

Hall Ticket Number:

--	--	--	--	--	--	--	--	--

24IT403**II/IV B. Tech (Regular) DEGREE EXAMINATION****May, 2026****Fourth Semester****Time: Three Hours****Information Technology
Frontend Web Technologies****Maximum: 60 Marks****Answer question 1 compulsorily.****(12X1 = 12 Marks)****Answer one question from each unit.****(4X12 = 48 Marks)**

CO	BL	M
CO1	L1	1M
CO1	L2	1M
CO1	L3	1M
CO2	L1	1M
CO2	L1	1M
CO2	L2	1M
CO2	L3	1M
CO2	L2	1M
CO3	L3	1M
CO3	L2	1M
CO4	L3	1M
CO4	L2	1M

1.
 - a) List any four HTML elements in detail
 - b) What is the difference between ROWSPAN and COLSPAN in a HTML table?
 - c) How do you create a hyperlink in HTML?
 - d) What are the advantages and disadvantages of Inline Styles?
 - e) What are the scopes of a variable in JavaScript?
 - f) List any three Mathematical functions in Java script
 - g) Mention any three Mouse events in JavaScript
 - h) Define Document Object Model (DOM).
 - i) What are the key components of a Node Structure in DOM?
 - j) What is JSX in the context of React?
 - k) Mention two differences between props and state.
 - l) Define version control in GitHub

Unit –I

2.
 - a) Create an HTML page named as “table.html” to display your class time table
 - b) Describe the various lists used in html with an example program.

(OR)

3.
 - a) Write an HTML program to create a web page that should display your college logo, college name and address by using image tag with necessary attributes
 - b) Design an HTML document for a Student registration form.

Unit –II

4.
 - a) Discuss CSS inline, internal and External Styles with an example. What are the advantages and disadvantages of each?
 - b) Using events write an example to create a simple animation.

(OR)

5.
 - a) Write a JavaScript program for generating Factorial of given number using functions.
 - b) Explain about Event Handling in JavaScript with an example.

Unit –III

6.
 - a) Define Object, explain how to use Objects in JavaScript with examples.
 - b) Write a JavaScript code for validating a form which contains Username with 10 characters, Email with username@mailaddress, Password with 6 characters and Mobile number with 10 digits using Regular Expressions

(OR)

7.
 - a) Explain in detail about ES6 Features with suitable examples.
 - b) Discuss the DOM (Document Object Model)? Explain getElementById() method with an example

Unit –IV

8.
 - a) Explain the concept of Routing in React with an Example.
 - b) Differentiate between Functional and Class Components in React with examples.

(OR)

9.
 - a) Write a simple React form application with a button when clicked that displays a counter.
 - b) Describe the role of GitHub in software development.



Scheme & Solutions

1. Answer All Questions.

a) List any four HTML elements in detail.

HTML elements are the building blocks of a web page. Four common HTML elements are:

(i) <html>

It is the root element of an HTML document.

All other HTML tags are written inside this tag.

(ii) <head>

It contains meta information such as title, styles, scripts, and links.

This part is not directly displayed on the web page.

(iii) <body>

It contains all visible content of the webpage like text, images, tables, forms, etc.

(iv) <p>

This tag is used to define a paragraph of text in HTML.

Example:

<p>This is a paragraph</p>

Any other valid HTML elements → 1M

b) What is the difference between ROWSPAN and COLSPAN in an HTML table?

ROWSPAN	COLSPAN
It merges multiple rows into one cell vertically.	It merges multiple columns into one cell horizontally.
Used with rowspan attribute.	Used with colspan attribute.
Example: <td rowspan="2">	Example: <td colspan="3">

Any two differences ----→ 1M

c) How do you create a hyperlink in HTML?

A hyperlink is created using the <a> (anchor) tag.

Syntax:

Google

Here:

- href specifies destination URL
- text between tags becomes clickable link

-----→ 1M

d) What are the advantages and disadvantages of Inline Styles?

