

K.S SCHOOL OF ENGINEERING & MANAGEMENT, BANGALORE-109

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



WEB TECHNOLOGY LAB

BCSL504

Prepared by: -

Ms. Nethravathi K G
Assistant Professor
Dept of CSE, KSSEM

Mrs. Dakshayini G R
Assistant Professor
Dept of CSE, KSSEM

LAB MANUAL

INSTITUTION VISION & MISSION

VISION:

“To impart quality education in engineering and management to meet technological, business and societal needs through holistic education and research”

MISSION:

K.S. School of Engineering and Management shall,

- ❖ Establish state-of-art infrastructure to facilitate effective dissemination of technical and Managerial knowledge.
- ❖ Provide comprehensive educational experience through a combination of curricular and experiential learning, strengthened by industry-institute-interaction.
- ❖ Pursuesocially relevant research and disseminate knowledge.
- ❖ Inculcate leadership skills and foster entrepreneurial spirit among students.

DEPARTMENT VISION & MISSION

VISION:

“To produce quality Computer Science professional, possessing excellent technical knowledge, skills, personality through education and research.”

MISSION:

Department of Computer Science and Engineering shall,

- ❖ Provide good infrastructure and facilitate learning to become competent engineers who meet global challenges.
- ❖ Encourages industry institute interaction to give an edge to the students.
- ❖ Facilitates experimental learning through interdisciplinary projects.
- ❖ Strengthen soft skill to address global challenges.

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	100
Examination type (SEE)	Practical		

Course objectives:

- Learn HTML 5 elements and their use.
- Use of 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.

Sl.NO	Experiments
1	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 (, <u>, <sub>, <sup> etc.)
2	Develop the HTML page named as “Table.html” to display your class time table. a) Provide the title as Time Table with table header and table footer, row-span and col-span etc. b) Provide various colour options to the cells (Highlight the lab hours and elective hours with different colours.) c) Provide colour options for rows.
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.
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.
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.

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.
7	Develop JavaScript program (with HTML/CSS) for: - Converting JSON text to JavaScript Object - Convert JSON results into a date - Converting From JSON To CSV and CSV to JSON - Create hash from string using crypto.createHash() method
8	- Develop a PHP program (with HTML/CSS) to keep track of the number of visitors visiting the webpage and to display this count of visitors, with relevant headings. - Develop a PHP program (with HTML/CSS) to sort the student records which are stored in the database using selection sort.
9	Develop jQuery script (with HTML/CSS) for: - Appends the content at the end of the existing paragraph and list. - Change the state of the element with CSS style using animate() method Change the color of any div that is animated.
10	Develop a JavaScript program with Ajax (with HTML/CSS) for: - Use ajax() method (without JQuery) to add the text content from the text file by sending ajax request. - Use ajax() method (with JQuery) to add the text content from the text file by sending ajax request. - Illustrate the use of getJSON() method in jQuery Illustrate the use of parseJSON() method to display JSON values.
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.	
Course outcomes (Course Skill Set): At the end of the course, the student will be able to: - Design the experiment for the given problem using HTML, Javascript and CSS. - Develop the solution for the given real-world problem using jQuery, Ajax and PHP. Analyze the results and produce substantial written documentation.	

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 answerscript 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.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.geeksforgeeks.org/jquery-tutorial/>

Xampp Folder:

Xampp -> need to start Apache & Mysql

open visual code -> file -> open folder -> local disk c -> xampp -> htdocs -> create new folder -> select folder -> trusted window

open -> local disk c -> xampp -> htdocs -> dashboard -> index -> open in notepad -> insert your folder name -> save.

Command for url:

...Url: localhost/dashboard/usn

...the output window

Program-1

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 (, <u>, <sub>, <sup> etc.)

Source Code: Myfirstwebpage.html

```
<!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>
```

```
  <!-- <style>
```

```
    body {
```

```
      font-family: Arial, sans-serif;
```

```
      line-height: 1.6;
```

```
      margin: 0;
```

```
      padding: 20px;
```

```
    }
```

```
</style> -->
```

```
<link rel="stylesheet" href="style.css">
```

```
</head>
```

```
<body>
```

```
<marquee>Basic HTML Tags</marquee>
```

```
<h1>This is Heading 1</h1>
```

```
<h2>This is Heading 2</h2>
```

```
<h3>This is Heading 3</h3>
```

```
<h4>This is Heading 4</h4>
```

```
<h5>This is Heading 5</h5>
```

```
<h6>This is Heading 6</h6>
```

<p>This is a paragraph. It demonstrates the use of the paragraph tag in HTML. Paragraphs are used to group related content together. </p>

```
<hr>
```

<p>This is another paragraph.
This text appears on a new line due to the line break tag. </p>

```
<blockquote>
```

This is a block quote. It's often used to highlight quoted text from another source.

```
</blockquote>
```

```
<pre>
```

This is preformatted text.

It preserves both spaces and

line breaks, making it useful

for displaying code or ASCII art.

```
</pre>
```

```
<p>
```

Here are examples of logical styles:

Bold text

<i>Italic text</i>

<u>Underlined text</u>

Strong text

Emphasized text

Text with _{subscript} and ^{superscript}

</p>

</body>

</html>

Output:

Basic HTML Tags

This is Heading 1

This is Heading 2

This is Heading 3

This is Heading 4

This is Heading 5

This is Heading 6

This is a paragraph. It demonstrates the use of the paragraph tag in HTML. Paragraphs are used to group related content together.

This is another paragraph.

This text appears on a new line due to the line break tag.

This is a block quote. It's often used to highlight quoted text from another source.

```
This is preformatted text.  
It preserves both spaces and  
line breaks, making it useful  
for displaying code or ASCII art.
```

Here are examples of logical styles:

Bold text

Italic text

Underlined text

Strong text

Emphasized text

Text with _{subscript} and ^{superscript}

Program -2

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

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

Source Code: Table.html

```
<!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>
    body {
      font-family: Arial, sans-serif;
      line-height: 1.6;
      margin: 0;
      padding: 20px;
    }
    table {
      width: 100%;
      border-collapse: collapse;
    }
    th, td {
      border: 1px solid #ddd;
      padding: 8px;
      text-align: center;
    }
  </style>
</head>
<body>
  <table>
    <tr>
      <th>Sl. No.</th>
      <th>Name</th>
      <th>Roll No.</th>
      <th>Lab Hours</th>
      <th>Elective Hours</th>
    </tr>
    <tr>
      <td>1</td>
      <td>Amit</td>
      <td>12345</td>
      <td>9:00 AM - 10:00 AM</td>
      <td>10:00 AM - 11:00 AM</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Priya</td>
      <td>67890</td>
      <td>11:00 AM - 12:00 PM</td>
      <td>12:00 PM - 1:00 PM</td>
    </tr>
    <tr>
      <td>3</td>
      <td>Ravi</td>
      <td>54321</td>
      <td>2:00 PM - 3:00 PM</td>
      <td>3:00 PM - 4:00 PM</td>
    </tr>
    <tr>
      <td>4</td>
      <td>Sneha</td>
      <td>98765</td>
      <td>4:00 PM - 5:00 PM</td>
      <td>5:00 PM - 6:00 PM</td>
    </tr>
  </table>
</body>
</html>
```

```
th {
    background-color: #f2f2f2;
}

.lab-hours {
    background-color: #ffcccb;
}

.elective-hours {
    background-color: #90ee90;
}

.lunch {
    background-color: #ffd700;
}

.odd-row {
    background-color: #f8f8f8;
}

tfoot {
    background-color: #e6e6e6;
    font-weight: bold;
}

</style>
</head>
<body>
<table>
<thead>
<tr>
    <th colspan="7">Class Time Table</th>
</tr>
<tr>
    <th>Time</th>
```

```
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 - 10:00</td>
<td>Math</td>
<td>English</td>
<td>Science</td>
<td>History</td>
<td>Geography</td>
<td rowspan="6">No Classes</td>
</tr>
<tr class="odd-row">
<td>10:00 - 11:00</td>
<td>Physics</td>
<td>Chemistry</td>
<td>Biology</td>
<td class="elective-hours">Computer Science</td>
<td class="elective-hours">Art</td>
</tr>
<tr>
<td>11:00 - 12:00</td>
<td colspan="2">Physics Lab</td>
```

```

        <td class="lab-hours" colspan="2">Chemistry Lab</td>

        <td class="lab-hours">Biology Lab</td>

    </tr>

    <tr class="odd-row">

        <td>12:00 - 13:00</td>

        <td class="lunch" colspan="5">Lunch Break</td>

    </tr>

    <tr>

        <td>13:00 - 14:00</td>

        <td>Literature</td>

        <td>Math</td>

        <td>English</td>

        <td>Physics</td>

        <td>Chemistry</td>

    </tr>

    <tr class="odd-row">

        <td>14:00 - 15:00</td>

        <td class="elective-hours">Music</td>

        <td class="elective-hours">Drama</td>

        <td class="lab-hours" colspan="3">Computer Lab</td>

    </tr>

</tbody>

<tfoot>

    <tr>

        <td colspan="7">* Lab hours are highlighted in pink, elective hours in light
green</td>

    </tr>

</tfoot>

</table>

