

Branch Name:	MCA
Program Code:	CS201
Course Title:	Full Stack Development-III
Course Code:	4CS2010311T
Pre-requisite Course:	Knowledge of HTML, CSS, JavaScript, React fundamentals, advanced React concepts, Node.js and Express basics, RESTful API development, database integration (MySQL/MongoDB), and full-stack application development.

Course Objective:

The objectives of the course are to:

1. Provide in-depth knowledge of advanced backend architecture and scalable system design using Node.js.
2. Introduce microservices architecture and real-time application development techniques.
3. Develop understanding of cloud computing, deployment strategies, and DevOps practices.
4. Enable students to implement performance optimization, caching, and scalability techniques.
5. Familiarize students with testing, monitoring, and debugging of full-stack applications.
6. Explore emerging technologies and modern trends in full-stack development.

Teaching and Examination Scheme:

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
Lecture	Tutorial	Practical	Credit	Theory (Marks)		Practical (Marks)		Total (Marks)
				University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	
3	-	-	3	60	40	-	-	100

Course Contents:

Unit No	Topic	Total Hours	Weightage (%)
1	Advanced Backend Architecture and API Design • Node.js Architecture & Performance: Deep understanding of Node.js internals including event loop phases, asynchronous execution model, clustering, and performance optimization techniques for scalable applications. • Advanced Express.js Design Patterns: Implementation of scalable backend architecture using layered structure, modular routing, service-controller pattern, and reusable middleware design. • API Design & Versioning: Designing robust REST APIs with proper versioning strategies, request validation, standardized response formats, and API documentation practices (Swagger basics). • Advanced Authentication & Authorization: Implementation of secure authentication mechanisms using JWT refresh tokens, role-based and permission-based access control, and session management strategies. • Backend Security & Best Practices: Advanced security techniques including input sanitization, rate limiting, CORS configuration, environment-based configuration, and protection against common vulnerabilities.	10	25
2	Microservices, Real-Time Systems and Performance Optimization	10	25

	● Microservices Architecture: Introduction to microservices-based application design, service decomposition, communication between services (REST/Message Queues), and advantages over monolithic architecture. ● API Gateway & Service Communication: Role of API Gateway, request routing, load balancing, and inter-service communication using REST and event-driven approaches. ● Real-Time Application Development: Implementation of real-time systems using WebSockets and Socket.io for features like live chat and notifications. ● Caching & Performance Optimization: Use of caching mechanisms (Redis basics), in-memory caching, and strategies to improve backend performance and reduce latency. ● Scalability & Load Handling: Techniques for handling high traffic including horizontal scaling, clustering, and load balancing concepts.		
3	Cloud Deployment and DevOps Practices ● Cloud Computing Fundamentals: Introduction to cloud computing concepts, service models (IaaS, PaaS, SaaS), and deployment models for modern web applications. ● Application Deployment: Deployment of full-stack applications on cloud platforms such as AWS, Vercel, and Netlify, including build and production configuration. ● Containerization: Basics of containerization using Docker, creating Docker images, and managing application environments for consistent deployment. ● CI/CD Pipeline: Introduction to Continuous Integration and Continuous Deployment, automated build, testing, and deployment using tools like GitHub Actions. ● Version Control & Collaboration: Advanced Git workflows, branching strategies, and collaborative development using GitHub.	10	25
4	Testing, Monitoring and Emerging Technologies ● Application Testing: Introduction to testing strategies including unit testing, integration testing, and API testing using tools like Jest and Postman. ● Error Handling & Logging: Advanced error handling techniques, centralized error management, and logging using tools like Winston and Morgan. ● Monitoring & Performance Analysis: Application monitoring concepts, performance tracking, and basic use of monitoring tools for analyzing system health. ● Debugging & Optimization: Debugging backend applications, identifying performance bottlenecks, and optimizing application performance. ● Emerging Trends in Full Stack Development: Overview of modern trends such as serverless architecture, GraphQL APIs, and Progressive Web Applications (PWA).	9	25

Text Books:

1. Vasan Subramanian, "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node", 2nd Edition, Apress.
2. Ethan Brown, "Web Development with Node and Express", 2nd Edition, O'Reilly Media.
3. Mario Casciaro and Luciano Mammino, "Node.js Design Patterns", 3rd Edition, Packt Publishing.
4. Frank Zammetti, "Modern Full-Stack Development", 1st Edition, Apress.

Reference Books:

1. David Herron, "Node.js Web Development", 5th Edition, Packt Publishing.
2. Sam Newman, "Building Microservices", 2nd Edition, O'Reilly Media.
3. Kelsey Hightower et al., "Kubernetes: Up and Running", 2nd Edition, O'Reilly Media.
4. Kyle Simpson, "You Don't Know JS Yet", 2nd Edition, O'Reilly Media.
5. Marijn Haverbeke, "Eloquent JavaScript: A Modern Introduction to Programming", 3rd Edition, No Starch Press.

List of Open Source Software/learning website:

- <https://nodejs.org>
- <https://expressjs.com>
- <https://react.dev>
- <https://www.mongodb.com/docs>
- <https://developer.mozilla.org>
- <https://www.docker.com>
- <https://kubernetes.io>
- <https://aws.amazon.com>
- <https://www.w3schools.com>
- <https://github.com/explore>

Course Learning Outcomes (CLO): On completion of this course, the students will be able to:

CLO	Description	Bloom's Taxonomy Level
CLO1	Understand advanced backend architecture, Node.js internals, and API design concepts.	1.Remembering 2.Understanding
CLO2	Design scalable backend systems using microservices architecture and real-time communication techniques.	3 Applying, 4 Analyzing
CLO3	Implement performance optimization, caching, and scalability techniques in full-stack applications.	3 Applying, 4 Analyzing
CLO4	Deploy full-stack applications using cloud platforms, containerization, and CI/CD pipelines.	3 Applying, 6 Creating
CLO5	Apply testing, debugging, and monitoring techniques for improving application reliability and performance.	4 Analyzing, 5 Evaluating
CLO6	Explore and implement modern trends such as serverless architecture, GraphQL, and progressive web applications.	5 Evaluating, 6 Creating

Mapping of CLOs with POs & PSOs

Course Learning Outcomes	Program Outcomes (POs)												Program Specific Outcomes (PSOs)	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO11	PO12	PSO1	PSO2
CLO1	H	M	M	–	H	–	–	–	–	–	–	M	H	M
CLO2	H	H	H	M	H	–	–	M	M	–	M	M	H	H
CLO3	H	H	H	H	H	–	–	M	–	–	M	M	H	H
CLO4	H	H	H	M	H	–	–	M	–	–	H	M	H	H
CLO5	H	H	H	H	H	–	–	M	–	–	M	M	H	H
CLO6	H	H	H	M	H	M	–	H	M	L	M	H	H	H

H:High, M:Medium, L:Low