



**EAST  
POINT COLLEGE OF ENGINEERING &  
TECHNOLOGY**

‘Jnana Prabha’, Virgo Nagar Post, Bengaluru-560049

**Department of Computer Science and Engineering**

**Academic Year: 2024-25**

**LABORATORY MANUAL**

**Semester**

**: V**

**Subject**

**: Web Technology Laboratory**

**Subject Code**

**: BCSL504**

**NAME:** \_\_\_\_\_

**USN:** \_\_\_\_\_

**SECTION:** \_\_\_\_\_



CamScanner

### **PROGRAM OUTCOMES**

**PO1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2: Problem analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**PO6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9: Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: Project management and finance:** Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12: Life-long learning:** Recognize the need for and have the preparation and ability to



**Department of Computer Science and Engineering**

**INSTITUTE VISION AND MISSION**

**VISION**

The East Point College of Engineering and Technology aspires to be a globally acclaimed institution, recognized for excellence in engineering education, applied research and nurturing students for holistic development.

**MISSION**

**M1:** To create engineering graduates through quality education and to nurture innovation, creativity and excellence in teaching, learning and research

**M2:** To serve the technical, scientific, economic and societal developmental needs of our communities

**M3:** To induce integrity, teamwork, critical thinking, personality development and ethics in students and to lay the foundation for lifelong learning



**Department of Computer Science and Engineering**

**DEPARTMENT VISION AND MISSION**

**VISION**

The department aspires to be a Centre of excellence in Computer Science & Engineering to develop competent professionals through holistic development.

**MISSION**

**M1:** To create successful Computer Science Engineering graduates through effective pedagogies, the latest tools and technologies, and excellence in teaching and learning.

**M2:** To augment experiential learning skills to serve technical, scientific, economic, and social developmental needs.

**M3:** To instil integrity, critical thinking, personality development, and ethics in students for a successful career in Industries, Research, and Entrepreneurship.

**PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

**PEO 1:** To produce graduates who can perform technical roles to contribute effectively in software industries and R&D Centre.

**PEO 2:** To produce graduates having the ability to adapt and contribute in key domains of computer science and engineering to develop competent solutions.

**PEO 3:** To produce graduates who can provide socially and ethically responsible solutions while adapting to new trends in the domain to carve a successful career in the industry.

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

**PSO1:** To conceptualize, model, design, simulate, analyse, develop, test, and validate computing systems and solve technical problems arising in the field of computer science & engineering.

**PSO2:** To specialize in the sub-areas of computer science & engineering systems such as cloud computing, Robotic Process Automation, cyber security, big data analytics, user interface design, and IOT to meet industry requirements.

**PSO3:** To build innovative solutions to meet the demands of the industry using appropriate

tools and techniques.

### **COURSE LEARNING OBJECTIVES**

- To learn HTML 5 elements and their use.
- To use CSS for enhanced user interface presentation.
- Gain knowledge of JavaScript, AJAX and jQuery for dynamic presentation.
- Use of PHP to build Web applications.
- Design and develop Websites and Web applications.

### **COURSE OUTCOMES**

At the end of the course the student will be able to:

**CO1:** Design web pages for the given problem using HTML, Javascript and CSS.

**CO2** Develop the solution for the given real-world problem using jQuery, Ajax and PHP.

**CO3:** Analyze the results and produce substantial written documentation.

Web Technology Lab		Semester	5
Course Code	BCSL504	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	0:0:2:0	SEE Marks	50
Credits	01	Exam Hours	2
Examination type (SEE)	Practical		
Sl. No	Experiments		
1	<p>Develop the HTML page named as “Myfirstwebpage.html”. Add the following tags with relevant content.</p> <ol style="list-style-type: none"> <li>1. Set the title of the page as “My First Web Page”</li> <li>2. Within the body use the following tags: <ol style="list-style-type: none"> <li>a) Moving text = “Basic HTML Tags”</li> <li>b) Different heading tags (h1 to h6)</li> <li>c) Paragraph</li> <li>d) Horizontal line</li> <li>e) Line Break</li> <li>f) Block Quote</li> <li>g) Pre tag</li> <li>h) Different Logical Style (&lt;b&gt;, &lt;u&gt;, &lt;sub&gt;, &lt;sup&gt; etc.)</li> </ol> </li> </ol>		
2	<p>Develop the HTML page named as “Table.html” to display your class time table.</p> <ol style="list-style-type: none"> <li>a) Provide the title as Time Table with table header and table footer, row-span and col-span etc.</li> <li>b) Provide various colour options to the cells (Highlight the lab hours and elective hours with different colours.)</li> <li>c) Provide colour options for rows.</li> </ol>		
3	<p>Develop an external style sheet named as “style.css” and provide different styles for h2, h3, hr, p, div, span, time, img &amp; a tags. Apply different CSS selectors for tags and demonstrate the significance of each.</p>		
4	<p>Develop HTML page named as “registration.html” having variety of HTML input elements with background colors, table for alignment &amp; provide font colors &amp; size using CSS styles.</p>		
5	<p>Develop HTML page named as “newspaper.html” having variety of HTML semantic elements with background colors, text-colors &amp; size for figure, table, aside, section, article, header, footer... etc.</p>		
6	<p>Apply HTML, CSS and JavaScript to design a simple calculator to perform the following operations: sum, product, difference, remainder, quotient, power, square-root and square.</p>		
7	<p>Develop JavaScript program (with HTML/CSS) for:</p> <ol style="list-style-type: none"> <li>a) Converting JSON text to JavaScript Object</li> <li>b) Convert JSON results into a date</li> <li>c) Converting From JSON To CSV and CSV to JSON</li> <li>d) Create hash from string using crypto.createHash() method</li> </ol>		
8	<ol style="list-style-type: none"> <li>a. Develop a PHP program (with HTML/CSS) to keep track of the number of visitors visiting the web page and to display this count of visitors, with relevant headings.</li> <li>b. Develop a PHP program (with HTML/CSS) to sort the student records which are stored in</li> </ol>		

	the database using selection sort.
9	Develop jQuery script (with HTML/CSS) for: <ul style="list-style-type: none"> <li>a. Appends the content at the end of the existing paragraph and list.</li> <li>b. Change the state of the element with CSS style using animate() method</li> <li>c. Change the color of any div that is animated.</li> </ul>
10	Develop a JavaScript program with Ajax (with HTML/CSS) for: <ul style="list-style-type: none"> <li>a. Use ajax() method (without JQuery) to add the text content from the text file by sending ajax request.</li> <li>b. Use ajax() method (with JQuery) to add the text content from the text file by sending ajax request.</li> <li>c. Illustrate the use of getJSON() method in jQuery</li> <li>d. Illustrate the use of parseJSON() method to display JSON values.</li> </ul>

**Programming Assignment (5 marks):**

Construct a Website (multiple Web pages) containing 'Resume' and Bio -data by using relevant HTML elements and appropriate styling for presentation with CSS/jQuery/JavaScript. Host the Website on a cloud platform.

**Programming Assignment (5 marks):** Build a Web application with HTML, CSS, JavaScript, jQuery and PHP for online application/registration form. Form should accept the information and print/display on a browser with formatting/styling upon submission (Button click) on success. Host the application on a cloud platform.

**Assessment Details (both CIE and SEE)**

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

**Continuous Internal Evaluation (CIE):**

CIE marks for the practical course are **50 Marks**.

The split-up of CIE marks for record/ journal and test are in the ratio **60:40**.

Each experiment is to be evaluated for conduction with an observation sheet and record write-up.

Rubrics for the evaluation of the journal/write-up for hardware/software experiments are designed by the faculty who is handling the laboratory session and are made known to students at the beginning of the practical session.

Record should contain all the specified experiments in the syllabus and each experiment write-up will be evaluated for 10 marks.

Total marks scored by the students are scaled down to **30 marks** (60% of maximum marks).

Weightage to be given for neatness and submission of record/write-up on time.

Department shall conduct a test of 100 marks after the completion of all the experiments listed in the syllabus.

In a test, test write-up, conduction of experiment, acceptable result, and procedural knowledge will carry a weightage of 60% and the rest 40% for viva-voce.

The suitable rubrics can be designed to evaluate each student's performance and learning ability.

The marks scored shall be scaled down to **20 marks** (40% of the maximum marks).

The Sum of scaled-down marks scored in the report write-up/journal and marks of a test is the total CIE marks scored by the student.

### **Semester End Evaluation (SEE):**

SEE marks for the practical course are 50 Marks.

SEE shall be conducted jointly by the two examiners of the same institute, examiners are appointed by the Head of the Institute.

The examination schedule and names of examiners are informed to the university before the conduction of the examination. These practical examinations are to be conducted between the schedule mentioned in the academic calendar of the University.

All laboratory experiments are to be included for practical examination.

(Rubrics) Breakup of marks and the instructions printed on the cover page of the answer script to be strictly adhered to by the examiners. **OR** based on the course requirement evaluation rubrics shall be decided jointly by examiners.

Students can pick one question (experiment) from the questions lot prepared by the examiners jointly. Evaluation of test write-up/ conduction procedure and result/viva will be conducted jointly by examiners.

General rubrics suggested for SEE are mentioned here, writeup-20%, Conduction procedure and result in -60%, Viva-voce 20% of maximum marks. SEE for practical shall be evaluated for 100 marks and scored marks shall be scaled down to 50 marks (however, based on course type, rubrics shall be decided by the examiners)

Change of experiment is allowed only once and 15% of Marks allotted to the procedure part are to be made zero.

The minimum duration of SEE is 02 hours

### **Suggested Learning Resources:**

#### **Books:**

1. Randy Connolly and Ricardo Hoar, Fundamentals of Web Development, 3rd edition, Pearson, 2021
2. Robert W Sebesta, Programming the World Wide Web, 8th Edition, Pearson Education, 2020.

#### **Web Links:**

- <https://www.w3schools.com/html/default.asp>
- <https://www.w3schools.com/css/default.asp>
- [https://www.w3schools.com/js/js\\_examples.asp](https://www.w3schools.com/js/js_examples.asp)
- <https://www.geeksforgeeks.org/javascript-examples/>
- <https://www.w3schools.com/php/default.asp>
- <https://www.w3schools.com/jquery/default.asp>
- [https://www.w3schools.com/js/js\\_ajax\\_intro.asp](https://www.w3schools.com/js/js_ajax_intro.asp)
- <https://www.geeksforgeeks.org/jquery-tutorial/>



Index					
Sl. No.	Program List	CO	PO	RBT	Page No.
1	<p>Develop the HTML page named as “Myfirstwebpage.html”. Add the following tags with relevant content.</p> <ol style="list-style-type: none"> <li>Set the title of the page as “My First Web Page”</li> <li>Within the body use the following tags:               <ol style="list-style-type: none"> <li>Moving text = “Basic HTML Tags”</li> <li>Different heading tags (h1 to h6)</li> <li>Paragraph</li> <li>Horizontal line</li> <li>Line Break</li> <li>Block Quote</li> <li>Pre tag</li> <li>Different Logical Style (&lt;b&gt;, &lt;u&gt;, &lt;sub&gt;, &lt;sup&gt; etc.)</li> </ol> </li> </ol>	CO1, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L3	26
2	<p>Develop the HTML page named as “Table.html” to display your class time table.</p> <ol style="list-style-type: none"> <li>Provide the title as Time Table with table header and table footer, row-span and col-span etc.</li> <li>Provide various colour options to the cells (Highlight the lab hours and elective hours with different colours.)</li> <li>Provide colour options for rows.</li> </ol>	CO1, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L3	28
3	Develop an external style sheet named as “style.css” and provide different styles for h2, h3, hr, p, div, span, time, img & a tags. Apply different CSS selectors for tags and demonstrate the significance of each.	CO1, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L3	31
4	Develop HTML page named as “registration.html” having variety of HTML input elements with background colors, table for alignment & provide font colors & size using CSS styles.	CO1, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L3	35
5	Develop HTML page named as “newspaper.html” having variety of HTML semantic elements with background colors, text-colors & size for figure, table, aside, section, article, header, footer... etc.	CO1, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L3	41
6	Apply HTML, CSS and JavaScript to design a simple calculator to perform the following operations: sum, product, difference, remainder, quotient, power, square-root and square.	CO1, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L3	49

