| | | |
|---|---|---|
| **Program** | Bachelor of Technology (BTech) | **Semester - 3** |
| **Type of Course** | Professional Core | |
| **Prerequisite** | | |
| **Course Objective** | Students will be able to understand modern web technologies based on Java Script. Student can develop Single Page Application (SPA) Development. Students will be able to provide efficient ways of building web applications by offering reusable components, state management solutions, and optimized rendering mechanisms. | |

| Teaching Scheme (Contact Hours) | | | | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | Theory Marks | | Practical Marks | | |
| Lecture | Tutorial | Practical | Credit | SEE | CIA | SEE | CIA | Total Marks |
| 3 | 0 | 4 | 5 | 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 | **Java Script** | 9 | 20 |
| | Syntax of JavaScript, Execution of JavaScript, Internal, Embedded and External Javascript, JavaScript : variables, arrays, functions, conditions, loops, Pop up boxes, JavaScript objects and DOM, JavaScript inbuilt functions, JavaScript validations and Regular expressions, Event handling with JavaScript, Callbacks in Javascript, Function as arguments in JavaScript. Introduction to JQuery, Exploring JQuery. | | |
| 2 | **NodeJS** | 9 | 20 |
| | Introduction to Node JS, Setup Development Environment, Node JS Modules, Node Package Manager, Creating Web Server, File System, Debugging Node JS Application, Events. | | |
| 3 | **ExpressJS** | 9 | 20 |
| | Express JS, Serving Static Resources, Database Connectivity, API using NodeJS and ExpressJS. | | |
| 4 | **Introduction to ReactJS** | 9 | 20 |
| | Introduction to ReactJS, Introducing JSX, Rendering Elements, Component and Props, State and Lifecyle, Handling Events, Conditional Rendering, Routing, React CSS styling. | | |
| 5 | **Consuming API in ReactJS** | 9 | 20 |
| | Hooks, Forms, Consuming API in ReactJS. | | |
| | **Total** | **45** | **100** |

**Suggested Distribution Of Theory Marks Using Bloom's Taxonomy**

| Level | Remembrance | Understanding | Application | Analyze | Evaluate | Create |
|---|---|---|---|---|---|---|
| **Weightage** | 10 | 35 | 55 | 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 | **articulate** clientside scripting using JavaScript |
| CO2 | **prepare** NodeJS projects using Node Package Manager. |
| CO3 | **implement** Restful APIs using ExpressJS |
| CO4 | **use** ReactJS to create Front-End. |
| CO5 | **manipulate** API using ReactJS. |

## Reference Books

| | |
|---|---|
| 1. | **Developing Web Application**<br>By Ralph Moseley \| Wiley India |
| 2. | **Professional Node.js**<br>By Pedro Teixeira \| John Wiley & Sons |
| 3. | **Node.js in Action**<br>By Alex Young \| Dreamtech Press |
| 4. | **Learning React: Fundamental Web Development with react and redux**<br>By Alex Banks \| O'Reilly |
| 5. | **Introduction to Web Technology**<br>By Pankaj Sharma \| S.K.Kataria & Sons \| 5th |

## List of Practical

| | |
|---|---|
| **1.** | **Demonstrate the use of popup boxes in JavaScript**<br>1. WAP in JavaScript to make simple calculator using popup box. (A)<br>2. WAP in JavaScript to check whether the given no. is prime or not. (A)<br>3. WAP in JavaScript to fnd the factorial of given number. (A)<br>4. WAP in JavaScript to print the Fibonacci series of a number. (A)<br>5. WAP in JavaScript to check whether the given number is palindrome or not. (A)<br>6. WAP in JavaScript to check whether the given number is Armstrong or not. (A) |
| **2.** | **Implement basic JavaScript programs**<br>1. WAP in JavaScript to print the factors of given number. (A)<br>2. WAP in JavaScript to prime number between the given range of number. (A)<br>3. WAP to display given patterns using JavaScript. (A)<br>4. WAP in JavaScript to print the GCD of two number. (A)<br>5. Write a JavaScript to take 2-digit number and then separate these 2 digits, then multiply frst digit by itself for second digit times. (B)<br>(For example, 23 should be separated as 2 and 3. 2 should multiply with itself 3 times).<br>6. Write an HTML page with JavaScript that takes a number from popup box in the range 0-999 and display it in words. If the number is out of range, it should show "out of range" and if it is not a number ,it should show "not a number" message in the result box. (A)<br>7. Write an HTML fle with JavaScript that fnds position of frst occurrence of vowel "a", last occurrence of vowel "a" in a given word and returns the string between them. For example, ajanta- then script would return frst occurrence of "a"-that is position 1 and last occurrence-6 and string between them is "jant". (B) |
| **3.** | **Implement basic array programs in JavaScript**<br>1. Demonstrate the use of array to find maximum number. (A)<br>2. WAP to read an array from the user and sort them in ascending order. (A)<br>3. WAP to read an array from the user and sort them using bubble sort. (A)<br>4. WAP to read an array form the user and sort them using quick sort. (B) |

