



# MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

## COURSECONTENT

WEB SERVICES								
B.Tech. IV Year II Sem.								
CourseCode	Category	Hours/Week			Credits	Maximum Marks		
24X0552	Foundation	L	T	P	C	CIA	SEE	Total
		3	0	0	3	40	60	100
ContactClasses:45	TutorialClasses:Nil	PracticalClasses:Nil			TotalClasses:45			
Prerequisites:Object Oriented Programming, Computer Networks, Database Management Systems, XML								

### CourseOverview:

This course introduces the concepts, standards, and technologies used to build **web services**—software systems designed to support interoperable machine-to-machine interaction over a network. Students learn how distributed systems communicate using web protocols and how to design, implement, secure, and consume web services.

### CourseObjectives:

- To understand the concepts of web services as a means of distributed computing.
- To learn about the technologies involved in the creation and consumption of web services, specifically, SOAP and REST.
- To acquire knowledge of web service description, discovery, and security mechanisms.
- To gain practical skills in developing and integrating web services in various applications.

### CourseOutcomes:AfterCompletionoftheCourse,Studentsshouldbeableto

**CO1:** Explain the evolution of distributed computing and the fundamental concepts of Service-Oriented Architecture (SOA) and Web Services.

**CO2:** Apply XML and related technologies for data representation and exchange in web service communication.

**CO3:** Design and implement web services using the SOAP protocol and WSDL for service description.

**CO4:** Develop and consume RESTful web services following REST architectural principles.

**CO5:** Analyze and integrate mechanisms for web service discovery and security, along with understanding advanced web service concepts.

**UNIT - I:** Fundamentals of Web Services and SOA Evolution of Distributed Computing: Client/Server, CORBA, JAVA RMI, Microsoft DCOM, MOM. Challenges in Distributed Computing. Role of J2EE and XML in distributed computing. Emergence of Web Services and Service-Oriented Architecture (SOA). Definition of Web Services, Basic operational model of web services. Characteristics and benefits of web services. Web Services Architecture and its core building blocks. Service Providers, Service Consumers, Service Registry. Web Services Communication models.

**UNIT - II:** XML and SOAP (Simple Object Access Protocol) Part A: XML and Related Technologies: XML fundamentals: XML document structure, elements, attributes, XML declaration. Well-formed vs. Valid XML. Document Type Definition (DTD): internal and external DTDs. XML Schema: complex types, simple types, elements, attributes. XML Namespaces. XML Parsers: SAX (Simple API for XML), DOM (Document Object Model).

XPath for navigating XML documents. XSLT for transforming XML documents. Introduction to JSON for data interchange. Part B: SOAP Protocol: Introduction to SOAP: Messaging framework, structure of a SOAP message (Envelope, Header, Body, Fault). SOAP with HTTP binding. SOAP communication model. SOAP faults and error handling. Implementing SOAP-based web services using Java (e.g., JAX-WS, Apache Axis). Consuming SOAP web services from client applications.

**UNIT - III:** WSDL (Web Services Description Language) & UDDI WSDL in the context of Web Services. Anatomy of a WSDL definition document: Types, Message, PortType, Binding, Service. WSDL bindings to SOAP. WSDL Tools. Web Services life cycle. Introduction to UDDI (Universal Description, Discovery, and Integration): Role of service discovery in SOA. UDDI registries, uses of UDDI Registry. UDDI data structures, Publishing and searching information in a UDDI Registry. Limitations of UDDI.

**UNIT - IV:** RESTful Web Services Introduction to REST (Representational State Transfer) architectural style. Principles of REST: Statelessness, client-server, cacheable, uniform interface, layered system, code-on-demand (optional). Resources and their identification (URIs). HTTP methods for RESTful services (GET, POST, PUT, DELETE, PATCH). Mediatypes (JSON, XML) for representation. Building RESTful services (e.g., using popular frameworks like Node.js with Express, Spring Boot, Python with Flask/Django REST Framework). Consuming RESTful web services from various client applications.

**UNIT - V:** Web Service Security and Advanced Topics Web Service Security Threats. Security mechanisms for Web Services: WS-Security specification (XML Signature, XML Encryption). Transport-level security (SSL/TLS). Authentication and Authorization in Web Services. Overview of Microservices Architecture and its relationship to Web Services. Introduction to API Gateways. Practical considerations in web service development and deployment.

### **TEXT BOOKS:**

- Developing Java Web Services, R. Nagappan, R. Skoczylas, R.P. Sriganesh, Wiley.
- Web Services & SOA Principles and Technology, Michael P. Papazoglou, Pearson Education.
- RESTful Web Services, Leonard Richardson and Sam Ruby, O'Reilly Media.

### **REFERENCE BOOKS:**

- Java Web Services Architecture, James McGovern, Sameer Tyagi et al., Elsevier.
- Building Web Services with Java, 2nd Edition, S. Graham and others, Pearson Education.
- Web Services: Concepts, Architectures and Programming, Sandeep Chatterjee, James Webber, Pearson.

### **ELECTRONICRESOURCES:**

1. <https://www.geeksforgeeks.org/computer-science-fundamentals/what-are-web-services>
2. <https://www.ibm.com/docs/en/was/9.0.5?topic=services-web><https://www.w3schools.com/webservices/>
3. <https://www.techtarget.com/searchapparchitecture/definition/Web-services>

### **MATERIALSONLINE:**

1. Coursetemplate
2. Tutorialquestionbank
3. TechtalkandConceptVideotopics

4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper-I
8. Model question paper-II
9. Lecture notes
10. E-Learning Readiness Videos (ELRV)