed Instances
- Create Azure Database for MySQL Server for DB Creation & Replication

Azure Cloud Operations

section 4

- Applications on Azure Cloud
- ServiceStatus & ActivityLogs at Subscription Level and Creating Alerts
- or High-Low CPU Utilization for Health Monitoring in Azure
- (Monitoring VMs Configuration:CPU Memory Disk Usage Metrics and NSG: FlowLogs TrafficAnalytics and NetworkConnectionWatcher in Azure
- Crate an AzureAutomationRunBook to Start Stop Terminate VM using ActionGroup and Use WorkBooks to get info from Templates and Explore options under AzureMonitoring
- Using AzureMonitorLogs feature and Querying Logs by using KQL – Kusto Query Language and Creating a Time Series Chart using Azure Monitor Logs
- BC & DR: Replication Failover & Fallback Types; Azure Site Recovery (ASR) the One-Stop Public Cloud Solution for Disaster Recovery

Migration

section 5

- Migration Overview
- Azure migration checklist
- Need for Migration and Understanding Migration from On-Premises to Azure Cloud
- How Azure Works and Knowing Azure Subscription PayAsYouGo AD Tenant ResourceGroup Resources and Cloud Adoption Strategy for Migration Basics
- Discovery and assessment
- Migration plan
- Migrating from onprem to Azure cloud, Azure cloud to other cloud AWS
- Migration of MySQL Database to Azure DMS
- Post Migration Checks

Azure ARM Templates

section 6

- Introduction of Azure Resource Manager Templates Concepts and Briefing of JSON & YAML Formats
- Structure & Syntax of ARM Templates and Shown Creating a Virtual Network with Two App & Web Subnets by Manual and ARM Template ways
- Creating a VNet with 3 AppWebDB Subnets & AddressPrefixes and Try Realizing n-Tier Architecture VNet with 6 Subnets & NetworkSecurityGroup using ARM Template Concepts
- Creating an Ntier architecture VNet with Multiple 6 Subnets along with NetworkSecurityGroup using dependsOn resourceId options in ARM Templates
- basic parameter, add perameter to subnet cidr ranges, object perameter for adding cidr ranges and using variable for network & subnet in ARM Template
- Extending the Ntier architecture VNet with Multiple Subnets NSG with a NIC and IP Address and fetching PublicIP & VM Details as Outputs Section of ARM Templates in Azure
- Add sql server to ARM Templates && Create sql database using existing template && Deploy ARM Templates using azure cli
- Deploy ARM Templates using Azure DevOps

Azure App service and Function App, KeyVault(Dev

section 7

- Understanding a 3 Tier Architecture Application of sma
  Exploring a Windows VM and Sample APP Service in Azu
- Explained about Azure AppService Billing Details and Create
  Application and Deploy it using AppService in Azure
- Azure App Service Environments, Azure App Service Plans, Deploying
  Configuring Database to web apps, Introduction to Serverless Computing
- Azure Functions, Pricing plan, Creating Azure Functions Using Portal and deplo
  .Net Code, Configuring Local Developer Environment With Visual Studio Code
- Introduction To Triggers & Bindings and Creating a Blob-Triggered Function
- Azure Functions triggers and bindings concepts and creates a copy of the blob using Azure Blob Trigger
- Azure KeyVault Overview

Azure Networking

section 8

- Networking Basic: Ip Address, Networkid, Hostid and example for subnet mask
- (Networking Fundamentals, Types of class in networking, Classful networks, Classless networks and subnet mask
- Cidr Notation, Cidr Ranges, Creating Private Networking with 4 subnets
- Basic Networking : network, subnets, router, routetable, Traceroute
- Underestanding Regions in AZure, Creating Virtual Network (Vnet),subnets ,Creating virtual machine with public ip and without public network, connect to public vm to private vm
- Network Security Groups, Network Security Rules, Inbound rules and Outbound rules with: [Priority, Source, destination, Protocol,Port, Action
- Network Security Groups (NSG) and Creating Subnet Level network security group rules
- Installation Azure cli and Powershell,cli and Powershell Extension in vscode, Creating Network in Azure using Azure-cli Commands
- Creating Virtual Network With 6 subnets,Creating NSG Rules for Allow and Denies by Using Azure cli/Portal
- Understanding the VNet’s Peering Concepts
- Understanding Virtual Private Network,Types of VPN,Point to Site & Site to Site, Virtual Network Gateway, Local Gateway,Configuring Customer Local Gateway to Virtual Network Gateway

