

Experiment-1

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.)

Laboratory Components

1. HTML Basics "Myfirstwebpage.html" - Fundamental HTML tags and structures
2. HTML Tables "Table.html" - Create a class timetable with advanced table features
3. External CSS Styling "style.css" - Demonstrate various CSS selectors and styles
4. HTML Forms with CSS "registration.html" - Input elements styled with CSS
5. Semantic HTML and CSS "newspaper.html" - Use of semantic elements with styling
6. JavaScript Calculator Simple calculator with multiple operations
7. JavaScript and JSON JSON manipulation, conversion, and hashing
8. PHP and Databases Visitor counter and database sorting with PHP
9. jQuery DOM Manipulation Content appending, animation, and color changing
10. Ajax Operations Ajax requests with and without jQuery, JSON handling

Explanation

Step 1: Document Structure

<!DOCTYPE html>: This declaration tells the browser that this is an HTML5 document.

<html lang="en">: The root element of the HTML page. The "lang" attribute specifies that

the language is English.

<head>: This section contains metadata about the document.

<body>: This section contains the visible content of the webpage.

Step 2: The Head Section

<meta charset="UTF-8">: Specifies the character encoding for the document (UTF-8 supports many languages).

<meta name="viewport" ...>: This tag helps with responsive design, making the page display properly on different devices.

<title>: Sets the title of the webpage, which appears in the browser tab.

<link rel="stylesheet" href="style.css">: Links to an external CSS file for styling the page

Step 3: The Body Content

<marquee>: Creates scrolling text (Note: This tag is outdated and not recommended for modern websites).

Headings: <h1> to <h6> tags represent six levels of section headings, with <h1> being the highest (most important) and <h6> the lowest.

<p>: Defines a paragraph of text.

<hr>: Creates a horizontal line, often used to separate content.

: Inserts a single line break.

<blockquote>: Used for quoting content from another source.

<pre>: Defines preformatted text, which preserves both spaces and line breaks.

Step 4: Text Formatting

 and : Both typically display text in bold. is preferred as it adds semantic meaning of strong importance.

<i> and : Both typically display text in italics. is preferred as it adds semantic meaning of emphasis.

<u>: Underlines text (use with caution as it can be confused with hyperlinks).

<sub>: Defines subscript text.

<sup>: Defines superscript text.

Code:index.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 | VNEC</title>

</head>

<body>

    <marquee>Welcome to VNEC </title></marquee>

    <!-- Different heading tags -->

    <h1>This is an H1 heading</h1>

    <h2>This is an H2 heading</h2>

    <h3>This is an H3 heading</h3>

    <h4>This is an H4 heading</h4>

    <h5>This is an H5 heading</h5>

    <h6>This is an H6 heading</h6>

    <!-- Paragraph -->

    <p>This is a paragraph demonstrating the use of the paragraph tag in
    HTML.</p>

    <!-- Horizontal line -->

    <hr>

    <!-- Line break -->

    <p>This is a line of text before the break.<br>This is a line of text
    after the break.</p>

    <!-- Block Quote -->

    <blockquote>

        This is a blockquote. It is used to display a quotation or excerpt
        from another source.

    </blockquote>
```

```
<!-- Pre tag -->
<pre>

    This is preformatted text.

    It preserves spaces and line breaks.

</pre>

<!-- Different Logical Style tags -->
<p>This is <b>bold</b> text.</p>
<p>This is <i>italicized</i> text.</p>
<p>This is <u>underlined</u> text.</p>
<p>This is <sup>superscript</sup> text.</p>
<p>This is <sub>subscript</sub> text.</p>
<p>This is <em>emphasized</em> text.</p>
<p>This is <strong>strong</strong> text.</p>
<p>This is <mark>highlighted</mark> text.</p>
<p>This is <small>small</small> text.</p>
<p>This is <del>deleted</del> text.</p>
<p>This is <ins>inserted</ins> text.</p>    <p>This is <code>inline
code</code> text.</p>
</html>
```

Output:

This is an H1 heading
This is an H2 heading
This is an H3 heading
This is an H4 heading
This is an H5 heading
This is an H6 heading
This is a paragraph demonstrating the use of the paragraph tag in HTML.
This is a line of text before the break.
This is a line of text after the break.
This is a blockquote. It is used to display a quotation or excerpt from another source.
This is preformatted text.
It preserves spaces and line breaks.
This is bold text.
This is italicized text.
This is underlined text.
This is superscript text.
This is subscript text.
This is emphasized text.
This is strong text.
This is highlighted text.
This is small text.
This is deleted text.
This is inserted text.
This is inline code text.

Explanation

Step 1: Document Structure

The document starts with the standard HTML5 declaration and structure. The `<head>` section contains metadata and styling information. The `<body>` section contains the visible content, which in this case is a table.

Step 2: Styling (CSS)

The `<style>` tag in the `<head>` section defines the appearance of the table and its elements. It sets the font, adjusts spacing, and defines colors for different types of classes. Specific classes are created for lab hours, elective hours, lunch, and alternating row colors.

Step 3: Table Structure

The table is created using the `<table>` tag. It has three main parts: `<thead>` (table header), `<tbody>` (table body), and `<tfoot>` (table footer).

Step 4: Table Header () The first row spans all columns and displays "Class Time Table". The second row shows the days of the week and a "Time" column.

Step 5: Table Body () Each represents a row in the timetable. The first column in each row shows the time slot. Subsequent columns represent different subjects for each day. Special formatting is applied to certain cells: Lab hours are highlighted in pink. Elective hours are highlighted in light green. The lunch break spans across multiple columns. The "Saturday" column spans multiple rows to show "No Classes". Alternate rows have a slightly different background color for better readability.

Step 6: Special Cell Attributes `colspan` is used to make cells span multiple columns (e.g., for labs and lunch). `rowspan` is used to make the "Saturday" cell span multiple rows. Classes are applied to cells for specific styling (e.g., "lab-hours", "elective-hours").

Step 7: Table Footer () The footer provides a legend explaining the color coding used in the table. This HTML document demonstrates several important concepts: Table structure and formatting Use of CSS for styling Cell spanning (both rows and columns) Color coding for different types of classes Responsive design considerations (though the table itself isn't responsive)

Experiment-02

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.

Code:

```
<!DOCTYPE html>

<html lang="en"></html>

<head>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Time Table | vtucode</title>
    <style>

        body {
            font-family: Arial, sans-serif;
        }

        table {
            width: 80%;
            margin: 20px auto;
            border-collapse: collapse;
        }

        th, td {
            border: 1px solid #ddd;
            padding: 8px;
            text-align: center;
        }

    </style>

```

```
th {  
    background-color: #f4f4f4;  
    color: #333;  
}  
  
tr:nth-child(even) {  
    background-color: #f9f9f9;  
}  
  
tr:nth-child(odd) {  
    background-color: #e6f7ff;  
}  
  
.lab-hour {  
    background-color: #ffcccc;  
}  
  
.imp-sub{  
    background-color: slateblue;  
}  
  
.elective-hour {  
    background-color: #ccffcc;  
}  
  
.highlight {  
    font-weight: bold;  
    color: #d63384;  
}  
  
tfoot {  
    background-color: #e0e0e0;  
    font-weight: bold;  
}  
  
</style>  
</head>  
<body>
```

```
<h1 style="text-align: center;">VNEC CSE</h1>
<h2 style="text-align: center;">CLASS TIME TABLE OF ACADEMIC YEAR-2025-
26(ODD)</h2>
<p style="padding-left: 200px;">SEM:VI A&B</p>
<table>
  <thead>
    <tr>
      <th>Day/Time</th>
      <th>Session 1 9:45am - 11:15am</th>
      <th>Session 2 11:30am - 1:00pm</th>
      <th> 1:00pm - 1:45pm</th>
      <th>Session 3 1:45pm - 3:15pm</th>
      <th>Session 4 3:30pm - 4:50pm</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Monday</td>
      <td>RM</td>
      <td class="imp-sub">AI</td>
      <td rowspan="6" class="highlight">Lunch Break</td>
      <td>TOC</td>
      <td class="lab-hour">CN Lab B 1</td>
    </tr>
    <tr>
      <td>Tuesday</td>
      <td class="elective-hour">SEPM</td>
      <td>RM</td>
      <td>CN</td>
      <td class="lab-hour">WEB Lab 2</td>
    </tr>
```

