GANPAT UNIVERSITY

FACULTY OF ENGINEERING & TECHNOLOGY

Programme	Diploma in Computer Engineering / Information Technology				
Semester	II		Version	1.0.0.0	
Effective from Academic Year	2026-27		Effective for the batch Admitted in	JULY 2025	
Course code	1CEIT2102	Course Name	Version Control System & JavaScript		

I.TEACHING-LEARNING AND ASSESSMENT SCHEME

Course Type	Course Code	Learning Scheme						Assessment Scheme										
		Actual Contact Hrs./Week			SLH	NLH	Credits	Theory				Practical				Based on SL		Total Marks
		CL	TL	LL				FA-TH	SA-TH	TOTAL		FA-PR	SA-PR	TOTAL		SLA		
								MAX	MAX	MAX	MIN	MAX	MAX	MAX	MIN	MAX	MIN	
SEC	1CEIT2102	-	-	4	2	6	3	-	-	-	-	60	40	100	40	20	8	120

Abbreviation:	**CL**- Classroom Learning	**TL**- Tutorial Learning	**LL**-Laboratory Learning
	SLH-Self Learning Hours	**NLH**-Notional Learning Hours	**SLA**-Self Learning Assessment
	FA-Formative Assessment (Term work+Mid Sem Exam+ Attendance)	**SA**-Summative Assessment	

II. PRE-REQUISITES

Basic knowledge of programming and file systems.

III.INDUSTRY /EMPLOYER EXPECTED OUTCOMES

Collaborate effectively on projects using GitHub and Contribute to open-source projects and Demonstrate industry-relevant skills in JavaScript tooling, code quality (linting), and team-based development practices.

IV. COURSE LEARNING OUTCOMES

At the end of the course, students will be able to achieve the following course learning outcomes:

CO1. Develop JavaScript code using arrays, objects, DOM, and events for web interactivity.

CO2. Implement asynchronous programming using Promises and Async/Await.

CO3. Apply functional programming principles and modular JavaScript concepts, manage client-side storage and JavaScript tooling using NPM and linters.

CO4. Collaborate on code using Git and GitHub including version control.

CO5. Understand project workflows and work on mini projects.

VI. LABORATORY LEARNING OUTCOME AND ALIGNED PRACTICAL

Sr	Practical/Laboratory Learning Outcome (LLO)	Practical Titles	Relevant COs
1	**LLO 1.1**: Perform basic Git operations locally and connect to GitHub.	**Git Fundamentals and GitHub Integration** Initialize a repo, commit changes, and push to GitHub.	CO4
2	**LLO 2.1**: Collaborate using GitHub features like branches, pull requests, and issues.	**Collaborative Development with GitHub** Create a team repo, work on different branches, and submit a pull request.	CO4
3	**LLO 3.1**: Apply project planning and teamwork on a full-stack mini project.	**Team-Based Full-Stack Project Development**	CO5

		Design and build a simple notes/weather app with GitHub collaboration.	
4	**LLO 4.1**: Deploy a JavaScript-based static site using GitHub Pages.	**Static Site Deployment** Publish the final mini project online using GitHub Pages.	CO5
5	**LLO 5.1**: Understand and implement JavaScript arrays, objects, and DOM manipulation.	**Dynamic Web Page Manipulation** Write JS code to dynamically update a webpage using DOM elements and array data.	CO1
6	**LLO 6.1**: Handle form events and validate inputs using JavaScript.	**Interactive Form Validation** Create a registration form with real-time validation using JS event listeners.	CO1
7	**LLO 7.1**: Demonstrate use of timers and complex user interactions.	**Advanced User Interface Components** Build a stopwatch/timer or image slider using JavaScript setTimeout and setInterval.	CO1
8	**LLO 8.1**: Use Promises and Async/Await for asynchronous code execution.	**API Integration and Asynchronous Programming** Fetch JSON data from a public API and render it in a webpage using fetch() & async.	CO2
9	**LLO 9.1**: Apply functional programming concepts like higher-order functions and closures.	**Functional Programming Implementation** Create a mini JS library using higher-order functions and closures.	CO3
10	**LLO 10.1**: Modularize JavaScript code using ES6 module syntax.	**ES6 Module System Implementation** Split a JS app into modules using import and export.	CO3
11	**LLO 11.1**: Store and retrieve data using localStorage and sessionStorage.	**Client-Side Data Persistence** Build a to-do list app with persistent data using localStorage.	CO3
12	**LLO 12.1**: Configure JavaScript linting and formatting tools (ESLint, Prettier).	**Code Quality and Development Tools Setup** Set up a JS project with ESLint and Prettier and demonstrate auto-formatting.	CO3

