Minor Stream in Cloud Administration in collaboration with Google Cloud

Course Code	Course Name	Level	L	T	P	C	Total
2501CS81	Cloud Computing Foundations	FC	2		2	4	6
2501CS82	Cloud Data Analytics	FC	2		2	4	6
2501CS83	Cloud Cybersecurity	FC	2		2	4	6
2501CS84	Cloud GenAI	IC	2		1	3	4
2501CS25	Cloud IOT &Edge ML	AC	2		1	3	4
2501CS85	Cloud Digital Leader	AC	1		0	1	1
2501CS86	Cloud Engineering	AC	2		2	4	6
2501CS26	Block Chain Technologies	AC	2		1	3	4
2501AI24	API and Micro Services	IC	2		1	3	4
2501CS87	Devops Essentials	AC	2		1	3	4
Total			19		13	32	45

S.No	Course Code	Course Name	Web Link
1	2501CS81	Cloud Computing Foundations	Google Certification
2	2501CS82	Cloud Data Analytics	Google Certification
3	2501CS83	Cloud Cybersecurity	Google Certification
4	2501CS84	Cloud GenAI	Google Certification
5	2501CS85	Cloud Digital Leader	Google Certification
6	2501CS86	Cloud Engineering	Google Certification

Cloud IoT & Edge ML

(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501CS25	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the course, student will be able to:

- **CO1:** Explain technologies behind the communication and management of fog and edge resources.
- **CO2:** Discuss techniques for storage and computation in fog, edge, 5G and cloud
- **CO3:** Implement Internet of Everything (IoE) applications through fog computing architecture.
- **CO4:** Illustrate optimization strategies in fog and Edge Architectures.
- **CO5:** Summarize applications in Fog and Edge Computing.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	1	2	-	-	2	-	-	2	2	-	2
CO2	2	2	-	-	2	-	-	2	2	-	2
CO3	1	2	3	-	2	-	-	2	2	-	2
CO4	2	1	1	3	2	-	-	2	2	-	2
CO5	2	1	1	-	2	-	-	2	2	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	1	-
CO2	1	-
CO3	1	-
CO4	-	2
CO5	-	2

UNIT – I

Internet of Things (IoT) and New Computing Paradigms: Introduction, Relevant Technologies, Fog and Edge Computing Completing the Cloud, Hierarchy of Fog and Edge Computing, Business Models, Edge Computing Platforms, Opportunities and Challenges.

Practice:

1. Building a Simple Edge Computing Application with Data Processing and Cloud Integration.

UNIT – II

Challenges in Federating Edge Resources: Introduction, Methodology, Integrated C2F2T

Literature by Modeling Technique, Integrated C2F2T Literature by Use Case Scenarios, Integrated C2F2T Literature by Metrics, Threads, Standards.

Practice:

1. Setting Up a Simple IoT Device with Edge Computing

UNIT – III

Orchestration of Network Slices in Fog, Edge, and Clouds: Introduction, Background, Network Slicing, Network Slicing in Software, Defined Clouds, Network Slicing Management in Edge and Fog, Internet of Vehicles (IoV): Architecture, Protocols and Seven-layer security model architecture for Internet of Vehicles, IoV: Network Models, Challenges and future aspects.

Practice:

1. Implementing a Simple Network Slicing Scenario for Internet of Vehicles (IoV) Using Edge and Fog Computing.

UNIT – IV

Optimization Problems in Fog and Edge Computing: Preliminaries, The Case for Optimization in Fog Computing, Formal Modeling Framework for Fog Computing, Metrics ,Further Quality Attributes ,Optimization Opportunities along the Fog Architecture, Optimization Opportunities along the Service Life Cycle , Toward a Taxonomy of Optimization Problems in Fog Computing.

Practice:

1. Optimizing Task Scheduling in Fog Computing

UNIT – V

Applications of Fog and Edge Computing: Exploiting Fog Computing in Health Monitoring-Smart Surveillance Video Stream Processing at the Edge for Real - Time Human Objects Tracking-Fog Computing Model for Evolving Smart Transportation Applications - Testing Perspectives of Fog - Based IoT Applications - Legal Aspects of Operating IoT Applications in the Fog.

