Full Stack Development

Experiment 1: Create a Well-Structured HTML Page for a College Profile.

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>College Profile<title>
</head>
<body>
  <h1>Welcome to ABC Engineering College</h7>

  <p>ABC Engineering College was <b>established in 2001<p>

  <h2>Departments Offered</h2>
  <ul>
    <li>CSE
    <li>ECE
    <li>Mechanical
  </ul>

  <h2>Useful Links</h2>
  <a hre="https://www.abcengg.ac.in">Visit Official Website</a>
</body>
</html>
```

Bug Explanations

1. **<title> not closed properly**
   - Written as <title>College Profile<title> instead of <title>College Profile</title>.
   - Browser may not show the correct page title.
2. **Mismatched heading tag**

    - <h1> ... </h7> used → HTML only allows <h1> to <h6>.
    - Invalid tag breaks heading structure.
3. **Paragraph not closed before new <p>**
    - <p>ABC Engineering College was <b>established in 2001<p> → missing </b> and </p>.
    - Causes formatting errors in text.
4. **Unclosed <li> items in the list**
    - <li>CSE, <li>ECE, <li>Mechanical don't have closing </li>.
    - Works in some browsers but is invalid HTML.
5. **Wrong attribute in <a> link**
    - Used hre= instead of href=.
    - Link won't work when clicked.

Experiment 2: Design a Student Resume Webpage using Headings, Lists, and Links

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Student Resume<title>
</head>
<body>
  <h1>Student Resume<h2>

  <p>Name: <b>John Doe</p>
  <p>Email: john.doe@example,com</p>

  <h2>Skills</h3>
  <ul>
    <li>HTML
    <li>CSS
    <li>JavaScript
  </ul>
```

```
    <h2>Profile Links</h2>

    <a href-"https://www.linkedin.com/in/johndoe">LinkedIn</a>

</body>

</html>
```

Bug Explanations

1. **Title tag not closed properly**
   - <title>Student Resume<title> instead of </title>.
2. **Heading tags mismatched**
   - <h1>Student Resume<h2> instead of closing with </h1>.
3. **Unclosed <b> tag & wrong email**
   - <b> never closed.
   - example,com should be example.com.
4. **List items missing </li>**
   - <li> tags not closed properly → invalid HTML.
5. **Broken link attribute**
   - Used href-"..." instead of href="...".

Experiment 3: Create a College Departments Information Webpage using Ordered, Unordered, and Nested Lists.

Program:

```
<!DOCTYPE html>

<html>

<head>

    <title>College Departments<title>

</head>

<body>

    <h1>ABC Engineering College Departments</h1>


    <h2>Undergraduate Programs</h3>

    <ul>

        <li>CSE

        <li>ECE
        <li>Mechanical
    </ul>

    <h2>Postgraduate Programs</h2>
    <ol type="decimalx">
        <li>M.Tech in CSE</li>
        <li>M.Tech in ECE
        <li>MBA</li>
    </ol>

    <h2>Nested List Example</h2>
    <ul>
        <li>Science
            <ol>
                <li>Physics</li>
                <li>Chemistry
            </ol>
        </li>
        <li>Arts
            <ul>
                <li>History
                <li>Political Science</ul>
        </li>
    </ul>
</body>
</html>
```

Bug Explanations:

1. **Title tag not closed properly**

    - <title>College Departments<title> instead of </title>.
2. **Mismatched heading tags**
    - <h2>Undergraduate Programs</h3> → closed with </h3> instead of </h2>.
3. **Unclosed <li> in UG program list**
    - <li>CSE, <li>ECE, <li>Mechanical not closed → invalid list.
4. **Invalid attribute in ordered list**
    - type="decimalx" is invalid. Only "1", "A", "a", "I", "i" are valid.
5. **Unclosed list items in nested list**
    - <li>Chemistry and <li>History not closed

Experiment 4: Develop a College Website Navigation Page using Basic and In-Page Links.

