FULL STACK DEVELOPMENT
(SKILL ENHANCEMENT COURSE)
Experiment:1
(1. Lists, Links and Images)

1a)

Aim: Write a HTML program, to explain the working of lists. Note: It should have an ordered list, unordered list, nested lists and ordered list in an unordered list and definition lists.

Source code:

```
<doctype html>
    <html>
        <head>
            <title>work with lists</title>
        </head>
        <body>
            <h1>Types of lists</h1>
            <ol>
                <li>ordered list</li>
                <li>unordered list</li>
                <li>description list</li>
                <li>nested list</li>
            </ol>
            <h2>Description about the Lists</h2>
            <dl>
                <dt>ordered list</dt>
                <dd>An ordered list starts with the ol tag. Each list item starts with the li tag.
                    The list items will be marked with numbers by default:
                </dd>
                <dt>unordered list</dt>
                <dd>An unordered list starts with the ul tag. Each list item starts with the li tag.

                    The list items will be marked with bullets (small black circles) by default:</dd>
                <dt>description list</dt>
                <dd>A description list is a list of terms, with a description of each term.
                    The dl tag defines the description list, the dt tag defines the term (name), and
the dd tag describes each term:
                </dd>
                <dt>nested list</dt>
                <dd>Lists can be nested (list inside list):</dd>

            </dl>
            <h2>fullstack technologies</h2>
            <ol>
                <li>FRONT END TECHNOLOGIES</li>
                <ul type="square">
                    <li>HTML</li>
                    <li>CSS</li>
                    <li>JAVASCRIPT</li>
                </ul>
                <li>SERVER SIDE TECHNOLOGIES</li>
                <li>DATABASE SIDE TECHNOLOGIES</li>
            </ol>
       </body>
    </html>
```

Types of lists

1. ordered list
2. unordered list
3. description list
4. nested list

Description about the Lists

ordered list

An ordered list starts with the ol tag. Each list item starts with the li tag. The list items will be marked with numbers by default:

unordered list

An unordered list starts with the ul tag. Each list item starts with the li tag. The list items will be marked with bullets (small black circles) by default:

description list

A description list is a list of terms, with a description of each term. The dl tag defines the description list, the dt tag defines the term (name), and the dd tag describes each term:

nested list

Lists can be nested (list inside list):

fullstack technologies

1. FRONT END TECHNOLOGIES
    - HTML
    - CSS
    - JAVASCRIPT
2. SERVER SIDE TECHNOLOGIES
3. DATABASE SIDE TECHNOLOGIES

1b)
Aim: Write a HTML program, to explain the working of hyperlinks using tag and href, target Attributes.

Source code:

```
<!--working with anchor tag-->
<!doctype html>
<html>
    <head><title>navigation bar</title></head>
    <body>
        <div style="background-color: bisque;">
        <h3 align="center">ST.ANN'S COLLEGE OF ENGINEERING & TECHNOLOGY</h3>
        <a href="#" target="_blank">Home</a>
        <a href="1c.html" target="_blank">contact us</a>

    </div>
    <pre>
        HTML Links
        Links are found in nearly all web pages. Links allow users to click their way from page to
page.

        HTML Links - Hyperlinks
HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

Note: A link does not have to be text. A link can be an image or any other HTML element!
HTML Links - Syntax
The HTML <a> tag defines a hyperlink.

The most important attribute of the a element is the href attribute, which indicates the link's
destination.

The link text is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

HTML Links - The target Attribute
By default, the linked page will be displayed in the current browser window. To change this, you must
specify another target for the link.

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

_self - Default. Opens the document in the same window/tab as it was clicked
_blank - Opens the document in a new window or tab
_parent - Opens the document in the parent frame
_top - Opens the document in the full body of the window


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

Output:

ST.ANN'S COLLEGE OF ENGINEERING & TECHNOLOGY

Home contact us

HTML Links
Links are found in nearly all web pages. Links allow users to click their way from page to page.

HTML Links - Hyperlinks
HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

Note: A link does not have to be text. A link can be an image or any other HTML element!
HTML Links - Syntax
The HTML tag defines a hyperlink.

The most important attribute of the a element is the href attribute, which indicates the link's destination.

The link text is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

HTML Links - The target Attribute
By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

_self - Default. Opens the document in the same window/tab as it was clicked
_blank - Opens the document in a new window or tab
_parent - Opens the document in the parent frame
_top - Opens the document in the full body of the window

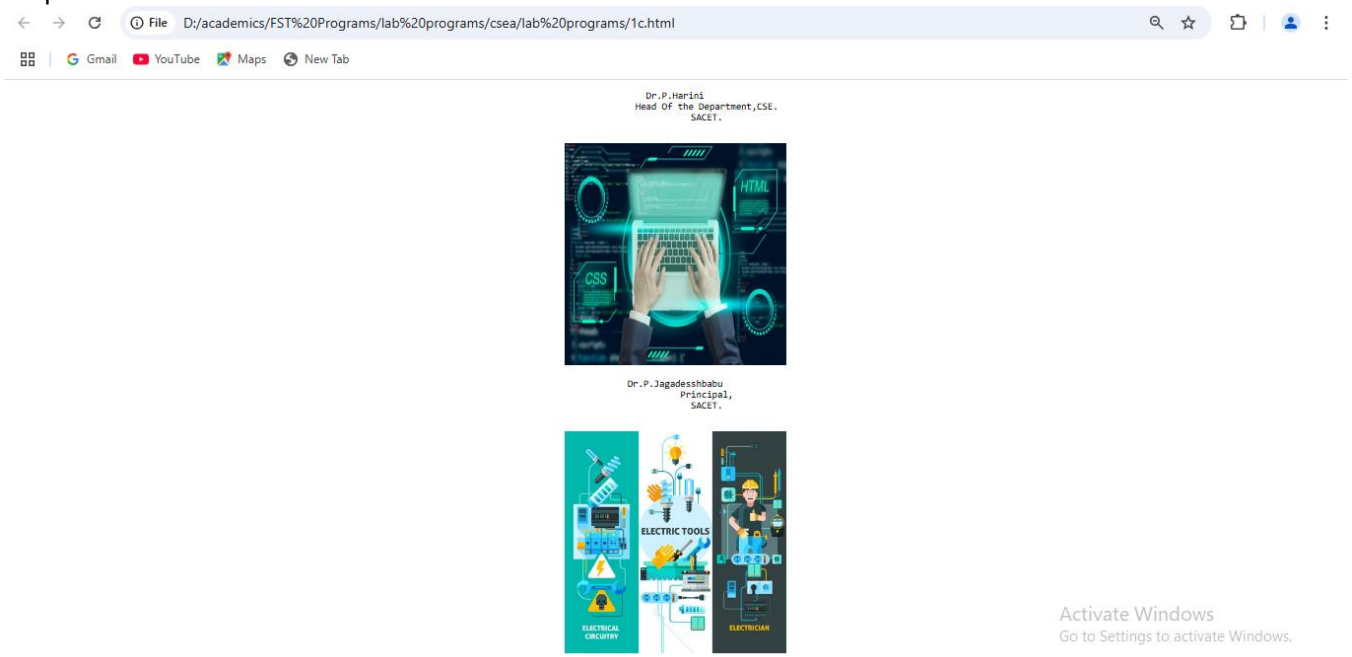
c.

Aim: Create a HTML document that has your image and your friend's image with a specific height and width. Also when clicked on the images it should navigate to their respective profiles.

Source code:

```
<!--create a html document that has your image and yours friends image with specific height and
width-->
<!doctype html>
<html>
    <head>
        <title>images</title>
    </head>
    <body align="center">
        <pre>Dr.P.Harini
            Head Of the Department,CSE.
            SACET.
        </pre>
        <a href="2a.html" target="_blank">
        <img src="csedept.jpg" alt="HOD" width="300" height="300">
    </a>
        <pre>Dr.P.Jagadesshbabu
            Principal,
            SACET.
        </pre>
        <a href="2a.html" target="_blank">
        <img src="eeedept.jpg" alt="Principal" width="300" height="300">
    </a>
    </body>
</html>
```

Output:



1d)

Aim:Write a HTML program, in such a way that, rather than placing large images on a page, the preferred technique is to use thumbnails by setting the height and width parameters to something like to 100*100 pixels. Each thumbnail image is also a link to a full sized version of the image. Create an image gallery using this technique.

source code:

```
<!-- Write a HTML program, in such a way that, rather than placing large images on a page,
the preferred technique is to use thumbnails by setting the height and width parameters
to something like to 100*100 pixels. -->
<!doctype html>
<html>
<head>
    <title>thumbnails</title>
</head>
<body>
<h3>gallery</h3>
<a href="hod.png" taget="_blank">
<img src="hod.png" alt="hod" widht="100" height="100">
</a>
<a href="mypic.jpg" taget="_blank">
<img src="mypic.jpg" alt="tss" widht="100" height="100">
</a>
<a href="hod.png" taget="_blank">
<img src="atp.png" alt="atp" widht="100" height="100">
</a>
<a href="hod.png" taget="_blank">
<img src="dnb.png" alt="dnb" widht="100" height="100">
</a>
<a href="hod.png" taget="_blank">
<img src="pvss.png" alt="pvss" widht="100" height="100">
</a>
<a href="hod.png" taget="_blank">
<img src="ramesh.jpg" alt="ramesh" widht="100" height="100">
</a>

</body>
</html>
```

output:

File D:/academics/FST%20Programs/lab%20programs/csea/lab%20programs/hod.png

Gmail YouTube Maps New Tab

File D:/academics/FST%20Programs/lab%20programs/csea/lab%20programs/mypic.jpg

Gmail YouTube Maps New Tab

Experiment:2
HTML Tables
2a)
Aim:write a html program to explain the working of tables using <table><tr>td><th> tags and attributes :border,rowspan and colspan.

Source code:

```
<!working with table,th,tr,td elements with cell spacing and cell padding
<!doctype html>
<html>
    <head></head>
    <style>
        table,th,td{
            border: 3px solid red;
            border-spacing: 20px;
        }
        th,td{
            padding:20px;
        }
    </style>
    <body>
        <table>
            <th>sino</th>
            <th>name of the student</th>
            <th>hallticket number</th>
            <th>phone number</th>
            <th>email id</th>
            <th>address</th>
            <tr>
                <td>1</td>
                <td>steve jobs</td>
                <td>566</td>
                <td>9885924188</td>
                <td>stevejobs@gmail.com</td>
                <td>kothapeta,chirala</td>
            </tr>
            <tr>
                <td>2</td>
                <td>steveharris</td>
                <td>567</td>
                <td>9885924189</td>
                <td>sacet623@gmail.com</td>
                <td>kothapet,chirala</td>
            </tr>
            <tr>
                <td>3</td>
                <td>jhon smith</td>
                <td>568</td>
                <td>9885924187</td>
                <td>jhonsmith2gmail.com</td>
                <td>iltd colony,chirala</td>
            </tr>
            <tr>
                <td>4</td>
                <td>petterpaul</td>
                <td>569</td>
                <td>9885924187</td>
```

```
                <td>petterpaul@gmail.com</td>
                <td>perala,chirala</td>
            </tr>
            <tr>
                <td>5</td>
                <td>richarson</td>
                <td>570</td>
                <td>9885924156</td>
                <td>ricjard@gmail.com</td>
                <td>perala,chirala</td>
            </tr>
        </table>
    </body>
</html>
```

Output:

sino	name of the student	hallticket number	phone number	email id	address
1	steve jobs	566	9885924188	stevejobs@gmail.com	kothapeta,chirala
2	steveharris	567	9885924189	sacet623@gmail.com	kothapet,chirala
3	jhon smith	568	9885924187	jhonsmith2gmail.com	iltd colony,chirala
4	petterpaul	569	9885924187	petterpaul@gmail.com	perala,chirala
5	richarson	570	9885924156	ricjard@gmail.com	perala,chirala

b. Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use tag to set the caption to the table & also use cell spacing, cell padding, border, rowspan, colspan etc.).

SOURCE CODE:

```
<!--working with table and colspan and rowspan-->
<!doctype html>
<html>
    <head></head>
    <style>
        th,td{
            padding:20px;
            border:2px solid red;
        }
    </style>
    <body>
        <table border="3" style="border-collapse: collapse;">
              <caption>TIMETABEL</caption>
            <th>day</th>
            <th>9.00-10.00</th>
            <th>10.00-11.00</th>
            <th>11.00-12.00</th>
            <th>12.00-12.50</th>
            <th>12.50-1.50</th>
            <th>1.50-2.50</th>
            <th>2.50-3.50</th>
            <tr>
                <td>mon</td>
                <td colspan="3">lab</td>
                <td rowspan="6">lunch</td>
                <td>os</td>
                <td>java</td>
                <td>msd</td>
            </tr>
            <tr>
                <td>tue</td>
                <td>cns</td>
                <td>os</td>
                <td>java</td>
                <td colspan="3">msd lab</td>
            </tr>
            <tr>
                <td>wed</td>
                <td colspan="3">lab</td>
                <td>os</td>
                <td>java</td>
                <td>msd</td>
            </tr>
            <tr>
                <td>thu</td>
                <td colspan="3">lab</td>

                <td>os</td>
                <td>java</td>
                <td>msd</td>
            </tr>

<tr>
```

```
            <td>fri</td>
            <td colspan="3">lab</td>

            <td>os</td>
            <td>java</td>
            <td>msd</td>
        </tr>
        <tr>
            <td>sat</td>
            <td colspan="3">lab</td>

            <td>os</td>
            <td>java</td>
            <td>msd</td>
        </tr>

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

OUTPUT:

TIMETABLE

day	9.00-10.00	10.00-11.00	11.00-12.00	12.00-12.50	12.50-1.50	1.50-2.50	2.50-3.50
mon	lab			lunch	os	java	msd
tue	cns	os	java		msd lab		
wed	lab				os	java	msd
thu	lab				os	java	msd
fri	lab				os	java	msd
sat	lab				os	java	msd

c. Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field, checkboxes,radio buttons,list boxes using and two buttons ie:submit and reset. Use tables to provide a better view).

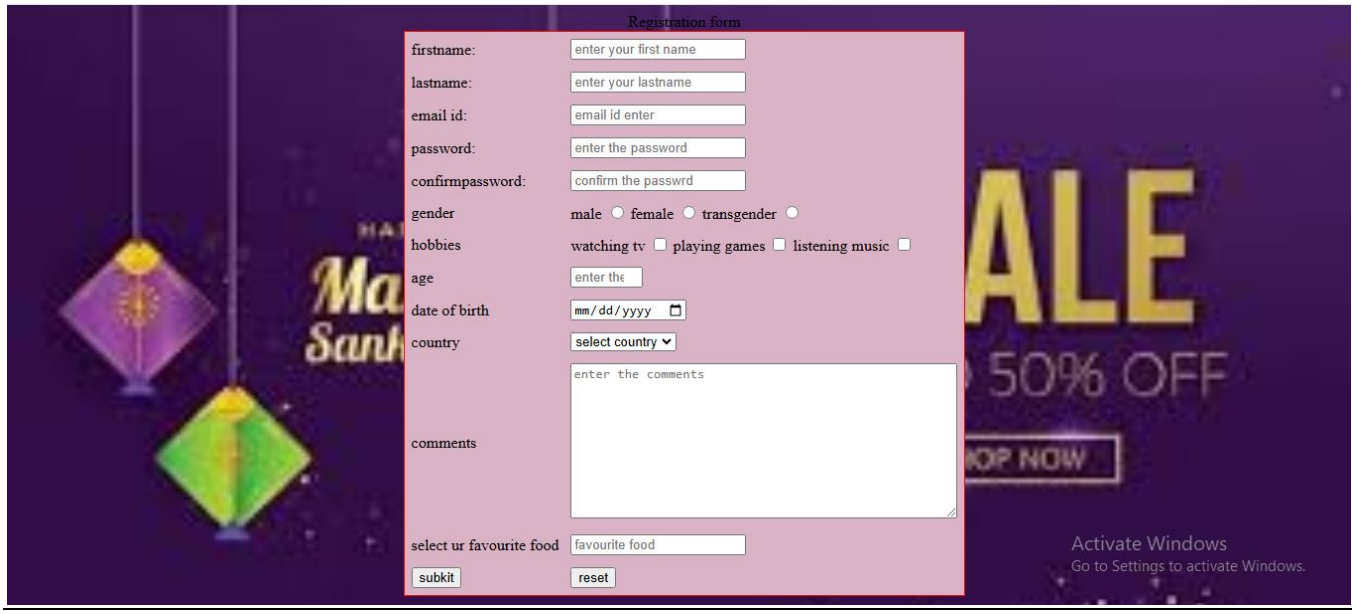
Source Code:

```
<!DOCTYPE html>
<html>
    <head></head>
    <style>
        table,th,td{
            padding:10px;
            border-spacing: 10px;
        }
        </style>
    <body  style="background-color: beige;">
        <form>
            <table>
                <caption>Registration Form</caption>
                <tr>
                    <td>
                    <label for="firstname">Enter first name</lable>
                </td>
                <td>
                    <input type="text" id="firstname" name="firstname" placeholder="enter firstname">
                </td>
                </tr>
            <tr>
                <td>
                    <label for="lastname">lastname</lable>
                </td>
                <td>
                    <input type="text" id="lastname" name="lastname" placeholder="eter lastname">
                </td>
            </tr>
            <tr>
                <td>
                    <label for="emailid">enter emailid</label>
                </td>
                <td>
                    <input type="email" id="email" name="email" placeholder="enter email id">
                </td>

            </tr>
            <tr>
                <td>
                    <lable for="password">new password</lable>
                </td>
                <td>
                    <input type="password" id="password" name="password" placeholder="set password">
                </td>
            </tr>
            <tr>
                <td>
                    <lable for="password">re-enter password</lable>
                </td>


<td>
```

```
                    <input type="password" id="password" name="password" placeholder="re-
typepassword">
                </td>
            </tr>
            <tr>
                <td>
                <lable for="age">enter age</lable>
            </td>
            <td>
                <input type="number" id="age" name="age" placeholder="select age" >
            </td>
            </tr>
            <tr>
                <td>
                <lable for="dob">date of birth</lable>
            </td>
            <td>
                <input type="date" id="dob" name="dob"  >
            </td>
            </tr>
            <tr>
                <td>
                <span>gender</span>
            </td>
            <td>
                <label for="male">male</label>
                <input type="radio" id="male" name="gender"  >
                <label for="female">female</label>
                <input type="radio" id="female" name="gender"  >
                <label for="transgender">transgender</label>
                <input type="radio" id="transgender" name="gender"  >
            </td>
            </tr>
            <tr>
                <td>
                <span>Hobbies</span>
            </td>
            <td>
                <label for="tv">watching tv</label>
                <input type="checkbox" id="tv" name="hobbies"  >
                <label for="games">playing games</label>
                <input type="checkbox" id="games" name="hobbies"  >
                <label for="reading">reading books</label>
                <input type="checkbox" id="reading" name="hobbies"  >

            </td>
            </tr>
            <td>
                <label for="country">country</label>
            </td>
            <td>
                <select id="country" name="country">
                    <option value="">seelct your country</option>
                    <option value="india">india</option>
                    <option value="russia">russia</option>
                    <option value="australia">australia</option>
                </select>            </td>
            <tr>
                <td>
```

```
                    <lable for="comments">comments</lable>
                </td>
                <td>
                    <textarea id="comments" name="comments" placeholder="enter comments"></textarea>
                </td>
            </tr>
            <tr>
                <td>
                    <span>favourite food</span>
                </td>
                <td>
                    <input list="favfood" placeholder="favourite food">
                    <datalist id="favfood">
                        <option value="idly">idly</option>
                        <option value="idly">idly</option>
                        <option value="idly">idly</option>
                    </datalist>
                </td>

            </tr>
        <tr>
            <td>
                <button>submit</button>

            </td>
            <td>
                <button>reset</button>
            </td>
        </tr>
     </table>
        </form>
    </body>
</html>
```

Output:



d. Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. hyperlink. And also◊ paragraph, third frame ◊ image, second frame ◊(Note: first frame make sure of using “no frame” attribute such that frames to be fixed).

