Version 08/06-06

Consolidated Academic Administration Plan for the Course

Distributed Systems (Core) – Sem.V –
Computer Engineering – 2025-2026 – Odd Semester

Dr. Amit K. Nerurkar (Subject Teacher)
Dr. Prakash Parmar (Lab Subject Teacher)
Dr. Amit K. Nerurkar (Cluster Mentor)

The academic resources available in VIT –

VMIS (ERP)	V-Refer and V-Live	VIT Library	VAC & MOOC Courses
Institute & Department Vision and Mission	Former IA question papers and solutions (prepared by faculty)	Former IA question papers solutions - hardcopy	Value Added Courses (VAC) are conducted throughout the semester & in the semester break - Enrol for the VACs
Program Educational Objectives (PEO)	MU end semester examination question papers and solutions (prepared by faculty)	MU end semester exam question paper & solutions - by faculty, hardcopy	
Program Specific Outcome (PSO)	Class notes and Digital Content for the subject (scanned / typed by faculty)	All text books, reference books, e -books mentioned in the syllabus & AAP	Online courses from NPTEL, Coursera etc. are pursued throughout the semester - Register for the course & get certified
Program Outcome (PO)	Comprehensive question bank, EQ, GQ, PPT, Class Test papers	Technical journals and magazines for reference	
Departmental Knowledge Map	Academic Administration Plan & Beyond Syllabus Activity report	VIT library has many resources e:g :- IEEE, Nimbus, xplore, EBSCO etc.	Watch former lectures captured in LMS at VIT

1.a Course Objectives (Write in detail – as per NBA guidelines)

Cognitive	What do you want students to know?	Student should gain fundamental knowledge in Distributed systems and its applications	
Affective	What do you want students to think / care about?	Student should analyse the importance of the performance of parallel and distributed systems.	
Behavioural	What do you want students to be able to do?	Student should apply appropriate algorithms for synchronization in DS.	

Advice to Students:

Attend every class!!! Missing even one class can have a substantial effect on your ability to understand the course. Be prepared to think and concentrate, in the class and outside. I will try to make the class very interactive. Participate in the class discussions. Ask questions when you don't understand something. Keep up with the class readings. Start assignments and homework early. Meet me in office hour to discuss ideas, solutions or to check if, what you understand is correct.

The v-Refer Link

Example:http://vidyalankarlive.com/vrefer/index.php/apps/files/?dir=/vRefer/FE/SEM%20I/202122/Engineering%20Mechanics/GM&fileid=586945 /

Creation of microsite or teams link

Example:- https://cs50.harvard.edu/college/2021/spring/)

Collaboration Policy:

We encourage discussion between students regarding the course material. However, no discussion of any sort is allowed with anyone on the assignment and homework for the class. If you find solution to some problems in a book or on the internet, you may use their idea for the solution; provided you acknowledge the source (name and page in the book or the website, if the idea is found on the internet). Even though you are allowed to use ideas from another source, you must write the solution in your own words. If you are unsure whether or not certain kinds of collaboration is possible, please ask the teacher.

1.b Course Outcome (CO) Statements and Module-Wise Mapping (follow NBA guideline)

CO No.	Statements	Related Module/s
CO1	Explain the key concepts, challenges, and architectural principles that underpin distributed systems.	1
CO2	Analyze the functional components, algorithms, and coordination techniques used to manage computation and communication.	2
CO3	Apply appropriate tools, algorithms, and middleware frameworks to build and manage reliable and scalable systems.	3
CO4	Evaluate the efficiency and effectiveness of resource management, consistency, replication, and scheduling strategies.	4
CO5	Design distributed solutions integrating modern technologies like containerization and distributed storage.	5
CO6	Demonstrate initiative in self-learning and adapt to emerging technologies in distributed computing.	6

1.c Course Outcome (CO) Statements and Module-Wise Mapping (follow NBA guideline)

	Mapped to Learning Outcomes
CO1	1.1, 1.2, 1.3, 3.1, 5.1, 6.1
CO2	1.1, 1.4, 2.1, 2.2, 3.1, 3.3, 3.5, 4.1
CO3	1.3, 1.4, 2.1, 2.3, 3.2, 3.4, 4.3, 5.4
CO4	2.2, 3.4, 3.5, 4.1, 4.2, 4.3, 5.1, 5.2, 5.3, 6.2, 6.3
CO5	4.4, 5.3, 5.4, 6.2, 6.3, 6.4, 6.5
CO6	1.5, 2.4, 3.6, 4.4, 5.5, 6.5

1.d	**Mapping of COs with POs (mark S: Strong, M: Moderate, W: Weak, Dash '–': not mapped)** (List of POs is available in V-refer)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
	Knowledge A	Analysis	Design	Investigation	Modern Tools	Society	Environment &sustainability	Ethics T	Teamwork C	Communication	Project Mgt	Life long learning
CO 1	S	M	-	-	-	-	-	-	-	-	-	-
CO 2	M	S	S	-	-	-	-	-	-	-	-	-
CO 3	M	S	S	-	-	-	-	-	-	-	-	-
CO 4	M	S	M	-	-	-	-	-	-	-	-	-
CO 5	S	S	-	-	-	-	-	-	-	-	-	W
CO6	M	M	-	-	-	-	-	-	-	-	-	W

1.e	**Mapping of COs with PSOs (mark S: Strong, M: Moderate, W: Weak, Dash '–': not mapped)**

	PSO 1	PSO 2	PSO 3
CO 1	S	-	-
CO 2	S	-	-
CO 3	S	-	-
CO 4	S	-	-
CO 5	M	-	-
CO6	M	-	-

1.f	**Teaching and Examination Scheme (As specified by the autonomous syllabus) for the Course**

Verticals	BSC/ESC	Program Courses	Multidisciplinary Courses	Skill Courses	HSSM	Experiential Learning	Liberal Learning
Tick suitable category		✓					

Subject Code	Subject Name	**Teaching Scheme**			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/Practical	Tutorial	Total
PCCE16T	Distributed Systems	2	-	-	2	-	-	2
PCCE16P	Distributed Systems Lab	-	2	-	-	1	-	1

Subject Code	Subject Name				Examination Scheme				
		Theory			Total (Theory)	Practical			Total (Practical)
		ISA	MSE	ESE		ISA	ESE	ORAL	
PCCE16T	Distributed Systems	15	20	40	75	25	25	--	50

Subject Code	Subject Name	MSE-1*		
		Q, No	Module wise % Distribution	Relevant to Bloom Taxonomy
PCCE16T	Distributed Systems	1	Module 1: 100%	Understand, Remember
		2	Module 2: 100%	Apply

Subject Code	Subject Name	MSE-2*		
		Q, No	Module wise % Distribution	Relevant to Bloom Taxonomy
PCCE16T	Distributed Systems	1	Module 3: 100%	Analyse, Apply
		2	Module 5: 100%	Apply

Subject Code	Subject Name	ESE#		
		Q, No	Distribution	Relevant to Bloom Taxonomy
PCCE16T	Distributed Systems	1	Attempt any one out of two (10 Marks Each) 10 Marks from Assignment 10 Marks based on assignments with slightly enhance difficulty	Understand, Apply
		2	Attempt any one out of two (10 Marks Each) 10 Marks based on assignments with slightly enhance difficulty 10 Marks from thought provoking	Apply, Apply
		3	Attempt any one out of two (10 Marks Each) 10 Marks based on assignments with slightly enhance difficulty 10 Marks from thought provoking	Analyse, Remember
		4	Attempt any one out of two (10 Marks Each) 10 Marks from Assignment	Understand, Remember

			10 Marks from thought provoking	

* **Recommended distribution: -** 30 Marks from Assignments, 40 marks based on assignments with slightly enhance difficulty /complex, 30 marks from thought provoking
 # Recommended distribution: - 30 Marks from Assignments, 40 marks based on assignments/MSE with slightly enhance difficulty /complex, 30 marks from thought provoking

1.g Faculty-Wise Distribution of all Lecture-Practical-Tutorial Hours for the Course

Divisions	Lecture (Hrs.)	Practical (Hrs.)				Tutorial (Hrs.)			
		Batch 1	Batch 2	Batch 3	Batch 4	Batch 1	Batch 2	Batch 3	Batch 4
A	AKN-2	AKN	AKN	AKN	AKN	--	--	--	--
B	AKN-2	AKN	AKN	AKN	AKN	--	--	--	--
C	AKN-2	PJP	PJP	PJP	PJP				

1.h Office Hours (Faculty will be available in office in this duration for solving students' query)

Division	Day	Time (at least 1 Hr. / Division)	Venue (Office Room No.)
A	Monday	4.30pm	M 209
B	Friday	4.30pm	M 209
C	Tuesday	4pm	M 209

2.a Syllabus: Module Wise Teaching Hours and % Weightage in autonomous syllabus Question Paper