| | |
|---|---|
| | 5. Write a JavaScript that uses a loop that searches a word in sentence held in an array, returning the index of the word. (B) |
| **4.** | **Render Array elements using JavaScript**<br>1. WAP to display Faculties stored in array. (A)<br>2. Demonstrate different array functions like push, pop, shift, unshift, splice, sort etc… (A)<br>3. WAP to display students stored in array. (A)<br>4. WAP to display products stored in array. (B) |
| **5.** | **Implement Validation of form in JavaScript**<br>1. WAP to create a basic graphical Calculator using HTML and JavaScript. (A)<br>2. WAP to create a static login page using HTML and JavaScript. (B)<br>3. WAP to create a scientifc graphical calculator using HTML and JavaScript. (C)<br>4. Write a JavaScript program to validate user data given from the HTML form (A)<br>1) username must be of minimum 8 characters<br>2) password must contain at least 1 digit and 1 special character and should be between 8-12 characters<br>3) password and repeat password must be same<br>4) age must be greater than 18 (calculate from date of birth)<br>5) enrollment must be of 11 digits<br>6) email validation |
| **6.** | **Demonstrate Document Object Model methods in JavaScript**<br>1. WAP to change background color on click of button. (A)<br>2. WAP to recognize which mouse event is fired. (A)<br>3. WAP to recognize which keyboard event is fired. (B)<br>4. WAP to recognize which form event is fired. (B)<br>5. WAP to demonstrate change in properties of HTML element using JavaScript. (A)<br>6. WAP to prepare student registration form and validate it using JavaScript. (A)<br>7. WAP to perform CRUD operation on Array using HTML and JavaScript. (C)<br>8. Write a JavaScript that handles following mouse events. Add necessary elements. (B)<br>If the mouse is over the heading should turn yellow and if the mouse goes out of the heading it should turn black.<br>If fndtime button is clicked show time and date information.<br>If button named “red” is clicked, background should change to red and If button named “green” is clicked, background should change to green.<br>9. Write a JavaScript that handles following mouse events. Add necessary elements. (B)<br>JavaScript gives the key code for the key pressed.<br>If the key pressed is “a”,”e”,”i”,”o”,”u”, the script should announce that vowel is pressed.<br>When the key is released background should change to blue. |
| **7.** | **Demonstrate ES6 features in JavaScript**<br>1. WAP to demonstrate callbacks in JavaScript. (A)<br>2. Demonstrate the difference between let and var. (A)<br>3. Demonstrate the default parameter while using a function. (A)<br>4. Demonstrate the spread operator. (A)<br>5. Demonstrate the ‘for of’ loop. (A)<br>6. Demonstrate the Array and Object Destructuring. (B)<br>7. Demonstrate the Arrow functions. (B)<br>8. Demonstrate how to create a class in Java Script. (B)<br>9. Create a Snake game using Java script. (C)<br>10. Write JavaScript that handles following mouse event. (B)<br>• If mouse left button pressed on browser, it displayed message “Left Clicked”.<br>• If mouse right button pressed on browser, it displayed message “Right Clicked”.<br>11. Write a JavaScript having a list of checkbox and by clicking on checkbox, it should show list of selected value in comma separated format. (e.g. list of roll number as checkbox value, and display selected roll number in comma separated format) (C) |
| **8.** | **Setting up ReactJS development Environment** |