```
<tr>
    <td>Wednesday</td>
    <td>TOC</td>
    <td class="imp-sub">AI</td>
    <td class="elective-hour">SEPM</td>
    <td class="lab-hour">CN  Lab 2</td>
</tr>
<tr>
    <td>Thursday</td>
    <td>CN</td>
    <td>TOC</td>
    <td class="lab-hour">WEB  Lab 1</td>
    <td>Mini Project</td>
</tr>
<tr>
    <td>Friday</td>
    <td class="imp-sub">AI</td>
    <td>CN</td>
    <td>BNSK508</td>
    <td>Mini Project</td>
</tr>
<tr>
    <td>Saturday</td>
    <td class="elective-hour">SEPM</td>
    <td>TOC</td>
    <td colspan="2">NSS/PE/YOGA</td>
</tr>
</tbody>
<tfoot>
<tr>
```

```

<td colspan="8">End of Timetable</td>
</tr>
</tfoot>
</table>
</body>
</html>

```

OutPut:

VNEC CSE

CLASS TIME TABLE OF ACADEMIC YEAR-2025-26(ODD)

SEM:VI A&B

Day/Time	Session 1 9:45am - 11:15am	Session 2 11:30am - 1:00pm	1:00pm - 1:45pm	Session 3 1:45pm - 3:15pm	Session 4 3:30pm - 4:50pm
Monday	RM	AI		TOC	CN Lab B 1
Tuesday	SEPM	RM		CN	WEB Lab 2
Wednesday	TOC	AI		SEPM	CN Lab 2
Thursday	CN	TOC		WEB Lab 1	Mini Project
Friday	AI	CN		BNSK508	Mini Project
Saturday	SEPM	TOC			NSS/PE/YOGA
End of Timetable					

Experiment-03

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.

Note: make two file one is style.css and other is index.html then type the below code for different file.

```
<!DOCTYPE html>

<head>
    <title>Styled HTML Page </title>
    <link rel="stylesheet" href="3exp.css">
</head>

<body>
    <h2>Main Heading</h2>
    <h3>Subheading</h3>
    <hr>
    <p>This is a paragraph demonstrating the basic text styling applied by CSS.</p>
    <div>
        This is a styled <strong>div</strong> element with padding and a light border. Inside the div, we can also use
        <span>span elements</span> that have their own styles, like this bold and orange text.
    </div>
    <p>Current time: <time>10:30 AM</time></p>
    
    <p>Visit <a href="http://veerappanistyeecs.org/">http://veerappanistyeecs.org/</a> to know more about our college.</p>
    <p class="highlight">This paragraph is highlighted with a yellow background.</p>
    <p class="center">This text is centered using a class selector.</p>
    <p id="special-paragraph">This is a special paragraph with unique styles applied through an ID selector.</p>
```

```
</body>  
</html>  
Style.css:  
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
  
h2 {  
    color: #2c3e50;  
    font-size: 2em;  
    margin-bottom: 10px;  
}  
  
h3 {  
    color: #34495e;  
    font-size: 1.5em;  
    margin-bottom: 8px;  
}  
  
hr {  
    border: 0;  
    height: 2px;  
    background-color: #e74c3c;  
    margin: 20px 0;  
}  
  
p {  
    font-family: 'Arial', sans-serif;  
    line-height: 1.6;  
    margin: 10px 0;  
}  
  
div {
```

```
padding: 15px;  
border: 1px solid #bdc3c7;  
background-color: #ecf0f1;  
}  
  
span {  
color: #e67e22;  
font-weight: bold;  
}  
  
time::before {  
content: '⌚ ';  
color: #16a085;  
}  
  
img {  
margin-left: 15px;  
height: 300px;  
width: 400px;  
border-radius: 8px;  
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);  
max-width: 100%;  
}  
  
a {  
text-decoration: none;  
color: #ea0e4c;  
}  
  
a:hover {  
color: #6200ee;  
text-decoration: underline;  
}  
  
.highlight {  
background-color: yellow;
```

```
padding: 3px;  
}  
  
.center {  
    text-align: center;  
}  
  
#special-paragraph {  
    font-size: 1.2em;  
    color: #8e44ad;  
    background-color: #f5f5f5;  
    padding: 10px;  

```

Explanation

Step 1: HTML Structure

The document starts with the standard HTML5 declaration and basic structure.

The <head> section includes metadata, title, and a link to an external CSS file (style.css).

The <body> contains a <main> element with the id "main-content", which holds all the visible content.

Step 2: Main Content Structure

The main content includes various HTML elements like h2, h3, p, hr, section, article, img, and a. These elements demonstrate different CSS selectors and properties.

Step 3: CSS Styling (from style.css)

Element Selectors:

h2: Styled with color, font-family, border-bottom, and padding.

h3: Has a hover effect changing color with a transition.

hr: Styled as a gradient line.

Attribute Selectors:

Class Selector: p[lang]: Applies italic style to paragraphs with a lang attribute.

time[datetime]: Styles time elements with a datetime attribute.

img[alt]: Ensures responsive sizing for images with alt text.

.highlight: Applies background color and padding.

ID Selector: #main-content: Sets max-width, margin, padding, and background color.

Descendant Selector:

div p: Styles paragraphs inside divs (note: no divs in this HTML, so this won't apply).

Child Selector:

div > span: Styles spans that are direct children of divs (also won't apply here).

Adjacent Sibling Selector:

h2 + p: Styles paragraphs immediately following h2 elements.

Pseudo-element Selector:

p::first-letter: Styles the first letter of each paragraph.

Multiple Selectors:

img, a: Applies border and padding to both images and links.

Pseudo-class Selectors for Links:

Combining Selectors:

a:link, a:visited, a:hover, a:active: Different styles for various link states.

div.special p: Would style paragraphs in divs with class "special" (not applicable here).

Step 4: Specific Content and Styling Examples

The h2 and following p demonstrate the adjacent sibling selector. The h3 has a hover effect.

A paragraph with lang="en" shows the attribute selector in action.

The "highlight" class is applied to a span within a paragraph.

A section with a paragraph and span demonstrates descendant and child selectors (though the CSS for div won't apply).

The <time> element shows the datetime attribute selector.

An article with class="special" is included, but the CSS for div.special won't apply.

An image and a link are included to demonstrate various selectors and responsive design.

OutPut

Main Heading

Subheading

This is a paragraph demonstrating the basic text styling applied by CSS.

This is a styled div element with padding and a light border. Inside the div, we can also use **span elements** that have their own styles, like this bold and orange text.

Current time:  10:30 AM



Visit <http://veerappanistyles.org/> to know more about our college.

This paragraph is highlighted with a yellow background.

This text is centered using a class selector.

| This is a special paragraph with unique styles applied through an ID selector.

Experiment-04

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.

Index.html

```
<!DOCTYPE html>

<head>
    <title>Registration Form | VTU CSE VNEC</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f4f8;
            margin: 0;
            padding: 20px;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
        }
        .container {
            width: 100%;
            max-width: 600px;
            background-color: #fff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            display: flex;
            flex-direction: column;
            gap: 20px;
        }
    </style>
</head>
<body>
    <div class="container">
        <table border="1">
            <tr>
                <td>First Name</td>
                <td>Last Name</td>
            </tr>
            <tr>
                <td>Address</td>
                <td>City</td>
            </tr>
            <tr>
                <td>Email ID</td>
                <td>Phone No.</td>
            </tr>
            <tr>
                <td>Gender</td>
                <td>Age</td>
            </tr>
            <tr>
                <td>Date of Birth</td>
                <td>Blood Group</td>
            </tr>
            <tr>
                <td>Hobbies</td>
                <td>Interests</td>
            </tr>
            <tr>
                <td>Upload Photo</td>
                <td></td>
            </tr>
        </table>
    </div>
</body>
</html>
```

```
h2 {  
    text-align: center;  
    color: #333;  
    margin: 0;  
}  
.form-group {  
    display: flex;  
    flex-direction: column;  
    gap: 5px;  
    margin-bottom: 10px;  
}  
label {  
    font-size: 14px;  
    color: #555;  
}  
input[type="text"],  
input[type="email"],  
input[type="password"],  
input[type="date"],  
select,  
textarea {  
    padding: 10px;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
    font-size: 14px;  
}  
.gender-options {  
    display: flex;  
    gap: 10px;  
    align-items: center;  
}
```

```
        }

        input[type="submit"],
        input[type="reset"] {
            padding: 10px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            font-size: 16px;
            flex: 1;
        }