7	Develop JavaScript program (with HTML/CSS) for: a) Converting JSON text to JavaScript Object b) Convert JSON results into a date c) Converting From JSON To CSV and CSV to JSON d) Create hash from string using crypto.createHash() method	CO1, CO2, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L3	52
8	Develop a PHP program (with HTML/CSS) to keep track of the number of visitors visiting the web page and to display this count of visitors, with relevant headings. b. Develop a PHP program (with HTML/CSS) to sort the student records which are stored in the database using selection sort.	CO2, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L3	58
9	Develop jQuery script (with HTML/CSS) for: a. Appends the content at the end of the existing paragraph and list. b. Change the state of the element with CSS style using animate() method c. Change the color of any div that is animated.	CO2, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L3	64
10	Develop a JavaScript program with Ajax (with HTML/CSS) for: a. Use ajax() method (without JQuery) to add the text content from the text file by sending ajax request. b. Use ajax() method (with JQuery) to add the text content from the text file by sending ajax request. c. Illustrate the use of getJSON() method in jQuery d. Illustrate the use of parseJSON() method to display JSON values.	CO2, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L3	68
<b>Programming Assignment (5 marks):</b> Construct a website (multiple Web pages) containing 'Resume' and Bio -data by using relevant HTML elements and appropriate styling for presentation with CSS/jQuery/JavaScript. Host the Website on a cloud platform.		CO1, CO2, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L4	
<b>Programming Assignment (5 marks):</b> Build a Web application with HTML, CSS, JavaScript, jQuery and PHP for online application/registration form. Form should accept the information and print/display on a browser with formatting/styling upon submission (Button click) on success. Host the application on a cloud platform.		CO1, CO3	PO1, PO2, PO3, PO5, PSO1,2,3	L4	
Viva Questions and Answers					73

**Course Articulation Matrix**

Cos	Pos												PSOs		
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3
CO1	3	3	3	1	3	2	-	-	2	2	-	1	3	1	2
CO2	3	3	3	1	3	2	-	-	2	2	-	1	3	1	2
CO3	3	3	3	1	3	2	-	-	2	2	-	1	3	1	2

**3 - High Correlation****2 - Medium Correlation****1 – Low Correlation****INTRODUCTION TO WEB TECHNOLOGY**

Web technology refers to the tools, frameworks, and techniques used to create, manage, and operate websites and web applications. It encompasses various components that enable the development, deployment, and functioning of the World Wide Web. Key aspects of web technology include:

1. Front-end technologies: These are tools used to create the client-side (what users see and interact with in a browser), including:
  - HTML (Hypertext Markup Language) for structuring content.
  - CSS (Cascading Style Sheets) for styling and layout.
  - JavaScript for interactive and dynamic elements.
2. Back-end technologies: These manage server-side operations (behind-the-scenes functionality), including:
  - Server-side scripting languages like PHP, Python, Ruby, and Node.js.
  - Databases like MySQL, MongoDB, and PostgreSQL to store and manage data.
  - Web servers like Apache and Nginx.
3. Web protocols and standards: These ensure smooth communication between servers and clients, such as:
  - HTTP/HTTPS (Hypertext Transfer Protocol) for transferring web pages.
  - WebSockets for real-time, bidirectional communication.
  - RESTful APIs for interaction between front-end and back-end services.
4. Web development frameworks: These simplify the creation of web applications:
  - Front-end frameworks like React, Angular, and Vue.js.
  - Back-end frameworks like Django, Flask, Ruby on Rails, and Express.js.
5. Web hosting: This involves storing website files on servers and making them accessible on the internet.

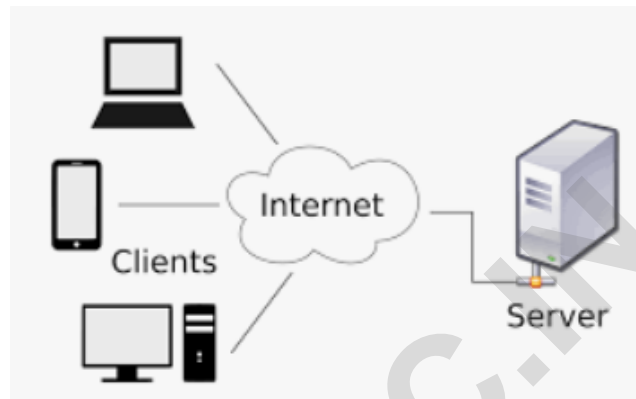
**Web Applications:**

A web application is a software application that runs on a web server and can be accessed through a web browser using the internet. Unlike traditional desktop applications, which are installed on a local computer, web applications are accessed remotely and require only a browser to operate, making

them platform-independent. Ex: Facebook, Netflix, Gmail, Dropbox etc.

### **Client –Server Model:**

The client–server model is a distributed application structure in computing that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network. A server is a host that is running one or more server programs which share their resources with clients. A client requests a server's content or service function.



### **WWW:**

WWW stands for the World Wide Web, which is a system of interlinked hypertext documents and multimedia content that can be accessed via the internet. When a user accesses the web using a browser, they can view web pages that may contain text, images, videos, and other media, as well as links to other pages.

Key components of the World Wide Web include:

1. **Web Pages:** These are documents written in HTML (Hypertext Markup Language) and are displayed by web browsers like Chrome, Firefox, or Safari.
2. **Web Browsers:** Software applications that allow users to access and navigate the web (e.g., Chrome, Firefox, Edge).
3. **URLs (Uniform Resource Locators):** The web address used to locate a web page or resource on the internet (e.g., <https://www.example.com>).
4. **HTTP/HTTPS:** Protocols used to request and transfer data over the web. HTTPS adds a layer of security (encryption) to the connection.
5. **Hyperlinks:** Links between different web pages or websites, allowing users to navigate the vast content on the web.

**Web site:** A set of interconnected web pages, usually including a homepage, generally located on the same server, and prepared and maintained as a collection of information by a person, group, or organization.

### **Types of Web Pages:**

- **Static web page:** is delivered exactly as stored, as web content in the web server's file system. Contents cannot be changed.
- **Dynamic web page:** is generated by a web application that is driven by server-side software

or client-side scripting. Dynamic web pages help the browser (the client) to enhance the web page through user input to the server. Contents can be changed as evolution over time.

### **Browsers & their types:**

A web browser (commonly referred to as a browser) is a software application for retrieving, presenting and traversing information resources on the World Wide Web. The major web browsers are Google Chrome, Firefox, Internet Explorer, Opera, and Safari.

## **INTRODUCTION TO HTML**

HTML (HyperText Markup Language) is the standard language used to create and design web pages. It provides the structure for a web page by defining elements like headings, paragraphs, images, links, and more. HTML is not a programming language but a markup language that uses tags to annotate text, images, and other content, making it interpretable by web browsers.

HTML was developed by Tim Berners-Lee in 1991 and is currently maintained by the World Wide Web Consortium (W3C). Modern HTML (HTML5) also supports multimedia, interactive features, and responsive design.

### **Structure of an HTML Document**

A basic HTML document has the following structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Page Title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph of text.</p>
    
    <a href="https://example.com">This is a link</a>
  </body>
</html>
```

### **Key Components of an HTML Document**

#### **1. DOCTYPE Declaration:**

<!DOCTYPE html> tells the browser that this document is written in HTML5, the latest

version of HTML.

## 2. HTML Element:

`<html>` encloses the entire HTML document. It is the root element of the page.

## 3. Head Section:

- The `<head>` element contains meta-information about the document, like the page title, character encoding (`<meta charset="UTF-8">`), and other resources like CSS styles or JavaScript files.
- The `<title>` element specifies the title of the page, shown on the browser tab.

## 4. Body Section:

The `<body>` element contains the content that will be visible on the web page, including text, images, links, forms, etc.

---

## HTML Elements

HTML documents are made up of elements. An HTML element consists of:

- Opening tag: `<tagname>`
- Content: Between the opening and closing tag (optional for some tags)
- Closing tag: `</tagname>`

For example:

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

## Common HTML Elements

Element	Description	Example
<code>&lt;html&gt;</code>	Root element that wraps the entire HTML document.	<code>&lt;html&gt; ... &lt;/html&gt;</code>
<code>&lt;head&gt;</code>	Contains meta-information about the document (title, links to resources, etc.).	<code>&lt;head&gt; ... &lt;/head&gt;</code>
<code>&lt;title&gt;</code>	Specifies the title of the document shown in the browser tab.	<code>&lt;title&gt;My Webpage&lt;/title&gt;</code>
<code>&lt;body&gt;</code>	Contains the content of the webpage visible to the user.	<code>&lt;body&gt; ... &lt;/body&gt;</code>
<code>&lt;h1&gt;</code> to <code>&lt;h6&gt;</code>	Defines headings, where <code>&lt;h1&gt;</code> is the largest, <code>&lt;h6&gt;</code> is the smallest.	<code>&lt;h1&gt;Main Heading&lt;/h1&gt;</code>
<code>&lt;p&gt;</code>	Defines a paragraph.	<code>&lt;p&gt;This is a paragraph of text.&lt;/p&gt;</code>

Element	Description	Example
 	Inserts a line break.	This is a line. This is another line.
<hr>	Inserts a horizontal line (thematic break).	<hr>
<div>	Block-level container for grouping elements (used for layout).	<div>Content goes here</div>
<span>	Inline container for grouping elements (used for styling parts of text).	<span style="color: red;">Red text</span>
<a>	Defines a hyperlink.	<a href="https://www.example.com">Visit Example</a>
<img>	Embeds an image in the webpage.	
<audio>	Embeds audio content.	<audio controls> <source src="audio.mp3" type="audio/mpeg"></audio>
<video>	Embeds video content.	<video controls><source src="video.mp4" type="video/mp4"></video>
<ul>	Defines an unordered (bulleted) list.	<ul> <li>Item 1</li> <li>Item 2</li> </ul>
<ol>	Defines an ordered (numbered) list.	<ol> <li>First item</li> <li>Second item</li> </ol>
<li>	Represents a list item in ordered or unordered lists.	<li>List Item</li>
<table>	Creates a table to organize data in rows and columns.	<table> ... </table>
<tr>	Defines a row in a table.	<tr> ... </tr>
<th>	Defines a header cell in a table.	<th>Header</th>



Element	Description	Example
<td>	Defines a data cell in a table.	<td>Data</td>
<form>	Defines a form for user input.	<form action="/submit" method="post"> ... </form>
<input>	Specifies an input field where the user can enter data.	<input type="text" name="username">
<button>	Creates a clickable button.	<button type="submit">Submit</button>
<label>	Defines a label for an input element.	<label for="username">Username:</label>
<meta>	Provides metadata (like description, keywords, etc.) about the webpage.	<meta charset="UTF-8">
<link>	Links external resources like stylesheets to the document.	<link rel="stylesheet" href="style.css">

### Uses of HTML

- **Webpage Structure:** HTML provides the foundational structure for all web pages and applications. It defines the layout of content and ensures web browsers display the text, images, and interactive elements correctly.
- **Hyperlinks:** HTML enables links between pages on the web, creating the interconnected nature of the World Wide Web.
- **Multimedia Integration:** Modern HTML (HTML5) supports embedding multimedia elements such as audio, video, and interactive games directly into web pages.
- **Form Handling:** HTML forms enable user interaction, like logging in, submitting data, or making online purchases.
- **Responsive Design:** HTML5 includes new elements like <header>, <footer>, <article>, and <section>, which enhance the structure and accessibility of web pages, making them easier to style responsively.

### CSS: Cascading Style Sheets

CSS (Cascading Style Sheets) is a language used to define the look and formatting of web pages written in HTML. It allows developers to control the layout, colors, fonts, and overall visual appearance of websites. CSS separates content (HTML) from presentation, making it easier to manage and maintain.

### Advantages of CSS

- **Separation of Content and Design:** Keeps HTML focused on content while CSS manages all styling, making code cleaner and easier to maintain.



- **Consistency:** External stylesheets ensure a uniform look across multiple pages, and updates to the CSS file reflect on all linked pages.
- **Faster Loading:** CSS reduces repetitive code and allows browsers to cache stylesheets, speeding up page loads.
- **Responsive Design:** CSS allows web pages to adapt to different screen sizes and devices, improving user experience.
- **Easier Maintenance:** Making changes to a single CSS file updates the design across the entire site without altering HTML.

### Using CSS

CSS can be added to HTML documents in 3 ways:

- a) **Inline** - by using the style attribute inside HTML elements
- b) **Internal** - by using a <style> element in the <head> section
- c) **External** - by using a <link> element to link to an external CSS file

#### a)Inline CSS:

- An inline CSS is used to apply a unique style to a single HTML element.
- An inline CSS uses the **style** attribute of an HTML element.

The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

Example: <h1 style="color:blue;">A Blue coloured Heading</h1>

<p style=" background-color:lightblue;font-family:calibri;font-size:20px;">A red paragraph.</p>

#### b)Internal CSS:

- An internal CSS is used to define a style for a single HTML page.
- An internal CSS is defined in the <head> section of an HTML page, within a <style> element.

Example:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {background-color: powderblue;}
```

```
h1 {color: blue;}
```

```
p {color: red;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

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