Module No.	Module Title and Brief Details	Teaching Hrs. for each module	% Weightage in autonomous syllabus Question Papers			Performance Indicator Mapping
			ISA	MSE	ESE	
1	**Foundations of Distributed Systems** Characteristics, Goals, and Issues in Distributed Systems Distributed Computing Models: Mini Computer, Workstation, Workstation Server, Processor Pool, Hybrid. Hardware and Software Concepts of Distributed Systems Middleware and APIs: **Self-Learning Topics:** REST, gRPC	4	10%	50%	10%	
Learning Outcome	1. A learner will be able to: 2. Describe the fundamental characteristics, goals, and issues of distributed systems. (P.I.: 1.4.1, 2.1.2) (CO1, CO2) 3. Identify and distinguish various distributed computing models such as minicomputer, workstation, processor pool, and hybrid models. (P.I.: 2.2.1, 2.2.2) (CO1) 4. Explain the role of hardware and software components in the architecture of distributed systems. (P.I.: 2.2.2, 1.4.1) (CO1, CO3)					

	5. Apply knowledge of middleware and APIs (e.g., REST, gRPC) in enabling communication and coordination within distributed environments. (P.I.: 5.1.1, 1.4.1) (CO2, CO3) 6. Demonstrate awareness of emerging middleware technologies and pursue self-learning in distributed computing. (P.I.: 12.2.1) (CO6)					
2	**Communication** Remote Procedure Call (RPC), RMI (Remote Method Invocation), RPC Failures. Message-Oriented **Self-Learning Topics:** Middleware (Kafka, RabbitMQ	5	20%	50%	20%	
Learning Outcome	A learner will be able to: 1. Explain the working and use-cases of Remote Procedure Call (RPC) and Remote Method Invocation (RMI) in distributed systems. (P.I.: 1.4.1, 2.2.2) (CO2, CO3) 2. Analyze the causes and handling mechanisms for RPC failures in distributed environments. (P.I.: 2.1.2, 2.2.1) (CO2, CO4) 3. Describe the need for message-oriented communication and the functioning of message brokers in distributed systems. (P.I.: 1.4.1, 2.2.2) (CO3) 4. Explore modern middleware technologies like Kafka and RabbitMQ through self-directed learning. (P.I.: 12.2.1) (CO6)					
3	**Synchronization and Consensus** Physical Clock Synchronization (Centralized and Distributed). Logical Clock Synchronization (Lamport's Algorithm). Election Algorithm (Bully). Mutual Exclusion (Centralized, Lamport, Ricart–Agrawala, Maekawa, Token-Based Approaches: Suzuki-Kasami, Raymond Tree).. Deadlock Management. **Self-Learning Topics:** Distributed Consensus: Paxos and Raft	9	30%	100%	30%	
Learning Outcome	A learner will be able to: 1. Describe the need for and methods of physical and logical clock synchronization in distributed environments. 2. (P.I.: 1.4.1, 2.1.2) (CO1, CO2) 3. Apply Lamport's algorithm to implement logical clocks and establish event ordering in distributed systems. (P.I.: 2.3.1, 4.1.2) (CO3) 4. Explain and compare election algorithms such as the Bully algorithm for coordinator selection. (P.I.: 2.1.2, 2.2.1) (CO2) 5. Apply mutual exclusion algorithms including Lamport, Ricart–Agrawala, Maekawa, and token-based methods for process synchronization. 6. (P.I.: 3.1.1, 4.1.2) (CO3, CO4) 7. Describe strategies for deadlock management in distributed systems, including avoidance, detection, and prevention. 8. (P.I.: 2.3.1, 1.4.1) (CO2, CO4) 9. Explore distributed consensus mechanisms such as Paxos and Raft through self-directed learning. (P.I.: 12.2.1) (CO6)					
4	**Resource Management and Process Handling** Distributed Scheduling: Load Balancing, Load Sharing, Task Assignment, Process Migration. **Self-Learning Topics:** Containerization &	6	20%	NA	20%	

	Orchestration (Docker, Kubernetes)					
Learning Outcome	A learner will be able to: 1. Describe key concepts of distributed scheduling such as load balancing, load sharing, task assignment, and process migration. (P.I.: 1.4.1, 2.1.2) (CO2, CO4) 2. Analyze different strategies for resource allocation and task management in a distributed environment. (P.I.: 2.2.1, 3.2.1) (CO4) 3. Apply task migration techniques to improve system efficiency and reduce bottlenecks. (P.I.: 3.1.1, 4.1.2) (CO3, CO4) 4. Demonstrate understanding of containerization and orchestration tools like Docker and Kubernetes through self-learning. (P.I.: 12.2.1) (CO5, CO6)					
5	**Consistency, Replication and Fault Tolerance** CAP Theorem, BASE vs. ACID. Consistency Models: (Data Centric & Client Centric). Replication Strategies & Sharding. Distributed Caching (Redis) **Self-Learning Topics:** Redis	3	10%	NA	10%	
Learning Outcome	A learner will be able to: 1. Explain the CAP theorem and distinguish between BASE and ACID models in the context of distributed systems. (P.I.: 1.4.1, 2.1.2) (CO1, CO4) 2. Analyze data-centric and client-centric consistency models for managing distributed data. (P.I.: 2.2.1, 2.2.2) (CO4) 3. Evaluate various replication strategies and sharding techniques for scalability and fault tolerance. (P.I.: 3.2.1, 4.3.1) (CO4, CO5) 4. Apply knowledge of distributed caching systems like Redis for improving system performance. (P.I.: 5.1.1, 5.2.1) (CO3, CO5) 5. Demonstrate self-learning by exploring advanced caching use cases and tools. (P.I.: 12.2.1) (CO6)					
6	**Distributed Shared Memory & Storage** Concept of DSM and its Motivation, General architecture of DSM system, Replacement Strategy, Thrashing, Distributed File Systems: GFS, HDFS **Self-Learning Topics -** Case Studies: Ivy, TreadMarks	3	10%	NA	10%	
Learning Outcome	1. Describe the concept and motivation behind Distributed Shared Memory (DSM) in distributed systems. (P.I.: 1.4.1, 2.1.2) (CO1) 2. Explain the general architecture, replacement strategies, and issues such as thrashing in DSM systems. (P.I.: 2.2.1, 3.2.1) (CO4, CO5) 3. Analyze the design principles and components of distributed file systems like GFS and HDFS. (P.I.: 4.3.1, 3.2.1) (CO4, CO5) 4. Apply knowledge of distributed memory and storage systems to propose scalable architectures using real-world case studies. (P.I.: 5.2.1, 12.2.1) (CO5) 5. Explore DSM case studies such as Ivy and TreadMarks to understand practical implementations. (P.I.: 12.2.1) (CO5, CO6)					
	Total	**30**	**100**	**100**	**100**	

**Learning Outcome should be in bulleted form.

2.b Prerequisite Courses

No.	Semester	Name of the Course	Topic/s

1	4	Computer Networks	Tcp/ip networking
2	4	Operating Systems	Kernel, Threads, Process, Mutual Exclusion, Message passing

2.c Relevance to Future Courses

No.	Semester	Name of the Course
1	7	Big Data analytics

2.d See :- Identify real life scenarios/examples which uses the knowledge of the subject ,(Discussion on how to prepare examples and case studies e.g. "Boeing Plane": C Programming Language – Intro to Computer Science – Harvard's CS50 (2018) – Bing video)

Real Life Scenario	Concept Used
Real-time process control: aircraft control systems	Synchronization
Real-time tracking systems	Communication
Web Browsers	Communication
Distributed Databases	Consistency
Content Delivery Networks (CDNs)	Communication + Load Distribution

3 Past Results – Division-Wise

Details	Target – DEC 2025	DEC 2024	DEC 2023	DEC 2022
Course Passing % – Average of 2 Divisions	100%	--	--	--
Marks Obtained by Course Topper (mark/100)	90	--	--	--

	Division A		Division B	
Year	Initials of Teacher	% Result	Initials of Teacher	% Result
Dec 2024	NA	NA	NA	NA
Dec 2023	NA	NA	NA	NA
Dec 2022	NA	NA	NA	NA

4 All the Learning Resources – Books and E-Resources

4.a List of Textbooks (T – Symbol for Textbooks) to be Referred by Students

Sr. No	Textbook Titles	Author/s	Publisher	Edition	Module Nos.	Available in our Library
1	Distributed Systems: Principles and Paradigms	Andrew S. Tanenbaum and Maarten Van Steen.	Pearson	2	1 to 6	Yes
2	Distributed Systems: Concepts and Design	George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair.	Pearson	5	1 to 6	Yes

4.b List of Reference Books (R – Symbol for Reference Books) to be Referred by Students

Sr. No	Reference Book Titles	Author/s	Publisher	Edition	Module Nos.	Available in our Library
1	Distributed Computing: Principles, Algorithms, and Systems	Ajay D. Kshemkalyani and Mukesh Singhal.	Cambridge	2	1,2,3,4,5	Yes
2	Distributed Operating Systems	P.K. Sinha.	PHI	1	1,2,4,6	Yes

4.c List of E - Books (E – Symbol for E-Books) to be Referred by Students

Sr. No	E- Book Titles	Author/s	Publisher	Edition	Module Nos.	Available in our Library
1	Distributed Systems (https://www.distributed-systems.net/index.php/books/ds4/)	Maarten Steen	Distributed Systems.NET	3rd	4-6	Available Online
2	Distributed Systems: An Algorithmic Approach (https://jagdishkapadnis.wordpress.com/wp-content/uploads/2018/08/santosh-kumar-distributed-systems-algorithmic-approach-information-812.pdf)	Sukumar Ghosh	CRC	1st	1,2,3,4	

4.d Reading latest / top rated research papers (at least 5 papers)

