

Hall Ticket Number:

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

II/IV B.Tech (Regular) DEGREE EXAMINATION

May, 2026

Fourth Semester

Time: Three Hours

Common to CB, CM, & DS

Client Side Web Technologies

Maximum: 60 Marks

Answer question 1 compulsorily.

(12X1 = 12 Marks)

Answer one question from each unit.

(4X12 = 48 Marks)

		CO	BL	M
1.	a) Define HTML5.	CO1	L1	1M
	b) What are metadata elements? Give an example.	CO1	L1	1M
	c) Define flow elements in HTML5.	CO1	L1	1M
	d) What is CSS?	CO2	L1	1M
	e) What is the use of display property?	CO2	L1	1M
	f) What is JavaScript?	CO2	L1	1M
	g) Define array in JavaScript	CO2	L1	1M
	h) Define Document Object.	CO3	L1	1M
	i) Compare between let and const.	CO3	L2	1M
	j) Define spread operator.	CO3	L1	1M
	k) What are components in React.js?	CO4	L1	1M
	l) Compare between MERN and MEAN?	CO4	L1	1M

Unit –I

2.	a) Explain the structure of an HTML5 document with example markup code.	CO1	L2	6M
	b) Explain about the form elements and implement a “Student Registration Form”.	CO1	L2	6M

(OR)

3.	a) Discuss the following HTML5 elements for arranging text and provide detailed example markup code. i) <div> ii) iii) <table> iv)	CO1	L2	6M
	b) Describe hyperlinks and structure of URL with examples.	CO1	L2	6M

Unit –II

4.	a) Explain different types of CSS (inline, internal, external) with example markup code.	CO2	L2	6M
	b) Discuss the CSS box model in detail with diagram and example.	CO2	L3	6M

(OR)

5.	a) Discuss about various CSS3 selectors used with suitable example code.	CO2	L2	6M
	b) Discuss array and functions in JavaScript and implement JavaScript application to find maximum and minimum elements in array using functions.	CO2	L3	6M

Unit –III

6.	a) Explain DOM structure and how it represents an HTML document.	CO3	L2	6M
	b) Discuss about Event handling mechanism in JavaScript with example code.	CO3	L3	6M

(OR)

7.	a) Explain Node, Document, and Element interfaces in detail.	CO3	L2	6M
	b) Explain arrow function, template literals with example code.	CO3	L2	6M

Unit –IV

8.	a) Explain the architecture and working of a basic React.js application.	CO4	L2	6M
	b) Discuss advantages and disadvantages of React.js.	CO4	L3	6M

(OR)

9.	a) Explain props and state in React.js with examples.	CO4	L2	6M
	b) Develop a React application for a menu component with an explanation.	CO4	L4	6M



II/IV B.Tech (Regular) DEGREE EXAMINATION

May, 2026

Seventh Semester

Time: Three Hours

CSE(Cyber Security/Data Science/AIML)

Client Side Web Technologies(24CB403/24CM403/24DS403)

Maximum : 60 Marks

SCHEME OF EVALUATION

Prepared By : J.KUMARARAJA
Assistant Professor,
Dept. of CSE(Cyber Security)
BEC, BAPATLA.

(Signature)

Internal Faculty Signature:

1.

2.

External Faculty Signature

HOD Signature

1.

2.

SCHEME OF EVALUATION

Marks-Allocation

UNIT – I

2 a) Explain the structure of an HTML5 document with example markup code.

Explanation	–	3 Marks
Program	–	3 Marks

2 b) Explain about the form elements and implement a “Student Registration Form”.

Explanation	–	3 Marks
Program	–	3 Marks

(or)

3. a) Discuss the following HTML5 elements for arranging text and provide a detailed example markup code.

i) <div> ii) iii) <table> iv)

Explanation	–	4 Marks
Program	–	2 Marks

3. b) Describe hyperlinks and structure of URL with examples.

Explanation	–	3 Marks
Program/Example	–	3 Marks

UNIT – II

4. a) Explain different types of CSS (inline, internal, external) with example markup code.

Explanation	–	3 Marks
Program(s)	–	3 Marks

4. b) Discuss the CSS box model in detail with diagram and example.

Explanation	–	3 Marks
Program/Example	–	3 Marks

(or)

5. a) Discuss about various CSS3 selectors used with suitable example code.

Explanation	–	3 Marks
Program/Example	–	3 Marks

5. b) Discuss array and functions in JavaScript and implement JavaScript application to find maximum and minimum elements in array using functions.

Explanation	–	3 Marks
Program/Example	–	3 Marks

UNIT – III

6. a) Explain DOM structure and how it represents an HTML document.

Explanation	–	3 Marks
Program/Example	–	3 Marks

6. b) Discuss about Event handling mechanism in JavaScript with example code.

Explanation	–	3 Marks
Program/Example	–	3 Marks

(or)

7. a) Explain Node, Document, and Element interfaces in detail.

Explanation	–	3 Marks
Program/Example	–	3 Marks

7. b) Explain arrow function, template literals with example code.

Explanation	–	3 Marks
Program/Example	–	3 Marks

UNIT – IV

8. a) Explain the architecture and working of a basic React.js application.

Explanation with Diagram	–	3 Marks
Program/Example	–	3 Marks

8. b) Discuss advantages and disadvantages of React.js.

Any 5/6 Points	–	6 Marks
----------------	---	---------

(or)

9. a) Explain props and state in React.js with examples.

Explanation with Diagram	–	3 Marks
Program/Example	–	3 Marks

9. b) Develop a React application for a menu component with an explanation.

