NIRMA UNIVERSITY

Institute:	**Institute of Technology, School of Technology**
Name of Programme:	**BTech CSE**
Course Code:	4CS502ME25
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	0	2	-	-	-	4

Course Learning Outcomes:

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

1. recognise the key advantages and complexities present in microservice architectures (BL1)
2. apply the appropriate architectural approach for the design of microservices (BL2)
3. implement microservice applications effectively with suitable techniques and technologies (BL3)
4. design and deploy microservice applications on cloud platforms. (BL4)

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, Service discovery and registration, best practices of microservice architecture	06
Unit-III	**Communication and Messaging in Microservices:** IPC in a microservice architecture, Synchronous and asynchronous messaging patterns, API layer design and implementation alternatives, API Gateway, REST, GraphQL and gRPC-based communication, Message and Payload Formats, Service bus for commands and events, Message queuing systems, Message broker, JMS, Rabbit MQ, and Kafka, Message driven microservice application	12
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 Flink 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, Healthcheck and observability, Securing Microservices	10

Self-Study:

The self-study contents will be declared at the commencement of the semester. Around 10% of the questions will be asked from self-study contents

Suggested Readings/ References:

1. Chris Richardson, *Microservices Patterns With examples in Java*, Manning publication
2. Binildas C, *Practical Microservices Architectural Patterns*, Apress
3. Sam Newman, *Building Microservices: Designing fine-grained systems*, O'Reilly
4. Sam Newman, *Monolith to Microservices*, O'Reilly
5. Irakli Nadareishvili, Ronnie Mitra, Matt McLarty, Mike Amundsen, *Microservice Architecture: Aligning Principles, Practices, and Culture*, Shroff/O'Reilly
6. Susan J. Fowler, *Microservices in Production*, O'Reilly
7. Morgan Bruce, Paulo A. Pereira, *Microservices in Action*, Manning publication
8. Bill Wagner, *Mike Rousos, .NET Microservices: Architecture for containerized .Net applications*, Microsoft Corporation.

Suggested List of Experiments:

Sr. No.	Name of Experiments/Exercises	Hours
1	Git - understanding its fundamentals with basic operations and commands for version control management	02
2	Experimenting with basic commands of the Docker Platform	02
3	Design and implement REST API-based Application	04
4	Design and implement gRPC-based micro-service application	04
5	Implement Message queuing system-based micro-service application	02
6	Developing transaction orchestration-based micro-service application	04
7	API Gateway integration with a group of 5 micro-services with an internal synchronous and asynchronous communication mechanism	04
8	Configure a continuous integration and continuous deployment (CI/CD) pipeline	04
9	Implement resilience and security mechanisms in microservices, focusing on the circuit breaker pattern.	02
10	Design and develop scalable microservices using container orchestration tools like Kubernetes and Docker Swarm.	02