```

</body>

</html>

Output:

Class Time Table							
Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	
9:00 - 10:00	Math	English	Science	History	Geography	No Classes	
10:00 - 11:00	Physics	Chemistry	Biology	Computer Science	Art		
11:00 - 12:00	Physics Lab		Chemistry Lab		Biology Lab		
12:00 - 13:00	Lunch Break						
13:00 - 14:00	Literature	Math	English	Physics	Chemistry		
14:00 - 15:00	Music	Drama	Computer Lab				
* Lab hours are highlighted in pink, elective hours in light green							

KSSEM

Program - 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.

Source Code: Pgm3.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Sample Styled Page (No Div)</title>

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

</head>

<body>

    <main id="main-content">

        <h2>Welcome to Our Styled Page</h2>

        <p>This is a paragraph right after an h2. It demonstrates the adjacent sibling selector.</p>

        <h3>Hover over me!</h3>

        <hr>

        <p lang="en">This paragraph has a lang attribute, demonstrating the attribute selector.</p>

        <p>Here's a <span class="highlight">highlighted</span> word using the class selector.</p>

        <section>

            <p>This paragraph is inside a section, showing the descendant selector.</p>

            <span>This span is a direct child of the section.</span>

        </section>

        <p>The current date and time: <time datetime="2023-08-31">September 13, 2024</time></p>

        <p>Notice how the first letter of each paragraph is styled differently.</p>
```

```
<article class="special">
```

```
<p>This paragraph is inside an article with class="special".</p>
```

```
</article>
```

```

```

```
<p>Check out this <a href="https://kssem.edu.in/">link</a> to see the updates in kssem.</p>
```

```
</main>
```

```
</body>
```

```
</html>
```

style.css

```
/* Element Selector */
```

```
h2 {
```

```
    color: #2c3e50;
```

```
    font-family: 'Arial', sans-serif;
```

```
    border-bottom: 2px solid #3498db;
```

```
    padding-bottom: 10px;
```

```
}
```

```
/* Element Selector with Pseudo-class */
```

```
h3:hover {
```

```
    color: #e74c3c;
```

```
    cursor: pointer;
```

```
    transition: color 0.3s ease;
```

```
}
```

```
/* Element Selector */
```

```
hr {  
    border: 0;  
    height: 1px;  
    background-image: linear-gradient(to right, rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.75), rgba(0, 0, 0,  
0));  
}  
  
/* Element Selector with Attribute */  
p[lang] {  
    font-style: italic;  
}  
  
/* Class Selector */  
.highlight {  
    background-color: #f1c40f;  
    padding: 5px;  
}  
  
/* ID Selector */  
#main-content {  
    max-width: 800px;  
    margin: 0 auto;  
    padding: 20px;  
    background-color: #ecf0f1;  
}  
  
/* Descendant Selector */  
div p {  
    line-height: 1.6;  
    margin-bottom: 15px;  
}
```

```
/* Child Selector */  
  
div > span {  
    font-weight: bold;  
    color: #16a085;  
}  
  
/* Adjacent Sibling Selector */  
  
h2 + p {  
    font-size: 1.1em;  
    color: #7f8c8d;  
}  
  
/* Attribute Selector */  
  
time[datetime] {  
    color: #8e44ad;  
    font-weight: bold;  
}  
  
/* Pseudo-element Selector */  
  
p::first-letter {  
    font-size: 1.5em;  
    font-weight: bold;  
    color: #c0392b;  
}  
  
/* Multiple Selectors */  
  
img, a {  
    border: 1px solid #bdc3c7;  
    padding: 5px;  
}  
  
/* Pseudo-class Selector for Links */
```

```
a:link, a:visited {  
    color: #3498db;  
    text-decoration: none;  
}  
  
a:hover, a:active {  
    color: #e74c3c;  
    text-decoration: underline;  
}  
  
/* Attribute Selector for Images */  
img[alt] {  
    max-width: 100%;  
    height: auto;  
}  
  
/* Combining Selectors */  
div.special p {  
    text-indent: 20px;  
    color: #27ae60;  
}
```

Output:

Welcome to Our Styled Page

This is a paragraph right after an h2. It demonstrates the adjacent sibling selector.

Hover over me!

This paragraph has a lang attribute, demonstrating the attribute selector.

Here's a **highlighted** word using the class selector.

This paragraph is inside a section, showing the descendant selector.

This span is a direct child of the section.

The current date and time: **September 13, 2024**

Notice how the first letter of each paragraph is styled differently.

This paragraph is inside an article with class="special".



Check out this [link](#) to see the updates in kssem.

Program-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.

Source Code: registration.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Registration Form</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f0f0f0;

            margin: 0;

            padding: 20px;

        }

        h1 {

            color: #333;

            text-align: center;

        }

        table {

            width: 100%;

            max-width: 600px;

            margin: 0 auto;

            background-color: #c08686;

            padding: 20px;

            border-radius: 8px;

            box-shadow: 0 0 10px rgba(0,0,0,0.1);
```

```
}  
td {  
    padding: 10px;  
}  
label {  
    color: #555;  
    font-weight: bold;  
}  
input[type="text"], input[type="email"], input[type="password"], select, textarea {  
    width: 100%;  
    padding: 8px;  
    border: 1px solid #ddd;  
    border-radius: 4px;  
    box-sizing: border-box;  
    font-size: 16px;  
}  
input[type="radio"], input[type="checkbox"] {  
    margin-right: 5px;  
}  
input[type="submit"] {  
    background-color: #4CAF50;  
    color: white;  
    padding: 10px 20px;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
    font-size: 18px;  
}  
input[type="submit"]:hover {
```

```
        background-color: #45a049;
    }

    .error {
        color: #ff0000;
        font-size: 14px;
    }
</style>
</head>
<body>
    <h1>Registration Form</h1>
    <form action="#" method="post">
        <table>
            <tr>
                <td><label for="fullname">Full Name:</label></td>
                <td><input type="text" id="fullname" name="fullname" required></td>
            </tr>
            <tr>
                <td><label for="email">Email:</label></td>
                <td><input type="email" id="email" name="email" required></td>
            </tr>
            <tr>
                <td><label for="password">Password:</label></td>
                <td><input type="password" id="password" name="password" required></td>
            </tr>
            <tr>
                <td><label for="confirm_password">Confirm Password:</label></td>
                <td><input type="password" id="confirm_password" name="confirm_password"
required></td>
            </tr>
        </table>
    </form>
</body>
```

```
<tr>

  <td><label>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>

    <input type="radio" id="other" name="gender" value="other" required>

    <label for="other">Other</label>

  </td>

</tr>

<tr>

  <td><label for="birthdate">Date of Birth:</label></td>

  <td><input type="date" id="birthdate" name="birthdate" required></td>

</tr>

<tr>

  <td><label for="country">Country:</label></td>

  <td>

    <select id="country" name="country" required>

      <option value="">Select a country</option>

      <option value="usa">India</option>

      <option value="uk">United Kingdom</option>

      <option value="canada">Canada</option>

      <option value="australia">Australia</option>

      <option value="other">Other</option>

    </select>

  </td>

</tr>

<tr>
```

```
<td><label for="interests">Interests:</label></td>

<td>

  <input type="checkbox" id="sports" name="interests[]" value="sports">

  <label for="sports">Sports</label>

  <input type="checkbox" id="music" name="interests[]" value="music">

  <label for="music">Music</label>

  <input type="checkbox" id="reading" name="interests[]" value="reading">

  <label for="reading">Reading</label>

  <input type="checkbox" id="travel" name="interests[]" value="travel">

  <label for="travel">Travel</label>

</td>

</tr>

<tr>

  <td><label for="bio">Bio:</label></td>

  <td><textarea id="bio" name="bio" rows="4"></textarea></td>

</tr>

<tr>

  <td colspan="2" style="text-align: center;">

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

  </td>

</tr>

</table>

</form>

</body>

</html>
```