Explanation with Diagram	–	3 Marks
Program/Example	–	3 Marks

II/IV B.Tech (Regular) DEGREE EXAMINATION

May, 2026

Seventh Semester

Time: Three Hours

CSE(Cyber Security/Data Science/AIML)

Client Side Web Technologies(24CB403/24CM403/24DS403)

Maximum : 60 Marks

SCHEME OF EVALUATION

Answer question 1 compulsorily.

(12X1 = 12 Marks)

Answer one question from each unit.

(4X12 = 48 Marks))

1. Answer all questions

(14 X 1 = 14 Marks)

a) Define HTML5.

HTML5 is the fifth and current major version of HyperText Markup Language (HTML) - the standard language used to create and structure content on the web.

In simple terms:

- HTML5 is the language that tells a web browser
- What content to display (text, images, videos, forms, etc.)
- How that content is structured

Applications of Node.js (any three):

- i. Real-time chat applications
- ii. RESTful web services and APIs
- iii. Streaming applications (audio/video)

b) What are metadata elements? Give an example.

Metadata elements are HTML elements that provide information about the web page rather than displaying content directly on the webpage. They are usually placed inside the **<head>** section of an HTML document.

Common Metadata Elements

`<meta>` `<title>` `<link>` `<style>` `<base>`

c) Define flow elements in HTML5.

Flow elements in HTML5 are the elements that are used to structure and display the main content of a webpage.

They are generally placed inside the `<body>` section and appear in the normal flow of the document.

Examples of Flow Elements

- `<p>` – Paragraph
- `<div>` – Division
- `<h1>` to `<h6>` – Headings
- `<ul>` / `<ol>` – Lists
- `<table>`
- `<section>`
- `<article>` - `<img>`

d) What is CSS?

CSS (Cascading Style Sheets) is a stylesheet language used to control the appearance and layout of HTML webpages.

CSS separates the content (HTML) from the presentation (design), making webpages attractive and easier to maintain.

e) What is the use of display property?

The display property in CSS is used to control how an HTML element is displayed on a webpage.

It determines:

- Whether the element appears as a block or inline element
- How elements are arranged
- Whether the element is visible or hidden

f) What is JavaScript?

JavaScript is a high-level scripting language used to make web pages **interactive and dynamic**.

It is mainly used for:

- Validating forms
- Creating animations
- Handling events
- Updating webpage content dynamically
- Building interactive web applications

JavaScript works together with:

HTML → Structure

CSS → Styling

JavaScript → Behavior and interactivity

g) Define array in JavaScript.

An array in JavaScript is a special variable used to store multiple values in a single variable.

Arrays can store: Numbers, Strings, Objects, Mixed data types

Each value in an array is called an **element**, and elements are accessed using an **index** starting from 0.

Example: `let fruits = ["Apple", "Mango", "Orange"];`

h) Define Document Object.

Anonymous Function in TypeScript – Code Snippet: The Document Object is an object in JavaScript that represents the entire HTML document displayed in a web browser.

It is part of the DOM (Document Object Model) and is used to access and manipulate webpage elements such as text, images, forms, and styles.

i) Compare between let and const.

Feature	Let	const
Purpose	Declares a variable	Declares a constant variable
Reassignment	Allowed	Not allowed
Initialization	Can be declared without value	Must be initialized while declaring
Scope	Block scope	Block scope
Value Change	Value can change	Value cannot be reassigned

j) Define spread operator.

The Spread Operator in JavaScript is represented by *three dots (...)*.

It is used to expand or spread elements of an array, object, or iterable into individual elements.

Uses of Spread Operator

Copy arrays

Merge arrays

Copy objects

Pass multiple values to functions

Example

```
let arr1 = [1, 2, 3];
let arr2 = [...arr1, 4, 5];
console.log(arr2);
```

k) What are components in React.js?

Components in React.js are reusable and independent pieces of code used to build the user interface (UI) of a web application.

Each component represents a part of the webpage, such as:

- Header
- Footer
- Button
- Form
- Navigation bar

Components help make code:

- Reusable
- Organized
- Easy to maintain

I) Compare between MERN and MEAN?

Feature	MERN Stack	MEAN Stack
Full Form	MongoDB, Express.js, React.js, Node.js	MongoDB, Express.js, Angular, Node.js
Frontend Technology	React.js	Angular
Language Used	JavaScript	JavaScript / TypeScript
Architecture	Library-based	Framework-based
Learning Curve	Easier and flexible	More structured and complex
Performance	Fast UI rendering using Virtual DOM	Good performance with two-way data binding
Best For	Dynamic single-page applications	Enterprise-level applications

UNIT – I

2. a) Explain the structure of an HTML5 document with example markup code.

An HTML5 document is the basic framework used to create web pages. It contains a set of elements that define the structure, content, and metadata of the webpage. HTML5 introduces semantic elements that improve readability and organization of web content.

The structure of an HTML5 document begins with the `<!DOCTYPE html>` declaration, which tells the browser that the document uses HTML5. The `<html>` element is the root element that encloses the entire webpage.

The document is mainly divided into two sections:

1. Head Section

The `<head>` section contains metadata about the webpage such as:

- Title of the page
- Character encoding
- CSS links
- JavaScript links
- Other meta information

The <title> element inside the head section specifies the title displayed on the browser tab.