Source code:

```
<!DOCTYPE html>
<html>
<head>
    <title>Frames Example</title>
</head>
<body>
    <iframe src="image.html" width="1250" height="2100" frameborder="0"></iframe>
    <iframe src="paragraph.html" width="1200" height="100" frameborder="0"></iframe>
    <iframe src="hyperlink.html" width="1200" height="100" frameborder="0"></iframe>

</body>
</html>
```

Image.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Image Frame</title>
</head>
<body>
    <img src="ajio1.PNG" alt="Example Image" style="width:100%; height:100%;">
    <img src="ajio2.PNG" alt="Example Image" style="width:100%; height:100%;">
    <img src="ajio3.PNG" alt="Example Image" style="width:100%; height:100%;">
    <img src="ajio4.PNG" alt="Example Image" style="width:100%; height:100%;">
    <img src="ajio5.PNG" alt="Example Image" style="width:100%; height:100%;">

</body>
</html>
```

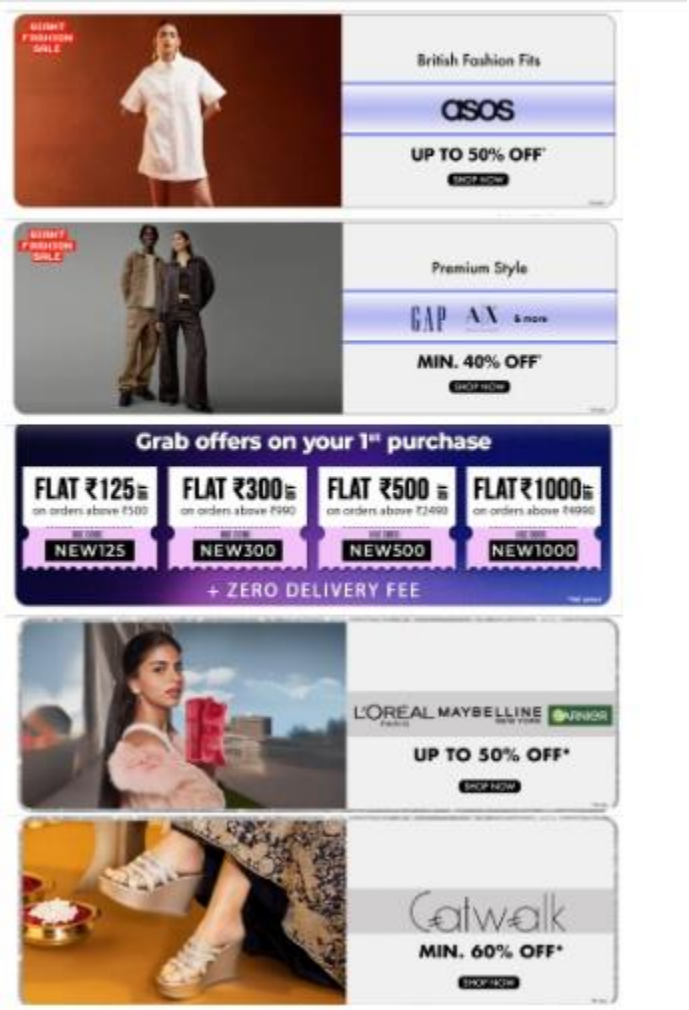
Paragraph.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Paragraph Frame</title>
</head>
<body>
    <p>This is an example paragraph to demonstrate the use of frames in HTML. Frames allow you to
divide the browser window into multiple sections, each capable of displaying a different
document.</p>
</body>
</html>
```

Hyperlink.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Hyperlink Frame</title>
</head>
<body>
    <a href="https://www.example.com" target="_blank">Visit Example.com</a>
</body>
</html>
```

Output:



This is an example paragraph to demonstrate the use of frames in HTML. Frames allow you to divide the browser window into multiple sections, each capable of displaying a different document.

Visit Example.com

Experiment:3

HTML 5 and Cascading Style Sheets, Types of CS

a.Write a HTML Program that makes use of <article>,<aside>,<figure>,<figcaption>,<footer>,<header>,<main>,

<nav>,<section>, , tags.

Souce code:

```
<!DOCTYPE html>
<html>
    <head></head>
    <body>
        <section style="text-align: center;">
            <header>
                <h1>baapare shopping</h1>
                <nav>
                    <a href="#" target="_blank">home</a>
                    <a href="#" target="_blank">login</a>
                    <a href="#" target="_blank">registration</a>
                </nav>
            </header>
        </section>
        <main>
            <article style="border-style: double;">
                <h3 style="text-align: center;">welcome to bapaare online shopping</h3>
                <aside style="float:left;width:150px;height:170px;">
                    <figure>
                    <img src="shopping.jpg" alt="welcome image" width="100" height="100">
                    <figcaption>welcome</figcaption>
                </figure>
                </aside>
                <p>We appreciate that you chose us for your shopping. We value your trust and
confidence in us.
                    We’re here to support you every step of the way. If you have any questions or
need assistance, don’t hesitate to reach out to our support team.
                    Welcome to the bapaare family, where we take great satisfaction in helping people
like you achieve needs through our Services.
                </p>
            </article>
            <div>
                <h4>AJIO is an online shopping platform  </h4>
                <span>that sells clothing, shoes, accessories, beauty products, home decor, and more.
It's a digital commerce initiative of Reliance Retail and is headquartered in Bangalore, India.
                </span>

            </div>
        </main>
        <br>


        <footer>
            <div>
               <span>bapaare<span>
               <nav>
                <a href="#" target="_blank">who we are</a>
                <br>
                <a href="#" target="_blank">join our team</a>
                <br>
                <a href="#" target="_blank">terms and conditions</a>
               </nav>
            </div>
        </footer>
    </body></html>
```

Output:

127.0.0.1:5500/3a.html

baapare shopping

home login registration

welcome to bapaare online shopping

welcome

We appreciate that you chose us for your shopping. We value your trust and confidence in us. We're here to support you every step of the way. If you have any questions or need assistance, don't hesitate to reach out to our support team. Welcome to the bapaare family, where we take great satisfaction in helping people like you achieve needs through our Services.

bapaare is an online shopping platform

that sells clothing, shoes, accessories, beauty products, home decor, and more. It's a digital commerce initiative of Reliance Retail and is headquartered in Bangalore, India.

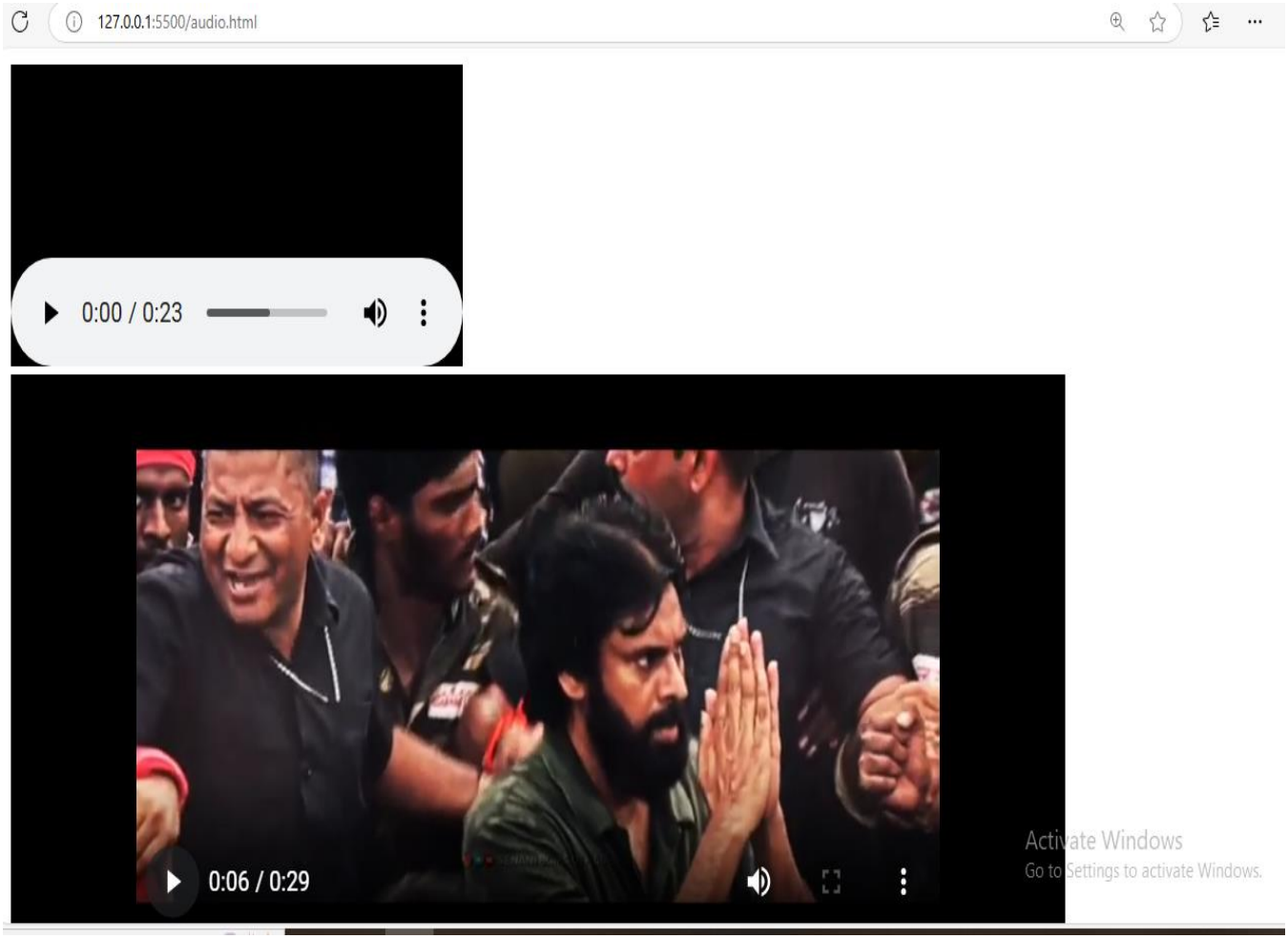
bapaare
who we are
join our team
terms and conditions

b. Write a HTML program, to embed audio and video into HTML web page.

Source code:

```
<!DOCTYPE html>
<html>
    <head></head>
    <body>
        <embed src="gabbar_singh.mp3">
            <br>
            <embed src="pspkmania.mp4" width="700" height="300">
    </body>
</html>
```

Output:



c. Write a program to apply different types (or levels of styles or style specification formats)- inline, internal, external styles to HTML elements. (identify selector, property and value).

Sourcecode:

```
<!DOCTYPE html>
<html>
    <head>
        <!-- external style -->
        <link rel="stylesheet" href="externalcss.css" type="text/css">
    </head>
    <!-- internal style -->
    <style>
        header{
            background-color: rgb(167, 112, 45);
        }
    </style>
    <body style="background-color: antiquewhite;">
        <section style="text-align: center;">
            <header>
                <!-- <!inline element-->
                <h1 style="color:gold">baapare shopping</h1>
                <nav>
                    <a href="#" target="_blank">home</a>
                    <a href="#" target="_blank">login</a>
                    <a href="#" target="_blank">registration</a>
                </nav>
            </header>
        </section>
        <main>
            <article style="border-style: double;">
                <h3 style="text-align: center;">welcome to bapaare online shopping</h3>
                <aside style="float:left;width:150px;height:170px;">
                    <figure>
                    <img src="shopping.jpg" alt="welcome image" width="50" height="50">
                    <figcaption>welcome</figcaption>
                </figure>
                </aside>
                <p>We appreciate that you chose us for your shopping. We value your trust and
confidence in us.
                    We’re here to support you every step of the way. If you have any questions or
need assistance, don’t hesitate to reach out to our support team.
                    Welcome to the bapaare family, where we take great satisfaction in helping people
like you achieve needs through our Services.
                </p>
            </article>
            <div>
                <h4>bapaare is an online shopping platform  </h4>
                <span>that sells clothing, shoes, accessories, beauty products, home decor, and more.
It's a digital commerce initiative of Reliance Retail and is headquartered in Bangalore, India.
                </span>

            </div>
        </main>
        <br>
        <footer>
            <div>
               <span>bapaare<span>
               <nav>
                <a href="#" target="_blank">who we are</a>
                <br>
                <a href="#" target="_blank">join our team</a>
                <br>
                <a href="#" target="_blank">terms and conditions</a>
               </nav>
            </div>
```

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

externalcss.css

```
a{
    color:black;
}
a:hover{
    color:white;
}
article{
    background-color: burlywood;
}
h1{
    color:red;
    font-family: cursive;
}
h3{
    color:green;
}
```

Output:

Experiment:4 Selector forms

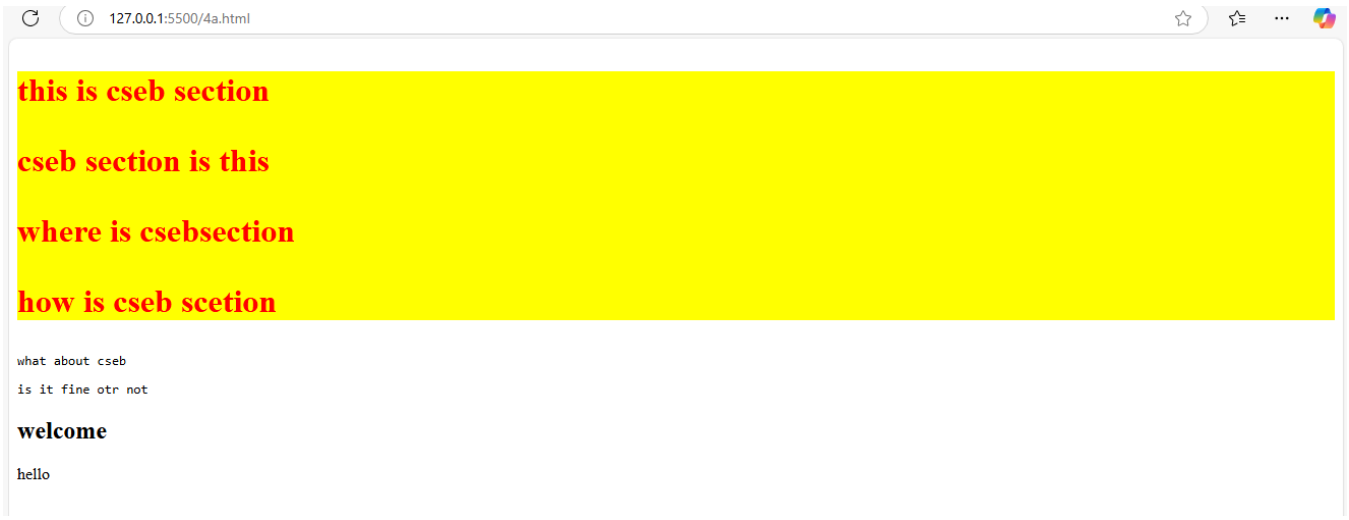
4a. Write a program to apply simple selector forms

i.using element selector

Source code:

```
<!-- write a html program to apply selector forms -->
<!doctype html>
<html>
    <haed></haed>
    <style>
           h1{
        color:red;
        background-color: yellow;;
    }

    </style>
    <body>
        <h1 id="parag">
            <p class="pa">this is cseb section</p>
            <p>cseb section is this</p>
            <p >where is csebsection</p>
            <p>how is cseb scetion</p>

        </h1>
        <pre >what about cseb</pre>
        <pre >is it fine otr not</pre>
        <h2>welcome</h2>
        <p class="pa" >hello</p>

    </body>
</html>
```

Output:

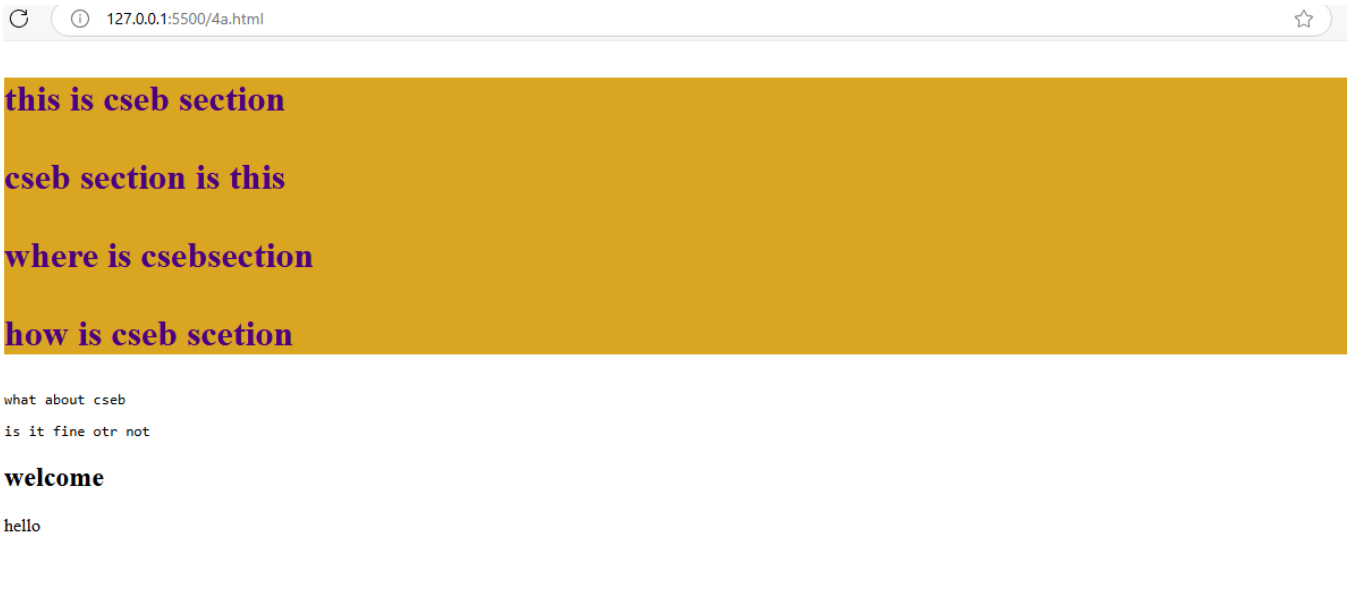


ii. id selector

Source code:

```
<!-- write a html program to apply selector forms -->
<!doctype html>
<html>
    <haed></haed>
    <style>

     /* id element style */
#parag{
        color:indigo;
        background-color: goldenrod;
     }


    </style>
    <body>
        <h1 id="parag">
            <p class="pa">this is cseb section</p>
            <p>cseb section is this</p>
            <p >where is csebsection</p>
            <p>how is cseb scetion</p>

        </h1>
        <pre >what about cseb</pre>
        <pre >is it fine otr not</pre>
        <h2>welcome</h2>
        <p class="pa" >hello</p>

    </body>
</html>
```

Output:

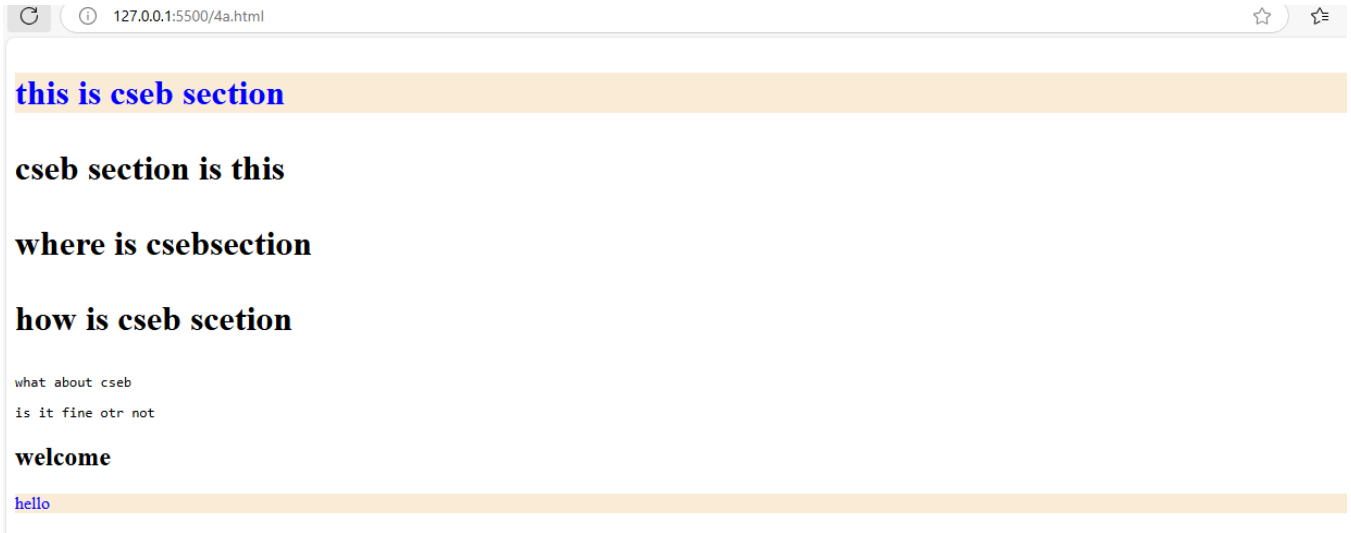


iii. using class selector

Source code:

```
<!-- write a html program to apply selector forms -->
<!doctype html>
<html>
    <haed></haed>
    <style>
     /* class element style */
     .pa{

        color:blue;
        background-color: antiquewhite;
     }
    </style>
    <body>
        <h1 id="parag">
            <p class="pa">this is cseb section</p>
            <p>cseb section is this</p>
            <p >where is csebsection</p>
            <p>how is cseb scetion</p>

        </h1>
        <pre >what about cseb</pre>
        <pre >is it fine otr not</pre>
        <h2>welcome</h2>
        <p class="pa" >hello</p>

    </body>
</html>
```

Output:



iv. using group selector

Source code:

```
<!-- write a html program to apply selector forms -->
<!doctype html>
<html>
    <haed></haed>
    <style>
/* group selector */
     h2,pre{

        color:aqua;
        background-color: black;
     }

    </style>
    <body>
        <h1 id="parag">
            <p class="pa">this is cseb section</p>
            <p>cseb section is this</p>
            <p >where is csebsection</p>
            <p>how is cseb scetion</p>