Program:

```
<!DOCTYPE html>

<html>

<head>

  <title>College Navigation<title>

</head>

<body>

  <h1>ABC Engineering College Website</h1>


  <h2>Navigation Menu</h2>

  <a hre="https://www.abcengg.ac.in">Official Website</a><br>

  <a href="#aboutcollege">About College</a><br>

  <a href="#departments">Departments</a><br>

  <a href="#contact">Contact</a><br>


  <h2 id="aboutcollege">About College<h2>

  <p>ABC Engineering College was established in 2001 and offers various UG and PG
programs.</p>


  <h2 id="departments">Departments</h3>

  <ul>

    <li>CSE
```

```
        <li>ECE
        <li>Mechanical
    </ul>

    <h2 id="contact">Contact Us</h2>
    <p>Email: info@abcengg,ac.in</p>
</body>
</html>
```

Bug Explanations

1. **Title tag not closed properly**
   - `<title>College Navigation<title>` instead of `</title>`.
2. **Broken external link attribute**
   - Used `hre` instead of `href`. → link won't work.
3. **Mismatched heading tags**
   - `<h2 id="aboutcollege">About College<h2>` → should be `</h2>`.
   - `<h2 id="departments">Departments</h3>` → closed with `</h3>` instead of `</h2>`.
4. **Unclosed list items in Departments list**
   - `<li>CSE`, `<li>ECE`, `<li>Mechanical` not closed.
5. **Invalid email format**
   - Written as `info@abcengg,ac.in` (comma instead of dot).

Experiment 5: Create a Webpage with Images and Use them as Links.

Program:

```
<!DOCTYPE html>

<html>

<head>

    <title>Image Example<title>

</head>

<body>

    <h1>Our College Gallery</h1>
    <img src="college.jpg" alt=College Image>   <!-- missing quotes -->

    <br>



    <a href="https://www.abcengg.ac.in">

        <img scr="logo.png" alt="College Logo">

    </a>



    <img src="students.jpg" width="300px height="200px">

</body>

</html>
```

Bug Explanations

1. Title tag not closed properly.
2. alt=College Image → missing quotes.
3. Used scr instead of src for image.
4. Width/height attribute not closed properly.
5. Missing caption/description text for accessibility.

Experiment 6: Design a Webpage to Embed Audio and Video Files using HTML5 Media Tags.

```
<!DOCTYPE html>

<html>

<head>

    <title>Media Page<title>

</head>

<body>

    <h1>College Media Files</h1>
    <audio control>

        <source src="anthem.mp" type="audio/mp3">

    </audio>



    <video width="400" heigt="300" controls>

        <source src="introvideo.mp4" type="video/mp-4">

    </video>

</body>

</html>
```

Bug Explanations

1. Title not closed.
2. Used control instead of controls.
3. Wrong extension .mp instead of .mp3.
4. Misspelled heigt instead of height.
5. Wrong MIME type → video/mp-4 should be video/mp4.

Experiment 7: Develop a Webpage with a Student Marks Table using Caption, Rowspan, and Colspan.

Program:

```
<!DOCTYPE html>

<html>

<head>

    <title>Student Table<title>

</head>

<body>

    <h1>Student Marks</h1>



    <table border="1">
```

```
        <caption>Semester Marks</caption>
        <tr>
            <th>Name</th>
            <th colspan="2">Subjects<th>
        </tr>
        <tr>
            <td rowspan=2>John</td>
            <td>Maths</td>
            <td>90
        </tr>
        <tr>
            <td>Science</td>
            <td>85</td>
    </table>
</body>
</html>
```

Bug Explanations

1. Title not closed.
2. `<th colspan="2">Subjects<th>` missing `</th>`.
3. `<td rowspan=2>` → value should be in quotes.
4. `<td>90` not closed.
5. Last row `<tr>` not closed.

Experiment 8: Create a Student Registration Form using Labels, Form Controls, Fieldset, and Legend.

```
<!DOCTYPE html>
<html>
<head>
  <title>Registration Form<title>
</head>
<body>
  <h1>Student Registration</h1>

  <form action"submit.php" method="post">
    <fieldset>
      <legend>Personal Details</legend>

      <label Name:</label>
      <input type=text name="studentname"><br><br>

      <label>Email:</label>
      <input type="email" name"email"><br><br>

      <input type="submit" value=Register>
    </fieldset>
  </form>
</body>
</html>
```

Bug Explanations

1. Title not closed.
2. <form action"submit.php"> → missing =.
3. <label Name:</label> → invalid label syntax.

4. <input type=text> → type should be quoted.
5. name"email" → missing = sign.

Experiment 9: Create a Webpage using Frames and Frameset Elements.

Program:

```
<!DOCTYPE html>

<html>

<head>

  <title>Frames Example<title>

</head>

<frameset rows="50%, 50%">

  <frame scr="header.html" name="top">

  <frame src="content.html" name="main">

</frameset>

<body>

  <p>Frames Example Page</p>

</body>

</html>
```

Bug Explanations

1. Title not closed properly.
2. Used scr instead of src.
3. <frameset> cannot have <body> inside → invalid structure.
4. Missing closing tag for <frameset>.
5. Frame content should include noresize or title attributes for accessibility.