2. Body Section

The <body> section contains all visible webpage content such as:

- Headings
- Paragraphs
- Images
- Tables
- Forms
- Links

HTML5 also provides semantic elements like <header>, <section>, <article>, <nav>, and <footer> to organize content meaningfully.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML5 Document</title>
  <meta charset="UTF-8">
</head>

<body>

<header>
  <h1>Welcome to HTML5</h1>
</header>

<section>
  <p>This is a sample HTML5 webpage.</p>
</section>

<footer>
  <p>Copyright 2026</p>
</footer>

</body>
</html>
```

2. b) Explain the form elements and implement a “Student Registration Form”.

HTML forms are used to collect user input such as text, passwords, email addresses, phone numbers, and selections. Forms are created using the <form> element along with different form controls.

Common Form Elements

- <form>** - Defines the form section used to collect user data.
- <input>** - Used to create input fields such as:
 - Text box
 - Password box
 - Radio buttons
 - Checkboxes
 - Submit button
- <label>** - Provides labels for form controls.

<textarea>

- Used for multi-line text input.

<select> and <option>

- Used to create drop-down lists.

<button>

- Creates clickable buttons.

<fieldset> and <legend>

- Used to group related form elements together.

Forms help users enter and submit information to a web server for processing.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
    <title>Student Registration Form</title>
</head>
<body>

<h2>Student Registration Form</h2>
<form>
<label>Student Name:</label>
<input type="text" name="sname"><br><br>

<label>Password:</label>
<input type="password" name="pwd"><br><br>

<label>Email:</label>
<input type="email" name="email"><br><br>
<label>Gender:</label>
<input type="radio" name="gender"> Male
<input type="radio" name="gender"> Female<br><br>

<label>Courses:</label>
<input type="checkbox"> CSE
<input type="checkbox"> ECE
<input type="checkbox"> IT<br><br>

<label>Address:</label><br>
<textarea rows="4" cols="30"></textarea><br><br>

<label>Department:</label>
<select>
    <option>CSE</option>
    <option>ECE</option>
    <option>EEE</option>
</select><br><br>

<input type="submit" value="Register">
<input type="reset" value="Clear">

</form>

</body>
</html>
```

3. a) Discuss the following HTML5 elements for arranging text and provide detailed example markup code.
i) <div> ii) iii) <table> iv)

HTML5 provides several elements to organize and arrange text content on a webpage. Some important elements are <div>, , <table>, and .

i) <div> Element

The <div> element is a block-level container used to group HTML elements together. It is mainly used for page layout and styling using CSS.

Features

- Starts on a new line
- Used to divide sections of a webpage
- Supports CSS styling and JavaScript manipulation

Example Code

```
<!DOCTYPE html>
<html>
<body>

<div style="background-color:lightblue; padding:10px;">
    <h2>Student Information</h2>
    <p>Name: Ravi Kumar</p>
    <p>Department: CSE</p>
</div>

</body>
</html>
```

From the Above Code The <div> groups related content into one section with a background color.

ii) Element

The element is an inline container used to style or manipulate a small portion of text within a line.

Features

- Does not start on a new line - Used for styling specific text - Mostly used with CSS

Example Code:

```
<!DOCTYPE html>
<html>
<body>

    <p>
        Welcome to
        <span style="color:red; font-weight:bold;">HTML5</span>
        Tutorial.
    </p>

</body>
</html>
```

iii) <table> Element

The <table> element is used to display data in rows and columns.

Important Tags

- <table> → Creates table
- <tr> → Table row
- <th> → Table heading
- <td> → Table data

Example Code

```
<!DOCTYPE html>
<html>
<body>

<table border="1">

<tr>
  <th>Name</th>
  <th>Department</th>
</tr>

<tr>
  <td>Ravi</td>
  <td>CSE</td>
</tr>

<tr>
  <td>Sita</td>
  <td>ECE</td>
</tr>

</table>

</body>
</html>
```

The table displays student details in rows and columns.

iv) Element

The element creates an ordered list where list items are displayed with numbers or letters.

Features

Displays items in sequence

Uses for list items

Example Code

```
<!DOCTYPE html>
<html>
<body>

<h3>Steps to Login</h3>

<ol>
  <li>Open Browser</li>
  <li>Enter Website URL</li>
  <li>Enter Username and Password</li>
  <li>Click Login Button</li>
</ol>

</body>
</html>
```

The `<ol>` element displays steps in numbered order.

The HTML5 elements `<div>`, `<span>`, `<table>`, and `<ol>` are important for arranging and organizing webpage content. They help create structured, readable, and visually organized web pages.

3. b) Describe hyperlinks and structure of URL with examples.

A **hyperlink** is a link that connects one webpage to another webpage, document, image, email address, or location within the same page. Hyperlinks allow users to navigate between web resources.

In HTML, hyperlinks are created using the `<a>` (anchor) element.

Syntax of Hyperlink

`<a href="URL">Link Text</a>`

Attributes

- `href` → Specifies the destination address of the link.
- `target="_blank"` → Opens the link in a new tab.

Example Code:

```
<!DOCTYPE html>
<html>
<body>

<h2>Visit OpenAI Website</h2>

<a href="https://www.openai.com">
Click Here
</a>

</body>
</html>
```

When the user clicks “Click Here,” the browser opens the specified website.

Structure of URL

A **URL (Uniform Resource Locator)** is the address used to locate a resource on the Internet.

General Structure of URL

protocol://domain-name/path/file-name

Example URL

<https://www.example.com/products/index.html>

Parts of URL

Part	Description
https	Protocol used for communication
www.example.com	Domain name or website address
/products/	Folder or path
index.html	Webpage or file name

Detailed Example

<https://www.google.com/search?q=html>

Component	Meaning
https	Secure HyperText Transfer Protocol
www.google.com	Domain name
/search	Path
?q=html	Query string parameter

Hyperlinks are used for navigation between web resources, while URLs specify the exact address of those resources on the Internet. Both are essential components of web browsing and web development.

Unit-II

4. a) Explain different types of CSS (inline, internal, external) with example markup code.

CSS (Cascading Style Sheets) is used to style HTML webpages. There are three main types of CSS:

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

Each type applies styles to HTML elements in different ways.

