

	vulnerabilities inherent in common web implementations.
Unit I: (Theory) 15 Hours	Introduction to Web Technology and Web Designing <ul style="list-style-type: none"> • Web Technology: HTTP; System Architecture of a Web server; Client-side Scripting versus Server-side Scripting. • Introduction to HTML: What is HTML-HTML Documents- Basic structure of an HTML document. CSS: What is CSS, Structure of CSS. Advantages of CSS. • Javascrpts: What is JavaScript? -Client-Side JavaScript -Advantages of JavaScript-Limitations of JavaScript.
UNIT-II: (Practical) 30 Hours	Hyper Text Markup Language (HTML5) <p>1. HTML5 Basics: Structure of an HTML5 document (<!DOCTYPE html>, <html>, <head>, <title>, <body>), Semantic elements (<header>, <nav>, <section>, <article>, <footer>, etc.)</p> <p>2. Text and Multimedia: Text formatting (headings, paragraphs, emphasis, etc.), Adding images (tag) and multimedia content (<video>, <audio> tags), Using HTML entities for special characters</p> <p>3. Links, Lists, and Tables: Creating hyperlinks (<a> tag) and anchor links, Lists (unordered , ordered , and definition <dl> lists), Creating tables (<table>, <tr>, <th>, <td>)</p> <p>4. Forms and Input Elements: Building forms (<form> tag) with various input types (text, password, email, etc.), Radio buttons, checkboxes, and dropdown lists, Form validation using HTML5 attributes (required, pattern, min/max, etc.)</p> <p>5. Media and Embedding: Embedding multimedia content (videos, audio) from external sources, Using the <iframe> tag for embedding content from other websites</p> <p>6. HTML5 APIs : Geolocation API for obtaining user location, Canvas API for drawing graphics and animations, Local Storage and Session Storage for client-side data storage</p> <p>7. Accessibility and SEO: Importance of semantic HTML for accessibility and SEO, Using ARIA attributes for enhancing accessibility, Optimizing HTML for search engines (meta tags, title tags, alt attributes)</p> <p>8. Responsive Design and Mobile Compatibility: Creating responsive layouts using HTML5 and CSS3,</p>

	<p>Meta viewport tag for mobile responsiveness, Mobile-friendly forms and input elements</p> <p>9. Advanced HTML5 Features: Web components and custom elements, Drag and drop functionality, Web storage (local Storage, session Storage)</p>
Suggested Practical Assignment:	<p>1. Create a Web Page Structure: Design a web page structure using HTML5 semantic elements such as <header>, <nav>, <section>, <article>, <footer>, and <aside>.</p> <p>2. Create a web page for a cake shop to display all the different types of cakes and price to choose from.</p> <p>3. Multimedia Embedding: Embed an audio or video file using the <audio> or <video> tag with appropriate attributes like controls, autoplay, and loop.</p> <p>4. Responsive Image Gallery: Build a responsive image gallery using HTML5 <figure> and <figcaption> elements. Ensure that the gallery adjusts smoothly on different screen sizes.</p> <p>5. Interactive Form Validation: Develop an HTML5 form with input fields like text, email, password, and a submit button. Implement HTML5 form validation using attributes like required, pattern, and min/max.</p> <p>6. Create a HTML page with controls to take data for a College Admission with all the proper validations in the form.</p> <p>7. Geolocation API Integration: Implement the HTML5 Geolocation API to display the user's current location on a map or show nearby places based on latitude and longitude.</p> <p>8. Local Storage Usage: Create a web page that allows users to store data locally using HTML5 localStorage or session Storage. Develop functionality to add, edit, and delete stored items.</p> <p>9. Create a HTML Page to display the number of the times the web page was visited using local storage.</p> <p>10. Semantic Markup for SEO: Optimize an existing web page for search engines using semantic HTML5 tags. Use <header>, <nav>, <main>, <article>, <section>, <aside>, and <footer> tags appropriately.</p>
UNIT-III: (Practical) 30 Hours	<p>Cascading Style Sheets (CSS)</p> <p>1. Introduction to CSS: What is CSS? Importance and benefits, CSS syntax: selectors, properties, and values, External, internal, and inline CSS</p> <p>2. CSS Selectors and Specificity: Basic selectors: element selectors, class selectors, ID</p>

