Program	Bachelor of Technology (BTech)	**Semester - 4**
Type of Course	Professional Core	
Prerequisite	Web Desiging	
Course Objective	Students will be able to understand modern web technologies based on Java Script.	

Teaching Scheme (Contact Hours)				**Examination Scheme**				
Lecture	**Tutorial**	**Practical**	**Credit**	**Theory Marks**		**Practical Marks**		**Total Marks**
				SEE (T)	**CIA (T)**	**SEE (P)**	**CIA (P)**	
4	0	4	6	40	30	20	10	100

__SEE__ - Semester End Examination, __CIA__ - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content — **T** - Teaching Hours | **W** - Weightage

Sr.	Topics	T	W
1	**NodeJS** Introduction to Node JS, Setup Development Environment, Node JS Modules, Node Package Manager, Creating Web Server, File System, Debugging Node JS Application, Events.	**9**	**20**
2	**ExpressJS** Express JS, Serving Static Resources, Database Connectivity, API using NodeJS and ExpressJS.	**10**	**20**
3	**Introduction to ReactJS** Introduction to ReactJS, Introducing JSX, Rendering Elements, Component and Props, State and Lifecyle, Handling Events, Conditional Rendering.	**10**	**20**
4	**Routing and Hooks in ReactJS** Routing, React CSS styling, Hooks, Forms, Consuming API in ReactJS.	**9**	**20**
5	**Authentication and best practices for API** Importance of API authentication, API authentication and authorization methods like basic, API key, TLS encryption, OAuth2.0, JWT, OIDC, best practices for creating and consuming API, creating API documentation.	**7**	**20**
	Total	**45**	**100**

Suggested Distribution Of Theory Marks Using Bloom's Taxonomy

Level	Remembrance	Understanding	Application	Analyze	Evaluate	Create
Weightage	10	45	45	0	0	0

NOTE : This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes

At the end of this course, students will be able to:	
CO1	**setup** NodeJS projects using Node Package Manager.
CO2	**implement** Restful APIs using ExpressJS.
CO3	**use** ReactJS to create Front-End.
CO4	**consume** API using ReactJS.
CO5	**understand** importance of API authentication and documenting API.

Reference Books

1. **Developing Web Application**
   By Ralph Moseley, M.T. Savaliya | Wiley India | Latest Edition
2. **Professional Node.js**
   By Pedro Teixeira | John Wiley & Sons
3. **Node.js in Action**
   By Alex Young | Dreamtech Press
4. **Learning React: Fundamental Web Development with react and redux**
   By Alex Banks | O'Reilly

List of Practical

1. Demonstrate the use of Node Package Manage (NPM).

Demonstrate the use of Node Package Manage (NPM).

- installing npm
- installing new package
- installing package globally
- updating package
- removing package

2. Understanding package.json file

Understanding package.json file and its fields

3. Demonstrate the use of core NodeJS modules (part-01)

- Demonstrate "path" core module in NodeJS.
- Demonstrate "fs" core module in NodeJS.

4. Demonstrate the use of core NodeJS modules (part-02)

- Demonstrate "child_process" core module in NodeJS.
- Demonstrate the use of EventEmmiter in NodeJS.

5. Demonstrate the use of core NodeJS modules (part-03)

- WAP in NodeJS to store the student details in text file.(A)
- 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 (B)
- WAP to read student details from the file named students.txt, student details are stored line by line with following comma seperated fields (C)
  - StudentID
  - StudentName
  - StudentEnrollmentNumber
  - StudentMobileNumber
  - StudentDepartment
  - StudentSPI
- WAP to read student detail specified as per previous program and filter the students with less than 5 SPI (C)

6. Demonstrate the use of http core module in NodeJS

- Create a hello world webapp using "http" core module in NodeJS. (A)
- Create a webapp with 5 pages like about, contact etc.. using "http" core module in NodeJS. (B)
- Create a webapp in NodeJS which reads files like about.txt, contact.txt and display it using http core module. (C)