1. Inline CSS

Inline CSS is used to apply styles directly inside an HTML element using the style attribute.

Features

- Applies style to a single element
- Highest priority
- Useful for quick styling

Example Code:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue; text-align:center;">Welcome to HTML5</h1>

</body>
</html>
```

The style is written directly inside the <h1> tag.

2. Internal CSS

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

Features

- Applies styles to a single webpage
- Useful for page-level styling

Example Code:

```
<!DOCTYPE html>
<html>

<head>
<style>

h1 {
    color:red;
    text-align:center;
}

p {
    font-size:18px;
}

</style>
</head>

<body>

<h1>Internal CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Styles for <h1> and <p> are defined inside the <style> element.

3. External CSS

External CSS stores styles in a separate .css file and links it to the HTML document.

Features

- Reusable across multiple webpages
- Makes code cleaner and easier to maintain

Example Code (HTML File)

```
<<!DOCTYPE html>
<html>

<head>
<link rel="stylesheet" href="style.css">
</head>

<body>

<h1>External CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Example Code (CSS File (style.css))

```
h1 {  
    color:green;  
    text-align:center;  
}  
  
p {  
    font-size:20px;  
}
```

The HTML file links the external stylesheet using the <link> element.

Inline CSS is used for single elements, Internal CSS is used for one webpage, and External CSS is used for multiple webpages. External CSS is the most preferred method for large websites because it improves reusability and maintainability.

4. b) Discuss the CSS box model in detail with a diagram and example.

CSS Box Model

The CSS Box Model describes how HTML elements are displayed as rectangular boxes on a webpage. Every HTML element is treated as a box consisting of content, padding, border, and margin.

The box model helps control:

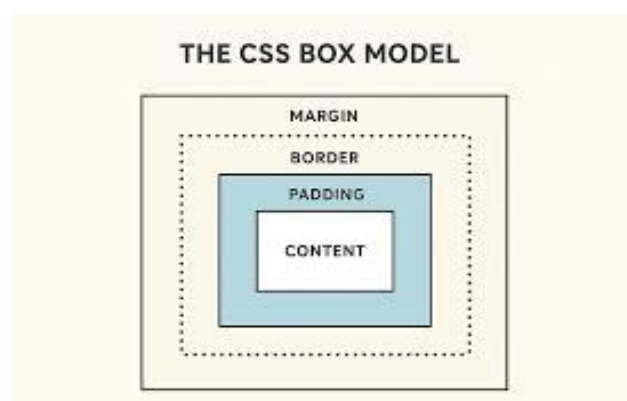
- Element size
- Spacing
- Layout
- Borders

Components of CSS Box Model

The CSS box model contains four parts:

1. Content
2. Padding
3. Border
4. Margin

Diagram of CSS Box Model



Explanation of Components

1. Content

The actual text, image, or data displayed inside the element.

Example:

```
width: 200px;
height: 100px;
```

2. Padding

- Space between the content and border
- Padding increases the inner spacing of the element.

Example:

```
padding: 20px;
```

3. Border

- A line surrounding the padding and content.

Example:

```
border: 2px solid black;
```

4. Margin

- Space outside the border used to separate elements from other elements.

Example:

```
margin: 15px;
```

Example Code (CSS File (style.css))

```
<!DOCTYPE html>
<html>

<head>
<style>

div {
    width: 200px;
    padding: 20px;
    border: 5px solid blue;
    margin: 25px;
    background-color: lightgray;
}

</style>
</head>

<body>

<div>
This is CSS Box Model Example
</div>

</body>
</html>
```

5. a) Discuss about various CSS3 selectors used with suitable example code.

CSS3 selectors are used to select and style HTML elements on a webpage. They help apply styles efficiently to specific elements based on names, classes, IDs, attributes, and positions. CSS3 provides different types of selectors for flexible webpage styling.

1. Element Selector

- The element selector selects HTML elements based on the tag name.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
<style>

p {
    color: blue;
}

</style>
</head>

<body>

<p>This is a paragraph.</p>

</body>
</html>
```

All <p> elements are displayed in blue color.

2. ID Selector

- The ID selector selects a specific element using the id attribute. It uses #.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
<style>

#title {
    color: red;
}

</style>
</head>

<body>

<h1 id="title">Welcome</h1>

</body>
</html>
```

The style applies only to the element with id="title".

3. Class Selector

- The class selector selects elements using the class attribute. It uses ..

Example Code:

```
<!DOCTYPE html>
<html>
<head>
<style>

.note {
    background-color: yellow;
}

</style>
</head>

<body>

<p class="note">Important Message</p>

</body>
</html>
```

All elements with class note get yellow background color.

4. Universal Selector

- The universal selector selects all HTML elements.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
<style>

* {
    font-family: Arial;
}

</style>
</head>

<body>

<h1>Heading</h1>
<p>Paragraph</p>

</body>
</html>
```

The font style applies to all elements.

5. Group Selector

- The group selector applies the same style to multiple elements.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
<style>

h1, p {
    color: green;
}

</style>
</head>

<body>

<h1>Heading</h1>
<p>Paragraph</p>

</body>
</html>
```

Both <h1> and <p> elements appear in green color.

6. Attribute Selector

- Selects elements based on attributes.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
<style>

input[type="text"] {
    background-color: lightblue;
}

</style>
</head>

<body>

<input type="text">

</body>
</html>
```

Styles apply only to text input fields.

7. Descendant Selector

- Selects elements inside another element.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
<style>

div p {
    color: purple;
}

</style>
</head>

<body>

<div>
    <p>Inside Div</p>
</div>

<p>Outside Div</p>

</body>
</html>
```

Only paragraphs inside <div> are styled.

8. Pseudo-class Selector

- Used to define special states of elements.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
<style>

a:hover {
    color: red;
}

</style>
</head>

<body>

<a href="#">Move Mouse Here</a>

</body>
</html>
```

The link color changes when the mouse pointer moves over it.