	<p>selectors, Combinators: descendant, child, adjacent sibling, general sibling, Pseudo classes and pseudo-elements, CSS specificity and inheritance</p> <p>3. CSS Box Model: Understanding the box model: content, padding, border, margin, Box sizing: content-box vs. border-box, Margin collapsing</p> <p>4. Layout and Positioning: Display property: block, inline, inline-block, flex, grid, Position property: static, relative, absolute, fixed, sticky, Floats and clearing floats, CSS Grid and Flexbox layouts</p> <p>5. Typography and Fonts: Font properties: font-family, font-size, font-weight, font-style, line-height, Text properties: color, text-align, text-decoration, text-transform, letter spacing, word-spacing, Google Fonts and custom font usage</p> <p>6. Colors and Backgrounds :Color values: named colors, hexadecimal, RGB, RGBA, HSL, HSLA, Background properties: background-color, background-image, background repeat, background-position, background-size</p> <p>7. Responsive Design and Media Queries: Responsive design principles, Media queries syntax and usage, Designing responsive layouts for different screen sizes (mobile-first approach)</p> <p>8. CSS Transitions and Animations: Transition properties: transition-property, transition duration, transition-timing-function, transition-delay, CSS animations: keyframes, animation properties, animation-duration, animation-timing-function, animation-delay</p> <p>9. Flexbox and Grid Layouts: Flexbox properties: flex-direction, justify-content, align-items, align-self, flex-grow, flex-shrink, CSS Grid properties: grid-template-columns, grid-template-rows, grid-gap, grid-template-areas</p> <p>10. CSS Frameworks and Preprocessors: Introduction to CSS frameworks (Bootstrap, Tailwind), Overview of CSS preprocessors (Sass): variables, mixins, nesting, inheritance</p> <p>11. Advanced CSS Techniques: Transformations: translate, rotate, scale, skew, CSS variables (custom properties), CSS gradients, shadows, and filters, Cross-browser compatibility and vendor prefixes</p>
Suggested Practical on the topics	<p>1. CSS Selectors and Box Model:</p> <ul style="list-style-type: none"> ○ Create a webpage with different elements styled using basic selectors, class selectors, and ID selectors. Apply different properties such as background color,

	<p>padding, border, and margin to understand the box model.</p> <p>2. Layout and Positioning:</p> <ul style="list-style-type: none"> ○ Design a web page layout using CSS display properties (e.g., flexbox or grid) for header, navigation, content, and footer sections. Use positioning (static, relative, absolute) to position elements within the layout. <p>3. Typography and Fonts:</p> <ul style="list-style-type: none"> ○ Style text on a webpage with different font families, sizes, weights, styles, colors, and text alignments. Experiment with line height, letter spacing, and text decorations. <p>4. Colors and Backgrounds:</p> <ul style="list-style-type: none"> ○ Create a webpage with various background colors, gradients, images, and patterns. Apply different background properties such as background-size, background-position, and background-repeat. <p>5. Responsive Design with Media Queries:</p> <ul style="list-style-type: none"> ○ Develop a responsive webpage that adjusts its layout and styling based on different screen sizes using media queries. Test the responsiveness on mobile devices and desktop screens. <p>6. CSS Transitions and Animations:</p> <ul style="list-style-type: none"> ○ Add transitions to elements (e.g., hover effects) using CSS transition properties (transition-duration, transition-property, transition-timing-function). simple animations using keyframes and animation properties. <p>7. Flexbox and Grid Layouts:</p> <ul style="list-style-type: none"> ○ Design a webpage layout using CSS Flexbox properties (flex-direction, justify-content, align-items) for a navigation menu or card-based layout. Create a grid based layout using CSS Grid properties (grid-template-columns, grid-template-rows, grid-gap). <p>8. Customizing CSS Frameworks:</p> <ul style="list-style-type: none"> ○ Customize a CSS framework (e.g., Bootstrap) by modifying variables, adding custom styles, and overriding default styles to create a unique design. <p>9. Advanced CSS Techniques:</p> <ul style="list-style-type: none"> ○ Implement CSS transformations (translate, rotate, scale, skew) on elements to create interactive effects. Use CSS gradients, shadows, and filters to enhance visual elements. ○ Optimize CSS code by minifying, concatenating, and compressing stylesheets. <p>Use browser developer tools to debug and optimize CSS for performance.</p>
--	---

