NIRMA UNIVERSITY

Institute:	Institute of Technology, School of Technology
Name of Programme:	MTech (Cyber Security), MTech (Data Science)
Course Code:	6CS467ME25
Course Title:	Microservice Architecture and Programming
Course Type:	Department Elective-III
Year of Introduction:	2025-26

L	T	Practical Component				C
		LPW	PW	W	S	
3	-	2	-	-	-	4

Course Learning Outcomes (CLOs):

At the end of the course, the student will be able to –

1. infer the key advantages and complexities present in microservice architectures (BL2)
2. apply the appropriate architectural approach for the design of microservices (BL3)
3. choose suitable techniques and technologies to develop microservice applications effectively (BL5)
4. test the deployment of microservice applications on cloud platforms. (BL6)

Unit	Contents	Teaching Hours (Total 45)
Unit-I	**Introduction to Microservices:** Monolithic architecture, Web Services and Service Oriented Architecture, SOA and Microservice architecture	03
Unit-II	**Microservice Architecture Concepts:** Microservice software architecture: patterns and techniques, Overall topology and core architecture components, Architectural characteristics, Service components and granularity, Bounded context, Data domains, API Ecosystem for Microservice, API layer design and implementation alternatives, API Gateway, Service discovery and registration, best practices of microservice architecture	08
Unit-III	**Messaging Middleware:** IPC in microservice architecture, Synchronous and asynchronous messaging patterns, REST and gRPC based messaging, Service bus for commands and events, Message queuing systems, Message broker, JMS, Rabbit MQ and Kafka, Message driven micro service application	10
Unit-IV	**Managing Databases for Microservices:** Distributed databases, NoSQL based systems, CAP and BASE consistency models for microservices, CRUD operations, Shared databases and Database per microservice pattern, Scaling and replicating databases	06
Unit-V	**Transactions and Data Streaming in Microservices:** Managing transactions with Sagas: choreographed, orchestrated, Event sourcing and CQRS Pattern, CDC with Transactional outbox pattern, Transaction log tailing, Streaming data in microservices, Streaming SQL, Data streaming approaches with Apache Spark and Kafka	08

Unit-VI	**Hybrid Architectures and Deployment:** Event-driven architecture for microservices, Architectural modularity, Serverless microservices architecture pattern, Caching, Load balancing, Circuit Breaker, Deployment patterns and strategies with containers, Virtual machines and clusters, Container Orchestration Approaches, Microservices deployment on Public Cloud platforms, Microservices Testing, Health check and observability, Securing Microservices.	10

Suggested List of Experiments:

Sr. No.	Name of Experiments/Exercises	Hours
1	To understand the fundamentals of Git by performing basic operations such as initializing repositories, committing changes, branching, merging, resolving conflicts, etc.	2
2	Experiment with Docker by learning how to create, manage, and deploy containers. This includes tasks such as building Docker images and running containers	2
3	Learn to design and implement RESTful APIs using Java Spring Boot. It includes setting up a Spring Boot project, defining REST endpoints, handling HTTP requests and responses, and integrating with a database.	4
4	To implement the microservices using gRPC and Python. This experiment involves defining service contracts with Protocol Buffers, implementing gRPC services and clients, and understanding the benefits of using gRPC for inter-service communication	4
5	Develop a microservice application that utilizes message queuing systems for asynchronous communication. This includes experimenting with open-source solutions like RabbitMQ or Kafka, as well as AWS services, such as SQS/SNS, to implement reliable message queuing	4
6	To develop to orchestrate AWS Lambda functions using AWS Step Functions. This involves creating and managing state machines, defining workflows, handling errors, and integrating with other AWS services to build complex serverless applications	4
7	Create a multi-microservice application with both synchronous and asynchronous communication mechanisms. Also. Cover internal service communication, using an API Gateway for external access, and ensuring efficient and reliable interactions between services	4
8	Set up a continuous integration and continuous deployment (CI/CD) pipeline using Jenkins, Gitops and other open-source tools. This includes automating code builds, running tests, and deploying applications to various environments, ensuring a smooth and efficient development workflow	2
9	Implement resilience and security patterns in microservices, focusing on the circuit breaker pattern. This practical involves configuring circuit breakers to handle service failures gracefully, ensuring robustness, and enhancing the reliability of your microservices architecture	2
10	To design and develop the scalable microservices using container orchestration tools like Kubernetes and Docker Swarm. It includes deploying microservices in a cluster, managing scaling, load balancing, and ensuring high availability and fault tolerance in your applications.	2