9. Pseudo-element Selector

- Styles specific parts of an element.

Example Code:

```
<!DOCTYPE html>
<html>
<head>
<style>

p::first-letter {
    font-size: 30px;
    color: red;
}

</style>
</head>

<body>

<p>Welcome to CSS3</p>

</body>
</html>
```

The first letter of the paragraph becomes larger and red.

CSS3 selectors help developers efficiently target and style HTML elements. They improve webpage design, maintainability, and flexibility by allowing precise selection of webpage elements.

5. b) Discuss array and functions in JavaScript and implement JavaScript application to find maximum and minimum elements in array using functions.

Arrays in JavaScript

An array is a special variable used to store multiple values in a single variable. Array elements are stored using indexes starting from 0.

Arrays can store:

- Numbers
- Strings
- Objects
- Mixed data types

Features of Arrays

- Easy storage of multiple values
- Access elements using index values
- Dynamic in size

Example

```
let numbers = [10, 20, 30, 40];
```

Functions in JavaScript

A function is a block of reusable code designed to perform a specific task. Functions help reduce code repetition and improve modularity.

Features of Functions

- Reusable
- Improves readability
- Reduces code duplication

Syntax

```
function functionName() {  
    // code  
}
```

JavaScript Application to Find Maximum and Minimum Elements in Array Using Functions

Example Code:

```
<!DOCTYPE html>  
<html>  
  
<head>  
    <title>Maximum and Minimum in Array</title>  
</head>  
<body>  
  
    <h2>Find Maximum and Minimum Elements</h2>  
<script>  
  
let arr = [25, 10, 45, 5, 60, 15];  
  
function findMaximum(array) {  
    let max = array[0];  
    for(let i = 1; i < array.length; i++) {  
        if(array[i] > max) {  
            max = array[i];  
        }  
    }  
    return max;  
}  
function findMinimum(array) {  
    let min = array[0];  
    for(let i = 1; i < array.length; i++) {  
        if(array[i] < min) {  
            min = array[i];  
        }  
    }  
    return min;  
}  
document.write("Array Elements: " + arr + "<br><br>");  
  
document.write("Maximum Element: " + findMaximum(arr) + "<br>");  
  
document.write("Minimum Element: " + findMinimum(arr));  
  
</script>  
  
</body>  
</html>
```

Unit-III

6. a) Explain DOM structure and how it represents an HTML document.

The DOM (Document Object Model) is a programming interface that represents an HTML document as a tree structure. It allows JavaScript to access, modify, add, or delete HTML elements dynamically.

When a webpage is loaded, the browser converts the HTML document into a DOM tree where each element becomes an object called a node.

DOM Structure

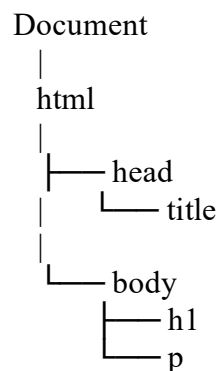
The DOM represents the HTML document in a hierarchical tree format.

Example HTML Document

```
<!DOCTYPE html>
<html>
<head>
  <title>DOM Example</title>
</head>

<body>
  <h1>Welcome</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

DOM Tree Representation



Explanation of DOM Structure

Node	Description
Document	Root of the webpage
<html>	Main HTML element
<head>	Contains metadata
<title>	Page title
<body>	Visible webpage content
<h1> and <p>	Child elements inside body

Each HTML tag becomes an element node in the DOM tree.

How DOM Represents an HTML Document

1. Browser reads the HTML document.
2. It converts HTML elements into objects (nodes).
3. These nodes are arranged in a tree structure.
4. JavaScript uses the DOM to interact with elements.

Using DOM, JavaScript can:

- Change text content
- Modify styles
- Add or remove elements
- Handle events

Example Using DOM

```
<!DOCTYPE html>
<html>

<body>

<h1 id="title">Hello</h1>

<script>

document.getElementById("title").innerHTML = "Welcome to DOM";

</script>

</body>
</html>
```

- document.getElementById() accesses the HTML element.
- innerHTML changes the text dynamically.

Advantages of DOM

- ✓ Enables dynamic webpage updates
- ✓ Allows interaction between HTML and JavaScript
- ✓ Supports event handling
- ✓ Makes webpages interactive

The DOM represents an HTML document as a hierarchical tree of objects. It enables JavaScript to access and manipulate webpage content dynamically, making modern web applications interactive and responsive.

6. b) Discuss about Event handling mechanism in JavaScript with example code.

Event handling in JavaScript is the process of responding to user actions or browser activities on a webpage. These actions are called **events**.

Examples of events:

- Mouse click
- Keyboard press
- Mouse movement
- Page loading
- Form submission

JavaScript uses event handlers to execute code when an event occurs.

Common Events in JavaScript

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 webpage loads
Onsubmit	Occurs when form is submitted

Event Handling Mechanism

The event handling process works as follows:

1. User performs an action.
2. Browser detects the event.
3. JavaScript event handler executes associated function.
4. Required action is performed.

Syntax of Event Handling

`<element event="functionName()">`

Example Using DOM

```
<!DOCTYPE html>
<html>

<head>
  <title>Event Handling Example</title>

  <script>

    function showMessage() {

      alert("Button Clicked Successfully");