Output:

Registration Form

Full Name:

Email:

Password:

Confirm Password:

Gender:

☐ Male ☐ Female ☐ Other

Date of Birth:

Country:

Select a country ▼

Interests:

☐ Sports ☐ Music ☐ Reading ☐ Travel

Bio:

Register

Program-5

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

Source code: newspaper.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>The Daily Chronicle</title>
  <style>
    body {
      font-family: 'Georgia', serif;
      line-height: 1.6;
      color: #333;
      max-width: 1200px;
      margin: 0 auto;
      padding: 20px;
      background-color: #f4f4f4;
    }
    header {
      background-color: #1a1a1a;
      color: #fff;
      padding: 20px;
      text-align: center;
    }
    header h1 {
      margin: 0;
      font-size: 2.5em;
```

```
}  
  
nav {  
    background-color: #333;  
    color: #fff;  
    padding: 10px;  
}  
  
nav ul {  
    list-style-type: none;  
    padding: 0;  
    margin: 0;  
    display: flex;  
    justify-content: center;  
}  
  
nav ul li {  
    margin: 0 10px;  
}  
  
nav ul li a {  
    color: #fff;  
    text-decoration: none;  
}  
  
main {  
    display: flex;  
    margin-top: 20px;  
}  
  
section {  
    flex: 2;  
    margin-right: 20px;  
}  
  
article {
```

```
background-color: #fff;

padding: 20px;

margin-bottom: 20px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

article h2 {

color: #1a1a1a;

font-size: 1.8em;

}

aside {

flex: 1;

background-color: #e6e6e6;

padding: 20px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

figure {

margin: 0;

text-align: center;

}

figure img {

max-width: 100%;

height: auto;

}

figcaption {

font-style: italic;

color: #666;

font-size: 0.9em;

}

table {
```

```
        width: 100%;

        border-collapse: collapse;

        margin-bottom: 20px;
    }

    th, td {

        border: 1px solid #ddd;

        padding: 10px;

        text-align: left;
    }

    th {

        background-color: #f2f2f2;
    }

    footer {

        background-color: #1a1a1a;

        color: #fff;

        text-align: center;

        padding: 10px;

        margin-top: 20px;
    }
</style>
</head>
<body>
    <header>
        <h1>The Daily Chronicle</h1>
    </header>
    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Politics</a></li>
```

```
<li><a href="#">Technology</a></li>

<li><a href="#">Sports</a></li>

<li><a href="#">Entertainment</a></li>

</ul>

</nav>

<main>

<section>

<article>

<h2>Breaking News: Major Technological Breakthrough</h2>

<p>Scientists have announced a groundbreaking discovery in the field of quantum
computing, potentially revolutionizing the tech industry.</p>

<figure>



<figcaption>A state-of-the-art quantum computer at the research
facility</figcaption>

</figure>

</article>

<article>

<h2>Local Sports Team Wins Championship</h2>

<p>In a thrilling match, our local team secured victory in the national championship,
bringing pride to our city.</p>

<table>

<tr>

<th>Team</th>

<th>Score</th>

</tr>

<tr>

<td>Local Heroes</td>
```

```
<td>3</td>

</tr>

<tr>

<td>Visiting Challengers</td>

<td>2</td>

</tr>

</table>

</article>

</section>

<aside>

<h3>Weather Update</h3>

<p>Expect sunny skies with a high of 75°F (24°C) today.</p>

<h3>Upcoming Events</h3>

<ul>

<li>City Festival - This Weekend</li>

<li>Tech Conference - Next Month</li>

<li>Charity Run - In Two Weeks</li>

</ul>

</aside>

</main>

<footer>

<p>&copy; 2024 The Daily Chronicle. All rights reserved.</p>

</footer>

</body>

</html>
```

Output:

The Daily Chronicle

[Home](#)[Politics](#)[Technology](#)[Sports](#)[Entertainment](#)

Breaking News: Major Technological Breakthrough

Scientists have announced a groundbreaking discovery in the field of quantum computing, potentially revolutionizing the tech industry.



A state-of-the-art quantum computer at the research facility

Local Sports Team Wins Championship

In a thrilling match, our local team secured victory in the national championship, bringing pride to our city.

Team	Score
Local Heroes	3
Visiting Challengers	2

Weather Update

Expect sunny skies with a high of 75°F (24°C) today.

Upcoming Events

- City Festival - This Weekend
- Tech Conference - Next Month
- Charity Run - In Two Weeks

© 2024 The Daily Chronicle. All rights reserved.

Program -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.

Source Code: calculator.html

(Using PHP)

```
<html>

<head>

<style>

table, td, th {

border: 1px solid black;

width: 35%;

text-align: center;

background-color: lightgray;

}

table { margin: auto; }

input,p { text-align:right; }

</style>

</head>

<body>

<form method="post" action="calci.php">

<table>

<caption><h2> SIMPLE CALCULATOR </h2></caption>

<tr>

<td>First Number:</td><td><input type="text" name="num1" /></td>

<td rowspan="2"><button type="submit" name="submit"

value="calculate">Calculate</td></tr>

<tr>
```

```
<td>Second Number:</td><td><input type="text" name="num2"/></td></tr></form>
```

```
<?php
```

```
if(isset($_POST['submit'])) // it checks if the input submit is filled
```

```
{
```

```
    $num1 = $_POST['num1'];
```

```
    $num2 = $_POST['num2'];
```

```
    if(is_numeric($num1) and is_numeric($num2) )
```

```
    {
```

```
        echo "<tr><td> Sum :</td><td><p>".($num1+$num2)."</p></td>";
```

```
        echo "<tr><td> Difference :</td><td><p> ".($num1-$num2)."</p></td>";
```

```
        echo "<tr><td> Product :</td><td><p>".($num1*$num2)."</p></td>";
```

```
        echo "<tr><td>Quoient :</td><td><p> ".($num1/$num2)."</p></td>";
```

```
        echo "<tr><td>Remainder :</td><td><p> ".($num1%$num2)."</p></td>";
```

```
        echo "<tr><td>Square1:</td><td><p> ".($num1*$num1)."</p></td>";
```

```
        echo "<tr><td>Power:</td><td><p> ".($num2*$num2)."</p></td>";
```

```
        echo "<tr><td>Squareroot1:</td><td><p> ".(sqrt($num1))."</p></td>";
```