Practice:

1. Real-Time Human Object Tracking Using Fog Computing in a Smart Surveillance System

Text Books:

1 Fog and Edge computing: Principles and Paradigms ,Buyya, Rajkumar, and Satish Narayana Srirama, John Wiley & Sons, USA, 1st edition, 2019, ISBN: 978-1-119-52498-4.

2 Internet of Things –From Research and Innovation to Market Deployment, OvidiuVermesan, Peter Friess, "River Publishers, India, 1st edition, ISBN: 978-8793102941

Reference Books:

1 Cloud computing: A hands-on approach, Bahga, Arshdeep, and Vijay Madisetti , CreateSpace Independent Publishing Platform, USA, 2nd edition, ISBN: 978-1494435141

Web Links:

1 https://onlinecourses.nptel.ac.in/noc24_cs66/preview

Block Chain Technologies
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501CS26	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Demonstrate blockchain and crypto currency basics
CO2: Compare and contrast permissioned and permissionless blockchains
CO3: Explain different types of crypto currency wallets
CO4: Explain how to compile and deploy smart contracts using Ethereum
CO5: Illustrate Hyperledger fabric and use cases of block chain technologies

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO11
CO1	3	1	-	-	-	-	-	-	-	-	2
CO2	2	3	1	2	-	-	-	-	-	-	2
CO3	2	3	1	1	-	-	-	-	-	-	2
CO4	2	2	3	2	-	-	-	-	-	-	2
CO5	2	2	3	2	-	-	-	-	-	-	2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	-	2
CO2	-	2
CO3	-	2
CO4	-	3
CO5	-	3

UNIT – I

Introduction: Overview of Block chain, History of Blockchain, Real-world applications, Peer to Peer Network, Smart Contract, Wallet, Digital Currency, Ledgers, Types of Blockchain Platforms.

Practice:

1. Creating Merkle tree

UNIT – II

Consensus Mechanism: Permissioned Blockchain, Permisionless Blockchain, Different Consensus Mechanisms- Proof of Work, Proof of Stake, Proof of Activity, Proof of Burn, Proof of Elapsed Time, Proof of Authority, Proof of Importance, voting-based consensus algorithms, and federated consensus.

Practice:

1. Creation of Block.

UNIT – III

Crypto currency and Wallet: Types of Wallet, Desktop Wallet, App based Wallet, Browser based wallet, Metamask, Creating an account in Metamask, Use of faucet to fund wallet, transfer of cryptocurrency in meta mask, Merkle tree, hard and soft forks, network models and properties of secure state machine replication (SMR)

Practice:

1. Block chain Implementation Programming code

UNIT – IV

Smart contract and Ethereum: Overview of Ethereum, Writing Smart Contract in Solidity, Remix IDE , Different networks of ethereum, understanding blocks practically at blockhcain.com, how to compile and deploy smart contract in remix

Practice:

1. Java code to implement blockchain in Merkle Trees

UNIT – V

Understanding Hyperledger Fabric: Hyperledger Fabric- Architecture, Identities and Policies, Membership and Access Control, Channels, Transaction Validation, Writing smart contracts using Hyperledger Fabric.
Use Cases: Cross border payments, Know Your Customer (KYC), Food Security, Block chain enabled Trade, Anti-Money Laundering (AML)

Practice:

1. Creating a Crypto-currency Wallet

Text Books:

1 Blockchain: Blueprint for a New Economy, Melanie Swan, O'Reilly, ISBN:9781491920473
2 Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks by Imran Bashier, Packt publishing, ISBN: 9781787125445

Reference Books:

1 Mastering Ethereum: Building Smart Contracts and DApps by Andrews, ISBN:978-1491971949
2 Mastering Bitcoin: Programming the Open Blockchain, by Andreas M. Antonpoulos, O'Reilly, ISBN: 9781491954362

Web Links:

1 https://onlinecourses.swayam2.ac.in/aic21_ge01/
2 https://github.com/blockchainedindia/resources
3 https://github.com/HyperledgerHandsOn/trade-finance-logistics

API & Micro Services
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501AI24	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Apply Dependency Injection principles using Constructor and Setter Injection.
CO2: Analyze Spring Boot fundamentals and Spring AOP
CO3: Apply transaction management and custom repository implementation
CO4: Explain web services using SOAP and REST
CO5: Evaluate the importance of exception handling and data validation in RESTful services

Mapping of Course Outcomes with Program Outcomes:

CO/ PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	2	3	-	1	-	-	1	1	-	-
CO2	3	2	2	-	1	-	-	1	1	-	-
CO3	2	2	2	-	1	-	-	1	1	-	-
CO4	2	2	2	-	1	-	-	1	1	-	-
CO5	2	2	3	-	1	-	-	1	1	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/ PSO	PSO1	PSO2
CO1	2	-
CO2	1	-
CO3	2	-
CO4	1	-
CO5	1	-

UNIT – I

Spring 5 Basics: Why Spring, What is Spring Framework, Spring Framework - Modules, Configuring IoC container using Java-based configuration, Introduction To Dependency Injection, Constructor Injection, Setter Injection, What is Auto Scanning

Practice:

1. Setting up Spring Framework
   a. Create a new Spring project using Spring Initializr.
   b. Configure the project structure and dependencies.
   c. Write a simple "Hello World" application using Spring
2. Spring Modules
   a. Explore Spring Framework modules (e.g., Core, Data Access, Web).

b. Implement a simple data access application using Spring JDBC.
c. Create a web application using Spring MVC.

UNIT – II

Spring Boot: Creating a Spring Boot Application, Spring Boot Application Annotation,What is Auto wiring, Scope of a bean, Logger, Introduction to Spring AOP, Implementing AOP advices, Best Practices: Spring Boot Application

Practice:

1. Creating a Spring Boot Application with Auto Wiring and Logging
   a. Create a new Spring Boot project using Spring Initializr.
   b. Define two classes: Employee and EmployeeService.
   c. Annotate EmployeeService with @Service and Employee with @Component.
   d. Use @Autowired to inject Employee into EmployeeService.
   e. Configure logging using Logback.
   f. Write a test class to verify auto-wiring and logging.
2. Implementing AOP Advices in Spring Boot
   a. Create a new Spring Boot project using Spring Initializr.
   b. Define two classes: Student and StudentService.
   c. Annotate StudentService with @Service and Student with @Component.
   d. Implement AOP advice using @Aspect and @Before.
   e. Apply AOP advice to StudentService methods.
   f. Write a test class to verify AOP functionality.

UNIT – III

Spring Data JPA with Boot: Limitations of JDBC API, Why Spring Data JPA, Spring Data JPA with Spring Boot, Spring Data JPA Configuration, Pagination and Sorting, Query Approaches, Named Queries and Query, Why Spring Transaction, Spring Declarative Transaction, Update Operation in Spring Data JPA, Custom Repository Implementation, Best Practices-Spring Data JPA

Practice:

1. Setting up Spring Data JPA with Spring Boot
   a. Create a new Spring Boot project.
   b. Add Spring Data JPA dependency.
   c. Configure database connection properties.
   d. Create an entity class (e.g., User).
   e. Create a repository interface (e.g., UserRepository).
   f. Test CRUD operations
2. Pagination and Sorting with Spring Data JPA
   a. Create a repository interface with pagination methods.
   b. Use Pageable and Sort to implement pagination and sorting.
   c. Test pagination and sorting functionality.
   d. Implement custom sorting using @Query

UNIT – IV

Web Services: Why Web services, SOA-Service Oriented Architecture, What are Web Services, Types of Web Services, SOAP based Web Services, REST ful Web Services, How to create REST ful Services

Practice:

1. Creating a Simple RESTful Web Service
   a. Create a new Spring Boot project.
   b. Define a resource class (e.g., User).
   c. Create a REST controller (e.g., UserController).
   d. Implement CRUD operations.
   e. Test using Postman or curl.
2. SOAP-based Web Service with Spring Boot
   a. Create a new Spring Boot project.
   b. Define a service interface (e.g., UserService).
   c. Implement service using @Service.
   d. Create a WSDL file.
   e. Test using SoapUI.