    }

  </script>

</head>

<body>

  <h2>JavaScript Event Handling</h2>

  <button onclick="showMessage()" ">
    Click Here
  </button>

</body>
</html>
```

Advantages of Event Handling

- Makes webpages interactive
- Improves user experience
- Enables dynamic webpage behavior
- Supports real-time user interaction

7. a) Explain Node, Document, and Element interfaces in detail.

In **TypeScript**, *function parameters* can be defined in different ways to make functions flexible and type-safe. Below are **three commonly used types of functional parameters**, explained clearly with **example code**.

Node, Document, and Element Interfaces in JavaScript DOM

The DOM (Document Object Model) represents an HTML document as a tree structure of objects. In the DOM, Node, Document, and Element are important interfaces used to access and manipulate webpage content dynamically.

1. Node Interface

The **Node** interface is the base interface for all objects in the DOM tree. Every part of an HTML document such as elements, text, attributes, and comments is considered a node.

Types of Nodes

- Element Node
- Text Node
- Attribute Node
- Comment Node

Features of Node Interface

- Represents each object in the DOM tree
- Provides properties and methods for navigation
- Used to access parent, child, and sibling nodes

Common Node Properties

Property	Description
nodeName	Returns the name of the node
nodeType	Returns the type of node
nodeValue	Returns node value

Common Node Methods

Method	Description
appendChild()	Adds a child node
removeChild()	Removes a child node
cloneNode()	Creates a copy of a node

Example Code:

```
<p id="demo">Hello</p>

<script>
let x = document.getElementById("demo");

document.write(x.nodeName);
</script>
```

2. Document Interface

The Document interface represents the entire HTML document loaded in the browser. It is the root object of the DOM.

JavaScript accesses it using:

Document

Features of Document Interface

- Represents the whole webpage
- Provides methods to access HTML elements
- Allows modification of webpage content

Common Document Methods

Method	Description
getElementById()	Selects element by ID
getElementsByTagName()	Selects elements by tag name
querySelector()	Selects first matching element
write()	Writes content to webpage

Example Code:

```
<h1 id="title">Welcome</h1>

<script>

document.getElementById("title").innerHTML = "Hello HTML5";

</script>
```

The document interface accesses the <h1> element and changes its text.

3. Element Interface

The **Element** interface represents an individual HTML element in the DOM tree.

Each HTML tag such as <p>, <div>, <h1>, etc., is an element object.

Features of Element Interface:

- Accesses specific HTML elements
- Allows style and content modification
- Supports attribute manipulation

Common Element Properties

Property	Description
innerHTML	Gets or sets HTML content
style	Accesses CSS styles
className	Accesses class name

Common Element Methods

Method	Description
setAttribute()	Sets attribute value
getAttribute()	Gets attribute value
removeAttribute()	Removes attribute

Example Code:

```
<p id="para">JavaScript Example</p>

<script>

let p = document.getElementById("para");

p.style.color = "blue";

</script>
```

The element interface changes the paragraph text color to blue.

Difference Between Node, Document, and Element

Interface	Description
Node	Base interface for all DOM objects
Document	Represents entire HTML document
Element	Represents a specific HTML element

Node, Document, and Element interfaces are core parts of the DOM. They allow JavaScript to access, navigate, and manipulate webpage content dynamically, making webpages interactive and responsive.

7. b) Explain arrow function, template literals with example code.

JavaScript ES6 introduced several modern features to make coding simpler and more efficient. Two important features are Arrow Functions and Template Literals.

Arrow Function

An Arrow Function is a shorter and simpler way to write functions in JavaScript. It uses the => symbol.

Syntax:

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

Features of Arrow Functions

- Shorter syntax
- Improves readability
- Useful for simple functions
- Does not have its own this keyword

Example Program

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

Short Form of Arrow Function

```
const square = x => x * x;  
console.log(square(5));
```

Template Literals

Template literals are used to create strings easily using backticks (` `) instead of single or double quotes.

They support:

- Multi-line strings
- String interpolation
- Embedded expressions

Syntax:

```
`string text`
```

Example Program

```
let name = "Ravi";  
let marks = 85;  
  
console.log(`Student Name: ${name}  
Marks: ${marks}`);
```

- Backticks are used for template literals.
- \${} inserts variable values into the string.

Advantages of Template Literals

- Easy string formatting
- Supports multi-line text
- Simplifies variable embedding

Difference Between Normal Function and Arrow Function

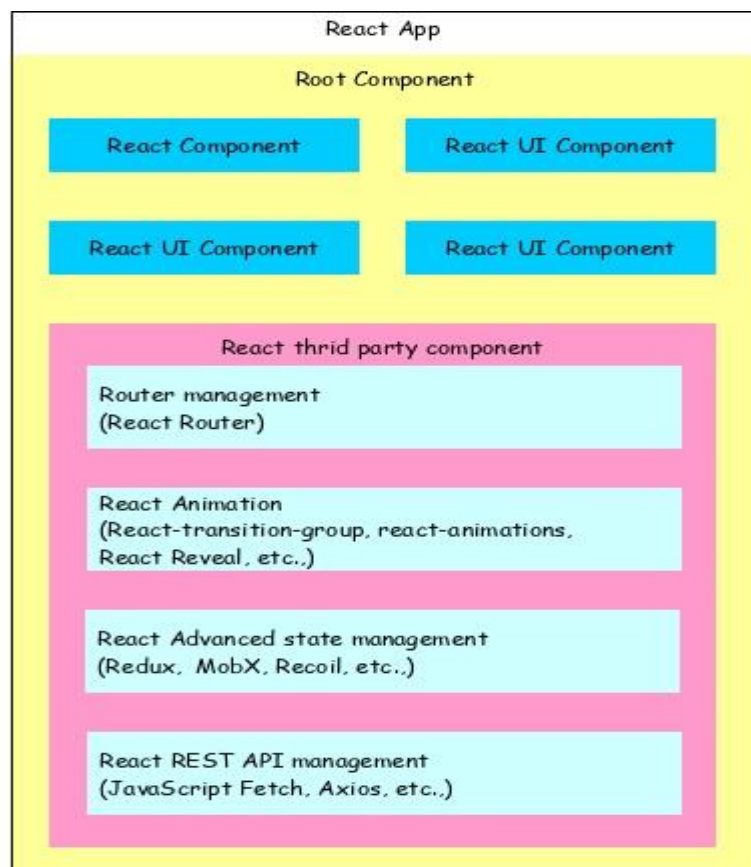
Normal Function	Arrow Function
Uses function keyword	Uses => symbol
Longer syntax	Shorter syntax
Has its own this	Does not have its own this

Arrow functions provide a concise way to write functions, while template literals simplify string creation and formatting. Both features improve JavaScript code readability and efficiency.

Unit-IV

8. a) Explain the architecture and working of a basic React.js application.

React library is just UI library and it does not enforce any particular pattern to write a complex application. Developers are free to choose the design pattern of their choice. React community advocates certain design pattern. One of the patterns is Flux pattern. React library also provides lot of concepts like Higher Order component, Context, Render props, Refs etc., to write better code. React Hooks is evolving concept to do state management in big projects. Let us try to understand the high level architecture of a React application.



- React app starts with a single root component.
- Root component is build using one or more component.
- Each component can be nested with other component to any level.
- Composition is one of the core concepts of React library. So, each component is build by composing smaller components instead of inheriting one component from another component.
- Most of the components are user interface components.
- React app can include third party component for specific purpose such as routing, animation, state management, etc.

Workflow of a React application

Let us understand the workflow of a React application in this chapter by creating and analyzing a simple React application.

Open a command prompt and go to your workspace.

