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

5		**Demonstrate the use of core NodeJS modules**
	A	1. Demonstrate "path" core module in NodeJS.
	A	2. Demonstrate "fs" core module in NodeJS.
	B	3. Demonstrate "child_process" core module in NodeJS.
	C	4. WAP in NodeJS to store the student details in text file.
6		**Demonstrate the use of http core module in NodeJS.**
	A	1. Demonstrate the use of http core module in NodeJS.
	A	2. Create a hello world webapp using "http" core module in NodeJS.
	A	3. Create a webapp with 5 pages like about, contact etc.. using "http" core module in NodeJS.
	B	4. Create a webapp in NodeJS which reads files like about.txt, contact.txt and display it using http core module.
	C	**5.** Demonstrate the use of EventEmmiter in NodeJS.
7		**Setup MongoDB**
	A	1. Install MongoDB and MongoDBC.
	B	**2.** Setup documents in MongoDB.
8		**Setup documents in MongoDB**
	A	1. Install Mongoose library using NPM.
	A	2. Demonstrate the use mongoose functions.
	A	3. Create a Database using MongoDBCompass for faculty.
	B	4. Create a Database using MongoDBCompass for student.
	B	5. Create a Database using MongoDBCompass for product.
9		**Demonstrate the basic ExpressJS web application**
	A	1. Create a hello world webapp using ExpressJS.
	A	2. Create a webapp with 5 pages like about, contact etc.. using ExpressJS.
	B	**3.** 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	1. Demonstrate the use of middleware in Express.
	B	2. Demonstrate the use of static middleware in Express.
11		**Create a restful CRUD API**
	A	Create a restful CRUD API using NodeJS, Express and MongoDB for faculty (getAll,getById).

12		**Create a restful CRUD API**
	A	Create a restful CRUD API using NodeJS, Express and MongoDB for faculty (updateById,deleteById).

13		**Setup ReactJS development environment and demonstrate reactJS component**
	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

14		**Demonstrate components and its properties in ReactJS**
	A	1. Create a function component in separate file and link with App.js
	A	2. Demonstrate the ReactJS props.
	B	3. Demonstrate the Event Handling in ReactJS.
	C	4. WAP in ReactJS to display the element if it has attribute called isDisplay to be true (using conditional rendering).

15		**Demonstrate the use of map,filter and reduce methods in ReactJS to display array**
	A	1. Display Faculties stored in array using ReactJS.
	B	2. Display Students stored in array using ReactJS.
	B	3. Display Products stored in array using ReactJS which price are more then 20000.
	C	4. Display the total price of Products stored in array using reduce method.

16		**Implement Routing in ReactJS.**
	A	Develop basic website using 5 different component (pages) and implement Routing in it. (i.e. About, Contact etc…).

17		**Demonstrate Hooks in ReactJS**
	A	1. Demonstrate useState hook in ReactJS.
	B	2. Demonstrate useEffect hook in ReactJS.
	B	3. WAP to create a simple calculator using ReactJS.
	C	4. Demonstrate useParams hook in ReactJS to fetch parameters from url.

18		**WAP to do CRUD operation using ReactJS.**
	A	1. WAP to do CRUD operation on students stored as array using ReactJS.
	B	2. WAP to do CRUD operation on faculties stored as array using ReactJS.

19		**Implement CRUD operation on different MockAPIs using ReactJS.**
	A	Implement CRUD operation on different MockAPIs using ReactJS. (minimum 3 APIs)

20		**Create API using NodeJS, ExpressJS, MongoDB and consume it using ReactJS**
	A	1. Create reactJS component which serv getAll, GetById using API consumption.
	B	2. Create reactJS component which serv updateByID, deleteById using API consumption.

21		**Create basic component in Angular**
	A	1. Setup Environment for Angular.
	A	2. Create first project in Angular.
	B	3. Hello world webapp using Angular.
	B	4. Demonstrate how to create component in Angular.

22		**Demonstrate routing and binding in Angular**
	A	1. Demonstrate basic routing in Angular.
	A	2. Demonstrate the routing parameters in Angular.
	B	3. Demonstrate the Interpolation.
	B	4. Demonstrate the Property Binding.
	C	5. Demonstrate the Event Binding.
	C	**6.** Demonstrate Two-way Data Binding.

23		**Demonstrate directives and observable in Angular**
	A	1. Demonstrate built-in attribute directives in Angular.
	B	2. Demonstrate built-in structural directives in Angular.
	B	3. Demonstrate Observable in Angular.
	C	**4.** Demonstrate Dependency Injection in Angular.

24		**Design forms using Angular**
	A	1. Demonstrate form handling in Angular
	B	2. Demonstrate Template Driven forms in Angular.

25		**CRUD Operation using Angular**
	A	1. Create webapp using Angular to perform CRUD operation on students stored as array.
	B	2. Create webapp using Angular to perform CRUD operation on products stored as array.
	C	3. Create webapp using Angular to perform CRUD operation on faculties stored as array.

26	A	**Consume API created on MockAPI using Angular** Create a MockAPI online with following fields. • FacultyID • FacultyName • FacultyExp • FacultyImage Perform CRUD operation on MockAPI using Angular.
27	A	**Implement CRUD operation on different MockAPIs using Angular.** Implement CRUD operation on different MockAPIs using Angular. (minimum 2 APIs)
28	A B	**Create API using NodeJS, ExpressJS, MongoDB and consume it using AngularJS** 1. Createan angular component which serv getAll, GetById using API consumption. 2. Create an angular component which serv updateByID, deleteById using API consumption.
29		**React or Angular project development.**
30		**React or Angular project development.**