Experiment 10: Apply CSS Styles for Text Formatting and Pseudo-Classes.

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Text Example<title>
  <style>
    body {
      font-famly: Arial;
      background-colr: lightblue;
    }
    h1 {
      color blue;
    }
    a:hover {
      textdecoration: underline;
    }
  </style>
</head>
<body>
  <h1>Welcome to CSS Page</h1>
  <p>This is a paragraph with styled text.</p>
  <a href="https://www.abcengg.ac.in">Visit Website</a>
</body>
</html>
```

Bug Explanations:

1. Title not closed.
2. font-famly misspelled → should be font-family.
3. background-colr → should be background-color.
4. color blue; → missing : (should be color: blue;).
5. textdecoration → should be text-decoration.

Experiment 11: Use CSS Selectors (Universal, ID, Class, Child, Descendant, Sibling).

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Selectors<title>
  <style>
    * { margin 0; padding 0; }
    #header { background:lightgrey; font-size=20px; }
    .content { color:blue font-weight:bold; }
    div > p { colr: green; }
    h2 + p { font-stle: italic; }
  </style>
</head>
<body>
  <div id="header">
    <h1>Welcome Header</h1>
  </div>
  <div class="content">
    <h2>Section</h2>
    <p>This is a child paragraph.</p>
  </div>
```

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

Bug Explanations

1. Title not closed.
2. margin 0; → missing :.
3. font-size=20px; → should use : not =.
4. color:blue font-weight:bold; → missing ; between properties.
5. font-stle misspelled → should be font-style.

Experiment 12: Demonstrate CSS Box Model with Backgrounds, Lists, and Tables.

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Box Model<title>
  <style>
    .box {
      widht: 300px;
      heigt: 200px;
      border 2px solid black;
      padding 10;
      background-clor: yellow;
    }
    table {
      border-collapse collapse;
    }
  </style>
</head>
<body>
  <div class="box">
```

```
        <p>This is an example of CSS Box Model.</p>
    </div>


    <table border="1">
        <tr><th>Name<th><th>Marks</th></tr>
        <tr><td>John<td>90</tr>
    </table>
</body>
</html>
```

Bug Explanations

1. Title not closed.
2. widht and heigt misspelled.
3. border 2px solid black; → missing :.
4. padding 10; → should be padding: 10px;.
5. border-collapse collapse; → missing :.

Experiment 13: Write a JavaScript Program to Display Output using Alert, Console, and Document Write.

Program:

```
<!DOCTYPE html>
<html>
<head>
    <title>JS Output<title>
    <script>
        // Display different outputs
        alrt("Welcome to JavaScript");
        console,log("This is console output");
        document.write("Page Loaded")
    </script>
</head>
```

```
<body>

   <h1>JavaScript Output Example</h1>

</body>

</html>
```

Bug Explanations

1. Title not closed.
2. alrt misspelled → should be alert.
3. console,log → should be console.log.
4. document.write("Page Loaded") missing ;.
5. Script is written directly in <head> → better inside <body> or with defer.

Experiment 14: Create an External JavaScript File and Link it to a Webpage.

Program:

```
<!DOCTYPE html>

<html>

<head>

   <title>External JS<title>

   <script src="script.jss"></script>

</head>

<body>

   <h1>External Script Example</h1>

   <button onclick="sayhello()">Click Me</button>

</body>

</html>
```

script.js

```
funtion sayhello() {

   alert("Hello, Welcome to JS");

}
```

Bug Explanations

1. Title not closed.

2. Wrong file extension .jss → should be .js.
3. Function name sayhello not matched correctly with button (case-sensitive issues possible).
4. funtion misspelled → should be function.
5. Missing semicolon inside JS function.

Experiment 15: Write a JavaScript Program using Variables, Operators, and String Concatenation.

Program:

```
<!DOCTYPE html>

<html>

<head>

   <title>Variables<title>

   <script>

      var name = "John"

      var age = 20;

      var msg = "My name is " + name + " I am " + age years old;

      alret(msg);