```
cd /go/to/your/workspace
```

Next, create a folder, *static_site* and change directory to newly created folder.

```
mkdir static_site  
cd static_site
```

Example

Next, create a file, *hello.html* and write a simple React application.

Example Code:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8" />  
    <title>React Application</title>  
  </head>  
  <body>  
    <div id="react-app"></div>  
    <script type="module">  
      import React from 'https://esm.sh/react';  
      import ReactDOM from 'https://esm.sh/react-dom/client';  
  
      function App() {  
        return React.createElement('h1', null, 'Hello React!');  
      }  
  
      const root = ReactDOM.createRoot(document.getElementById('react-app'));  
      root.render(React.createElement(App));  
    </script>  
  </body>
```

Next, serve the application using serve web server.

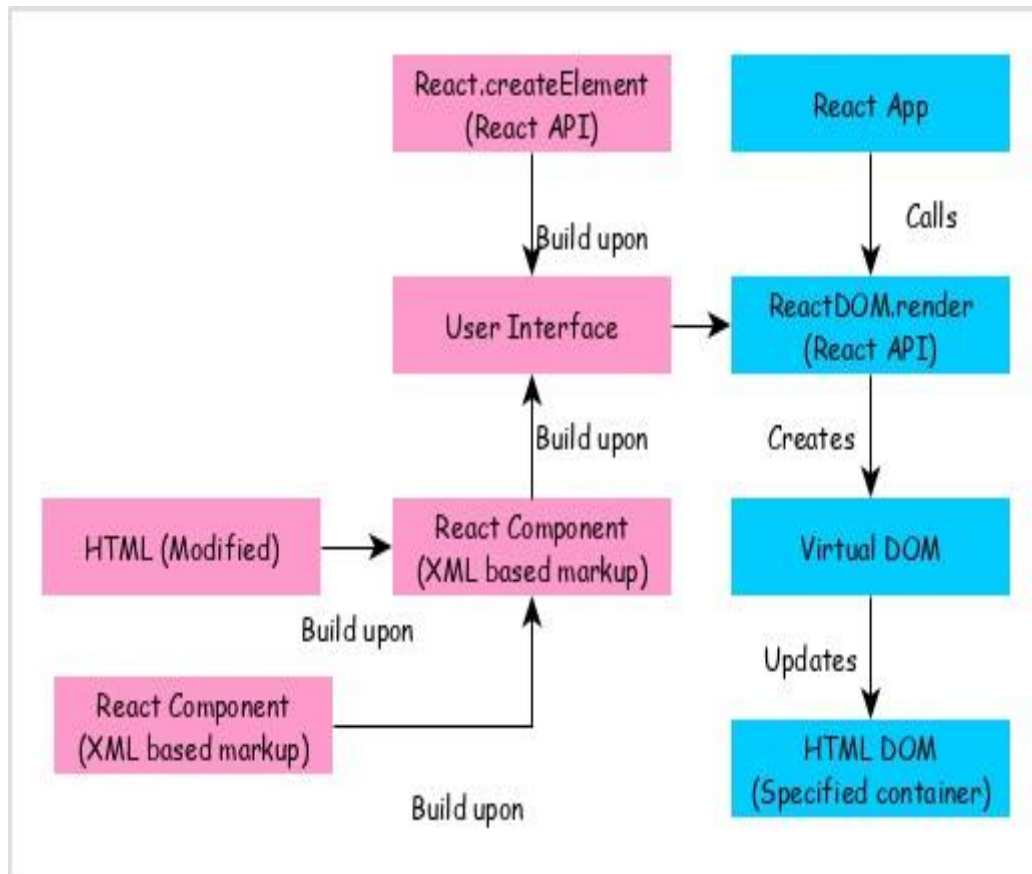
```
serve ./hello.html
```

Output

Next, open your favorite browser. Enter **http://localhost:3000** in the address bar and then press enter.

Hello React!

By analyzing the application, we can visualize the workflow of the React application as shown in the below diagram.