.button-group {
    display: flex;
    gap: 10px;
    justify-content: center;
}

input[type="submit"] {
    background-color: #4CAF50;
    color: white;
}

input[type="reset"] {
    background-color: #f44336;
    color: white;
}

.form-group textarea {
    margin-bottom: 10px;
}

</style>

</head>

<body>

<div class="container">
```

```
<h2>Registration Form</h2>

<form action="#" method="post">

    <div class="form-group">
        <label for="firstName">First Name:</label>
        <input type="text" id="firstName" name="firstName" required>
    </div>

    <div class="form-group">
        <label for="lastName">Last Name:</label>
        <input type="text" id="lastName" name="lastName" required>
    </div>

    <div class="form-group">
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
    </div>

    <div class="form-group">
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>
    </div>

    <div class="form-group">
        <label for="dob">Date of Birth:</label>
        <input type="date" id="dob" name="dob">
    </div>

    <div class="form-group">
        <label>Gender:</label>
        <div class="gender-options">
            <input type="radio" id="male" name="gender" value="male">
            <label for="male">Male</label>
            <input type="radio" id="female" name="gender" value="female">
            <label for="female">Female</label>
        </div>
    </div>

```

```
</div>

<div class="form-group">
    <label for="country">Country:</label>
    <select id="country" name="country">
        <option value="usa">USA</option>
        <option value="canada">Canada</option>
        <option value="uk">UK</option>
        <option value="india">India</option>
    </select>
</div>

<div class="form-group">
    <label for="bio">Bio:</label>
    <textarea id="bio" name="bio" rows="4"></textarea>
</div>

<div class="button-group">
    <input type="submit" value="Register">
    <input type="reset" value="Reset">
</div>
</form>
</div>
</body>
</html>
```

OutPut:

The screenshot shows a registration form titled "Registration Form". The form includes fields for First Name, Last Name, Email, Password, Date of Birth (with a date input and a calendar icon), Gender (with radio buttons for Male and Female, where Male is selected), Country (with a dropdown menu showing "India"), and Bio (with a text area containing a single character). At the bottom are two buttons: a green "Register" button and a red "Reset" button.

Registration Form	
First Name:	<input type="text"/>
Last Name:	<input type="text"/>
Email:	<input type="text"/>
Password:	<input type="text"/>
Date of Birth:	<input type="text"/> mm / dd / yyyy
Gender:	<input checked="" type="radio"/> Male <input type="radio"/> Female
Country:	<input type="text" value="India"/>
Bio:	<input type="text" value="I"/>
Register Reset	

Explanation

Step 1: Document Structure

The document starts with the standard HTML5 declaration and structure. The <head> section contains metadata, title, and embedded CSS styles. The <body> section contains the form.

Step 2: Styling (CSS)

The style section defines the appearance of the form and its elements. It sets the font, colors, layout, and responsive design for the form. Specific styles are applied to different input types, labels, and the submit button. A class for error messages is defined but not used in the HTML.

Step 3: Form Structure

The form is created using the <form> tag with a placeholder action "#". A table is used to structure the form layout.

Step 4: Form Fields

The form includes the following fields:

Full Name (text input)

Email (email input)

Password (password input)

Confirm Password (password input)

Gender (radio buttons)

Date of Birth (date input)

Country (select dropdown)

Interests (checkboxes)

Bio (textarea)

Submit button

Step 5: Input Attributes and Validation

Most fields are marked as required. Appropriate input types are used (e.g., email for email address, date for birthdate). The password field has a confirmation field. The country dropdown includes a default "Select a country" option.

Step 6: Styling Details The form has a white background with rounded corners and a shadow. Input fields have consistent styling with padding and border-radius. The submit button has a green color with a hover effect. The layout is responsive, with a maximum width set for larger screens. This registration form demonstrates several important concepts: HTML form structure and various input types Basic client-side form validation using HTML5 attributes Responsive design using CSS Styling form elements for a consistent and appealing look Use of tables for form layout (though modern practices often prefer CSS grid or flexbox).

Experiment-05

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

```
<!DOCTYPE html>

<head>
  <title>Newspaper Page | VNEC  </title>
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }
    body {
      padding: 20px;
      font-family: 'Arial', sans-serif;
      color: #000000;
      display: flex;
      flex-direction: column;
      min-height: 100vh;
    }
    header {
      margin-bottom: 30px;
      border-radius: 10px;
      align-items: center;
      background-color: #7b38f7;
      color: #fff;
      padding: 20px; display: flex;
      justify-content: space-between; text-align: center;
      box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    }
  </style>
</head>
<body>
  <header>
    <h1>VNEC</h1>
    <ul>
      <li>Home</li>
      <li>About</li>
      <li>Contact</li>
    </ul>
  </header>
  <main>
    <h2>Latest News</h2>
    <ul>
      <li>Breaking News: Major Earthquake in Japan</li>
      <li>U.S. President Visits South Korea</li>
      <li>Global Climate Change Conference in Paris</li>
    </ul>
  </main>
  <footer>
    <p>Copyright © 2017 VNEC. All Rights Reserved.</p>
  </footer>
</body>

```

```
}

header a {

    font-size: 25px;
    font-weight: 600;
    color: #fff;
    text-decoration: none;
}

nav {

    display: flex;
    gap: 20px;
    color: #fff;
    text-align: center;
}

nav a {

    font-size: 18px;
    color: #fff;
    text-decoration: none;
    font-weight: bold;
}

nav a:hover {

    text-decoration: underline;
}

.content {

    display: flex;
    justify-content: space-between;
    flex: 1;
    margin: auto;
    padding: 20px 0;
    gap: 20px;
    position: relative;
}
```

```
}

.main-content {
    cursor: pointer;
    flex: 1;
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
    gap: 20px;
    background-color: #fff;
    border-radius: 12px;
    padding: 25px;

box-shadow: rgba(9, 30, 66, 0.25) 0px 1px 1px, rgba(9, 30, 66, 0.13) 0px 0px
1px 1px;

}

aside {
    border: 1px solid #ccc;
    padding: 20px;
    width: 350px;
    border-radius: 8px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    position: -webkit-sticky;
    position: sticky;
    top: 20px;
    color: #333;
    right: 0;
    margin-left: 20px;
}

.related-news h3 {
    text-align: center;
    border-radius: 7px;
    padding: 8px;
    background: #000;
```

```
        color: #ffffff;
        font-size: 1.4em;
        margin-bottom: 15px;
    }

    .related-news ul {
        list-style: outside;
        padding: 7px;
        margin: 0;
    }

    .related-news li {
        margin-bottom: 12px;
    }

    .related-news a {
        text-decoration: none;
        color: #7b38f7;
        font-weight: bold;
        transition: color 0.3s ease;
    }

    .related-news a:hover {
        text-decoration: underline;
    }

    footer {
        border-radius: 10px;
        background-color: #7b38f7;
        color: #fff;
        padding: 20px;
        font-weight: 500;
        text-align: center;
        margin-top: auto;
        font-size: 18px;
    }
```

```
}

article {
    transition: all 0.3s ease;
    background-color: #fff;
    padding: 15px;
    box-shadow: rgba(9, 30, 66, 0.25) 0px 1px 1px, rgba(9, 30, 66, 0.13) 0px 0px 1px 1px;
    border-radius: 7px;
    color: #000000;
}

figure {
    background-color: #fafafa;
    padding: 10px;
    border: 1px solid #ddd;
    border-radius: 8px;
    text-align: center;
    margin: 0;
}

figcaption {
    font-size: 0.9em;
    color: #666;
}

img {
    max-width: 100%;
    height: auto;
    border-radius: 8px;
}

section {
    padding: 20px;
    width: 100%;
    background-color: #fff;
```

```
border-radius: 8px;  
box-shadow: rgba(9, 30, 66, 0.25) 0px 1px 1px, rgba(9, 30, 66,  
0.13) 0px 0px 1px 1px;  
}  
  
section h2 {  
color: #fff;  
background: #000;  
font-size: 24px;  
border-radius: 10px;  
text-align: center;  
padding: 10px;  
margin-bottom: 30px;  
}  
  
table {  
width: 100%;  
border-collapse: collapse;  
}  
  
caption {  
font-size: 18px;  
margin-bottom: 10px;  
color: #555;  
}  
  
thead {  
background-color: #007BFF;  
color: #fff;  
}  
  
th, td {  
padding: 12px;  
text-align: left;  
}  
  
th {
```

```
        font-weight: bold;
    }

    tbody tr:nth-child(even) {
        background-color: #f9f9f9;
    }

tbody tr:hover {
    background-color: #eaeaea;
}

@media (max-width: 600px) {
    th,
    td {
        padding: 8px;
        font-size: 14px;
    }
}

caption {
    background-color: #d9d9d9;
    padding: 10px;
    font-weight: bold;
    border-bottom: 2px solid #ddd;
    border-radius: 8px 8px 0 0;
    font-size: 1.1em;
    color: #333;
}

section {
    margin-top: 40px;
    margin-bottom: 50px;
}

article h2 {
    color: #7b38f7;
```

```
        font-size: 1.3em;
        margin-bottom: 12px;
    }

    article p {
        text-align: left;
        line-height: 1.2;
        margin-top: 10px;
    }

    article:hover {
        background-color: #e7ddfb;
    }

    @media (max-width: 768px) {
        .content {
            flex-direction: column;
            padding: 10px;
        }

        aside {
            width: 100%;
            margin-top: 20px;
            position: static;
            margin-left: 0;
        }