<p>UNIT-IV: (Practical) 30 Hours</p>	<p>Java Scripts</p> <p>1. JavaScript Basics :JavaScript syntax: variables, data types, operators, expressions, statements, Functions: defining functions, function expressions, arrow functions, Control flow: if statements, switch statements, loops (for, while)</p> <p>2. Arrays and Objects: Arrays: creating arrays, accessing elements, array methods (push, pop, shift, unshift, slice, splice), Objects: creating objects, object properties, methods, constructor functions, prototypes</p> <p>3. DOM Manipulation: Accessing DOM elements: get Element ById, querySelector, querySelectorAll, Manipulating DOM elements: changing content, styles, attributes, adding/removing elements</p> <p>4. Events and Event Handling: click, mouseover, keydown, submit, etc. Event listeners: adding event listeners, event propagation (bubbling, capturing) Handling user interactions with events</p> <p>5. Forms and Validation: Working with HTML forms in JavaScript, Form validation: validating input fields, displaying error messages, preventing default form submission</p> <p>6. Error Handling: Handling errors in JavaScript: try-catch blocks, Debugging JavaScript code using browser developer tools</p>
<p>Suggested Practical Assignments</p>	<p>1. Basic JavaScript Concepts:</p> <ul style="list-style-type: none"> ○ Write JavaScript code to declare variables of different data types (string, number, boolean). ○ Implement arithmetic operations, comparison operators, and logical operators in JavaScript. <p>2. Functions and Control Flow:</p> <ul style="list-style-type: none"> ○ Create a function to calculate the factorial of a number using recursion. ○ Write a JavaScript program to check if a number is prime or not using a function. ○ Write a Javascript program to print all the perfect numbers from 1 to n. <p>3. Arrays and Objects:</p> <ul style="list-style-type: none"> ○ Create an array of numbers and write JavaScript code to find the sum, average, maximum, and minimum value in the array. ○ Define an object representing a person with properties like name, age, and country. Use object methods to display information about the person. <p>4. DOM Manipulation and Events:</p> <ul style="list-style-type: none"> ○ Build an HTML form with input fields for username and password. Use JavaScript to validate the form on submission and display appropriate messages.

	<ul style="list-style-type: none"> ○ Create a webpage with a button that changes the background color of a div element when clicked using event handling. <p>5. Project-Based Assignments:</p> <ul style="list-style-type: none"> ○ Choose a project idea (e.g., interactive quiz, weather app, budget tracker) and implement it using JavaScript. Use concepts learned throughout the syllabus to build the project.
Suggested Readings	<ol style="list-style-type: none"> 1. David Flanagan, "JavaScript: The Definitive Guide" by, O'Reilly Media. 2022. 8th Edition 2. Elizabeth Castro and Bruce Hyslop, "HTML and CSS: Visual QuickStart Guide", Peachpit Press, 9th Edition 3. Jennifer Niederst Robbins, "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics". 4. Marijn Haverbeke, "Eloquent JavaScript: A Modern Introduction to Programming".
Requirements	<ul style="list-style-type: none"> • Computers • Software • Internet Access • External Storage • Printers and Scanners • Projector and Screens • Any other item as required
Qualified Instructors	<ul style="list-style-type: none"> • Instructors with experience in Web Designing and teaching. • Certifications or relevant qualifications in Web Designing