

Course Code	UCA20D01J	Course Name	WEB DEVELOPMENT USING NODE JS AND MONGO	Course Category	D	Discipline Specific Elective Course	L	T	P	C
							4	0	4	6

Pre-requisite Courses	Nil	Co-requisite Courses	Nil	Progressive Courses	Nil
Course Offering Department	Computer Applications	Data Book / Codes/Standards	Nil		

Course Learning Rationale (CLR):	The purpose of learning this course is to:	Learning	Program Learning Outcomes (PLO)
----------------------------------	--	----------	---------------------------------

CLR-1 :	Full Stack Development using Node.js	1	2	3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CLR-2 :	Design social media websites,music players and mini games via scripting																		
CLR-3 :	Building own modules																		
CLR-4 :	Understanding Node Package Manager																		
CLR-5 :	Interface with Mongo DB																		
CLR-6 :	Master NoSQL database																		

Course Learning Outcomes (CLO):	At the end of this course, learners will be able to:	Level of Thinking (Bloom)	Expected Proficiency (%)	Expected Attainment (%)	Fundamental Knowledge	Application of Concepts	Link with Related Disciplines	Procedural Knowledge	Skills in Specialization	Ability to Utilize Knowledge	Skills in Modeling	Analyze, Interpret Data	Investigative Skills	Problem Solving Skills	Communication Skills	Analytical Skills	ICT Skills	Professional Behavior	Life Long Learning
CLO-1 :	Install Node.js	2	80	70	H	H	H	H	H	M	L	M	H	M	-	H	H	H	M
CLO-2 :	Create basic web applications using Node.js	3	85	75	H	H	H	H	H	M	L	M	H	M	-	H	H	H	M
CLO-3 :	Build HTTP server using core modules	3	75	70	H	H	H	H	H	M	L	M	H	M	-	H	H	H	M
CLO-4 :	Use stream IO to efficiently serve the web page	3	85	80	H	H	H	H	H	M	L	M	H	M	-	H	H	H	M
CLO-5 :	Create Modules to organize server	3	85	75	H	H	H	H	H	M	L	M	H	M	-	H	H	H	M
CLO-6 :	Quering and performing CRUD on Mongo DB	3	80	70	H	H	H	H	H	M	L	M	H	M	-	H	H	H	M

Duration (hour)	24	24	24	24	24
S-1	SLO-1	Need of Scripting Language	Array Methods :indexOf, join, lasIndexOf, toString	Add HTTP header	Streams – Reading a Stream
	SLO-2	Difference between client and server side scripting	Array Methods : reduce, reverse, slice, some, sort	Example programs	Stream – Writing to a stream
S-2	SLO-1	Script tag in HTML	Function Definition	Read the Query String	Piping the Stream
	SLO-2	Java Script declaration	Function Parameters	Split the Query String	Chaining the Streams
S-3	SLO-1	Output printing – document. Write, innerHTML	Calling a Function	Node.js URL Module	Node.js as a File Server