        .main-content {
            grid-template-columns: 1fr;
        }
    }
}

</style>

</head>

<body>

    <header>
```

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

<nav>

    <a href="#">Home</a>
    <a href="#">About</a>
    <a href="#">Contact</a>
    <a href="#">Services</a>
    <a href="#">Marketing</a>
    <a href="#">Updates</a>

</nav>

</header>

<div class="content">

    <main class="main-content">

        <article>

            <h2>Article Title 1</h2>

            <figure>

                <figcaption>Image Caption</figcaption>

            </figure>

            <p>This is the content of the first article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

        </article>

        <article>

            <h2>Article Title 2</h2>

            <figure>

                <figcaption>Image Caption</figcaption>

            </figure>

            <p>This is the content of the second article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

        </article>

    </main>

</div>
```

```
<article>

    <h2>Article Title 3</h2>

    <figure>

        <figcaption>Image Caption</figcaption>

    </figure>

    <p>This is the content of the third article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

</article>

<article>

    <h2>Article Title 4</h2>

    <figure>

        <figcaption>Image Caption</figcaption>

    </figure>

    <p>This is the content of the fourth article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

</article>

<article>

    <h2>Article Title 5</h2>

    <figure>

        <figcaption>Image Caption</figcaption>

    </figure>

    <p>This is the content of the fourth article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

</article>

<article>

    <h2>Article Title 6</h2>

    <figure>
```

```

    <figcaption>Image Caption</figcaption>
</figure>

<p>This is the content of the fourth article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
</article>
</main>
<aside class="related-news">
    <h3>Related News</h3>
    <ul>
        <li><a href="#">Related News 1</a></li>
        <li><a href="#">Related News 2</a></li>
        <li><a href="#">Related News 3</a></li>
    </ul>
</aside>
</div>
<section>
    <h2>Recent Posts</h2>
    <div>
        <table>
            <caption>List of Posts</caption>
            <thead>
                <tr>
                    <th>Post Title</th>
                    <th>Date</th>
                    <th>Author</th>
                </tr>
            </thead>
            <tbody>
                <tr>
```

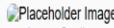
```
<td>Post 1</td>
<td>2024-08-30</td>
<td>Author 1</td>
</tr>
<tr>
<td>Post 2</td>
<td>2024-08-29</td>
<td>Author 2</td>
</tr>
<tr>
<td>Post 3</td>
<td>2024-08-28</td>
<td>Author 3</td>
</tr>
</tbody>
</table>
</div>
</section>
<footer>
<p>© 2024 Newspaper. All rights reserved.</p>
</footer>
</body>
</html>
```

OutPut:

Newspaper

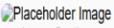
Home About Contact Services Marketing Updates

Article Title 1

Image Caption

This is the content of the first article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Article Title 2

Image Caption

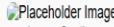
This is the content of the second article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Article Title 3

Image Caption

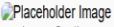
This is the content of the third article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Article Title 4

Image Caption

This is the content of the fourth article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Article Title 5

Image Caption

This is the content of the fourth article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Article Title 6

Image Caption

This is the content of the fourth article. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Related News

- Related News 1
- Related News 2
- Related News 3

Recent Posts

Experiment-6

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.

```
<!DOCTYPE html>
```

```
<head>
    <title>Simple Calculator | VTU</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }
        .calculator {
            background: #fff;
            padding: 20px;
            border-radius: 12px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
            width: 320px;
            text-align: center;
        }
        h1 {
            border-radius: 8px;
            background: #000;
            font-size: 24px;
            padding: 10px;
        }
    </style>
</head>
<body>
    <h1>CALCULATOR</h1>
    <div class="calculator">
        <input type="text" id="display" value="0" style="width: 280px; height: 40px; border: none; outline: none; font-size: 16px; margin-bottom: 10px;" />
        <div>
            <button>7</button>
            <button>8</button>
            <button>9</button>
            <button>/</button>
        </div>
        <div>
            <button>4</button>
            <button>5</button>
            <button>6</button>
            <button>*</button>
        </div>
        <div>
            <button>1</button>
            <button>2</button>
            <button>3</button>
            <button>-</button>
        </div>
        <div>
            <button>0</button>
            <button>.0</button>
            <button>+</button>
            <button>= </button>
        </div>
    </div>
</body>
</html>
```

```
        color: #ffffff;  
        margin-bottom: 30px;  
    }  
  
    input,  
    select,  
    button {  
        width: 100%;  
        margin: 10px 0;  
        padding: 12px;  
        border: 1px solid #0808081d;  
        border-radius: 8px;  
        font-size: 16px;  
        box-sizing: border-box;  
        transition: border-color 0.3s, box-shadow 0.3s;  
    }  
  
    #operation {  
        cursor: pointer;  
    }  
  
    input:focus,  
    select:focus,  
    button:focus {  
        outline: none;  
        border-color: #007bff;  
        box-shadow: 0 0 0 3px rgba(38, 143, 255, 0.25);  
    }  
  
    option {  
        background-color: #fff;  
        color: #000;  
        padding: 10px;  
        border: none;  
    }
```

```
}

option:hover {
    background-color: #f1f1f1;
}

button {
    background-color: #007bff;
    color: white;
    border: none;
    cursor: pointer;
    font-size: 18px;
    transition: box-shadow 0.3s, transform 0.3s;
}

button:hover {
    box-shadow: 0 0 0 2px #fff, 0 0 0 4px #007bff;
}

button:focus {
    box-shadow: 0 0 0 2px #fff, 0 0 0 4px #007bff;
}

#result.error {
    background: #ffdddd;
    border-color: #ff0000;
}

#result.success {
    font-size: 17px;
    font-weight: 500;
    color: #000;
    background: #6ef08d38;
    border-color: #47e56d;
}

#result {
```

```
        font-size: 18px;
        color: #000000;
        border-radius: 8px;
        background: #afffe2;
        border: 1px solid #ccc;
        box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
        transition: opacity 0.5s, transform 0.5s;
        opacity: 0;
        transform: translateY(-20px);
    }

    #result.show {
        cursor: not-allowed;
        opacity: 1;
        margin-top: 20px;
        padding: 15px;
        transform: translateY(0);
    }
}

</style>

</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" required>

                <option value="">Select Operation</option>
                <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="sqrt">Square Root</option>
<option value="square">Square</option>
</select>
<button type="button" onclick="calculate()">Calculate</button>
</form>
<div id="result"></div>
</div>
<script>
    function calculate() {
        const num1 = parseFloat(document.getElementById('num1').value);
        const num2 = parseFloat(document.getElementById('num2').value);
        const operation = document.getElementById('operation').value;
        let result = '';
        let resultClass = 'success';
        if (isNaN(num1) || isNaN(num2)) {
            result = 'Please enter valid numbers.';
            resultClass = 'error';
        } else {
            switch (operation) {
                case 'sum':
                    result = `Sum: ${num1 + num2}`;
                    break;
                case 'product':
                    result = `Product: ${num1 * num2}`;
                    break;
                case 'difference':

```

```
        result = `Difference: ${num1 - num2}`;

        break;

    case 'remainder':

        result = `Remainder: ${num1 % num2}`;

        break;

    case 'quotient':

        if (num2 === 0) {

            result = 'Cannot divide by zero';

            resultClass = 'error';

        } else {

            result = `Quotient: ${num1 / num2}`;

        }

        break;

    case 'power':

        result = `Power: ${Math.pow(num1, num2)}`;

        break;

    case 'sqrt':

        if (num1 < 0 || num2 < 0) {

            result = 'Square root is not defined for negative numbers';

            resultClass = 'error';

        } else {

            result = `Square Root of ${num1}: ${Math.sqrt(num1)} <br> Square Root of
${num2}: ${Math.sqrt(num2)}`;

        }

        break;

    case 'square':

result = `Square of ${num1}: ${Math.pow(num1, 2)} <br> Square of ${num2}:
${Math.pow(num2, 2)}`;

        break;

    default:

        result = 'Please select an operation.';
```

```
        resultClass = 'error';

    }

}

const resultDiv = document.getElementById('result');

resultDiv.innerHTML = result;

resultDiv.classList.remove('show', 'error', 'success');

resultDiv.classList.add(resultClass, 'show');

}

</script>

</body>

</html>
```

OUTPUT:

The screenshot shows a mobile application titled "Simple Calculator". It has two input fields containing "10" and "2", a dropdown menu set to "Sum", and a blue "Calculate" button. Below the button is a green box displaying the result "Sum: 12".