Advantages:

- Easy to apply style to a single element
- Quick testing and debugging
- Overrides external styles

Disadvantages:

- Makes HTML code lengthy
- Difficult to maintain large websites
- Repetition of style code
- Not reusable

Any two valid points---→1M

Example:

```
<p style="color:red;">Hello</p>
```

e) What are the scopes of a variable in JavaScript?

JavaScript variables have mainly three scopes:

1. Global Scope – accessible anywhere in the program.
2. Function Scope – accessible only inside the function.
3. Block Scope – accessible only inside { } block using let and const.

----→ 1M

f) List any three Mathematical functions in JavaScript.

JavaScript provides Math object functions.

Any three are:

- Math.sqrt(x) → returns square root
- Math.pow(x,y) → returns x raised to power y
- Math.random() → returns random number

Example:

```
Math.sqrt(25); //5
```

Any valid three functions→1M

g) Mention any three Mouse events in JavaScript.

Three common mouse events are:

1. onclick – triggered when mouse is clicked
2. onmouseover – triggered when mouse pointer moves over element
3. onmouseout – triggered when mouse leaves the element

Any three valid events -→ 1M

h) Define Document Object Model (DOM).

The Document Object Model (DOM) is a programming interface that represents an HTML or XML document as a tree structure of objects.

Using DOM, JavaScript can:

- access HTML elements
- modify content
- change styles dynamically

-----→ 1M

i) What are the key components of a Node Structure in DOM?

DOM treats every part of HTML as a node.

Main components are:

1. Element Node – HTML tags like <p>, <div>
2. Text Node – text inside elements
3. Attribute Node – attributes like id, class
4. Comment Node – HTML comments

These nodes together form a hierarchical tree.

----→ 1M

10. What is JSX in the context of React?

JSX (JavaScript XML) is a syntax extension used in React that allows writing HTML-like code inside JavaScript.

Example:

```
const element = <h1>Hello React</h1>;
```

JSX makes React components easier to write and understand.

-----→ 1M

11. Mention two differences between props and state.

Props	State
Props are passed from parent component to child component.	State is maintained inside the component itself.
Props are read-only and cannot be modified.	State can be changed using setState() or hooks.

Any two valid differences ----→ 1M

12. Define version control in GitHub.

Version Control is the process of managing and tracking changes made to source code over time.

In GitHub, version control helps to:

- save multiple code versions
- track modifications
- collaborate with team members
- restore previous versions if needed

Git is the version control system used by GitHub.

---→ 1M

Unit-I

2. a) Create an HTML Page named “table.html” to Display Your Class Timetable.

[6M]

HTML Program: table.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Class Time Table</title>
```

```
</head>
```

```
<body>
```

```
<h2 align="center">B.Tech CSE - Class Time Table</h2>
```

```
<table border="1" align="center" cellpadding="10" cellspacing="0">
```

```
<tr>
```

```
<th>Day / Period</th>
```

```
<th>9:00-10:00</th>
```

```
<th>10:00-11:00</th>
```

```
<th>11:00-12:00</th>
```

```
<th>12:00-1:00</th>
```

```
<th>2:00-3:00</th>
```

```
<th>3:00-4:00</th>
```

-----→ 2M

```
</tr>
```

```
<tr>
```

```
<td>Monday</td>
```

```
<td>DS</td>
```

```
<td>WT</td>
```

```
<td>OS</td>
```

```
<td rowspan="5">LUNCH</td>
```

```
<td>CN</td>
```

```
<td>SE</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Tuesday</td>
```

```
<td>WT</td>
```

```
<td>DS Lab</td>
```

```
<td>DS Lab</td>
```

```
<td>OS</td>
```

```

        <td>CN</td>
    </tr>

    <tr>
        <td>Wednesday</td>
        <td>CN</td>
        <td>SE</td>
        <td>WT</td>
        <td>DS</td>
        <td>Library</td>
    </tr>