Name of Paper	Name of Authors (Background)	Published in		Problem Statement	Available in our Library
		Date	Journal		
Big Data Computing with Distributed Computing Frameworks	Gurjit Singh Bhathal Punjabi University, Patiala · Department of Computer	January 2019	Innovations in Electronics and Communication Engineeri	The rapid growth in the volume, velocity, and veracity of data presents significant challenges in processing, storing, and analyzing large datasets within short timeframes using traditional approaches. As single	https://www.researchgate.net/publication/330931223_Big_Data_Computing_with_Distributed_Computing_Frameworks

Name of Paper	Name of Authors (Background)	Published in		Problem Statement	Available in our Library
		Date	Journal		
	Science and Engineering Doctor of Engineering		ng (pp.467-477)	systems struggle to manage these massive datasets efficiently, distributed computing frameworks such as Hadoop, utilizing MapReduce for distributed processing and HDFS for storage, offer a scalable solution. However, with the increasing need for real-time data processing, newer technologies like Apache Spark, Storm, and Flink have evolved to handle stream and real-time data more effectively. The problem lies in designing and implementing distributed computing solutions that not only address scalability, fault tolerance, and data consistency but also improve the efficiency and speed of processing large datasets in real-time or near-real-time environments.	
Dynamic scheduling in distributed transactional memory	Costas BuschView Augusta University, Augusta, GA, USA Maurice Herlihy Brown University, Providence, RI, USA	30 November 2021	Springer	The problem addressed in this paper is the scheduling of transactions in distributed transactional memory systems, where transactions operate on shared, mobile objects across a communication network. These transactions request, execute, and transfer objects between nodes, and the challenge is to efficiently schedule these transactions to minimize execution time. While previous studies focused on offline batch scheduling, where transactions are predetermined, it is known that minimizing execution time for arbitrary communication graphs in this setting is NP-hard. The key problem is to develop efficient and near-optimal scheduling algorithms for dynamic, online scheduling, where transactions arrive over time and are not known in advance. The goal is to design scheduling algorithms with provable performance	https://par.nsf.gov/servlets/purl/10315663

Name of Paper	Name of Authors (Background)	Published in		Problem Statement	Available in our Library
		Date	Journal		
				guarantees, ensuring efficient execution even in specialized network architectures, and to adapt these solutions from centralized to distributed schedulers in a way that maintains performance and scalability.	
TOPS: Two-Phase Scheduling for Distributed Real-time Systems	Mohammed Alghamdi Albaha University	July 2020	The IEEE Computing, Communications and IT Applications Conference (ComComAp)At: Beijing, China.	This work proposes a two-phase scheduling technique (TOPS) for distributed real-time systems, where the first phase generates a scheduling sequence and the second dispatches tasks to computing nodes. The two phases are independent, allowing for flexibility in modifying policies. TOPS helps assess the impact of sorting and scheduling policies on system performance. A prototype implementation with three sorting policies and two scheduling policies reveals that combining the Earliest-Deadline-First (EDF) and As-Early-As-Possible (AEAP) policies results in optimized performance among six candidate algorithms.	https://ieeexplore.ieee.org/document/7017186
Techniques of Enhancing Synchronization Efficiency of Distributed Real Time Operating Systems	A.W.H.P.Alagalla, U.U.S.K. Rajapaksha	Feb 2022	2022 2nd International Conference on Advanced Research in Computing (ICARC) 308-313. Available from: 10.1109/ICARC54489.2022.9754095	This research focuses on enhancing the accuracy and efficiency of Real-Time Operating Systems (RTOS) through synchronization techniques, addressing the lack of comprehensive analysis in optimizing synchronization methods for RTOS and Distributed RTOS (DRTOS). It explores various synchronization platforms, including backup synchronization, on-chip memory handling, location-based network systems, and hardware-oriented synchronization. The paper highlights the challenges faced by existing RTOS and DRTOS systems and proposes solutions to improve the efficiency of distributed node networks using multi-core processors. Multiple	https://ieeexplore.ieee.org/document/9754095

Name of Paper	Name of Authors (Background)	Published in		Problem Statement	Available in our Library
		Date	Journal		
				synchronization techniques, including lock-based, lock-free, semaphore-based, mutex-based, and hardware-related approaches, are discussed to optimize performance. This work is particularly beneficial for developing high-performance processing systems in IoT-based distributed networks, supporting desired objectives and features.	
A modified physical clock synchronization algorithm	Subho Chaudhuri,Somnath Nandy	2013	2013 3rd IEEE International Advance Computing Conference (IACC) 34-39. Available from: 10.1109/IADCC.2013.6506811	This paper addresses the problem of clock synchronization in distributed systems, where maintaining synchronized clocks is crucial for improving performance. The goal is to provide all parts of a distributed system with a common notion of time. The paper proposes a clock synchronization algorithm with bounded clock drift, incorporating both external and internal synchronization methods. The algorithm employs a two-level synchronization approach to synchronize the local clocks of nodes while ensuring fault tolerance. This approach combines external synchronization with an external reference and internal synchronization to maintain clock consistency across the distributed system.	https://ieeexplore.ieee.org/document/6506811
Distributed computing systems synchronization modeling for solving machine learning tasks	T V Azarnova, P V Polukhin	2021	Journal of Physics: Conference Series 1902 (2021) 012050 IOP Publishing doi:10.1088/1742-6596/190	Despite the growing use of distributed computing systems for handling large-scale data and machine learning tasks, these systems often face inefficiencies due to unpredictable processing delays, inadequate synchronization, and suboptimal resource allocation. There is a need for a robust framework that can model, evaluate, and predict system behavior—specifically processing delays and query execution—using formal	https://iopscience.iop.org/article/10.1088/1742-6596/1902/1/012050/pdf

Name of Paper	Name of Authors (Background)	Published in		Problem Statement	Available in our Library
		Date	Journal		
			2/1/012050	methods such as queuing theory. Such a framework would enhance system efficiency by improving request processing mechanisms and ensuring consistent computational performance in distributed environments.	

4.e **Based on research paper an identify the current Problem statement**

Problem Statement			Used in				
	Quiz	Assignment	Lab	Mini Project	Poster Presentation	Test	Any Other
Despite the growing use of distributed computing systems for handling large-scale data and machine learning tasks, these systems often face inefficiencies due to unpredictable processing delays, inadequate synchronization, and suboptimal resource allocation. There is a need for a robust framework that can model, evaluate, and predict system behavior—specifically processing delays and query execution—using formal methods such as queuing theory.		Assignment					

Such a framework would enhance system efficiency by improving request processing mechanisms and ensuring consistent computational performance in distributed environments.							
This paper addresses clock synchronization in distributed systems, aiming to provide a common notion of time across all nodes. It proposes an algorithm with bounded clock drift, using a two-level synchronization approach. The approach combines external synchronization with a reference and internal synchronization, ensuring fault tolerance and maintaining clock consistency throughout the system.						OBT	
This research focuses on improving the efficiency and accuracy of Real-Time Operating Systems (RTOS) and Distributed RTOS (DRTOS)						OBT	

through optimized synchronization techniques. It examines various synchronization methods, including backup synchronization, on-chip memory handling, location-based networks, and hardware-oriented approaches. The paper addresses challenges in current systems and proposes solutions to enhance distributed node network performance using multi-core processors. Several synchronization techniques, such as lock-based, lock-free, semaphore-based, mutex-based, and hardware-related methods, are discussed, with a focus on optimizing performance for high-performance IoT-based distributed networks.							

4.f **Identify Companies / Industries which use the knowledge of the subject and thus may provide Internships and final Placements**

Name of the Company	To be / Contacted for

	Student Internship	Student Final Placement	Faculty Internship
Tata Consultancy Services (TCS)	√	√	
Cyberfrat	√		√
BMS		√	
IRCTC		√	

4.g **Identify suitable relevant TOP Guest Speakers from Industry,**
Example: - (CS50 Lecture by Mark Zuckerberg - 7 December 2005 - YouTube)

Name of the Identified Guest Speaker	Designation	Name of the Company
Robert Morris	Professor	MIT Computer Science and Artificial Intelligence Laboratory (CSAIL) http://nil.lcs.mit.edu/rtm/
Chris Colohan	Distributed Systems Expert Google	Freelancer https://www.colohan.com/

4.h **Identify relevant technical competitions to participate [Competitions -Paper Presentations, Projects, Hackathons, IVs etc..]**

Name of the Relevant Technical Competition Identified to participate	Organized by	Date of the Event
Hackathon	TSEC	August 25
Automation Expo	IED	August 25

4.i **Identify faculty in TOP schools / Universities who are teaching same / similar subject and develop rapport e.g. Exchange Lecture Material (Assignments / Tests / Project etc..), Joint Paper Publication**

University	Name of the Course	Name of Faculty	Type of Collaboration		
			Exchange of Lecture Material	Joint Publication/ Research	Other
MIT	MIT 6.824 Distributed Systems https://www.youtube.com/watch?v=cQP8WApzIQQ&list=PLrw6a1wE39_tb2fErI4-WkMbsvGQk9_UB	Robert Morris	Yes		
Palo Alto	**Distributed Systems** https://www.distribute	Chris Colohan	Yes		

	dsystemscourse.com/				

4.j **Module Best Available in – Title of the best resource [from *4.a* to *4.d* in this AAP] and other details as necessary**