React app calls **ReactDOM.render** method by passing the user interface created using React component (coded in either JSX or React element format) and the container to render the user interface.

ReactDOM.render processes the JSX or React element and emits Virtual DOM.

Virtual DOM will be merged and rendered into the container.

8. b) Discuss advantages and disadvantages of React.js.

React.js is a popular JavaScript library used for building user interfaces, especially single-page applications. It provides fast rendering and reusable components for modern web development.

Advantages of React.js

1. Component-Based Architecture

React applications are built using reusable components, which makes code modular and easy to maintain.
Benefit

- Reusability of code
- Easier development and maintenance

2. Virtual DOM for Fast Performance

React uses a Virtual DOM to update only changed elements instead of reloading the entire webpage.
Benefit

- Faster rendering
- Improved application performance

3. Reusable Components

Components can be reused across multiple parts of the application.
Benefit

- Reduces code duplication
- Saves development time

4. Easy to Learn

React is easier to understand compared to large frameworks because it mainly focuses on the UI layer.
Benefit

- Beginner-friendly
- Simple syntax using JSX

5. Strong Community Support

React is maintained by Meta Platforms and has a large developer community.
Benefit

- Large number of tutorials and libraries
- Better support and updates

6. One-Way Data Binding

Data flows in one direction from parent to child components.

Benefit

- Better control over application data
- Easier debugging

7. SEO Friendly

React supports server-side rendering, which improves search engine optimization.

Disadvantages of React.js

1. Only UI Library

React handles only the user interface and not complete application development.

Limitation

Developers need additional libraries for:

- Routing
- State management
- API handling

2. JSX Complexity

JSX combines HTML and JavaScript together, which may confuse beginners initially.

3. Frequent Updates

React technologies and tools change frequently.

Limitation

Developers need continuous learning.

4. Large Application Setup

For large projects, configuring tools and libraries may become complex.

5. Learning Curve for Advanced Concepts

Advanced topics such as:

- Hooks
- State management
- Lifecycle methods

may require extra learning effort.

React.js is a powerful and efficient library for building modern web applications. Its component-based architecture and Virtual DOM improve performance and code reusability. However, it also has some limitations such as dependency on additional libraries and frequent updates.

9. a) Explain props and state in React.js with examples.

In React.js, **Props** and **State** are important concepts used to manage and transfer data between components.

Props in React.js

Props (Properties) are used to pass data from one component to another component, usually from parent to child.

Features of Props

- Read-only
- Passed as arguments to components
- Used for component communication
- Cannot be modified by child component

Example of Props

```
function Student(props) {  
  return <h2>Student Name: {props.name}</h2>;  
}  
  
function App() {  
  return <Student name="Ravi" />;  
}  
  
export default App;
```

- name="Ravi" is passed as a prop.
- props.name receives and displays the value.

State in React.js

State is used to store and manage dynamic data within a component. State values can change during program execution.

Features of State

- Mutable (can change)
- Managed inside component
- Used for dynamic updates
- Re-renders component when updated

Example of State:

```
import React, { useState } from 'react';  
  
function Counter() {  
  const [count, setCount] = useState(0);  
  return (  
    <div>
```

```

    <h2>Count: {count}</h2>

    <button onClick={() => setCount(count + 1)}>
      Increment
    </button>
  </div>
);
}

```

export default Counter;

- useState(0) creates state variable count.
- setCount() updates the state.
- Clicking button increases count value.

Difference Between Props and State

Props	State
Used to pass data between components	Used to manage component data
Read-only	Can be modified
Passed from parent component	Managed within component
Immutable	Mutable
Used for communication	Used for dynamic updates

Props and State are core concepts in React.js. Props are used for passing data between components, while State is used to manage dynamic data within a component. Together, they help build interactive and reusable React applications.

9. b) Develop a React application for a menu component with explanation.

A Menu component in React.js is used to display navigation links or menu items in a webpage or web application. React components make menus reusable, interactive, and easy to maintain.

In this application, a simple menu is created using React functional components. The menu contains different navigation items such as Home, About, Services, and Contact.

React Application – Menu Component

Cont... in Next Page..

Example Code (App.js)

```
import React from 'react';
import './App.css';

function Menu() {
  return (
    <div className="menu">
      <h1>Website Menu</h1>

      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </div>
  );
}

function App() {
  return (
    <div>
      <Menu />
    </div>
  );
}

export default App;
```

Example Code (App.css)

```
.menu {
  width: 300px;
  margin: 50px auto;
  padding: 20px;
  border: 2px solid black;
  text-align: center;
  background-color: lightblue;
}

.menu ul {
  list-style-type: none;
  padding: 0;
}

.menu li {
  margin: 10px 0;
}

.menu a {
  text-decoration: none;
  color: darkblue;
  font-size: 18px;
  font-weight: bold;
}

.menu a:hover {
  color: red;
}
```

Explanation of the Program

1. Import React

```
import React from 'react';
```

This statement imports the React library.

2. Menu Component

```
function Menu()
```

The Menu component is a functional component that displays menu items.

3. Unordered List

```
<ul>
```

The unordered list is used to create navigation menu items.

4. Anchor Tags

```
<a href="#">Home </a>
```

Anchor tags create clickable menu links.

5. CSS Styling

The CSS file styles the menu using:

- Background color
- Text alignment
- Hover effects
- Spacing

Output

Website Menu

```
Home
About
Services
Contact
```

The menu items are displayed in a styled navigation box.

Advantages of Menu Component

- Reusable component
- Easy navigation
- Better user interface
- Easy maintenance
- Supports styling and interactivity

The React Menu component demonstrates how reusable components can be used to create navigation systems in React applications. React makes UI development modular, efficient, and easy to manage.