    <tr>
        <td>Thursday</td>
        <td>OS</td>
        <td>WT Lab</td>
        <td>WT Lab</td>
        <td>SE</td>
        <td>CN</td>
    </tr>

    <tr>
        <td>Friday</td>
        <td>DS</td>
        <td>CN</td>
        <td>OS</td>
        <td colspan="2" align="center">Seminar / Project</td>
    </tr>
</table>

</body>
</html>
-----→ 4M

```

2. b) Describe the various lists used in html with an example program. [6M]

Ans: In HTML, a list is used to display a group of related items in an organized manner.

Lists improve readability and help present information clearly on web pages.

HTML provides mainly three types of lists:

1. Ordered List
2. Unordered List
3. Description List

1. Ordered List ()

An Ordered List displays items in a numbered or alphabetic sequence.

Each item is represented using tag.

Syntax

```
<ol>
```

```
<li>Item1</li>
```

```
<li>Item2</li>
```

```
</ol>
```

Example

```
<ol>
```

```
<li>HTML</li>
```

```
<li>CSS</li>
```

```
<li>JavaScript</li>
```

```
</ol>
```

Output

1. HTML

2. CSS

3. JavaScript

Ordered List Types

We can also specify numbering style:

- type="1" → numbers
- type="A" → uppercase letters
- type="a" → lowercase letters
- type="I" → Roman capitals

Example:

```
<ol type="A">
```

```
<li>CSE</li>
```

```
<li>IT</li>
```

```
</ol>
```

-----→ 2M

2. Unordered List ()

An Unordered List displays items with bullets instead of numbers.

Syntax

```
<ul>
```

```
<li>Item1</li>
```

```
<li>Item2</li>
```

```
</ul>
```

Example

```
<ul>
```

```
<li>Apple</li>
```

```
<li>Mango</li>
```

```
<li>Orange</li>
```

```
</ul>
```

Output

- Apple
- Mango
- Orange

Bullet Types

- disc
- circle
- square

Example:

```
<ul type="square">
```

-----→ 2M

3. Description List (<dl>)

A Description List is used to display terms and their descriptions.

It uses:

- <dl> → description list
- <dt> → description term
- <dd> → description definition

Syntax

```
<dl>
```

```
<dt>HTML</dt>
```

```
<dd>Hyper Text Markup Language</dd>
```

```
</dl>
```

Example Output

HTML

→ Hyper Text Markup Language

CSS

→ Cascading Style Sheets

-----→ 2M

(OR)

3. a) Write an HTML program to create a web page that should display your college logo, college name and address by using image tag with necessary attributes. [6M]

Ans:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>College Information Page</title>
```


</head>

<body bgcolor="lightyellow">

-----→ 2M

<center>

<h1>BAPATLA ENGINEERING COLLEGE</h1>

<h3>Bapatla, Andhra Pradesh - 522101</h3>

<h4>An Autonomous Institution Affiliated to JNTUK</h4>

</center>

</body>

</html>

-----→ 4M

3. b) Design an HTML document for a Student registration form.

[6M]

<!DOCTYPE html>

<html>

<head>

<title>Student Registration Form</title>

</head>

<body bgcolor="lightyellow">

<center>

<h1>STUDENT REGISTRATION FORM</h1>

</center>

<form>

<table align="center" cellpadding="8">

<tr>

<td>Student Name :</td>

<td><input type="text" name="sname" placeholder="Enter Full Name"></td>

</tr>

<tr>

<td>Father Name :</td>

```

        <td><input type="text" name="fname" placeholder="Enter Father Name"></td>
    </tr>

    <tr>
        <td>Mother Name :</td>
        <td><input type="text" name="mname" placeholder="Enter Mother Name"></td>
    </tr>

    <tr>
        <td>Password :</td>
        <td><input type="password" name="pwd"></td>
    </tr>

    <tr>
        <td>Email ID :</td>
        <td><input type="email" name="mail"></td>
    </tr>

    <tr>
        <td>Mobile Number :</td>
        <td><input type="tel" name="mobile"></td>
    </tr>

    <tr>
        <td>Gender :</td>
        <td>
            <input type="radio" name="gender">Male
            <input type="radio" name="gender">Female
        </td>
    </tr>

    <tr>
        <td>Date of Birth :</td>
        <td><input type="date" name="dob"></td>
    </tr>