						ii) Query an array
	SLO-2	window .alert, console.log	Return Statements	Node.js File Server	Create Files, Reading Files	ii) Query an array of nested documents iv) Geospatial Queries Query Operation Examples
S-4	SLO-1	Java script statements	Nested Functions	Node.js – NPM Package	Delete Files	Update Operation: updateOne(), updateMany()
	SLO-2	Comments and Variables	Example Programs	Downloading and Using a Package	Update and rename files	replaceOne(), findAndModify() Update operation :Examples
S 5-8	SLO-1	Lab 1 – Java Script Input and Output	Lab 4 – Functions	Lab 7 –Query String	Lab 10 – Streams and Files	Lab :Working with CURD operations Insert and Query
	SLO-2					
S-9	SLO-1	Java script Operators -Logical, Bitwise	Web stacks introduction	Callback – Blocking code example	Creating a Upload Form	Delete Operation: deleteMany(), deleteOne()
	SLO-2	Arithmetic and Assignment operators	LAMP, LEMP, MEAN, MERN	Callback – Non- Blocking code example	Parse the uploaded files	iii)findOneAndDelete() Delete operation Examples
S-10	SLO-1	Java Script Datatypes - numeric	Difference between php and java script	Event Driven Programming	Save the files	Operation on Mongodb Data: projection
	SLO-2	Java Script Datatypes – non numeric	Node introduction and evolution	Working of node Application	Display the uploaded files	Limiting Records Sorting Records
S-11	SLO-1	Conditional statements	Installing node.js and npm in windows	Node Even emitter class	Nodemailer Modules	Indexes in Mongodb, default _id index
	SLO-2	If else statements	Installing node.js and npm in Linux	add Listener(), on(), once()	Sending a email	Creating and Index createIndex method
S-12	SLO-1	Switch statements	Built in modules in node.js – http, https	removeListener(), removeAllListeners()	Multiple Receivers	Single Field, Compound, Multikey
	SLO-2	Iteration statements	Built in modules in node.js – querystring, readline	setMaxListeners(), listeners()	Sending HTML	Geospatial, text Index, Hashed Index
S 13-16	SLO-1	Lab 2 – Java Script Operators and Conditions	Lab 5 – Installing Node.js	Lab 8 – Event Driver classes	Lab 11 – Sending Mail	Lab :Working with CRUD operations Update and Delete
	SLO-2					
S-17	SLO-1	Loop Controls – for loop	Include modules	Creating Buffers, writing to buffers	Mongodb Datatypes: i)Integer ii)Boolean iii)Double iv)String v)Arrays vi)Object vii)NULL viii)Regular expression ix)Timestamp x)Date xi)Object ID	Properties of Index i)Unique Indexes ii)Partial Indexes
	SLO-2	While loop	Writing first sample application	Reading from Buffers	Installing Mongo DB in Windows, Linux and Mac Operating Systems	iii)Sparse Indexes iv)TTL Indexes
S-18	SLO-1	Do while Loop	Creating own modules	Converting Buffer to JSON	Installing and Working with MongoDB interfaces: i)Mongo Shell, ii)Mongo Compass	Aggregation in Mongodb: i)aggregate() method Aggregate expressions: i) \$sum ii) \$avg iii) \$min iv) \$max

	SLO-2	For each loop	Including your own module	Concatenate Buffer	Introduction to entities of MongoDB: i)Databases i)Collections and iii)Documents	v) \$push vi) \$addToSet vii) \$first viii) \$last
S-19	SLO-1	Arrays Introduction and declaring	Node.js – REPL Terminal – Read, Eval	Compare, Copy Buffer	Database: i)createDatabase() method with example	Mongodb Backup: Export/Import data backup using shell i)mongodump ii)mongorestore
	SLO-2	Accessing arrays	Node.js – REPL Terminal – Print, Loop	Slice Buffer and Buffer Length	ii)dropDatabase() method with example	Mongodb Backup: Export/Import data backup using Mongo Compass
S-20	SLO-1	Array Properties : index, input length, prototype	Node.js as built in HTTP module	isEncoding(), isBuffer()	Collections: i)createCollection() method with example	Monitoring Deployment using Mongodb: i)mongostat, mongotop
	SLO-2	Array Methods :concat, every, forEach	Node.js as a Web Server	byteLength	ii)dropCollection() method with example	iii)serverStatus, dbStats, collStats
S 21- 24	SLO-1	Lab 3 - Looping Statements	Lab 6 - Running sample application using node.js	Lab 9 - Buffers	Lab 12 – Working with MongoDB – create,drop,working with Collections	Lab: i)Creating different types of indexes ii)Aggregate data using different Aggregate expressions iii)Perform Mongodb data Export and Import using shell as well as mongo compass. iv)Working with mongo deployment commands
	SLO-2					

Learning Resources	Online Official Documentation 1. NodeJS v13 10.1 Docs: https://nodejs.org/latest-v13.x/api/documentation.html 2. MongoDB: https://docs.mongodb.com/manual/tutorial/getting-started/
--------------------	---

Learning Assessment											
Level	Bloom's Level of Thinking	Continuous Learning Assessment (50% weightage)								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (10%)		CLA – 3 (20%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember	20%	20%	15%	15%	15%	15%	15%	15%	15%	15%
	Understand										
Level 2	Apply	20%	20%	20%	20%	20%	20%	20%	20%	20%	20%
	Analyze										
Level 3	Evaluate	10%	10%	15%	15%	15%	15%	15%	15%	15%	15%
	Create										
	Total	100 %		100 %		100 %		100 %		100 %	

CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

Course Designers		
Experts from Industry	Experts from Higher Technical Institutions	Internal Experts
Mr.G.Muruganandam, Group Project Manager, HCL Technologies, Chennai	Dr.S.Gopinathan, Professor, University of Madras, Chennai	Mrs. Ramla, SRM IST
Mr.M. Hemachandar, Tech Lead, Wipro Limited, Chennai		Mrs. Anita Jasmine, SRM IST