```
</body>
```

```
</html>
```

#### External CSS

An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the <head> section of each HTML page:

Ex:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension. Here is what the "styles.css" file looks like:

"styles.css":

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

### Structure of a CSS Rule:

```
selector {
  property: value;
  property: value;
}
```

- Selector: Identifies the HTML element or elements you want to style (e.g., h1, p, .class, #id).
- Property: A style attribute (e.g., color, font-size, margin) that affects how the element is displayed.
- Value: The value of the property (e.g., red, 16px, 20px) that defines the specific appearance.

### CSS Simple Selectors

The simple selectors select elements based on element-name, id, and class. In addition, there is the **universal selector** (\*).

Selector	Example	Example description
<i>element</i>	P	Selects all <p> elements
<i>#id</i>	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
<i>.class</i>	.intro p.intro	Selects all elements with class="intro" Selects all <p> elements with class="intro"

### JavaScript

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

### Features of JavaScript

There are following features of JavaScript:

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.
6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
8. It provides good control to the users over the web browsers.

### Applications of JavaScript

1. Client-Side Scripting: JavaScript runs in the user's web browser, allowing for immediate interaction with web pages without needing to communicate with the server.

2. Event Handling: JavaScript can respond to user actions such as clicks, keypresses, and mouse movements, enabling dynamic changes to the webpage.
3. Manipulating HTML and CSS: JavaScript can change the content and style of HTML elements on the fly, creating interactive and responsive web pages.
4. Asynchronous Operations: JavaScript supports asynchronous operations using techniques like AJAX (Asynchronous JavaScript and XML) and Fetch API, allowing web pages to load data and update content without reloading.
5. Cross-Platform: JavaScript is supported by all major web browsers, making it a universal tool for web development.
6. Versatility: JavaScript can be used both in the browser (client-side) and on the server (using environments like Node.js), making it a full-stack development language.

### The <script> Tag

In HTML, JavaScript code is inserted between <script> and </script> tags.

#### Syntax:

<script>

document.getElementById("demo").innerHTML = "My First JavaScript";

</script>

#### Example:

```
<html>
<body>
<h2>JavaScript in Body</h2>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
</body>
</html>
```

**Note:** Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

### JavaScript in <head>

In this example, a JavaScript function is placed in the <head> section of an HTML page. The function is invoked (called) when a button is clicked:

#### Example

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

```
</script>
</head>
<body>
<h2>Demo JavaScript in Head</h2>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

### JavaScript in <body>

In this example, a JavaScript function is placed in the <body> section of an HTML page. The function is invoked (called) when a button is clicked:

#### Example

```
<!DOCTYPE html>
<html>
<body>
<h2>Demo JavaScript in Body</h2>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</body>
</html>
```

### External JavaScript

Scripts can also be placed in external files:

#### External file: myScript.js

```
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

#### html file:

```
<!DOCTYPE html>
<html>
<body>
<h2>Demo External JavaScript</h2>
<p id="demo">A Paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
<p>This example links to "myScript.js".</p>
<p>(myFunction is stored in "myScript.js")</p>
<script src="myScript.js"></script>
</body>
</html>
```

## JavaScript Output

JavaScript can "display" data in different ways:

- **Writing into an HTML element, using innerHTML:** To access an HTML element, JavaScript can use the `document.getElementById(id)` method. The `id` attribute defines the HTML element. The `innerHTML` property defines the HTML content:  
`document.getElementById("demo").innerHTML = "hello";`
- Writing into the HTML output using `document.write()`.  
`<script>document.write(5 + 6); </script>`
- Writing into an alert box, using `window.alert()`.  
`<script> window.alert("well done"); </script>`
- Writing into the browser console, using `console.log()`  
`<script>console.log(5 + 6); </script>`

## PHP

PHP (Hypertext Preprocessor) is a popular server-side scripting language designed for web development. It is embedded within HTML and executed on the server to generate dynamic web content based on user interactions or data. PHP excels in creating interactive and data-driven websites, offering robust support for database integration and session management. Being open-source and platform-independent, PHP is widely used by developers for its ease of use, extensive community support, and ability to handle complex server-side tasks efficiently.

**PHP syntax is designed to be simple and flexible, allowing for easy integration with HTML and providing powerful scripting capabilities. Below is a quick guide to understanding PHP syntax.**

### 1. Basic PHP Syntax

- **PHP Code Block:** PHP code is written between `<?php ... ?>` tags.
- **Echo Statement:** `echo` is used to output text or variables to the browser.  
`<?php  
    echo "Hello, World!";  
?>`
- **Comments:**
  - Single-line comments: `//` or `#`
  - Multi-line comments: `/* ... */`

### 2. Variables

- Variables in PHP start with a `$` symbol, followed by the variable name.
- PHP variables are dynamically typed, meaning they don't require explicit type declarations.  
`<?php  
    $name = "Alice";`

```
$age = 25;
echo "Name: $name, Age: $age";
?>
```

### 3. Data Types

PHP supports several data types:

- String: Textual data, e.g., "Hello"
- Integer: Whole numbers, e.g., 42
- Float: Decimal numbers, e.g., 3.14
- Boolean: true or false
- Array: Collection of values, e.g., [1, 2, 3]
- Object: Instance of a class
- NULL: Variable with no value

### 4. Operators

- **Arithmetic Operators:** +, -, \*, /, %

```
<?php
$a = 10;
$b = 5;
$sum = $a + $b;
echo $sum; // Outputs: 15
?>
```

- Comparison Operators: ==, !=, <, >, <=, >=
- Logical Operators: &&, ||, !

### 5. Control Structures

- **If-Else Statement:**

```
<?php
$score = 85;
if ($score >= 90) {
    echo "Grade: A";
} elseif ($score >= 75) {
    echo "Grade: B";
} else {
    echo "Grade: C";
}
?>
```

- **Switch Statement:**

```
<?php
$day = "Monday";
switch ($day) {
    case "Monday":
        echo "It's Monday!";
        break;
    case "Friday":

```

```
        echo "It's Friday!";
        break;
    default:
        echo "Another day!";
    }
?>
```

- **Loops:**

- **For Loop:**

```
<?php
    for ($i = 1; $i <= 5; $i++) {
        echo $i . " ";
    }
?>
```

- **While Loop:**

```
<?php
    $i = 1;
    while ($i <= 5) {
        echo $i . " ";
        $i++;
    }
?>
```

- **Foreach Loop: Used for arrays.**

```
<?php
    $colors = array("Red", "Green", "Blue");
    foreach ($colors as $color) {
        echo $color . " ";
    }
?>
```

## 6. Functions

Functions are reusable blocks of code that perform specific tasks. Defined with the function keyword.

```
<?php
    function greet($name) {
        return "Hello, $name!";
    }
    echo greet("Alice"); // Outputs: Hello, Alice!
?>
```

## 7. Including Files

PHP includes external files using include or require:

```
<?php
    include "header.php";
    require "footer.php";
```



?>

### 8. Connecting to a Database

PHP can connect to databases like MySQL using functions like mysqli\_connect() or PDO.

```
<?php
    $conn = new mysqli("localhost", "username", "password", "database");
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Connected successfully!";
?>
```

## EXPERIMENT - 01

Develop the HTML page named as “Myfirstwebpage.html”. Add the following tags with relevant content.

1. Set the title of the page as “My First Web Page”
2. Within the body use the following tags:
  - a) Moving text = “Basic HTML Tags”
  - b) Different heading tags (h1 to h6)
  - c) Paragraph
  - d) Horizontal line
  - e) Line Break
  - f) Block Quote
  - g) Pre tag
  - h) Different Logical Style (<b>, <u>, <sub>, <sup> etc.)

### Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My First Web Page</title>
</head>
<body>
  <!-- Moving Text -->
  <marquee>Basic HTML Tags</marquee>
  <!-- Different Heading Tags -->
  <h1>Heading 1</h1>
  <h2>Heading 2</h2>
  <h3>Heading 3</h3>
  <h4>Heading 4</h4>
  <h5>Heading 5</h5>
  <h6>Heading 6</h6>
  <!-- Paragraph -->
  <p>This is a paragraph demonstrating basic HTML tags and their usage.</p>
  <!-- Horizontal Line -->
  <hr>
  <!-- Line Break -->
  This is some text. <br> This is the next line after a line break.
  <!-- Block Quote -->
  <blockquote>
    "This is a block quote, usually used for quoting someone or something significant."
  </blockquote>
```

<!-- Preformatted Text -->

<pre>

This is preformatted text.

It preserves  
the whitespace  
and line breaks.

</pre>

<!-- Different Logical Styles -->

<p>

This is <b>bold</b> text. <br>

This is <i>italic</i> text. <br>

This is <u>underlined</u> text. <br>

This is <del>strikethrough</del> text. <br>

This is <sup>superscript</sup> text. <br>

This is <sub>subscript</sub> text. <br>

This is <strong>strong</strong> text. <br>

This is <em>emphasized</em> text.

</p>

</body>

</html>

### Sample Output:

## Heading 1

### Heading 2

#### Heading 3

#### Heading 4

#### Heading 5

#### Heading 6

This is a paragraph demonstrating basic HTML tags and their usage.

This is some text.

This is the next line after a line break.

"This is a block quote, usually used for quoting someone or something significant."

This is preformatted text.  
It preserves  
the whitespace  
and line breaks.

This is **bold** text.

This is *italic* text.

This is underlined text.

This is ~~strikethrough~~ text.

This is <sup>superscript</sup> text.

This is <sub>subscript</sub> text.

This is **strong** text.

This is *emphasized* text.

## EXPERIMENT 2

Develop the HTML page named as “Table.html” to display your class time table.

- Provide the title as Time Table with table header and table footer, row-span and col-span etc.
- Provide various colour options to the cells (Highlight the lab hours and elective hours with different colours.)
- Provide colour options for rows.

### Program

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Time Table</title>
  <style>
    table {
      width: 60%;
      border-collapse: collapse;
      text-align: center;
    }
    th, td {
      border: 2px solid black;
      padding: 10px;
    }
    th {
      background-color: rgb(135, 2, 71);
      color: white;
    }
    thead th { background-color: rgb(57, 132, 61); }

    tfoot td {
      background-color: rgb(150, 234, 136);
      font-weight: bold;
    }
    .lab {
      background-color: lightcoral; /* Highlight lab hours */
    }
    .elective {
      background-color: rgba(119, 53, 233, 0.4); /* Highlight elective hours */
    }
    tr:nth-child(even) {
```

```
        background-color: rgb(234, 186, 145);
    }
    tr:nth-child(odd) {
        background-color: rgb(165, 212, 241);
    }
</style>
</head>
<body>
    <center>
```

```
<h1>Class Time Table</h1>
```

```
<table>
    <thead>
        <tr>
            <th colspan="9">Time Table</th>
        </tr>
    </thead>
    <tr>
        <th>Day</th>
        <th>8:45 - 9:45</th>
        <th>9:45 - 10:45</th>
        <th>10:45 - 11:00</th>
        <th>11:00 - 12:00</th>
        <th>12:00 - 1:00</th>
        <th>1:00 - 1:45</th>
        <th>1:45 - 2:45</th>
        <th>2:45 - 3:45</th>
    </tr>

    <tbody>
        <tr>
            <td>Monday</td>
            <td>TOC</td>
            <td class="elective">Elective: AI</td>

            <td rowspan="5">BREAK</td>
            <td>CN</td>
            <td>EVS</td>
            <td rowspan="5">BREAK</td>
            <td>SE</td>
            <td>RM</td>
        </tr>
        <tr>
```

```

        <td>Tuesday</td>
        <td>SE</td>
        <td>TOC</td>
        <td>RM</td>
        <td>CN</td>

        <td class="lab" colspan="2">CN/WEB LAB (2 Hours)</td>
    </tr>
    <tr>
        <td>Wednesday</td>
        <td>TOC</td>
        <td>SE</td>
        <td>CN</td>
        <td class="elective">Elective: AI</td>
        <td colspan="2" class="lab">CN/WEB LAB (2 Hours)</td>
    </tr>
    <tr>
        <td>Thursday</td>
        <td>CN</td>
        <td>SE</td>
        <td>TOC</td>
        <td>EVS</td>
        <td class="elective">Elective: AI</td>
        <td>SE</td>
    </tr>
    <tr>
        <td>Friday</td>
        <td>RM</td>
        <td>TOC</td>
        <td>CN</td>
        <td>EVS</td>
        <td colspan="2" class="lab">CN/WEB LAB (2 Hours)</td>
    </tr>
</tbody>
<tfoot>
    <tr>
        <td colspan="9">End of Timetable</td>
    </tr>
</tfoot>
</table>
</center>
</body>
</html>

```

## Sample Output:

Class Time Table

Time Table								
Day	8:45 - 9:45	9:45 - 10:45	10:45 - 11:00	11:00 - 12:00	12:00 - 1:00	1:00 - 1:45	1:45 - 2:45	2:45 - 3:45
Monday	TOC	Elective: AI	BREAK	CN	EVS	BREAK	SE	RM
Tuesday	SE	TOC		RM	CN		CN/WEB LAB (2 Hours)	
Wednesday	TOC	SE		CN	Elective: AI		CN/WEB LAB (2 Hours)	
Thursday	CN	SE		TOC	EVS		Elective: AI	SE
Friday	RM	TOC		CN	EVS		CN/WEB LAB (2 Hours)	
End of Timetable								

## EXPERIMENT 3

Develop an external style sheet named as “style.css” and provide different styles for h2, h3, hr, p, div, span, time, img & a tags. Apply different CSS selectors for tags and demonstrate the significance of each.

## style.css:

```

/* Element Selector: Applies to all h2 elements */
h2 {
    font-family: 'Courier New', Courier, monospace;
    color: darkblue;
    text-align: center;
    border-bottom: 3px solid #333;
}

/* Element Selector: Applies to all h3 elements */
h3 {
    font-size: 1.5em;
    color: darkgreen;
    margin-bottom: 10px;
}

/* Element Selector: Applies to all hr elements */
hr {
    border: 1px solid #ccc;

```

```
margin: 20px 0;
}

/* Class Selector: Applies only to paragraphs with the 'intro' class */
p.intro {
    font-size: 1.2em;
    line-height: 1.6;
    background-color: #f0f0f0;
    padding: 15px;
}

/* ID Selector: Targets only the div with ID 'content' */
#content {
    background-color: #e8e8e8;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

/* Class Selector: Highlights text inside span elements */
span.highlight {
    color: red;
    font-weight: bold;
}

/* Element Selector: Applies styles to time elements */
time {
    color: #555;
    font-style: italic;
    font-size: 0.9em;
}

/* Attribute Selector: Applies styles to img elements with an alt attribute */
img[alt] {
    border: 3px solid #ddd;
    border-radius: 8px;
    max-width: 100%;
}
```



/\* Pseudo-class Selector: Styles the a tag when hovered \*/

```
a:hover {  
    color: orange;  
    text-decoration: underline;  
}
```

/\* Universal Selector: Applies a border-box layout model to all elements \*/

```
* {  
    box-sizing: border-box;  
}
```

### HTML File

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>CSS Selectors Demonstration</title>  
    <!-- Linking the external stylesheet -->  
    <link rel="stylesheet" href="style.css">  
</head>  
<body>  
  
    <!-- Applying the h2 style -->  
    <h2>Welcome to My Website</h2>  
  
    <!-- Applying the h3 style -->  
    <h3>Latest Updates</h3>  
  
    <!-- Applying the hr style -->  
    <hr>  
  
    <!-- Applying the p.intro style using class selector -->  
    <p class="intro">This is an introduction to the website. The paragraph is styled using a class  
selector to emphasize its importance.</p>  
  
    <!-- Applying the div style using ID selector -->  
    <div id="content">  
        <p>This is a div styled using an ID selector. It contains a highlighted <span  
class="highlight">word</span> and some important information.</p>
```

```
<!-- Applying the time element style -->
<p>Published on: <time datetime="2024-09-24">September 24, 2024</time></p>
</div>

<!-- Applying the img style using attribute selector -->


<!-- Applying the a:hover pseudo-class -->
<p>Check out our <a href="#">latest article</a> to learn more!</p>

</body>
</html>
```

### Sample Output:

## Welcome to My Website

### Latest Updates

This is an introduction to the website. The paragraph is styled using a class selector to emphasize its importance.

This is a div styled using an ID selector. It contains a highlighted **word** and some important information.

Published on: *September 24, 2024*



Check out our [latest article](#) to learn more!

## EXPERIMENT 4

**Develop HTML page named as “registration.html” having variety of HTML input elements with background colors, table for alignment & provide font colors & size using CSS styles.**

**CSS file:**

```
/* General styles */
```

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f4f4f4;  
    padding: 20px;  
}
```

```
h1 {  
    color: darkblue;  
    text-align: center;  
}
```

```
form {  
    background-color: #fff;  
    border-radius: 10px;  
    padding: 20px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
    max-width: 700px;  
    margin: auto;  
}
```

```
table {  
    width: 100%;  
    border-collapse: collapse;  
    margin: 20px 0;  
}
```

```
table td {  
    padding: 10px;  
}
```

```
/* Input field styling */
```

```
input[type="text"], input[type="email"], input[type="number"],
```

```
input[type="password"], input[type="file"], input[type="date"],
input[type="url"], select, textarea {
    width: 100%;
    padding: 10px;
    margin-top: 5px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

/* Background colors and font styles */
.bg-light {
    background-color: #f9f9f9;
}

.highlight {
    background-color: #d3f0d1;
}

/* Styling for buttons */
input[type="submit"], input[type="reset"] {
    background-color: darkblue;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

input[type="submit"]:hover, input[type="reset"]:hover {
    background-color: blue;
}

/* Font sizes and colors */
td label {
    font-size: 1.1em;
    color: darkgreen;
}

td select, td input, td textarea {
```

```
font-size: 1em;
color: #333;
}
```

```
td {
    vertical-align: top;
}
```

```
/* Image and file input styling */
input[type="file"] {
    border: none;
}
```

```
/* Error message styles (for potential client-side validation using JS) */
.error-message {
    color: red;
    font-size: 0.9em;
}
```

#### HTML file:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>University Student Registration</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <h1>University Student Registration Form</h1>

    <form action="#" method="POST" enctype="multipart/form-data">
        <table class="bg-light">
            <tr>
                <td><label for="firstname">First Name:</label></td>
                <td><input type="text" id="firstname" name="firstname" required
placeholder="Enter your first name"></td>
            </tr>
```

```

<tr>
  <td><label for="lastname">Last Name:</label></td>
  <td><input type="text" id="lastname" name="lastname" required
placeholder="Enter your last name"></td>
</tr>
<tr>
  <td><label for="email">Email Address:</label></td>
  <td><input type="email" id="email" name="email" required placeholder="Enter a
valid email address"></td>
</tr>
<tr>
  <td><label for="phone">Phone Number:</label></td>
  <td><input type="number" id="phone" name="phone" required placeholder="Enter
your phone number" min="1000000000" max="9999999999"></td>
</tr>
<tr>
  <td><label for="password">Password:</label></td>
  <td><input type="password" id="password" name="password" required
placeholder="Create a password" minlength="8"></td>
</tr>
<tr>
  <td><label for="confirmpassword">Confirm Password:</label></td>
  <td><input type="password" id="confirmpassword" name="confirmpassword"
required placeholder="Confirm your password" minlength="8"></td>
</tr>
<tr class="highlight">
  <td><label for="program">Select Program:</label></td>
  <td>
    <select id="program" name="program" required>
      <option value="">--Select Program--</option>
      <option value="B.Sc Computer Science">B.Sc in Computer Science</option>
      <option value="BBA">BBA</option>
      <option value="B.Com">B.Com</option>
      <option value="B.A.">B.A.</option>
    </select>
  </td>
</tr>
<tr>
  <td><label for="gender">Gender:</label></td>

```

```

        <td>
            <input type="radio" id="male" name="gender" value="Male" required>
            <label for="male">Male</label>
            <input type="radio" id="female" name="gender" value="Female" required>
            <label for="female">Female</label>
        </td>
    </tr>
    <tr>
        <td><label for="address">Address:</label></td>
        <td><textarea id="address" name="address" rows="4" required placeholder="Enter
your address"></textarea></td>
    </tr>
    <tr class="highlight">
        <td><label for="dob">Date of Birth:</label></td>
        <td><input type="date" id="dob" name="dob" required></td>
    </tr>
    <tr>
        <td><label for="nationality">Nationality:</label></td>
        <td><input type="text" id="nationality" name="nationality" required
placeholder="Enter your nationality"></td>
    </tr>
    <tr>
        <td><label for="photo">Upload Photo:</label></td>
        <td><input type="file" id="photo" name="photo" accept="image/*" required></td>
    </tr>
    <tr>
        <td><label for="website">Personal Website (Optional):</label></td>
        <td><input type="url" id="website" name="website" placeholder="Enter your
personal website URL (optional)"></td>
    </tr>
    <tr>
        <td><label for="newsletter">Subscribe to Newsletter:</label></td>
        <td><input type="checkbox" id="newsletter" name="newsletter" value="yes"> Yes,
I would like to receive the newsletter.</td>
    </tr>
    <tr>
        <td><label for="terms">Agree to Terms and Conditions:</label></td>
        <td><input type="checkbox" id="terms" name="terms" required> I agree to the
terms and conditions.</td>
    </tr>

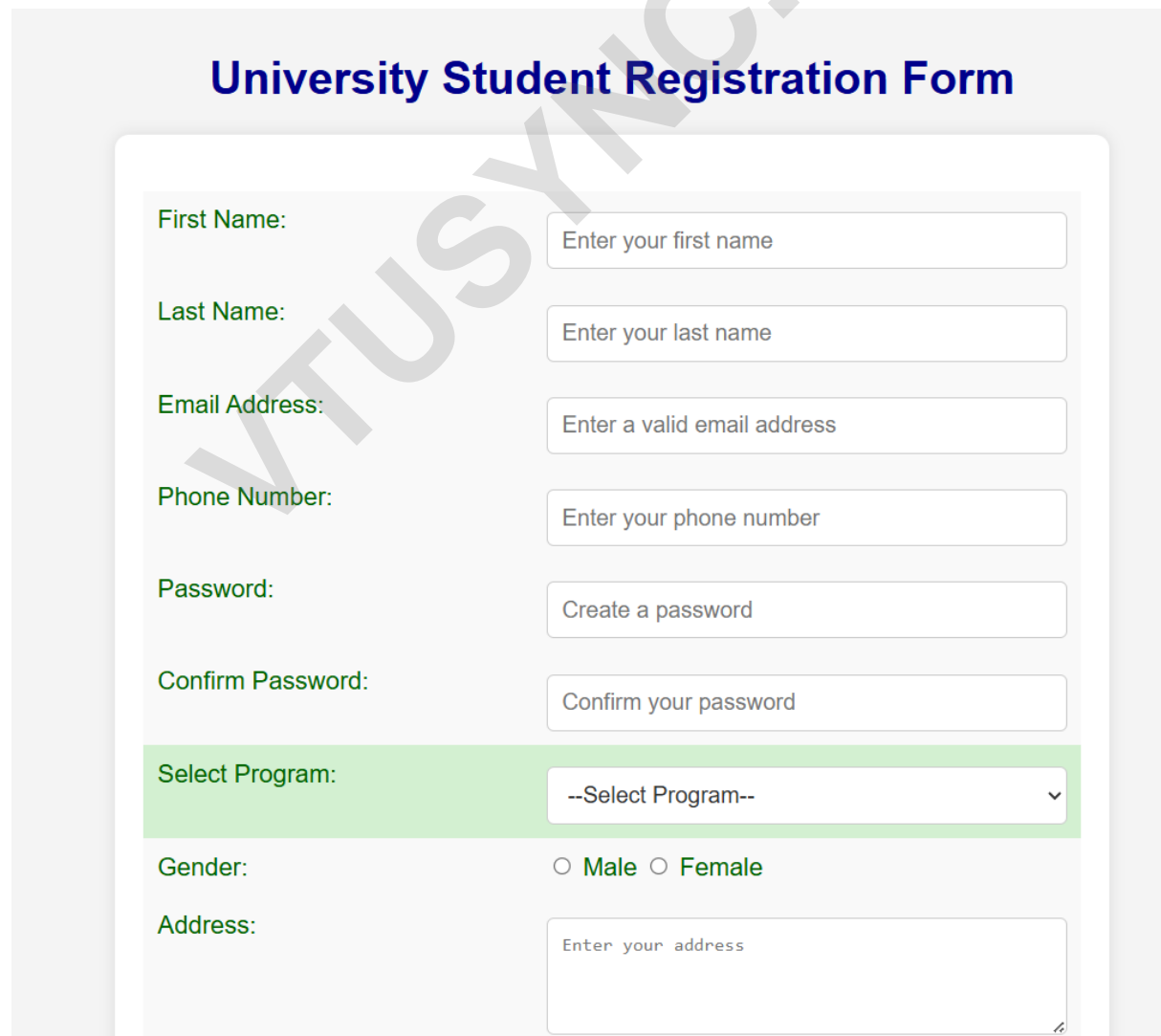
```

```
</tr>
</table>

<table>
  <tr>
    <td><input type="submit" value="Submit"></td>
    <td><input type="reset" value="Reset"></td>
  </tr>
</table>
</form>

</body>
</html>
```

**Sample Output:**



The sample output shows a web form titled "University Student Registration Form". The form contains the following fields and controls:

- First Name:** A text input field with the placeholder "Enter your first name".
- Last Name:** A text input field with the placeholder "Enter your last name".
- Email Address:** A text input field with the placeholder "Enter a valid email address".
- Phone Number:** A text input field with the placeholder "Enter your phone number".
- Password:** A text input field with the placeholder "Create a password".
- Confirm Password:** A text input field with the placeholder "Confirm your password".
- Select Program:** A dropdown menu with the placeholder "--Select Program--".
- Gender:** Two radio buttons labeled "Male" and "Female".
- Address:** A text input field with the placeholder "Enter your address".