```

-----→ 3M

```

    <tr>
        <td>Department :</td>
        <td>

```

```

        <select name="dept">
            <option>CSE</option>
            <option>IT</option>
            <option>ECE</option>
            <option>EEE</option>
            <option>MECH</option>
        </select>
    </td>
</tr>

<tr>
    <td>Languages Known :</td>
    <td>
        <input type="checkbox">Telugu
        <input type="checkbox">English
        <input type="checkbox">Hindi
    </td>
</tr>

<tr>
    <td>Address :</td>
    <td><textarea rows="4" cols="25"></textarea></td>
</tr>

<tr>
    <td>Upload Photo :</td>
    <td><input type="file"></td>
</tr>

<tr>
    <td colspan="2" align="center">
        <input type="submit" value="Register">
        <input type="reset" value="Clear">
    </td>
</tr>
</table>
</form>
</body>
</html>

```

Unit-II

4. a) Discuss CSS inline, internal and External Styles with an example. What are the advantages and disadvantages of each? [6M]

Ans: CSS (Cascading Style Sheets) is used to control the appearance and layout of web pages. CSS can be applied to HTML in three different ways:

1. Inline CSS
2. Internal CSS
3. External CSS

1. Inline CSS

Inline CSS is written directly inside an HTML element using the style attribute.

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>Inline CSS</title>
</head>
<body>

<h1 style="color: blue; font-size: 30px;">
  Welcome to CSS
</h1>

<p style="color: green;">
  This is an example of Inline CSS.
</p>
</body>
</html>
```

Advantages of Inline CSS

- Easy to use for small changes.
- No separate CSS file required.
- Useful for testing styles quickly.
- Styles apply only to a specific element.

Disadvantages of Inline CSS

- Makes HTML code lengthy and difficult to read.
- Cannot reuse styles across multiple pages.
- Increases maintenance difficulty.
- Violates separation of content and presentation.
- Not suitable for large websites.

2. Internal CSS

Internal CSS is written inside the <style> tag within the <head> section of the HTML document.

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>Internal CSS</title>

  <style>
    h1 {
      color: red;
      text-align: center;
    }

    p {
      color: purple;
      font-size: 20px;
    }
  </style>

</head>
<body>

  <h1>Welcome to CSS</h1>

  <p>This is an example of Internal CSS.</p>

</body>
</html>
```

Advantages of Internal CSS

- Easy to apply styles to a single webpage.
- No external file needed.
- Better readability compared to inline CSS.
- Useful for small websites or single-page designs.

Disadvantages of Internal CSS

- Styles cannot be reused across multiple pages.
- Increases page size.
- Difficult to maintain for large projects.
- Every page needs separate style definitions.

3. External CSS

External CSS is written in a separate .css file and linked to the HTML document using the <link> tag.

HTML File (index.html)

```
<!DOCTYPE html>

<html>

<head>
  <title>External CSS</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
<h1>Welcome to CSS</h1>
<p>This is an example of External CSS.</p>
</body>
</html>
```

CSS File (style.css)

```
h1 {
  color: orange;
  text-align: center;
}
p {
  color: brown;
  font-size: 22px;
}
```

Advantages of External CSS

- Styles can be reused across multiple pages.
- Easy maintenance and updates.
- Keeps HTML code clean and organized.
- Reduces code duplication.
- Improves website consistency.
- Faster page loading due to browser caching.

Disadvantages of External CSS

- Requires an additional CSS file.
- Webpage may appear unstyled if the CSS file fails to load.
- Slightly more complex for beginners.
- Requires proper file linking.