Azure Load balancing Strategies

section 9

- Creating Image from Lamp server & Creat Two vm using image with out public ip and Create a public load balancer to load balance VMs
- Backend Resources,Backend pool, layer4 Load Balancer, Public & Internal Load Balancer and Health probes to monitor Load Balancer Resources
- Azure Application Gateway (Layer7 Load Balancing
- App Gateway path Based Routing, Azure DNS and Creating a DNS Zone for already Purchased domain and mapping public ip to domain
- Azure FrontDoor LoadBalancer
- Disater Recovery with Traffic Manger
- Azure Private Enpoints

Azure SQL DB &Cosmos DB

section 10

- Database overview
- Types of databases
- Choose right dabases for the application
- Setup and Configure Azure SQL databse
- Intrgrate Database with Realtime Application

Azure SQL DB &Cosmos DB

section 11

- Monitoring Overview
- Metrics, Logs & Alerts
- Logs Analytics Workspace
- Application Insights
- Resource Monitoring with Azure Monitor

Azure Security

section 12

- Azure Service Principal
- Managed identities
- Virtual Machine Security Best Practices
- Database Security Best Practices
- App Service Security Best Practices
- Networking Security Best Practices

Disaster Recovery & BCP with Azure Cloud

section 13

- Disaster Recovery & BCP Overview
- RPO & RTO
- Setup and Configure Disaster recovery strategy with Azure Cloud
- Test DR with Realtime Application

Disaster Recovery & BCP with Azure Cloud

section 14

- Disaster Recovery & BCP Overview
- RPO & RTO
- Setup and Configure Disaster recovery strategy with Azure Cloud
- Test DR with Realtime Application

section 15

- Generative AI, ML & IOT - Cloud & DevSecOps

- I. GenAI & ML Basics
- What is GenAI? - Defining Generative AI and its capabilities.
- Why it's important. - Exploring the impact and potential of GenAI.
- ML refresh. - Brief overview of key machine learning concepts.
- Cloud & DevSecOps for GenAI. - Why cloud platforms and secure development are essential.

- II. Building GenAI Apps
- App basics. - Understanding the structure of a GenAI application.
- Code setup. - Configuring a development environment.
- Example apps. - Building simple GenAI applications.

- III. GenAI & Copilot
- GenAI platforms. - Exploring cloud platforms for GenAI development.
- Model deployment. - Deploying trained models for use.
- Using Copilots. - Integrating AI assistants into workflows.

- IV. Small Language Models (SLMs)
- What are SLMs? - Defining and explaining smaller language models.
- How they work. - Simplified explanation of SLM operation.
- SLM uses. - Practical applications of SLMs.

- V. Large Language Models (LLMs) & RAG
- What are LLMs? - Understanding the power of large language models.
- RAG explained. - How Retrieval Augmented Generation works.
- Using LLMs. - Prompt engineering and fine-tuning.

- VI. Agentic AI
- What is it? - Defining agentic AI and autonomous agents.
- How it's used. - Applications of agentic AI.

- VII. Cloud for GenAI
- Azure . – Azure OpenAI services for GenAI and ML.
- AWS . - AWS Bedrock services for GenAI and ML.

- VIII. DevSecOps
- Security basics. - Fundamental security concepts for GenAI.
- DevSecOps for GenAI. - Secure development practices for GenAI.

- IX. GenAI & IoT
- IoT basics. - Introduction to the Internet of Things.
- GenAI for IoT. - Combining GenAI with IoT data.

- X. Final Project
- Build a GenAI app. - Developing a practical GenAI application.
- Use cloud & DevSecOps. - Applying cloud services and secure practices.

Academy.ITasCode.in | +91 970 5535 444

Course Duration: 75 Days

section 16

Manage IoT devices with Azure IoT Hub

- Securely communicate with devices using AWS IoT Core protocols.
- Monitor and manage devices at scale with AWS IoT Core services.
- Build and deploy applications to interact with device data in AWS IoT Core

section 17

Microsoft AzureCloud Architecture best practices

- Microsoft Azure Well-Architected Framework and use cases Choose an Azure compute service for your application Choose the right load balancing solution for business

section 18

Internship with Realtime project setup

section 19

ITasCode Team Guidance

- DevOps Roles & Responsibilities Mock Interview
- Support Resume Preparation
- Real Time Work Demonstration with Practicality.
- Q&A – Interview Discussions