UNIT – V

Spring REST: Spring REST - An Introduction, Creating a Spring REST Controller, @RequestBody and Response Entity, Parameter Injection, Usage of @PathVariable, @RequestParam and @MatrixVariable, Exception Handling, Data Validation, Creating a REST Client, Versioning a Spring REST endpoint, Enabling CORS in Spring REST, Securing Spring REST endpoints

Practice:

1. Creating a Simple Spring REST Controller
   a. Create a new Spring Boot project.
   b. Define a resource class (e.g., User).
   c. Create a REST controller (e.g., UserController).
   d. Implement CRUD operations.
   e. Test using Postman or curl.
2. Using @RequestBody and ResponseEntity
   a. Create a REST controller.
   b. Use @RequestBody to accept JSON data.
   c. Use ResponseEntity to return HTTP responses.
   d. Test using Postman or curl.

Note: The student must Complete & Submit Spring5 Basics with Spring Boot, Spring DataJPA with Boot, SpringREST Certificate Course offered by Infosys Spring Board at the end of the Practice Session.

Text Books:

1 Spring in action, 5th Edition, Author: Craig Walls, Ryan Breidenbach, Manning, ISBN: 978-1617294945

Web Reference:

1 Dependency Injection in spring-javat point
2 Auto wiring in Spring- javat point
3 https://docs.spring.io/spring-boot/docs/2.0.x/reference/html/using-boot-using-spring boot application-annotation.html
4 Auto wiring in Spring – javat point,https://www.baeldung.com/spring-bean-scopes
5 Spring Boot Logging|How does logging works inspring boot with example (educba.com)
6 Spring AOP Tutorial|Aspect Oriented Programming-javat point
7 Spring Boot Best Practices(javaguides.net)
8 Introduction to Spring Data JPA | SpringHow
9 https://asbnotebook.com/spring-data-jpa-crud-example/ ,https://www.bezkoder.com/spring-boot-jpa-crud-rest-api/
10 Pagination and Sorting using Spring Data JPA – Paging And Sorting Repository (javaguides.net)
11 https://www.javaguides.net/2018/11/spring-data-jpa-query-creation-from-method-names.html,https://www.javaguides.net/2022/02/spring-data-jpa-namedqueries-example.html
12 https://javadeveloperzone.com/spring/spring-declarative-transaction-management/
13 https://javadeveloperzone.com/spring/spring-declarative-transaction-management/
14 https://javabeat.net/spring-data-custom-repository/
15 https://www.jrebel.com/blog/jpa-application-performance-best-practices
16 https://www.javatpoint.com/service-oriented-architecture,https://www.javatpoint.com/web-services-tutorial
17 https://www.javatpoint.com/soap-web-services,https://www.javatpoint.com/restful-web-services
18 RESTful Web Services - javatpoint
19 https://www.javatpoint.com/restful-web-services-spring-boot
20 https://www.javatpoint.com/restful-web-services-spring-boothttps://dzone.com/articles/lifecycle-of-a-request-response-process-for-a-spri
21 https://www.ibm.com/docs/en/was/8.5.5?topic=applications-defining-uri-patterns-resources-in-restful
22 https://www.baeldung.com/exception-handling-for-rest-with-spring
23 https://howtodoinjava.com/spring-boot2/resttemplate/spring-restful-client-resttemplate-example/
24 https://www.javatpoint.com/restful-web-services-versioning

Web Links:

1 https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01296689056211763272_shared/ overview[Spring5BasicswithSpringBoot]
2 https://infyspringboard.onwingspan.com/en/app/toc/lex_4313461831752789500_shared/overvi ew[Spring DataJPAwithBoot]
3 https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01273190096390553619 0_shared/overview[SpringREST]