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.**

```
<!DOCTYPE html>

<head>

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

<title>Simple Converter | VNEC</title>

<style>

* {
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}

body {
    font-family: Arial, sans-serif;
    color: #000000;
}

.container {
    width: 60%;
    margin: 0 auto;
    padding: 20px;
}

.head-title h1 {
    font-size: 28px;
    padding: 10px;
    color: #fff;
    margin-bottom: 50px;
}
```

```
}

.head-title {
    width: 100%;
    background: #000;
    text-align: center;
    border-radius: 10px;
}

.section {
    margin-bottom: 40px;
    padding: 20px;
    border-radius: 8px;
    background: #fff;
    box-shadow: rgba(0, 0, 0, 0.1) 0px 1px 3px 0px, rgba(0, 0, 0, 0.06) 0px 1px 2px 0px;
    transition: all 0.3s;
    overflow: hidden;
}

.section h2 {
    color: #000000;
    font-size: 20px;
    margin-bottom: 15px;
}

textarea {
    font-size: 14px;
    width: 100%;
    height: 120px;
    margin-bottom: 15px;
    padding: 12px;
    border-radius: 8px;
    border: 1px solid #00000022;
    box-sizing: border-box;
}
```

```
        transition: border-color 0.3s ease, box-shadow 0.3s ease;
    }

    textarea:focus {
        background: transparent;
        border: 1px solid #00000022;
        border-color: #007BFF;
        box-shadow: 0 0 12px rgba(0, 123, 255, 0.5);
        outline: none;
    }

    input[type="text"] {
        width: calc(100% - 24px);
        padding: 12px;
        border-radius: 8px;
        border: 1px solid #ddd;
        box-sizing: border-box;
        transition: border-color 0.3s ease, box-shadow 0.3s ease;
        margin-bottom: 15px;
    }

    input[type="text"]:focus {
        border-color: #007BFF;
        box-shadow: 0 0 8px rgba(0, 123, 255, 0.5);
        outline: none;
    }

    button {
        display: inline-block;
        padding: 15px 15px;
        margin: 10px 0;
        font-weight: 600;
        border: none;
        border-radius: 7px;
    }
```

```
background-color: #007BFF;
color: #fff;
cursor: pointer;
font-size: 16px;
transition: box-shadow 0.3s ease, transform 0.3s ease;
}

button:hover {
    box-shadow: 0 0 0 2px #fff, 0 0 0 4px #007BFF;
}

button:focus {
    box-shadow: 0 0 0 2px #fff, 0 0 0 4px #007BFF;
}

pre {
    display: none;
    background: #f8f9fa;
    border: 1px solid #ddd;
    padding: 15px;
    border-radius: 8px;
    overflow: auto;
    transition: opacity 0.3s ease;
}

.error {
    margin-top: 10px;
    font-size: 14px;
    color: #000;
    background: #ffdddd;
    border-color: #ff0000;
    padding: 10px;
}
```

```
.success {  
    margin-top: 10px;  
    font-size: 14px;  
    color: #000;  
    background: #6ef08d38;  
    border-color: #47e56d;  
    padding: 10px;  
}  
.adjust-area {  
    margin-top: 30px;  
}  
</style>  
</head>  
<body>  
    <div class="container">  
        <div class="head-title">  
            <h1>Simple Converter</h1>  
        </div>  
        <div class="section">  
            <h2>1. Convert JSON Text to JavaScript Object</h2>  
            <textarea id="jsonInput" placeholder="Enter JSON here..."></textarea>  
            <button onclick="convertJsonToObject()">Convert JSON</button>  
            <pre id="jsonOutput" class="output"></pre>  
        </div>  
        <div class="section">  
            <h2>2. Convert JSON Results into Date</h2>  
            <textarea id="jsonDateInput" placeholder='Enter JSON with date in "yyyy-mm-dd" format'></textarea>  
            <button onclick="convertJsonToDate()">Convert to Date</button>  
            <pre id="jsonDateOutput" class="output"></pre>  
        </div>  
    </div>
```

```
<div class="section">

    <h2>3. Convert JSON to CSV and CSV to JSON</h2>

    <textarea id="jsonCsvInput" placeholder="Enter JSON for CSV conversion..."></textarea>

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

    <pre id="csvOutput" class="output"></pre>

    <textarea id="csvInput" placeholder="Enter CSV here..." class="adjust-area"></textarea>

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

    <pre id="jsonCsvOutput" class="output"></pre>

</div>

<div class="section">

    <h2>4. Create Hash from String</h2>

    <input type="text" id="hashInput" placeholder="Enter string to hash">

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

    <pre id="hashOutput" class="output"></pre>

</div>

</div>

<script>

    function showResult(id, text, isSuccess) {

        const element = document.getElementById(id);

        element.textContent = text;

        element.className = isSuccess ? 'success' : 'error';

        element.style.display = 'block';

        element.style.opacity = '1';

    }

    function convertJsonToObject() {

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

        try {

            const jsonObject = JSON.parse(jsonInput);


```

```
        showResult('jsonOutput', JSON.stringify(jsonObject, null, 2),
true);

    } catch (error) {

        showResult('jsonOutput', 'Invalid JSON', false);
    }

}

function convertJsonToDate() {

    const jsonDateInput =
document.getElementById('jsonDateInput').value;

    try {

        const data = JSON.parse(jsonDateInput);

        if (data.date && !isNaN(new Date(data.date).getTime())) {

            const date = new Date(data.date);

            showResult('jsonDateOutput', date.toString(), true);

        } else {

            showResult('jsonDateOutput', 'Invalid Date Format', false);

        }
    } catch (error) {

        showResult('jsonDateOutput', 'Invalid JSON', false);
    }
}

function convertJsonToCsv() {

    const jsonInput = document.getElementById('jsonCsvInput').value;
try {

    const jsonArray = JSON.parse(jsonInput);

    if (Array.isArray(jsonArray) && jsonArray.length > 0) {

        const keys = Object.keys(jsonArray[0]);

        const csv = [
            keys.join(','),
            ...jsonArray.map(row => keys.map(key =>
JSON.stringify(row[key])).join(','))

        ].join('\n');
    }
}
}
```

```
        showResult('csvOutput', csv, true);

    } else {

showResult('csvOutput', 'Invalid JSON: Expected an array with objects.', false);

    }

} catch (error) {

    showResult('csvOutput', 'Invalid JSON', false);

}

}

function convertCsvToJson() {

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

    try {

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

        if (lines.length > 1) {

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

            if (keys.length > 0) {

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

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

                    return keys.reduce((obj, key, index) => {

                        obj[key] = values[index];

                        return obj;

                    }, {});

                });

                showResult('jsonCsvOutput', JSON.stringify(jsonArray, null, 2), true);

            } else {

showResult('jsonCsvOutput', 'Invalid CSV: No columns found.', false);

            }

        } else {

showResult('jsonCsvOutput', 'Invalid CSV: No data found.', false);

        }

    } catch (error) {
```

```
        showResult('jsonCsvOutput', 'Invalid CSV', false);

    }

}

function createHash() {

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

if (hashInput.length > 0) {

    const hash = CryptoJS.SHA256(hashInput).toString();

    showResult('hashOutput', hash, true);

} else {

    showResult('hashOutput', 'Input cannot be empty', false);

}

}

</script>

</body>

</html>
```