-----→ 2M

4. b)

Ans: <!DOCTYPE html>

<html>

<head>

<title>Simple Animation using Events</title>

<style>

```
#container {  
    width: 600px;  
    height: 100px;  
    border: 2px solid black;  
    position: relative;  
}
```

```
#box {  
    width: 50px;  
    height: 50px;  
    background-color: red;  
    position: absolute;  
    left: 0px;  
    top: 25px;  
}
```

</style>

</head>

<body>

<h2>Animation using Click Event</h2>

<div id="container">

<div id="box"></div>

</div>

<button onclick="startAnimation()">Start Animation</button>

-----→ 3M

<script>

```
function startAnimation() {  
    var box = document.getElementById("box");  
    var position = 0;  
    var animation = setInterval(frame, 10);  
    function frame() {  
        if (position >= 550) {  
            clearInterval(animation);
```

```

    }
    else {
        position++;
        box.style.left = position + "px";
    }
}
}
</script>
</body>
</html>

```

-----→ 3M

(OR)

5. a) Write a JavaScript program for generating Factorial of given number using functions. 6M

Ans: <!DOCTYPE html>

```

<html>
<head>
    <title>Factorial Program</title>
</head>
<body>
<h2>Factorial of a Number</h2>
<script>
    // Function to calculate factorial
    function factorial(n) {
        let fact = 1;
        for(let i = 1; i <= n; i++) {
            fact = fact * i;
        }
        return fact;
    }
    // Input from user
    let num = prompt("Enter a number:");
    // Calling function
    let result = factorial(num);
    // Display output
    document.write("Factorial of " + num + " is: " + result);
</script>
</body>
</html>

```

-----→ 3M

-----→ 3M

5. b) Explain about Event Handling in JavaScript with an example.**6M**

Event handling in JavaScript is the process of responding to user actions or browser activities such as:

- Mouse Events
- Key Board Events
- Form Events

| Event | Description |
|-------------|---|
| onclick | Occurs when an element is clicked |
| onmouseover | Occurs when mouse moves over an element |
| onmouseout | Occurs when mouse leaves an element |
| onkeydown | Occurs when a key is pressed |
| onload | Occurs when page is loaded |
| onsubmit | Occurs when form is submitted |
| onchange | Occurs when value of element changes |

Any two events from Mouse or Keyboard with examples ----→ 3+ 3= 6M

Unit-III**6. a) Define Object, explain how to use Objects in JavaScript with examples.****6M**

Ans: An Object in JavaScript is a collection of properties and methods.

It is used to store data in the form of key–value pairs.

- Property → stores values
- Method → stores functions

Objects help represent real-world entities such as student, employee, car, etc.

Creating and Using Objects in JavaScript

Objects can be created using:

1. Object Literal
2. Object Constructor
3. Using new Object()

Explanation for any one method --→ 3M

Example 1: Object Literal

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Object Example</title>
</head>
<body>
<script>
  // Creating object
  var student = {
    name : "Ravi",
    rollno : 101,
    branch : "IT",
```

```

display : function() {
    return "Name: " + this.name +
        "<br>Roll No: " + this.rollno +
        "<br>Branch: " + this.branch;
}
};
// Accessing object properties
document.write(student.display());
</script>
</body>
</html>

```

Any one example -----→ 3M

6. b) Write a JavaScript code for validating a form which contains Username with 10 characters, Email with username@mailaddress, Password with 6 characters and Mobile number with 10 digits using Regular Expressions. **6M**

Ans:

```

<!DOCTYPE html>
<html>
<head>
    <title>Form Validation</title>
    <script>
        function validateForm() {
            // Getting values from form
            var username = document.getElementById("uname").value;
            var email = document.getElementById("email").value;
            var password = document.getElementById("pass").value;
            var mobile = document.getElementById("mobile").value;