Module No.	Title of the Module	Textbook	Mention the Tile					
			Reference Book	E-books	Journal	E-Journal	Available in our Library	V-refer
1	Introduction	T1,T2	R1,R2	E2		International Journal of Distributed Systems and Technologies (IJDST) https://www.igi-global.com/journal/international-journal-distributed-systems-technologies/1164	√	
2	Communication	T1,T2	R1,R2	E2			√	
3	Synchronization	T1,T2	R1	E2			√	
4	Process & Resource Management	T1,T2	R1,R2	E1,E2			√	
5	Consistency Model	T1,T2	R1	E			√	
6	Distributed File Systems	T1,T2	R2	E1		International Journal of Distributed Systems and Technologies (IJDST) https://www.igi-global.com/journal/international-journal-distributed-systems-technologies/1164	√	

4.k **Referred to any top-rated university in that subject for content**

University	Name of the Course	Name of Faculty	Date of Delivery of the Course	Remarks
University of Waterloo	Distributed Computer Systems	S. Keshav	Winter 2012	In CS 436, Professor S. Keshav explores the core principles of distributed systems, including communication models, fault tolerance, and distributed file systems. Recorded in Winter 2012, these lectures provide students with a deep understanding of distributed computing, preparing them for real-world challenges in the field. Students are encouraged to actively engage with the material

				through notes, discussions, and problem sets to strengthen their knowledge.
University of Cambridge	Concurrent and Distributed Systems	https://www.cl.cam.ac.uk/teaching/2122/ConcDisSys/	January 2021	The aim of the first half of the course is to introduce concurrency control concepts and their implications for system design and implementation. The aims of the latter half of the course are to study the fundamental characteristics of distributed systems, including their models and architectures; the implications for software design; some of the techniques that have been used to build them; and the resulting details of good distributed algorithms and applications

4.l **Faculty received any certification related to this subject. List of Certifications Identified / Done**

Course	Certifying Agency	No. of Hours	Level of the Course		Certification		Remarks
			Introductory	Advance Skill Development	Done on	Proposed to be on	
Big Data 301	Infosys Spring Board	35H		√		October 2025	This course will cover Advanced Hive, Sqoop, Spark Core and Spark SQL. Course is blended with demos, quizzes and hands on try out assignments.

4.m **Completed subject wise/cluster wise training with cluster mentor. List of relevant Refresher Course Identified / Done**

Course	Certifying Agency (As suggested by DAB/Cluster Mentor/Industry/University other than MU)	Certification		Remarks
		Done on	Proposed to be on	
Pedagogy	Gamification		August 2025	Students learn the theory at home (via videos or readings) and use class time for discussions or hands-on activities.
PBL	Case Study based PBLE		Entire Semester	Assign a scenario like designing a distributed system for a ride-sharing application. Let students identify challenges (e.g., fault tolerance, load balancing) and propose solutions
Sub. Content Training	Concurrent and Distributed Computing with Python https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0130944073676636161763_shared/overview			This course will help you resolve these difficulties. You will start by exploring the basic concepts of concurrency and distributed computing, and you'll learn which Python libraries are relevant to these. You will not only learn to see Celery as a way to build-in concurrency into your apps, but also Pyro as an alternative to Celery. You will create processes and manage processes along with interprocess communication; combine coroutines with threads and processes; practice the management of process pools; implement asynchronous tasks/job queues using AsyncResult and Celery backends; invoke remote methods in your Python-based code, and use these skills and concepts when working with AWS for Python.

4.n Best Practices Identified and adopted

No.	Item	Best Practices Identified		
		Univ. 1	Univ. 2	Univ. 3
1	Microsite	Distributed Systems https://www.distributedsystemscourse.com/		
2	Video Lectures	Pallab Dasgupta, IIT Kharagpur https://www.youtube.com/watch?v=dlBVWMdGhqw&list=PLUJ7JmcrTifBROWODSG8wgyl20XgBuE-N	University of Cambridge Distributed Systems https://www.youtube.com/watch?v=UEAMfLPZZhE&list=PLeKd45zvjcDFUEv_ohr_HdUFe97RItdiB	
3	Assignments			
4	Mini Project			
5	Assessment Metric			
6	Quizzes			

7	Labs/ Practical (PBL)	Problems in Distributed ML feat. Richard Liaw | Stanford https://www.youtube.com/watch?v=R7N3quJcGNQ		
8	Tests			
9	Peer Assessment			
10	Any Other	Distributed Systems University of Cambridge https://www.cl.cam.ac.uk/teaching/2122/ConcDisSys/dist-sys-notes.pdf		

4.o Web Links for Online Notes/YouTube/VIT Digital Content/VIT Lecture Capture/NPTEL Videos

Students can view lectures by VIT professors, captured through LMS 'Lecture Capture' in VIT campus for previous years.

No.	Websites / Links	Module Nos.
1	1. http://micronica.com.au/catalog/sharer/ 2.https://www.youtube.com/watch?v=mm3r8EG4wLQ&index=6&list=PLmPpJG5-RKf0RUQ7VYjHn6pnoWT2__4CM 3. https://www.youtube.com/watch?v=YY0SvestyaQ 4. https://www.youtube.com/watch?v=rYK-kTBUrK4	1
2	1. http://www.sanfoundry.com/computer-networks-questions-answers-rpc/ 2. https://www.youtube.com/watch?v=-6Uoku-M6oY&t=556s 3. https://www.youtube.com/watch?v=I_3zU9HeDOs 4. https://www.youtube.com/watch?v=gr7oaiUsxSU 5. https://www.youtube.com/watch?v=ILeAeFZOkMI 6. https://www.youtube.com/watch?v=dQE6BDNc3z0	2
3	1. https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjgnIXS2Y_RAhULtI8KHdkMD9cQjRwIBw&url=http%3A%2F%2Fsynch.dj%2F&psig=AFQjCNGrTbMb6PNac0ysly5Tz3sRov8g2Q&ust=1482767502851734 2. http://synch.dj/ 3. https://www.youtube.com/watch?v=GAZAT068Hbg 4. https://www.youtube.com/watch?v=bnrD2n55dfk 5. https://www.youtube.com/watch?v=7_9CR9aRKBk 6. https://www.youtube.com/watch?v=XDdqF8FfRx8 7. https://www.youtube.com/watch?v=xaISZOQ-PWY 8. https://www.youtube.com/watch?v=aWne_qIR2XI 9. https://www.youtube.com/watch?v=jFULOEHPqgo 10. https://www.youtube.com/watch?v=KC3J52L0jFE	3,4
4	1. https://www.youtube.com/watch?v=fUrKt-AQYtE 2. https://www.youtube.com/watch?v=Lz5DNqfpN1w 3. https://www.youtube.com/watch?v=L_sUyOAjC6M	5,6

5	LEC RECORDING: MS TEAMS	All

4.p Recommended MOOC Courses like Coursera / NPTEL / MIT-OCW / edX/VAC etc.

Sr. No.	MOOC Course Link	Course conducted by – Person / University / Institute / Industry	Course Duration	Certificate (Y / N)
1	https://nptel.ac.in/courses/106/106/106106168/#	Dr. Rajiv Misra/IIT Patna	8 weeks	Y
2	A Parallel Programming for Distributed Systems in Java https://www.coursera.org/specializations/pcdp	Dr. Eric Allen, Senior Vice President, Two Sigma	4 weeks	Y
3	Git for Distributed Development https://www.coursera.org/learn/git-distributed-development	Jerry Cooperstein	3 Weeks	Y

5 Consolidated Course Lesson Plan

	From (date/month/year)	From (date/month/year)	Total Number of Weeks
Semester Duration	6/7/25	18/10/25	15

Week	Lecture no.	Module No.	Lecture Topics / MSE / BSA planned to be covered	Actual date of Completion (Handwritten)	COs Mapped	Mapping Bloom Taxonomy level	Recommended Prior Viewing / Reading: Lecture No. (on LMS)	Recommended Prior Viewing / Reading: Chapter No./ Books/ Web Site
1	1	1	Characterization of Distributed Systems: Issues, Goals		CO1	"Understanding" (Level 2)		T1/1/1 T2/1/1
	2	1	Distributed System Models		CO1	"Understanding" (Level 2)		T1/1/17 T2/1/37
2	3	1	Hardware concepts, Software Concept.		CO1	"Understanding" (Level 2)		T1/1/17 T2/1/37
	4	1	Middleware: Introduction to DCE		CO1	"Understanding" (Level 2)		T1/1/14 T2/1/37
3	5	2	Layered Protocols, Interprocess communication (IPC): MPI,		CO2	"Applying" (Level 3)		T1/4/116 T2/2/146