OutPut:

The screenshot shows a web application interface for converting JSON text to JavaScript objects. The title bar reads "Simple Converter". Below the title, there is a section titled "1. Convert JSON Text to JavaScript Object". A code editor window contains the following JSON text:

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

Below the code editor is a blue button labeled "Convert JSON". To the right of the code editor is a vertical scroll bar. At the bottom of the application window, there is a green highlighted area containing the converted JavaScript object:

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

2. Convert JSON Results into Date

```
{  
    "date": "2024-09-01"  
}
```

Convert to Date

Sun Sep 01 2024 05:30:00 GMT+0530 (India Standard Time)

3. Convert JSON to CSV and CSV to JSON

```
[  
    {"name": "Alice", "age": 30, "city": "New York"},  
    {"name": "Bob", "age": 25, "city": "San Francisco"},  
    {"name": "Charlie", "age": 35, "city": "Chicago"}  
]
```

JSON to CSV

```
name,age,city  
"Alice",30,"New York"  
"Bob",25,"San Francisco"  
"Charlie",35,"Chicago"
```

CSV to JSON

```
[  
    {  
        "name": "Alice",  
        "age": "30",  
        "city": "New York"  
    },  
    {  
        "name": "Bob",  
        "age": "25",  
        "city": "San Francisco"  
    },  
    {  
        "name": "Charlie",  
        "age": "35",  
        "city": "Chicago"  
    }  
]
```

4. Create Hash from String

VNEC Shorapur

Create Hash

2102e9ce9f52df0665dcbd5fc75e9432a9b98bcd560f2b72844d758667b5b20c

Experiment-8A

8A. 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.

- Create a file name called **track.php**, write the below code and save it.
- Copy **track.php** file and open XAAMP directory if installed else install it [click here](#)
- There you'll find a folder named "htdocs".
- Inside the "htdocs" folder, paste **track.php** file.
- After then open your XAAMP and start the Apache server.
- Open your favorite [browser](#); we recommend using Google Chrome or Mozilla Firefox.
- Then, go to the URL "http://localhost/track.php".

Track.php:

```
<?php

$counterFile = "counter.txt";

if (!file_exists($counterFile)) {
    file_put_contents($counterFile, "0");
}

$currentCount = file_get_contents($counterFile);
$newCount = $currentCount + 1;
file_put_contents($counterFile, $newCount);

?>

<!DOCTYPE html>

<html lang="en">
<head>

    <title>Visitor Counter | vtucode</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }
    </style>

```

```
        text-align: center;
        margin: 0;
        padding: 0;
        display: flex;
        flex-direction: column;
        justify-content: center;
        height: 100vh;
        background-color: #f4f4f9;
        color: #333;
    }

    .container {
        background: #fff;
        padding: 20px;
        box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
        border-radius: 8px;
        margin: 0 auto;
        width: 60%;
    }

    h1 {
        font-size: 2.5em;
        margin: 0;
    }

    p {
        font-size: 1.2em;
        color: #555;
    }

```

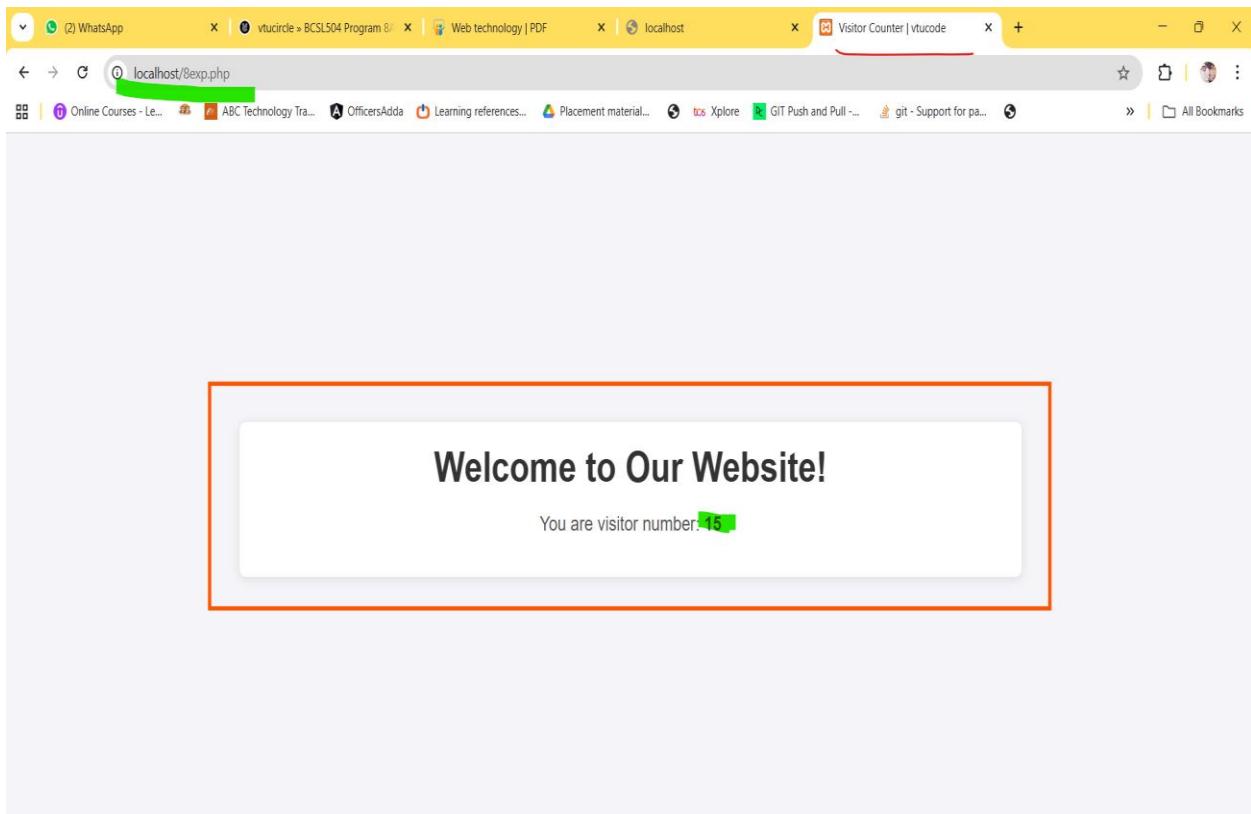
</style>

</head>

<body>

```
<div class="container">  
    <h1>Welcome to Our Website!</h1>  
    <p>You are visitor number: <strong><?php echo $newCount;  
?></strong></p>  
  </div>  
</body>  
</html>
```

OUTPUT:



Experiment-8B

8B .Develop a PHP program (with HTML/CSS) to sort the student records which are stored in the database using selection sort.

- Create a database name called students or download and import [click here](#)
- Create a file name called **sort_students.php**, copy the below code and paste it and save it.
- Copy **sort_students.php** file and open XAAMP directory if installed else install it [click here](#)
- There you'll find a folder named "htdocs".
- Inside the "htdocs" folder, paste **sort_students.php** file.
- After then open your XAAMP and start the Apache server.
- Open your favorite [browser](#); we recommend using Google Chrome or Mozilla Firefox.
- Then, go to the URL "http://localhost/sort_students.php".