```
        echo "</table>";
```

```
    }
```

```
    else
```

```
    {
```

```
        echo"<script> alert(' ENTER VALID NUMBER');</script>";
```

```
    }
```

```
}
```

```
?>
```

```
</body>
```

```
</html>
```

(Using JavaScript)

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
table, td, th
```

```
{
```

```
border: 1px solid black;
```

```
width: 33%;
```

```
text-align: center;
```

```
background-color: DarkGray;
```

```
border-collapse: collapse;
```

```
}
```

```
table {margin: auto; }
```

```
input {text-align:right; }
```

```
</style>
```

```
<script type="text/javascript">
```

```
function calc(clicked_id)
```

```
{
```

```
var val1 = parseFloat(document.getElementById("value1").value);
```

```
var val2 = parseFloat(document.getElementById("value2").value);
```

```
if (isNaN(val1)||isNaN(val2))
```

```
    alert("ENTER VALID NUMBER");
```

```
else if(clicked_id=="add")
    document.getElementById("answer").value=val1+val2;
else if(clicked_id=="sub")
    document.getElementById("answer").value=val1-val2;
else if(clicked_id=="mul")
    document.getElementById("answer").value=val1*val2;
else if(clicked_id=="div")
    document.getElementById("answer").value=val1/val2;
else if(clicked_id=="mod")
    document.getElementById("answer").value=val1%val2;
else if(clicked_id=="square")
    document.getElementById("answer").value=val1*val1;
else if(clicked_id=="power")
    document.getElementById("answer").value=val2*val2;
else if(clicked_id=="root")
    document.getElementById("answer").value=(Math.sqrt(val1));
}

function cls()
{
    value1.value="0";
    value2.value="0";
    answer.value="";
}

</script>

</head>

<body>

<table>
```

```

<tr>

<th colspan="4"> SIMPLE CALCULATOR </th>

</tr>

<tr>

<td>value1</td><td><input type="text" id="value1" value="0"/></td>

<td>value2</td><td><input type="text" id= "value2" value="0"/></td>

</tr>

<tr>

<td><input type="button" value="Addition" id = "add" onclick="calc(this.id)"/></td>

<td><input type="button" value="Subtraction" id = "sub" onclick="calc(this.id)"/></td>

<td><input type="button" value="Multiplication" id = "mul" onclick="calc(this.id)"/></td>

<td><input type="button" value="Division" id ="div" onclick="calc(this.id)"/></td>

</tr><tr>

<td><input type="button" value="Quoient" id ="mod" onclick="calc(this.id)"/></td>

<td><input type="button" value="Square" id ="square" onclick="calc(this.id)"/></td>

<td><input type="button" value="Power" id ="power" onclick="calc(this.id)"/></td>

<td><input type="button" value="Square root" id ="root" onclick="calc(this.id)"/></td>

</tr>

<tr>

<td>Answer:</td><td> <input type= "text" id="answer" value="" disabled/></td>

<td colspan="2"><input type= "button" value="CLEAR ALL" onclick="cls()"/><td>

</tr>

</table>

</body>

</html>

```

Output:

SIMPLE CALCULATOR			
value1	0	value2	0
Addition	Subtraction	Multiplication	Division
Quoient	Square	Power	Square root
Answer:		CLEAR ALL	

Program-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**

source Code: pgm7.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>JSON/CSV Converter and Hash Generator</title>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.1.1/crypto-js.min.js"></script>

  <style>

    body {

      font-family: Arial, sans-serif;

      line-height: 1.6;

      margin: 0;

      padding: 20px;

      background-color: #f4f4f4;

    }

    .container {

      max-width: 800px;

      margin: auto;

      background: white;

      padding: 20px;
```

```
border-radius: 5px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

h1 {

    color: #333;

}

textarea {

    width: 100%;

    height: 100px;

    margin-bottom: 10px;

}

button {

    background-color: #4CAF50;

    color: white;

    padding: 10px 15px;

    border: none;

    border-radius: 4px;

    cursor: pointer;

    margin-right: 10px;

}

button:hover {

    background-color: #45a049;

}

#result {

    margin-top: 20px;

    padding: 10px;

    background-color: #e7e7e7;
```

```
        border-radius: 4px;

    }

</style>

</head>

<body>

    <div class="container">

        <h1>JSON/CSV Converter and Hash Generator</h1>

        <h2>a) Convert JSON to JavaScript Object</h2>

        <textarea id="jsonInput" placeholder="Enter JSON here"></textarea>

        <button onclick="convertJsonToObject()">Convert to Object</button>

        <h2>b) Convert JSON to Date</h2>

        <textarea id="jsonDateInput" placeholder="Enter JSON date string (e.g., {"date": "2023-05-15T12:00:00Z"})"></textarea>

        <button onclick="convertJsonToDate()">Convert to Date</button>

        <h2>c) Convert JSON to CSV and CSV to JSON</h2>

        <textarea id="dataInput" placeholder="Enter JSON or CSV here"></textarea>

        <button onclick="convertJsonToCsv()">JSON to CSV</button>

        <button onclick="convertCsvToJson()">CSV to JSON</button>

        <h2>d) Create Hash from String</h2>

        <textarea id="hashInput" placeholder="Enter string to hash"></textarea>

        <button onclick="createHash()">Generate Hash</button>

        <div id="result"></div>
```

```
</div>
```

```
<script>
```

```
function convertJsonToObject() {  
    try {  
        const jsonInput = document.getElementById('jsonInput').value;  
        const jsObject = JSON.parse(jsonInput);  
        document.getElementById('result').innerText = 'Converted Object: ' +  
JSON.stringify(jsObject, null, 2);  
    } catch (error) {  
        document.getElementById('result').innerText = 'Error: ' + error.message;  
    }  
}  
  
function convertJsonToDate() {  
    try {  
        const jsonInput = document.getElementById('jsonDateInput').value;  
        const jsObject = JSON.parse(jsonInput);  
        const date = new Date(jsObject.date);  
        document.getElementById('result').innerText = 'Converted Date: ' + date.toString();  
    } catch (error) {  
        document.getElementById('result').innerText = 'Error: ' + error.message;  
    }  
}  
  
function convertJsonToCsv() {  
    try {
```

```
const jsonInput = document.getElementById('dataInput').value;

const jsObject = JSON.parse(jsonInput);

const headers = Object.keys(jsObject[0]);

const csvRows = [

    headers.join(','),

    ...jsObject.map(row => headers.map(fieldName =>
JSON.stringify(row[fieldName])).join(','))

];

const csvString = csvRows.join('\n');

document.getElementById('result').innerText = 'Converted CSV:\n' + csvString;

} catch (error) {

    document.getElementById('result').innerText = 'Error: ' + error.message;

}

}

function convertCsvToJson() {

    try {

        const csvInput = document.getElementById('dataInput').value;

        const lines = csvInput.split('\n');

        const headers = lines[0].split(',');

        const jsonArray = lines.slice(1).map(line => {

            const values = line.split(',');

            return headers.reduce((obj, header, index) => {

                obj[header] = values[index];

                return obj;

            }, {});

        });

    }

};
```

```
        document.getElementById('result').innerText = 'Converted JSON:\n' +
JSON.stringify(jsonArray, null, 2);

    } catch (error) {

        document.getElementById('result').innerText = 'Error: ' + error.message;

    }

}

function createHash() {

    try {

        const input = document.getElementById('hashInput').value;

        const hash = CryptoJS.SHA256(input);

        document.getElementById('result').innerText = 'Generated Hash (SHA-256): ' + hash;

    } catch (error) {

        document.getElementById('result').innerText = 'Error: ' + error.message;

