

Web Technology Lab-2

Name: Pratyush Tiwari
Roll No: 22MC3024
Branch: Mathematics & Computing

T1. Make a little webpage to tell a little about yourself. Use at least three levels of headings and some paragraphs.



The screenshot shows a code editor interface with the following details:

- File tabs: index.html, styles.css, script.js.
- Toolbar: NEW, HTML (selected), RUN, and other icons.
- Code area (index.html):

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>About Me</title>
7  </head>
8  <body>
9      <h1>About Me</h1>
10     <h2>Introduction</h2>
11     <p>I am Shubham Kumar, B-tech undergrad at Rajiv Gandhi Institute of Petroleum Technology.</p>
12     <h2>Skills</h2>
13     <p>I'm proficient in Problem-Solving.</p>
14     <h2>Interests</h2>
15     <p>My interests include artificial intelligence, machine learning, natural language processing, and web development.</p>
16 </body>
17 </html>
```
- Preview area:
 - About Me**
 - Introduction**

I am Shubham Kumar, B-tech undergrad at Rajiv Gandhi Institute of Petroleum Technology.
 - Skills**

I'm proficient in Problem-Solving.
 - Interests**

My interests include artificial intelligence, machine learning, natural language processing, and web development.

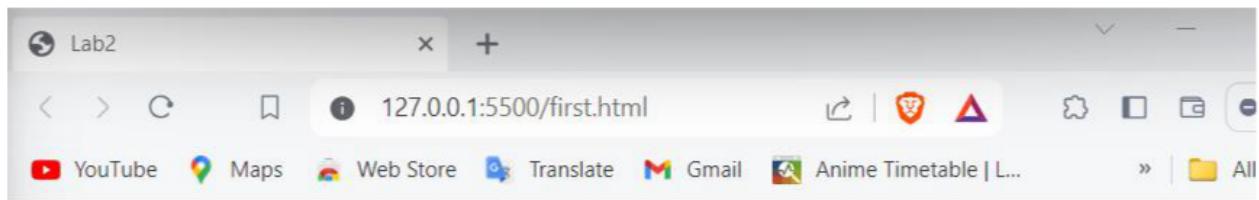
T2. Write an html page that includes a header element giving a title to some image then include the image in the page, sized appropriately, and a paragraph that describes the image.

Ans

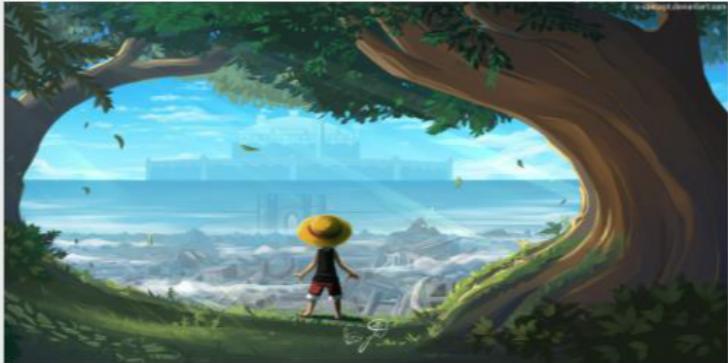
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lab2</title>
</head>
<body>
    <header>
        <h2><u>Wallpaper</u></h2>
```

```
</header>

<p>This is a Wallpaper with height=200px and width=400px.</p>
</body>
</html>
```



Wallpaper



This is a Wallpaper with height=200px and width=400px.

T3. Make an ordered list that has capital roman numerals at the highest level and lowercase roman numerals for the second level items. List three countries at the first level, and for each of those list two spoken languages in that country.

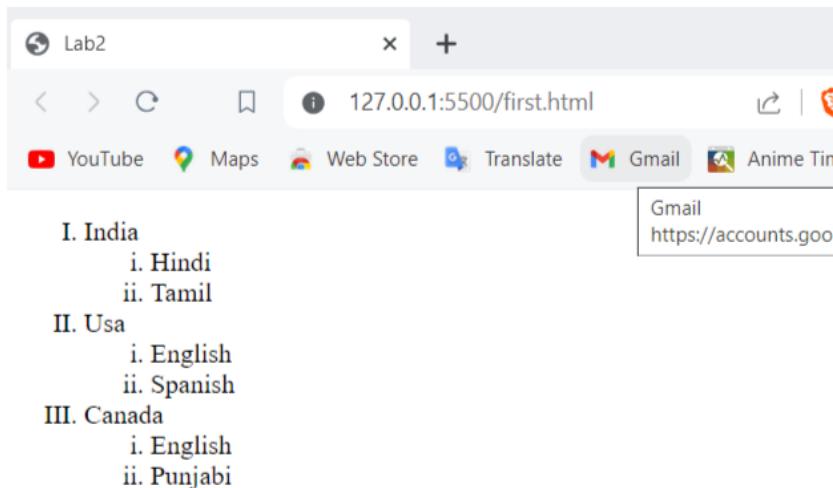
Ans

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Lab2</title>
```

```

</head>
<body>
    <ol type="I">
        <li>India
            <ol type="i">
                <li>Hindi</li>
                <li>Tamil</li>
            </ol>
        </li>
        <li>Usa
            <ol type="i">
                <li>English</li>
                <li>Spanish</li>
            </ol>
        </li>
        <li>Canada
            <ol type="i">
                <li>English</li>
                <li>Punjabi</li>
            </ol>
        </li>
    </ol>
</body>
</html>

```



T4. Make a webpage that uses a table to create two columns. In the left column should be a bulleted list of 5 countries. In the right column should be a numbered list of capitals of those countries.

Ans

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lab2</title>
</head>
<body>
    <table>
        <thead> <tr>
            <th>Column 1</th>
            <th>Column 2</th>
        </tr>
        </thead>
        <tbody> <tr>

<td><ul><li>India</li><li>USA</li><li>China</li><li>Bangladesh</li><li>Nepal</li></ul></td>
            <td><ol><li>New Delhi</li><li>Washington D.C.</li><li>Beijing</li><li>Dhaka</li><li>Kathmandu</li></ol></td>
        </tr>
        <tr>
        </tr>
        </tbody>
        <tfoot> </tfoot>
    </table>
</body>
</html>
```

Column 1	Column 2
• India:	1. New Delhi
• USA:	2. Washington D.C.
• China	3. Beijing
• Bangladesh	4. Dhaka
• Nepal	5. Kathmandu

T5. Make a table that looks like this:

Country	Politicians	
Turkmenistan	President	Gurbanguly
	Berdimuhamedow	
	Vice-President	Raşit Meredow
Eritrea	President	Isaias Afwerki
	Vice-President	None

Ans

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lab2</title>
</head>
<body>
    <table border="1">
        <thead>
            <tr>
                <th>Country</th>
                <th colspan="2">Politicians</th>
            </tr>
        </thead>
        <tbody>
```

```

<tr>
    <th rowspan="3">Turkmenistan</th>
    <td>President</td>
    <td>Gurbanguly</td>
</tr>
<tr>

    <td></td>
    <td>Berdimuhamedow</td>
</tr>
<tr>
    <td>Vice-President</td>
    <td>Rasit Meredow</td>
</tr>
<tr>
    <th rowspan="2">Eritrea</th>
    <td>President</td>
    <td>Isaias Afwerki</td>
</tr>
<tr>
    <td>Vice-President</td>
    <td>None</td>
</tr>
</tbody>
</table>
</body>
</html>

```

Screenshot of a web browser window titled "Lab2" showing the rendered HTML table.

The browser's address bar displays the URL: 127.0.0.1:5500/first.html

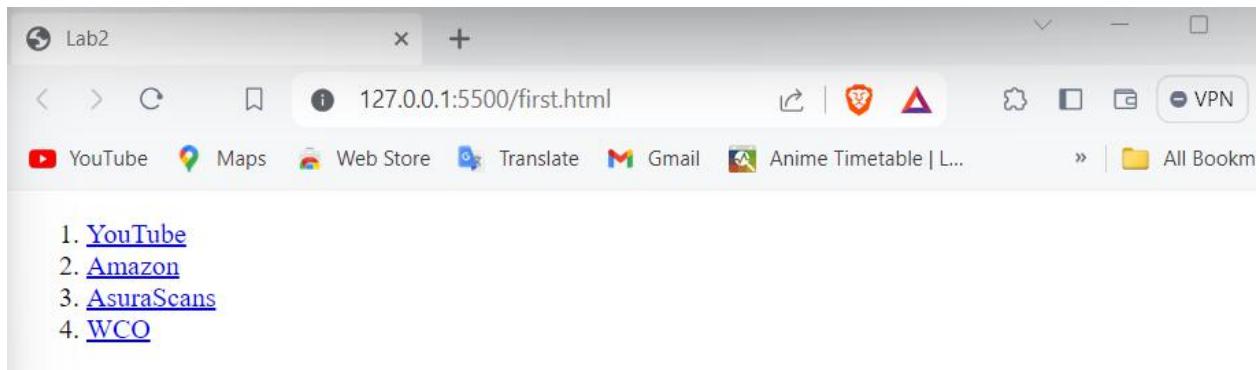
The rendered table has the following structure:

Country	Politicians	
Turkmenistan	President	Gurbanguly
		Berdimuhamedow
	Vice-President	Rasit Meredow
Eritrea	President	Isaias Afwerki
	Vice-President	None

T6. Make a webpage that links to four of your favorite websites.

Ans

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lab2</title>
</head>
<body>
    <nav>
        <ol>
            <li><a href="www.youtube.com">YouTube</a></li>
            <li><a href="www.amazon.com">Amazon</a></li>
            <li><a href="asuratoon.com">AsuraScans</a></li>
            <li><a href="wcofun.com">WCO</a></li>
        </ol>
    </nav>
</body>
</html>
```



T7. Make a webpage with a link at the bottom that when clicked will jump you all the way to the top of the page.

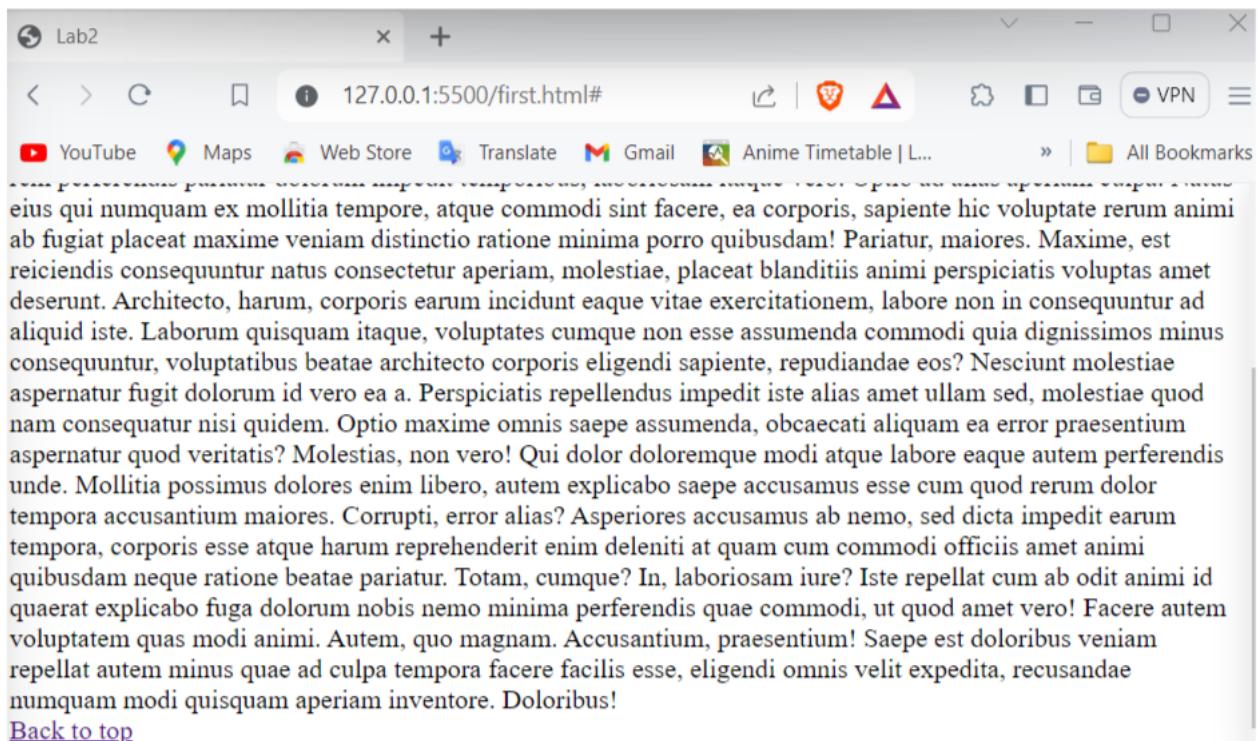
Ans

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Lab2</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<header><h1>Top</h1></header>
<div>Lorem*10</div>
<footer>
    <a href="#">Back to top</a>
</footer>
</body>
</html>

```



T8. Make a webpage with a link at the top of it that when clicked will jump all the way to the bottom of the page. At the bottom of the page there should be a link to jump back to the top of the page.

Ans