Course Duration: 75 Days

IT AS CODE
Pvt Ltd.,

For more details,

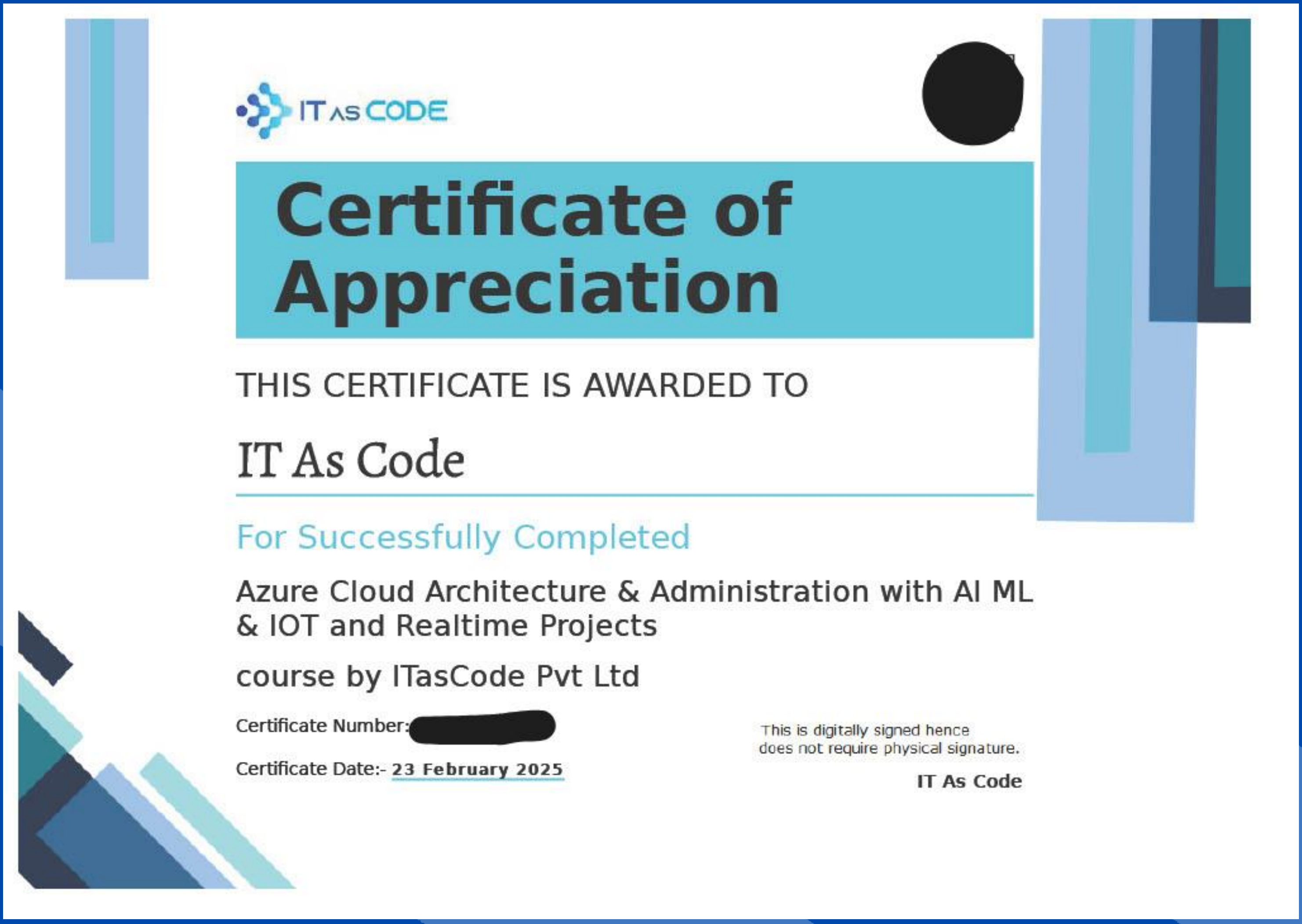
Visit: Academy.itascode.in

Email: contact@itascode.in

Google Play Store App Name – ITasCode Academy

Course Completion Certificate

- **•Successful Completion:**
  •I Students must fulfill all program requirements.
  •This includes completing assigned tasks and projects.
  •Comprehensive Performance Evaluation:
  •Students will be assessed based on their overall performance.
  •Evaluation will cover technical skills, participation, and work ethic.
  •Prestigious Course Completion Certificate:
  •A formal document will be awarded.
  •The certificate signifies successful completion and achievement.
  •The certificate adds value to the student's professional portfolio.



IT AS CODE

[redacted]

Certificate of Appreciation

THIS CERTIFICATE IS AWARDED TO

IT As Code

For Successfully Completed

Azure Cloud Architecture & Administration with AI ML & IOT and Realtime Projects

course by ITasCode Pvt Ltd

Certificate Number: [redacted]

Certificate Date:- 23 February 2025

This is digitally signed hence does not require physical signature.