The form is titled 'Date of Birth:' and has a text input field with a placeholder 'dd - mm - yyyy' and a calendar icon. Below this is a 'Nationality:' field with a placeholder 'Enter your nationality'. The 'Upload Photo:' field has a 'Choose File' button and the text 'No file chosen'. The 'Personal Website (Optional):' field has a placeholder 'Enter your personal website URL (optional)'. At the bottom are 'Submit' and 'Reset' buttons.

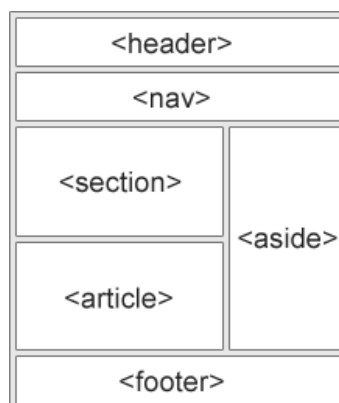
## EXPERIMENT 5

Develop HTML page named as “newspaper.html” having variety of HTML semantic elements with background colors, text-colors & size for figure, table, aside, section, article, header, footer... etc.

In HTML there are some **semantic elements** that can be used to define different parts of a web page. Semantic tags in HTML are elements that clearly describe their meaning in a human- and machine-readable way. These tags provide additional information about the content they contain, making it easier for browsers, developers, and assistive technologies to understand the structure and purpose of the content.

- ❖ Semantic tags help screen readers and other assistive devices to better interpret content for users with disabilities.
- ❖ Code becomes more readable and maintainable for developers, making it easier to understand the structure of the page at a glance.

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>



**Newspaperstyle.css:**

```
/* General styling */
body {
    font-family: Arial, sans-serif;
    line-height: 1.6;
    margin: 0;
    background-color: #f0f0f0;
}

h1, h2, h3 {
    color: darkblue;
    margin-bottom: 10px;
}

p {
    color: #333;
    font-size: 18px;
}

/* Header styling */
header {
    background-color: navy;
    color: white;
    padding: 10px 0;
    text-align: center;
}

header nav ul {
    list-style: none;
    padding: 0;
}

header nav ul li {
    display: inline;
    margin: 0 10px;
}

header nav ul li a {
    color: white;
    text-decoration: none;
    font-weight: bold;
}
```

```
header nav ul li a:hover {
    text-decoration: underline;
}

/* Main content and aside */
main {
    display: flex;
    justify-content: space-between;
    margin: 20px;
}

section {
    flex: 70%;
    background-color: white;
    padding: 20px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

aside {
    flex: 25%;
    background-color: #f9f9f9;
    padding: 20px;
    margin-left: 20px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

aside h3 {
    color: darkgreen;
    font-size: 20px;
}

aside ul {
    list-style-type: square;
    margin: 0;
    padding: 0;
}

aside ul li {
    margin-bottom: 10px;
}

/* Table styling */
table {
    width: 100%;
```

```
border-collapse: collapse;
margin-top: 20px;
background-color: white;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
```

```
table th, table td {
padding: 10px;
border: 1px solid #ddd;
text-align: left;
}
```

```
table th {
background-color: #4CAF50;
color: white;
}
```

```
table tr:nth-child(even) {
background-color: #f2f2f2;
}
```

/\* Footer styling \*/

```
footer {
background-color: #333;
color: white;
text-align: center;
padding: 10px 0;
margin-top: 20px;
}
```

```
footer a {
color: lightblue;
text-decoration: none;
}
```

```
footer a:hover {
text-decoration: underline;
}
```

/\* Image and figure styling \*/

```
figure {
margin: 20px 0;
text-align: center;
}
```

```
figure img {  
    border: 3px solid #ddd;  
    border-radius: 10px;  
}
```

```
figcaption {  
    font-size: 16px;  
    color: #555;  
}
```

### **Newspaper.html:**

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Bangalore News</title>  
    <link rel="stylesheet" href="newspaper-style.css">  
</head>  
<body>  
  
    <!-- Header -->  
    <header>  
        <h1>Bangalore News</h1>  
        <nav>  
            <ul>  
                <li><a href="#">Home</a></li>  
                <li><a href="#">Local</a></li>  
                <li><a href="#">Technology</a></li>  
                <li><a href="#">Events</a></li>  
                <li><a href="#">Contact</a></li>  
            </ul>  
        </nav>  
    </header>  
  
    <!-- Main Section with Aside -->  
    <main>
```

<section>

<article>

<h2>Breaking News: Bengaluru Metro Expansion</h2>

<p>Bangalore is set to further expand its metro network. The Purple and Green Lines will connect new areas across the city, reducing traffic congestion and making commutes easier. The new metro routes will benefit residents in Whitefield, Kengeri, and Bannerghatta Road.</p>

<figure>



<figcaption>New Metro routes under development</figcaption>

</figure>

<p>The expansion project is expected to be completed by 2026 and is part of a broader effort to enhance public transportation in the city.</p>

</article>

<article>

<h2>Local News: Cubbon Park Renovation</h2>

<p>The Bangalore City Corporation has announced a major renovation of Cubbon Park, one of the city's most beloved green spaces. The project will introduce eco-friendly lighting, improved walking paths, and better security features.</p>

<figure>



<figcaption>Ongoing Renovation at Cubbon Park</figcaption>

</figure>

<p>Residents are excited about the initiative, which aims to maintain the park's natural beauty while making it safer and more accessible for visitors.</p>

</article>

<article>

<h2>World News: Bangalore Joins Global Climate Summit</h2>

<p>As a major tech hub, Bangalore is participating in the Global Climate Summit. The city's efforts in promoting renewable energy and reducing carbon emissions have gained international recognition. The government is pushing for green initiatives in IT parks and residential areas.</p>

</article>

</section>

<!-- Aside for additional information -->

<aside>

### <h3>Popular Articles</h3>

<ul>

<li><a href="#">How Bangalore's Startups are Embracing AI</a></li>

<li><a href="#">Best Cafes to Visit in Koramangala</a></li>

<li><a href="#">Traffic Solutions for Outer Ring Road</a></li>

<li><a href="#">Weekend Getaways Near Bangalore</a></li>

</ul>

### <h3>Weather Report</h3>

<p>Sunny with a chance of rain. High of 29°C, low of 20°C. Typical Bangalore weather for this time of year, so be prepared for unexpected showers!</p>

</aside>

</main>

<!-- Table for recent sports events -->

<section>

## <h2>Recent Sports Events in Bangalore</h2>

<table>

<thead>

<tr>

<th>Event</th>

<th>Date</th>

<th>Location</th>

<th>Winner</th>

</tr>

</thead>

<tbody>

<tr>

<td>Kabaddi League Final</td>

<td>September 22, 2024</td>

<td>Kanteerava Stadium</td>

<td>Bengaluru Bulls</td>

</tr>

<tr>

<td>Cricket Premier League</td>

<td>September 19, 2024</td>

<td>M. Chinnaswamy Stadium</td>

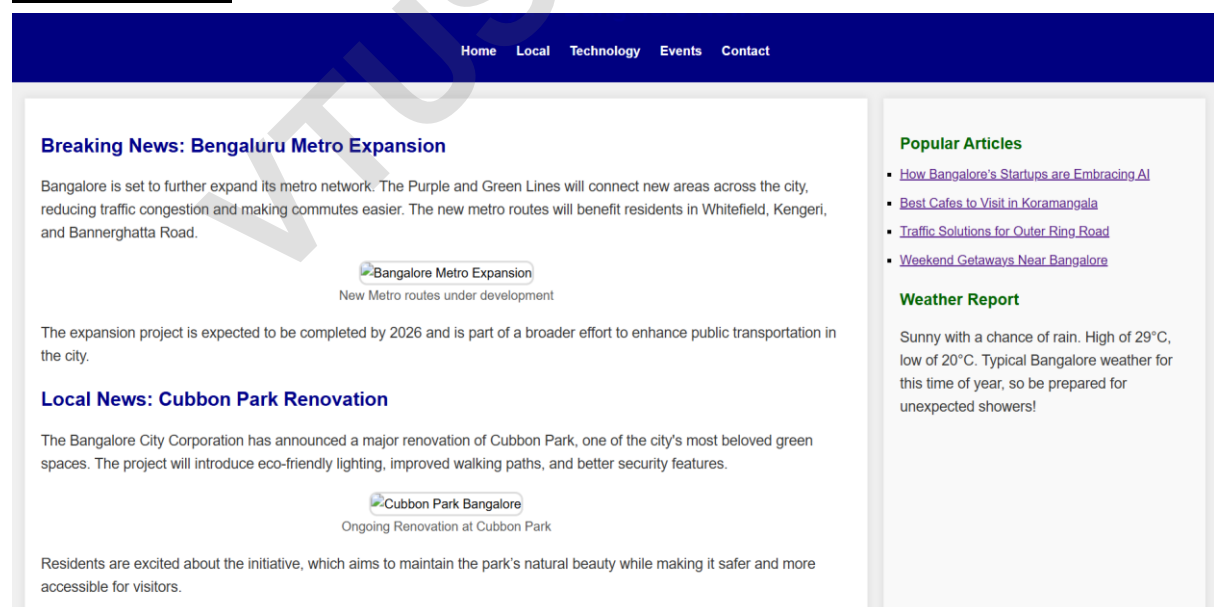
<td>Royal Challengers Bangalore</td>

</tr>

```
<tr>
  <td>Bangalore Marathon</td>
  <td>September 16, 2024</td>
  <td>Cubbon Park</td>
  <td>Runner A</td>
</tr>
</tbody>
</table>
</section>

<!-- Footer -->
<footer>
  <p>&copy; 2024 Bangalore News. All rights reserved.</p>
  <p>Contact us: <a
href="mailto:info@bangalorenews.com">info@divyasnews.com</a></p>
</footer>
</body>
</html>
```

### Sample Output:





## EXPERIMENT 6

**Apply HTML, CSS and JavaScript to design a simple calculator to perform the following operations: sum, product, difference, remainder, quotient, power, square-root and square.**

### Calculator.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple Calculator</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="calculator">
    <h1>Simple Calculator</h1>
    <form id="calculator-form">
      <input type="number" id="num1" placeholder="Enter first number" required>
      <input type="number" id="num2" placeholder="Enter second number" required>
      <select id="operation">
        <option value="sum">Sum</option>
        <option value="product">Product</option>
        <option value="difference">Difference</option>
        <option value="remainder">Remainder</option>
        <option value="quotient">Quotient</option>
        <option value="power">Power</option>
        <option value="square-root">Square Root</option>
        <option value="square">Square</option>
      </select>
      <button type="button" onclick="calculate()">Calculate</button>
    </form>
    <div id="result">
      <h2>Result:</h2>
      <p id="output">Please enter numbers and select an operation.</p>
    </div>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

### Calculator-styles.css:

/\* General body styling \*/

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f4f4f4;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    height: 100vh;  
    margin: 0;  
}
```

/\* Calculator container styling \*/

```
.calculator {  
    background-color: #fff;  
    border-radius: 8px;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
    padding: 20px;  
    width: 300px;  
    text-align: center;  
}
```

/\* Input and select styling \*/

```
input[type="number"], select {  
    width: calc(100% - 22px);  
    padding: 10px;  
    margin: 10px 0;  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    font-size: 16px;  
}
```

/\* Button styling \*/

```
button {  
    background-color: #4CAF50;  
    color: white;  
    border: none;  
    padding: 10px 20px;  
    margin: 10px 0;  
    border-radius: 4px;  
    font-size: 16px;  
    cursor: pointer;  
}  
button:hover {  
    background-color: #45a049;  
}
```

/\* Result section styling \*/

```
#result {
```

```
margin-top: 20px;
}
#output {
  font-size: 18px;
  color: #333;
}
```

### Java script file

#### Calculator.js

```
function calculate() {
  const num1 = parseFloat(document.getElementById('num1').value);
  const num2 = parseFloat(document.getElementById('num2').value);
  const operation = document.getElementById('operation').value;
  let result;
  if (isNaN(num1) || isNaN(num2)) {
    result = "Please enter valid numbers.";
  } else {
    switch (operation) {
      case 'sum':
        result = num1 + num2;
        break;
      case 'product':
        result = num1 * num2;
        break;
      case 'difference':
        result = num1 - num2;
        break;
      case 'remainder':
        result = num1 % num2;
        break;
      case 'quotient':
        result = num1 / num2;
        break;
      case 'power':
        result = Math.pow(num1, num2);
        break;
      case 'square-root':
        result = Math.sqrt(num1);
        break;
      case 'square':
        result = num1 * num1;
        break;
      default:
        result = "Unknown operation.";
    }
  }
}
```

```
    }  
  }  
  document.getElementById('output').textContent = `The result is: ${result}`;  
}
```

### Explanation of js code:

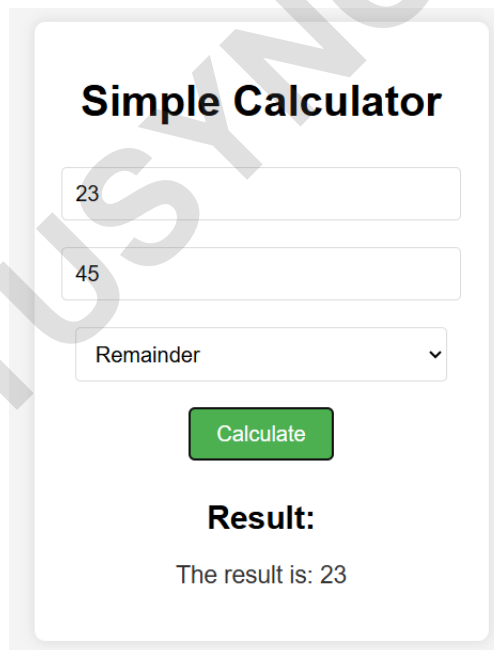
The function, calculate(), performs various mathematical operations based on user input.