   </script>

</head>

<body>

   <h1>JS Variables Example</h1>

</body>

</html>
```

Bug Explanations

1. Title not closed.
2. Missing semicolon after var name = "John".
3. "I am " + age years old; → invalid string concatenation (should be "I am " + age + " years old").
4. alret misspelled → should be alert.
5. Best practice: use let or const instead of var.

Experiment 16: Write a JavaScript Program to Demonstrate Functions, Conditional Statements, and Loops.

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Functions<title>
  <script>
    fucntion checkEvenOdd(num) {
      if(num % 2 = 0) {
        return "Even"
      } else {
        return "Odd"
    }

    for(var i=1; i<=5; i++) {
      console.log("Number " + i + " is " + checkEvenOdd(i));
  </script>
</head>
<body>
  <h1>JS Functions and Loops</h1>
</body>
</html>
```

Bug Explanations

1. Title not closed.
2. fucntion misspelled → should be function.
3. if(num % 2 = 0) → assignment instead of comparison (should be == or ===).

4. Missing closing brace } for function.
5. Missing closing brace } for loop.

Experiment 17: Demonstrate JavaScript Predefined Objects (Window, Document, Math, String, Array).

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Predefined Objects<title>
  <script>
    // Window object
    windw.alert("Welcome to JS Objects");

    // Document object
    docment.write("This text is written using document object<br>");

    // Math object
    var num = Math.squrt(16);

    // String object
    var str = "Hello World;
    console.log(str.toUperCase());

    // Array object
    var arr = [10,20,30];
    console.log(arr.lenght);
  </script>
</head>
<body>
```

```
    <h1>Predefined Object Demo</h1>
</body>
</html>
```

Bug Explanations

1. Title not closed.
2. windw.alert → misspelled window.
3. docment.write → misspelled document.
4. Math.squrt → should be Math.sqrt.
5. str.toUperCase() → misspelled toUpperCase; also missing closing quote in string.

Experiment 18: Access DOM Elements by ID, Class, and TagName.

Program:

```
<!DOCTYPE html>
<html>
<head>
    <title>Access DOM<title>
    <script>
        function changeText() {
            var heading = document.getElementId("title");
            heading.innerHtml = "Updated Heading";

            var paras = document.getElementsByClasName("para");
            paras[0].innerText = "First paragraph updated";

            var allDivs = document.getElementByTagName("div");
            allDivs[0].style.backgrounColor = "yellow";
        }
    </script>
</head>
<body onload="changeText()">
```

```
    <h1 id="title">Original Heading</h1>

    <p class="para">Paragraph 1</p>

    <p class="para">Paragraph 2</p>

    <div>Some content</div>

</body>

</html>
```

Bug Explanations

1. Title not closed.
2. getElementId → should be getElementById.
3. innerHtml → should be innerHTML.
4. getElementsByClasName → misspelled Class.
5. backgrounColor → should be backgroundColor.

Experiment 19: Update Content Using textContent, innerText, and innerHTML.

Program:

```
<!DOCTYPE html>

<html>

<head>

    <title>Update DOM<title>

    <script>

        function updateContent() {

            document.getElementById("p1").textcontent = "Updated with textContent";

            document.getElementById("p2").innertext = "Updated with innerText";

            document.getElementById("p3").innerHtml("Updated with innerHTML");

        }

    </script>

</head>

<body>

    <p id="p1">Original P1</p>

    <p id="p2">Original P2</p>
    <p id="p3">Original P3</p>

    <button onclick="updateContent()">Click to Update</button>

</body>

</html>
```

Bug Explanations

1. Title not closed.
2. textcontent → should be textContent (case-sensitive).
3. innertext → should be innerText.
4. innerHtml("...") → invalid, should be innerHTML = "...";.
5. Missing semicolons after each statement.

Experiment 20: Add and Remove Elements Dynamically from the DOM Tree.

Program:

```
<!DOCTYPE html>

<html>

<head>

    <title>DOM Manipulation<title>

    <script>

        function addElement() {

            var newPara = document.createElemnt("p");

            newPara.innerHTML = "New Paragraph Added";

            document.body.apendChild(newPara);

        }


        function removeElement() {

            var para = document.getElementById("removeme");

            document.body.removechild(para);

        }

    </script>

</head>
```

```
<body>
    <h1>DOM Add/Remove Example</h1>
    <p id="removeme">This will be removed</p>
    <button onclick="addElement()">Add Paragraph</button>
    <button onclick="removeElement()">Remove Paragraph</button>
</body>
</html>
```

Bug Explanations:

1. Title not closed.
2. createElemnt → should be createElement.
3. apendChild → should be appendChild.
4. removechild → should be removeChild (case-sensitive).
5. If element not found, removeChild will throw an error (needs check).