        </h1>
        <pre >what about cseb</pre>
        <pre >is it fine otr not</pre>
        <h2>welcome</h2>
        <p class="pa" >hello</p>

    </body>
</html>
```

Output:

v.using universal selector

Source code:

```
<!-- write a html program to apply selector forms -->
<!doctype html>
<html>
    <haed></haed>
    <style>
/* universal selector */
     *{
        color:green;
        background-color: chocolate;
     }

    </style>
    <body>
        <h1 id="parag">
            <p class="pa">this is cseb section</p>
            <p>cseb section is this</p>
            <p >where is csebsection</p>
            <p>how is cseb scetion</p>

        </h1>
        <pre >what about cseb</pre>
        <pre >is it fine otr not</pre>
        <h2>welcome</h2>
        <p class="pa" >hello</p>

    </body>
</html>
```

Output:

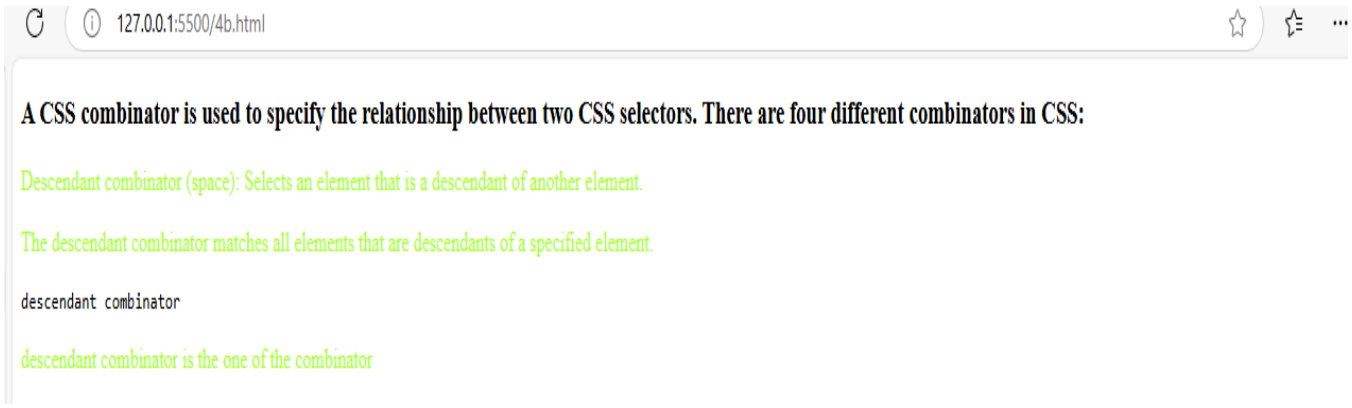
4b. Write a program to apply Combinator selector

i.using Descendant selector

source code:

```
<!DOCTYPE html>
<html>
    <head></head>
     <style>

        /* descending style */
        /* applyied color to the all the childs of the parent */
      div p{
        color:greenyellow;
      }
       </style>
    <body>
       <div>

            <h3>A CSS combinator is used to specify the relationship between two CSS selectors. There are four different combinators in CSS:</h3>
            <p>Descendant combinator (space): Selects an element that is a descendant of another element.</p>
            <p>The descendant combinator matches all elements that are descendants of a specified element.</p>
            <section>
                <pre>descendant combinator</pre>
                <p>descendant combinator is the one of the combinator</p>
            </section>
        </div>
    </body>
</html>
```

Output:

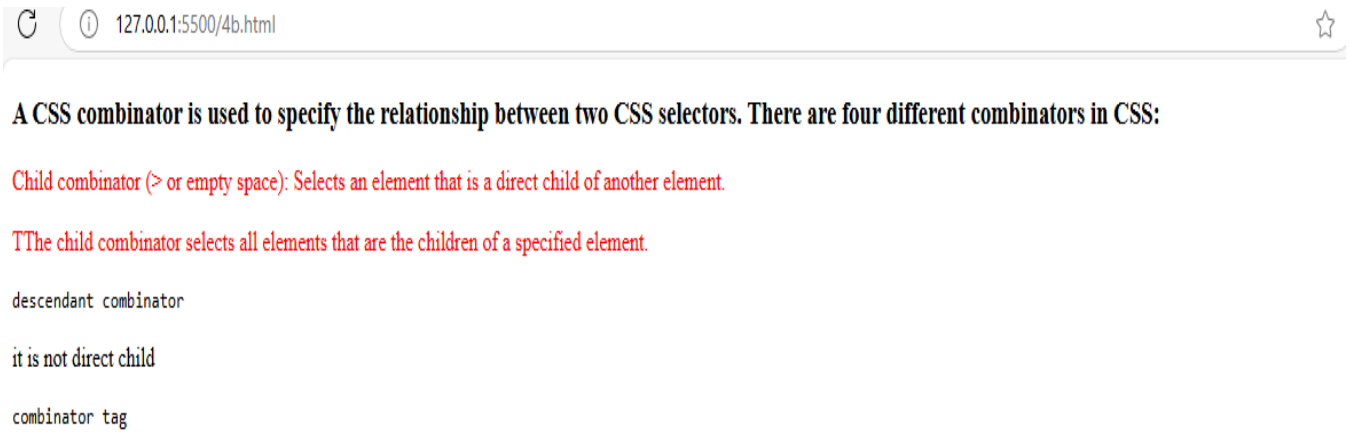


ii.using child selector

source code:

```
<!DOCTYPE html>
<html>
    <head></head>
     <style>

       /* child seelctor forms */

      /* apply styles to dirct childerens of the parent */
      div>p{
        color:red;
      }

       </style>
    <body>
       <div>

            <h3>A CSS combinator is used to specify the relationship between two CSS selectors. There are four different combinators in CSS:</h3>
            <p>Child combinator (> or empty space): Selects an element that is a direct child of another element.</p>
            <p>TThe child combinator selects all elements that are the children of a specified element.</p>
            <section>
                <pre>descendant combinator</pre>
                <p>it is not direct child</p>
            </section>
        </div>
    </body>

</html>
```

Output:

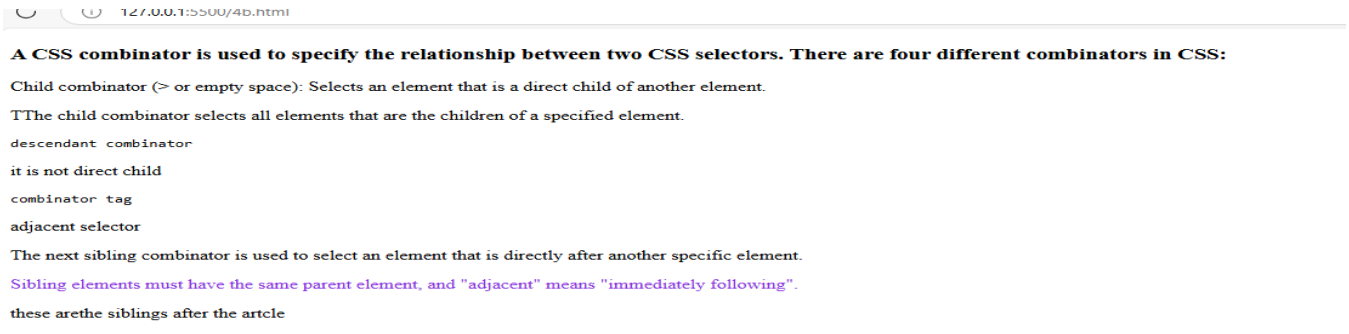


iii.using adjacent sibling selector

source code:

```
<!DOCTYPE html>
<html>
    <head></head>
     <style>

       /* adjacent sibling selector

      direct first nextsibling of child sto apply */
      article+p{
        color:blueviolet;
      }
       </style>
    <body>
       <div>

            <h3>A CSS combinator is used to specify the relationship between two CSS selectors. There are four different combinators in CSS:</h3>
            <p>Child combinator (> or empty space): Selects an element that is a direct child of another element.</p>
            <p>TThe child combinator selects all elements that are the children of a specified element.</p>
            <section>
                <pre>descendant combinator</pre>
                <p>it is not direct child</p>
            </section>
        </div>
        <pre>combinator tag</pre>

        <article>
            <p>adjacent selector</p>
            <p>The next sibling combinator is used to select an element that is directly after another specific element.</p>
        </article>
        <p>Sibling elements must have the same parent element, and "adjacent" means "immediately following".</p>
        <p>these arethe siblings after the artcle</p>
    </body>

</html>
```

Output:

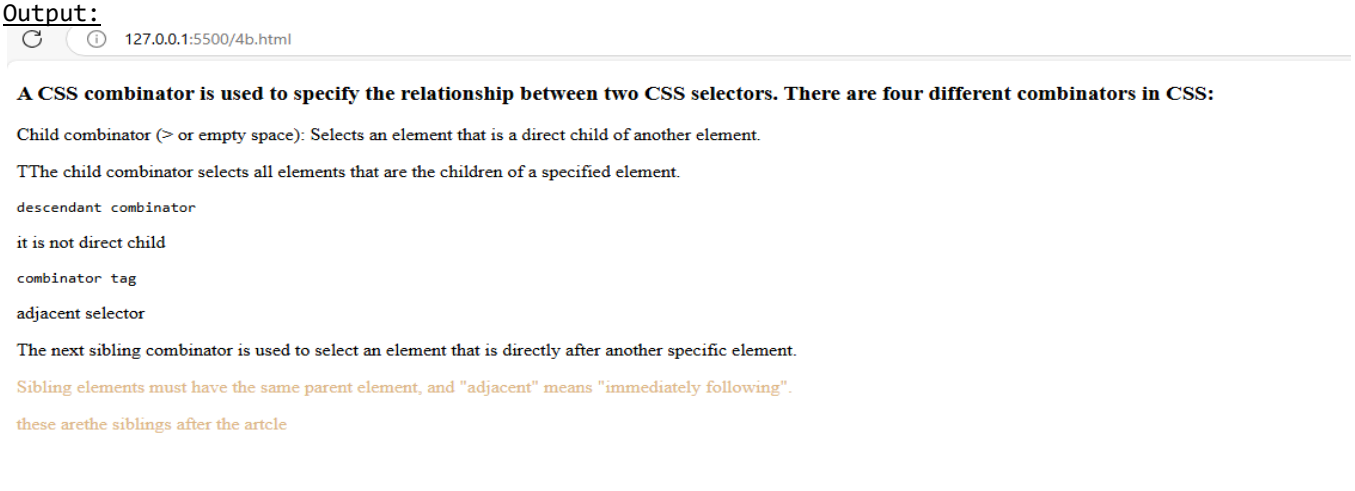


iv.using general sibling selector

source code:

```
<!DOCTYPE html>
<html>
    <head></head>
<style>
/* general sibling selector
      using tilled sysmbol and styles applied to all sibling after parent element */
      article~p{
        color:burlywood;
      }
</style>
<body>


        <div>
            <h3>A CSS combinator is used to specify the relationship between two CSS selectors.
There are four different combinators in CSS:</h3>
            <p>Child combinator (> or empty space): Selects an element that is a direct child of
another element.</p>
            <p>TThe child combinator selects all elements that are the children of a specified
element.</p>
            <section>
                <pre>descendant combinator</pre>
                <p>it is not direct child</p>
            </section>
        </div>
        <pre>combinator tag</pre>

        <article>
            <p>adjacent selector</p>
            <p>The next sibling combinator is used to select an element that is directly after
another specific element.</p>
        </article>
        <p>Sibling elements must have the same parent element, and "adjacent" means "immediately
following".</p>
        <p>these arethe siblings after the artcle</p>
        </body>
</html>
```

Output:

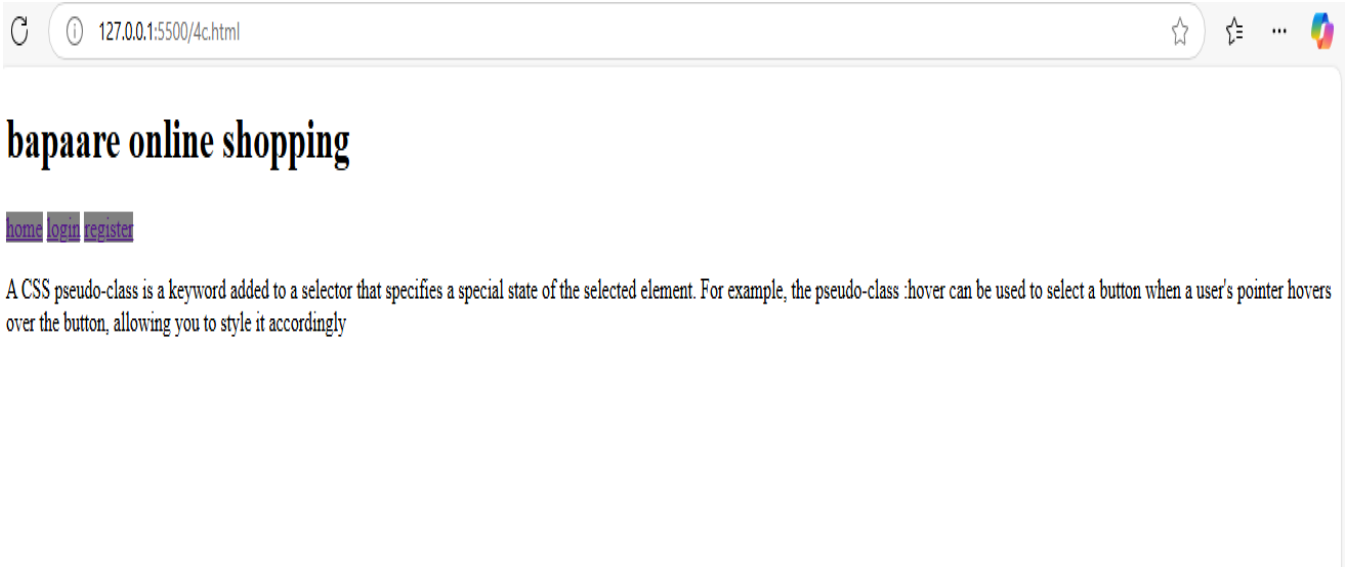


4c. Write a program to apply pseudo class selector

Source code:

```
<!doctype html>
 <html>
    <head></head>
    <style>
    a:link{
        color:brown;
        background-color: grey;
    }
    a:hover{
        color:white;
        background-color: green;
    }

    </style>
    <body>
        <h1>bapaare online shopping</h1>
        <a href="#" target="_parent">home</a>
        <a href="#" target="_parent">login</a>
        <a href="#" target="_parent">register</a>


        <p>A CSS pseudo-class is a keyword added to a selector that specifies a special state of the selected element. For example, the pseudo-class :hover can be used to select a button when a user's pointer hovers over the button, allowing you to style it accordingly</p>
    </body>
 </html>
```

Output:

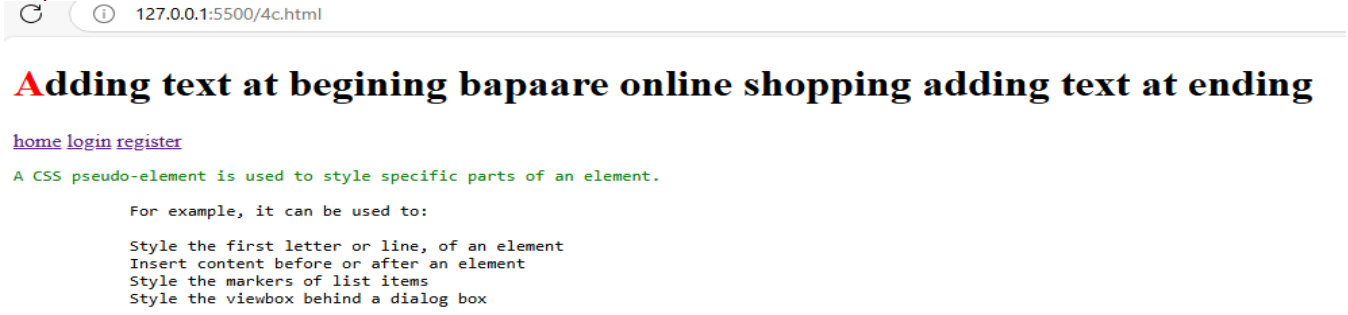


4d. Write a program to apply pseudo element selector

Source code:

```
<!doctype html>
<html>
   <head></head>
    <style>
   h1::first-letter{
       color:red;
       text-transform: uppercase;
   }
   pre::first-line{
       color:green;
   }
   h1::before{
      content:" adding text at begining ";
   }
   h1::after{
      content:" adding text at ending ";
   }
   p::selection{
       background-color:grey;
       color:white;
   }

   </style>
   <body>
       <h1>bapaare online shopping</h1>
       <a href="#" target="_parent">home</a>
       <a href="#" target="_parent">login</a>
       <a href="#" target="_parent">register</a>

       <pre>A CSS pseudo-element is used to style specific parts of an element.

           For example, it can be used to:

           Style the first letter or line, of an element
           Insert content before or after an element
           Style the markers of list items
           Style the viewbox behind a dialog box
           </pre>
   </body>
</html>
```

Output:



4e. Write a program to apply attribute selector

Source code:

```
<!doctype html>
<html>
   <head></head>
   <style>
       /* select all title attribute */
       [title]{
           border:5px solid yellow;
       }
       /* select exact value */
       [title='sacet cse dept']
       {
           color:white;
           background-color: aqua;
       }
       /* select any where matches the word */
       [title~="sacet"]
       {
           color:yellow;
           background-color: black;;
       }
       /* select when word comes at starting */
       [title^="sacp"]{
           color:green;
           background-color: blanchedalmond;
       }
       /* select word at end */
[title$="sacet"]{
    color:red;
    background-color: aqua;
}
    </style>
    <body>
        <img src="#" title="sacet cse dept" width="200" height="200">
        <br>
        <br>
        <img src="#" title="sacet ece dept" width="200" height="200">
        <img src="#" title="sacet eee dept" width="200" height="200">
        <img src="#" title="sacet mech dept" width="200" height="200">
        <img src="#" title="mech dept sacet" width="200" height="200">
        <img src="#" title="sacp pharma dept" width="200" height="200">
        <img src="#" title="sacp analysis dept" width="200" height="200">
    </body>
</html>
```

Output:


127.0.0.1:5500/4d.html
sacet cse dept
sacet ece dept
sacet eee dept
sacet mech dept
mech dept sacet
sacp pharma dept
sacp analysis dept

Experiment:5 CSS with Color, Background, Font, Text and CSS Box Model

5a. Write a program to demonstrate the various ways you can reference a color in CSS

Source code:

```
<!doctype html>
<html>
    <head></head>
    <style>
        #colorname{
            color:indianred;
            background-color: aqua;

        }
        #colorhex{
            color:#ffffff;
            background-color: #c24b90;
        }
        #colorrgb{
            color:rgb(10, 23, 50);
            background-color: rgb(55, 107, 107);
        }
        #colorrgba{
            color:rgb(29, 220, 39,1.0);
            background-color: rgb(242, 7, 117);
        }
        #colorothers{
            color:hwb(0 36% 20%);
            background-color: aqua;
        }