- It retrieves two numbers from input fields with IDs num1 and num2 and converts them to floating-point numbers using parseFloat. It also fetches the desired operation from a dropdown or selection input with ID operation.
- It checks if the inputs num1 or num2 are not numbers (NaN). If either input is invalid, it assigns a message "Please enter valid numbers." to result.
- Using a switch statement, it evaluates the selected operation and performs the corresponding calculation. Ex: option 'sum' performs num1+num2.

The function call calculate() happens when user clicks the button provided in the web page using html code:

```
<button onclick="calculate()">Calculate</button>
```

### Sample output:



**Simple Calculator**

23

45

Remainder ▼

Calculate

**Result:**

The result is: 23

## EXPERIMENT 7

### 7. Develop JavaScript program (with HTML/CSS) for:

- a) Converting JSON text to JavaScript Object
- b) Convert JSON results into a date
- c) Converting From JSON To CSV and CSV to JSON
- d) Create hash from string using `crypto.createHash()` method

About the Experiment:

#### ➤ Convert JSON Text to JavaScript Object

- JSON (JavaScript Object Notation) is a text format for representing data. Converting JSON text to a JavaScript object means parsing it, so you can work with the data in JavaScript.

**Example:** Use `JSON.parse()` to turn a JSON string into a JavaScript object.

#### ➤ Convert JSON Results into a Date

- If the JSON data contains a date in string format (e.g., "2024-11-20"), you can convert it into a JavaScript Date object, allowing you to work with it as a date.

#### ➤ Convert JSON to CSV and CSV to JSON

- CSV (Comma-Separated Values) is another format for storing data, especially in spreadsheets.
- This task requires converting a JSON object to CSV format and vice versa. JSON-to-CSV and CSV-to-JSON conversion functions can be written in JavaScript.

#### ➤ Create a Hash from a String Using `crypto.createHash()`

- Hashing is a way to convert a string into a fixed-size string, usually for security. The `crypto.createHash()` method (available in Node.js) is used to generate hashes in JavaScript.

**About JSON:** JSON is a **text-based format** that represents structured data in a way that both humans and machines can understand. It consists of key-value pairs, arrays, and objects, making it flexible and lightweight. JSON is often used to send data between a server and a web application. JSON is built from one or more keys and values. Keys can be any string, while values can be strings, numbers, arrays, additional objects, or the literals false, true, and null.

Ex:

```
{ "name": "Alice",  
  "age": 30,  
  "isStudent": false,  
  "hobbies": ["reading", "traveling", "sports"],  
  "address": { "street": "123 Main St", "city": "Anytown", "zip": "12345" } }
```

JSON is language-independent, making it ideal for exchanging data across different programming languages and platforms.

**JavaScript Objects :** Objects are variables in javascript. But objects can contain many values.

Ex: we can assign **many values** (Fiat, 500, white) to an **object** named car as follows:

```
const car = {type:"Fiat", model:"500", color:"white"};
```

**CSV(comma-separated values):** A .csv file is a text file format that stores data in a table structure, with commas separating values and newlines separating records.

**Html file:**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Conversion and Hashing Tool</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 20px;
    }
    textarea, input, button {
      margin: 10px 0;
      padding: 10px;
      width: 100%;
      max-width: 500px;
    }
    button {
      cursor: pointer;
    }
    pre {
      background: #f4f4f4;
      padding: 10px;
      white-space: pre-wrap;
    }
  </style>
</head>
<body>

  <h1>Conversion and Hashing Tool</h1>

  <!-- Convert JSON to JavaScript Object -->
  <h3>Convert JSON to JavaScript Object</h3>
  <textarea id="jsonInput" rows="5" placeholder="Enter JSON text"></textarea>
  <button onclick="convertJsonToObject()">Convert to JS Object</button>
  <pre id="jsonObjectOutput"></pre>
```

```
<!-- Convert JSON Date String to Date Object -->
<h3>Convert JSON Date String to JavaScript Date Object</h3>
<input id="jsonDataInput" type="text" placeholder="Enter JSON date string">
<button onclick="convertJsonToDate()">Convert to Date</button>
<pre id="jsonDataOutput"></pre>
```

```
<!-- Convert CSV to JSON -->
<h3>Convert CSV to JSON</h3>
<textarea id="csvInput" rows="5" placeholder="Enter CSV text"></textarea>
<button onclick="convertCsvToJson()">Convert CSV to JSON</button>
<textarea id="jsonOutput" rows="5" readonly></textarea>
```

```
<!-- Convert JSON to CSV -->
<h3>Convert JSON to CSV</h3>
<button onclick="convertJsonToCsv()">Convert JSON to CSV</button>
<pre id="csvOutput"></pre>
```

```
<!-- Create Hash from a String -->
<h3>Create SHA-256 Hash from String</h3>
<input id="hashInput" type="text" placeholder="Enter string to hash">
<button onclick="createHash()">Create Hash</button>
<pre id="hashOutput"></pre>
```

```
<script>
  // Function to convert JSON text to a JavaScript Object
  function convertJsonToObject() {
    const jsonInput = document.getElementById('jsonInput').value;
    try {
      const jsonObject = JSON.parse(jsonInput);
      document.getElementById('jsonObjectOutput').textContent =
JSON.stringify(jsonObject, null, 2);
    } catch (error) {
      document.getElementById('jsonObjectOutput').textContent = 'Invalid JSON input.';
    }
  }

  // Function to convert JSON date string to Date object
  function convertJsonToDate() {
    const jsonString = document.getElementById('jsonDataInput').value;
    try {
      const date = new Date(jsonString);
      if (isNaN(date.getTime())) {
        throw new Error('Invalid date string');
      }
    }
  }
</script>
```

```
        document.getElementById('jsonDataOutput').textContent = date.toString();
    } catch (error) {
        document.getElementById('jsonDataOutput').textContent = 'Invalid JSON date
string.';
    }
}
```

// Function to convert CSV to JSON

```
function convertCsvToJson() {
    const csvInput = document.getElementById('csvInput').value;
    const lines = csvInput.trim().split('\n');
    const result = [];
    const headers = lines[0].split(',');
    for (let i = 1; i < lines.length; i++) {
        const obj = {};
        const currentLine = lines[i].split(',');
        headers.forEach((header, index) => {
            obj[header] = currentLine[index];
        });
        result.push(obj);
    }
    document.getElementById('jsonOutput').value = JSON.stringify(result, null, 2);
}
```

// Function to convert JSON to CSV

```
function convertJsonToCsv() {
    const jsonInput = document.getElementById('jsonOutput').value;
    const jsonArray = JSON.parse(jsonInput);
    const csvRows = [];
    const headers = Object.keys(jsonArray[0]);
    csvRows.push(headers.join(','));
    for (const row of jsonArray) {
        const values = headers.map(header => row[header]);
        csvRows.push(values.join(','));
    }
    document.getElementById('csvOutput').textContent = csvRows.join('\n');
}
```

// Function to create a hash from a string using crypto API

```
function createHash() {
    const crypto = window.crypto || window.msCrypto; // For IE 11 and other browsers
    const text = document.getElementById('hashInput').value;
    const encoder = new TextEncoder();
    const data = encoder.encode(text);
```



```
crypto.subtle.digest('SHA-256', data).then(hashBuffer => {
  const hashArray = Array.from(new Uint8Array(hashBuffer));
  const hashHex = hashArray.map(byte => byte.toString(16).padStart(2, '0')).join("");
  document.getElementById('hashOutput').textContent = hashHex;
}).catch(err => {
  document.getElementById('hashOutput').textContent = 'Error creating hash.';
});
}
</script>
```

</body>

</html>

### Sample output:

## Conversion and Hashing Tool

### Convert JSON to JavaScript Object

```
{
  "name": "John",
  "age": 30,
  "city": "New York"
}
```

Convert to JS Object

```
{
  "name": "John",
  "age": 30,
  "city": "New York"
}
```

### Convert JSON Date String to JavaScript Date Object

Convert to Date

Thu Oct 03 2024 21:00:00 GMT+0530 (India Standard Time)

### Convert CSV to JSON

```
name,age,city
Alice,28,New York
Bob,32,San Francisco
Charlie,25,Chicago
```

Convert CSV to JSON

```
[
  {
    "name": "Alice",
    "age": "28",
    "city": "New York"
  }
]
```

### Convert JSON to CSV

Convert JSON to CSV

```
name,age,city
Alice,28,New York
Bob,32,San Francisco
Charlie,25,Chicago
```

### Create SHA-256 Hash from String

Create Hash

## EXPERIMENT 8

**8. a) Develop a PHP program (with HTML/CSS) to keep track of the number of visitors visiting the web page and to display this count of visitors, with relevant headings.**

**b) Develop a PHP program (with HTML/CSS) to sort the student records which are stored in the database using selection sort.**

### **a. PHP Program to Track and Display the Number of Visitors**

**Steps to follow:**

#### **Step 1: Set Up Your Project Files**

1. Create a New Folder for your project in Visual Studio Code.
2. Inside this folder, create:
  - index.php: This will contain your PHP and HTML code.
  - visitor\_count.txt: An empty file where the visitor count will be stored.
  - phpStyle.css (optional): Any CSS you want to style your page.

#### **Step 2: Save Your PHP and HTML Code in index.php**

#### **Step 3: Run the PHP File with a Local Server**

1. Open the Integrated Terminal in Visual Studio Code by selecting View > Terminal.
2. Start PHP's Built-In Server by navigating to the folder where index.php is saved and running:

**php -S localhost:8000**

3. **Access the Page in Your Browser:** Open a browser and go to <http://localhost:8000/index.php>.
4. **Check the Visitor Counter:** Each time you refresh the page, the visitor count should increase, as the PHP code increments the count and saves it in visitor\_count.txt.

#### **index.php:**

```
<?php
// File to store the visitor count
$counterFile = 'visitor_count.txt';
// Check if the counter file exists
if (!file_exists($counterFile)) {
    // If not, create the file and initialize the count to 0
    file_put_contents($counterFile, '0');
}
// Read the current count
$visitorCount = (int)file_get_contents($counterFile);
// Increment the count
$visitorCount++;
// Write the new count back to the file
file_put_contents($counterFile, $visitorCount);
?>
```

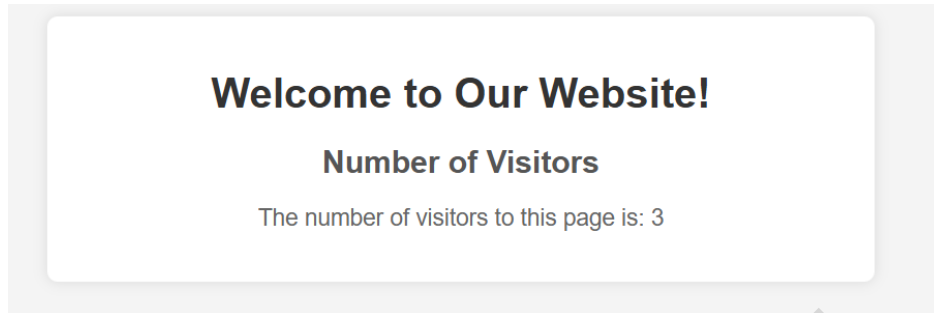
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Visitor Counter</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Welcome to Our Website!</h1>
    <h2>Number of Visitors</h2>
    <p>The number of visitors to this page is: <?php echo $visitorCount; ?></p>
  </div>
</body>
</html>
```

**styles.css:**

```
/* General body styling */
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 0;
}
/* Container styling */
.container {
  max-width: 600px;
  margin: 20px auto;
  padding: 20px;
  background-color: #fff;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  text-align: center;
}
/* Heading styling */
h1 {
  color: #333;
}
h2 {
  color: #555;
}
/* Paragraph styling */
```

```
p {  
    font-size: 18px;  
    color: #666;  
}
```

**Output:**



**b. PHP Program to Sort Student Records Using Selection Sort**

Steps:

**Step 1: Start MySQL Server**

**1. Open MySQL:**

- If you're using MySQL Workbench, launch it and connect to your MySQL server.
- If you're using the command line, open a terminal and type: `mysql -u root -p`
- Enter your MySQL root password to access the MySQL prompt.