```

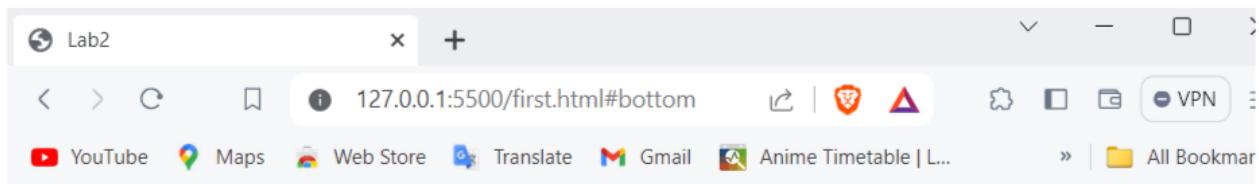
<!DOCTYPE html>
<html lang="en">

```

```

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lab2</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header><a href="#bottom">To the Bottom</a></header>
    <div>Lorem*10</div>
    <footer id="bottom"></footer>
</body>
</html>

```



To the Bottom

Lorem ipsum dolor sit amet consectetur adipisicing elit. Eum quos accusamus qui voluptatum voluptate suscipit neque illum repudiandae placeat aspernatur sit ducimus, iste nam ullam numquam facere. Facere, impedit eligendi! Accusamus porro molestiae fuga odio, aspernatur inventore possimus unde officia quo fugiat itaque cupiditate doloremque ex aperiam at? Corporis adipisci quia doloremque ratione ab corrupti pariatur autem velit maiores repudiandae? Maiores deleniti minima aliquam at voluptate velit optio reprehenderit! Vel in nisi est asperiores. Tempore alias dignissimos repellendus accusamus qui voluptatibus eius? Provident quos tenetur deserunt illum quis earum eligendi. Aliquam tempora cum illum, fuga quas odio ducimus, enim, qui modi provident accusamus possimus doloremque fugiat impedit cumque id quasi temporibus aspernatur rerum doloribus nisi? Tempora aliquid iure eius deserunt! Placeat, ratione. Autem in exercitationem praesentium sed fugit ipsam sapiente sit facilis magnam delectus provident ratione dignissimos sequi, eveniet laudantium voluptate dicta cumque quo minus. Esse fuga atque dolorum debitis. Repudiandae, consectetur earum voluptatum quasi optio incidunt, veniam distinctio aliquam dolore vel omnis velit vitae quae odio sequi. Rem accusamus animi reiciendis! Quibusdam sit deleniti doloribus corrupti ex eos repellat! Delectus hic debitis assumenda sapiente possimus numquam, nemo suscipit, inventore necessitatibus ad quo odit minus saepe repudiandae mollitia laudantium cum voluptatem rem dolore est corrupti. Est expedita ipsum porro corporis. Natus odio, laborum quos aspernatur consequatur eaque ad. Laboriosam expedita corrupti totam doloremque deleniti sunt iusto, ex quibusdam sed perspiciatis consectetur odio dolorum incidunt assumenda magni ipsa molestiae ipsum nihil!

T9. Make a webpage with an image that has a border of size 2, a width of 200, and a height of 200.

Ans

```

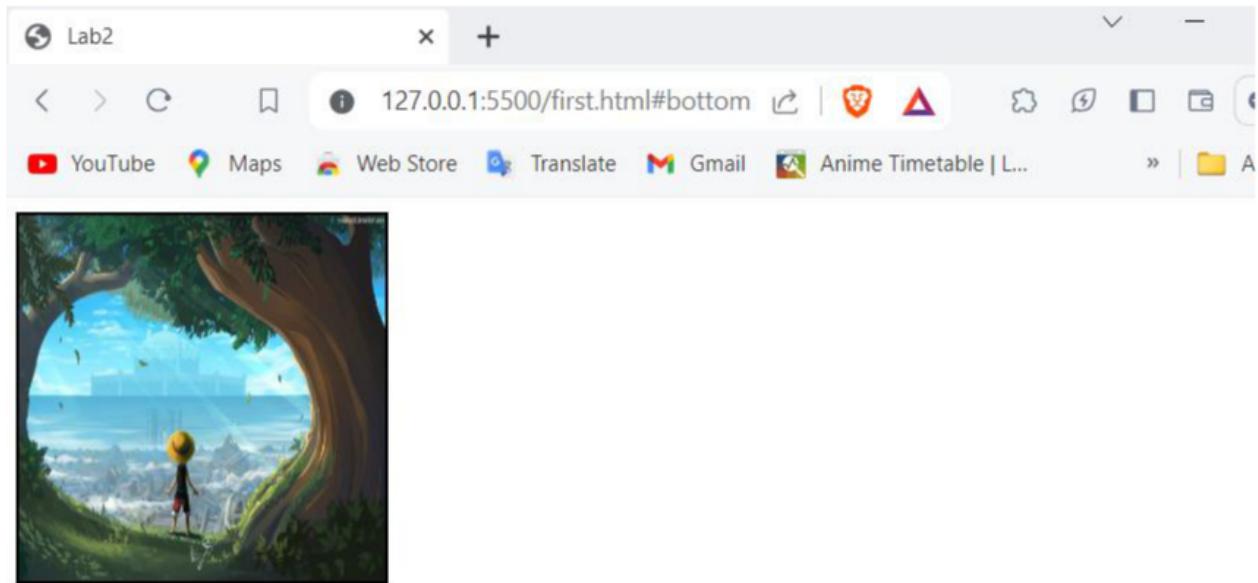
<!DOCTYPE html>
<html lang="en">
<head>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Lab2</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
    
</body>
</html>

```



**T10. Make a webpage with five different images. Skip two lines between each image (Use
 tag). Each image should have a caption.**

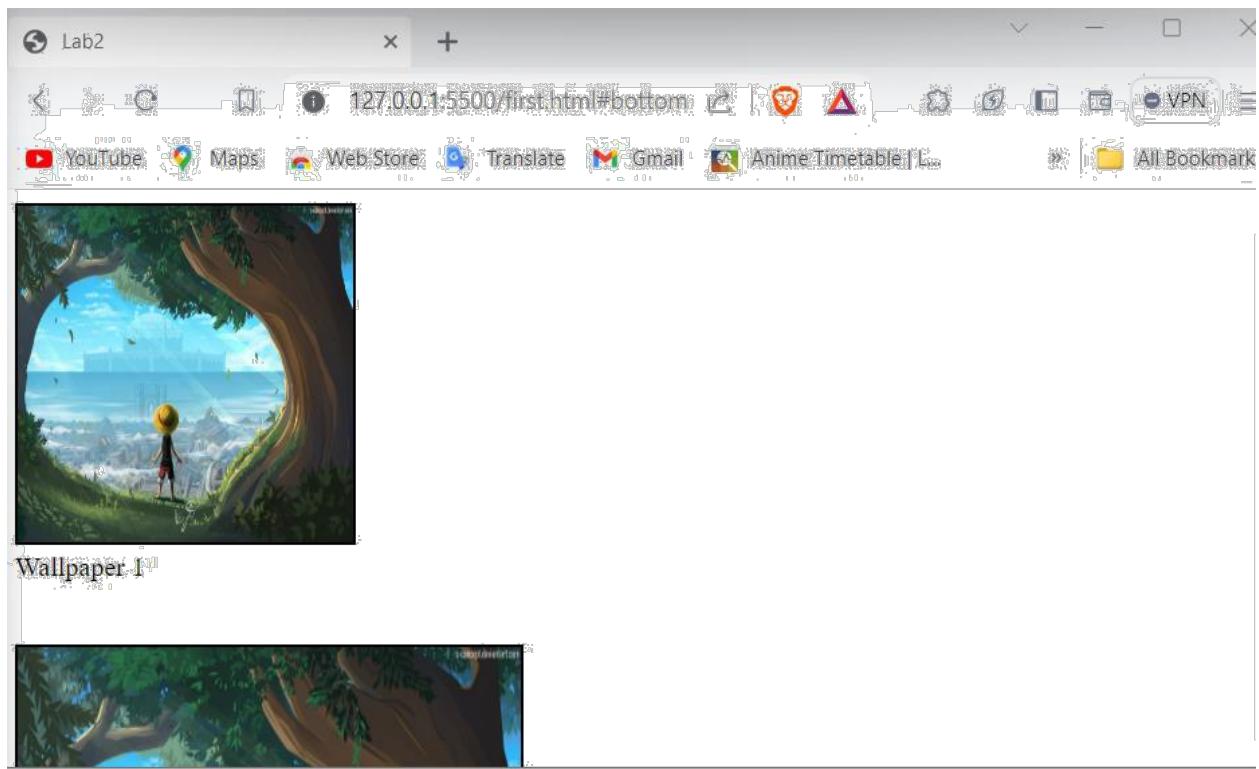
Ans

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lab2</title>
    <link rel="stylesheet"
href="style.css"> </head>

```

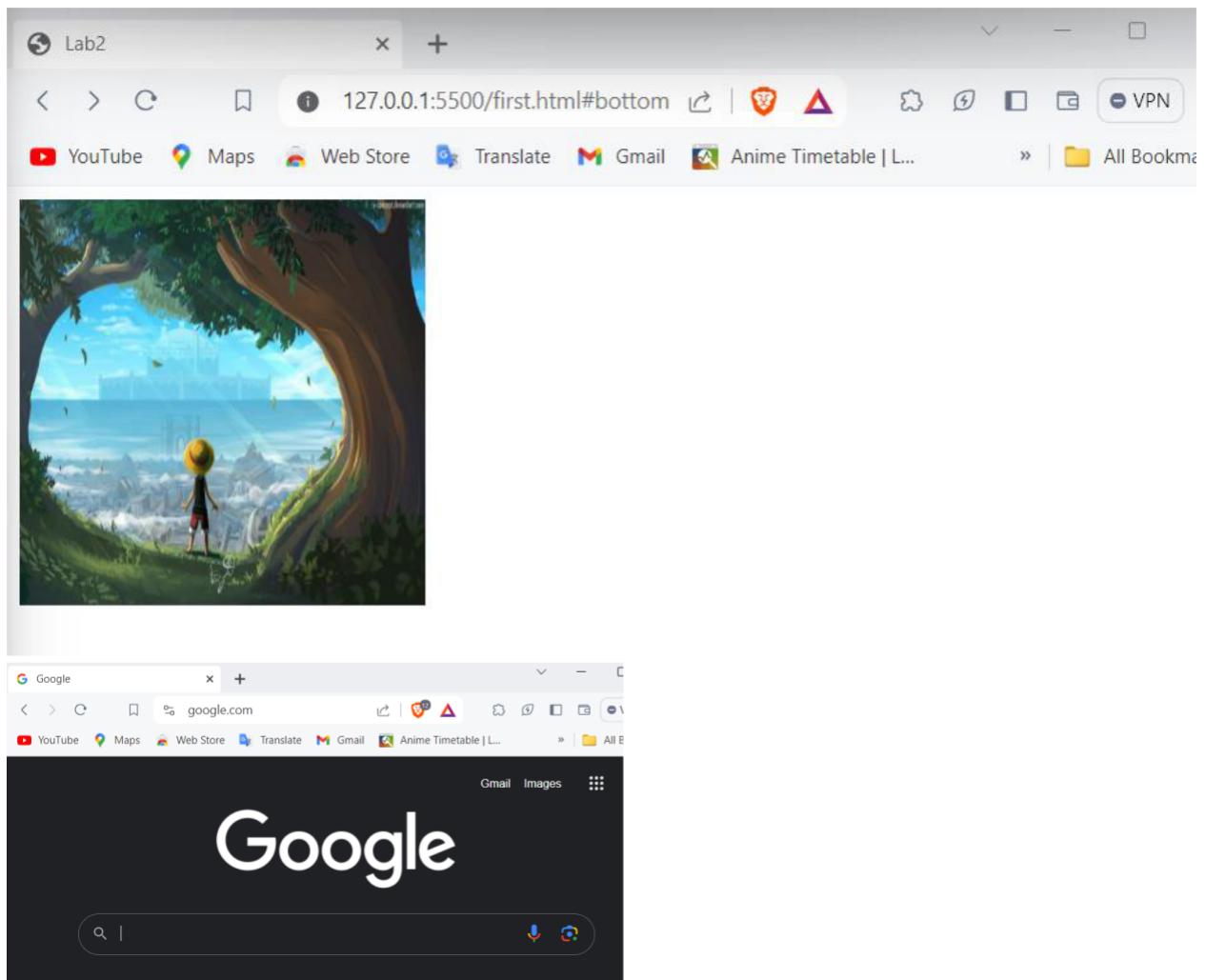
```
<body>
  
  <figcaption>Wallpaper 1</figcaption>
  <br><br>
  
  <figcaption>Wallpaper 2</figcaption>
  <br><br>
  
  <figcaption>Wallpaper 3</figcaption>
  <br><br>
  
  <figcaption>Wallpaper 4</figcaption>
  <br><br>
  
  <figcaption>Wallpaper 4</
  figcaption> </body>
</html>
```



T11. Make a webpage with an image that when clicked will link to a search engine of your choice.

Ans

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Lab2</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <a href="https://www.google.com">
        
    </a>
</body>
</html>
```



LAB ASSIGNMENT - 3

Name:- Pratyush Tiwari

Roll No. :- 22MC3024

Branch :- Mathematics and Computing

T1. Add a CSS rule and appropriate html to style every other line of the unordered list with

a light blue background color. Specify the background color using the rgb method.

- Eswatini
- Ecuador
- Tanzania
- Spain
- Slovakia
- Iraq

Ans.)

The screenshot shows a code editor interface with two tabs: "Assignment-3.html" and "# Assgn-3.css".

The "Assignment-3.html" tab contains the following HTML code:

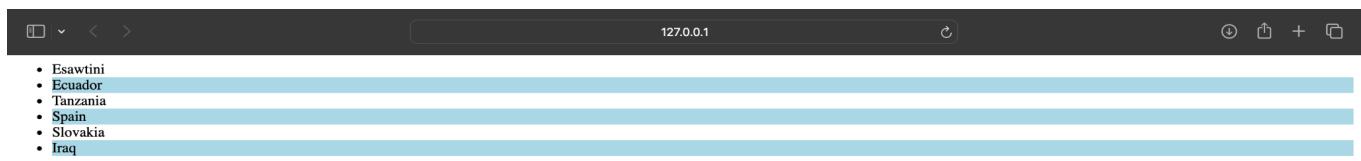
```
<!DOCTYPE html>
<html>
  <head>
    </head>
    <link rel="stylesheet" href="Assgn-3.css">
  <body>
    <ul>
      <li class="eswatini"> Eswatini</li>
      <li>Ecuador</li>
      <li class="tanzania">Tanzania</li>
      <li>Spain</li>
      <li class="slovakia">Slovakia</li>
      <li>Iraq</li>
    </ul>
  </body>
</html>
```

The "# Assgn-3.css" tab contains the following CSS code:

```
.eswatini {background-color: #ADD8E6;}
```

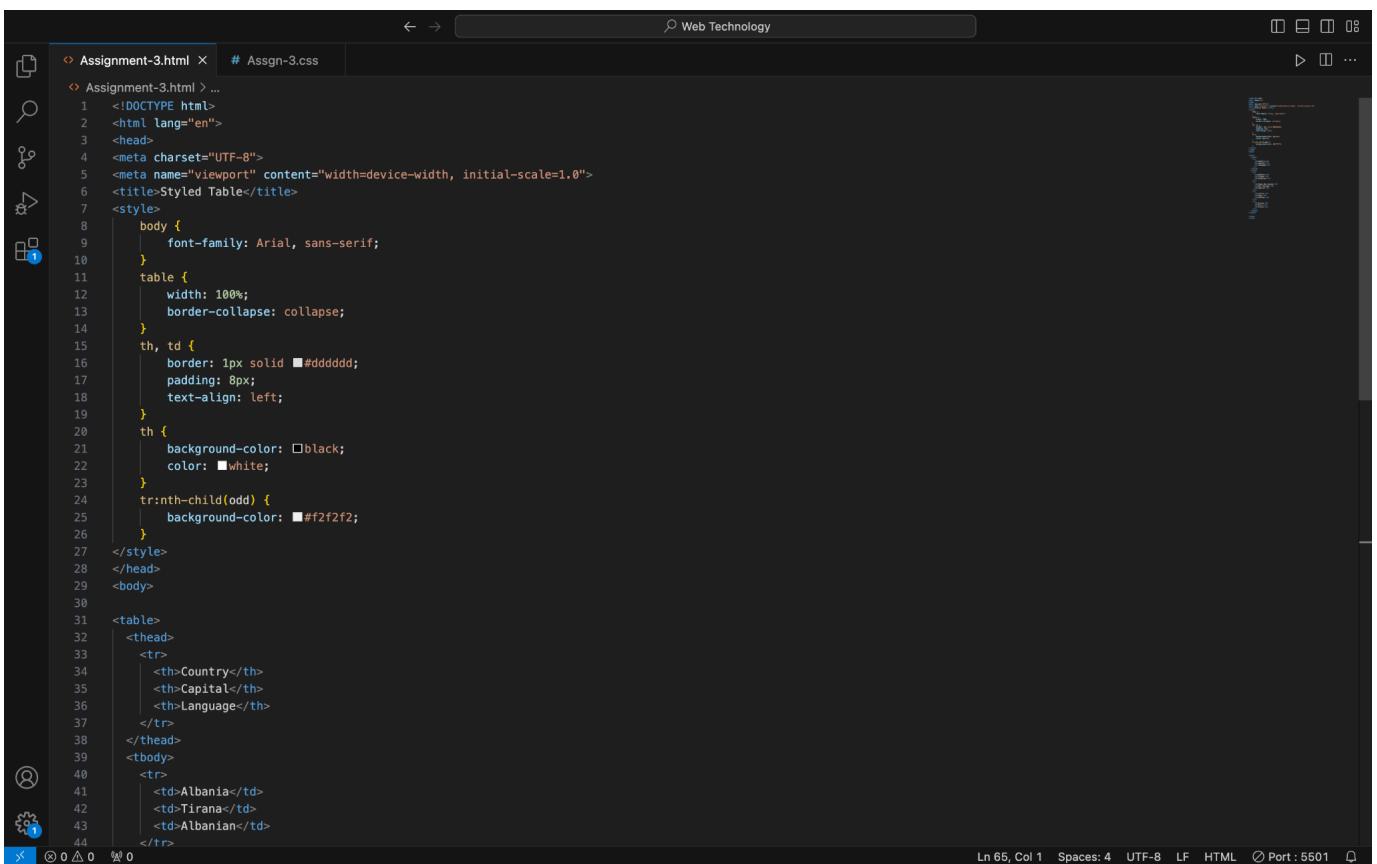
The code editor has a dark theme and includes icons for file operations like Open, Save, and Close, as well as a search bar at the top. The bottom status bar shows file details: Line 17, Column 8, Spaces: 4, UTF-8, LF, HTML, Port: 5501.

```
# Assgn-3.css > ...
# Assgn-3.css
1 ul li:nth-child(even) {
2   background-color: #rgb(173, 216, 230);
3 }
4
```

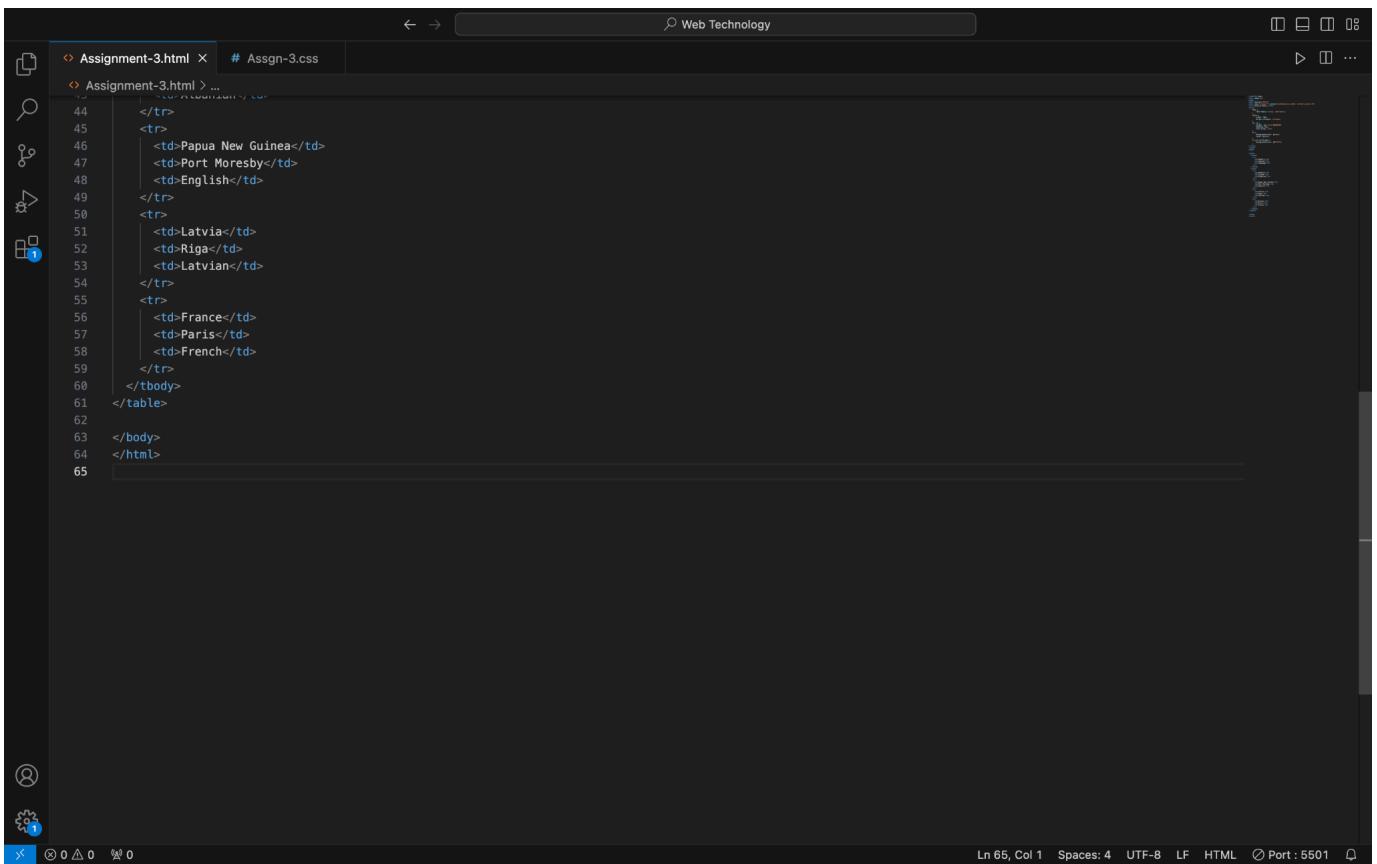


T2. Given the table in the question style the table so that the odd rows have a light gray background. In addition add a header row to the table and style that row so it has white text and a black background color. Change the font so that they are all 14 point sans serif of some kind.

Ans)



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Styled Table</title>
<style>
body {
    font-family: Arial, sans-serif;
}
table {
    width: 100%;
    border-collapse: collapse;
}
th, td {
    border: 1px solid black;
    padding: 8px;
    text-align: left;
}
th {
    background-color: black;
    color: white;
}
tr:nth-child(odd) {
    background-color: #f2f2f2;
}
</style>
</head>
<body>
<table>
<thead>
<tr>
<th>Country</th>
<th>Capital</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albania</td>
<td>Tirana</td>
<td>Albanian</td>
</tr>
<tr>
<td>Papua New Guinea</td>
<td>Port Moresby</td>
<td>English</td>
</tr>
<tr>
<td>Latvia</td>
<td>Riga</td>
<td>Latvian</td>
</tr>
<tr>
<td>France</td>
<td>Paris</td>
<td>French</td>
</tr>
</tbody>
</table>
</body>
</html>
```



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Styled Table</title>
<style>
body {
    font-family: Arial, sans-serif;
}
table {
    width: 100%;
    border-collapse: collapse;
}
th, td {
    border: 1px solid black;
    padding: 8px;
    text-align: left;
}
th {
    background-color: black;
    color: white;
}
tr:nth-child(odd) {
    background-color: #f2f2f2;
}
</style>
</head>
<body>
<table>
<thead>
<tr>
<th>Country</th>
<th>Capital</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albania</td>
<td>Tirana</td>
<td>Albanian</td>
</tr>
<tr>
<td>Papua New Guinea</td>
<td>Port Moresby</td>
<td>English</td>
</tr>
<tr>
<td>Latvia</td>
<td>Riga</td>
<td>Latvian</td>
</tr>
<tr>
<td>France</td>
<td>Paris</td>
<td>French</td>
</tr>
</tbody>
</table>
</body>
</html>
```

Country	Capital	Language
Albania	Tirana	Albanian
Papua New Guinea	Port Moresby	English
Latvia	Riga	Latvian
France	Paris	French

T3. Suppose you are working on a recipe website. Recipes have a title, a list of ingredients, and an instructions section. Make a webpage that shows the recipe for grilled cheese sandwiches with bacon and tomato. The ingredients should be displayed as an unordered list, with no bullets. Meat ingredients should have a light red background color, vegetables should have a light green background, and dairy products should have a light yellow. The title should be in a sans serif font of your choice, and the instructions should have a heading “Instructions” and the font for the instructions should be italic.

Ans.)

```
Assignment-3.html # Assgn-3.css
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Grilled Cheese Sandwich with Bacon and Tomato</title>
7 <style>
8   body {
9     font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
10    background-color: #f5f5f5;
11    margin: 0;
12    padding: 20px;
13  }
14  h1 {
15    font-size: 28px;
16    color: #333333;
17    text-align: center;
18    margin-bottom: 20px;
19  }
20  .recipe-container {
21    background-color: #ffffff;
22    border-radius: 8px;
23    padding: 20px;
24    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
25  }
26  h2 {
27    font-size: 20px;
28    color: #444444;
29    margin-top: 0;
30  }
31  ul {
32    list-style-type: none;
33    padding: 0;
34    margin: 0;
35  }
36  li {
37    margin-bottom: 10px;
38  }
39  .meat {
40    background-color: #ffc6b3;
41    padding: 5px 10px;
42    border-radius: 5px;
43  }
44  .vegetable {
```

Ln 93, Col 1

```
Assignment-3.html # Assgn-3.css
Assignment-3.html ...
44  .vegetable {
45    background-color: #c6ffb3;
46    padding: 5px 10px;
47    border-radius: 5px;
48  }
49  .dairy {
50    background-color: #ffffcc;
51    padding: 5px 10px;
52    border-radius: 5px;
53  }
54  ol {
55    padding-left: 20px;
56  }
57  p {
58    margin: 10px 0;
59  }
60  .instructions {
61    font-style: italic;
62  }
63 </style>
64 </head>
65 <body>
66
67 <div class="recipe-container">
68   <h1>Grilled Cheese Sandwich with Bacon and Tomato</h1>
69
70   <h2>Ingredients:</h2>
71   <ul>
72     <li class="meat">Bacon</li>
73     <li class="vegetable">Tomato slices</li>
74     <li class="dairy">Cheddar cheese slices</li>
75     <li class="dairy">Butter</li>
76     <li>Bread slices</li>
77   </ul>
78
79   <h2>Instructions:</h2>
80   <ol>
81     <li class="instructions">Preheat a skillet over medium heat.</li>
82     <li class="instructions">Spread butter on one side of each bread slice.</li>
83     <li class="instructions">Place one slice of bread, buttered side down, on the skillet.</li>
84     <li class="instructions">Layer with cheese slices, bacon, and tomato slices.</li>
85     <li class="instructions">Top with another bread slice, buttered side up.</li>
86     <li class="instructions">Cook until golden brown on both sides and cheese is melted, flipping halfway through.</li>
```

```
<Assignment-3.html> # Asgn-3.css
Assignment-3.html > ...
75     <ul class="list-group">
76         <li>Bread slices</li>
77     </ul>
78
79     <h2>Instructions:</h2>
80     <ol>
81         <li><span class="list-item">Preheat a skillet over medium heat.</span></li>
82         <li><span class="list-item">Spread butter on one side of each bread slice.</span></li>
83         <li><span class="list-item">Place one slice of bread, buttered side down, on the skillet.</span></li>
84         <li><span class="list-item">Layer with cheese slices, bacon, and tomato slices.</span></li>
85         <li><span class="list-item">Top with another bread slice, buttered side up.</span></li>
86         <li><span class="list-item">Cook until golden brown on both sides and cheese is melted, flipping halfway through.</span></li>
87         <li><span class="list-item">Remove from skillet, slice, and serve hot.</span></li>
88     </ol>
89 </div>
90
91 </body>
92 </html>
93
```

Ln 93, Col 1 Spaces: 4 UTF-8 LF HTML Port: 5501

Grilled Cheese Sandwich with Bacon and Tomato

Ingredients:

- Bacon
- Tomato slices
- Cheddar cheese slices
- Butter
- Bread slices

Instructions:

1. Preheat a skillet over medium heat.
2. Spread butter on one side of each bread slice.
3. Place one slice of bread, buttered side down, on the skillet.
4. Layer with cheese slices, bacon, and tomato slices.
5. Top with another bread slice, buttered side up.
6. Cook until golden brown on both sides and cheese is melted, flipping halfway through.
7. Remove from skillet, slice, and serve hot.

T4. Using CSS and HTML Make a webpage that has two columns. Each column should use half of the width of the page. The left half should have a light gray background and

the right half should have a light green background. The left half should have a list of the 5 best selling books in Amazon's kindle store, and the right should have a list of your five favorite celebrities or athletes.

Ans.)

A screenshot of a code editor showing two files: Assignment-3.html and Assgn-3.css. The Assgn-3.css file contains the following CSS:

```
Assignment-3.html # Assgn-3.css

body {
    font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f9f9f9;
}
.container {
    display: flex;
    width: 100%;
    justify-content: space-around;
    padding: 20px;
}
.column {
    width: 45%;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.left-column {
    background-color: #ffe6e6; /* light pink */
}
.right-column {
    background-color: #e6ffe6; /* light green */
}
ul {
    padding: 0;
    margin: 0;
    list-style: none;
}
li {
    margin-bottom: 10px;
}
h2 {
    margin-top: 0;
    text-align: center;
    color: #333333;
}
```

A screenshot of a code editor showing two files: Assignment-3.html and Assgn-3.css. The Assignment-3.html file contains the following HTML:

```
Assignment-3.html # Assgn-3.css

...
.celebrity {
    font-weight: bold;
}
</style>
</head>
<body>
<div class="container">
    <div class="column left-column">
        <h2>Best Selling Books on Kindle</h2>
        <ul>
            <li>The Silent Patient by Alex Michaelides</li>
            <li>Where the Crawdads Sing by Delia Owens</li>
            <li>Becoming by Michelle Obama</li>
            <li>Girl, Wash Your Face by Rachel Hollis</li>
            <li>The Night Circus by Erin Morgenstern</li>
        </ul>
    </div>
    <div class="column right-column">
        <h2>Favorite Celebrities/Athletes</h2>
        <ul>
            <li><span class="celebrity">Lionel Messi</span></li>
            <li>Rihanna</li>
            <li><span class="celebrity">Tom Hanks</span></li>
            <li>Serena Williams</li>
            <li>Leonardo DiCaprio</li>
        </ul>
    </div>
</div>
</body>
</html>
```

The Silent Patient by Alex Michaelides
Where the Crawdads Sing by Delia Owens
Becoming by Michelle Obama
Girl, Wash Your Face by Rachel Hollis
The Night Circus by Erin Morgenstern

Lionel Messi
Rihanna
Tom Hanks
Serena Williams
Leonardo DiCaprio

T5. Using the nav element, make a navbar for at the top of a sample webpage. Add at least three links to the navbar using an unordered list that is displayed as inline rather than block. Make the navbar have a light blue background color, but change the background of the title to a darker blue when the mouse hovers over it.

Ans.)

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Sample Webpage with Navbar</title>
<style>
body {
    font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f5f5f5;
}
nav {
    background-color: #333333; /* dark gray */
    padding: 10px 0;
    text-align: center;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}
li {
    display: inline;
    margin-right: 20px;
}
li:first-child {
    margin-left: 20px;
}
a {
    text-decoration: none;
    color: #ffffff;
    transition: color 0.3s ease;
}
a:hover {
    color: #a9d1ff; /* light blue */
}
h1 {
    display: inline-block;
    margin: 0;
    padding: 10px 20px;
    background-color: #333333; /* dark gray */
}
```

```
Assignment-3.html # Assgn-3.css
Assignment-3.html > html > body > nav > h1
40   h1 {
41     display: inline-block;
42     margin: 0;
43     padding: 10px 20px;
44     background-color: #333333; /* dark gray */
45     color: #ffffff;
46     font-size: 24px;
47     font-weight: bold;
48     transition: background-color 0.3s ease;
49   }
50   h1:hover {
51     background-color: #286090; /* darker blue */
52   }
53   .content {
54     padding: 20px;
55   }
56 </style>
57 </head>
58 <body>
59
60 <nav>
61   <h1>Hello</h1>
62   <ul>
63     <li><a href="#">Home</a></li>
64     <li><a href="#">About</a></li>
65     <li><a href="#">Contact</a></li>
66   </ul>
67 </nav>
68
69 <div class="content">
70   <p>My name is Pratyush Tiwari and I am a second year undergraduate student at RGIPT studying Mathematics and Computing.</p>
71 </div>
72
73 </body>
74 </html>
75
```

127.0.0.1 Ln 61

Hello

Home About Contact

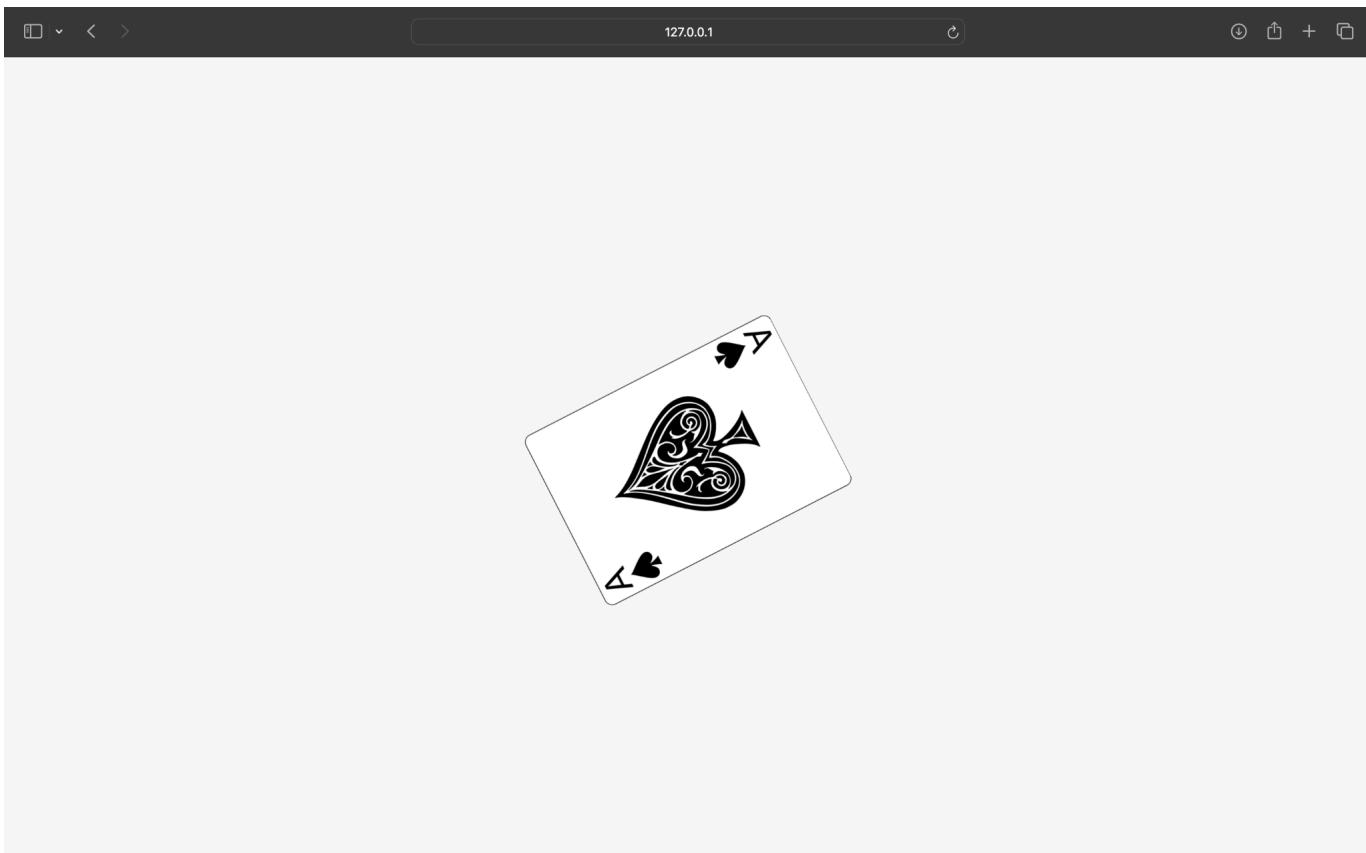
My name is Pratyush Tiwari and I am a second year undergraduate student at RGIPT studying Mathematics and Computing.

T6. Using the image of the playing card, use transformations and animations to make the playing card spin around endlessly.
Ans.)

```
Assignment-3.html # Assgn-3.css

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Spinning Card</title>
7 <link rel="stylesheet" href="Assgn-3.css">
8 </head>
9 <body>
10 <div class="card">
11   
12 </div>
13
14
15 </body>
16 </html>
17
```

```
# Assgn-3.css > .card
1 body {
2   display: flex;
3   justify-content: center;
4   align-items: center;
5   height: 100vh;
6   margin: 0;
7   background-color: #f5f5f5;
8 }
9
10 .card {
11   width: 200px;
12   height: 300px;
13   position: relative;
14   animation: spin 5s linear infinite;
15 }
16
17 .card img {
18   width: 100%;
19   height: 100%;
20   object-fit: contain;
21 }
22
23 @keyframes spin {
24   0% {
25     transform: rotate(0deg);
26   }
27   100% {
28     transform: rotate(360deg);
29   }
30 }
31
```



LAB ASSIGNMENT - 4

Name:- Pratyush Tiwari

Roll No. :- 22MC3024

Branch :- Mathematics and Computing

T1. Create a webpage that has a heading and a button. Write a script to change the background color and text when the button is clicked.

Ans.)

The screenshot shows a code editor on the left and a browser preview on the right. The code editor displays the following HTML and JavaScript:

```
index.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>T1</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      text-align: center;
11    }
12    h1 {
13      color: #333;
14    }
15   #changeButton {
16     padding: 10px 20px;
17     font-size: 16px;
18     background-color: #007bff;
19     color: #fff;
20     border: none;
21     cursor: pointer;
22     border-radius: 5px;
23   }
24   </style>
25 </head>
26 <body>
27   <h1 id="heading">Hello!</h1>
28   <button id="changeButton">Click Here</button>
29
30   <script>
31     document.getElementById('changeButton').addEventListener('click',
32       function() {
33         // Change background color
34         document.body.style.backgroundColor = '#F5A302';
35         // Change text
36         document.getElementById('heading').textContent = 'I am Pratyush';
37       });
38   </script>
39 </body>
40 </html>
```

The browser preview shows a white page with a central "Hello!" heading and a blue "Click Here" button. The console tab at the bottom of the code editor shows no errors.

The screenshot shows a code editor on the left and a browser preview on the right. The code editor displays the following modified HTML and JavaScript:

```
index.html
18   background-color: #007bff;
19   color: #fff;
20   border: none;
21   cursor: pointer;
22   border-radius: 5px;
23 }
24 </style>
25 </head>
26 <body>
27   <h1 id="heading">Hello!</h1>
28   <button id="changeButton">Click Here</button>
29
30 <script>
31   document.getElementById('changeButton').addEventListener('click',
32     function() {
33       // Change background color
34       document.body.style.backgroundColor = '#F5A302';
35       // Change text
36       document.getElementById('heading').textContent = 'I am Pratyush';
37     });
38 </script>
39 </body>
40 </html>
```

The browser preview shows a yellow page with a central "I am Pratyush" heading and a blue "Click Here" button. The console tab at the bottom of the code editor shows no errors.

T2. For T1, experiment with the Javascript code by trying the following things:

a. Make the background light green.

The screenshot shows a code editor with an open file named `index.html`. The code contains HTML, CSS, and JavaScript. The CSS styles the body and h1 elements. The JavaScript adds a click event listener to a button that changes the background color to light green and updates the heading text.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>T1</title>
7 <style>
8   body {
9     font-family: Arial, sans-serif;
10    text-align: center;
11  }
12  h1 {
13    color: #333;
14  }
15  #changeButton {
16    padding: 10px 20px;
17    font-size: 16px;
18    background-color: #0007bff;
19    color: #fff;
20    border: none;
21    cursor: pointer;
22    border-radius: 5px;
23  }
24 </style>
25 </head>
26 <body>
27   <h1 id="heading">Hello!</h1>
28   <button id="changeButton">Click Here</button>
29
30 <script>
31   document.getElementById('changeButton').addEventListener('click', function() {
32     // Change background color to light green
33     document.body.style.backgroundColor = '#90EE90';
34     // Change text
35     document.getElementById('heading').textContent = 'I am Pratyush';
36   });
37 </script>
38 </body>
39 </html>
```

The screenshot shows the same code editor and browser preview as the previous one, but the browser now displays a light green background with the text "I am Pratyush". This indicates that the JavaScript has successfully changed the background color and updated the heading text.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta name="viewport" content="width=device-width, initial-scale=1.0">
5   <style>
6     body {
7       font-family: Arial, sans-serif;
8       text-align: center;
9     }
10    h1 {
11      color: #333;
12    }
13    #changeButton {
14      padding: 10px 20px;
15      font-size: 16px;
16      background-color: #0007bff;
17      color: #fff;
18      border: none;
19      cursor: pointer;
20      border-radius: 5px;
21    }
22 </style>
23 </head>
24 <body>
25   <h1 id="heading">Hello!</h1>
26   <button id="changeButton">Click Here</button>
27
28 <script>
29   document.getElementById('changeButton').addEventListener('click', function() {
30     // Change background color to light green
31     document.body.style.backgroundColor = '#90EE90';
32     // Change text
33     document.getElementById('heading').textContent = 'I am Pratyush';
34   });
35 </script>
36 </body>
37 </html>
```

b. Add a line that makes the text red.

The screenshot shows a code editor with an open file named `index.html`. The code contains HTML, CSS, and JavaScript. The browser window shows the rendered page with the text "Hello!" and a blue button labeled "Click Here". The browser's developer tools are open, showing the local and network URLs.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>T1</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      text-align: center;
11    }
12   h1 {
13     color: #333;
14   }
15   #changeButton {
16     padding: 10px 20px;
17     font-size: 16px;
18     background-color: #007bff;
19     color: #fff;
20     border: none;
21     cursor: pointer;
22     border-radius: 5px;
23   }
24 </style>
25 </head>
26 <body>
27   <h1 id="heading">Hello!</h1>
28   <button id="changeButton">Click Here</button>
29
30 <script>
31   document.getElementById('changeButton').addEventListener('click', function() {
32     // Change background color to light green
33     document.body.style.backgroundColor = '#90EE90';
34     // Change text
35     document.getElementById('heading').textContent = 'I am Pratyush';
36     // Change text color to red
37     document.getElementById('heading').style.color = 'red';
38   });
39 </script>
40 </body>
41 </html>
```

The screenshot shows a code editor with an open file named `index.html`. The code is identical to the one in the previous screenshot, but the browser window now displays the text "I am Pratyush" in red, indicating that the JavaScript has successfully changed the text content and color of the heading element.