    }

}

</script>
</body>
</html>
```

Output:

a) Convert JSON to JavaScript Object: Input should be given in the format of `{"name":"John","age":30,"city":"New York"}` When we click the "Convert to Object" button, the `convertJsonToObject()` function will parse the JSON string and display the resulting JavaScript object in the `#result` div. The `JSON.stringify()` method is used to convert the JavaScript object to a string, with indentation and spacing for readability. The resulting string is then displayed in the `#result` div. Note that the output will be a string representation of the JavaScript object, with property names and values enclosed in double quotes.

b) Convert JSON to Date: input should be given in the format of `{"date": "2023-05-15T12:00:00Z"}`

a) Convert JSON to JavaScript Object

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

Convert to Object

b) Convert JSON to Date

```
{"date": "2023-05-15T12:00:00Z"}
```

Convert to Date

c) Convert JSON to CSV and CSV to JSON

Enter JSON or CSV here

JSON to CSV

CSV to JSON

d) Create Hash from String

Enter string to hash

Generate Hash

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

Converted Date: Mon May 15 2023 17:30:00 GMT+0530 (India Standard Time)

c) Convert JSON to CSV and CSV to JSON: Input should be given in the format of `[{"name": "John", "age": 30, "city": "New York"}, {"name": "Jane", "age": 25, "city": "Los Angeles"}]` for convention JSON to CSV and From converting CSV to JSON the input should be given as :

name,age,gender

john,20,male

jane,30,female

bob,25,male

d) Create Hash from String: Input should be given in the format of **"Hello, World!"**

c) Convert JSON to CSV and CSV to JSON

```
[{"name": "John", "age": 30, "city": "New York"}, {"name": "Jane", "age": 25, "city": "Los Angeles"}]
```

JSON to CSV

CSV to JSON

d) Create Hash from String

"Hello, World!"

Generate Hash

Converted CSV:

name,age,city

"John",30,"New York"

"Jane",25,"Los Angeles"

Generated Hash (SHA-256): cc82ebbcf8b60a5821d1c51c72cd79380ecea47de343ccb3b158938a2b3bf764

c) Convert JSON to CSV and CSV to JSON

```
name,age,gender  
john,20,male  
jane,30,female  
bob,25,male
```

[JSON to CSV](#)[CSV to JSON](#)

d) Create Hash from String

Enter string to hash

[Generate Hash](#)

Converted JSON:

```
[  
{  
  "name": "john",  
  "age": "20",  
  "gender": "male"  
},  
{  
  "name": "jane",  
  "age": "30",  
  "gender": "female"  
},  
{  
  "name": "bob",  
  "age": "25",  
  "gender": "male"  
},  
{  
  "name": ""  
}  
]
```

Program - 8A

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

Source Code: pgm8a.php

```
<!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>

    <style>

        body {

            font-family: Arial, sans-serif;

            line-height: 1.6;

            margin: 0;

            padding: 20px;

            background-color: #f4f4f4;

        }

        .container {

            max-width: 600px;

            margin: auto;

            background: white;

            padding: 20px;

            border-radius: 5px;

            box-shadow: 0 0 10px rgba(0,0,0,0.1);

        }

        h1 {
```

```
        color: #333;

        text-align: center;
    }

    .counter {

        font-size: 24px;

        text-align: center;

        margin-top: 20px;

    }

</style>

</head>

<body>

    <div class="container">

        <h1>Welcome to Our Website</h1>

        <div class="counter">

            <?php

                $counterFile = 'count.txt';

                // Read the current count

                if (file_exists($counterFile)) {

                    $count = (int)file_get_contents($counterFile);

                } else {

                    $count = 0;

                }

                // Increment the count

                $count++;

                // Save the new count

                file_put_contents($counterFile, $count);

                // Display the count
```

```
echo "<h2>Visitor Count</h2>";

echo "<p>You are visitor number: $count</p>";

$name="counter.txt";

$file = fopen($name,"r");

$hits= fscanf($file,"%d");

fclose($file);

$hits[0]++;

$file = fopen($name,"w");

fprintf($file,"%d",$hits[0]);

fclose($file);

print "Total number of views: ".$hits[0];

?>

</div>

</div>

</body>

</html>
```

Output:

Welcome to Our Website

Visitor Count

You are visitor number: 22

Total number of views: 22

Program - 8B

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

Source Code: pgm8bb.php

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Student Record Sorter</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            line-height: 1.6;

            margin: 0;

            padding: 20px;

            background-color: #f4f4f4;

        }

        .container {

            max-width: 800px;

            margin: auto;

            background: white;

            padding: 20px;

            border-radius: 5px;

            box-shadow: 0 0 10px rgba(0,0,0,0.1);

        }

        h1 {
```

```
        color: #333;

        text-align: center;
    }

    table {

        width: 100%;

        border-collapse: collapse;

        margin-top: 20px;
    }

    th, td {

        padding: 10px;

        border: 1px solid #ddd;

        text-align: left;
    }

    th {

        background-color: #f2f2f2;
    }

</style>

</head>

<body>

    <div class="container">

        <h1>Student Records</h1>

        <?php

            // Database connection details

            $host = 'localhost';

            $dbname = 'nkg';

            $username = 'root';

            $password = '';
```

```
try {  
  
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);  
  
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
  
    // Fetch student records  
  
    $stmt = $pdo->query("SELECT * FROM students");  
  
    $students = $stmt->fetchAll(PDO::FETCH_ASSOC);  
  
    // Selection sort function  
  
    function selectionSort(&$arr, $n) {  
        for ($i = 0; $i < $n - 1; $i++) {  
            $min_idx = $i;  
            for ($j = $i + 1; $j < $n; $j++) {  
                if ($arr[$j]['gpa'] > $arr[$min_idx]['gpa']) {  
                    $min_idx = $j;  
                }  
            }  
            if ($min_idx != $i) {  
                $temp = $arr[$i];  
                $arr[$i] = $arr[$min_idx];  
                $arr[$min_idx] = $temp;  
            }  
        }  
    }  
  
    // Sort students by GPA
```

```
selectionSort($students, count($students));

// Display sorted student records

echo "<table>";

echo "<tr><th>ID</th><th>Name</th><th>GPA</th></tr>";

foreach ($students as $student) {

    echo "<tr>";

    echo "<td>" . htmlspecialchars($student['id']) . "</td>";

    echo "<td>" . htmlspecialchars($student['name']) . "</td>";

    echo "<td>" . htmlspecialchars($student['gpa']) . "</td>";

    echo "</tr>";

}

echo "</table>";

} catch(PDOException $e) {

    echo "Connection failed: " . $e->getMessage();

}

?>

</div>

</body>

</html>
```

Output:**commands in shell:****open shell in xampp****>>mysql -u root****>>show databases;****>> create database weblab;****>>show databases;**

>>use weblab; # use the craeted database by the student

>> create table student;

>> create table student (usn varchar(30), name varchar(30), marks varchar(10));