            // Regular Expressions
            var userPattern = /^[A-Za-z0-9]{10}$/;
            var emailPattern = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
            var passPattern = /^[0-9]{6}$/;
            var mobilePattern = /^[0-9]{10}$/;

            // Username validation
            if(!userPattern.test(username)) {
                alert("Username must contain exactly 10 characters");
                return false;
            }
        }
    </script>

```

```

// Email validation
if(!emailPattern.test(email)) {
    alert("Enter valid Email address");
    return false;
}

// Password validation
if(!passPattern.test(password)) {
    alert("Password must contain minimum 6 characters");
    return false;
}

// Mobile validation
if(!mobilePattern.test(mobile)) {
    alert("Mobile number must contain exactly 10 digits");
    return false;
}
alert("Form Submitted Successfully");
return true;
}

```

</script>

-----> 3M

</head>

<body>

<h2>Registration Form Validation</h2>

<form onsubmit="return validateForm()">

Username:

<input type="text" id="uname">

Email:

<input type="text" id="email">

Password:

<input type="password" id="pass">

Mobile Number:

<input type="text" id="mobile">

<input type="submit" value="Submit">

</form>

</body>

</html>

-----> 3M

(OR)

7. a) Explain in detail about ES6 Features with suitable examples.

6M

Ans: ES6 (ECMAScript 2015) is an important version of JavaScript that introduced many new features to make programming easier, cleaner, and more efficient.

Important ES6 Features

1. let and const
2. Arrow Functions
3. Template Literals
4. Default Parameters
5. Destructuring
6. Spread Operator
7. Rest Parameters
8. Classes
9. Modules
10. Promises

1. let and const

Before ES6, variables were declared using var.

ES6 introduced:

- let → block scoped variable
- const → constant value

Example

```
let name = "Ravi";  
name = "Kiran";  
const pi = 3.14;  
console.log(name);  
console.log(pi);
```

Advantages

- Prevents scope issues
- Improves code safety
- const prevents reassignment

2. Arrow Functions

Arrow functions provide a shorter syntax for writing functions.

Syntax

```
const functionName = () => {  
  // code  
}
```

Example

```
const add = (a, b) => a + b;  
console.log(add(10, 20));
```

Output

30

Advantages

- Short and clean syntax
- Easy to write anonymous functions

3. Template Literals

Template literals use backticks `` to create strings.

They allow:

- Multi-line strings
- String interpolation

Example

```
let name = "Suresh";
```

```
let age = 22;
```

```
console.log(`My name is ${name} and age is ${age}`);
```

Output

My name is Suresh and age is 22

4. Default Parameters

Function parameters can have default values.

Example

```
function greet(name = "Guest") {  
  console.log("Hello " + name);  
}
```

```
greet();
```

```
greet("Ravi");
```

Output

Hello Guest

Hello Ravi

5. Destructuring

Destructuring extracts values from arrays or objects.

Array Destructuring Example

```
let numbers = [10, 20, 30];
```

```
let [a, b, c] = numbers;
```

```
console.log(a);
```

```
console.log(b);
```

Output

10

20

6. Spread Operator (...)

The spread operator expands elements of an array or object.

Example

```
let arr1 = [1, 2, 3];
```

```
let arr2 = [...arr1, 4, 5];
```

```
console.log(arr2);
```

Output: [1, 2, 3, 4, 5]

Any three features with examples--→ 3+3=6M

7. b) Discuss the DOM (Document Object Model)? Explain getElementById() method with an example. 6M

Ans: The Document Object Model (DOM) is a programming interface for HTML and XML documents. DOM represents a web page as a tree structure of objects.

Using JavaScript, we can:

- Access HTML elements
- Modify content
- Change styles
- Add or remove elements
- Respond to user actions dynamically

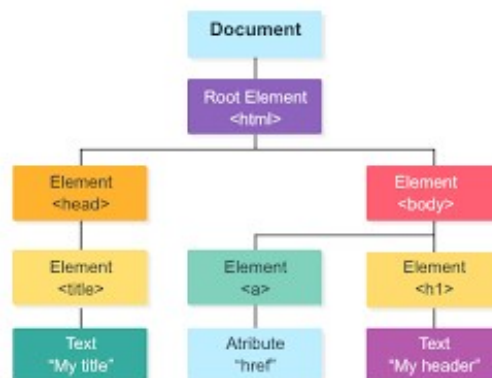
DOM Structure

In DOM, every element in an HTML document is treated as an object.

Example HTML:

```
<html>
  <body>
    <h1>Hello</h1>
  </body>
</html>
```

DOM Tree Representation:



-----→ 3M

Features of DOM

- Represents document as objects
- Allows dynamic webpage manipulation
- Platform and language independent
- Supports event handling
- Enables interactive web pages

DOM Methods

Some important DOM methods are:

Method	Description
getElementById()	Selects element using id
getElementsByName()	Selects elements using tag name
getElementsByClassName()	Selects elements using class name
querySelector()	Selects first matching element
querySelectorAll()	Selects all matching elements

getElementById() Method

The getElementById() method is used to access an HTML element using its unique id.

Syntax

```
document.getElementById("idname");
```

- Returns the element whose id matches the specified value.
- It is one of the most commonly used DOM methods.

Example Program

The following program changes the text of a paragraph when a button is clicked.

```
<!DOCTYPE html>
<html>
<head>
  <title>getElementById Example</title>
  <script>
    function changeText() {
      document.getElementById("msg").innerHTML =
        "Text Changed Successfully!";
    }
  </script>
</head>
<body>
<h2>DOM getElementById() Method</h2>

<p id="msg">Welcome to JavaScript</p>
```

```
<button onclick="changeText()">
```

```
  Click Here
```

```
</button>
```

```
</body>
```

```
</html>
```

----→ 3M

Unit-IV

8. a) Explain the concept of Routing in React with an Example.

6M

Ans: Routing in React is the process of navigating between different pages or components in a React application without reloading the entire webpage.

React uses the library React Router for routing.

It helps create Single Page Applications (SPA) where content changes dynamically while the browser remains on the same page.

Why Routing is Needed

Routing is used to:

- Navigate between pages
- Improve user experience
- Avoid full page reload
- Build Single Page Applications
- Manage URLs dynamically

Example:

Home Page → /

About Page → /about

Contact Page → /contact

Main Components of React Router

Component	Description
BrowserRouter	Wraps entire application
Routes	Contains all route definitions
Route	Defines path and component
Link	Navigation links
useNavigate()	Programmatic navigation

-----→ 3M

Example of Routing in React

Step 1: Create Components

Home.js

```
function Home() {  
  return <h2>Welcome to Home Page</h2>;  
}  
export default Home;
```


About.js

```
function About() {  
  return <h2>Welcome to About Page</h2>;  
}  
export default About;
```

Contact.js

```
function Contact() {  
  return <h2>Welcome to Contact Page</h2>;  
}  
export default Contact;
```

Step 2: Configure Routing in App.js

```
import React from "react";  
import { BrowserRouter, Routes, Route, Link } from "react-router-dom";  
import Home from "./Home";  
import About from "./About";  
import Contact from "./Contact";  
function App() {  
  return (  
    <BrowserRouter>  
      <nav>  
        <Link to="/">Home</Link> |  
        <Link to="/about">About</Link> |  
        <Link to="/contact">Contact</Link>  
      </nav>  
      <Routes>  
        <Route path="/" element={<Home />} />  
        <Route path="/about" element={<About />} />  
        <Route path="/contact" element={<Contact />} />  
      </Routes>  
    </BrowserRouter>  
  );  
}  
export default App;
```

-----→ 3M

8. b) Differentiate between Functional and Class Components in React with examples. 6M

Ans: In React, components are the building blocks of a user interface.

There are two main types of components:

1. Functional Components
2. Class Components

Functional Components

Functional components are simple JavaScript functions that return JSX.

They are mainly used for presenting UI elements.

Syntax