**Step 2: Create the Database and Table**

**1. Create Database school:**

- At the MySQL prompt, run: `create database school;`
- Create the students table with columns for id, name, and score:

```
CREATE TABLE students (  
    id CHAR(10) PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    score INT NOT NULL  
);
```

**Step 3: Insert Sample Data**

Add some sample student records to test your selection sort program.

```
INSERT INTO students (name, score) VALUES
```

```
('1EP22CS010','Alice', 85),  
( '1EP22CS012', 'Bob', 92),  
( '1EP22CS016', 'Charlie', 78),  
( '1EP22CS020','David', 90),  
( '1EP22CS022','Eve', 88);
```

**Step 5: Run Your PHP Program**

Now that the database and table are set up, follow these steps to run the PHP program:

**1. Save Your Files:**

- Save your PHP files (e.g., db\_config.php for the database connection, and sort\_students.php for the main sorting program).
  - Save styles.css if you want to apply additional styling.
2. Start PHP's Built-In Server:
- Open the terminal in Visual Studio Code and navigate to the folder containing your PHP files.
  - Start the PHP server by running: `php -S localhost:8000`
  - This starts a local server at <http://localhost:8000>.
3. Access the Program in Your Browser:
- Open a web browser and go to `http://localhost:8000/sort_students.php`.
  - You should see a table displaying the sorted student records by score in descending order.

**db\_config.php:**

```
<?php
$servername = "localhost";
$username = "root"; // Update with your database username
$password = ""; // Update with your database password
$dbname = "school";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

**sort\_students.php**

```
<?php
include 'db_config.php';
// Fetch student records
$sql = "SELECT id, name, score FROM students";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    $students = [];
    // Store all student records in an array
    while ($row = $result->fetch_assoc()) {
        $students[] = $row;
    }
    // Selection sort algorithm to sort by score in descending order
    $n = count($students);
```

```
for ($i = 0; $i < $n - 1; $i++) {
    $maxIdx = $i;
    for ($j = $i + 1; $j < $n; $j++) {
        if ($students[$j]['score'] > $students[$maxIdx]['score']) {
            $maxIdx = $j;
        }
    }
    // Swap the elements
    $temp = $students[$i];
    $students[$i] = $students[$maxIdx];
    $students[$maxIdx] = $temp;
}
} else {
    $students = [];
}

$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sorted Student Records</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Sorted Student Records</h1>
        <?php if (empty($students)): ?>
            <p>No student records found.</p>
        <?php else: ?>
            <table>
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Name</th>
                        <th>Score</th>
                    </tr>
                </thead>
                <tbody>
                    <?php foreach ($students as $student): ?>
                        <tr>
```

```
<td><?php echo htmlspecialchars($student['id']); ?></td>
<td><?php echo htmlspecialchars($student['name']); ?></td>
<td><?php echo htmlspecialchars($student['score']); ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php endif; ?>
</div>
</body>
</html>
```

### styles.css (Additional Styles)

```
/* Table styling */
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}
table th, table td {
    border: 1px solid #ddd;
    padding: 10px;
    text-align: left;
}
table th {
    background-color: #4CAF50;
    color: white;
}
table tr:nth-child(even) {
    background-color: #f2f2f2;
}
```

### Output:

### Sorted Student Records

ID	Name	Score
1EP22CS010	MANOJ	92
1EP22CS012	DINOY	80
1EP22CS020	SURESH	62

## EXPERIMENT 9

**Develop jQuery script (with HTML/CSS) for: a. Appends the content at the end of the existing paragraph and list. b. Change the state of the element with CSS style using animate () method c. Change the color of any div that is animated.**

**Instructions: Create a new folder to keep .html, .js and .css file. The execute .html file.**

### About the code:

- ❖ **Appending Content to Paragraph and List:**
  - The append() function in jQuery is used to add content to the end of an element.
  - We use #appendParagraphBtn and #appendListBtn buttons to trigger append() for the paragraph and list elements, respectively.
- ❖ **Animating Element Using animate():**
  - The animate() function changes the width, height, and fontSize of .animatedDiv.
  - When the .animatedDiv element is clicked, the animate() function runs, resizing the div over 1 second.
- ❖ **Change the Color of the Div After Animation:**
  - A callback function is used after the animate() method completes.
  - Inside the callback, css("background-color", "#ff5733") changes the background color of .animatedDiv to a different color (e.g., #ff5733).

### Step 1: Create html file

#### Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>jQuery User Input Example</title>
  <link rel="stylesheet" href="style.css">
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script src="script.js" defer></script>
</head>
<body>
  <div class="container">
    <h1>jQuery Interactions with User Input</h1>

    <!-- Input for paragraph content -->
    <p id="myParagraph">This is an existing paragraph.</p>
    <input type="text" id="paragraphInput" placeholder="Enter text to add to paragraph">
    <button id="appendParagraphBtn">Add to Paragraph</button>

    <!-- Input for list content -->
```



```
<ul id="myList">
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
<input type="text" id="listInput" placeholder="Enter text to add to list">
<button id="appendListBtn">Add to List</button>

<!-- Animated div -->
<div class="animatedDiv">Click Me</div>
</div>
</body>
</html>
```

## Step 2: Create CSS file

### Style.css

```
/* Basic Styling */
```

```
body {
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  background: linear-gradient(135deg, #6a11cb 0%, #2575fc 100%);
  color: #333;
}
.container {
  text-align: center;
  width: 350px;
  padding: 20px;
  background-color: #ffffffdd;
  border-radius: 12px;
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
}
```

```
/* Input and Button Styling */
```

```
input[type="text"] {
  width: 100%;
  padding: 8px;
  margin: 10px 0;
  border: 1px solid #ddd;
  border-radius: 6px;
}
```

```
button {
    margin: 10px 0;
    padding: 10px 15px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 8px;
    font-size: 0.9em;
    cursor: pointer;
    transition: background-color 0.3s;
    width: 100%;
}

button:hover {
    background-color: #45a049;
}

/* Animated Div Styling */
.animatedDiv {
    width: 100px;
    height: 100px;
    background-color: #007bff;
    margin: 20px auto;
    line-height: 100px;
    color: white;
    font-weight: bold;
    border-radius: 8px;
    cursor: pointer;
    transition: box-shadow 0.3s, transform 0.3s;
}

.animatedDiv:hover {
    box-shadow: 0 8px 15px rgba(0, 123, 255, 0.3);
    transform: scale(1.05);
}
```

### Create java script file

#### Script.js

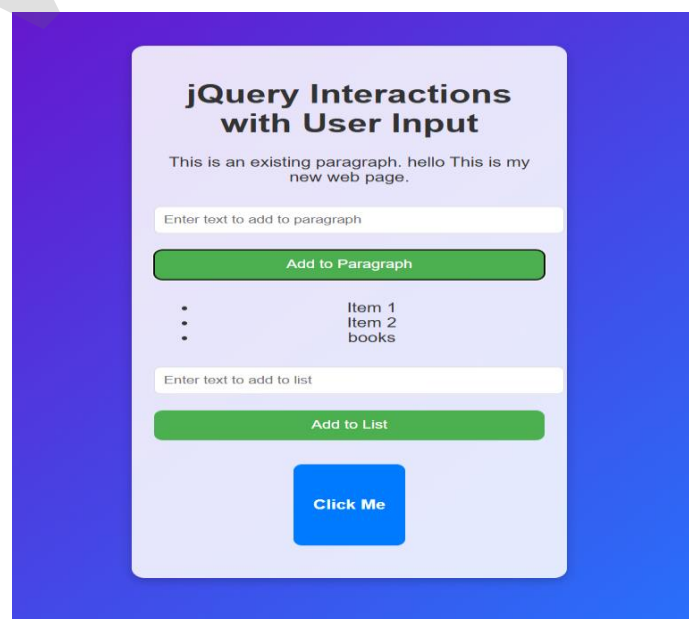
```
$(document).ready(function() {
    // a. Appends user input to paragraph and list
    $("#appendParagraphBtn").click(function() {
        const paragraphText = $("#paragraphInput").val();
        if (paragraphText) {
            $("#myParagraph").append(" " + paragraphText);
            $("#paragraphInput").val(""); // Clear the input field
        }
    });
});
```

```
    } else {  
        alert("Please enter text to add to the paragraph.");  
    }  
});  
  
$("#appendListBtn").click(function() {  
    const listItemText = $("#listInput").val();  
    if (listItemText) {  
        $("#myList").append("<li>" + listItemText + "</li>");  
        $("#listInput").val(""); // Clear the input field  
    } else {  
        alert("Please enter text to add to the list.");  
    }  
});
```

// b. Animate the div and change CSS properties

```
$(".animatedDiv").click(function() {  
    $(this).animate({  
        width: "180px",  
        height: "180px",  
        fontSize: "1.2em",  
        lineHeight: "180px"  
    }, 800, function() {  
        // c. Change the color after animation  
        $(this).css("background-color", "#ff5733");  
    });  
});  
});
```

**Output:**



## EXPERIMENT 10

**Develop a JavaScript program with Ajax (with HTML/CSS) for:** a. Use `ajax()` method (without JQuery) to add the text content from the text file by sending ajax request. b. Use `ajax()` method (with JQuery) to add the text content from the text file by sending ajax request. c. Illustrate the use of `getJSON()` method in JQuery d. Illustrate the use of `parseJSON()` method to display JSON values.

### About the code:

- Vanilla JavaScript AJAX (`loadTextWithoutJQuery` function):
  - a. This function uses `XMLHttpRequest` to send an asynchronous request to `textfile.txt`.
  - b. Once the content is loaded successfully, it displays the text in `#textWithoutJQuery`.
- JQuery AJAX:
  - a. Uses the `$.ajax()` method to load `textfile.txt`.
  - b. On success, it displays the text content in `#textWithJQuery`.
  - c. On error, it shows an error message in the same div.
- `getJSON()` Method:
  - a. Fetches JSON data from `data.json` and iterates through each key-value pair using `$.each()`.
  - b. Displays the JSON content as a list inside `#jsonData`.
- `parseJSON()` Method:
  - a. A hardcoded JSON string is parsed into a JavaScript object using `$.parseJSON()`.
  - b. Each value is then displayed individually in `#parsedJsonData`.