Week	Lecture no.	Module No.	Lecture Topics / MSE / BSA planned to be covered	Actual date of Completion (Handwritten)	COs Mapped	Mapping Bloom Taxonomy level	Recommended Prior Viewing / Reading	
							Lecture No. (on LMS)	Chapter No./ Books/ Web Site
	6	2	Remote Procedure Call (RPC), Parameter-passing Semantics, RPC Failures & Process resilience,		CO2	"Applying" (Level 3)		T1/4/125 T2/2/186
4	7	2	Remote Method Invocation (RMI),		CO2	"Applying" (Level 3)		T1/4/116 T2/2/217
	8	2	Message Oriented Communication, Stream Oriented		CO2	"Applying" (Level 3)		T1/4/140 T2/6/230
5	9	2	Communication, Group Communication & Issues		CO2	"Applying" (Level 3)		T1/4/166 T2/6/230
	10	3	Clock Synchronization (Physical & Logical),		CO3	"Analyzing" (Level 4)		T1/6/232 T2/14/599
6	11	3	Mutual Exclusion, Distributed Mutual Exclusion-Classification of mutual Exclusion Algorithm.		CO3	"Analyzing" (Level 4)		T1/6/252 T2/14/633
	12	3	Non Token based Algorithms: Lamport Algorithm, Ricart–Agrawala's Algorithm, Maekawa's Algorithm		CO3	"Analyzing" (Level 4)		T1/6/252 T2/14/633
7	13	3	Token Based Algorithms: Raymond Tree Algorithm,		CO3	"Analyzing" (Level 4)		T1/6/252 T2/14/633

Week	Lecture no.	Module No.	Lecture Topics / MSE / BSA planned to be covered	Actual date of Completion (Handwritten)	COs Mapped	Mapping Bloom Taxonomy level	Recommended Prior Viewing / Reading	
							Lecture No. (on LMS)	Chapter No./ Books/ Web Site
	14	3	Suzuki-Kasami's Broardcast Algorithms,		CO3	"Analyzing" (Level 4)		T1/6/252 T2/14/633
8	15	3	Deadlock Management (Avoidance, Detection, Prevention)		CO3	"Analyzing" (Level 4)		T2/17/743
	16	3	Deadlock Management (Avoidance, Detection, Prevention)		CO3	"Analyzing" (Level 4)		T2/17/743
9	17	3	Election Algorithms		CO3	"Analyzing" (Level 4)		T1/6/263
	18	3	Election Algorithms		CO3	"Analyzing" (Level 4)		T1/6/263
10	19	4	Desirable Features of global Scheduling algorithm,		CO4	"Understanding" (Level 2)		https://www.researchgate.net/publication/267786164_Resource_Management_in_Distributed_System
	20	4	Task assignment approach		CO4	"Understanding" (Level 2)		
11	21	4	, Load balancing approach, load sharing approach		CO4	"Understanding" (Level 2)		

Week	Lecture no.	Module No.	Lecture Topics / MSE / BSA planned to be covered	Actual date of Completion (Handwritten)	COs Mapped	Mapping Bloom Taxonomy level	Recommended Prior Viewing / Reading	
							Lecture No. (on LMS)	Chapter No./ Books/ Web Site
	22	4	Introduction to process management,		CO4	"Understanding" (Level 2)		T1/3/69
12	23	4	Process migration.		CO4	"Understanding" (Level 2)		T1/3/69
	24	4	CAP Theorem, BASE vs. ACID		CO4	"Understanding" (Level 2)		T1/3/103
13	25	5	Data-Centric		CO5	"Understanding" (Level 2)		T1/7/274
	26	5	Client- Centric Consistency Models.		CO5	"Understanding" (Level 2)		T1/7/276
14	27	5	Replication Strategies & Sharding. Distributed Caching (Redis)		CO5	"Understanding" (Level 2)		T1/7/288
	28	6	Concept of DSM and its Motivation, General architecture of DSM system		CO6	"Understanding" (Level 2)		T1/11/491 T2/12/521
15	29	6	Replacement Strategy, Thrashing,		CO6	"Understanding" (Level 2)		T1/11/491 T2/12/522

Week	Lecture no.	Module No.	Lecture Topics / MSE / BSA planned to be covered	Actual date of Completion (Handwritten)	COs Mapped	Mapping Bloom Taxonomy level	Recommended Prior Viewing / Reading	
							Lecture No. (on LMS)	Chapter No./ Books/ Web Site
	30	6	Distributed File Systems: GFS, HDFS		CO6	"Understanding" (Level 2)		T1/11/491 T2/12/536 T2/21/916

6 Rubric for Grading and Marking of Term Work (inform students at the beginning of semester)

- Activity/ies should be designed as per reference of credit structure.
- If the subject is of 2 credit, activity/ assignment should be design for 2 hours with appropriate complexity and engaging time.

Theory (ISA=15)												
Class Participation	Activity-1	Activity-2	Activity-3	Activity-4	Activity-5	Activity-6	Activity-7	Activity-8	Activity-9	Activity-10	Activity-11	Total
TE CMPN	15	15	15	15	15	15	15	15	15	15	15	15 (Best of 10 Assignments)

Total			
Class Participation	Lab Participation (PBLE based)	Lab Work	Average/25
TE CMPN	15 Marks	10 Marks	Average of all 10 Task/25

Class Participation	MSE-1	MSE-2	ESE*	Total
TE CMPN	20	20	40	7

7 Assignments / Tutorials Details

<table>
<tr><th>Assignment/ Tutorial No.</th><th>Title of the Assignments / Tutorials</th><th>CO Map</th><th>Mapping Bloom Taxonomy Level</th><th>Assignment/ Tutorials given to Students on</th><th>Assignments to be submitted back on</th></tr>
<tr><td>1</td><td>

Transparency:

Reference: Distributed computing systems synchronization modeling for solving machine learning tasks https://iopscience.iop.org/article/10.1088/1742-6596/1902/1/012050/pdf

<table>
<tr><th>Sr. No.</th><th>Scenario</th><th>Transparency Type (To be filled by student)</th><th>Justification (To be filled by student)</th></tr>
<tr><td>1</td><td>A user accesses course material via a portal without knowing if it is stored on their local college server or central university server.</td><td></td><td></td></tr>
<tr><td>2</td><td>After a server crash, the system migrates the session to a backup server, and the user continues uninterrupted.</td><td></td><td></td></tr>
<tr><td>3</td><td>A cloud-based game server handles multiple players' moves without letting one player overwrite another's actions.</td><td></td><td></td></tr>
<tr><td>4</td><td>A student updates their email address in the system, and it reflects across all university applications instantly.</td><td></td><td></td></tr>
<tr><td>5</td><td>The system optimally distributes user traffic among various servers so that no single server is overloaded.</td><td></td><td></td></tr>
<tr><td>6</td><td>A teacher accesses attendance records using the same function, whether it's stored on a local device or on a remote server.</td><td></td><td></td></tr>
<tr><td>7</td><td>During system upgrade, active user sessions are</td><td></td><td></td></tr>
</table>

</td><td>CO 1</td><td>"Understanding" (Level 2)</td><td>6/7/25</td><td>21/07/25</td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td></td><td>shifted to another server automatically, without asking users to log in again.</td><td></td><td></td></tr>
<tr><td>8</td><td>A mobile banking app automatically retries a failed transaction due to server timeout without informing the user.</td><td></td><td></td></tr>
<tr><td>9</td><td>A content delivery network stores copies of a file in multiple locations but ensures users always get the latest version.</td><td></td><td></td></tr>
<tr><td>10</td><td>The system scales up by adding new nodes during peak exam form submissions, with no effect on user experience.</td><td></td><td></td></tr>
</table></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>Video Assignment 1
https://www.youtube.com/watch?v=5YCA3kCYw6Q
https://youtu.be/CESKgdNiKJw?si=ubZydtq4P8mrTNeo
https://youtu.be/cHTvVr1w_7s?si=Uh7hj5uXN1ltqO0Z

Write a short summary on every video & attempt quiz on it.</td><td>CO 1</td><td>"Understanding" (Level 2)</td><td>6/7/25</td><td>28/87/25</td></tr>
<tr><td>3</td><td>Communication
1. How does the choice of transport protocol (TCP vs. UDP) affect the performance and reliability of your RPC system?
2. Create a comparative table of RPC, RMI, and CORBA, focusing on features such as platform independence, ease of implementation, scalability, and fault tolerance.
Link: https://www.youtube.com/watch?v=u4nU6w2ZK3s (Computerphile)
Title: Difference Between TCP and UDP (with animation)
Link: https://www.youtube.com/watch?v=Vdc8TCESIg8</td><td>CO 2</td><td>"Applying" (Level 3)</td><td>6/7/25</td><td>4/8/25</td></tr>
<tr><td>4</td><td>NETWORK TIME PROTOCOL in LINUX:
1. Describe the key components of NTP: reference clock, NTP server, and NTP client
2. Discuss the NTP clock synchronization algorithm and how it compensates for network delays and jitter.
(https://www.youtube.com/watch?v=oCtkwEjhyD4)</td><td>CO 3</td><td>"Applying" (Level 3)</td><td>6/7/25</td><td>11/8/25</td></tr>
</table>