7. Demonstrate the basic ExpressJS web application

- Create a hello world webapp using ExpressJS. (A)
- Create a webapp with 5 pages like about, contact etc.. using ExpressJS. (B)

	• Create a webapp in NodeJS which reads files like about.txt, contact.txt and display it using http core module. (C)
8.	**Create middleware in ExpressJS** • Demonstrate the use of middleware in Express. (A) • Demonstrate the use of static middleware in Express. (A)
9.	**Setup MongoDB** • Install MongoDB and MongoDBCompass • Setup documents in MongoDB.
10.	**Setup documents in MongoDB** • Install Mongoose library using NPM. • Demonstrate the use mongoose functions. • Create a Database using MongoDBCompass for faculty. • Create a Database using MongoDBCompass for student. • Create a Database using MongoDBCompass for product.
11.	**Create a restful CRUD API using NodeJS, Express and MongoDB for faculty.** Create a restful CRUD API using NodeJS, Express and MongoDB for faculty.
12.	**Create a restful CRUD API using NodeJS, Express and MongoDB for student and product.** Create a restful CRUD API using NodeJS, Express and MongoDB for student and product.
13.	**Setting up ReactJS development Environment** • Setting up react environment. (A) • Hello world webapp using ReactJS. (A) • Demonstrate the use of JSX. (A) • WAP to create a simple class component in ReactJS. (A) • WAP to create a simple function component in ReactJS. (A)
14.	**Create Hello World ReactJS app** • Create a function component in separate file and link with App.js (A) • Create a class component in separate file and link with App.js (B)
15.	**Demonstrate props in ReactJS** • 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) (A)
16.	**Demonstrate the use of map method in ReactJS** • Demonstrate the use of map method in ReactJS to display array. (A) • Display Faculties stored in array using ReactJS. (B) • Display Students stored in array using ReactJS. (B) • Display Products stored in array using ReactJS (C)
17.	**Demonstrate Routing in ReactJS** • Implement Routing in ReactJS. (A) • Develop basic website using 5 different component (pages) and implement Routing in it. (i.e. About, Contact etc...) (A) • Develop full static website using 15 different component (pages) and implement Routing in it. (i.e. About, Contact etc...) (C)
18.	**Demonstrate the use of hooks in ReactJS** • Demonstrate useState hook in ReactJS. (A) • Demonstrate useEffect hook in ReactJS (A)
19.	**Create GUI Calculator using ReactJS** • WAP to create a simple calculator using ReactJS. (A) • WAP to create a scientific calculator using ReactJS. (C)

20.	**Implement CRUD operation on Array in ReactJS** • WAP to do CRUD operation on products stored as array using ReactJS. (A) • WAP to do CRUD operation on students stored as array using ReactJS. (B) • WAP to do CRUD operation on faculties stored as array using ReactJS. (C)
21.	**Perform CRUD operation on MockAPI using ReactJS** • Create a MockAPI online with following fields. (A) ◦ FacultyID ◦ FacultyName ◦ FacultyExp ◦ FacultyImage • Perform CRUD operation on MockAPI using ReactJS. (minimum 3 mock api) (A) • Perform CRUD operation on MockAPI using ReactJS. (minimum 8 mock api) (B) • Perform CRUD operation on MockAPI using ReactJS. (minimum 15 mock api) (C)
22.	**Demonstrate API Authentication Techniques and Create API Documentation** • Demonstrate API Authentication Techniques like OAuth, JWT etc.. • Create an API documentation for all previous practicals.
23.	**Create a mini project for library management system (part-01)** Create and consume Restfull API using MongoDB, Express, ReactJS and NodeJS (MERN stack) for library management system.
24.	**Create a mini project for library management system (part-02)** Create and consume Restfull API using MongoDB, Express, ReactJS and NodeJS (MERN stack) for library management system.
25.	**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.
26.	**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.

Useful Links
1. https://reactjs.org/docs/getting-started.html 2. https://nodejs.org/en/docs/