### HTML file:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AJAX and JSON Examples</title>
  <link rel="stylesheet" href="style.css">
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script src="script.js" defer></script>
</head>
<body>
  <div class="container">
    <h1>AJAX and JSON with JavaScript and jQuery</h1>
```

```
<!-- Vanilla JavaScript AJAX -->
<button onclick="loadTextWithoutjQuery()">Load Text (Without jQuery)</button>
<div id="textWithoutjQuery"></div>

<!-- jQuery AJAX -->
<button id="loadTextWithjQuery">Load Text (With jQuery)</button>
<div id="textWithjQuery"></div>

<!-- jQuery getJSON() -->
<button id="loadJson">Load JSON Data (Using getJSON)</button>
<div id="jsonData"></div>

<!-- JSON.parse Example -->
<button id="parseJsonButton">Parse JSON Data</button>
<div id="parsedJsonData"></div>
</div>
</body>
</html>
```

#### Javascript file: script.js

```
function loadTextWithoutjQuery() {
    const xhr = new XMLHttpRequest();
    xhr.open("GET", "textfile.txt", true);
    xhr.onreadystatechange = function () {
        if (xhr.readyState === 4 && xhr.status === 200) {
            // Replace content instead of appending
            document.getElementById("textWithoutjQuery").innerText = xhr.responseText;
        }
    };
    xhr.send();
}

$(document).ready(function() {
    $("#loadTextWithjQuery").click(function() {
        $.ajax({
            url: "textfile.txt",
            type: "GET",
            success: function(data) {
                // Replace content instead of appending
                $("#textWithjQuery").text(data);
            },
            error: function() {
                $("#textWithjQuery").text("Failed to load content.");
            }
        });
    });
});
```

```
    });  
  });  
});  
$(document).ready(function() {  
  $("#loadJson").click(function() {  
    $.getJSON("data.json", function(data) {  
      let content = "<ul>";  
      $.each(data, function(key, value) {  
        content += `<li>${key}: ${value}</li>`;  
      });  
      content += "</ul>";  
      $("#jsonData").html(content);  
    });  
  });  
});  
$(document).ready(function() {  
  $("#parseJsonButton").click(function() {  
    // Simulate JSON data  
    const jsonString = '{"name": "John", "age": 25, "city": "New York"}';  
    const parsedData = $.parseJSON(jsonString);  
  
    let output = `<p>Name: ${parsedData.name}</p>`;  
    output += `<p>Age: ${parsedData.age}</p>`;  
    output += `<p>City: ${parsedData.city}</p>`;  
  
    $("#parsedJsonData").html(output);  
  });  
});
```

### Stylesheet:style.css

```
/* Styling for container and buttons */  
body {  
  font-family: Arial, sans-serif;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
  margin: 0;  
  background: #f5f5f5;  
}  
  
.container {  
  text-align: center;
```

```
width: 400px;
padding: 20px;
background-color: #fff;
border-radius: 8px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}
```

```
button {
margin: 10px 0;
padding: 10px 15px;
background-color: #4CAF50;
color: white;
border: none;
border-radius: 8px;
font-size: 0.9em;
cursor: pointer;
width: 100%;
}
```

```
button:hover {
background-color: #45a049;
}
```

```
div {
margin: 10px 0;
padding: 10px;
border: 1px solid #ddd;
border-radius: 6px;
background-color: #f9f9f9;
}
```

#### **Textfile: textfile.txt**

Welcome to our website! Here's some information loaded dynamically using AJAX.

#### **JSON file: data.json**

```
{
  "name": "Alice",
  "age": 30,
  "city": "Los Angeles"
}
```

**Output:**

## AJAX and JSON with JavaScript and jQuery

Load Text (Without jQuery)

Welcome to our website! Here's some information loaded dynamically using AJAX.

Load Text (With jQuery)

Welcome to our website! Here's some information loaded dynamically using AJAX.

Load JSON Data (Using getJSON)

•  
•  
•

name: Alice  
age: 30  
city: Los Angeles

Parse JSON Data

Name: John  
Age: 25  
City: New York



### VIVA QUESTIONS

1. What is web technology?  
**A:** Web technology encompasses the tools and techniques used to communicate between different devices over the internet, including protocols, languages (HTML, CSS, JavaScript), frameworks, and server-side technologies.
2. What is the difference between a website and a web application?  
**A:** A website is a collection of static or dynamic pages accessible via the internet, typically for information display. A web application, however, is an interactive program that enables user actions and processing, often featuring complex interactions and data handling (e.g., online banking, social media).
3. What are front-end and back-end technologies?  
**A:** Front-end technologies manage the user interface and experience, using HTML, CSS, and JavaScript. Back-end technologies handle server-side operations, databases, and business logic, typically using languages like PHP, Python, Node.js, and databases like MySQL or MongoDB.
4. What is the role of HTML in web development?  
**A:** HTML is a markup language that structures the content on a web page, defining elements like headings, paragraphs, images, links, tables, and forms.
5. What are semantic HTML tags?  
**A:** Semantic HTML tags, like `<header>`, `<footer>`, `<article>`, and `<section>`, provide meaning to the content, enhancing accessibility and SEO by indicating the role of each section of content.
6. What is the purpose of the `<meta>` tag in HTML?  
**A:** The `<meta>` tag provides metadata about the HTML document, such as character encoding, author information, and viewport settings, and can improve SEO and page behavior.
7. What is CSS used for in web development?  
**A:** CSS is used to style and layout HTML elements, controlling design aspects like colors, fonts, spacing, and layout to create visually appealing and responsive web pages.
8. What are CSS selectors?  
**A:** CSS selectors are patterns used to select and style specific elements on a web page. Examples include class selectors (`.className`), ID selectors (`#idName`), and tag selectors (`elementName`).
9. How do you make a website responsive using CSS?  
**A:** A responsive website adapts to different screen sizes using CSS techniques like media queries, flexible grids, and responsive units (e.g., %, vw, vh), making content accessible on mobile, tablet, and desktop devices.
10. What is JavaScript, and why is it important?  
**A:** JavaScript is a client-side scripting language that adds interactivity, dynamic content, and logic to web pages. It enables features like animations, form validation, data fetching, and user interaction.

11. What is DOM manipulation in JavaScript?

**A:** DOM (Document Object Model) manipulation involves changing the structure, style, or content of a webpage dynamically using JavaScript methods like `getElementById`, `querySelector`, and `appendChild`.

12. What is an event in JavaScript?

**A:** An event is an action or occurrence detected by JavaScript, such as a click, hover, or form submission. JavaScript can use events to trigger specific functions, making web pages interactive.

13. What is AJAX?

**A:** AJAX (Asynchronous JavaScript and XML) is a technique for updating parts of a web page without reloading the entire page. It uses JavaScript to send and retrieve data asynchronously from a server.

14. What is JSON, and why is it used?

**A:** JSON (JavaScript Object Notation) is a lightweight data format used for data exchange between client and server. It is easy to read and parse in JavaScript, making it ideal for web applications.

15. How does AJAX improve user experience on a website?

**A:** AJAX allows for seamless data loading and updates without page refreshes, providing a smoother and faster user experience by only reloading the necessary parts of a page.

16. What is jQuery?

**A:** jQuery is a JavaScript library that simplifies tasks like DOM manipulation, event handling, animation, and AJAX. It allows developers to write less code to achieve common JavaScript tasks.

17. How do you select an element in jQuery?

**A:** Elements in jQuery are selected using the `$()` function, similar to CSS selectors. For example, `$("#id")` selects an element by ID, and `$(".class")` selects elements by class.

18. What is the purpose of the `.animate()` method in jQuery?

**A:** The `.animate()` method in jQuery enables custom animations by allowing developers to specify CSS properties and animate their changes over time.

19. What is PHP used for in web development?

**A:** PHP is a server-side scripting language commonly used for building dynamic websites, handling form data, managing databases, and generating HTML content on the server.

20. How do you connect to a database in PHP?

**A:** You can connect to a database in PHP using `mysqli_connect()` or PDO (PHP Data Objects) to establish a connection, execute queries, and fetch data.

21. What is the difference between `echo` and `print` in PHP?

**A:** Both `echo` and `print` are used to output data in PHP, but `echo` is slightly faster and allows multiple arguments, while `print` only accepts one argument and returns a value.

22. What is HTTP?

**A:** HTTP (Hypertext Transfer Protocol) is the foundational protocol for data communication on the web, defining how messages are formatted and transmitted.

23. What is HTTPS, and why is it important?  
**A:** HTTPS (Hypertext Transfer Protocol Secure) is an encrypted version of HTTP that provides a secure connection for transmitting sensitive data, protecting users' privacy and security.
24. What are cookies in web development?  
**A:** Cookies are small pieces of data stored on the client's browser that allow websites to remember user information, like login credentials or user preferences, across sessions.
25. What is a Web API?  
**A:** A Web API (Application Programming Interface) allows different software applications to communicate with each other over the internet, enabling web applications to access data or services from other systems.
26. What is the Fetch API in JavaScript?  
**A:** The Fetch API provides a modern way to make asynchronous HTTP requests in JavaScript, replacing the older XMLHttpRequest method. It is widely used for making AJAX requests to retrieve or send data.
27. What is REST in web development?  
**A:** REST (Representational State Transfer) is an architectural style for designing networked applications, often used to create APIs that perform CRUD (Create, Read, Update, Delete) operations over HTTP.
28. Why is responsive design important in web development?  
**A:** Responsive design ensures that a website adapts to different screen sizes and devices, providing an optimal user experience on mobile, tablet, and desktop devices.
29. What is the purpose of using an external style sheet?  
**A:** An external style sheet separates CSS from HTML, making it easier to maintain, reuse, and update styles across multiple pages.
30. Why is code validation important for web pages?  
**A:** Code validation checks for errors and ensures adherence to web standards, improving compatibility, accessibility, and SEO.
31. What is the purpose of the <title> tag in HTML?  
**A:** The <title> tag defines the title of the webpage, which appears on the browser tab and helps with search engine optimization.
32. How do you create scrolling text in HTML?  
**A:** By using the <marquee> tag, although it's deprecated. CSS animations are preferred for creating moving text.
33. What is the difference between <h1> and <h6> tags?  
**A:** <h1> is the highest-level heading (largest), and <h6> is the lowest-level heading (smallest) for organizing content hierarchy.
34. How do <b> and <strong> tags differ?  
**A:** <b> applies bold styling, while <strong> indicates important text and may affect accessibility.
35. What attributes are used for merging cells in a table?  
**A:** rowspan and colspan attributes.

36. How do you add color to table rows or cells?  
**A:** Using the style attribute or CSS classes with background-color properties.
37. What's the purpose of <thead>, <tbody>, and <tfoot> in a table?  
**A:** They separate different sections of a table: <thead> for the header, <tbody> for the body, and <tfoot> for the footer.
38. What is the difference between an ID and a class in CSS?  
**A:** An ID is unique and used once on a page (#id), while a class can be reused across multiple elements (.class).
39. How do CSS selectors work?  
**A:** CSS selectors apply styles to elements based on attributes like tag name, class, ID, or other attributes.
40. What is the purpose of an external CSS file?  
**A:** To separate styling from HTML, making code modular, maintainable, and reusable across multiple pages.
41. How do you change the color of form elements in CSS?  
**A:** By using CSS properties like background-color, color, and border-color.
42. What are different types of input elements in HTML?  
**A:** Common input types include text, password, checkbox, radio, submit, email, date, and number.
43. How do you align form fields using HTML and CSS?  
**A:** Using a table or CSS properties like display: flex or grid.
44. What is the purpose of semantic elements like <header>, <section>, and <article>?  
**A:** Semantic elements add meaning to the content structure, aiding in readability and accessibility.
45. How does <figure> differ from <img>?  
**A:** <figure> is used to group an image and caption with <figcaption>, providing context.
46. How do you capture input values from an HTML form in JavaScript?  
**A:** By using document.getElementById("inputID").value or similar methods.
47. What is the purpose of Math.pow() in JavaScript?  
**A:** It calculates the power of a number, e.g., Math.pow(base, exponent).
48. How do you display the result of a calculation in HTML?  
**A:** By using JavaScript to set the innerHTML or value of an HTML element.
49. How do you convert JSON text to a JavaScript object?  
**A:** Using JSON.parse(jsonText).
50. What is JSON, and why is it used?  
**A:** JSON (JavaScript Object Notation) is a lightweight data format used for exchanging data between servers and clients.
51. How can you convert JSON results to a date in JavaScript?  
**A:** By parsing the JSON date string and using new Date(dateString).
52. Q:How does PHP keep track of visitor counts?  
**A:** PHP can read from and write to a file to increment the visitor count each time the page is loaded.
53. What is the purpose of file\_get\_contents() in PHP?

- A:** It reads the content of a file into a string.
54. How do you display PHP data on an HTML page?  
**A:** By embedding PHP code within HTML using `<?php echo $variable; ?>`.
55. How does PHP connect to a MySQL database?  
**A:** By using the `mysqli_connect()` function or PDO.
56. What is the purpose of selection sort?  
**A:** Selection sort is an algorithm to sort elements by repeatedly finding the minimum or maximum element and placing it at the correct position.
57. How do you retrieve records from a database in PHP?  
**A:** By executing an SQL query with `mysqli_query()` and fetching results using `mysqli_fetch_assoc()`.
58. How do you append content to an element in jQuery?  
**A:** By using the `.append()` method.
59. What does the `animate()` method do in jQuery?  
**A:** It performs custom animations by changing CSS properties with gradual transitions.
60. How do you change the color of an animated element in jQuery?  
**A:** By using `.css()` after the animation completes or by chaining `.animate()`.
61. How do you make an AJAX request in JavaScript without jQuery?  
**A:** By using the XMLHttpRequest object.
62. What does the `getJSON()` method do in jQuery?  
**A:** It makes an AJAX request and automatically parses the JSON response.
63. How do you parse JSON in JavaScript?  
**A:** By using `JSON.parse(jsonString)` to convert JSON text to a JavaScript object.

\*\*\*\*\*