```
15   #changeButton {
16     padding: 10px 20px;
17     font-size: 16px;
18     background-color: #007bff;
19     color: #fff;
20     border: none;
21     cursor: pointer;
22     border-radius: 5px;
23   }
24 </style>
25 </head>
26 <body>
27   <h1 id="heading">Hello!</h1>
28   <button id="changeButton">Click Here</button>
29
30 <script>
31   document.getElementById('changeButton').addEventListener('click', function() {
32     // Change background color to light green
33     document.body.style.backgroundColor = '#90EE90';
34     // Change text
35     document.getElementById('heading').textContent = 'I am Pratyush';
36     // Change text color to red
37     document.getElementById('heading').style.color = 'red';
38   });
39 </script>
40 </body>
41 </html>
```

c. Change the `fontSize` to 48pt.

index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>T1</title>
7 <style>
8   body {
9     font-family: Arial, sans-serif;
10    text-align: center;
11  }
12  h1 {
13    color: #333;
14  }
15  #changeButton {
16    padding: 10px 20px;
17    font-size: 16px;
18    background-color: #007bff;
19    color: #fff;
20    border: none;
21    cursor: pointer;
22    border-radius: 5px;
23  }
24 </style>
25 </head>
26 <body>
27   <h1 id="heading">Hello!</h1>
28   <button id="changeButton">Click Here</button>
29
30 <script>
31   document.getElementById('changeButton').addEventListener('click', function() {
32     // Change background color to light green
33     document.body.style.backgroundColor = '#90EE90';
34     // Change text
35     document.getElementById('heading').textContent = 'I am Pratyush';
36     // Change text color to red
37     document.getElementById('heading').style.color = 'red';
38     // Change font size to 48pt
39     document.getElementById('heading').style.fontSize = '48pt';
40   });
41 </script>
42 </body>
43 </html>
```

Ln 44, Col 1 • Spaces: 2 History

AI HTML

Webview

Hello!

Click Here

index.html

```
17   font-size: 16px;
18   background-color: #007bff;
19   color: #fff;
20   border: none;
21   cursor: pointer;
22   border-radius: 5px;
23 }
24 </style>
25 </head>
26 <body>
27   <h1 id="heading">Hello!</h1>
28   <button id="changeButton">Click Here</button>
29
30 <script>
31   document.getElementById('changeButton').addEventListener('click', function() {
32     // Change background color to light green
33     document.body.style.backgroundColor = '#90EE90';
34     // Change text
35     document.getElementById('heading').textContent = 'I am Pratyush';
36     // Change text color to red
37     document.getElementById('heading').style.color = 'red';
38     // Change font size to 48pt
39     document.getElementById('heading').style.fontSize = '48pt';
40   });
41 </script>
42 </body>
43 </html>
```

Generate

Ln 44, Col 1 • Spaces: 2 History

AI HTML

Webview

I am Pratyush

Click Here

d. Change the wording inside the <h1> tag in the Javascript.

The screenshot shows a code editor with an open file named 'index.html'. The code contains HTML, CSS, and JavaScript. The CSS styles the body and a button. The JavaScript adds a click event listener to the button, changing the background color of the body to light green and the text content of the heading to 'I am in Mathematics and Computing branch'. The browser preview window shows the heading 'Hello!' and a blue button labeled 'Click Here'.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>T1</title>
<style>
body {
    font-family: Arial, sans-serif;
    text-align: center;
}
h1 {
    color: #333;
}
#changeButton {
    padding: 10px 20px;
    font-size: 16px;
    background-color: #007bff;
    color: #fff;
    border: none;
    cursor: pointer;
    border-radius: 5px;
}
</style>
</head>
<body>
<h1 id="heading">Hello!</h1>
<button id="changeButton">Click Here</button>
<script>
document.getElementById('changeButton').addEventListener('click', function() {
    // Change background color to light green
    document.body.style.backgroundColor = '#90EE90';
    // Change text
    document.getElementById('heading').textContent = 'I am in Mathematics and
Computing branch';
    // Change text color to red
    document.getElementById('heading').style.color = 'red';
    // Change font size to 48pt
    document.getElementById('heading').style.fontSize = '48pt';
});
</script>
</body>
</html>
```

The screenshot shows a code editor with an open file named 'index.html'. The code is identical to the one in the previous screenshot, except for the text content of the heading element. The browser preview window shows the heading 'I am in Mathematics and Computing branch' in large red text, and a blue button labeled 'Click Here'.

```
<!-- ... -->
<style>
body {
    border: none;
    cursor: pointer;
    border-radius: 5px;
}
</style>
</head>
<body>
<h1 id="heading">Hello!</h1>
<button id="changeButton">Click Here</button>
<script>
document.getElementById('changeButton').addEventListener('click', function() {
    // Change background color to light green
    document.body.style.backgroundColor = '#90EE90';
    // Change text
    document.getElementById('heading').textContent = 'I am in Mathematics and
Computing branch';
    // Change text color to red
    document.getElementById('heading').style.color = 'red';
    // Change font size to 48pt
    document.getElementById('heading').style.fontSize = '48pt';
});
</script>
</body>
</html>
```

e. Add <button>Click Me</button> after the closing </h1>.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>T1</title>
7 <style>
8   body {
9     font-family: Arial, sans-serif;
10    text-align: center;
11  }
12  h1 {
13    color: #333;
14  }
15  #changeButton {
16    padding: 10px 20px;
17    font-size: 16px;
18    background-color: #007bff;
19    color: #fff;
20    border: none;
21    cursor: pointer;
22    border-radius: 5px;
23  }
24 </style>
25 </head>
26 <body>
27 <h1 id="heading">Hello!<button id="clickMeButton">Click Me</button></h1>
28 <button id="changeButton">Click Here</button>
29
30
31 <script>
32   document.getElementById('changeButton').addEventListener('click', function() {
33     // Change background color to light green

```

```

30   // Change text color to red
31   document.body.style.backgroundColor = '#90EE90';
32   // Change text
33   document.getElementById('heading').textContent = 'I am in Mathematics and
34   Computing branch';
35   // Change font size to 48pt
36   document.getElementById('heading').style.fontSize = '48pt';
37
38   // Create a new button element
39   var clickMeButton = document.createElement('button');
40   // Set the button's text content
41   clickMeButton.textContent = 'Click Me';
42   // Set the button's ID
43   clickMeButton.id = 'clickMeButton';
44   // Append the button to the document body
45   document.getElementById('heading').appendChild(clickMeButton);
46
47   // Create a line break element
48   var lineBreak = document.createElement('br');
49   // Append the line break to the document body
50   document.getElementById('heading').appendChild(lineBreak);
51
52   // Create a line break element
53   var lineBreak2 = document.createElement('br');
54   // Append the line break to the document body
55   document.getElementById('heading').appendChild(lineBreak2);
56
57 </script>
58 </body>
59 </html>

```

T3. For T1, set the color of the background directly with the class attribute. Generate alerts before and after assigning the class attribute to document object.

Ans.)

The screenshot shows a code editor with an open file named 'index.html'. The code contains CSS and HTML. A modal dialog box is displayed in the center of the screen with the message 'Color Going to be changed' and an 'OK' button. To the right of the dialog, a browser window shows the rendered page. The page has a light green background, a white h1 header with black text, and a blue button labeled 'Click Here'. The browser interface includes tabs for 'index.html', 'Console', and 'Shell'.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>T1</title>
<style>
body {
    font-family: Arial, sans-serif;
    text-align: center;
}
.light-green-bg {
    background-color: #90EE90;
}
h1 {
    color: #333;
}
#changeButton {
    padding: 10px 20px;
    font-size: 16px;
    background-color: #007bff;
    color: #fff;
    border: none;
    cursor: pointer;
    border-radius: 5px;
}
</style>
</head>
<body>
<h1 id="heading">Hello!</h1>
<button id="changeButton">Click Here</button>
</body>
```

T4. Design a page having contents within and outside the main section. The page must contain a button that when clicked replaces the content of the main section with another content having different style.

Ans.)

The screenshot shows a code editor with an open file named 'index.html'. The code includes a 'header' section with a blue background and white text, and a 'section' section with a white background and black text. A blue button labeled 'Change Content' is located between them. To the right, a browser window displays the rendered page. The 'header' is a solid blue bar with the word 'Hello' in white. Below it is a white box containing the text 'Original Content' and 'This is the original content of the main section.' A blue button labeled 'Change Content' is visible above the content box. The browser interface includes tabs for 'index.html', 'Webview', 'Console', and 'Shell'.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Dynamic Content Replacement</title>
<style>
body {
    font-family: Arial, sans-serif;
    text-align: center;
}
header {
    background-color: #007bff;
    color: #fff;
    padding: 10px;
}
section {
    padding: 20px;
    border: 2px solid #ccc;
    margin: 20px;
}
#changeButton {
    padding: 10px 20px;
    font-size: 16px;
    background-color: #007bff;
    color: #fff;
    border: none;
    cursor: pointer;
    border-radius: 5px;
}
.new-content {
    border: 1px solid #ccc;
    padding: 10px;
}
</style>
</head>
<body>
<header>Hello</header>
<section>
    This is the original content of the main section.
</section>
<button id="changeButton">Change Content</button>
</body>
```

```

index.html
34   color: #333;
35   font-size: 20px;
36   padding: 20px;
37 }
38 </style>
39 </head>
40 <body>
41 <header>
42   <h1>Hello </h1>
43 </header>
44 <br>
45 <button id="changeButton">Change Content</button>
46 <section id="mainSection">
47   <h2>Original Content</h2>
48   <p>This is the original content of the main section.</p>
49 </section>
50
51 <script>
52   document.getElementById('changeButton').addEventListener('click', function() {
53     var mainSection = document.getElementById('mainSection');
54     var newContent = document.createElement('section');
55     newContent.classList.add('new-content');
56     newContent.innerHTML =
57       `<h2>New Content</h2>
58       <p>This is the new content of the main section with different
59       styles.</p>`;
60     mainSection.parentNode.replaceChild(newContent, mainSection);
61   });
62 </script>
63 </body>
64 </html>

```

Ln 20, Col 18 • Spaces: 2 History ⚙️

Console Shell ⚙️

T5. Consider a h1 heading lying outside the main section in the page. Try changing the color of h1 in at least 3 different ways.

Ans.)

```

index.html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Changing Color of h1</title>
7  <style>
8    body {
9      font-family: Arial, sans-serif;
10     text-align: center;
11   }
12   h1 {
13     color: #333; /* Default color */
14   }
15   .red-color {
16     color: red; /* Red color using a class */
17   }
18 </style>
19 </head>
20 <body>
21   <h1 id="heading">Heading Outside Main Section</h1>
22
23   <!-- Method 1: Using JavaScript -->
24   <button onclick="changeColorJS()">Change Color (JavaScript)</button>
25
26   <!-- Method 2: Using CSS class -->
27   <button onclick="changeColorClass()">Change Color (Class)</button>
28
29   <!-- Method 3: Using Inline Styling -->
30   <button onclick="changeColorInline()">Change Color (Inline)</button>
31
32 <script>

```

Ln 53, Col 1 • Spaces: 2 History ⚙️

Console Shell ⚙️

```
index.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Changing Color of h1</title>
<style>
body {
    font-family: Arial, sans-serif;
    text-align: center;
}
h1 {
    color: #333; /* Default color */
}
.red-color {
    color: red; /* Red color using a class */
}
</style>
</head>
<body>
<h1 id="heading">Heading Outside Main Section</h1>

<!-- Method 1: Using JavaScript -->
<button onclick="changeColorJS()">Change Color (JavaScript)</button>

<!-- Method 2: Using CSS class -->
<button onclick="changeColorClass()">Change Color (Class)</button>

<!-- Method 3: Using Inline Styling -->
<button onclick="changeColorInline()">Change Color (Inline)</button>

</script>
```

```
index.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Changing Color of h1</title>
<style>
body {
    font-family: Arial, sans-serif;
    text-align: center;
}
h1 {
    color: #333; /* Default color */
}
.red-color {
    color: red; /* Red color using a class */
}
</style>
</head>
<body>
<h1 id="heading">Heading Outside Main Section</h1>

<!-- Method 1: Using JavaScript -->
<button onclick="changeColorJS()">Change Color (JavaScript)</button>

<!-- Method 2: Using CSS class -->
<button onclick="changeColorClass()">Change Color (Class)</button>

<!-- Method 3: Using Inline Styling -->
<button onclick="changeColorInline()">Change Color (Inline)</button>

</script>
```

T6. Design a webpage using grid layout that displays the following:
The first row shows icons of various web browsers and the second row shows the latest version number.

Ans.)

index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Browser Versions</title>
6 </head>
7 <body>
8   <div class="container">
9     <div class="browser-icon"></div>
10    <div class="browser-icon"></div>
11    <div class="browser-icon"></div>
12    <div class="browser-icon"></div>
13    <div class="browser-icon"></div>
14  </div>
15  <div class="browser-version">Version 57.0</div>
16  <div class="browser-version">Version 16.0</div>
17  <div class="browser-version">Version 52.0</div>
18  <div class="browser-version">Version 10</div>
19  <div class="browser-version">Version 63.0</div>
20  <div class="browser-version">Version 44</div>
21 </body>
22 </html>
```

AI HTML

Ln 41, Col 50 • Spaces: 2 History

Webview

srcdoc

Devtools

New tab

Version 57.0 Version 16.0 Version 52.0 Version 10 Version 44

index.html

```
31 <div class="container">
32   <div class="browser-icon"></div>
33   <div class="browser-icon"></div>
34   <div class="browser-icon"></div>
35   <div class="browser-icon"></div>
36   <div class="browser-icon"></div>
37
38  <div class="browser-version">Version 57.0</div>
39  <div class="browser-version">Version 16.0</div>
40  <div class="browser-version">Version 52.0</div>
41  <div class="browser-version">Version 10</div>
42  <div class="browser-version">Version 63.0</div>
43  <div class="browser-version">Version 44</div>
44
45 </body>
46 </html>
```

AI HTML

Ln 41, Col 50 • Spaces: 2 History

Webview

srcdoc

Devtools

New tab

Version 57.0 Version 16.0 Version 52.0 Version 10 Version 44

Web Technology Lab-5

Name: Pratyush Tiwari

Roll No: 22MC3024

Branch: Mathematics & Computing

T1. Make a simple web page that contains an h2 with the word “Hello” a text input box, and a button. When the user types a word or phrase into the input box and presses the button, replace the old h2 with the word entered. Using animation, make the word spin.

Solution :

Hello
 Change Text

T2. Make a simple web page that contains a button and a paragraph with the id of count. Whenever this button is pressed increment the count by 1 and update the paragraph text. Also update the font size so that as the number gets larger, so does the font.

Solution :

0

T3. Repeat the previous exercise but make a list of numbers. In this case you will not be able to simply update the innerHTML of the paragraph, you will need to use

the `document.createElement()` and `document.appendChild()` functions to add a new list item.

Increment Count

- 1
- 2
- 3
- 4
- 5
- 6

T4. Given the following html. Every time the button is pressed you should add a row to the table, where the new row of the table contains the sum of the previous two rows. You should make use of the `lastChild`, `previousSibling`, and `innerText` attributes in this exercise.

Add Row

1	1	NaN	NaN	NaN	NaN	NaN	NaN
---	---	-----	-----	-----	-----	-----	-----

T5. Create an html page with two text input boxes and four buttons. The buttons should be labeled `+`, `-`, `*`, and `/`. When one of these buttons is pressed you should get the value from both text input boxes and add, subtract, multiply, or divide the numbers entered in the text input boxes. The result should be displayed below the buttons. Note In order to do math on the values you read from the text input boxes you will need to use `Number.parseInt` on the value. for example suppose you get a reference to input box 1 using `myIn1 = document.querySelector("#in1id");` then the statement `value1`

= `Number.parseInt(myIn1.value)` converts the string from the text input box to an integer. In fact most of the time Javascript will do the conversion for you automatically except for addition.

25	25	<input type="button" value="+"/> <input type="button" value="-"/> <input type="button" value="*"/> <input type="button" value="/"/>
----	----	---

Result: 50

T6. Starting with the code given, create a page that looks like the following image: The rest of the page must be created using javascript. You must use `document.createElement` and the `appendChild` functions.



