Lab	Type	Practical
I.	Javascript Revision	
1	 A A A B C	**Perform JavaScript Programs on Date Handling, Array Manipulation, and Utility Functions** 1. Write a function that formats a given Date into different formats like: YYYY/MM/DD, DD-MM-YYYY, or Month DD, YYYY (e.g., November 12, 2025) 2. Write a JS function to reverse an array using loop or reduce( ). (Without using reverse( )). 3. Create a random 6-digit numeric OTP or alphanumeric password using Math.random(). 4. Take a date input from user and display which day (e.g., Monday, Tuesday). 5. Input a future date and show live countdown in Days : Hours : Minutes : Seconds.
2	 A A A B C	**Perform JavaScript String Manipulation Programs** 1. Use object/map to count occurrences of each character in a string. 2. Use regex or conditional logic to count vowels and consonants. 3. Write a JS function to find and return the longest word and its length. 4. Write a JS function to find First Non-Repeating Character (B). 5. Write a JS function to compress String (Run-Length Encoding) Ex: "aaabbcccc" → "a3b2c4"
3	 A A A B C	**Demonstrate Core concepts of ES6** 1. Create a function using Rest Parameter to calculate the product of any number of arguments. 2. Convert an array of user objects into two arrays: names[] and ages[] using Destructuring. 3. Create a function to count the frequency of words in a sentence using Map. 4. Write a function that returns the longest word in a string using ES6 features. 5. Write an async function that fetches 3 APIs in parallel and prints all results.
II.	NodeJS	
4	 A A A A A	**Demonstrate the use of Node Package Manage (NPM).** 1. installing npm 2. installing new package 3. installing package globally 4. updating package 5. Understanding package.json file.
5	 A A	**Demonstrate the use of core NodeJS modules (part-01)** 1. Demonstrate “path” core module in NodeJS. 2. Demonstrate “fs” core module in NodeJS.

6		**Demonstrate the use of core NodeJS modules (part-02)**
	A	• WAP in NodeJS to store the student details in text file.
	B	• WAP in NodeJS to copy the content of a file named abc.txt to xyz.txt
	B	• WAP in NodeJS to count number of words in a file
	B	• WAP in NodeJS to count total vowels in a file
	C	• WAP to read student details from the file named students.txt, student details are stored line by line with following comma seperated fields
		◦ StudentID
		◦ StudentName
		◦ StudentEnrollmentNumber
		◦ StudentMobileNumber
		◦ StudentDepartment
		◦ StudentSPI
	C	• WAP to read student detail specified as per previous program and filter the students with less than 5 SPI
7		**Demonstrate the use of core NodeJS modules (part-03)**
	A	• Demonstrate “child_process” core module in NodeJS.
	A	• Demonstrate the use of EventEmmiter in NodeJS.
8		**Demonstrate the use of http core module in NodeJS**
	A	• Create a hello world webapp using “http” core module in NodeJS.
	B	• Create a webapp with 5 pages like about, contact etc.. using “http” core module in NodeJS.
	C	• Create a webapp in NodeJS which reads files like about.txt, contact.txt and display it using http core module.
III.	**ExpressJS**	
9		**Demonstrate the basic ExpressJS web application**
	A	• Create a hello world webapp using ExpressJS.
	A	• Create a webapp with 5 pages like about, contact etc.. using ExpressJS.
	A	• Create a webapp in NodeJS which reads files like about.txt, contact.txt and display it using http core module
10		**Create middleware in ExpressJS**
	A	• Demonstrate the use of middleware in Express.
	A	• Demonstrate the use of static middleware in Express.