>>desc student; #description of table

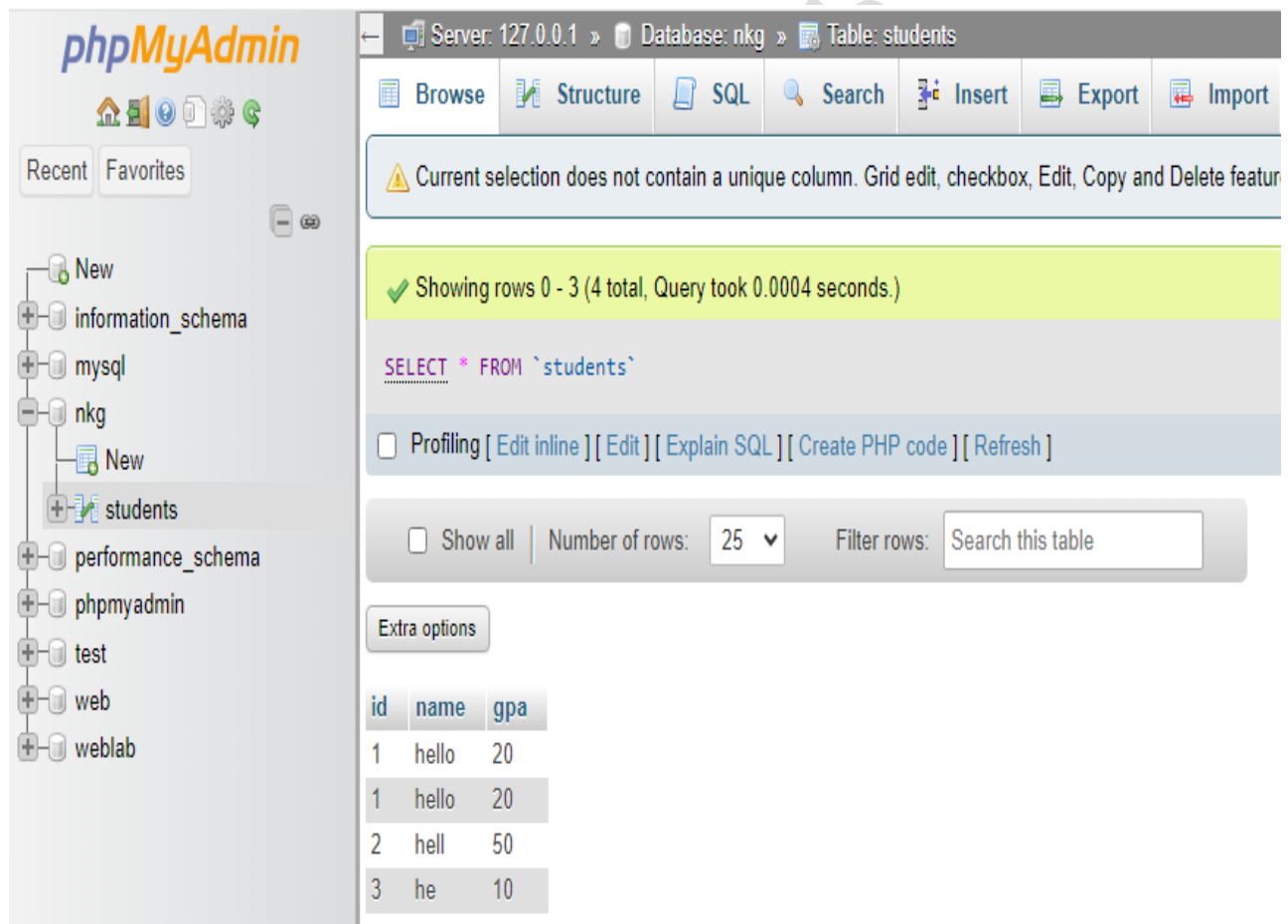
>> insert into student values ("1kg22cs20", "lmn", "5");

>> insert into student values ("1kg22cs40", "gfhe", "4");

>>select * from student;

OR

...localhost/dashboard -> phpMyAdmin -> craete a database name as nkg -> create a table as students -> insert the column names.



The screenshot displays the phpMyAdmin web interface. On the left, the database structure tree shows a database named 'nkg' containing a table named 'students'. The main panel shows the 'Table: students' view. A message at the top states: 'Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are disabled.' Below this, a green status bar indicates 'Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)'. The SQL query entered is 'SELECT * FROM `students`'. Below the query, there are links for 'Profiling', 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh'. A 'Show all' checkbox is present, along with a 'Number of rows' dropdown set to '25' and a 'Filter rows' search box. At the bottom, an 'Extra options' button is visible. The table data is displayed as follows:

id	name	gpa
1	hello	20
1	hello	20
2	hell	50
3	he	10

Sorted in greater to Lower range.

Student Records

ID	Name	GPA
2	hell	50
1	hello	20
1	hello	20
3	he	10


```
h1, h2 {
    color: #333;
}

.box {
    width: 100px;
    height: 100px;
    background-color: #3498db;
    margin: 20px 0;
}

button {
    padding: 10px 15px;
    background-color: #2ecc71;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    margin-right: 10px;
}

button:hover {
    background-color: #27ae60;
}

</style>
</head>
<body>
    <div class="container">
        <h1>jQuery Demonstration</h1>

        <h2>a. Append Content</h2>

        <p id="existingParagraph">This is an existing paragraph. </p>
```

```
<ul id="existingList">
    <li>Existing item 1</li>
    <li>Existing item 2</li>
</ul>

<input type="text" id="appendInput" placeholder="Enter text to append">
<button id="appendButton">Append Content</button>
```

```
<h2>b. Animate Element</h2>
```

```
<div id="animateBox" class="box"></div>
<button id="animateButton">Animate Box</button>
```

```
<h2>c. Change Color of Animated Div</h2>
```

```
<div id="colorBox" class="box"></div>
<button id="colorAnimateButton">Animate and Change Color</button>
</div>
```

```
<script>
```

```
$(document).ready(function() {
    // a. Append content
    $("#appendButton").click(function() {
        var userInput = $("#appendInput").val();
        $("#existingList").append("<li>" + userInput + "</li>").append("<h4>New data
        Appended</h4>");
    });
```

```
    // b. Animate element
```

```
    $("#animateButton").click(function() {
        $("#animateBox").animate({
            width: "200px",
```

```
        height: "200px",
        opacity: 0.5
    }, 1000);
});

// c. Animate and change color
$("#colorAnimateButton").click(function() {
    $("#colorBox").animate({
        width: "200px",
        height: "200px"
    }, {
        duration: 1000,
        step: function(now, fx) {
            if (fx.prop === "width") {
                $(this).css("background-color", rgb(${Math.round(now)}, 52, 219));
            }
        }
    });
});
</script>
</body>
</html>
```

Output:

jQuery Demonstration

a. Append Content

This is an existing paragraph.

- Existing item 1
- Existing item 2
- web technology

New data Appended

Append Content

b. Animate Element

Animate Box

c. Change Color of Animated Div

Animate and Change Color

Program-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.**

Source Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ajax Demo Program</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      line-height: 1.6;
      margin: 0;
      padding: 20px;
      background-color: #f4f4f4;
    }
    .container {
      max-width: 800px;
      margin: auto;
      background: white;
      padding: 20px;
      border-radius: 5px;
```

```
        box-shadow: 0 0 10px rgba(0,0,0,0.1);
    }
    h1 {
        color: #333;
    }
    h2 {
        color: #666;
    }
    button {
        background-color: #4CAF50;
        border: none;
        color: white;
        padding: 10px 20px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 16px;
        margin: 4px 2px;
        cursor: pointer;
        border-radius: 4px;
    }
    pre {
        background-color: #f8f8f8;
        border: 1px solid #ddd;
        border-radius: 4px;
        padding: 10px;
        white-space: pre-wrap;
        word-wrap: break-word;
    }
```

```
</style>

</head>

<body>

  <div class="container">

    <h1>Ajax Demo Program</h1>

    <h2>a. Ajax-like operation without jQuery</h2>

    <button onclick="operationWithoutjQuery()">Perform Operation (without
jQuery)</button>

    <pre id="result-a"></pre>

    <h2>b. Ajax-like operation with jQuery</h2>

    <button onclick="operationWithjQuery()">Perform Operation (with jQuery)</button>

    <pre id="result-b"></pre>

    <h2>c. jQuery-like getJSON() method</h2>

    <button onclick="getJSONOperation()">Get JSON</button>

    <pre id="result-c"></pre>

    <h2>d. jQuery parseJSON() method</h2>

    <button onclick="parseJSONExample()">Parse JSON</button>

    <pre id="result-d"></pre>
  </div>

  <script>

    // Simulated data

    const simulatedData = {

      text: "This is a sample text from a simulated server response.",

      json: {
```

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

// a. Ajax-like operation without jQuery

```
function operationWithoutjQuery() {
    setTimeout(function() {
        document.getElementById("result-a").textContent = simulatedData.text;
    }, 500);
}
```

// b. Ajax-like operation with jQuery