VII. SUGGESTED MICRO PROJECT / ASSIGNMENTS / ACTIVITIES FOR SELF LEARNING / SKILL DEVELOPMENT (SELF LEARNING)

- To-Do List App (Basic CRUD App)
- Student Result Management System (Frontend UI)
- Simple Game (e.g., Tic Tac Toe or Snake Game)

VIII. LIST OF INSTRUMENTS / EQUIPMENT / TRAINER BOARD

- Desktop or laptop computers with high-speed internet access.
- Code Editor & Browser: Visual Studio Code for writing code, Firefox/Chrome browsers for testing web pages
- Version Control: Git software for tracking code changes, GitHub account for storing and sharing projects online
- JavaScript Runtime: Node.js for running JavaScript programs, npm for installing additional code packages
- Web Hosting & Testing: GitHub Pages for publishing websites online, Postman for testing web APIs and data connections

VIII. LIST OF REFERENCE BOOKS

Sr. No	Title	Author	Publication
1	Eloquent JavaScript: A Modern Introduction to Programming	Marijn Haverbeke	No Starch Press, 3rd Edition
2	Pro Git (Free online book)	Scott Chacon and Ben Straub	Apress / Git SCM, 2nd Edition
3	Learning JavaScript Design Patterns	Addy Osmani	O'Reilly Media, Updated 2021

IX. LINK OF LEARNING WEB RESOURCE

1	https://medium.com/@wamna083/a-deep-dive-into-javascript-core-concepts-and-features-cd2d8edb3b8d
2	https://medium.com/@lydiahallie/javascript-visualized-promises-async-await-a3f1aad8a943
3	https://www.tutorialspoint.com/git/index.htm
4	https://www.w3schools.com/git/
5	https://www.geeksforgeeks.org/git-tutorial/
6	https://www.youtube.com/watch?v=RGOj5yH7evk

X. SUGGESTED WEIGHTAGE TO LEARNING EFFORTS & ASSESSMENT PURPOSE

Sr. No	Practical title	Aligned COs	Learning hours	R level	U level	A level	Total marks
1	Git Fundamentals and GitHub Integration	CO4	4	3	2	0	5
2	Collaborative Development with GitHub	CO4	4	2	2	1	5
3	Team-Based Full-Stack Project Development	CO5	8	0	2	4	6
4	Static Site Deployment	CO5	4	1	1	2	4
5	Dynamic Web Page Manipulation	CO1	6	1	2	2	5
6	Interactive Form Validation	CO1	5	1	1	3	5
7	Advanced User Interface Components	CO1	5	0	2	3	5

8	API Integration and Asynchronous Programming	CO2	6	0	2	4	6
9	Functional Programming Implementation	CO3	5	0	2	3	5
10	ES6 Module System Implementation	CO3	4	1	2	2	5
11	Client-Side Data Persistence	CO3	5	0	1	4	5
12	Code Quality and Development Tools Setup	CO3	4	2	2	0	4
		Grand Total:	60	11	21	28	60

XII. COs AND POs AND PSOs MAPPING

Course Outcome (Cos)	**Programme Outcomes (POs)**							**Programme Specific Outcomes (PSOs)**		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	0	0	0	3	3	2	0	3	0	3
CO2	0	3	3	3	3	0	2	3	0	3
CO3	3	2	3	3	2	2	0	3	0	2
CO4	3	2	3	2	0	2	0	3	0	0
CO5	3	2	3	3	0	3	0	3	0	2

Legends:- *3-High; 2-Moderate/Medium; 1-Slight/Low; 0-None*