Program sort.php:

```
<?php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "students";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {

    die("Connection failed: " . $conn->connect_error);

}

$sql = "SELECT * FROM students";

$result = $conn->query($sql);

$students = [];

if ($result->num_rows > 0) {

    while ($row = $result->fetch_assoc()) {
```

```
$students[] = $row;
}

}

function selectionSort(&$arr, $key)
{
    $n = count($arr);

    for ($i = 0; $i < $n - 1; $i++) {
        $minIndex = $i;

        for ($j = $i + 1; $j < $n; $j++) {
            if ($arr[$j][$key] < $arr[$minIndex][$key]) {
                $minIndex = $j;
            }
        }

        $temp = $arr[$i];
        $arr[$i] = $arr[$minIndex];
        $arr[$minIndex] = $temp;
    }
}

selectionSort($students, 'name');

?>

<!DOCTYPE html>

<head>

<title>Sorted Student Records | vtucode</title>

<style>

body {
```

```
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
background-color: #f0f2f5;  
color: #333;  
margin: 0;  
padding: 20px;  
}  
  
h2 {  
    text-align: center;  
    color: #4A90E2;  
    margin-bottom: 20px;  
}  
  
table {  
    width: 100%;  
    border-collapse: collapse;  
    background-color: #fff;  
    border-radius: 10px;  
    overflow: hidden;  
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);  
    margin: 0 auto;  
}  
  
th,td {  
    padding: 12px 15px;  
    text-align: left;  
    border-bottom: 1px solid #ddd;  
}
```

```
th {  
    background-color: #4A90E2;  
    color: white;  
    text-transform: uppercase;  
    letter-spacing: 0.03em;  
}  
  
tr {  
    transition: background-color 0.3s ease;  
}  
  
tr:hover {  
    background-color: #f1f1f1;  
}  
  
td {  
    font-size: 0.9em;  
    color: #555;  
}  
  
@media (max-width: 768px) {  
    table,  
    th, td {  
        display: block;  
        width: 100%;  
    }  
    th,td {  
        box-sizing: border-box;  
    }  
}
```

```
tr {  
    margin-bottom: 15px;  
    display: block;  
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);  
}  
  
th {  
    position: absolute;  
    top: -9999px;  
    left: -9999px;  
}  
  
td {  
    border: none;  
    position: relative;  
    padding-left: 50%;  
    text-align: right;  
}  
  
td:before {  
    content: attr(data-label);  
    position: absolute;  
    left: 0;  
    width: 50%;  
    padding-left: 15px;  
    font-weight: bold;  
    text-align: left;  
    text-transform: uppercase;
```

```
        color: #4A90E2;  
    }  
}  
</style>  
</head>  
  
<body>  
    <h2>VNEC CSE Dept Students Sorted Records by Name</h2>  
    <table>  
        <thead>  
            <tr>  
                <th>ID</th>  
                <th>Name</th>  
                <th>USN</th>  
                <th>Branch</th>  
                <th>Email</th>  
                <th>Address</th>  
            </tr>  
        </thead>  
        <tbody>  
            <?php foreach ($students as $student): ?>  
            <tr>  
                <td data-label="ID"><?php echo htmlspecialchars($student['id']); ?></td>  
                <td data-label="Name"><?php echo htmlspecialchars($student['name']); ?></td>  
                <td data-label="USN"><?php echo htmlspecialchars($student['usn']); ?></td>  
                <td data-label="Branch"><?php echo htmlspecialchars($student['branch']); ?></td>  
            </tr>  
        </tbody>  
    </table>
```

```

<td data-label="Email"><?php echo htmlspecialchars($student['email']); ?></td>

<td data-label="Address"><?php echo htmlspecialchars($student['address']); ?></td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

</body>

</html>

```

Output:

VNEC CSE Dept Students Sorted Records by Name					
ID	NAME	USN	BRANCH	EMAIL	ADDRESS
1	ABHILASH S KULKARNI	3VN23CS003	CSE	abkulkarni@gmail.com	Surpur
2	ABHISHEK SHIVRAJ	3VN23CS004	CSE	ABHISHEKSHIVRAJ@gmail.com	Shorapur
45	ABHISHEK SHIVRAJ	3VN23CS004	CSE	ABHISHEKSHIVRAJ@gmail.com	Shahapur
8	Arun Kumar	1ME21CS008	CSE	arun@gmail.com	Bhopal
5	BASAMMA	3VN23CS018	CSE	basamma@gmail.com	Gulbarga
6	BHUMIKA REDDY	3VN23CS023	AI&ML	bhomikareddy@gmail.com	Bidar, India
7	GIRIJA POLICEPATIL	3VN23CS030	DS	girija@gmail.com	Mangalore
10	LAYBA BATUL	3VN23CS036	ECE	layaba@gmail.com	Surpur
9	MOHAN KUMAR M	3VN23CS050	ISE	mohankumar@gmail.com	Yadgir
4	Shubham Kumar	1ME21CS004	CSE	shubham@gmail.com	Ghaziabad
3	Sunil Kumar	1ME21CS003	CIVIL	sunil@gmail.com	Delhi, India

Experiment-9

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.

```
<!DOCTYPE html>

<head>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <title>jQuery Example | vtucode</title>

    <style>

        body {

            font-family: 'Roboto', sans-serif;

            background-color: #f4f7f6;

            margin: 0;

            padding: 0;

            display: flex;

            justify-content: center;

            align-items: center;

            height: 100vh;

        }

        .container {

            text-align: center;

            background: #fff;

            padding: 30px;

            border-radius: 12px;

            box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);

            transition: transform 0.3s ease-in-out;

        }

        .container:hover {

            transform: scale(1.02);

        }

    </style>

</head>

<body>

    <div class="container">

        <p>Hello, I am a dynamic paragraph!</p>

        <ul>
            <li>Item 1</li>
            <li>Item 2</li>
            <li>Item 3</li>
        </ul>

    </div>

</body>

</html>
```

```
}

#paragraph {
    margin-bottom: 20px;
    color: #333;
    font-size: 18px;
    line-height: 1.5;
}

#list {
    margin-bottom: 20px;
    list-style: none;
    padding: 0;
}

#list li {
    background: #e8f0fe;
    margin: 5px 0;
    padding: 10px;
    border-radius: 8px;
    transition: background 0.3s;
}

#list li:hover {
    background: #d0e2fe;
}

.box {
    padding: 0 10px;
    width: 100px;
    height: 100px;
    background-color: #007bff;
    margin: 20px auto;
    line-height: 100px;
    color: white;
```

```
        text-align: center;
        border-radius: 8px;
        transition: all 0.3s ease;
    }

    button {
        padding: 12px 24px;
        margin: 10px;
        cursor: pointer;
        border: none;
        border-radius: 6px;
        font-size: 16px;
        background: #007bff;
        color: white;
        transition: box-shadow 0.3s, transform 0.2s;
        box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
    }

    button:hover {
        box-shadow: 0 0 0 2px #fff, 0 0 0 4px #007bff;
    }

    button:focus {
        box-shadow: 0 0 0 2px #fff, 0 0 0 4px #007bff;
    }

    button:active {
        background: #004494;
        transform: translateY(0);
    }

</style>

</head>

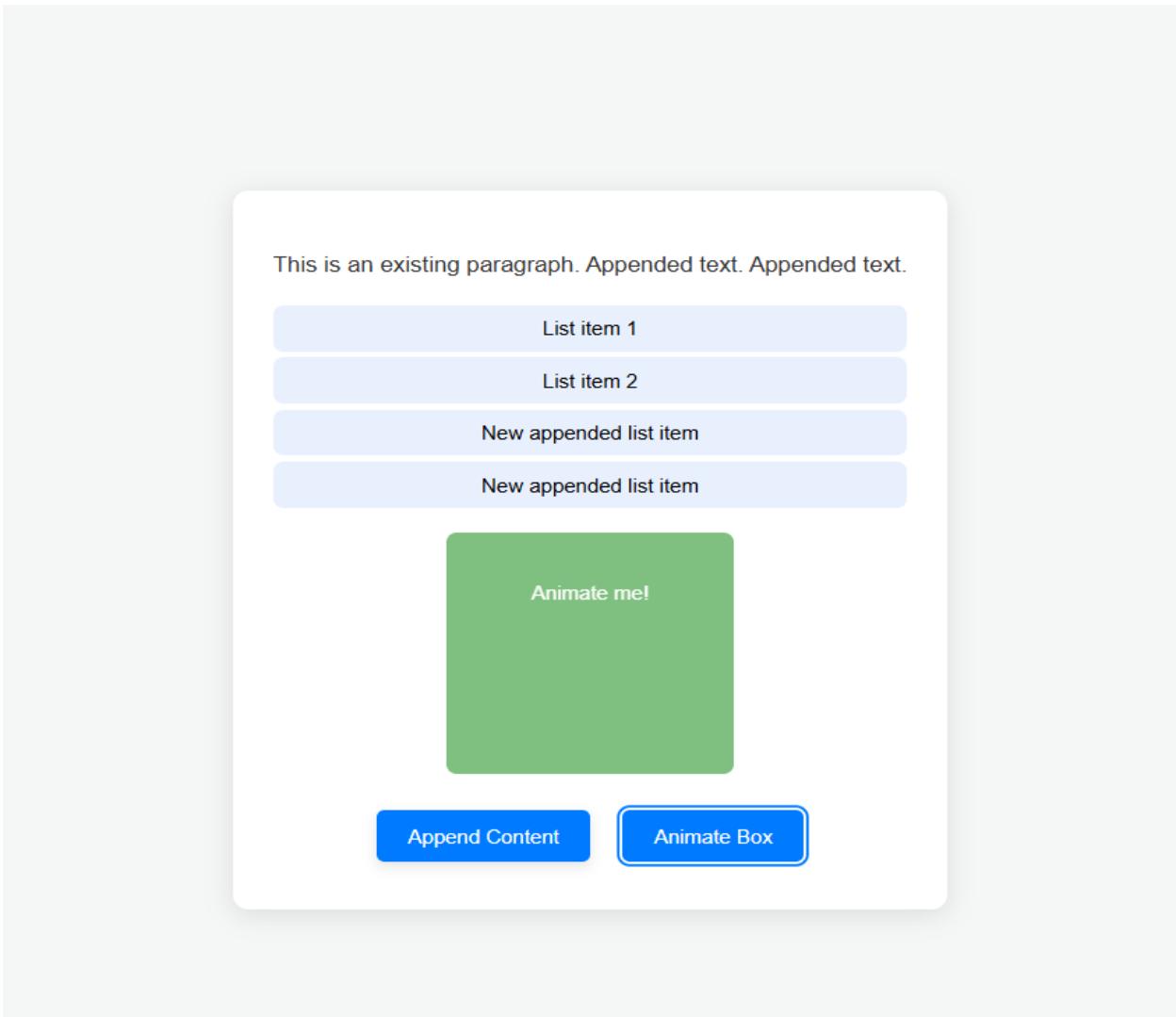
<body>

<div class="container">
```

```
<p id="paragraph">This is an existing paragraph.</p>
<ul id="list">
    <li>List item 1</li>
    <li>List item 2</li>
</ul>
<div class="box" id="box">Animate me!</div>
<button id="appendButton">Append Content</button>
<button id="animateButton">Animate Box</button>
</div>
<script>
$(document).ready(function () {
    $("#appendButton").click(function () {
        $("#paragraph").append(" Appended text.");
        $("#list").append("<li>New appended list item</li>");
    });
    $("#animateButton").click(function () {
        $("#box").stop(true, true).css({
            width: "100px",
            height: "100px",
            opacity: 1,
            backgroundColor: "blue"
        }).animate({
            width: "200px",
            height: "200px",
            opacity: 0.5
        }, 1000, function () {
            $(this).css("background-color", "green");
        });
    });
});
```