5	A distributed system has 5 nodes (P1 to P5). Each process uses the Ricart–Agrawala algorithm for entering the critical section. All processes request access to the critical section once. (a) How many total messages (request + reply) are exchanged for all processes to enter the critical section once (sequentially, not concurrently)? (b) If two processes request entry at the same time, explain how the tie is broken. (c) Assume logical clock timestamps: P2 sends a request at T = 5 P3 sends a request at T = 3 Who will enter the critical section first? Show the working. 1. A distributed system has 5 nodes (P1 to P5). Each process uses the centralized mutual exclusion algorithm for entering the critical section. All processes request access to the critical section once. (d) How many total messages (request + reply) are exchanged for all processes to enter the critical section once (sequentially, not concurrently)? (e) If two processes request entry at the same time, explain how the tie is broken. (f) Assume logical clock timestamps: P2 sends a request at T = 5 P3 sends a request at T = 3 2. Who will enter the critical section first? Show the working.	CO 3	"Applying " (Level 3)	6/7/25	18/8/25
6	**Process & Resource Management:** Welcome to the "Distributed Chef Challenge"! You are the lead developer for a futuristic **Cloud Kitchen**, where multiple kitchens (nodes) receive dynamic food orders from various customers across a city. Your goal is to ensure that the tasks (food orders) are scheduled, processed, and balanced efficiently using modern distributed computing techniques. You must **simulate**, **visualize**, or **prototype** how your smart kitchen handles **load balancing**, **task assignment**, and **process migration**, while also deploying the system using **Docker containers** and managing them using **Kubernetes**. **Q. Animated Storyboard/Comic Strip: Show how your cloud kitchen works with distributed scheduling using visual storytelling (e.g., Canva, Pixton).** **Resources:** **1. Pixton – Make Comics with Characters** • **Website: https://www.pixton.com/**	CO 4	"Applying " (Level 3)	6/7/25	25/8/25

	• Use: Drag-and-drop interface to create characters, scenes, and dialogues. • Best For: Comic strip representations of how tasks (orders) are assigned to kitchen nodes. • Tutorial: [How to Create Comics in Pixton (YouTube)]()				
	2. Canva Storyboard & Animation • Website: https://www.canva.com/create/storyboards/ • Use: Use templates for animated storyboards; add visuals, icons, and movement. • Best For: Explaining process migration or task overload handling visually. • Tutorial: [Storyboard Animation using Canva (YouTube)]()				
	3. Storyboard That – Digital Storyboard Tool • Website: https://www.storyboardthat.com/ • Use: Design step-by-step process flows using prebuilt characters and props. • Best For: Showing each kitchen node, its current load, and scheduling strategies visually. • Tutorial: [StoryboardThat Tutorial for Beginners]()				
	4. Animaker – Create Animated Explainer Videos • Website: https://www.animaker.com/ • Use: Build animated scenes to explain how Kubernetes handles container failures or how Docker deploys kitchen services. • Best For: Students who want to present through short explainer video format. • Tutorial: [How to Create Explainer Videos with Animaker]()				
	5. Sketchboard.io – Collaborative Visual Flow Tool • Website: https://sketchboard.io/ • Use: Simple drag-and-drop diagramming for system visualization and fun cartoon-style layouts. • Best For: Group work or whiteboard-style distributed system drawings.				

7	**Working with Apache Cassandra: Using Replication, Consistency, & CQL** https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013817297142874112131/overview	CO 5	"Applying" (Level 3)	6/7/25	1/9/25
8	**Distributed File System:** Course of Big Data Analytics https://www.simplilearn.com/big-data-tools-free-course-online-skillup **OR** Introduction to Big Data and Hadoop https://www.mygreatlearning.com/academy/learn-for-free/courses/mastering-big-data-analytics Refer: https://www.researchgate.net/publication/330931223_Big_Data_Computing_with_Distributed_Computing_Frameworks	CO 6	"Understanding" (Level 2)	6/7/25	8/9/25
9	**Guest Lecture Assignment** Based on the content presented during the lecture, this assignment will be designed to reinforce the concepts discussed and challenge students to apply their newly acquired knowledge in practical scenarios. The assignment will focus on key learning objectives and provide an opportunity for critical thinking and problem-solving in the context of the subject matter.		"Understanding" (Level 2)	6/7/25	15/9/25
10	**Video Assignment 2: Google File System - Paper that inspired Hadoop** https://www.youtube.com/watch?v=eRgFNW4QFDc Refer: https://www.researchgate.net/publication/330931223_Big_Data_Computing_with_Distributed_Computing_Frameworks	CO 6	"Understanding" (Level 2)	6/7/25	22/9/25
11	**Open Book Test**	CO 3-CO 5	"Understanding" (Level 2)	6/7/25	29/9/25

Analysis of Assignment / Tutorial Questions and Related Resources

Assignment /	Week No.	Type* (√)			Module No.	Based on #			Question Type (√)	
		OT	CS	DTP		Textbook	Reference Book	Other Learning Resource	Real Life Assignments	Thought Provoking
1	3			√	1	T1,T2	R2			√
2	4			√	1	T1,T2	R2	https://www.youtube.com/watch?v=eRgFNW4QFDc		√
3	5		√		2	T1, T2	R2		√	
4	6		√		3	T1	R1	(https://www.youtube.com/watch?v=oCtkwEjhyD4)	√	
5	7		√		3	T1	R1		√	
6	8			√	4	T2	R2			√
7	9			√	6	T2	R2			√
8	10			√	6	T2	R2			√
9	11			√	5	T1	R1			√
10	12			√	6	T2	R2	https://www.youtube.com/watch?v=eRgFNW4QFDc		√
11	13			√	5	T1	R1			√

* Tick (√) the Type of the Assignment: Online Tools (OT); Collaborative Assignments (CS); Design /Thought provoking (DTP)

Write number for textbook, reference book, other learning resource from this AAP – *from Points 4.a to 4.d*

8 In Semester Assessment (ISE) / Other Class Test / Open Book Test (OBT)/Take Home Test (THT) Details

Tests	Test Dates	Module No.	CO Map	MSE Question Paper Pattern	Policy
ISE					
Pop Quiz	2/10/25	3	CO3	Kahoot Quiz	
Open Book Test	2/10/25	3,4,5	CO3,CO4,CO5	5 Problem based Questions	All 5 to be attempted
Take Home Test					
Class tests / prelims					
Class tests / prelims					

Any other test/exams					

*** Failures of IA test (IA1+IA2) shall appear for IA test in the next semester. There is no provision for re-test in the same semester.**

9. Practical Activities

Div A: Problem Statement 1: Distributed Online Proctored Examination System
Detailed Problem Statement:
Educational institutions offering remote education need a secure, scalable, and fault-tolerant distributed online examination system. The platform must support:
Real-time question delivery
Live proctoring via video feed
Auto-save of answers with synchronization
Timer synchronization across clients
Single attempt submission with conflict prevention
System scalability across geographic regions
Data consistency, fault tolerance, and load balancing
Tasks

Task	Module No	Task Statement	Topics to be highlighted	CO	Bloom's Level	References
1	1	Design the system architecture covering exam UI, timer, proctoring, question services, and storage	Introduction to DS Framework	CO1	Create	System Design Primer Gaurav Sen - System Design
2	2	Develop RPC APIs to enable communication between services	RPC	CO2, CO3	Apply	gRPC Docs, REST API Guide gRPC Tutorial – Nana
3	2	Implement RMI/RPC calls for operations like startExam(), submitAnswer()	RMI	CO3	Apply	RPC vs RMI – GFG Java RMI – Apna College
4	3	Apply Lamport or NTP clock synchronization to maintain consistency in timestamps	Clock Synchronization	CO3	Apply	Lamport Clock - Stanford Lamport Clock – Gate Smashers
5	3	Apply Ricart–Agrawala mutual exclusion to prevent multiple submissions	Mutual Exclusion	CO4	Apply	Ricart–Agrawala Ricart-Agrawala – Simplified
6	3	Simulate and resolve deadlocks between auto-save and final submission	Deadlocks	CO4	Analyze	Deadlock in DS – GFG Deadlock – Neso
7	4	Implement load balancing strategy (Least Connections) on proctoring servers	Load Balancing	CO4	Apply	NGINX Load Balancing LB – Gaurav Sen
8	5	Implement eventual consistency for answer auto-save sync	Consistency Models	CO5	Evaluate	CAP Theorem & Consistency CAP Explained

Task	Module No	Task Statement	Topics to be highlighted	CO	Bloom's Level	References
9	6	Integrate Redis caching and database replication for fault tolerance	Caching and Replication	CO5	Apply	Redis Docs Redis Tutorial – Fireship
10	6	Deploy complete system using Docker Compose/Kubernetes and simulate exam load	Dockers	CO6	Create	Docker Compose Docs Docker – Nana

Div B: Problem Statement 2: Distributed Smart City Traffic Management System

Detailed Problem Statement:

Smart cities demand an intelligent distributed traffic management system where each intersection functions autonomously yet coordinates with nearby intersections to:

Optimize traffic flow dynamically

Synchronize signal states across intersections

Prioritize emergency vehicle routing

Prevent gridlocks

Achieve system-wide consistency and fault tolerance

Tasks

Task	Module no	Task Statement	Topics to be highlighted	CO	Bloom's Level	References
1	1	Design traffic control system architecture with distributed junction nodes	Introduction to DS Framework	CO1	Create	Smart City Architecture – IEEE
2	2	Develop REST/gRPC APIs for inter-junction communication	RPC	CO2, CO3	Apply	gRPC Docs
3	2	Implement RPC to trigger signal updates	RMI	CO3	Apply	RPC in DS – GFG
4	3	Apply clock sync (Berkeley/NTP) across controllers	Clock Synchronization	CO3	Apply	Clock Sync – GFG
5	3	Implement Suzuki-Kasami token-based mutual exclusion for signal control	Mutual Exclusion	CO4	Apply	Suzuki-Kasami – GFG
6	3	Simulate gridlock deadlock and resolve it	Deadlocks	CO4	Analyze	Deadlock in DS – Neso
7	4	Apply load balancing across traffic nodes	Load Balancing	CO4	Apply	LB Algorithms – Kemp
8	5	Apply consistency model for signal data sync	Consistency Models	CO5	Evaluate	Consistency Models – Microsoft Azure
9	6	Integrate Redis caching for route data	Caching and Replication	CO5	Apply	Redis Docs
10	6	Deploy via Docker Swarm/Kubernetes for multi-junction simulation	Dockers	CO6	Create	Docker Swarm Docs