```
<html>
<body>
<button onclick="makePage();">Click Here</button>
</body>

</html>
```

Click Here
Header
Main Content
Footer
Header
Main Content
Footer

T7. Create a Tip Calculator as a single page web application (SPA). Design an interface that allows you to enter the amount of the tip. The percentage you would like to tip, and the number of people to split the tip with. Do not use 3 text input elements! Calculate and dynamically display the tip.

Bill Amount: .
Tip Percentage: .
Split with: .

Tip Amount per person: \$8.00

Web Technology LAB ASSIGNMENT-6

Name:- Pratyush Tiwari

Roll No. :- 22MC3024

Branch :- Mathematics and Computing

Code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Shopping List</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f0f0f0;
    }

    h1 {
      text-align: center;
      margin-top: 30px;
      color: #007bff;
    }

    #inputBox {
      padding: 10px;
      font-size: 16px;
      width: 300px;
      margin-right: 10px;
      border: 1px solid #ced4da;
      border-radius: 5px;
      outline: none;
    }

    .btn {
      padding: 10px 20px;
      font-size: 16px;
      cursor: pointer;
      border: none;
      border-radius: 5px;
      margin-left: 10px;
    }

    #addItemBtn {
```

```
background-color: #28a745;
color: #fff;
}

#sortBtn {
background-color: #17a2b8;
color: #fff;
}

#clearBtn {
background-color: #dc3545;
color: #fff;
}

#priorityBtn {
background-color: #ffc107;
color: #000;
}

#filterDropdown {
padding: 10px;
font-size: 16px;
border: 1px solid #ced4da;
border-radius: 5px;
outline: none;
}

ul {
list-style-type: none;
padding: 0;
}

li {
margin-bottom: 10px;
background-color: #fff;
padding: 10px;
border-radius: 5px;
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
display: flex;
align-items: center;
cursor: grab;
}

li.completed {
background-color: #c3e6cb;
}

li.priority {
background-color: #ffc107;
}
```

```
li input[type="text"] {  
    padding: 5px;  
    font-size: 14px;  
    margin-right: 10px;  
    border: 1px solid #ced4da;  
    border-radius: 5px;  
    outline: none;  
}  
  
}
```

```
li button {  
    padding: 8px 16px;  
    font-size: 14px;  
    cursor: pointer;  
    border: none;  
    border-radius: 5px;  
    background-color: #dc3545;  
    color: #fff;  
    margin-left: auto;  
}  
  
}
```

```
.item-count {  
    text-align: center;  
    margin-top: 10px;  
}  
  
}
```

```
.theme-options {  
    text-align: center;  
    margin-top: 20px;  
    position: absolute;  
    top: 20px;  
    right: 20px;  
}  
  
}
```

```
.theme-btn {  
    padding: 10px 20px;  
    font-size: 16px;  
    cursor: pointer;  
    border: none;  
    border-radius: 5px;  
    margin: 0 5px;  
}  
  
}
```

```
.theme-dark {  
    background-color: #343a40;  
    color: #fff;  
}  
  
}
```

```
.theme-light {  
    background-color: #fff;  
    color: #000;  
}  
  
}
```

```
</style>
</head>
<body>
  <h1>Shopping List</h1>
  <div class="theme-options">
    <button id="themeBtn" class="theme-btn">Toggle Theme</button>
  <div style="text-align: center;">
    <input type="text" id="inputBox" placeholder="Enter an item...">
    <button id="addItemBtn" class="btn">Add Item</button>
    <button id="sortBtn" class="btn">Sort</button>
    <button id="clearBtn" class="btn">Clear All</button>
    <button id="priorityBtn" class="btn">Priority</button>
    <select id="filterDropdown">
      <option value="all">All</option>
      <option value="completed">Completed</option>
      <option value="uncompleted">Uncompleted</option>
      <option value="priority">Priority</option>
    </select>
  </div>
  <ul id="shoppingList"></ul>
  <div class="item-count" id="itemCount"></div>
  <div class="theme-options">
    <button class="theme-btn theme-light">Light Theme</button>
    <button class="theme-btn theme-dark">Dark Theme</button>
  </div>

<script>
  document.addEventListener('DOMContentLoaded', function() {
    const inputBox = document.getElementById('inputBox');
    const addItemBtn = document.getElementById('addItemBtn');
    const sortBtn = document.getElementById('sortBtn');
    const clearBtn = document.getElementById('clearBtn');
    const filterDropdown = document.getElementById('filterDropdown');
    const shoppingList = document.getElementById('shoppingList');
    const itemCount = document.getElementById('itemCount');
    const themeButtons = document.querySelectorAll('.theme-btn');
    const priorityBtn = document.getElementById('priorityBtn');

    addItemBtn.addEventListener('click', addItem);
    sortBtn.addEventListener('click', sortItems);
    clearBtn.addEventListener('click', clearAll);
    filterDropdown.addEventListener('change', filterItems);
    themeButtons.forEach(button => {
      button.addEventListener('click', changeTheme);
    });
    priorityBtn.addEventListener('click', togglePriority);

    let dragStartIndex;

    loadItems();

    function addItem() {
```

```
const itemText = inputBox.value.trim();
if (itemText !== '') {
    const li = createListItem(itemText);
    shoppingList.appendChild(li);
    inputBox.value = '';
    saveItems();
    updateItemCount();
}
}

function createListItem(text) {
    const li = document.createElement('li');
    const inputEdit = document.createElement('input');
    inputEdit.type = 'text';
    inputEdit.value = text;
    inputEdit.disabled = true;
    li.appendChild(inputEdit);

    const editBtn = document.createElement('button');
    editBtn.textContent = 'Edit';
    editBtn.addEventListener('click', function() {
        inputEdit.disabled = !inputEdit.disabled;
        inputEdit.focus();
    });
}

const removeBtn = document.createElement('button');
removeBtn.textContent = 'Remove';
removeBtn.addEventListener('click', function() {
    removeItem(li);
});

const completeBtn = document.createElement('button');
completeBtn.textContent = 'Complete';
completeBtn.addEventListener('click', function() {
    li.classList.toggle('completed');
    saveItems();
    updateItemCount();
});

li.appendChild(editBtn);
li.appendChild(removeBtn);
li.appendChild(completeBtn);
li.draggable = true;

li.addEventListener('dragstart', dragStart);
li.addEventListener('dragover', dragOver);
li.addEventListener('drop', dragDrop);

return li;
}
```

```
function removeItem(item) {
    shoppingList.removeChild(item);
    saveItems();
    updateItemCount();
}

function clearAll() {
    shoppingList.innerHTML = '';
    localStorage.clear();
    updateItemCount();
}

function sortItems() {
    const items = Array.from(shoppingList.children);
    items.sort(function(a, b) {
        const textA = a.querySelector('input').value.toLowerCase();
        const textB = b.querySelector('input').value.toLowerCase();
        return textA.localeCompare(textB);
    });
    items.forEach(function(item) {
        shoppingList.appendChild(item);
    });
}

function filterItems() {
    const filterValue = filterDropdown.value;
    shoppingList.querySelectorAll('li').forEach(function(item) {
        if (filterValue === 'all') {
            item.style.display = 'flex';
        } else if (filterValue === 'completed' && item.classList.contains('completed')) {
            item.style.display = 'flex';
        } else if (filterValue === 'uncompleted' && !item.classList.contains('completed')) {
            item.style.display = 'flex';
        } else if (filterValue === 'priority' && item.classList.contains('priority')) {
            item.style.display = 'flex';
        } else {
            item.style.display = 'none';
        }
    });
}

function dragStart() {
    dragStartIndex = Array.from(shoppingList.children).indexOf(this);
}

function dragOver(e) {
    e.preventDefault();
}

function dragDrop() {
    const dragEndIndex = Array.from(shoppingList.children).indexOf(this);
```

```
const items = Array.from(shoppingList.children);
const itemToMove = items[dragStartIndex];
items.splice(dragStartIndex, 1);
items.splice(dragEndIndex, 0, itemToMove);
shoppingList.innerHTML = '';
items.forEach(item => {
    shoppingList.appendChild(item);
});
saveItems();
}

function saveItems() {
    const items = [];
    shoppingList.querySelectorAll('li').forEach(function(item) {
        items.push({
            text: item.querySelector('input').value,
            completed: item.classList.contains('completed')
        });
    });
    localStorage.setItem('shoppingListItems', JSON.stringify(items));
}

function loadItems() {
    const items = JSON.parse(localStorage.getItem('shoppingListItems'));
    if (items) {
        items.forEach(function(item) {
            const li = createListItem(item.text);
            if (item.completed) {
                li.classList.add('completed');
            }
            shoppingList.appendChild(li);
        });
    }
    updateItemCount();
}

function updateItemCount() {
    const count = shoppingList.querySelectorAll('li').length;
    itemCount.textContent = `Total items: ${count}`;
}

function changeTheme() {
    const theme = this.classList.contains('theme-light') ? 'light' : 'dark';
    document.body.classList.remove('theme-light', 'theme-dark');
    document.body.classList.add(`theme-${theme}`);
    saveTheme(theme);
}

function saveTheme(theme) {
    localStorage.setItem('shoppingListTheme', theme);
}
```

```

function loadTheme() {
    const theme = localStorage.getItem('shoppingListTheme');
    if (theme) {
        document.body.classList.add(`theme-${theme}`);
    }
}

function togglePriority() {
    this.classList.toggle('priority');
    filterItems();
}

loadTheme();
});

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

```

Output:-

The screenshot shows a JupyterLab environment with a dark theme. A tab titled "1 Shopping List" is active. The main content area displays a "Shopping List" application. The list contains three items:

- Grocery
- Pen
- Notebook

Each item has associated actions:

- Edit**: A blue button.
- Remove**: A red button.
- Complete**: A red button.

At the top of the list, there are several buttons and controls:

- Enter an item... (text input field)
- Add Item (blue button)
- Sort (teal button)
- Clear All (red button)
- Priority (yellow button)
- All (dropdown menu)
- Light Theme (button)
- Dark Theme (button, currently selected)

A message at the bottom of the list states "Total items: 3".

Untitled2.ipynb (3) - JupyterLab 127.0.0.1

Shopping List

Light Theme Dark Theme

Enter an item... Add Item Sort Clear All Priority All

		Edit	Remove	Complete
Grocery				
Pen				
Notebook				

Total items: 3

Grocery

Pen

Notebook

Edit Remove Complete

Edit Remove Complete

Edit Remove Complete

Total items: 3

Web-Tech Lab 07

Name-Pratyush Tiwari
Roll No.- 22MC3024

T1. Develop prototype 3 continuing with the last lab. Confirm that the app now remembers your list even after a page refresh.

HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Shopping List</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h1>Shopping List</h1>
        <input type="text" id="itemInput" placeholder="Add new item">
        <button onclick="addItem()">Add Item</button>
        <ul id="itemList"></ul>
    </div>

    <script src="script.js"></script>
</body>
</html>
```

CSS code:

```
.container {
    max-width: 600px;
    margin: 50px auto;
    padding: 0 20px;
}

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

```

input[type="text"] {
    width: 70%;
    padding: 8px;
    margin-bottom: 10px;
}

button {
    padding: 8px 15px;
    background-color: #4CAF50;
    color: white;
    border: none;
    cursor: pointer;
}

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

ul {
    list-style-type: none;
    padding: 0;
}

li {
    margin-bottom: 5px;
    padding: 8px;
    background-color: #f2f2f2;
    border-radius: 5px;
}

```

Javascript Code:

script.js

```

// Function to add item to the list
function addItem() {
    var itemInput = document.getElementById('itemInput');
    var itemValue = itemInput.value.trim();

```

```

    if (itemValue !== '') {
        var itemList = document.getElementById('itemList');
        var listItem = document.createElement('li');
        listItem.textContent = itemValue;
        itemList.appendChild(listItem);
        saveListToStorage(); // Save list to localStorage
        itemInput.value = '';
    } else {
        alert('Please enter a valid item!');
    }
}

// Function to save the list to localStorage
function saveListToStorage() {
    var itemList = document.getElementById('itemList');
    var items = [];

    // Get all list items
    for (var i = 0; i < itemList.children.length; i++) {
        items.push(itemList.children[i].textContent);
    }

    // Save items to localStorage
    localStorage.setItem('shoppingList', JSON.stringify(items));
}

// Function to load the list from localStorage
function loadListFromStorage() {
    var itemList = document.getElementById('itemList');
    var storedItems = localStorage.getItem('shoppingList');

    if (storedItems) {
        var items = JSON.parse(storedItems);

        // Add items to the list
        items.forEach(function(item) {
            var listItem = document.createElement('li');
            listItem.textContent = item;
            itemList.appendChild(listItem);
        });
    }
}

```

```

        }
    }

// Load list from localStorage when the page loads
window.addEventListener('load', loadListFromStorage);

```

model.js

```

var shoppingListModel = {
    items: [],
    addItem: function(item) {
        this.items.push(item);
    }
};

```

controller.js

```

var shoppingListController = {
    addItem: function() {
        var itemInput = document.getElementById('itemInput');
        var itemValue = itemInput.value.trim();

        if (itemValue !== '') {
            shoppingListModel.addItem(itemValue);
            itemInput.value = '';
            shoppingListView.displayItems();
        } else {
            alert('Please enter a valid item!');
        }
    },
    init: function() {
        this.setupEventListeners();
        shoppingListView.displayItems();
    },
    setupEventListeners: function() {
        var addButton = document.querySelector('button');
        addButton.addEventListener('click', this.addItem);
        var itemInput = document.getElementById('itemInput');
        itemInput.addEventListener('keypress', function(event) {

```

```

        if (event.key === 'Enter') {
            shoppingListController.addItem();
        }
    });
}

shoppingListController.init();

```

view.js

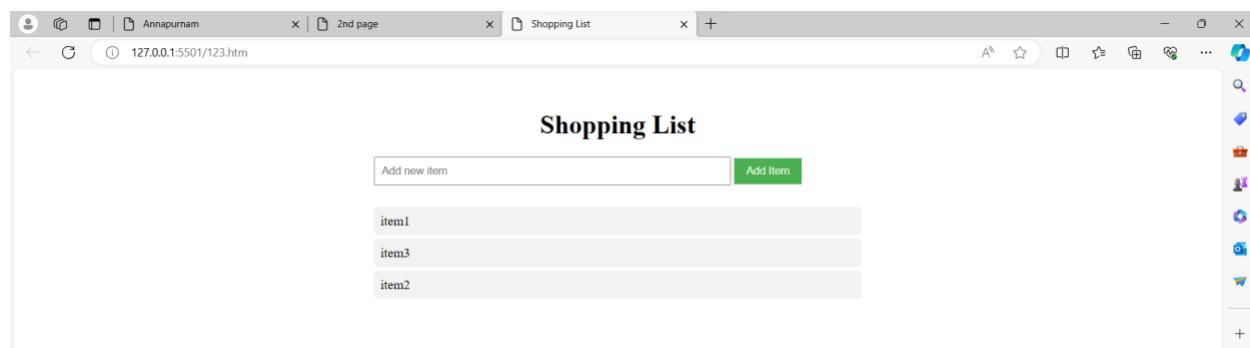
```

var shoppingListView = {
    displayItems: function() {
        var itemList = document.getElementById('itemList');
        itemList.innerHTML = '';

        shoppingListModel.items.forEach(function(item) {
            var listItem = document.createElement('li');
            listItem.textContent = item;
            itemList.appendChild(listItem);
        });
    }
};

```

OUTPUT:-



T2. Create a local storage that saves the number of times you have accessed the page and displays it.