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

OutPut:



Experiment-10

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.**

Note: Create two separate file within the same folder one is **textfield.txt** and other **data.json** then copy below text for the both separate file and paste it save it.

textfield.txt

hi this is example text...

data.json

```
{ "name": "JohnDoe",
  "age": 30,
  "city": "NewYork",
  "skills": ["JavaScript", "React", "Node.js"],
  "address": {"street": "123 Elm Street", "zipcode": "10001"},
  "projects": [ { "name": "WebsiteRedesign", "year": 2023 },
    "technologies": ["HTML", "CSS", "JavaScript"] ],
  { "name": "Mobile App", "year": 2024, "technologies": ["React Native", "Expo"] } ] }
```

PROGRAM:

```
<!DOCTYPE html>
<head>
  <title>AJAX Examples | VNEC CSE</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
```

```
padding: 0;
background-color: #f4f4f9;
}

h1 {
    text-align: center;
    color: #333;
    padding: 20px 0;
}

#content {
    flex-direction: column;
    display: flex;
    max-width: 600px;
    margin: 20px auto;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 8px;
    background-color: #fff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

button {
    display: inline-block;
    padding: 10px 15px;
    margin: 12px;
    border: none;
    border-radius: 5px;
    background-color: #007bff;
    color: #fff;
    font-size: 16px;
    cursor: pointer;
    transition: box-shadow 0.3s;
```

```
}

button:hover {
    box-shadow: 0 0 0 2px #fff, 0 0 0 4px #007bff;
}

button:focus {
    box-shadow: 0 0 0 2px #fff, 0 0 0 4px #007bff;
}

#output {
    display: none;
    margin-top: 20px;
    padding: 10px;
    border-radius: 5px;
    white-space: pre-wrap;
    max-height: 300px;
    overflow-y: auto;
}

#output/plain-ajax {
    background-color: #f0f8ff;
    border: 1px solid #b0c4de;
}

#output/jquery-ajax {
    background-color: #f5ffff;
    border: 1px solid #98fb98;
}

#output/jquery-json {
    background-color: #ffffaf0;
    border: 1px solid #ffd700;
}

#output/parse-json {
    background-color: #ffff0f5;
```

```
        border: 1px solid #ff69b4;
    }
</style>
</head>
<body>
<h1>AJAX Examples</h1>
<div id="content">
    <button id="plain-ajax-btn">Load Text (Plain AJAX)</button>
    <button id="jquery-ajax-btn">Load Text (jQuery AJAX)</button>
    <button id="jquery-json-btn">Load JSON (jQuery getJSON)</button>
    <button id="parse-json-btn">Load and Parse JSON (jQuery get)</button>
    <div id="output"></div>
</div>
<script>
    function showOutput(className) {
        const output = document.getElementById('output');
        output.className = className;
        output.style.display = 'block';
    }
document.getElementById('plain-ajax-btn').addEventListener('click', function () {
    var xhr = new XMLHttpRequest();
    xhr.open('GET', 'textfile.txt', true);
    xhr.onload = function () {
        if (xhr.status === 200) {
            document.getElementById('output').innerText = xhr.responseText;
        } else {
            document.getElementById('output').innerText = 'Error loading file.';
        }
        showOutput('plain-ajax');
    };
});
```

```
        xhr.send();

    });

    $('#jquery-ajax-btn').on('click', function () {
        $.ajax({
            url: 'textfile.txt',
            method: 'GET',
            success: function (data) {
                $('#output').text(data);
            },
            error: function () {
                $('#output').text('Error loading file.');
            }
        }).always(function () {
            showOutput('jquery-ajax');
        });
    });

    $('#jquery-json-btn').on('click', function () {
        $.getJSON('data.json')
            .done(function (data) {
                $('#output').text(JSON.stringify(data, null, 2));
            })
            .fail(function () {
                $('#output').text('Error loading JSON file.');
            })
            .always(function () {
                showOutput('jquery-json');
            });
    });

    $('#parse-json-btn').on('click', function () {
        $.get('data.json')
```

```
.done(function (data) {
    try {
        let jsonData;

        if (typeof data === 'string') {
            jsonData = JSON.parse(data);
        } else {
            jsonData = data;
        }
        $('#output').text(JSON.stringify(jsonData, null, 2));
    } catch (e) {
        $('#output').text('Error parsing JSON: ' + e.message);
    }
})
.fail(function () {
    $('#output').text('Error loading JSON file.');
})
.always(function () {
    showOutput('parse-json');
});
});
```

</script>

</body>

</html>

Output:

AJAX Examples

Load Text (Plain AJAX)

Load Text (jQuery AJAX)

Load JSON (jQuery getJSON)

Load and Parse JSON (jQuery get)

hi this is example text...

AJAX Examples

Load Text (Plain AJAX)

Load Text (jQuery AJAX)

Load JSON (jQuery getJSON)

Load and Parse JSON (jQuery get)

hi this is example text...

AJAX Examples

Load Text (Plain AJAX)

Load Text (jQuery AJAX)

Load JSON (jQuery getJSON)

Load and Parse JSON (jQuery get)

```
{  
  "name": "John Doe",  
  "age": 30,  
  "city": "New York",  
  "skills": [  
    "JavaScript",  
    "React",  
    "Node.js"  
,  
  "address": {  
    "street": "123 Elm Street",  
    "zipcode": "10001"  
,  
  "projects": [  
    {  
      "name": "Website Redesign",  
      "year": 2023,  
    },  
  ]},  
}
```

AJAX Examples

Load Text (Plain AJAX)

Load Text (jQuery AJAX)

Load JSON (jQuery getJSON)

Load and Parse JSON (jQuery get)

```
{  
  "name": "John Doe",  
  "age": 30,  
  "city": "New York",  
  "skills": [  
    "JavaScript",  
    "React",  
    "Node.js"  
,  
  "address": {  
    "street": "123 Elm Street",  
    "zipcode": "10001"  
,  
  "projects": [  
    {  
      "name": "Website Redesign",  
      "year": 2023,  
    },  
  ]},  
}
```