| | |
|---|---|
| | 1. Setting up react environment. (A)<br>2. Hello world webapp using ReactJS. (A)<br>3. Demonstrate the use of JSX. (A)<br>4. WAP to create a simple class component in ReactJS. (A)<br>5. WAP to create a simple function component in ReactJS. (A) |
| **9.** | **Create Hello World ReactJS app**<br>1. Create a function component in separate file and link with App.js (A)<br>2. Create a class component in separate file and link with App.js (B) |
| **10.** | **Demonstrate props in ReactJS**<br>1. Demonstrate the ReactJS props. (A)<br>2. Demonstrate the Event Handling in ReactJS. (A)<br>3. WAP in ReactJS to display the element if it has attribute called isDisplay to be true (using conditional rendering) (A) |
| **11.** | **Demonstrate the use of map method in ReactJS**<br>1. Demonstrate the use of map method in ReactJS to display array. (A)<br>2. Display Faculties stored in array using ReactJS. (B)<br>3. Display Students stored in array using ReactJS. (B)<br>4. Display Products stored in array using ReactJS (C ) |
| **12.** | **Demonstrate Routing in ReactJS**<br>1. Implement Routing in ReactJS. (A)<br>2. Develop basic website using 5 different component (pages) and implement Routing in it. (i.e. About, Contact etc…) (A)<br>3. Develop full static website using 15 different component (pages) and implement Routing in it. (i.e. About, Contact etc…) (C) |
| **13.** | **Demonstrate the use of hooks in ReactJS**<br>1. Demonstrate useState hook in ReactJS. (A)<br>2. Demonstrate useEffect hook in ReactJS (A) |
| **14.** | **Create GUI Calculator using ReactJS**<br>1. WAP to create a simple calculator using ReactJS. (A)<br>2. WAP to create a scientific calculator using ReactJS. (C ) |
| **15.** | **Implement CRUD operation on Array in ReactJS**<br>1. WAP to do CRUD operation on products stored as array using ReactJS. (A)<br>2. WAP to do CRUD operation on students stored as array using ReactJS. (B)<br>3. WAP to do CRUD operation on faculties stored as array using ReactJS. (C ) |
| **16.** | **Perform CRUD operation on MockAPI using ReactJS**<br>1. Create a MockAPI online with following fields. (A)<br>• FacultyID<br>• FacultyName<br>• FacultyExp<br>• FacultyImage<br>2. Perform CRUD operation on MockAPI using ReactJS. (minimum 3 mock api) (A)<br>3. Perform CRUD operation on MockAPI using ReactJS. (minimum 8 mock api) (B)<br>4. Perform CRUD operation on MockAPI using ReactJS. (minimum 15 mock api) (C) |
| **17.** | **Demonstrate the use of Node Package Manager**<br>1. Demonstrate the use of Node Package Manage (NPM). (A)<br>2. Demonstrate “path” core module in NodeJS. (A)<br>3. Demonstrate “fs” core module in NodeJS. (A)<br>4. Demonstrate “child_process” core module in NodeJS (A)<br>5. Explore minimum 3 other core module from the documentation of NodeJS (B)<br>6. Explore minimum 8 other core module from the documentation of NodeJS (C ) |
| | |

| | |
|---|---|
| **18.** | **Create basic webapp using http core module**<br>1. Create a hello world webapp using “http” core module in NodeJS. (A)<br>2. Create a webapp with 5 pages like about, contact etc.. using “http” core module in NodeJS. (B)<br>3. Create a webapp in NodeJS which reads files like about.txt, contact.txt and display it using http core module (C) |
| **19.** | **Create basic webapp using expressJS**<br>1. Create a hello world webapp using ExpressJS. (A)<br>2. Create a webapp with 5 pages like about, contact etc.. using ExpressJS. (B)<br>3. Create a webapp in NodeJS which reads files like about.txt, contact.txt and display it using express (C ) |
| **20.** | **Setup MongoDB and middleware in ExpressJS**<br>1. Demonstrate the use of middleware in Express. (A)<br>2. Demonstrate the use of static middleware in Express. (A)<br>3. Install MongoDB and MongoDBCompass (A)<br>4. Setup documents in MongoDB. (A) |
| **21.** | **Create a MongoDB collections**<br>1. Install Mongoose library using NPM. (A)<br>2. Demonstrate the use mongoose functions. (A)<br>3. Create a Database using MongoDBCompass for faculty. (A)<br>4. Create a Database using MongoDBCompass for student. (B)<br>5. Create a Database using MongoDBCompass for product. (C) |
| **22.** | **Create CRUD API using Express, MongoDB and NodeJS**<br>1. Create a restful CRUD API using NodeJS, Express and MongoDB for faculty. (A)<br>2. Create a restful CRUD API using NodeJS, Express and MongoDB for student. (B)<br>3. Create a restful CRUD API using NodeJS, Express and MongoDB for product. (C ) |
| **23.** | **Create a mini project for library management system**<br>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**<br>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**<br>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**<br>Create and consume Restfull API using MongoDB, Express, ReactJS and NodeJS (MERN stack) for attendance management system. |

**Miscellaneous**

| **Useful Links** |
|---|
| https://reactjs.org/docs/getting-started.html<br>https://nodejs.org/en/docs/ |