    </style>
    <body>
        <p id="colorname">to apply the colors in css we can use by their colorname</p>
        <p id="colorhex">to apply the colors to the content you can use hex decimal values</p>
        <p id="colorrgb">to aplly the colors to the content you can use the rgb values</p>
        <p id="colorrgba">to apply the colors to the content through the rgba</p>
        <p id="colorothers">to apply the coltrs to the content in otherways</p>
    </body>
</html>
```

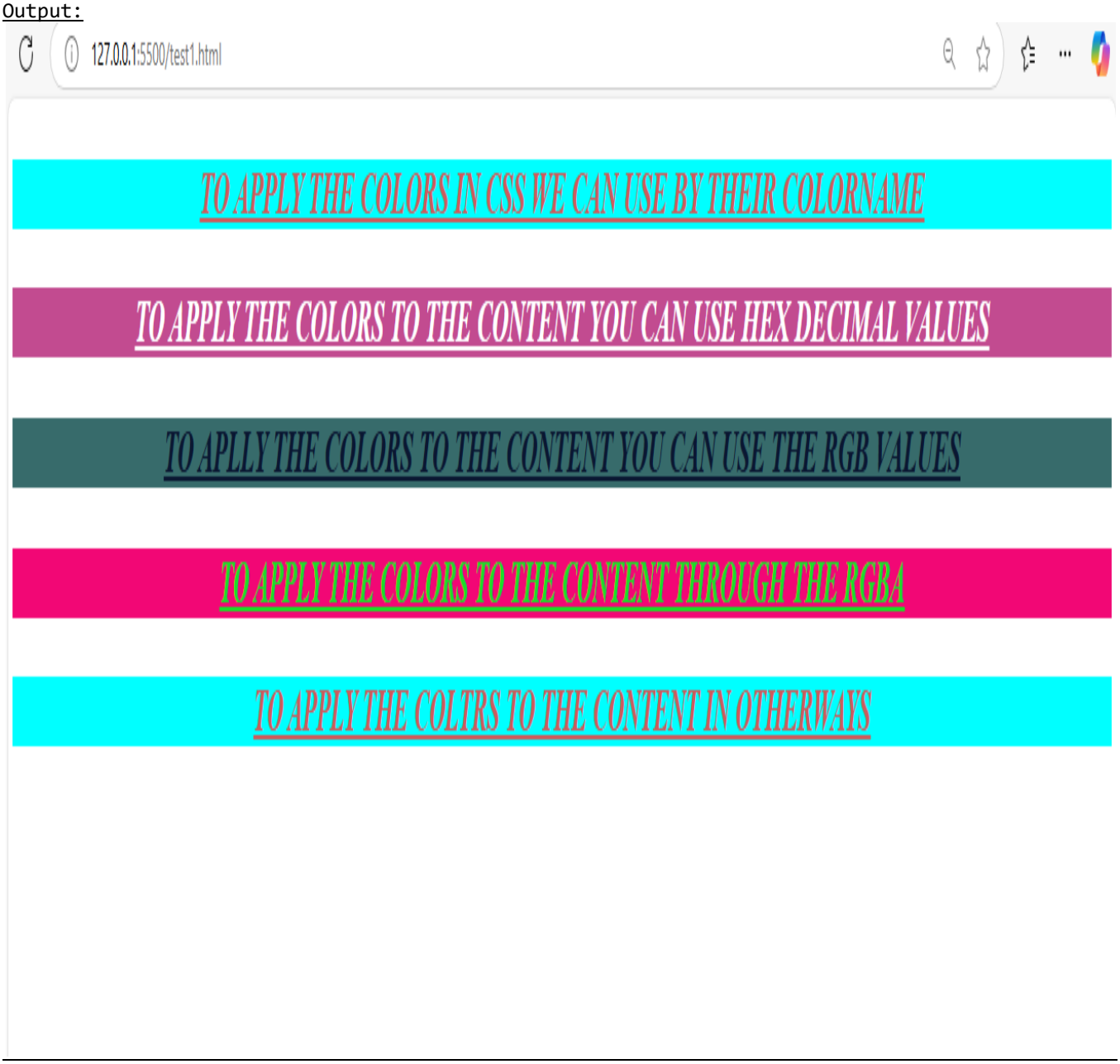
Output:

c. Write a program using the following terms related to CSS font and text: i. font-size ii. font-weight iii. font-style iv. text-decoration v. text-transformation vi. text-alignment

source code:

```
<!doctype html>
<html>
    <head></head>
    <style>
        #colorname{
            color:indianred;
            background-color: aqua;

        }
        #colorhex{
            color:#ffffff;
            background-color: #c24b90;
        }
        #colorrgb{
            color:rgb(10, 23, 50);
            background-color: rgb(55, 107, 107);
        }
        #colorrgba{
            color:rgb(29, 220, 39,1.0);
            background-color: rgb(242, 7, 117);
        }
        #colorothers{
            color:hwb(0 36% 20%);
            background-color: aqua;
        }
         p{

            font-size: 40px;
            font-weight: bold;
            font-style:italic;
            text-decoration: underline;
            text-transform: uppercase;
            text-align: center;
        }
    </style>
    <body>
        <p id="colorname">to apply the colors in css we can use by their colorname</p>
        <p id="colorhex">to apply the colors to the content you can use hex decimal values</p>
        <p id="colorrgb">to aplly the colors to the content you can use the rgb values</p>
        <p id="colorrgba">to apply the colors to the content through the rgba</p>
        <p id="colorothers">to apply the coltrs to the content in otherways</p>
    </body>
</html>
```

Output:


127.0.0.1:5500/test1.html
TO APPLY THE COLORS IN CSS WE CAN USE BY THEIR COLORNAME
TO APPLY THE COLORS TO THE CONTENT YOU CAN USE HEX DECIMAL VALUES
TO APLLY THE COLORS TO THE CONTENT YOU CAN USE THE RGB VALUES
TO APPLY THE COLORS TO THE CONTENT THROUGH THE RGBA
TO APPLY THE COLTRS TO THE CONTENT IN OTHERWAYS

d. Write a program, to explain the importance of CSS Box model using
i. Content ii. Border iii. Margin iv. Padding

source code:

```
<!doctype html>
<html>
    <head></head>
    <style>
        #colorname{
            color:indianred;
            background-color: aqua;
        }
        #colorhex{
            color:#ffffff;
            background-color: #c24b90;
        }
        #colorrgb{
            color:rgb(10, 23, 50);
            background-color: rgb(55, 107, 107);
        }
        #colorrgba{
            color:rgb(29, 220, 39,1.0);
            background-color: rgb(242, 7, 117);
        }
        #colorothers{
            color:hwb(0 36% 20%);
            background-color: aqua;
        }
         p{
            padding:20px;
            margin:100px;
            border:2px solid red;
        }
    </style>
    <body>
        <p id="colorname">to apply the colors in css we can use by their colorname</p>
        <p id="colorhex">to apply the colors to the content you can use hex decimal values</p>
        <p id="colorrgb">to aplly the colors to the content you can use the rgb values</p>
        <p id="colorrgba">to apply the colors to the content through the rgba</p>
        <p id="colorothers">to apply the coltrs to the content in otherways</p>
    </body>
</html>
```

Output:

b. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.

Source code:

```
<!doctype html>
<html>
    <head></head>
    <style>
        #colorname{
            color:indianred;
            background-color: aqua;
        }
        #colorhex{
            color:#ffffff;
            background-color: #c24b90;
        }
        #colorrgb{
            color:rgb(10, 23, 50);
            background-color: rgb(55, 107, 107);
        }
        #colorrgba{
            color:rgb(29, 220, 39,1.0);
            background-color: rgb(242, 7, 117);
        }
        #colorothers{
            color:hwb(0 36% 20%);
            background-color: aqua;
        }
         p{
            padding:20px;
            margin:100px;
            border:2px solid red;
            font-size: 40px;
            font-weight: bold;
            font-style:italic;
            text-decoration: underline;
            text-transform: uppercase;
            text-align: center;
        }
        body{
            background-image: url('gmg.jpg');
            background-repeat: no-repeat;
            background-size: cover;
            background-position: center 50%;
            background-attachment: fixed;
            transform: rotate(30deg);
        }
    </style>
    <body>
        <p id="colorname">to apply the colors in css we can use by their colorname</p>
        <p id="colorhex">to apply the colors to the content you can use hex decimal values</p>
        <p id="colorrgb">to aplly the colors to the content you can use the rgb values</p>
        <p id="colorrgba">to apply the colors to the content through the rgba</p>
        <p id="colorothers">to apply the coltrs to the content in otherways</p>
    </body></html>
```

Output:


127.0.0.1:5500/test1.html
PPLY THE COLORS IN CSS WE CAN USE BY THE
COLORNAME
TO APPLY THE COLORS TO THE CONTENT YOU
USE HEX DECIMAL VALUES
TO APLLY THE COLORS TO
USE THE
TO AP
Activate Windows
Go to Settings to activate Windows.

Javascript:

- JavaScript is a **scripting or programming language** that enables you to create dynamically updating content, control multimedia, animate images, and much more on web pages.
- It is an essential part of web development, standard web technologies, alongside HTML and CSS.
- Html is to define the content or structure of the web pages.
- Css is to specify the styles or layouts of the web pages.
- Java script is to provide the behavior of the web pages.
- Web browsers have a dedicated JavaScript engine that executes the client code..
- The most popular runtime system for non-browser usage is Node.js.

Key Features

- **Dynamic Content Updates**: JavaScript can update and change both HTML and CSS, allowing for interactive and responsive web pages
- **Event Handling**: It can respond to user actions like button clicks, form submissions, and other events
- **APIs**: JavaScript can interact with various APIs to perform tasks like retrieving geographical information, creating 2D/3D graphics, and handling multimedia.
- **Adding interactive to websites**
- **Developing mobile applications**
- **Create web browser based games.**
- **Backend web development.**

Experiment:6

Applying JavaScript - internal and external, I/O, Type Conversion

a. Write a program to embed internal and external JavaScript in a web page.

- Internal and External JavaScript are the two ways of adding JavaScript code to an HTML document.
- External JavaScript refers to adding JavaScript code in HTML from a separate .js file using the src attribute of <script> tag.
- Internal JavaScript refers to embedding JavaScript code directly within the HTML file using <script> tag, either inside the <head> or <body> tag. This method is useful for small scripts specific to a single page.

Source code:

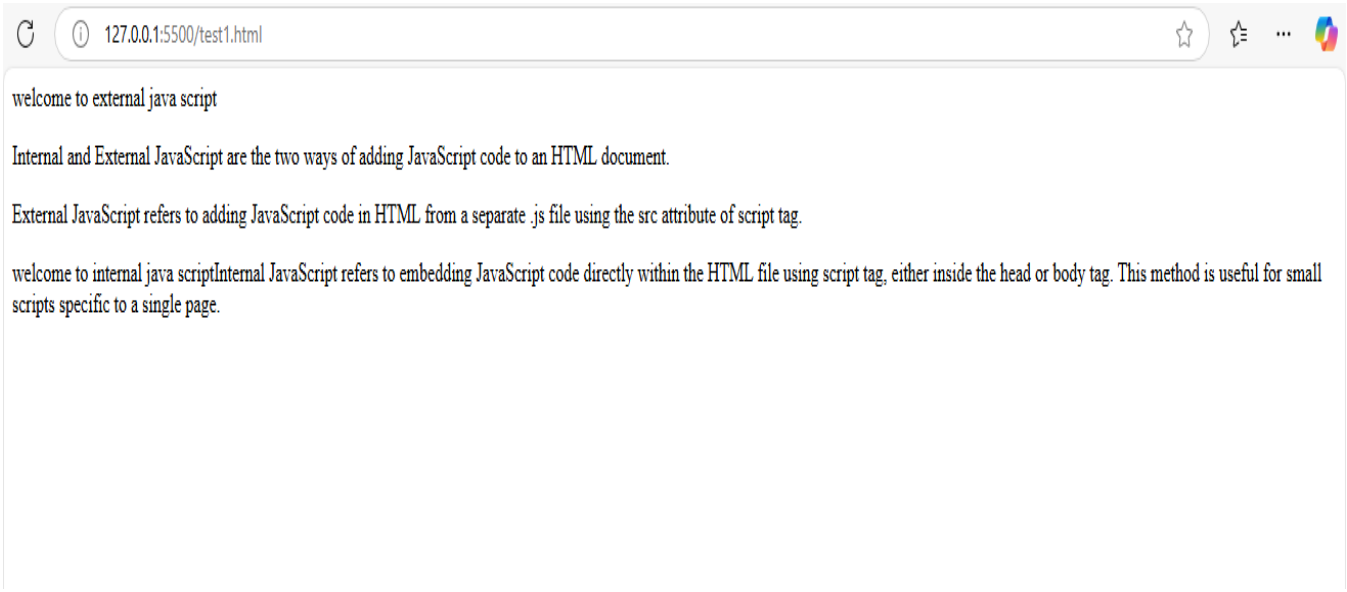
```
<!DOCTYPE html>
<html>
    <head>
        <title>java script intro</title>
    </head>
//external javascript
    <script src="samplejs.js"></script>
    <body>
        <p>
            Internal and External JavaScript are the two ways of adding JavaScript code to an
HTML document.
        </p>
        <p>
            External JavaScript refers to adding JavaScript code in HTML from a separate .js
file using the src attribute of script tag.
        </p>
```

```
//internal javascript
        <script>
            document.write("welcome to internal java script");
            document.write("Internal JavaScript refers to embedding JavaScript code directly
within the HTML file using script tag, either inside the head or body tag. This method is useful
for small scripts specific to a single page.");
        </script>

    </body>
</html>
```

Sample.js:

```
document.write("welcome to external java script");
```

output:



b. Write a program to explain the different ways for displaying output.

- There are 4 ways to display the output in JavaScript.
- Displaying the output in html elements using innerhtml attribute.
- Displaying the output using document.write().
- Displaying the output through console.log().
- Displaying the output through the alert box.

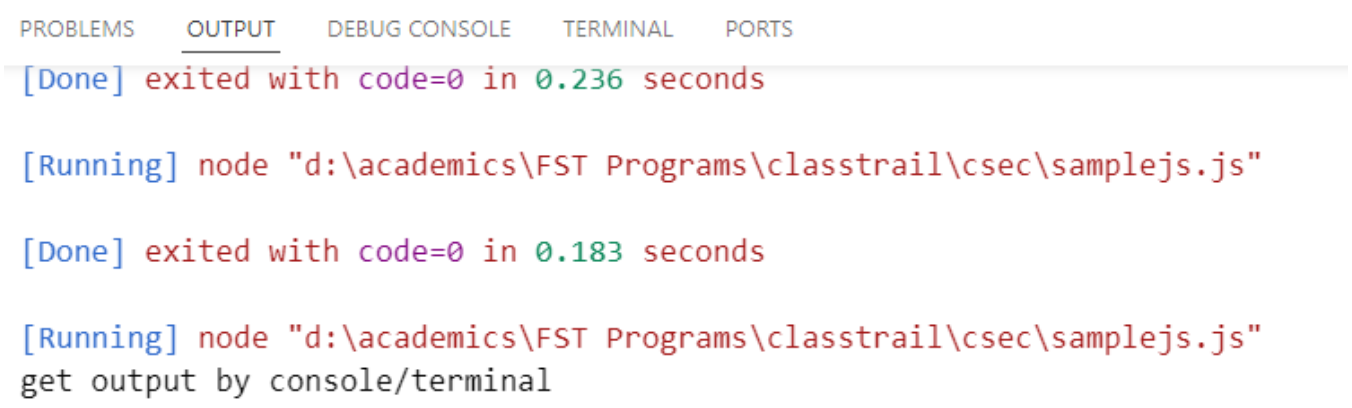
Source code:

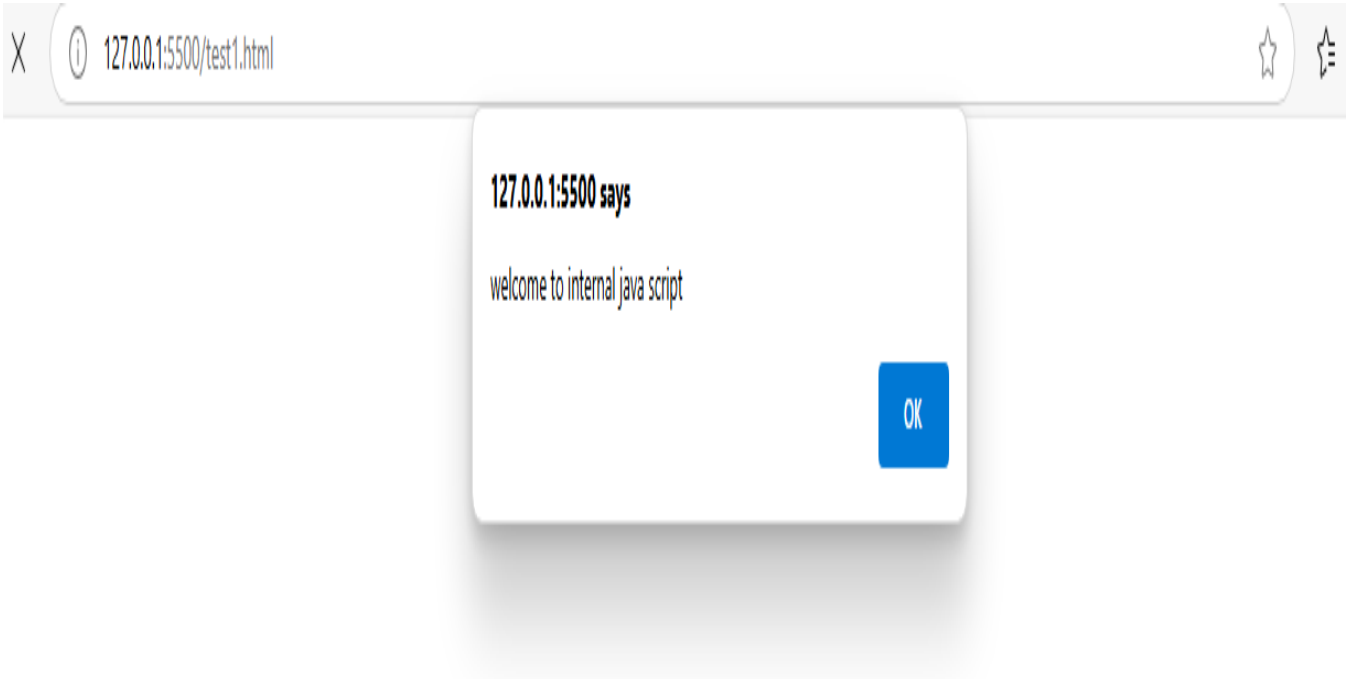
```
<!DOCTYPE html>
<html>
    <head>
        <title>java script intro</title>
    </head>

    <body>
       <p>There are 4 ways to display the output in JavaScript.</p>
       <ol>
        <li>Displaying the output in html elements using innerhtml attribute.</li>
        <li>Displaying the output using document.write().</li>
        <li>Displaying the output using console.log().</li>
        <li>Displaying the output using alert box.</li>
       </ol>
       <p id="display"></p>
        <script>
            document.getElementById("display").innerHTML = "display output by element";
            alert("welcome to internal java script");
            document.write("Internal JavaScript refers to embedding JavaScript code directly
within the HTML file using script tag, either inside the head or body tag. This method is useful
for small scripts specific to a single page.");
        </script>

    </body>
</html>
```

Console.js

```
console.log("get output by console/terminal");
```

output:





There are 4 ways to display the output in JavaScript.

1. Displaying the output in html elements using innerhtml attribute.
2. Displaying the output using document.write().
3. Displaying the output using console.log().
4. Displaying the output using alert box.

display output by element

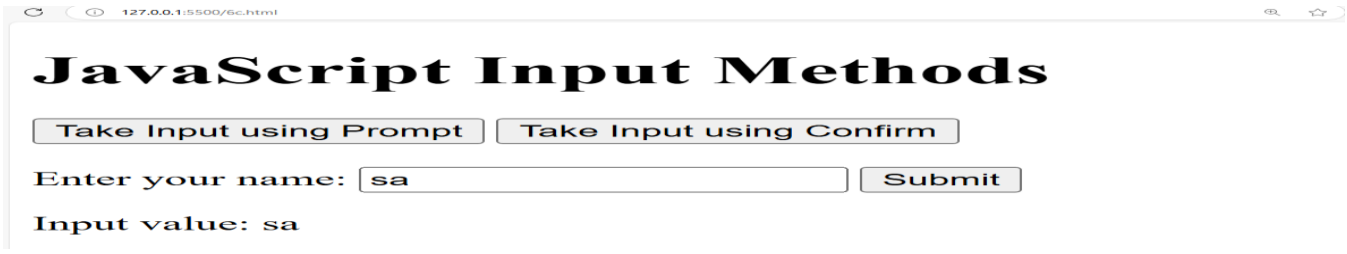
Internal JavaScript refers to embedding JavaScript code directly within the HTML file using script tag, either inside the head or body tag. This method is useful for small scripts specific to a single page.

c. Write a program to explain the different ways for taking input.

Source code:

```
<!DOCTYPE html>

<head>
       <title>JavaScript Input Methods</title>
</head>
<body>

    <h1>JavaScript Input Methods</h1>

    <button onclick="usePrompt()">Take Input using Prompt</button>
    <button onclick="useConfirm()">Take Input using Confirm</button>

    <br><br>

    <!-- Input field -->
    <label for="userInput">Enter your name: </label>
    <input type="text" id="userInput" placeholder="Type something">
    <button onclick="getInputValue()">Submit</button>

    <p id="output"></p>

    <script>
        // 1. Taking input using prompt()
        function usePrompt() {
            let name = prompt("Enter your name:");
            if (name) {
                document.getElementById("output").innerHTML = "You entered: " + name;
            } else {
                document.getElementById("output").innerHTML = "You did not enter anything.";
            }
        }

        // 2. Taking input using an HTML input field
        function getInputValue() {
            let inputValue = document.getElementById("userInput").value;
            document.getElementById("output").innerHTML = "Input value: " + inputValue;
        }

        // 3. Taking input using confirm()
        function useConfirm() {
            let response = confirm("Do you like JavaScript?");
            if (response) {
                document.getElementById("output").innerHTML = "You clicked OK! ";
            } else {
                document.getElementById("output").innerHTML = "You clicked Cancel. ";
            }
        }
    </script>