```
function ComponentName() {  
  return <h1>Hello</h1>;  
}
```

Example

```
function Welcome() {  
  return <h2>Welcome to React</h2>;  
}  
  
export default Welcome;
```

Features of Functional Components

- Simple and easy to write
- Faster and lightweight
- Uses Hooks for state and lifecycle features
- Mostly used in modern React applications

-----→ 3M

Class Components

Class components are ES6 classes that extend React.Component.

They can manage state and lifecycle methods.

Syntax

```
class ComponentName extends React.Component {  
  render() {  
    return <h1>Hello</h1>;  
  }  
}
```

Example

```
import React, { Component } from "react";  
  
class Welcome extends Component {  
  
  render() {  
    return <h2>Welcome to React</h2>;  
  }  
}
```

```

    }
  }
}
export default Welcome;

```

Features of Class Components

- Supports lifecycle methods
- Can maintain state using this.state
- More suitable for complex applications (before Hooks)

Difference Between Functional and Class Components

Functional Components	Class Components
Written as JavaScript functions	Written as ES6 classes
Simpler syntax	Complex syntax
Uses Hooks for state management	Uses this.state
No render() method required	render() method required
Better performance	Slightly slower
Easier to understand and test	More difficult for beginners
Preferred in modern React	Older approach

-----→ 3M

(OR)

9. a) Write a simple React form application with a button when clicked that displays a counter. 6M

Ans: Simple React Counter Application

The following React application contains:

- A button
- A counter value
- When the button is clicked, the counter increases

```

import React, { useState } from "react";
function CounterApp() {
  // State variable
  const [count, setCount] = useState(0);

  // Function to increase counter
  const incrementCounter = () => {
    setCount(count + 1);
  };

  return (
    <div>
      <h2>Simple Counter Application</h2>
      <h3>Counter Value: {count}</h3>

```

-----→ 3M

```

    <button onClick={incrementCounter}>
      Click Me
    </button>
  </div>
);
}
export default CounterApp; -----→ 3M

```

9. b) Describe the role of GitHub in software development. 6M

Ans: GitHub is a web-based platform used for version control and collaborative software development. It is built on Git, a distributed version control system created to track changes in source code. GitHub helps developers store, manage, and collaborate on software projects efficiently.

Main Roles of GitHub in Software Development

1. Version Control

GitHub tracks every change made to the source code.

Developers can:

- View previous versions
- Restore old code
- Compare changes
- Track project history

Example

If a bug occurs after updating code, developers can revert to an earlier working version.

2. Collaboration Among Developers

Multiple developers can work on the same project simultaneously.

GitHub allows team members to:

- Share code
- Review changes
- Merge updates
- Discuss issues

Example

In a web development project:

- One developer works on frontend
- Another works on backend
- Both can upload changes without overwriting each other's work

3. Repository Management

A **repository (repo)** is a storage location for project files and code.

GitHub repositories contain:

- Source code

- Documentation
- Images
- Configuration files

Repositories can be:

- Public
- Private

4. Branching and Merging

GitHub supports branching to develop features independently.

Branch

A branch is a separate copy of the project.

Example

main branch

feature-login branch

feature-payment branch

Developers can test features safely without affecting the main project.

Merging

After testing, branches are merged into the main branch.

5. Backup and Cloud Storage

GitHub stores projects in the cloud.

Benefits:

- Prevents data loss
- Access from anywhere
- Easy project recovery

6. Open Source Development

GitHub is widely used for open-source projects.

Developers around the world can:

- Contribute code
- Fix bugs
- Improve software

Example

Popular open-source projects hosted on GitHub:

- React
- Angular
- TensorFlow

Any three roles ----→ 3M

Scheme Prepared By:

K. Suresh Kumar,
Assistant Professor,
Department of IT, BEC

Signature of the HOD