```
function operationWithjQuery() {
    $.Deferred(function(deferred) {
        setTimeout(function() {
            deferred.resolve(simulatedData.text);
        }, 500);
    }).done(function(result) {
        $("#result-b").text(result);
    });
}
```

// c. jQuery-like getJSON() method

```
function getJSONOperation() {
    $.Deferred(function(deferred) {
        setTimeout(function() {
            deferred.resolve(simulatedData.json);
        }, 500);
    });
}
```

```
        }, 500);  
    }).done(function(result) {  
        $("#result-c").text(JSON.stringify(result, null, 2));  
    });  
}  
  
// d. jQuery parseJSON() method  
function parseJSONExample() {  
    var jsonString = JSON.stringify(simulatedData.json);  
    var Json Object = $.parseJSON(jsonString);  
    $("#result-d").text(JSON.stringify(Json Object, null, 2));  
}  
</script>  
</body>  
</html>
```

Output:

Ajax Demo Program

a. Ajax-like operation without jQuery

Perform Operation (without jQuery)

This is a sample text from a simulated server response.

b. Ajax-like operation with jQuery

Perform Operation (with jQuery)

This is a sample text from a simulated server response.

c. jQuery-like getJSON() method

Get JSON

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

d. jQuery parseJSON() method

Parse JSON

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

Viva Questions and Answers:

1. What are tags?

Content is placed in between HTML tags in order to properly format it. It makes use of the less than symbol (<) and the greater than symbol (>).

2. Do all HTML tags come in pair?

No, there are single HTML tags that does not need a closing tag. Examples are the tag and
 tags.

3. What is the difference between HTML elements and tags?

HTML elements communicate to the browser how to render text. When surrounded by angular brackets < > they form HTML tags. For the most part, tags come in pairs and surround text.

4. What is “Semantic HTML?”

- Semantic HTML is a coding style where the tags embody what the text is meant to convey.
- In Semantic HTML, tags like for bold, and <i> </i> for italic should not be used, reason being they just represent formatting, and provide no indication of meaning or structure.
- The semantically correct thing to do is use and .
- These tags will have the same bold and italic effects, while demonstrating meaning and structure (emphasis in this case).

5. How do you insert a copyright symbol on a browser page?

To insert the copyright symbol, you need to type © or & #169; in an HTML file.

6. Is there any way to keep list elements straight in an html file?

- By using indents, you can keep the list elements straight.
- If you indent each sub nested list in further than the parent list that contains it, you can at a glance determine the various lists and the elements that it contains.

7. Does a hyperlink apply to text only?

No, hyperlinks can be used on text as well as images. That means you can convert an image into a link that will allow user to link to another page when clicked. Just surround the image within the ... tag combinations.

8. What are style sheets?

Style sheets enable you to build consistent, transportable, and well-defined style templates. These templates can be linked to several different web pages, making it easy to maintain and change the look and feel of all the web pages within a site.

9. What bullet types are available?

With ordered lists, you can select to use a number of different list types including alphabetical and Roman numerals. The type attribute for unordered lists can be set to disc, square, or circle.

10. How do you make comments without text being picked up by the browser?

Comments are used to explain and clarify code or to prevent code from being recognized by the browser. Comments start with “*<!--” and end with ”--> “.

<!-- Insert comment here. -->

11. What is span in HTML?

The SPAN having notation as

< SPAN > < /SPAN >

is used for highlighting text of any color desired, for adding colored text, for adding background image to text. SPAN doesn't cause a line break. It delimits text and it allows styles to be applied to a 'elemental' region, or for the 'elemental' region to be identified, without causing a break in the text flow.

12. Is CSS case sensitive?

- Cascading Style Sheets (CSS) is not case sensitive.
- However, font families, URLs to images, and other direct references with the style sheet may be.
- The trick is that if you write a document using an XML declaration and an XHTML doctype, then the CSS class names will be case sensitive for some browsers.

13. What are the limitations of CSS ?

Limitations are:

- Ascending by selectors is not possible
- Limitations of vertical control

- No expressions
- No column declaration
- Pseudo-class not controlled by dynamic behavior
- Rules, styles, targeting specific text not possible

14. What is the difference between an ID selector and CLASS?

An ID selector identifies and sets style to only one occurrence of an element, while CLASS can be attached to any number of element

15. What is Contextual Selector?

- Contextual selector addresses specific occurrence of an element.
- It is a string of individual selectors separated by white space (search pattern), where only the last element in the pattern is addressed providing it matches the specified context

16. What are Pseudo Classes?

Pseudo classes allow you to identify HTML elements on characteristics (as opposed to their name or attributes). The classes are specified using a colon to separate the element name and pseudo class. A good example is the :link and :visited pseudo classes for the HTML A element. Another good example is first_child, which finds an element's first child element.

What does z-index do?

The z-index property specifies the stack order of an element. An element with greater stack order is always in front of an element with a lower stack order.

Here is the Example, where using z-index property you can display in front of image.

17. What is parent-child selector?

Parent-child selector is a selector representing the direct descendent of a parent element. Parent-child selectors are created by listing two or more tilde (~) separated selectors.

BODY ~ P {background: red; color: white}

The P element will be declared the specified style only if it directly descends from the BODY element:

18. Enumerate the differences between Java and JavaScript?

Java is a complete programming language. In contrast, JavaScript is a coded program that can be introduced to HTML pages. These two languages are not at all inter-dependent and are designed for the different intent. Java is an object-oriented programming (OOPS) or structured programming language like C++ or C whereas JavaScript is a client-side scripting language.

19. What is the use of isNaN function?

isNaN function returns true if the argument is not a number otherwise it is false.

20. What is negative infinity?

Negative Infinity is a number in JavaScript which can be derived by dividing negative number by zero.

21. What are global variables? How are these variable declared and what are the problems associated with using them?

Global variables are those that are available throughout the length of the code, that is, these have no scope. The var keyword is used to declare a local variable or object. If the var keyword is omitted, a global variable is declared.

22. What is a prompt box?

A prompt box is a box which allows the user to enter input by providing a text box. Label and box will be provided to enter the text or number.

23. What is === operator?

=== is called as strict equality operator which returns true when the two operands are having the same value without any type conversion.

24. What is called Variable typing in Javascript?

Variable typing is used to assign a number to a variable and the same variable can be assigned to a string.

25. What do mean by NULL in Javascript?

The NULL value is used to represent no value or no object. It implies no object or null string, no valid boolean value, no number and no array object.

26. What is the difference between an alert box and a confirmation box?

An alert box displays only one button which is the OK button.

But a Confirmation box displays two buttons namely OK and cancel.

27. Explain what is pop() method in JavaScript?

The pop() method is similar as the shift() method but the difference is that the Shift method works at the start of the array. Also the pop() method take the last element off of the given array and returns it. The array on which is called is then altered.

What are the two basic groups of datatypes in JavaScript?

They are as –

- Primitive
- Reference types

28. What is the use of type of operator?

'Typeof' is an operator which is used to return a string description of the type of a variable.

30. Explain window.onload and onDocumentReady?

The onload function is not run until all the information on the page is loaded. This leads to a substantial delay before any code is executed. onDocumentReady loads the code just after the DOM is loaded. This allows early manipulation of the code.

31. Define event bubbling?

JavaScript allows DOM elements to be nested inside each other. In such a case, if the handler of the child is clicked, the handler of parent will also work as if it were clicked too.

32. How are event handlers utilized in JavaScript?

Events are the actions that result from activities, such as clicking a link or filling a form, by the user. An event handler is required to manage proper execution of all these events. Event handlers are an extra attribute of the object. This attribute includes event's name and the action taken if the event takes place.

33. What is the correct and the most two common way to start and finish a PHP block of code?

The two most common ways to start and finish a PHP script are:

`<?php [ --- PHP code---- ] ?>` and `<? [--- PHP code ---] ?>`

34. How can we display the output directly to the browser?

To be able to display the output directly to the browser, we have to use the special tags `<?=` and `?>`.

35. How can PHP and HTML interact?

It is possible to generate HTML through PHP scripts, and it is possible to pass pieces of information from HTML to PHP.

36. What type of operation is needed when passing values through a form or an URL?

If we would like to pass values through a form or an URL, then we need to encode and to decode them using `htmlspecialchars()` and `urlencode()`.

37. How can PHP and Javascript interact?

PHP and Javascript cannot directly interact since PHP is a server side language and Javascript is a client-side language. However, we can exchange variables since PHP can generate Javascript code to be executed by the browser and it is possible to pass specific variables back to PHP via the URL

38. What is JSON?

JSON stands for JavaScript Object Notation but it does not use JavaScript to perform operations, it just uses the JavaScript syntax. It is a light-weighted and language-independent format mainly used to transmit the data between the server and the web app.

39. Explain the structure and format of JSON.

The JSON format follows the structure of the JavaScript object. It contains the key-value pairs in the form of strings that can be transmitted from server to web app and vice-versa. The values stored in JSON can be retrieved in the same way as the values of JavaScript object are retrieved.