HTML code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Page Access Counter</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <h1>Page Access Counter</h1>
    <p>You have accessed this page <span id="accessCount"></span> times.</p>
    <button id="incrementButton">Increment Access Count</button>
    <script src="script.js"></script>
</body>
</html>
```

CSS Code:-

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    text-align: center;
}

h1 {
    margin-top: 50px;
}

p {
    font-size: 18px;
}
```

Javascript Code:-

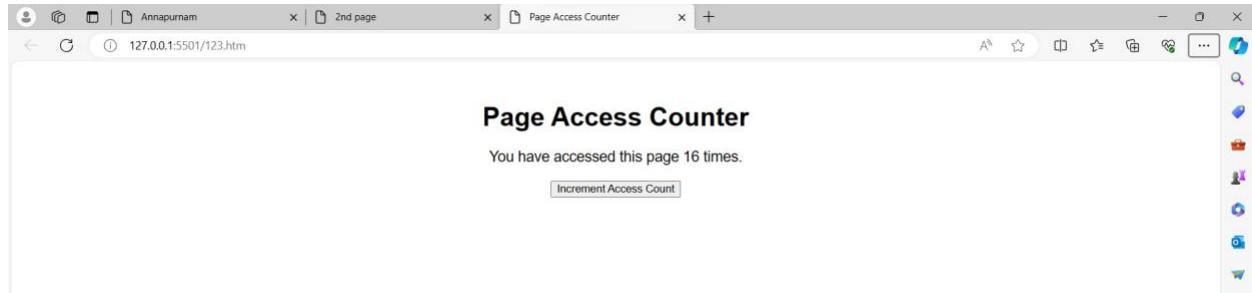
```
function updateAccessCount () {
    if (typeof(Storage) !== "undefined") {
        if (localStorage.pageAccessCount) {
            localStorage.pageAccessCount =
Number(localStorage.pageAccessCount) + 1;
        } else {
            localStorage.pageAccessCount = 1;
        }
        document.getElementById("accessCount").innerText =
localStorage.pageAccessCount;
    } else {
        document.getElementById("accessCount").innerText = "Sorry, your
browser does not support web storage...";
    }
}

function initializeAccessCount () {
    if (typeof(Storage) !== "undefined") {
        if (!localStorage.pageAccessCount) {
            localStorage.pageAccessCount = 0;
        }
        document.getElementById("accessCount").innerText =
localStorage.pageAccessCount;
    } else {
        document.getElementById("accessCount").innerText = "Sorry, your
browser does not support web storage...";
    }
}

window.onload = initializeAccessCount;

document.getElementById("incrementButton").addEventListener("click",
updateAccessCount);
```

OUTPUT:-



WEB TECHNOLOGY LAB ASSIGNMENT

Name : Pratyush Tiwari

Roll No. : 22MC3024

Branch : Mathematics & Computing

Q.1

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Currency Converter</title>
    <style>
        /* CSS styles */
        body {
            font-family: Arial, sans-serif;
        }
        label {
            margin-right: 10px;
        }
        select {
            margin-left: 5px;
        }
    </style>
</head>
<body>
    <div id="root"></div>
    <script
src="https://unpkg.com/react@17/umd/react.development.js"></script>
    <script
src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></s
cript>
```

```
<script>
// ReactJS code
const { useState } = React;

function CurrencyConverter() {
  const [amount, setAmount] = useState('');
  const [fromCurrency, setFromCurrency] = useState('USD');
  const [toCurrency, setToCurrency] = useState('EUR');
  const exchangeRate = 0.85; // Example exchange rate (1 USD =
0.85 EUR)

  const handleAmountChange = (event) => {
    setAmount(event.target.value);
  };

  const handleFromCurrencyChange = (event) => {
    setFromCurrency(event.target.value);
  };

  const handleToCurrencyChange = (event) => {
    setToCurrency(event.target.value);
  };

  const convertCurrency = () => {
    const convertedAmount = parseFloat(amount) * exchangeRate;
    return isNaN(convertedAmount) ? '' :
convertedAmount.toFixed(2);
  };

  return React.createElement(
    'div',
    null,
    React.createElement('h1', null, 'Currency Converter'),
    React.createElement(
      'div',
      null,
      React.createElement(

```

```
        'label',
        null,
        'Amount:',
        React.createElement('input', { type: 'number', value:
amount, onChange: handleAmountChange })
    )
),
React.createElement(
'div',
null,
React.createElement(
'label',
null,
'From:',
React.createElement(
'select',
{ value: fromCurrency, onChange:
handleFromCurrencyChange },
React.createElement('option', { value: 'USD' }, 'USD'),
React.createElement('option', { value: 'EUR' }, 'EUR')
)
)
),
React.createElement(
'div',
null,
React.createElement(
'label',
null,
'To:',
React.createElement(
'select',
{ value: toCurrency, onChange: handleToCurrencyChange
},
React.createElement('option', { value: 'USD' }, 'USD'),
React.createElement('option', { value: 'EUR' }, 'EUR')
)
)
```

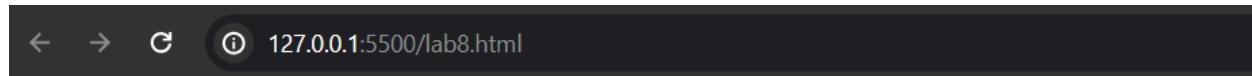
```

        )
    ) ,
    React.createElement(
        'div',
        null,
        React.createElement('p', null, 'Converted Amount: ',
convertCurrency()))
    )
);
}

ReactDOM.render(React.createElement(CurrencyConverter),
document.getElementById('root'));
</script>
</body>
</html>

```

Output:



Currency Converter

Amount:

From:

To:

Converted Amount: 28.90

Q.2

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>Stopwatch</title>
<style>
/* CSS styles */
body {
    font-family: Arial, sans-serif;
}
button {
    margin-right: 10px;
}
</style>
</head>
<body>
<div id="root"></div>
<script
src="https://unpkg.com/react@17/umd/react.development.js"></script>
<script
src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></s
cript>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.26.0/b
abel.min.js"></script>
<script type="text/babel">
// ReactJS code
const { useState, useEffect } = React;

function Stopwatch() {
    const [timer, setTimer] = useState(0);
    const [isActive, setIsActive] = useState(false);

    useEffect(() => {
        let interval;
        if (isActive) {
            interval = setInterval(() => {
                setTimer((prevTimer) => prevTimer + 1);
            }, 1000);
        }
    }, [isActive]);
}

Stopwatch();

```

```
        }, 1000);
    } else {
        clearInterval(interval);
    }
    return () => clearInterval(interval);
}, [isActive]);

const handleStart = () => {
    setIsActive(true);
};

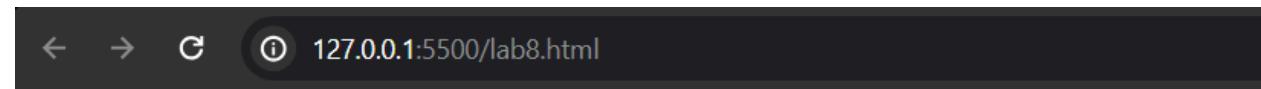
const handlePause = () => {
    setIsActive(false);
};

const handleReset = () => {
    setIsActive(false);
    setTimer(0);
};

return (
    <div>
        <h1>Stopwatch</h1>
        <div>
            <p>{timer}s</p>
        </div>
        <div>
            {!isActive ? (
                <button onClick={handleStart}>Start</button>
            ) : (
                <button onClick={handlePause}>Pause</button>
            )}
            <button onClick={handleReset}>Reset</button>
        </div>
    </div>
);
}
```

```
ReactDOM.render(<Stopwatch />, document.getElementById('root'));  
</script>  
</body>  
</html>
```

Output:



Stopwatch

5s

Q.3

```
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="UTF-8" />  
  <title>Simple Messaging App</title>  
  <style>  
    body {  
      font-family: Arial, sans-serif;  
    }  
    #conversations {  
      list-style-type: none;  
    }
```

```
padding: 0;
margin: 0;
}

#conversations li {
    padding: 10px;
    border-bottom: 1px solid #ddd;
    cursor: pointer;
}

#conversations li:hover {
    background-color: #f5f5f5;
}

#messages {
    list-style-type: none;
    padding: 0;
    margin: 0;
    height: 300px;
    overflow-y: auto;
}

#messages li {
    padding: 10px;
    border-bottom: 1px solid #ddd;
}

#message-input {
    width: 100%;
    padding: 10px;
}

#send-button {
    padding: 10px;
    background-color: #4CAF50;
    color: white;
    border: none;
    cursor: pointer;
}

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

</style>
```

```
</head>
<body>
  <h1>Simple Messaging App</h1>
  <ul id="conversations"></ul>
  <ul id="messages"></ul>
  <input id="message-input" type="text" placeholder="Type a message..." />
  <button id="send-button">Send</button>
  <script
src="https://unpkg.com/react@16.13.1/umd/react.development.js"></script>
  <script
src="https://unpkg.com/react-dom@16.13.1/umd/react-dom.development.js"></script>
  <script
src="https://unpkg.com/babel-standalone@6.26.0/babel.min.js"></script>
  <script
src="https://unpkg.com/push.js@1.0.12/push.min.js"></script>
  <script type="text/babel">
    class Conversation extends React.Component {
      constructor(props) {
        super(props);
        this.state = {
          messages: [],
        };
        this.sendMessage = this.sendMessage.bind(this);
      }
      componentDidMount() {
        this.getMessages();
        this.interval = setInterval(() => this.getMessages(),
1000);
      }
      componentWillUnmount() {
        clearInterval(this.interval);
      }
      getMessages() {
```

```

fetch(`/api/conversations/${this.props.conversation.id}/messages`)
  .then((response) => response.json())
  .then((data) => {
    this.setState({ messages: data });
  });
}

sendMessage() {
  const message = {
    text: document.getElementById("message-input").value,
  };

fetch(`/api/conversations/${this.props.conversation.id}/messages`, {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify(message),
})
  .then((response) => response.json())
  .then((data) => {
    this.getMessages();
    document.getElementById("message-input").value = "";
    Push.create("New message", {
      body: data.text,
      timeout: 4000,
    });
  });
}

render() {
  return (
    <div>
      <h2>{this.props.conversation.name}</h2>
      <ul id="messages">
        {this.state.messages.map((message) => (
          <li key={message.id}>{message.text}</li>
        )));
    </div>
}

```

```
</ul>
<input
    id="message-input"
    type="text"
    placeholder="Type a message..."
    onKeyPress={(event) => {
        if (event.key === "Enter") {
            this.sendMessage();
        }
    }}
/>
<button id="send-button" onClick={this.sendMessage}>
    Send
</button>
</div>
);
}
}

class Conversations extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            conversations: [],
        };
    }
    componentDidMount() {
        fetch("/api/conversations")
            .then((response) => response.json())
            .then((data) => {
                this.setState({ conversations: data });
            });
    }
    render() {
        return (
            <ul id="conversations">
                {this.state.conversations.map((conversation) => (