Div C: Problem Statement 3: Distributed Electronic Health Records (EHR) SystemDetailed Problem Statement:
Smart cities demand an intelligent distributed traffic management system where each intersection functions autonomously yet coordinates with nearby intersections to:
Optimize traffic flow dynamically
Synchronize signal states across intersections
Prioritize emergency vehicle routing
Prevent gridlocks
Achieve system-wide consistency and fault tolerance
Tasks

Task	**Module no**	**Task Statement**	**Topics to be highlighted**	**CO**	**Bloom's Level**	**References**
1	1	Design distributed EHR architecture connecting hospitals and labs	Introduction to DS Framework	CO1	Create	EHR Architecture – IEEE
2	2	Develop REST/gRPC APIs for secure record exchange	RPC	CO2, CO3	Apply	FHIR APIs
3	2	Implement RPC/RMI for patient record access	RMI	CO3	Apply	Java RMI Docs
4	3	Apply clock sync for patient log timestamps	Clock Synchronization	CO3	Apply	Clock Sync – Neso
5	3	Implement Maekawa's mutual exclusion algorithm for concurrent edits	Mutual Exclusion	CO4	Apply	Maekawa Algorithm – GFG
6	3	Simulate and resolve deadlocks in record updates	Deadlocks	CO4	Analyze	Deadlock Detection – GFG
7	4	Apply load balancing for patient query handling	Load Balancing	CO4	Apply	LB Strategies – Kemp
8	5	Implement quorum-based consistency for record synchronization	Consistency Models	CO5	Evaluate	Quorum Consensus – Raft
9	6	Use Redis caching for frequent queries and replicate data	Caching and Replication	CO5	Apply	Redis + MongoDB Sharding
10	6	Deploy EHR system using Docker Compose/Kubernetes	Dockers	CO6	Create	Docker Compose Docs

10 Uncovering syllabus with different Activities.

No.	Type of the Activity	Activities	Number of beneficiaries	Other Details – guest profile, feedback, mark sheet, report
1	**Experiential learning/Interaction with Outside World**	1- Guest Lectures by Industry Expert	235	
		2- Workshops		
		3- Mini Project	235	
		4- Industrial Visit		
		5- Any other activity		
2		6- Poster Presentation	235	

	Collaborative & Group Activity	7- Minute Papers	235	
		8- Students Seminars	235	
		9- Students Debates		
		10- Panel Discussion / Mock GD		
		11- Mock Interview		
		12- Any other activity		
3	**Co-Curricular Activity**	13- Informative videos (NPTEL/YouTube /TEDx/ MIT OW/edX)	235	
		14- Lecture Capture Usage	235	
		15- Any other activity		
4	**Tests & Assessments**	16- Class Tests/ Weekly Tests		
		17- Pop Quiz	235	
		18- Mobile App Based Quiz	235	
		19- Open Book Test	235	
		20- Take Home Test		
		21- Any other activity		

11 AAP

No.	Programme	Course	Uploaded on V-refer	Date
1	Computer Engineering	Distributed Systems	Yes	6/7/25

12 Lecture Guide

No.	Programme	Course	Uploaded on V-refer	Date
1	Computer Engineering	Distributed Systems	Yes	6/7/25

*** Do not delete any activity. Give details for planned events. Write ‘NA’ for activity Not Planned.**

Consolidated Academic Administration Plan Prepared by (mention all theory teaching faculty names with signature)

Please write below your name and sign with date of the external cluster mentor meeting

Prof Amit K. Nerurkar Faculty 1	Faculty 2	Faculty 3

Vikram Babar Mr. Vikram Babar External Industry Mentor	Dr. Vinaya Sawant External Academic Mentor	Prof Amit K. Nerurkar VIT Cluster Mentor	Program HOD

Annexure:

Learning Outcomes: The Learner will:	**Assessment Criteria:** The Learner can:	**Evaluated under ISA/MSE/ESE/LAB**
1.1 Describe the fundamental characteristics, goals, and issues of distributed systems. (P.I.: 1.4.1, 2.1.2) (CO1, CO2)	• The student can define distributed systems and their architecture. • The student can list key characteristics like scalability and fault tolerance. • The student can state major goals such as transparency and performance. • The student can identify core issues like synchronization and latency. • The student can classify examples based on goals and challenges.	MSE MSE ISA ESE ESE
1.2 Identify and distinguish various distributed computing models such as minicomputer, workstation, processor pool, and hybrid models. (P.I.: 2.2.1, 2.2.2) (CO1)	• The student can define distributed computing models and their importance. • The student can describe the minicomputer model and its setup. • The student can distinguish the workstation model and its independence. • The student can explain the processor pool model's shared resource usage. • The student can differentiate the hybrid model combining multiple approaches. • The student can compare models based on scalability and use cases.	MSE/ESE/ISA MSE MSE MSE ISA ISA
1.3 Explain the role of hardware and software components in the architecture of distributed systems. (P.I.: 2.2.2, 1.4.1) (CO1, CO3)	• The student can list key hardware components in distributed systems. • The student can identify core software components like middleware and OS. • The student can explain hardware–software interaction in system design. • The student can match components to their specific functions. • The student can analyze how choices impact performance and architecture.	MSE/ESE MSE/ESE MSE/ESE MSE/ESE/ISA MSE/ESE
1.4 Apply knowledge of middleware and APIs (e.g., REST, gRPC) in enabling communication and	• The student can define middleware and APIs used in distributed systems.	LAB LAB

Learning Outcomes: The Learner will:	**Assessment Criteria:** The Learner can:	**Evaluated under ISA/MSE/ESE/LAB**
coordination within distributed environments. (P.I.: 5.1.1, 1.4.1) (CO2, CO3)	• The student can explain REST and gRPC for service communication. • The student can describe how middleware enables coordination. • The student can apply APIs for inter-process communication. • The student can select suitable APIs based on system needs.	LAB LAB LAB
1.5 Demonstrate awareness of emerging middleware technologies and pursue self-learning in distributed computing. (P.I.: 12.2.1) (CO6)	• The student can identify recent middleware trends and tools. • The student can explore technologies like service mesh and event-driven platforms. • The student can relate new tools to distributed system needs. • The student demonstrates initiative in learning independently. • The student documents insights from self-directed exploration.	ISA ISA/LAB ISA ISA ISA
2.1 Explain the working and use-cases of Remote Procedure Call (RPC) and Remote Method Invocation (RMI) in distributed systems. (P.I.: 1.4.1, 2.2.2) (CO2, CO3)	• The student can define RPC and RMI in the context of distributed systems. • The student can describe how RPC enables procedure calls across networked systems. • The student can explain how RMI supports object-based communication in Java. • The student can differentiate between RPC and RMI in terms of architecture and use. • The student is able to identify real-world use cases where RPC or RMI is applied.	LAB/ESE/MSE LAB/ESE/MSE LAB/ESE/MSE LAB/ESE/MSE LAB/ESE/MSE
2.2 Analyze the causes and handling mechanisms for RPC failures in distributed environments. (P.I.: 2.1.2, 2.2.1) (CO2, CO4)	• The student can list common RPC failure causes. • The student can explain timeout and orphan issues. • The student can describe failure handling techniques. • The student can analyze fault-tolerant strategies. • The student can suggest solutions based on failure types.	ISA/MSE/ESE ISA/MSE/ESE ISA/MSE/ESE ISA/MSE/ESE ISA/MSE/ESE

Learning Outcomes: The Learner will:	**Assessment Criteria:** The Learner can:	**Evaluated under ISA/MSE/ESE/LAB**
2.3 Describe the need for message-oriented communication and the functioning of message brokers in distributed systems. (P.I.: 1.4.1, 2.2.2) (CO3)	• The student can define message-oriented communication. • The student can state its need in loosely coupled systems. • The student can describe the role of message brokers. • The student can explain how brokers handle message queues. • The student can relate brokers to real-world systems like Kafka or RabbitMQ.	ISA/MSE/ESE ISA/MSE/ESE ISA/MSE/ESE ISA/MSE/ESE/LAB
2.4 Explore modern middleware technologies like Kafka and RabbitMQ through self-directed learning. (P.I.: 12.2.1) (CO6)	• The student can identify features of Kafka and RabbitMQ. • The student can explore installation and usage independently. • The student can compare their use in real-world scenarios. • The student can document learnings through self-study. • The student shows initiative in exploring modern tools.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB
3.1 Describe the need for and methods of physical and logical clock synchronization in distributed environments. (P.I.: 1.4.1, 2.1.2) (CO1, CO2)	• The student can explain why clock synchronization is needed. • The student can differentiate between physical and logical clocks. • The student can describe methods like NTP and Cristian's algorithm. • The student can explain Lamport and vector clocks. • The student can relate synchronization methods to real-world scenarios.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB
3.2 Apply Lamport's algorithm to implement logical clocks and establish event ordering in distributed systems. (P.I.: 2.3.1, 4.1.2) (CO3)	• The student can describe Lamport's logical clock concept. • The student can assign timestamps to events using the algorithm. • The student can determine event ordering from timestamps. • The student can apply the algorithm in example scenarios.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB

Learning Outcomes: The Learner will:	**Assessment Criteria:** The Learner can:	**Evaluated under ISA/MSE/ESE/LAB**
	• The student can analyze causality and concurrency of events.	ISA/MSE
3.3 Explain and compare election algorithms such as the Bully algorithm for coordinator selection. (P.I.: 2.1.2, 2.2.1) (CO2)	• The student can define the purpose of election algorithms. • The student can explain steps of the Bully algorithm. • The student can compare Bully with other election methods. • The student can analyze efficiency and fault tolerance. • The student can apply election logic to select a coordinator.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE ISA/MSE/ESE/LAB
3.4 Apply mutual exclusion algorithms including Lamport, Ricart–Agrawala, Maekawa, and token-based methods for process synchronization. (P.I.: 3.1.1, 4.1.2) (CO3, CO4)	• The student can explain the need for mutual exclusion. • The student can describe each algorithm's working. • The student can apply algorithms to synchronize processes. • The student can compare algorithms on message complexity. • The student can select suitable methods for given scenarios.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE ISA/MSE/ESE
3.5 Describe strategies for deadlock management in distributed systems, including avoidance, detection, and prevention. (P.I.: 2.3.1, 1.4.1) (CO2, CO4)	• The student can define deadlock in distributed systems. • The student can describe avoidance techniques like wait-die and wound-wait. • The student can explain detection using wait-for graphs. • The student can list prevention strategies like resource ordering. • The student can compare approaches based on system constraints.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA
3.6 Explore distributed consensus mechanisms such as Paxos and Raft through self-directed learning. (P.I.: 12.2.1) (CO6)	• The student can identify the need for consensus in distributed systems. • The student can describe basics of Paxos and Raft. • The student can compare their approaches to consensus.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA ISA/MSE/ESE/LAB

Learning Outcomes: The Learner will:	**Assessment Criteria:** The Learner can:	**Evaluated under ISA/MSE/ESE/LAB**
	• The student can explore real-world use cases independently. • The student demonstrates initiative in self-learning advanced topics.	ISA/MSE/ESE/LAB
4.1 Describe key concepts of distributed scheduling such as load balancing, load sharing, task assignment, and process migration. (P.I.: 1.4.1, 2.1.2) (CO2, CO4)	• The student can define distributed scheduling and its importance. • The student can explain load balancing and load sharing strategies. • The student can describe task assignment methods. • • The student can explain process migration with examples. • The student can relate these concepts to real-world scheduling systems.	ISA/MSE ISA/MSE ISA/MSE ISA/MSE ISA/MSE
4.2 Analyze different strategies for resource allocation and task management in a distributed environment. (P.I.: 2.2.1, 3.2.1) (CO4)	• The student can identify resource allocation challenges in distributed systems. • The student can describe static and dynamic allocation strategies. • The student can analyze task management approaches. • The student can evaluate strategies based on efficiency and fairness. • The student can suggest appropriate strategies for given scenarios.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE ISA/MSE/ESE/LAB ISA
4.3 Apply task migration techniques to improve system efficiency and reduce bottlenecks. (P.I.: 3.1.1, 4.1.2) (CO3, CO4)	• The student can define task migration and its need. • The student can describe different types of migration (preemptive, non-preemptive). • The student can apply migration techniques in example scenarios. • The student can analyze impact on load distribution and performance. • The student can identify when task migration improves efficiency.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB
4.4 Demonstrate understanding of containerization and orchestration tools like Docker and Kubernetes	• The student can define containerization and orchestration.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB

Learning Outcomes: The Learner will:	**Assessment Criteria:** The Learner can:	**Evaluated under ISA/MSE/ESE/LAB**
through self-learning. (P.I.: 12.2.1) (CO5, CO6)	• The student can describe key features of Docker and Kubernetes. • The student can explore tool usage through hands-on self-study. • The student can relate tools to distributed deployment scenarios. • The student shows initiative in learning emerging technologies.	ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB ISA/MSE/ESE/LAB
5.1 Explain the CAP theorem and distinguish between BASE and ACID models in the context of distributed systems. (P.I.: 1.4.1, 2.1.2) (CO1, CO4)	• The student can state the CAP theorem and its components. • The student can explain trade-offs between consistency, availability, and partition tolerance. • The student can describe ACID and BASE properties. • The student can differentiate BASE from ACID with examples. • The student can relate models to distributed database design.	ISA/MSE/ESE/LAB ISA ISA ISA ISA
5.2 Analyze data-centric and client-centric consistency models for managing distributed data. (P.I.: 2.2.1, 2.2.2) (CO4)	• The student can define data-centric and client-centric consistency. • The student can describe models like linearizability, causal, and eventual consistency. • The student can explain client-centric models like monotonic reads/writes. • The student can compare consistency models with examples. • The student can analyze which model suits a given application.	ISA ISA ISA ISA ISA
5.3 Evaluate various replication strategies and sharding techniques for scalability and fault tolerance. (P.I.: 3.2.1, 4.3.1) (CO4, CO5)	• The student can define replication and sharding in distributed systems. • The student can describe replication types: synchronous, asynchronous, and quorum-based. • The student can explain horizontal and vertical sharding.	ISA ISA ISA/ESE ISA/ESE

Learning Outcomes: The Learner will:	**Assessment Criteria:** The Learner can:	**Evaluated under ISA/MSE/ESE/LAB**
	• The student can evaluate trade-offs in scalability and reliability. • The student can select suitable strategies for distributed data systems.	ISA/ESE
5.4 Apply knowledge of distributed caching systems like Redis for improving system performance. (P.I.: 5.1.1, 5.2.1) (CO3, CO5)	• The student can define the role of caching in distributed systems. • The student can describe features and architecture of Redis. • The student can apply caching to reduce latency and load. • The student can analyze scenarios where caching is beneficial. • The student can implement basic Redis-based caching solutions.	ISA/ESE ISA/ESE ISA/ESE ISA/ESE ISA/ESE
5.5 Demonstrate self-learning by exploring advanced caching use cases and tools. (P.I.: 12.2.1) (CO6)	• The student can research cutting-edge caching patterns (e.g., CDN edge, write-through, write-behind). • The student can investigate tools beyond Redis, such as Memcached and Hazelcast. • The student can experiment with tuning cache eviction and persistence options. • The student can document performance gains from chosen configurations. • The student shows initiative by sharing insights via reports or presentations	ISA/ESE ISA/ESE ISA/ESE ISA/ESE ISA/ESE
6.1 Describe the concept and motivation behind Distributed Shared Memory (DSM) in distributed systems. (P.I.: 1.4.1, 2.1.2) (CO1)	• The student can define Distributed Shared Memory (DSM). • The student can explain why DSM is used in distributed systems. • The student can describe how DSM abstracts memory across nodes. • The student can relate DSM to ease of programming and transparency. • The student can identify basic DSM use cases and benefits.	ISA/ESE ISA/ESE ISA/ESE ISA/ESE ISA/ESE
6.2 Explain the general architecture, replacement strategies, and issues	• The student can describe the basic architecture of DSM systems.	ISA/ESE

Learning Outcomes: The Learner will:	**Assessment Criteria:** The Learner can:	**Evaluated under ISA/MSE/ESE/LAB**
such as thrashing in DSM systems. (P.I.: 2.2.1, 3.2.1) (CO4, CO5)	• The student can explain common page replacement strategies. • The student can define and identify causes of thrashing. • The student can analyze how strategies affect performance. • The student can suggest ways to mitigate DSM-related issues.	ISA/ESE ISA/ESE ISA/ESE ISA/ESE
6.3 Analyze the design principles and components of distributed file systems like GFS and HDFS. (P.I.: 4.3.1, 3.2.1) (CO4, CO5)	• The student can describe the architecture of GFS and HDFS. • The student can identify key components like NameNode and DataNode. • The student can explain design principles such as fault tolerance and scalability. • The student can compare features of GFS and HDFS. • The student can analyze use cases and performance trade-offs.	ISA/ESE ISA/ESE ISA/ESE ISA/ESE ISA/ESE
6.4 Apply knowledge of distributed memory and storage systems to propose scalable architectures using real-world case studies. (P.I.: 5.2.1, 12.2.1) (CO5)	• The student can identify components of scalable distributed architectures. • The student can apply concepts of DSM and DFS in real-world contexts. • The student can analyze case studies like Netflix, Google, or Facebook. • The student can propose scalable solutions using appropriate tools. • The student demonstrates applied understanding through self-directed exploration.	ISA/ESE ISA/ESE ISA/ESE ISA/ESE ISA/ESE
6.5 Explore DSM case studies such as Ivy and TreadMarks to understand practical implementations. (P.I.: 12.2.1) (CO5, CO6)	• The student can describe the architecture of Ivy and TreadMarks. • The student can explore how DSM concepts are applied in practice. • The student can compare design choices in both systems. • The student can relate learnings to real-world distributed memory needs.	ISA/ESE/LAB ISA/ESE ISA/ESE ISA/ESE

Learning Outcomes: The Learner will:	**Assessment Criteria:** The Learner can:	**Evaluated under ISA/MSE/ESE/LAB**
	• The student shows initiative in studying practical DSM implementations.	ISA/ESE