```
{  
  "key1": "value1",  
  "key2": "value2",  
  "key3": "value3"
```

```
}
```

40. Who created JSON?

Douglas Crockford was the first man who introduced and created the JSON format first in 2000.

41. What is the extension for the JSON files?

The JSON files can be created and stored using the .json extension. You can add the data in the form of JSON format using the key-value pairs in the form of strings.

42. What is the reason for using JSON?

JSON is a light weighted format that is used to transfer the data on the network. The main reason for using JSON is that it provides a very simple and easy way to communicate between the server and the web app by data transmission. It is used as an alternative for the XML format.

43. What are the features of JSON?

There are different features of JSON as listed below:

- It is a light weight format that helps to optimize the performance of the web app.
- It is independent of different programming languages.
- It has a vast support from different programming languages.
- The JSON format is very easy to understand.
- It can represent the complex data structures.

44. List different data types supported by JSON.

JSON has a support for the following data-types:

- Number
- Boolean
- String
- Array
- Object
- null

45. How to create a JSON array?

A JSON array is just like the arrays in the other programming languages that contains a

collection of multiple items of different data-types. It can contain value of any data-type supported by JSON.

```
["value1", 12, true, null]
```

46. Is it possible to create function in JSON?

No, as we know JSON is just a data format and it is used to transfer data on the network. Therefore, we can not write any kind of executable code inside it.

47. How to retrieve the data stored in JSON object?

You can retrieve or access the values stored in an JSON object in the same way as we access them in JavaScript. The below methods can be used to access the values of JSON object.

- Using dot notation
- Using square brackets

48. Explain the process of accessing JSON object values using dot notation.

You can follow the below syntax to retrieve values of JSON object using dot notation.

```
const JSONObj = {  
  "name": "GFG",  
  "est": "2009",  
  "workForce": "200+"  
}  
  
console.log(JSONObj.name)  
console.log(JSONObj.est)  
console.log(JSONObj.workForce)
```

49. Explain the process of accessing JSON object values using square brackets.

You can follow the below syntax to retrieve values of JSON object using square brackets.

```
const JSONObj = '{  
  "name": "GFG",  
  "est": "2009",  
  "workForce": "200+"
```

```
}'  
  
console.log(JSONObj["name"])  
  
console.log(JSONObj["est"])  
  
console.log(JSONObj["workForce"])
```

13. Explain JSONP and how it is different from JSON?

JSONP stands for JSON with Padding. It is a method for sending the JSON data with padding to avoid cross domain restrictions. JSONP is different from JSON, because it allows the cross-origin requests.

50. List different browsers that support JSON.

JSON is supported by all the modern browsers. As we know, JSON follows the JavaScript syntax, hence every browser has built-in support for it. The below browsers support JSON:

- Google Chrome
- Mozilla Firefox
- Safari
- Microsoft Edge
- Opera, etc.

51. Explain the use of JSON.parse() method in JSON.

The JSON.parse() method is used to parse the JSON data into JavaScript and converts the JSON string into a JavaScript object.

```
JSON.parse(JSONString, function)
```

52. What is the JSON.stringify() method used for?

The JSON.stringify() method is used to convert the JavaScript object into a JSON string format.

```
JSON.stringify(JSONString, function, space)
```

53. Explain JSON schema.

JSON schema is a content specification language defined to validate the structure and the format of JSON data or string. It defines the basic structure, format and important constraints of JSON data.

54. How to write JSON in an HTML document?

You can write or include the JSON data inside and HTML document using a <script> tag with type attribute by passing a value as application/json to it.

```
<script type="application/json">
```

```
// JSON data
```

```
</script>
```

55. What are the methods used to encode and decode JSON in PHP?

We can use the below methods to encode and decode JSON in PHP:

`json_encode()`: This method is used to encode JSON in PHP. It takes the data in PHP array format and converts it into JSON data.

`json_decode()`: It is used to decode JSON in PHP. It takes a JSON string as input and returns corresponding PHP array.

56. What is the MIME type for JSON?

The MIME type for JSON is `application/json`. You can pass this type as a value to the `type` attribute inside the script tag to include JSON into your HTML document.

57. What are the benefits of using JSON?

The below list contains some benefits of using JSON:

- It is easy to implement and understand.
- It can be used with any language due to its language independency.
- It's a very lightweight data interchange format on the network.
- It's an alternative of XML.

58. Explain disadvantages of using JSON.

There are some disadvantages of using JSON:

- It does not have any Document Type Definition.
- It is not good for large scale data products.
- There's no type definition of it.
- It is not safe to use with untrusted services and browsers as it does not have a solid security mechanism.
- It can output a complex data in case of large data set which may be difficult to understand.

59. What are the applications of JSON?

JSON can be used in different situations:

- It serializes and transmits the structured data over a network connection.
- There are multiple web services and APIs available on the internet that returns the data in JSON format when the data is fetched from them using JavaScript or any other language.
- It transfers the data between the servers and web applications.

60. List some of the popular libraries for working with JSON in JavaScript.

Below are some of the popular libraries for working with JSON in JavaScript:

- lodash
- underscore.js
- jsonwebtoken, etc.

61. How to handle JSON parsing errors in JavaScript?

The JSON parsing errors can be handled using the try/catch block with the parsing statements in JavaScript.

62. Is it possible to create duplicate keys in an JSON object?

No, JSON does not allow to create duplicate keys in an object. If there's a duplicate key created in some scenario, then the last occurrence of that key will replace the previous occurrence.

63. Explain JSON web token and its implementation.

JSON web token or JWT is a self-contained and compact way to transmit the data securely between server and web app in JSON object format. It consist of three parts as the header, the payload, and the signature. All the parts of JWT are encode in base64 format which are concatenated to form a JWT.

64. How to transmit JSON data securely over the internet?

The JSON data can be securely transmitted over the internet using the HTTPS (Hyper Text Transfer Protocol Secure) protocol. It encrypts the data during its transmission over the internet.

65. What is the purpose of toJSON() method in JSON?

The toJSON() method in JSON is used to convert the date returned by the Date constructor in the string format.

dateObj.toJSON();

66. What are JSON formatter and Validator?

The JSON formatter and Validators are the tools used to format and validate the JSON data. These tools convert the JSON data into a human-readable format by ensuring its readability, correctness, and adherence to JSON syntax.

67. What are the alternatives of JSON?

There are some alternatives available for JSON as listed below:

- XML (Extensible Markup Language)
- Protocol Buffers (protobuf)
- YAML
- MessagePack
- BSON (Binary JSON)
- CBOR (Concise Binary Object Representation)