</body>
</html>
```

Output:



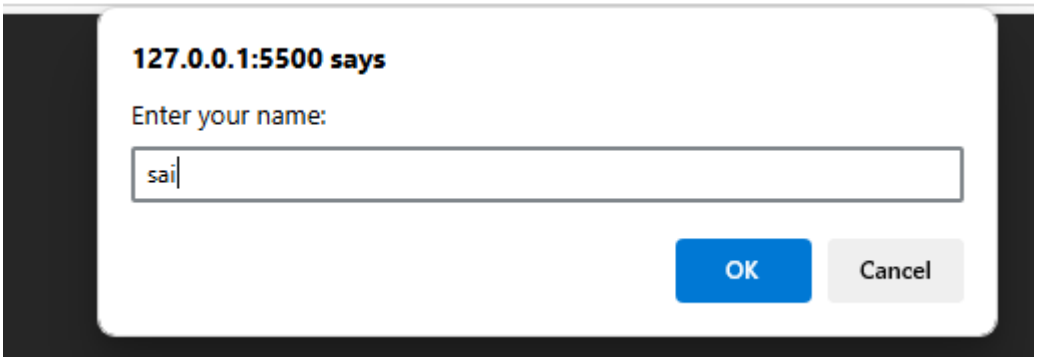
d. Create a webpage which uses prompt dialogue box to ask a voter for his name and age. Display the information in table format along with either the voter can vote or not.

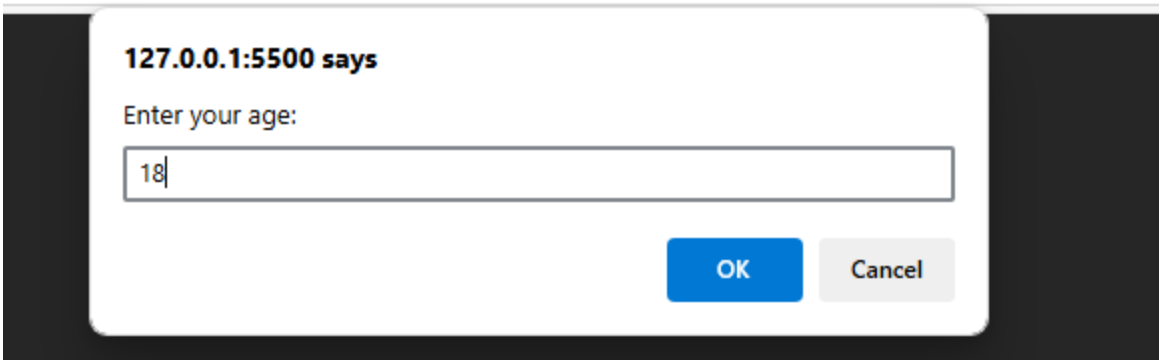
Source code:

```
<!DOCTYPE html>
<head>

    <title>Voter Eligibility Check</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin: 50px;
        }
        table {
            margin: 20px auto;
            border-collapse: collapse;
            width: 50%;
        }
        th, td {
            border: 1px solid black;
            padding: 10px;
            text-align: center;
        }
        th {
            background-color: #f2f2f2;
        }
    </style>
</head>
<body>

    <h1>Voter Eligibility Checker</h1>

    <script>
        // Ask user for Name and Age
        let name = prompt("Enter your name:");
        let age = prompt("Enter your age:");

        // Convert age to number
        age = Number(age);

        // Determine voter eligibility
        let eligibility = (age >= 18) ? "Eligible to Vote " : "Not Eligible to Vote ";

        // Display the result in a table format
        if (name && !isNaN(age)) {
            document.write(`
                <table>
                    <tr>
                        <th>Name</th>
                        <th>Age</th>
                        <th>Voting Eligibility</th>
                    </tr>
                    <tr>
                        <td>${name}</td>
                        <td>${age}</td>
                        <td>${eligibility}</td>
                    </tr>
                </table>
            `);
        } else {
            document.write("<p style='color:red;'>Invalid input. Please refresh and enter valid
details.</p>");
        }
    </script>
```

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

Output:

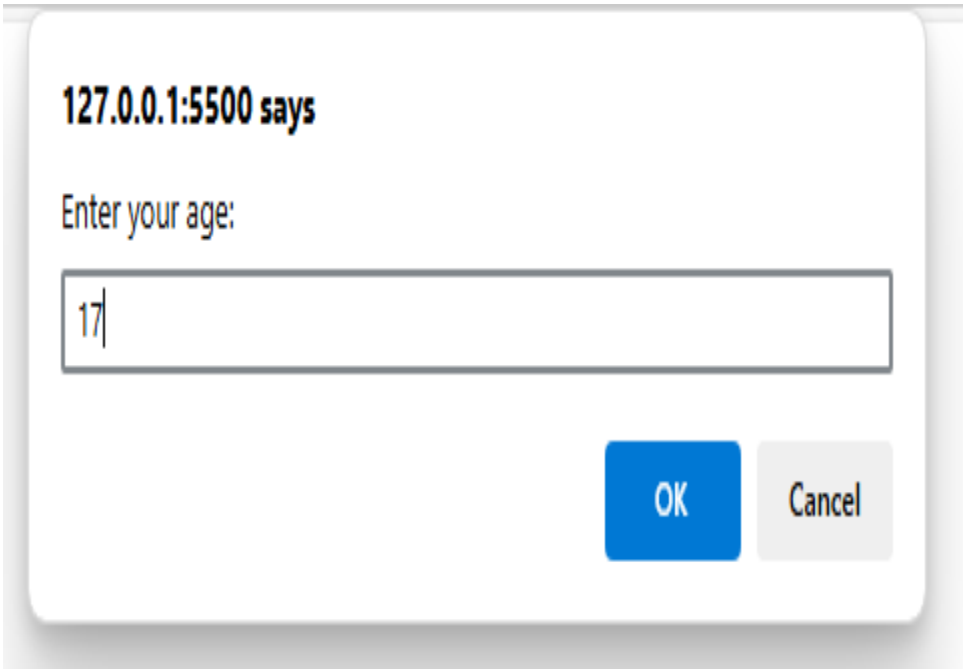




Voter Eligibility Checker

Name	Age	Voting Eligibility
sai	18	Eligible to Vote

Output2:



Voter Eligibility Checker

Name	Age	Voting Eligibility
steve	17	Not Eligible to Vote

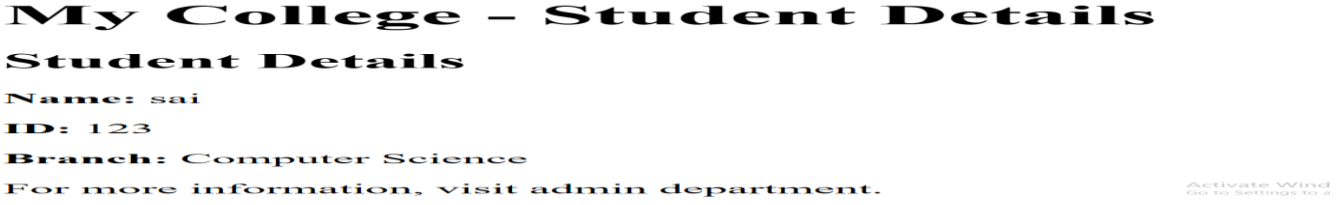
Experiment:7

Java Script Pre-defined and User-defined Objects

a. Write a program using document object properties and methods.
Source code:

```
<!DOCTYPE html>
<head>

  <title>My College - Student Details-7a</title>
  <script>
    // User-defined object constructor for Student
    function Student(name, id, branch) {
      this.name = name;
      this.id = id;
      this.branch = branch;

      // Method to display student details using document methods
      this.displayDetails = function() {
        // Create a div element to hold the student details
        var studentDiv = document.createElement('div');
        studentDiv.innerHTML = '<h2>Student Details</h2>' +
                               '<p><strong>Name:</strong> ' + this.name + '</p>' +
                               '<p><strong>ID:</strong> ' + this.id + '</p>' +
                               '<p><strong>Branch:</strong> ' + this.branch + '</p>';
        // Append the div to the body using the pre-defined document object
        document.body.appendChild(studentDiv);
      }
    }

    // Use window.onload to run code after the document is fully loaded
    window.onload = function() {
      // Create a header element using a document method
      var header = document.createElement('h1');
      header.textContent = 'My College - Student Details';
      document.body.appendChild(header);

      // Create an instance of the user-defined Student object
      var student1 = new Student("sai", "123", "Computer Science");

      // Call the method to display the student's details
      student1.displayDetails();

      // Create a paragraph element to provide additional information
      var infoPara = document.createElement('p');
      infoPara.textContent = 'For more information, visit admin department.';
      document.body.appendChild(infoPara);
    }
  </script>
</head>
<body>
</body>
</html>
```

Output:

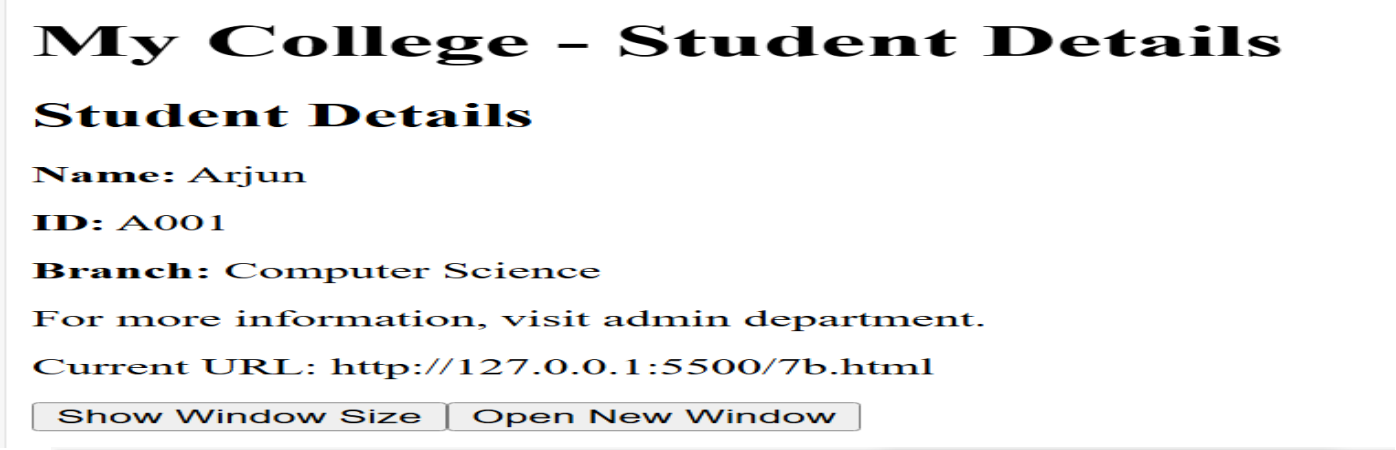


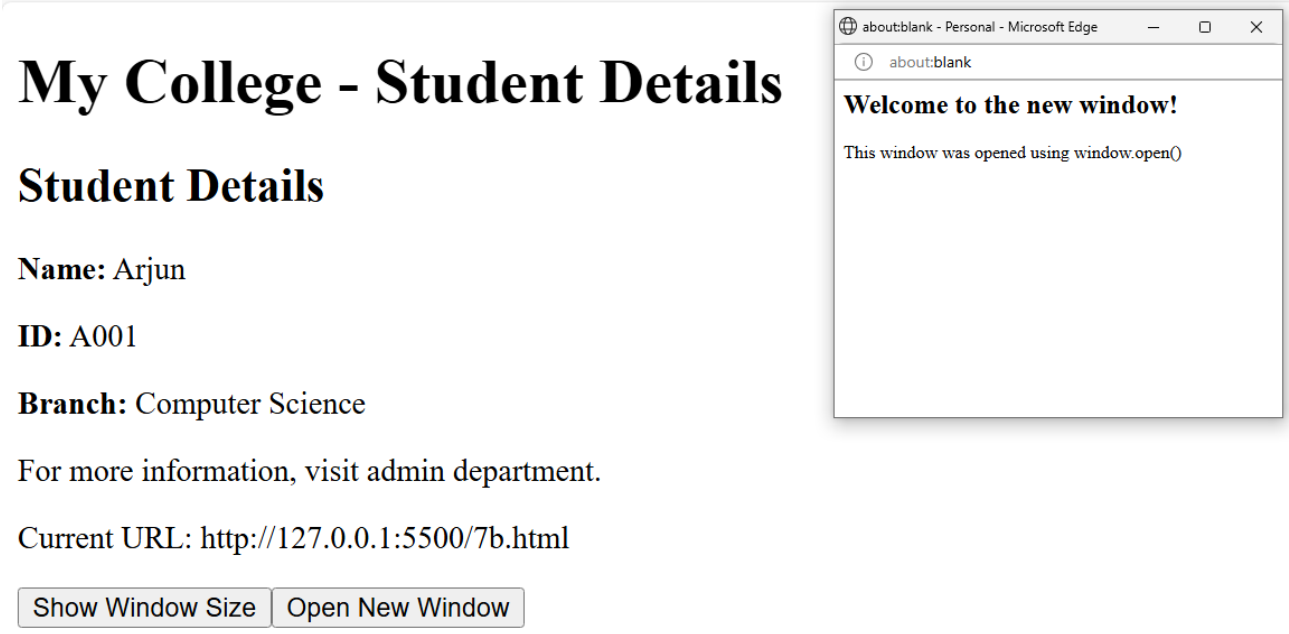
b. Write a program using window object properties and methods.
Source code:

```
<!DOCTYPE html>
<html>
<head>

  <title>My College - Student Details</title>
  <script>
    // User-defined object constructor for Student
    function Student(name, id, branch) {
      this.name = name;
      this.id = id;
      this.branch = branch;

      // Method to display student details using document methods
      this.displayDetails = function() {
        // Create a div element to hold the student details
        var studentDiv = document.createElement('div');
        studentDiv.innerHTML = '<h2>Student Details</h2>' +
                               '<p><strong>Name:</strong> ' + this.name + '</p>' +
                               '<p><strong>ID:</strong> ' + this.id + '</p>' +
                               '<p><strong>Branch:</strong> ' + this.branch + '</p>';
        // Append the div to the body using the pre-defined document object
        document.body.appendChild(studentDiv);
      }
    }

    // Use window.onload to run code after the document is fully loaded
    window.onload = function() {
      // Create a header element using a document method
      var header = document.createElement('h1');
      header.textContent = 'My College - Student Details';
      document.body.appendChild(header);

      // Create an instance of the user-defined Student object
      var student1 = new Student("Arjun", "A001", "Computer Science");

      // Call the method to display the student's details
      student1.displayDetails();

      // Create a paragraph element to provide additional information
      var infoPara = document.createElement('p');
      infoPara.textContent = 'For more information, visit admin department.';
      document.body.appendChild(infoPara);

      // ----- Demonstration of window object properties and methods -----

      // Display the current URL using window.location.href
      var urlPara = document.createElement('p');
      urlPara.textContent = 'Current URL: ' + window.location.href;
      document.body.appendChild(urlPara);

      // Create a button to display window dimensions using window.innerWidth and window.innerHeight
      var sizeButton = document.createElement('button');
      sizeButton.textContent = 'Show Window Size';
      sizeButton.onclick = function() {
        alert('Window Size - Width: ' + window.innerWidth + 'px, Height: ' + window.innerHeight +
'px');
      };
      document.body.appendChild(sizeButton);

      // Create another button that opens a new window using window.open()
      var newWindowButton = document.createElement('button');
      newWindowButton.textContent = 'Open New Window';
      newWindowButton.onclick = function() {
        // Open a new window with a simple HTML content
```

```
        var newWin = window.open("", "NewWindow", "width=400,height=300");
        newWin.document.write("<h2>Welcome to the new window!</h2><p>This window was opened using
window.open()</p>");
      };
      document.body.appendChild(newWindowButton);
    }
  </script>
</head>
<body>
</body>
</html>
```

Output:





c. Write a program using array object properties and methods.

```
<!DOCTYPE html>
<html >
<head>

  <title>My College - Student Details</title>
  <script>
    // User-defined object constructor for Student
    function Student(name, id, branch) {
      this.name = name;
      this.id = id;
      this.branch = branch;

      // Method to display student details using document methods
      this.displayDetails = function() {
        // Create a div element to hold the student details
        var studentDiv = document.createElement('div');
        studentDiv.innerHTML = '<h2>Student Details</h2>' +
                               '<p><strong>Name:</strong> ' + this.name + '</p>' +
                               '<p><strong>ID:</strong> ' + this.id + '</p>' +
                               '<p><strong>Branch:</strong> ' + this.branch + '</p>';
        // Append the div to the body using the pre-defined document object
        document.body.appendChild(studentDiv);
      }
    }

    // Use window.onload to run code after the document is fully loaded
    window.onload = function() {
      // Create a header element using a document method
      var header = document.createElement('h1');
      header.textContent = 'My College - Student Details';
      document.body.appendChild(header);

      // Create an instance of the user-defined Student object
      var student1 = new Student("sai", "123", "Computer Science");

      // Call the method to display the student's details
      student1.displayDetails();

      // Create a paragraph element to provide additional information
      var infoPara = document.createElement('p');
      infoPara.textContent = 'For more information, visit admin department.';
      document.body.appendChild(infoPara);

      // --------- Array Object Properties and Methods Demonstration ---------

      // Section header for Array Operations
      var arrayHeader = document.createElement('h2');
      arrayHeader.textContent = 'Array Object Properties and Methods';
      document.body.appendChild(arrayHeader);

      // Initialize an array of student names
      var studentNames = ["ramu", "seshu", "Romeo", "Priyanka"];

      // Display the initial array using join()
      var initialArrayPara = document.createElement('p');
      initialArrayPara.textContent = 'Initial Array: ' + studentNames.join(", ");
      document.body.appendChild(initialArrayPara);

      // Display the length property of the array
      var lengthPara = document.createElement('p');
      lengthPara.textContent = 'Array Length: ' + studentNames.length;
```

```
      document.body.appendChild(lengthPara);

      // Add a new student name using the push() method
      studentNames.push("Suresh");
      var pushPara = document.createElement('p');
      pushPara.textContent = 'After push("steveharris"): ' + studentNames.join(", ");
      document.body.appendChild(pushPara);

      // Remove the last element using the pop() method
      var removedName = studentNames.pop();
      var popPara = document.createElement('p');
      popPara.textContent = 'After pop(), removed: ' + removedName + '. Array now: ' +
studentNames.join(", ");
      document.body.appendChild(popPara);

      // Sort the array using the sort() method
      studentNames.sort();
      var sortPara = document.createElement('p');
      sortPara.textContent = 'After sort(): ' + studentNames.join(", ");
      document.body.appendChild(sortPara);

      // Reverse the array using the reverse() method
      studentNames.reverse();
      var reversePara = document.createElement('p');
      reversePara.textContent = 'After reverse(): ' + studentNames.join(", ");
      document.body.appendChild(reversePara);

      // Join array elements into a string using the join() method with a custom separator
      var joinStr = studentNames.join(" - ");
      var joinPara = document.createElement('p');
      joinPara.textContent = 'Using join(" - "): ' + joinStr;
      document.body.appendChild(joinPara);
    }
  </script>
</head>
<body>
</body>
</html>
```

Output:

My College - Student Details

Student Details

Name: sai

ID: 123

Branch: Computer Science

For more information, visit admin department.

Array Object Properties and Methods

Initial Array: ramu, seshu, Romeo, Priyaka

Array Length: 4

After push("Steveharris"): ramu, seshu, Romeo, Priyaka, Steveharris

After pop(), removed: Steveharris. Array now: ramu, seshu, Romeo, Priyaka

After sort(): Priyaka, Romeo, ramu, seshu

After reverse(): seshu, ramu, Romeo, Priyaka

Using join(" - "): seshu - ramu - Romeo - Priyaka

d. Write a program using math object properties and methods.