```

```
        <li key={conversation.id} onClick={() =>
this.props.onSelect(conversation)}>
    {conversation.name}
</li>
)) }
</ul>
);
}
}

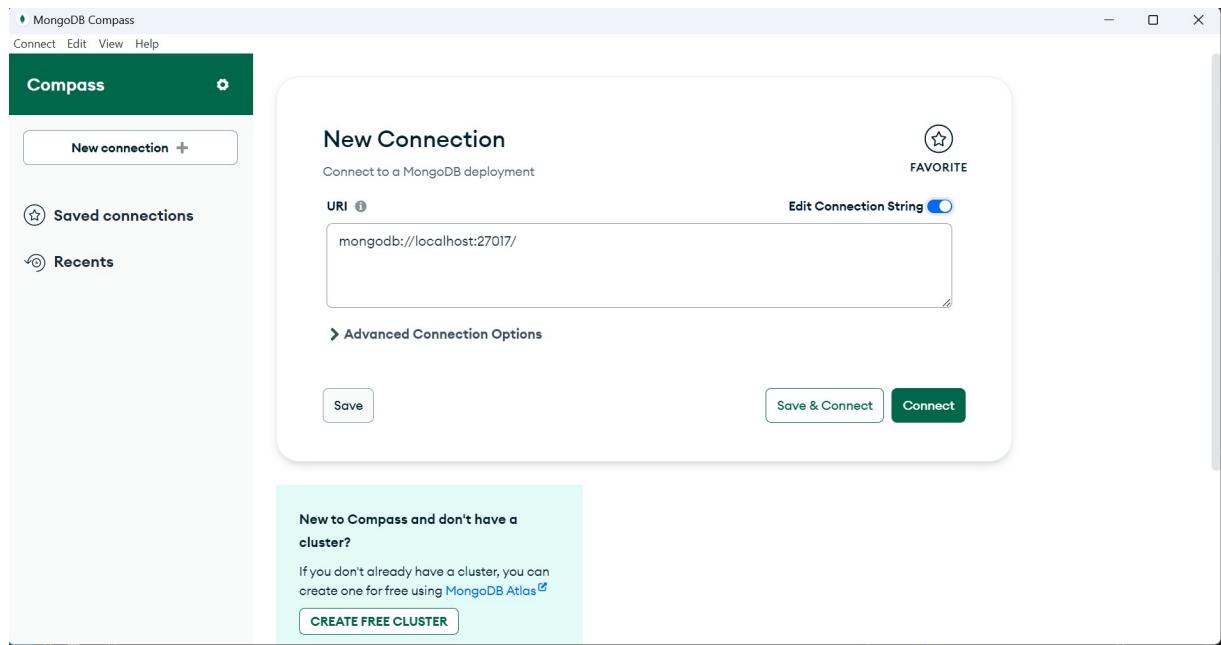
class App extends React.Component {
constructor(props) {
super(props);
this.state = {
selectedConversation: null,
};
this.onSelectConversation =
this.onSelectConversation.bind(this);
}
onSelectConversation(conversation) {
this.setState({ selectedConversation: conversation });
}
render() {
return (
<div>
<Conversations onSelect={this.onSelectConversation} />
{this.state.selectedConversation && (
<Conversation
conversation={this.state.selectedConversation} />
)}
</div>
);
}
ReactDOM.render(<App />, document.getElementById("root"));
</script>
</body>
</html>
```


Pratyush Tiwari
22MC3024
Mathematics and Computing

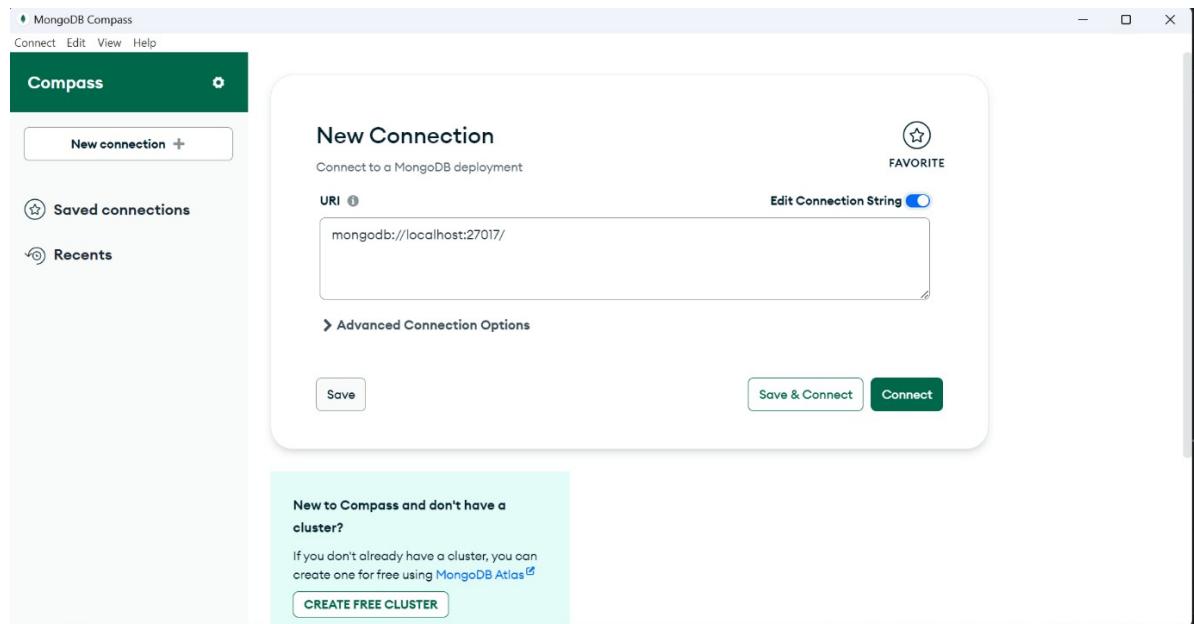
WEB TECHNOLOGY

LAB-10

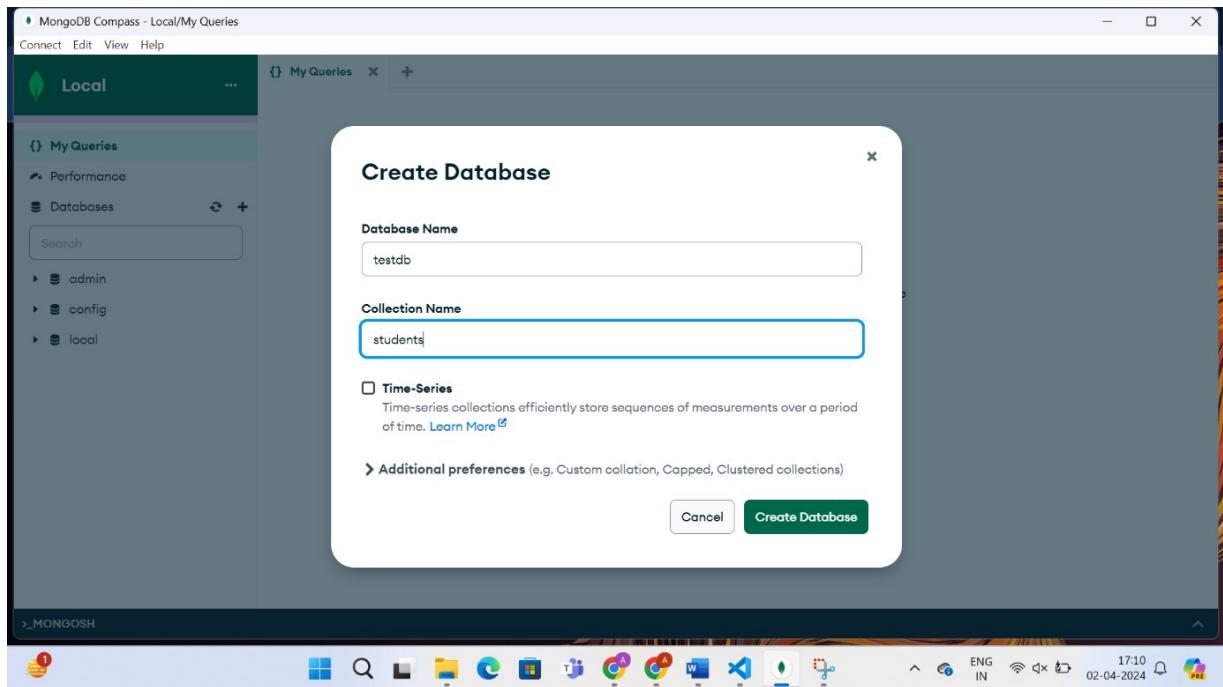
Connect to a MongoDB server using MongoDB Compass.



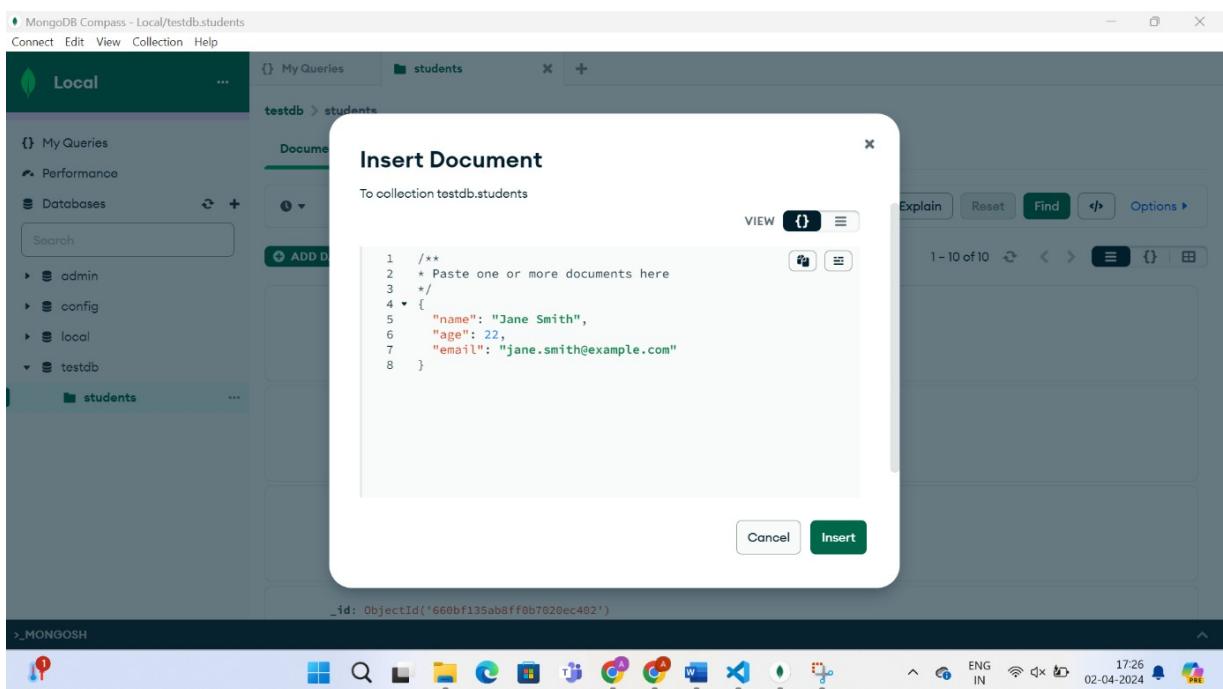
Create a new database named "testdb" in MongoDB Compass.



Create a new collection named "students" in the "testdb" database.



Insert ten documents into the "students" collection with the following fields: name, age, and email.



View the contents of the "students" collection.

MongoDB Compass - Local/testdb.students

Connect Edit View Collection Help

Local

My Queries

Performance

Databases

Search

admin

config

local

testdb

students

Documents 10 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

[Explain](#) [Reset](#) [Find](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

1-10 of 10

	_id	ObjectId	name	String	age	Int32	email	String
1	ObjectId('660bf0f7ab8ff0b...	"John Doe"	28	"john.doe@example.com"				
2	ObjectId('660bf113ab8ff0b...	"Jane Smith"	22	"jane.smith@example.com"				
3	ObjectId('660bf126ab8ff0b...	"Michael Johnson"	21	"michael.johnson@example...				
4	ObjectId('660bf135ab8ff0b...	"Emily Brown"	19	"emily.brown@example.com"				
5	ObjectId('660bf14dab8ff0b...	"David Wilson"	23	"david.wilson@example.com"				
6	ObjectId('660bf1b7ab8ff0b...	"Sarah Martinez"	20	"sarah.martinez@example.c...				
7	ObjectId('660bf1c9ab8ff0b...	"Christopher Taylor"	24	"christopher.taylor@example...				
8	ObjectId('660bf1daab8ff0b...	"Jessica Anderson"	22	"jessica.anderson@example...				
9	ObjectId('660bf1e8ab8ff0b...	"Ryan Thomas"	21	"ryan.thomas@example.com"				
10	ObjectId('660bf1f8aab8ff0b...	"Amanda Garcia"	20	"amanda.garcia@example.co...				

>_MONGOSH

ENG IN 17:28 02-04-2024

Update the age of a specific student in the "students" collection.

MongoDB Compass - Local/testdb.students

Connect Edit View Collection Help

Local

My Queries

Performance

Databases

Search

admin

config

local

testdb

students

Documents 10 Aggregations Schema Indexes 1 Validation

[Generate query](#) [Explain](#) [Reset](#) [Find](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

1-10 of 10

+	_id: ObjectId('660bf0f7ab8ff0b7020ec3fc')	ObjectId
	name: "John Doe"	String
	age: 28	Int32
	email: "john.doe@example.com"	String

Document modified.

```
_id: ObjectId('660bf113ab8ff0b7020ec3fe')
name: "Jane Smith"
age: 22
email: "jane.smith@example.com"
```

```
_id: ObjectId('660bf126ab8ff0b7020ec400')
name: "Michael Johnson"
age: 21
email: "michael.johnson@example.com"
```

>_MONGOSH

ENG IN 17:31 02-04-2024

Delete a document from the "students" collection based on a specific condition.

Use the aggregation pipeline to calculate the average age of all students in the "students" collection.

Create an index on the "name" field in the "students" collection.

MongoDB Compass - Local/testdb.students

Connect Edit View Collection Help

Local

My Queries

Performance

Databases

Search

testdb > students

Documents 9 Aggregations Schema Indexes 1 Validation

Create Index

Create Index

Name: testdb.students

Index fields:

- name (1 asc)

Options

Cancel Create Index

VIEWING INDEXES SEARCH INDEXES

Properties

UNIQUE

17:51 02-04-2024

Export the contents of the "students" collection to a JSON file.

MongoDB Compass - Local/testdb.students

Connect Edit View Collection Help

Local

My Queries

Performance

Databases

Search

testdb > students

Documents 9 Aggregations Schema Indexes 1 Validation

Generate query Explain Reset Find Options

EXPORT DATA

1-9 of 9

Export completed.
9 documents written.

SHOW FILE

17:54 02-04-2024

Perform a complex aggregation operation to find the top 5 oldest students in the "students" collection.

MongoDB Compass - Local/testdb.students

Connect Edit View Collection Help

Local

My Queries Aggregations Schema Indexes Validation

testdb > students

Documents 9 Aggregations

\$group \$sort

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT WIZARD

Stage 2 \$sort

```

1 [
2 {
3   "$sort": {
4     "age": -1
5   }
6 },
7 {
8   "$limit": 5
9 }
10 ]

```



Create a geospatial index on a field representing the location of students.

MongoDB Compass - Local/testdb.students

Connect Edit View Collection Help

Local

My Queries Aggregations Schema Indexes Validation

testdb > students

Documents 9 Aggregations Schema Indexes 1 Validation

Create Index Refresh

VIEWING INDEXES SEARCH INDEXES

Name and Definition	Type	Size	Usage	Properties
id	REGULAR	36.9 KB	13 (since Tue Apr 02 2024)	UNIQUE
name_1	REGULAR	20.5 KB	0 (since Tue Apr 02 2024)	
students_2dsphere	GEOSPATIAL	8.2 KB	0 (since Tue Apr 02 2024)	

