

Program	Bachelor of Technology (BTech)	Semester - 4
Type of Course	-	
Prerequisite	Web Technology – I	
Course Objective	After completing the course students will be able to create and consume APIs using Angular, ReactJS and NodeJS.	

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	30	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	Node Package Manager Introduction to NPM, Setting up development environment, NodeJS Modules, installing new package with NPM, global install, uninstalling package, updating package, working with specific version of a package, package.json	8	20
2	NodeJS Introduction to NodeJS, Core modules in NodeJS, path module, child_process module, os module, url module, util module, creating webserver, file system, Debugging NodeJS application, Events	9	20
3	ExpressJS Introduction to ExpressJS, creating basic webserver using ExpressJS, basic routing, routing methods, route paths, route parameters, route handlers, middlewares, database connectivity, creating RestAPI.	9	20
4	Angular Introduction to Angular, Project Setup, Creating Components, Routing, Databinding, String interpolation, Pipes, Property Binding, Event Binding, Two-way data Binding, Directives, Observable, Dependency Injection, Forms in Angular, consuming Rest API using Angular.	10	20
5	ReactJS Introduction to ReactJS, Introducing JSX, Rendering Elements, functional and class component, lifecycle of component, react props, events, conditional rendering, routing, CSS styling in react, Hooks, Forms, Consuming API.	9	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:

C01	discuss node package manager.
C02	prepare basic webserver using NodeJS.
C03	implement api using NodeJS, Express and MongoDB.
C04	create single page application (SPA) using Angular.
C05	design front-end using ReactJS.

Reference Books

1.	Professional Node.js By Pedro Teixeira John Wiley & Sons
2.	Node.js in Action By Alex Young Dreamtech Press
3.	Angular 2 Cookbook By Frishbie Packt Publishing
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). <ul style="list-style-type: none"> installing npm installing new package installing package globally updating package removing package
2.	Understanding package.json file Understanding package.json file
3.	Demonstrate the use of core NodeJS modules Demonstrate the use of core NodeJS modules <ul style="list-style-type: none"> Demonstrate "path" core module in NodeJS. Demonstrate "fs" core module in NodeJS. Demonstrate "child_process" core module in NodeJS. Demonstrate the use of EventEmitter in NodeJS. WAP in NodeJS to store the student details in text file.
4.	Demonstrate the use of http core module in NodeJS Demonstrate the use of http core module in NodeJS <ul style="list-style-type: none"> Create a hello world webapp using "http" core module in NodeJS. Create a webapp with 5 pages like about, contact etc.. using "http" core module in NodeJS. Create a webapp in NodeJS which reads files like about.txt, contact.txt and display it using http core module.
5.	Demonstrate the basic ExpressJS web application Demonstrate the basic ExpressJS web application <ul style="list-style-type: none"> Create a hello world webapp using ExpressJS. Create a webapp with 5 pages like about, contact etc.. using ExpressJS. Create a webapp in NodeJS which reads files like about.txt, contact.txt and display it using http core module

6.	Create middleware in ExpressJS Create middleware in ExpressJS <ul style="list-style-type: none"> Demonstrate the use of middleware in Express. Demonstrate the use of static middleware in Express.
7.	Setup MongoDB Install MongoDB and MongoDBCompass Setup documents in MongoDB.
8.	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.
9.	Create a restful CRUD API using NodeJS, Express and MongoDB for faculty. Create a restful CRUD API using NodeJS, Express and MongoDB for faculty.
10.	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. Create a restful CRUD API using NodeJS, Express and MongoDB for product.
11.	Create basic component in Angular Setup Environment for Angular. Create first project in Angular. Hello world webapp using Angular. Demonstrate how to create component in Angular.
12.	Demonstrate routing and binding in Angular Demonstrate basic routing in Angular. Demonstrate the routing parameters in Angular. Demonstrate the Interpolation. Demonstrate the Property Binding. Demonstrate the Event Binding. Demonstrate Two-way Data Binding.
13.	Demonstrate directives and observable in Angular Demonstrate built-in attribute directives in Angular. Demonstrate built-in structural directives in Angular. Demonstrate Observable in Angular. Demonstrate Dependency Injection in Angular.
14.	Demonstrate form handling in Angular Demonstrate Template Driven forms in Angular. Create webapp using Angular to perform CRUD operation on students stored as array. Create webapp using Angular to perform CRUD operation on products stored as array. Create webapp using Angular to perform CRUD operation on faculties stored as array.
15.	Consume API created on MockAPI using Angular Create a MockAPI online with following fields. <ul style="list-style-type: none"> FacultyID FacultyName FacultyExp FacultyImage Perform CRUD operation on MockAPI using Angular.

16.	Implement CRUD operation on different MockAPIs using Angular. (minimum 3 APIs) Implement CRUD operation on different MockAPIs using Angular. (minimum 3 APIs)
17.	Create API using NodeJS, ExpressJS, MongoDB and consume it using Angular Create API using NodeJS, ExpressJS, MongoDB and consume it using Angular
18.	Setup ReactJS development environment Setting up react environment. Hello world webapp using ReactJS. Demonstrate the use of JSX. WAP to create a simple class component in ReactJS. WAP to create a simple function component in ReactJS.
19.	Demonstrate components and its properties in ReactJS Create a function component in separate file and link with App.js Demonstrate the ReactJS props. Demonstrate the Event Handling in ReactJS. WAP in ReactJS to display the element if it has attribute called isDisplay to be true (using conditional rendering).
20.	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.
21.	Implement Routing in ReactJS. Develop basic website using 5 different component (pages) and implement Routing in it. (i.e. About, Contact etc...).
22.	Demonstrate Hooks in ReactJS Demonstrate useState hook in ReactJS. Demonstrate useEffect hook in ReactJS. WAP to create a simple calculator using 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.	Implement CRUD operation on different MockAPIs using ReactJS. (minimum 3 APIs) Implement CRUD operation on different MockAPIs using ReactJS. (minimum 3 APIs)
24.	Create API using NodeJS, ExpressJS, MongoDB and consume it using ReactJS Create API using NodeJS, ExpressJS, MongoDB and consume it using ReactJS