```
<!DOCTYPE html>
<html >
<head>
  <title>My College - Student Details</title>
  <script>
    // User-defined object constructor for Student
    function Student(name, branch, marks, totalMarks) {
      this.name = name;
      // Generate a random user id using Math.random() and Math.floor()
      this.id = "A" + Math.floor(Math.random() * 1000);
      this.branch = branch;
      this.marks = marks;
      this.totalMarks = totalMarks;
      // Calculate percentage and round it to the nearest integer using Math.round()
      this.percentage = Math.round((marks / totalMarks) * 100);

      // Method to display student details using document methods
      this.displayDetails = function() {
        // Create a div element to hold the student details
        var studentDiv = document.createElement('div');
        studentDiv.innerHTML = '<h2>Student Details</h2>' +
                               '<p><strong>Name:</strong> ' + this.name + '</p>' +
                               '<p><strong>ID:</strong> ' + this.id + '</p>' +
                               '<p><strong>Branch:</strong> ' + this.branch + '</p>' +
                               '<p><strong>Marks:</strong> ' + this.marks + ' out of ' +
this.totalMarks + '</p>' +
                               '<p><strong>Percentage:</strong> ' + this.percentage + '%</p>';
        // Append the div to the body using the pre-defined document object
        document.body.appendChild(studentDiv);
      }
    }

    // Run code after the document is fully loaded
    window.onload = function() {
      // Create a header element using a document method
      var header = document.createElement('h1');
      header.textContent = 'My College - Student Details';
      document.body.appendChild(header);

      // Create an instance of the Student object
      // Here, marks are 432 out of 500, and the percentage will be rounded
      var student1 = new Student("steveharris", "Computer Science", 432, 500);

      // Display the student's details
      student1.displayDetails();

      // Additional information regarding Math usage
      var infoPara = document.createElement('p');
      infoPara.textContent = 'Student details include a randomly generated ID and a rounded
percentage computed using Math methods.';
      document.body.appendChild(infoPara);
    }
  </script>
</head>
<body>
</body>
</html>
```

Output:

My College - Student Details

Student Details

Name: Steveharris

ID: A669

Branch: Computer Science

Marks: 432 out of 500

Percentage: 86%

Student details include a randomly generated ID and a rounded percentage computed using Math methods.

e. Write a program using string object properties and methods.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My College - Student Details</title>
  <script>
    // User-defined object constructor for Student
    function Student(name, branch, marks, totalMarks) {
      this.name = name;
      // Generate a random user id using Math.random() and Math.floor()
      this.id = "A" + Math.floor(Math.random() * 1000);
      this.branch = branch;
      this.marks = marks;
      this.totalMarks = totalMarks;
      // Calculate percentage and round it to the nearest integer using Math.round()
      this.percentage = Math.round((marks / totalMarks) * 100);

      // Method to display student details using document methods
      this.displayDetails = function() {
        // Create a div element to hold the student details
        var studentDiv = document.createElement('div');
        studentDiv.innerHTML = '<h2>Student Details</h2>' +
                               '<p><strong>Name:</strong> ' + this.name + '</p>' +
                               '<p><strong>ID:</strong> ' + this.id + '</p>' +
                               '<p><strong>Branch:</strong> ' + this.branch + '</p>' +
                               '<p><strong>Marks:</strong> ' + this.marks + ' out of ' +
this.totalMarks + '</p>' +
                               '<p><strong>Percentage:</strong> ' + this.percentage + '%</p>';
        // Append the div to the body using the pre-defined document object
        document.body.appendChild(studentDiv);
      }
    }

    // Run code after the document is fully loaded
    window.onload = function() {
      // Create a header element using a document method
      var header = document.createElement('h1');
      header.textContent = 'My College - Student Details';
      document.body.appendChild(header);

      // Create an instance of the Student object
      // Here, marks are 432 out of 500, and the percentage will be rounded
      var student1 = new Student("steveharris", "Computer Science", 432, 500);
      // Display the student's details
      student1.displayDetails();

      // Additional information regarding String usage
      var infoPara = document.createElement('p');
      infoPara.textContent = 'This example demonstrates string object properties and methods
applied to student details.';
      document.body.appendChild(infoPara);

      // --------- String Object Properties and Methods Demonstration ---------

      // Section header for String Operations
      var stringHeader = document.createElement('h2');
      stringHeader.textContent = 'String Object Properties and Methods';
      document.body.appendChild(stringHeader);

      // Using the student name for various string operations
      var nameStr = student1.name; // "Arjun Kumar"
```

```
      // Display the length property of the string
      var lengthPara = document.createElement('p');
      lengthPara.textContent = 'Length of student name: ' + nameStr.length;
      document.body.appendChild(lengthPara);

      // Convert the string to uppercase
      var upperPara = document.createElement('p');
      upperPara.textContent = 'Uppercase: ' + nameStr.toUpperCase();
      document.body.appendChild(upperPara);

      // Convert the string to lowercase
      var lowerPara = document.createElement('p');
      lowerPara.textContent = 'Lowercase: ' + nameStr.toLowerCase();
      document.body.appendChild(lowerPara);

      // Extract a substring (first name) using substring() and indexOf()
      var firstName = nameStr.substring(0, nameStr.indexOf(" "));
      var substringPara = document.createElement('p');
      substringPara.textContent = 'First Name (using substring): ' + firstName;
      document.body.appendChild(substringPara);

      // Find the index of the space character in the name using indexOf()
      var indexPara = document.createElement('p');
      indexPara.textContent = 'Index of space in name: ' + nameStr.indexOf(" ");
      document.body.appendChild(indexPara);

      // Split the full name into an array of words using split()
      var splitName = nameStr.split(" ");
      var splitPara = document.createElement('p');
      splitPara.textContent = 'Split name (first and last): ' + splitName.join(" | ");
      document.body.appendChild(splitPara);

      // Display the first character of the name using charAt()
      var charPara = document.createElement('p');
      charPara.textContent = 'First character of the name: ' + nameStr.charAt(0);
      document.body.appendChild(charPara);
    }
  </script>
</head>
<body>
</body>
</html>
```

Output:

My College – Student Details

Student Details

Name: Steveharris

ID: A846

Branch: Computer Science

Marks: 432 out of 500

Percentage: 86%

This example demonstrates string object properties and methods applied to student details.

String Object Properties and Methods

Length of student name: 11

Uppercase: STEVEHARRIS

Lowercase: steveharris

First Name (using substring):

Index of space in name: -1

Split name (first and last): Steveharris

First character of the name: S

f. Write a program using regex object properties and methods.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>My College - Student Details</title>
  <script>
    // User-defined object constructor for Student
    function Student(name, branch) {
      this.name = name;
      // Generate a random user id using Math.random() and Math.floor()
      this.id = "A" + Math.floor(Math.random() * 1000);
      this.branch = branch;

      // Method to display student details using document methods
      this.displayDetails = function() {
        // Create a div element to hold the student details
        var studentDiv = document.createElement('div');
        studentDiv.innerHTML = '<h2>Student Details</h2>' +
                               '<p><strong>Name:</strong> ' + this.name + '</p>' +
                               '<p><strong>ID:</strong> ' + this.id + '</p>' +
                               '<p><strong>Branch:</strong> ' + this.branch + '</p>';
        // Append the div to the body using the document object
        document.body.appendChild(studentDiv);
      }
    }

    // Run code after the document is fully loaded
    window.onload = function() {
      // Create a header element using a document method
      var header = document.createElement('h1');
      header.textContent = 'My College - Student Details';
      document.body.appendChild(header);

      // Create an instance of the Student object (marks and percentage dropped)
      var student1 = new Student("steveharris", "Computer Science");
      student1.displayDetails();

      // --------- Regex Object Properties and Methods Demonstration ---------
      var regexHeader = document.createElement('h2');
      regexHeader.textContent = 'Regex Object Properties and Methods';
      document.body.appendChild(regexHeader);

      // Define a regular expression to validate that the student's name contains only letters
and spaces
      var nameRegex = /^[A-Za-z\s]+$/;

      // Use test() to check if the name matches the regex
      var isNameValid = nameRegex.test(student1.name);
      var testPara = document.createElement('p');
      testPara.textContent = 'Using test(): The student name "' + student1.name + '" is ' +
(isNameValid ? 'valid' : 'invalid') + '.';
      document.body.appendChild(testPara);

      // Use match() to extract the first word (first name) from the student name
      var matchResult = student1.name.match(/[A-Z][a-z]+/);
      var matchPara = document.createElement('p');
      matchPara.textContent = 'Using match(): The first word in the name is "' + (matchResult ?
matchResult[0] : 'Not Found') + '".';
      document.body.appendChild(matchPara);

      // Use search() to find the index of the first space in the student name
```

```
      var searchIndex = student1.name.search(/\s/);
      var searchPara = document.createElement('p');
      searchPara.textContent = 'Using search(): The index of the first space in the name is ' +
searchIndex + '.';
      document.body.appendChild(searchPara);

      // Use replace() to remove all spaces from the student name
      var nameWithoutSpaces = student1.name.replace(/\s/g, "");
      var replacePara = document.createElement('p');
      replacePara.textContent = 'Using replace(): The name without spaces is "' +
nameWithoutSpaces + '".';
      document.body.appendChild(replacePara);

      // Use exec() to execute a search for a match (similar to test but returns an array)
      var execResult = nameRegex.exec(student1.name);
      var execPara = document.createElement('p');
      execPara.textContent = 'Using exec(): The result is ' + (execResult ? execResult[0] :
'null') + '.';
      document.body.appendChild(execPara);
    }
  </script>
</head>
<body>
</body>
</html>
```

Output:

My College - Student Details

Student Details

Name: Steveharris

ID: A338

Branch: Computer Science

Regex Object Properties and Methods

Using test(): The student name "Steveharris" is valid.

Using match(): The first word in the name is "Steveharris".

Using search(): The index of the first space in the name is -1.

Using replace(): The name without spaces is "Steveharris".

Using exec(): The result is Steveharris.

g. . Write a program using date object properties and methods.

```
<!DOCTYPE html>
<html >
<head>

  <title>My College - Student Details</title>
  <script>
    // User-defined object constructor for Student
    function Student(name, branch, dob, joinYear) {
      this.name = name;
      // Generate a random user id using Math.random() and Math.floor()
      this.id = "A" + Math.floor(Math.random() * 1000);
      this.branch = branch;
      // Store date of birth (as a Date object) and the year of joining
      this.dob = new Date(dob); // dob should be in a format like "1999-03-15"
      this.joinYear = joinYear;

      // Calculate the age using the Date object
      var today = new Date();
      var age = today.getFullYear() - this.dob.getFullYear();
      var m = today.getMonth() - this.dob.getMonth();
      if (m < 0 || (m === 0 && today.getDate() < this.dob.getDate())) {
          age--;
      }
      this.age = age;

      // Method to display student details using document methods
      this.displayDetails = function() {
        var studentDiv = document.createElement('div');
        studentDiv.innerHTML = '<h2>Student Details</h2>' +
                               '<p><strong>Name:</strong> ' + this.name + '</p>' +
                               '<p><strong>ID:</strong> ' + this.id + '</p>' +
                               '<p><strong>Branch:</strong> ' + this.branch + '</p>' +
                               '<p><strong>Date of Birth:</strong> ' + this.dob.toDateString() +
'</p>' +
                               '<p><strong>Age:</strong> ' + this.age + '</p>' +
                               '<p><strong>Year of Joining:</strong> ' + this.joinYear + '</p>';
        document.body.appendChild(studentDiv);
      }
    }

    // Run code after the document is fully loaded
    window.onload = function() {
      // Create a header element using document methods
      var header = document.createElement('h1');
      header.textContent = 'My College - Student Details';
      document.body.appendChild(header);

      // Create an instance of the Student object with a date of birth and year of joining
      var student1 = new Student("steveharris", "Computer Science", "1999-03-15", 2018);
      student1.displayDetails();

      // --------- Date Object Properties and Methods Demonstration ---------
      var dateHeader = document.createElement('h2');
      dateHeader.textContent = 'Date Object Properties and Methods';
      document.body.appendChild(dateHeader);

      // Create a Date object for the current date and time
      var currentDate = new Date();

      // Display the current date and time using toLocaleString()
      var currentDatePara = document.createElement('p');
```

```
        currentDatePara.textContent = 'Current Date and Time: ' + currentDate.toLocaleString();
        document.body.appendChild(currentDatePara);

        // Display individual properties of the current date
        var currentYearPara = document.createElement('p');
        currentYearPara.textContent = 'Current Year: ' + currentDate.getFullYear();
        document.body.appendChild(currentYearPara);

        var currentMonthPara = document.createElement('p');
        // getMonth() returns month index (0-11), add 1 for human-readable format
        currentMonthPara.textContent = 'Current Month: ' + (currentDate.getMonth() + 1);
        document.body.appendChild(currentMonthPara);

        var currentDayPara = document.createElement('p');
        currentDayPara.textContent = 'Current Day: ' + currentDate.getDate();
        document.body.appendChild(currentDayPara);

        var currentHoursPara = document.createElement('p');
        currentHoursPara.textContent = 'Current Hour: ' + currentDate.getHours();
        document.body.appendChild(currentHoursPara);

        var currentMinutesPara = document.createElement('p');
        currentMinutesPara.textContent = 'Current Minutes: ' + currentDate.getMinutes();
        document.body.appendChild(currentMinutesPara);

        var currentSecondsPara = document.createElement('p');
        currentSecondsPara.textContent = 'Current Seconds: ' + currentDate.getSeconds();
        document.body.appendChild(currentSecondsPara);
      }
    </script>
</head>
<body>
</body>
</html>
```

Output:

My College - Student Details

Student Details

Name: Steveharris

ID: A692

Branch: Computer Science

Date of Birth: Mon Mar 15 1999

Age: 26

Year of Joining: 2018

Date Object Properties and Methods

Current Date and Time: 4/7/2025, 2:21:38 PM

Current Year: 2025

Current Month: 4

Current Day: 7

Current Hour: 14

Current Minutes: 21

Current Seconds: 38

h. Write a program to explain user-defined object by using properties, methods, accessors,constructors and display.

```
<!DOCTYPE html>
<html>
<head>

  <title>My College - Student Details</title>
  <script>
    // User-defined object constructor for Student
    function Student(firstName, lastName, branch) {
      // Properties
      this.firstName = firstName;
      this.lastName = lastName;
      this.branch = branch;
      // Generate a random user ID using Math.random() and Math.floor()
      this.id = "A" + Math.floor(Math.random() * 1000);

      // Accessor: Define a getter and setter for fullName
      Object.defineProperty(this, 'fullName', {
        get: function() {
          return this.firstName + " " + this.lastName;
        },
        set: function(name) {
          var parts = name.split(" ");
          this.firstName = parts[0];
          this.lastName = parts[1] || "";
        },
        enumerable: true,
        configurable: true
      });

      // Method: Display student details on the page using document methods
      this.displayDetails = function() {
        var studentDiv = document.createElement('div');
        studentDiv.innerHTML = '<h2>Student Details</h2>' +
                               '<p><strong>ID:</strong> ' + this.id + '</p>' +
                               '<p><strong>Full Name:</strong> ' + this.fullName + '</p>' +
                               '<p><strong>Branch:</strong> ' + this.branch + '</p>';
        document.body.appendChild(studentDiv);
      }
    }

    // Run the code after the document is fully loaded
    window.onload = function() {
      // Create an instance of the Student object using the constructor
      var student1 = new Student("steve", "harris", "Computer Science");

      // Display the student details initially
      student1.displayDetails();

      // Demonstrate the accessor: update the full name using the setter
      var updatePara = document.createElement('p');
      updatePara.textContent = "Updating full name using the accessor...";
      document.body.appendChild(updatePara);

      // Update the fullName property (this will update firstName and lastName)
      student1.fullName = "steve smith";

      // Display the updated student details
      student1.displayDetails();
    }
  </script>
```

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

Output:

Student Details

ID: A936

Full Name: Steveharris sai

Branch: Computer Science

Updating full name using the accessor...

Student Details

ID: A936

Full Name: steve smith

Branch: Computer Science

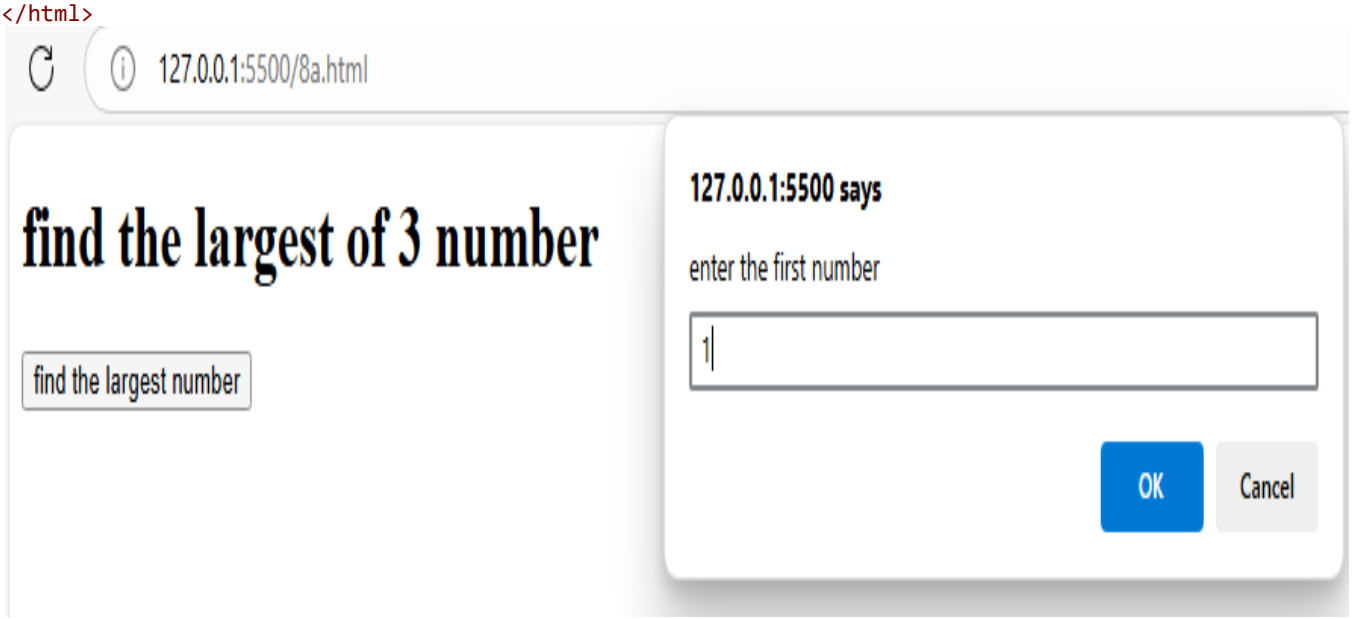
Experiment:8

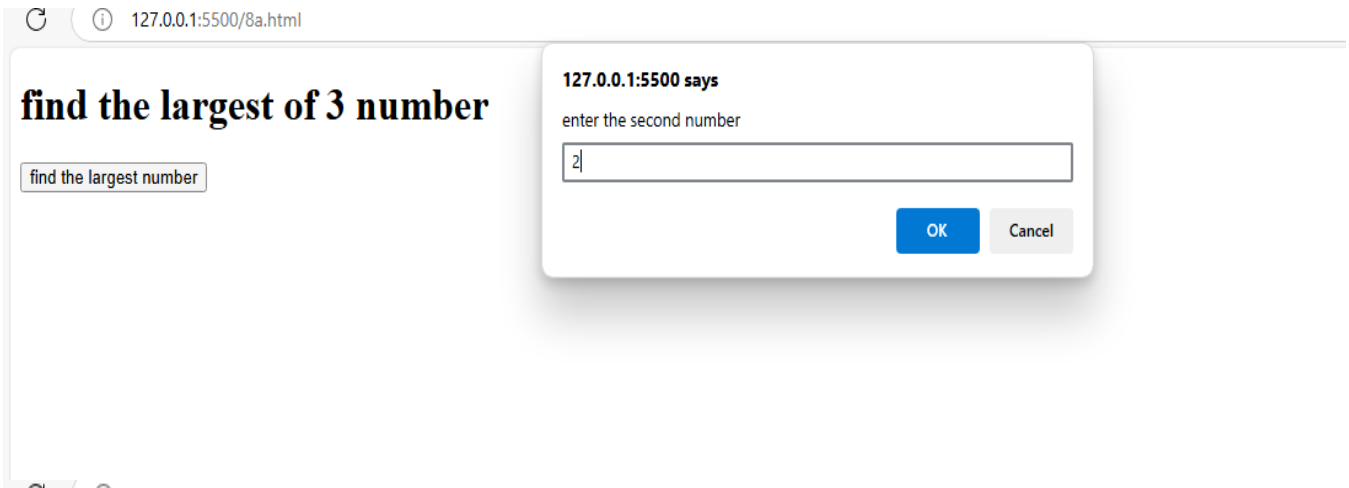
Java Script Conditional Statements and Loops

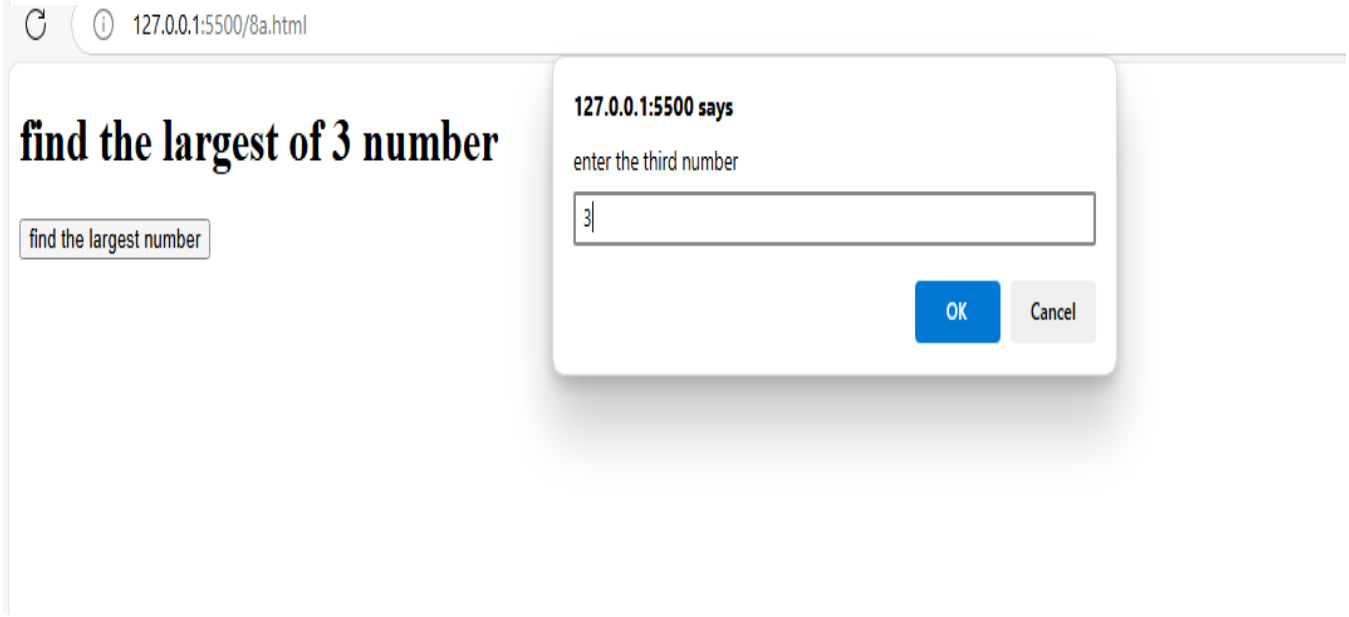
a. Write a program which asks the user to enter three integers, obtains the numbers from the user and outputs HTML text that displays the larger number followed by the words “LARGER NUMBER” in an information message dialog. If the numbers are equal, output HTML text as “EQUAL NUMBERS”.

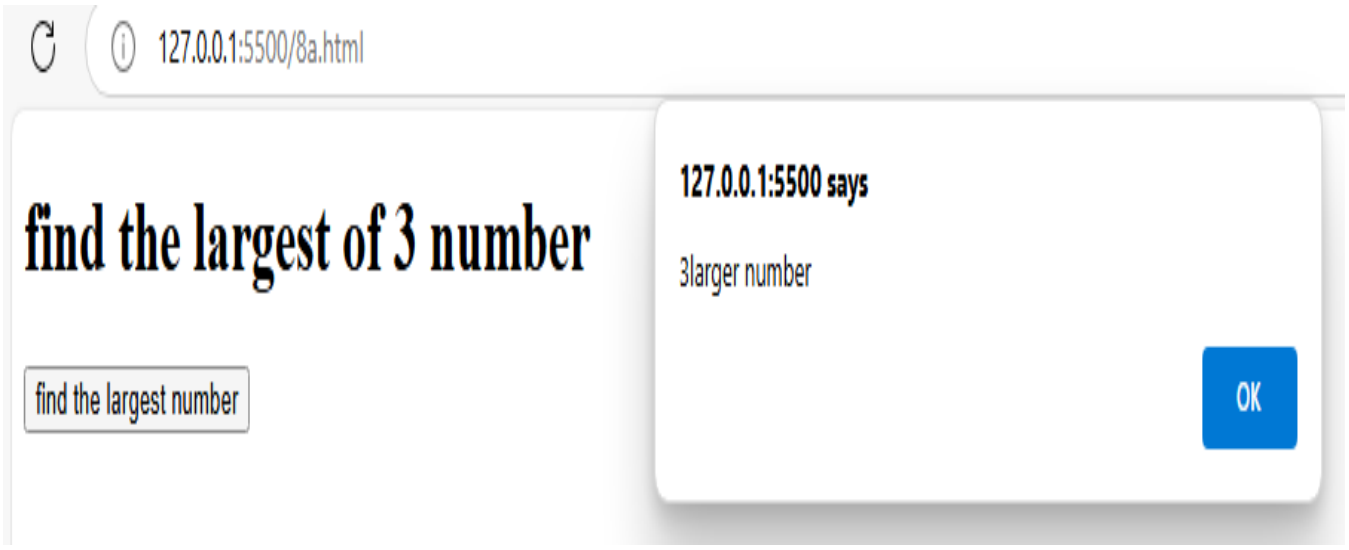
Source code:

```
<!DOCTYPE html>
<html>
    <head>
        <title>largest number</title>
        <script>
            function Largest()
{
                let num1=prompt("enter the first number");
                let num2=prompt("enter the second number");
                let num3=prompt("enter the third number");
                if(num1==num2&&num2==num3)

            {
                alert("equal numbers");


            }
            else{
                let larger=Math.max(num1,num2,num3);
                alert(larger+"larger number");
            }

            }
        </script>
</head>
    <body>
        <h1>find the largest of 3 number</h1>
        <button onclick="Largest()">find the largest number</button>


    </body>
</html>
```



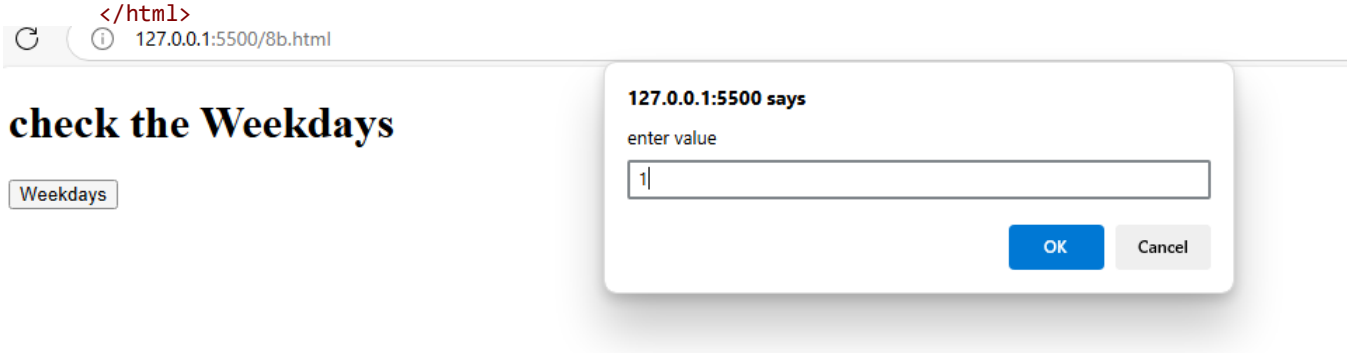

127.0.0.1:5500/8a.html
find the largest of 3 number
find the largest number
127.0.0.1:5500 says
enter the second number
2
OK
Cancel


127.0.0.1:5500/8a.html
find the largest of 3 number
find the largest number
127.0.0.1:5500 says
enter the third number
3
OK
Cancel

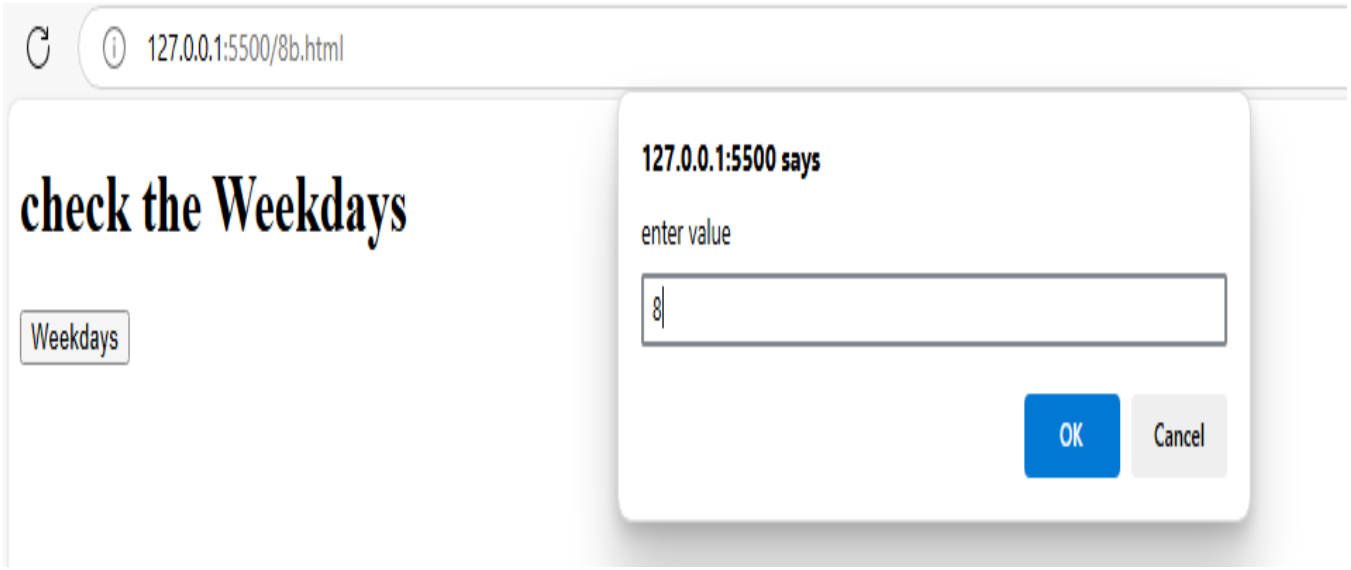

127.0.0.1:5500/8a.html
find the largest of 3 number
find the largest number
127.0.0.1:5500 says
3larger number
OK

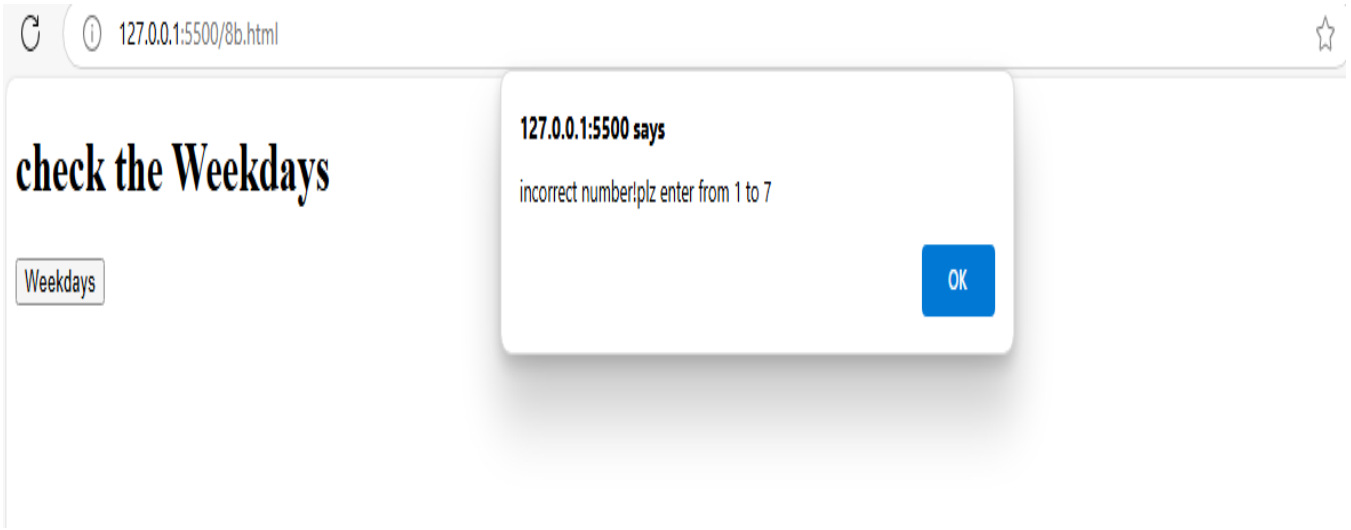
b. Write a program to display week days using switch case.

```
<!DOCTYPE html>
<html>
    <head>
        <title>sample</title>
        <script>
            function Weekdays()
            {
                let day=parseInt(prompt("enter value"));
                switch(day)
                {
                    case 1:
                        alert("sunday");
                        break;

                        case 2:
                            alert("monday ");
                            break;

                            case 3:
                                alert("tuesday");
                                break;

                                case 4:
                                    alert("wednesday");
                                    break;

                                    case 5:
                                        alert("thursday");
                                        break;

                                        case 6:
                                            alert("friday");
                                            break;

                                            case 7:
                                                alert("saturday");
                                                break;


                            default:
                                alert("incorrect number!plz enter from 1 to 7");

                }
            }
        </script>
    </head>
    <body>
        <h1>check the Weekdays</h1>
        <button onclick="Weekdays()">Weekdays</button>
    </body>
</html>
```




127.0.0.1:5500/8b.html
check the Weekdays
Weekdays
127.0.0.1:5500 says
sunday
OK

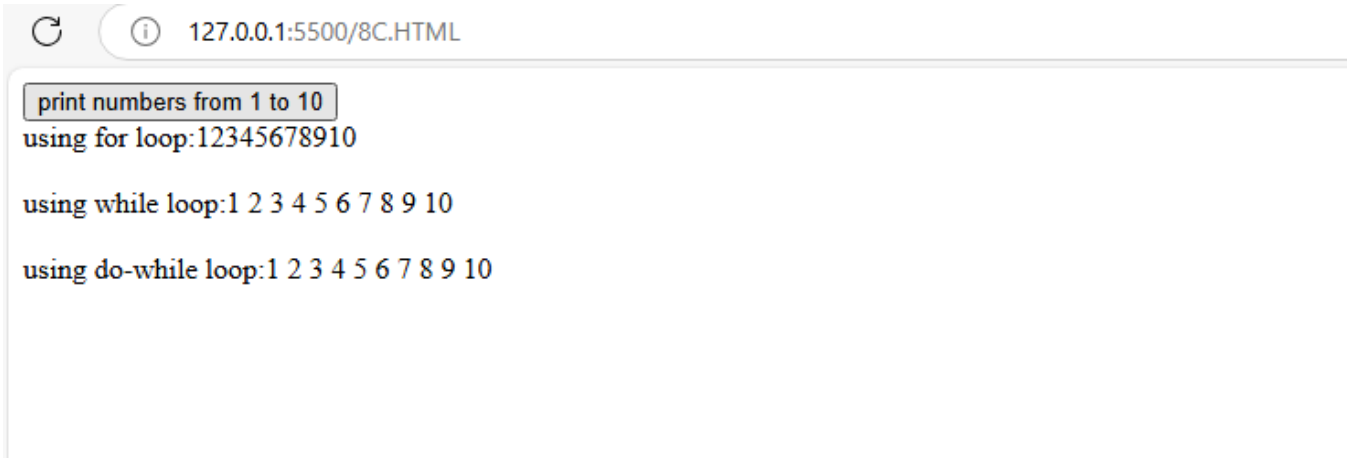

127.0.0.1:5500/8b.html
check the Weekdays
Weekdays
127.0.0.1:5500 says
enter value
8
OK
Cancel


127.0.0.1:5500/8b.html
check the Weekdays
Weekdays
127.0.0.1:5500 says
incorrect number!plz enter from 1 to 7
OK

c. Write a program to print 1 to 10 numbers using for, while and do-while loops.

```
<!DOCTYPE html>
<html>
    <head>
        <title>do while example</title>
        <script>
            function Printnumbers()
            {
                let output="using for loop:";
                for(let i=1;i<=10;i++)
            {
                output +=i;

            }
            document.getElementById("for").innerHTML=output;
            //whileloop
            output1="<br>using while loop:";
            let i=1;
            while(i<=10)
            {
                output1 +=i+" ";
                i++;
            }
            document.getElementById("while").innerHTML=output1;
            //do-whileloop
            output2="<br>using do-while loop:";
            i=1;
            do{
                output2 +=i+" ";
                i++;

            }while(i<=10);
            document.getElementById("dow").innerHTML=output2;
            }
        </script>
    </head>
    <body>
        <button onclick="Printnumbers()">print numbers from 1 to 10</button>
        <div id="for"></div>
        <div id="while"></div>
        <div id="dow"></div>
    </body>
</html>
```

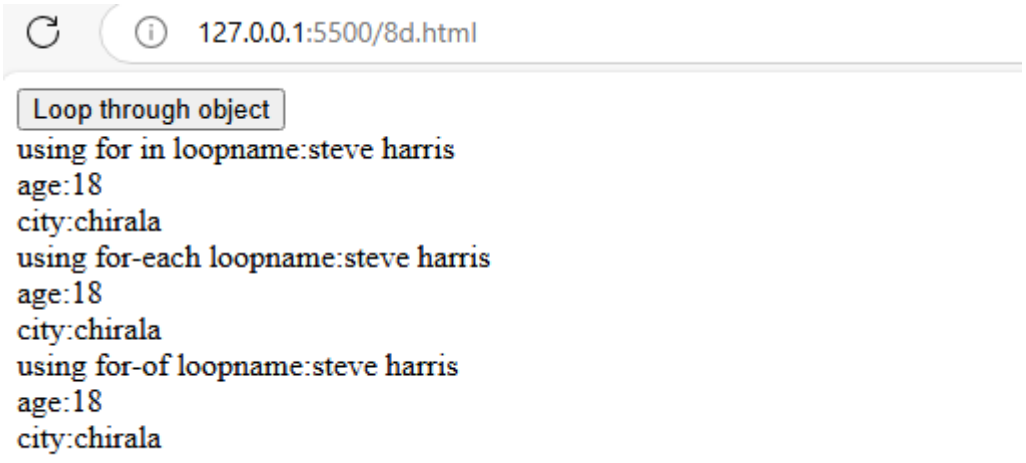
Output:



d. Write aprogram to print data in object using for-in, for-each and for-of loops.

```
<!DOCTYPE html>
<html>
    <head>
        <title>for in and for each</title>
        <script>
            function loopThroughObject()
            {
                let person={
                    name:"steve harris",
                    age:18,
                    city:"chirala"
                };
                let output="using for in loop";
                for(let key in person)
                {
                    output +=key + ":"+person[key]+"<br>";
                }
                output +="using for-each loop";
                Object.entries(person).forEach(([key,value])=>{
                    output +=key + ":"+value+"<br>";
                });

                output +="using for-of loop";
                for (const[key,value]of Object.entries(person)){
                    output +=key+":"+value+"<br>";
                }
                document.getElementById("output").innerHTML=output;
            }
        </script>
    </head>
    <body>
        <button onclick="loopThroughObject()">Loop through object</button>
        <div id="output"></div>