11		**Setup MongoDB and Create a Database**
	A	Install MongoDB, MongoDBCompass and Mongoose library
	A	Setup documents in MongoDB.
	A	Demonstrate the use mongoose functions.
	A	Create a Database using MongoDBCompass for faculty.
	A	Create a Database using MongoDBCompass for student.
	A	Create a Database using MongoDBCompass for product.
12		**Create a restful CRUD API**
	A	Create a restful CRUD API using NodeJS, Express and MongoDB for faculty (updateById,deleteById).
13		**Create a restful CRUD API**
	A	Create a restful CRUD API using NodeJS, Express and MongoDB for student (updateById,deleteById).
14		**Create a restful CRUD API**
	A	Create a restful CRUD API using NodeJS, Express and MongoDB for product (updateById,deleteById).
IV.	**Introduction to ReactJS**	
15		**Setup ReactJS development environment**
	A	1. Setting up react environment.
	A	2. Hello world webapp using ReactJS.
	A	3. Demonstrate the use of JSX.
	B	4. WAP to create a simple class component in ReactJS.
	B	5. WAP to create a simple function component in ReactJS
16		**Create Hello World ReactJS app**
	A	• Create a function component in separate file and link with App.js
	B	• Create a class component in separate file and link with App.js
17		**Demonstrate props in ReactJS**
	A	• Demonstrate the ReactJS props.
	A	• Demonstrate the Event Handling in ReactJS.
	A	• WAP in ReactJS to display the element if it has attribute called isDisplay to be true (using conditional rendering)

18	 A B B C	**Demonstrate the use of map method in ReactJS** • Demonstrate the use of map method in ReactJS to display array. • Display Faculties stored in array using ReactJS. • Display Students stored in array using ReactJS. • Display Products stored in array using ReactJS
V.	**Routing and Hooks in ReactJS**	
19	 A A C	**Demonstrate Routing in ReactJS** • Implement Routing in ReactJS. • Develop basic website using 5 different component (pages) and implement Routing in it. (i.e. About, Contact etc…) • Develop full static website using 15 different component (pages) and implement Routing in it. (i.e. About, Contact etc…)
20	 A A	**Demonstrate the use of hooks in ReactJS** • Demonstrate useState hook in ReactJS. • Demonstrate useEffect hook in ReactJS
21	 A C	**Create GUI Calculator using ReactJS** • WAP to create a simple calculator using ReactJS. • WAP to create a scientific calculator using ReactJS.
22	 A B C	**Implement CRUD operation on Array in ReactJS** • WAP to do CRUD operation on products stored as array using ReactJS. • WAP to do CRUD operation on students stored as array using ReactJS. • WAP to do CRUD operation on faculties stored as array using ReactJS.
23	 A	**Perform CRUD operation on MockAPI using ReactJS** • Create a MockAPI online with following fields. ◦ FacultyID ◦ FacultyName ◦ FacultyExp ◦ FacultyImage
24	 A B C	**Perform CRUD operation on MockAPI using ReactJS** • Perform CRUD operation on MockAPI using ReactJS. (minimum 3 mock api) • Perform CRUD operation on MockAPI using ReactJS. (minimum 8 mock api) • Perform CRUD operation on MockAPI using ReactJS. (minimum 15 mock api)

VI.	**Authentication and best practices for API**	
25	 A A	**Demonstrate API Authentication Techniques and Create API Documentation** • Demonstrate API Authentication Techniques like OAuth, JWT etc.. • Create an API documentation for all previous practicals.
26	A	**Create a mini project for attendance management system (part-01)** Create and consume Restfull API using MongoDB, Express, ReactJS and NodeJS (MERN stack) for attendance management system.
27	A	**Create a mini project for attendance management system (part-02)** Create and consume Restfull API using MongoDB, Express, ReactJS and NodeJS (MERN stack) for attendance management system.
28	A	**Create a mini project for attendance management system (part-03)** Create and consume Restfull API using MongoDB, Express, ReactJS and NodeJS (MERN stack) for attendance management system.
29	 A B	**Create API Documentation** • Create an API documentation for attendance management systerm • Create an API documentation for all previous practicals.
30		**Project Evaluation**