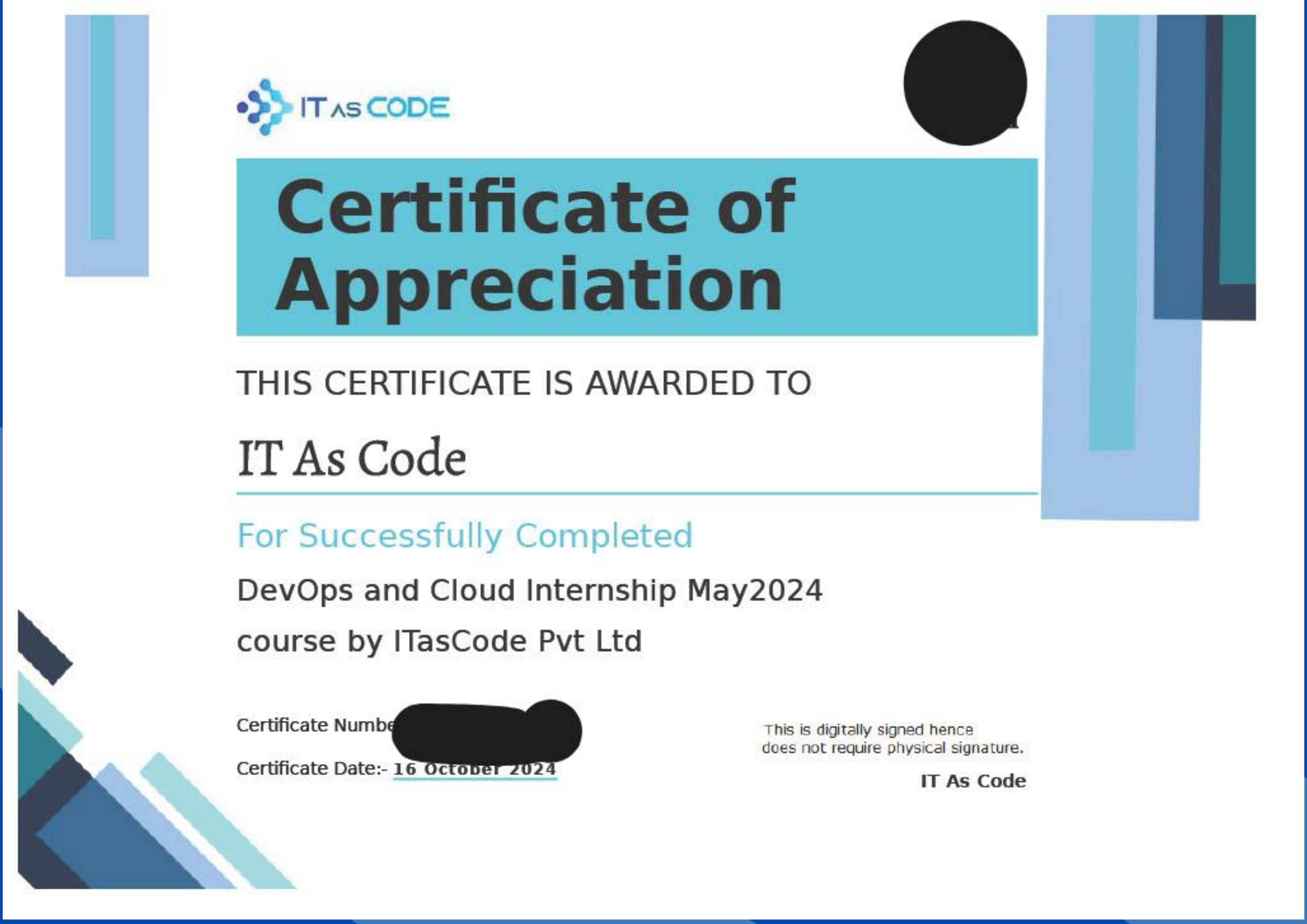
IT As Code

Academy.ITasCode.in | +91 970 5535 444

Course Duration: 120 Days

Award Internship Certificate

- **Successful Completion:**
- Interns must fulfill all program requirements.
- This includes completing assigned tasks and projects.
- **Comprehensive Performance Evaluation:**
- Interns will be assessed based on their overall performance.
- Evaluation will cover technical skills, participation, and work ethic.
- **Prestigious Internship Certificate:**
- A formal document will be awarded.
- The certificate signifies successful completion and achievement.
- The certificate adds value to the intern's professional portfolio.



IT AS CODE

[redacted]

Certificate of Appreciation

THIS CERTIFICATE IS AWARDED TO

IT As Code

For Successfully Completed

DevOps and Cloud Internship May2024

course by ITasCode Pvt Ltd

Certificate Numbe[redacted]

Certificate Date:- 16 October 2024

This is digitally signed hence
does not require physical signature.

IT As Code

Academy.ITasCode.in | +91 970 5535 444

Course Duration: 120 Days