    </body>
</html>
```

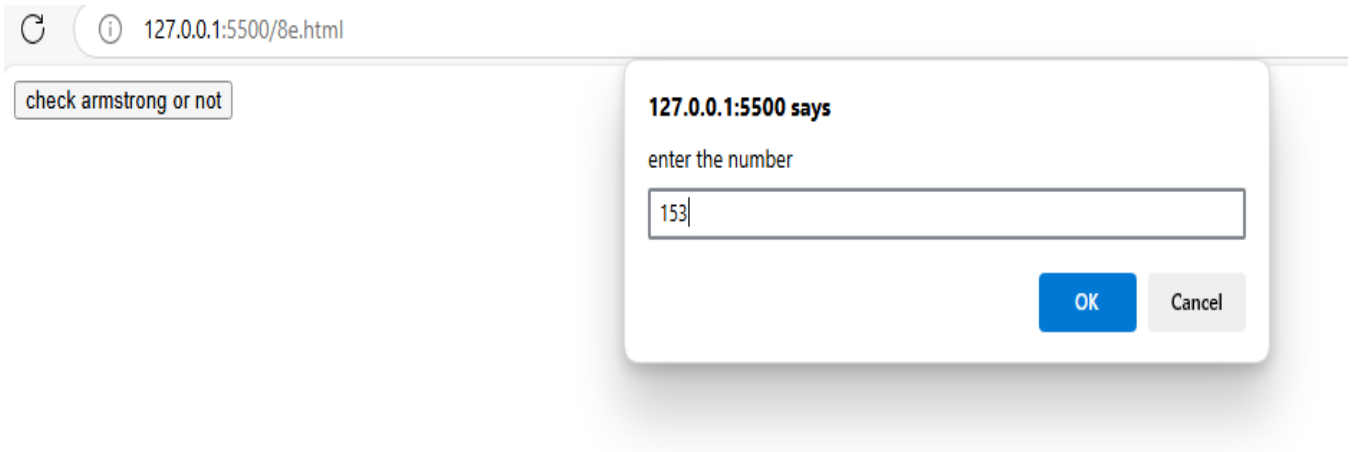
Output:

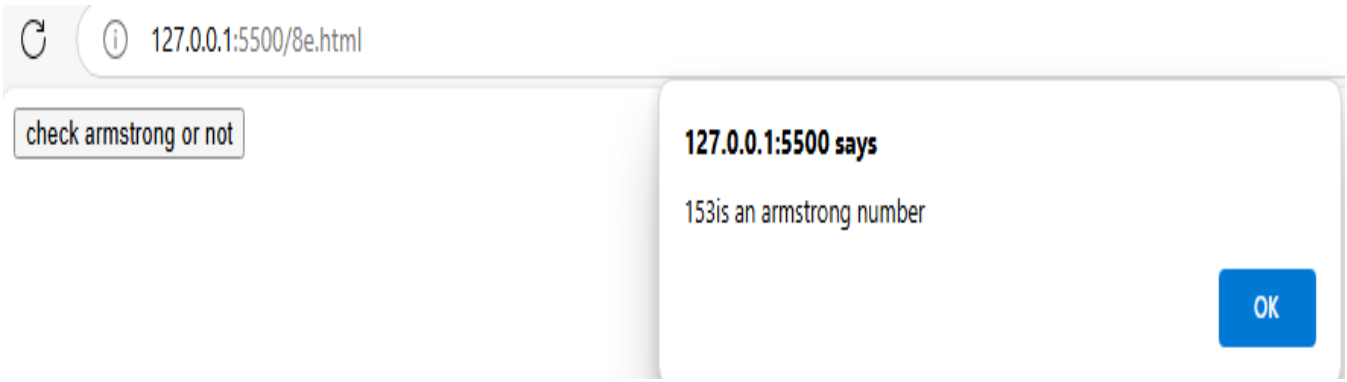


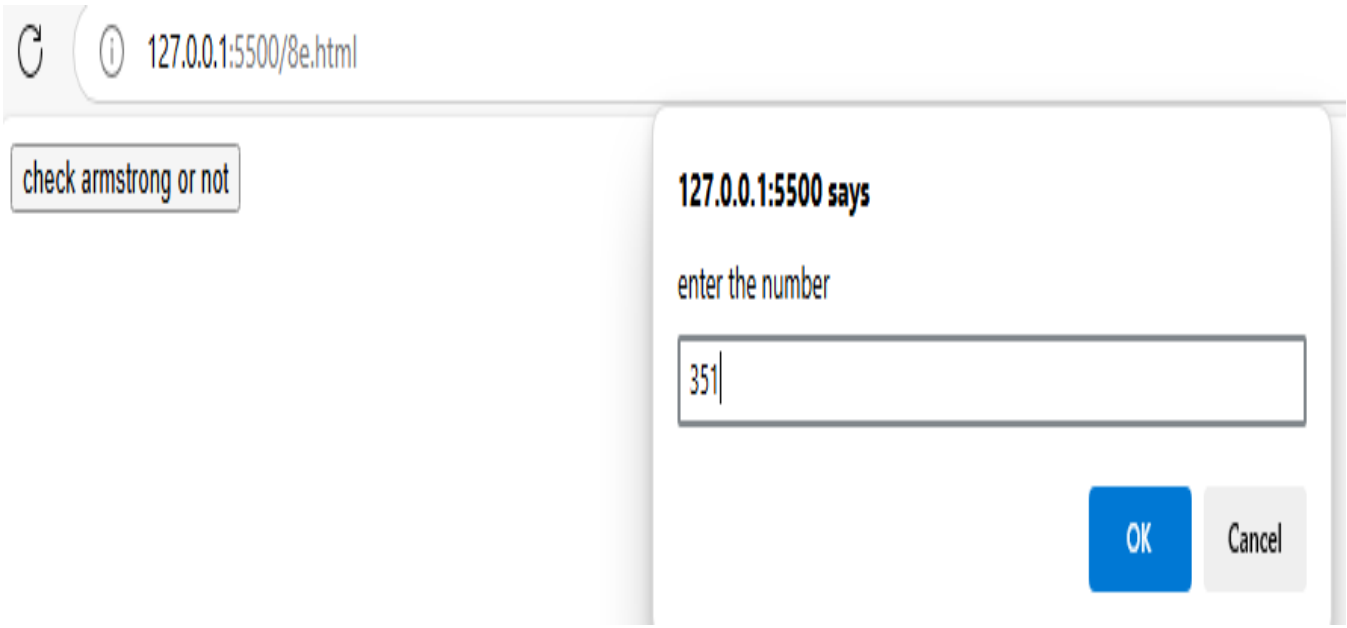
e. Develop a program to determine whether a given number is an 'ARMSTRONG NUMBER' or not. [Eg: 153 is an Armstrong number, since sum of the cube of the digits is equal to the number i.e.,13 + 53+ 33 = 153]

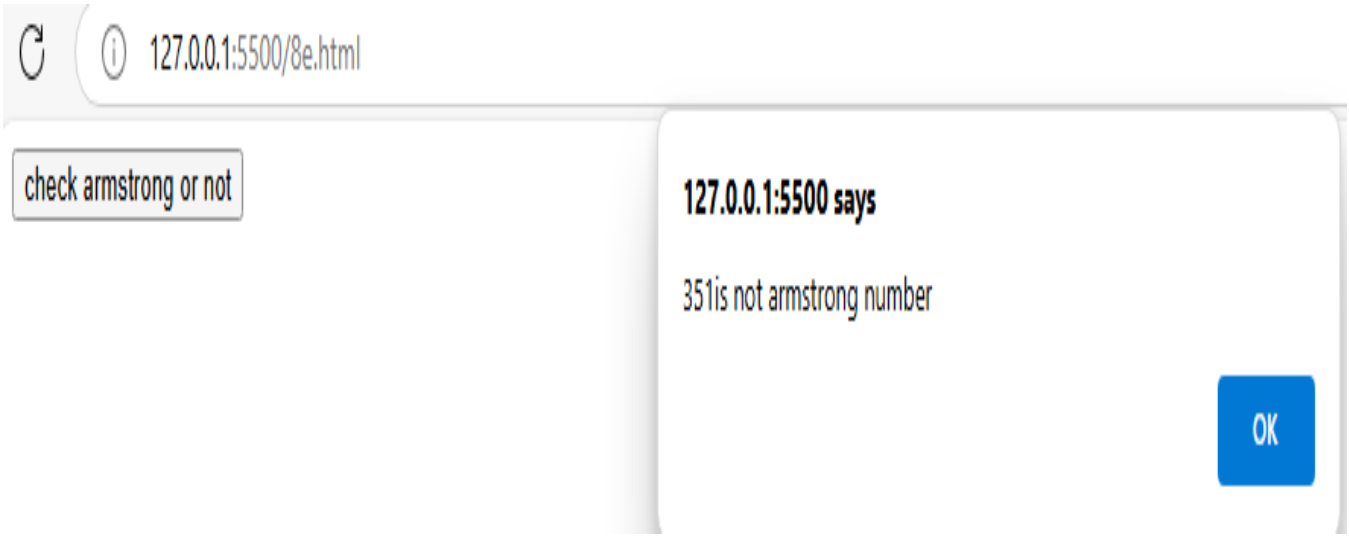
```
<!DOCTYPE html>
<html>
    <head>
        <title>armstrong number or not</title>
        <script>
            function Armstrong()
            {
                let num=parseInt(prompt("enter the number"));
                let sum=0;
                let temp=num;
                let digits=num.toString().length;

                while(temp>0)
            {
                let digit=temp%10;
                sum +=Math.pow(digit,digits);
                temp=Math.floor(temp/10);

            }
            if(sum==num)
            {
                alert(num+"is an armstrong number");
            }
            else{
                alert(num+"is not armstrong number");
            }

            }
        </script>
    </head>
    <body>
        <button onclick="Armstrong()">check armstrong or not</button>
    </body>
</html>
```

Output:




127.0.0.1:5500/8e.html
check armstrong or not
127.0.0.1:5500 says
153is an armstrong number
OK

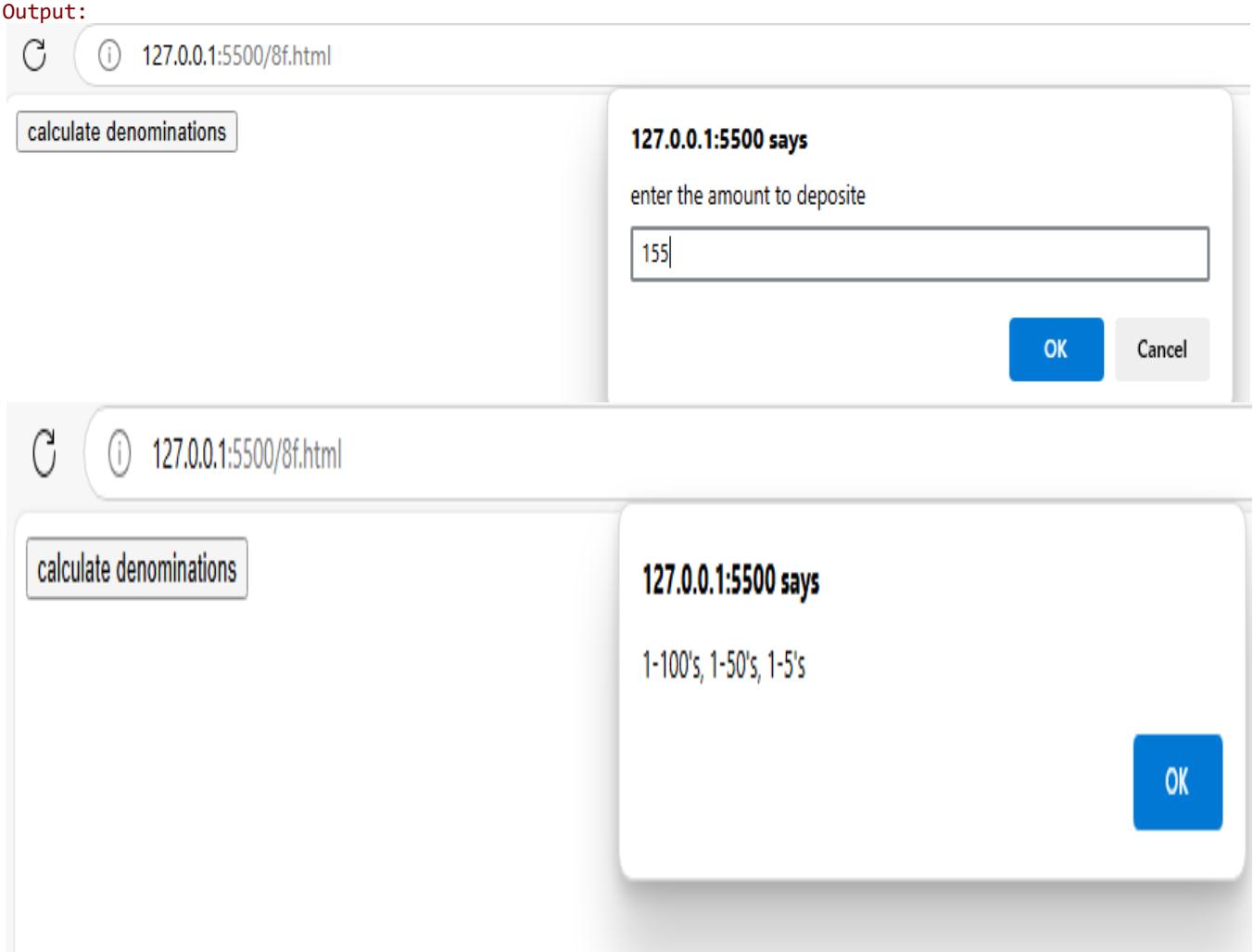

127.0.0.1:5500/8e.html
check armstrong or not
127.0.0.1:5500 says
enter the number
351
OK
Cancel


127.0.0.1:5500/8e.html
check armstrong or not
127.0.0.1:5500 says
351is not armstrong number
OK

f. Write a program to display the denomination of the amount deposited in the bank in terms of 100's, 50's, 20's, 10's, 5's, 2's & 1's. (Eg: If deposited amount is Rs.163, the output should be 1-100's, 1-50's, 1- 10's, 1-2's & 1-1's).

```
<!DOCTYPE html>
<head>
    <script>
        function calculateDenominations()
        {
            let amount=parseInt(prompt("enter the amount to deposite"));
            let denominations=[100,50,20,10,5,2,1];
            let result="";
            for(let i=0;i<denominations.length;i++)
        {
            let count=Math.floor(amount/denominations[i]);
            if(count>0)
        {
            result +=count+"-"+denominations[i]+"'s, ";
            amount=amount%denominations[i];
        }
        }
        alert(result.slice(0,-2));//to remove last comma and space
        }
    </script>
</head>
<body>
    <button onclick="calculateDenominations()">calculate denominations</button>
</body>
</html>
```

Output:



Experiment:9

Java Script Functions and Events

a. Design a appropriate function should be called to display i. Factorial of that number ii. Fibonacci series up to that number iii. Prime numbers up to that number iv. Is it palindrome or not.

b. Design a HTML having a text box and four buttons named Factorial, Fibonacci, Prime, and Palindrome. When a button is pressed an appropriate function should be called to display i. Factorial of that number ii. Fibonacci series up to that number iii. Prime numbers up to that number iv. Is it palindrome or not

```
<!DOCTYPE html>
<head>
    <title>number of operations</title>
</head>
<body>
    <div>
        <h2>number operations</h2>
        <input type="number" id="numberInput" placeholder="enter a number">
        <button onclick="Factorial()">Factorial</button>
        <button onclick="Fibonacci()">Fibonacci</button>
        <button onclick="Primenum()">Prime numbers</button>
        <button onclick="Palindrome()">Palindrome</button>
        <div id="result"></div>
    </div>
    <script>
        //Factorial
        function Factorial(){
            let num=document.getElementById('numberInput').value;
            num=parseInt(num);

            if(isNaN(num)||num<=0)
        {
            document.getElementById('result').innerHTML="please enter a positive number";
            return;
        }
        let result=1;
        for(let i=1;i<=num;i++)
        {
            result *=i;
        }
        document.getElementById('result').innerHTML='Factorial of'+num+'is:'+result;
        }
        //Fibonacci
        function Fibonacci(){
            let num=document.getElementById('numberInput').value;
            num=parseInt(num);
            if(isNaN(num)||num<=0){
                document.getElementById('result').innerHTML="please enter a positive integer";
                return;
            }
            let fib=[0,1];
            for(let i=2;i<num;i++)
            {
                fib.push(fib[i-1]+fib[i-2]);

            }
            document.getElementById('result').innerHTML='Fibonacci series
upto'+num+'is:'+fib.join(',');
        }

        //prime number
        function Primenum(){
            let num=document.getElementById('numberInput').value;
            num=parseInt(num);
```

```
            if(isNaN(num)||num<2){
                document.getElementById('result').innerHTML="please enter a number greaterthan or
eqaul to 2";
                return;
            }
            let primes=[];
            for(let i=2;i<=num;i++)
            {
                let isPrime=true;
                for(let j=2;j<=Math.sqrt(i);j++)
            {
                if(i%j===0)
            {
                isPrime=false;
                break;
            }
            }
            if(isPrime){
                primes.push(i);
            }

            }
            document.getElementById('result').innerHTML='prime numbers
 upto'+num+'is:'+primes.join(',');
        }
        //palindrome
        function Palindrome(){
            let num=document.getElementById('numberInput').value;
            if(isNaN(num))
        {
            document.getElementById('result').innerHTML="please enter valid number";
            return;
        }
        let strnum=num.toString();
        let reversed=strnum.split('').reverse().join('');
        if(strnum==reversed)
        {
            document.getElementById('result').innerHTML=num+'is a plindrome';
        }
        else {
            document.getElementById('result').innerHTML=num+'is not a plaindrome';
        }
                        }

    </script>

</body>
</html>
```

Output:

number operations

5 Factorial Fibonacci Prime numbers Palindrome

Fibonacci series upto5is:0,1,1,2,3

number operations

5 Factorial Fibonacci Prime numbers Palindrome

prime numbers upto5is:2,3,5

number operations

1010 Factorial Fibonacci Prime numbers Palindrome

1010is not a plaindrome

number operations

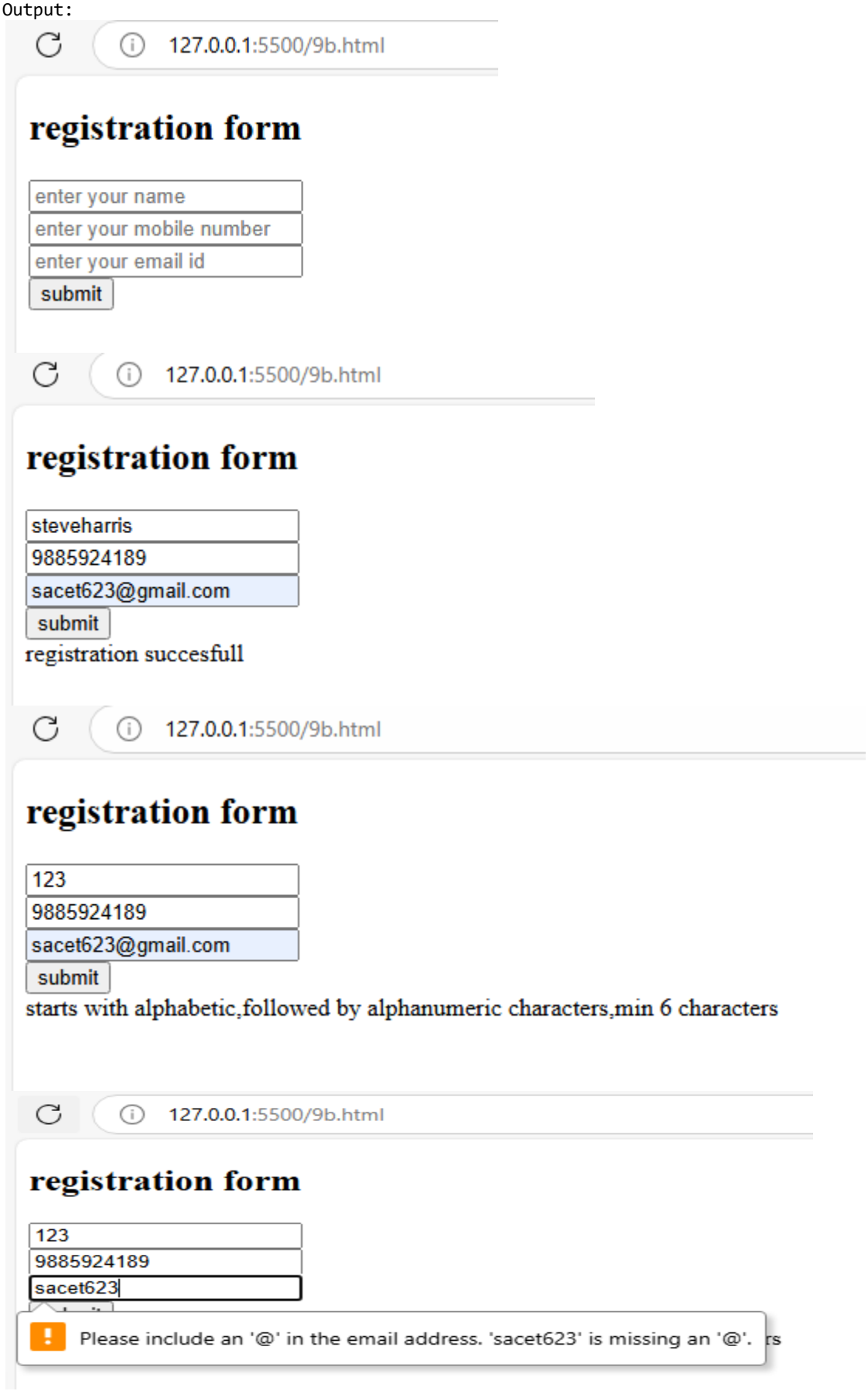
1001 Factorial Fibonacci Prime numbers Palindrome

1001is a plindrome

c. Write a program to validate the following fields in a registration page. i.Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters) ii. Mobile (only numbers and length 10 digits) iii. E-mail (should contain format like xxxxxxx@xxxxxx.xxx)

```
<!DOCTYPE html>
<html>
    <head>
        <title>registration form validation</title>
    </head>
    <body>
        <div>
            <h2>
                registration form
            </h2>
            <form id="registrationform">
            <input type="text" id="name" placeholder="enter your name" required><br>
            <input type="text" id="mobile" placeholder="enter your mobile number" required><br>
            <input type="email" id="email" placeholder="enter your email id" required><br>
                <button type="submit">submit</button>
            </form>
            <div id="errormessage"></div>
        </div>
        <script>
            function validatename(name){
                //starts with alphabetic,followed by alphanumeric characters,min 6 characters
                const nameregex=/^[A-Za-z][A-Za-z0-9]{5,}$/;
                return nameregex.test(name);
            }
            function validatemobile(mobile){
                const mobileregex=/^\d{10}$/;
                return mobileregex.test(mobile);
            }
            function validateemail(email){
                const emailregex=/^[a-zA-Z0-9._]+@[a-zA-Z0-9]+\.[a-zA-Z]{2,6}$/;
                return emailregex.test(email);
            }
            document.getElementById('registrationform').addEventListener('submit',function(event)
        {
            event.preventDefault();
            const name=document.getElementById('name').value;
            const mobile=document.getElementById('mobile').value;
            const email=document.getElementById('email').value;
            let errormessages='';

            if(!validatename(name)){
                errormessages +='starts with alphabetic,followed by alphanumeric characters,min 6
characters<br>';
                }
                if(!validatemobile(mobile)){
                errormessages +='mobile number should be numbers and 10 digits only<br>';
                }
                if(!validateemail(email)){
                errormessages +='email id should be in right format<br>';
                }

                if(errormessages){
                    document.getElementById('errormessage').innerHTML=errormessages;
                }
                else{
                    document.getElementById('errormessage').innerHTML='registration succesfull';
                }

        });
        </script>
    </body>
</html>
```

Output:


127.0.0.1:5500/9b.html
registration form
enter your name
enter your mobile number
enter your email id
submit
127.0.0.1:5500/9b.html
registration form
steveharris
9885924189
sacet623@gmail.com
submit
registration succesfull
127.0.0.1:5500/9b.html
registration form
123
9885924189
sacet623@gmail.com
submit
starts with alphabetic,followed by alphanumeric characters,min 6 characters
127.0.0.1:5500/9b.html
registration form
123
9885924189
sacet623
Please include an '@' in the email address. 'sacet623' is missing an '@'.