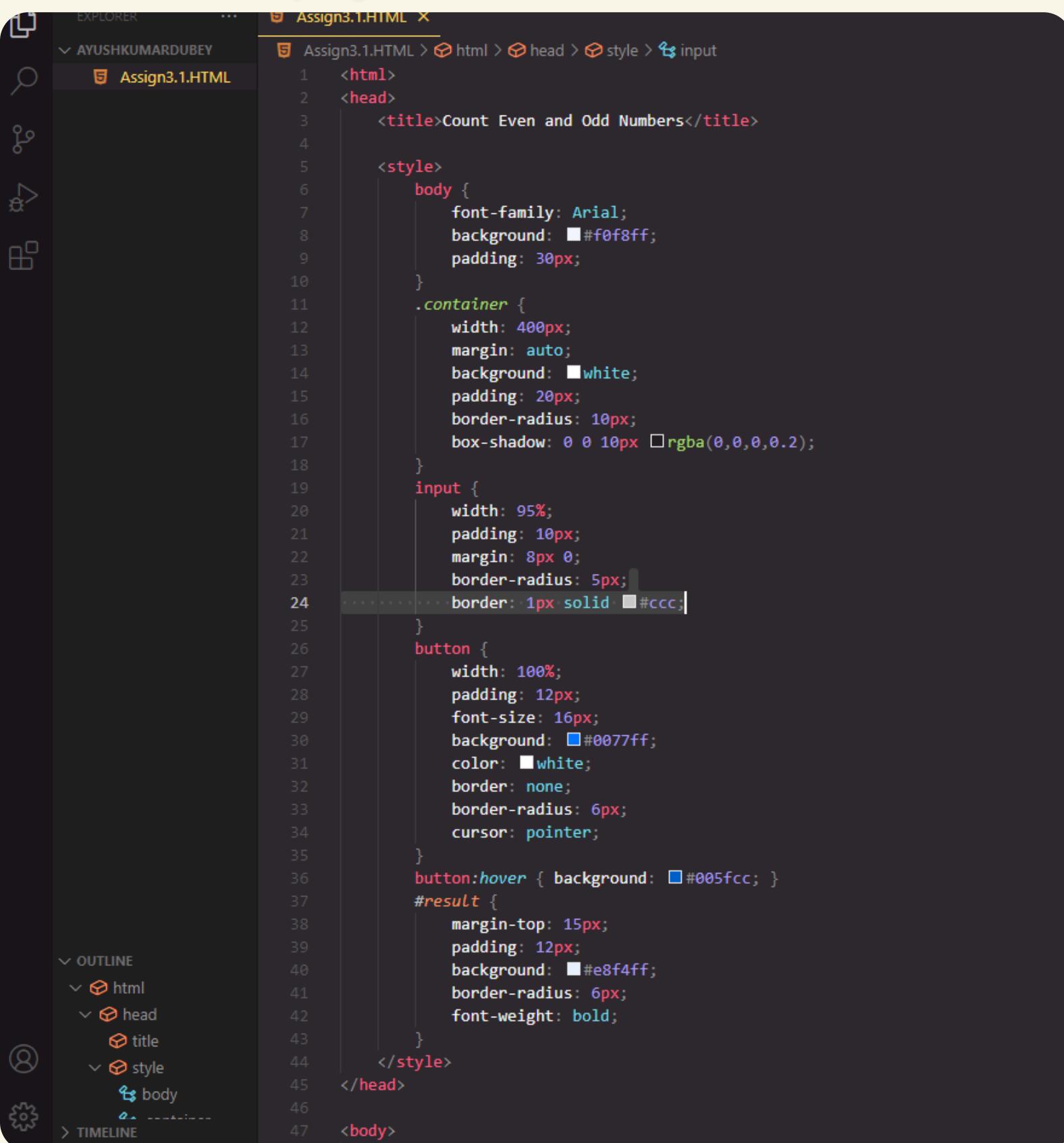


### 3.1 Count how many even and odd numbers are there.

This program takes a list of numbers as input and checks each value to determine whether it is even or odd. It counts the total number of even and odd values and displays the final result.

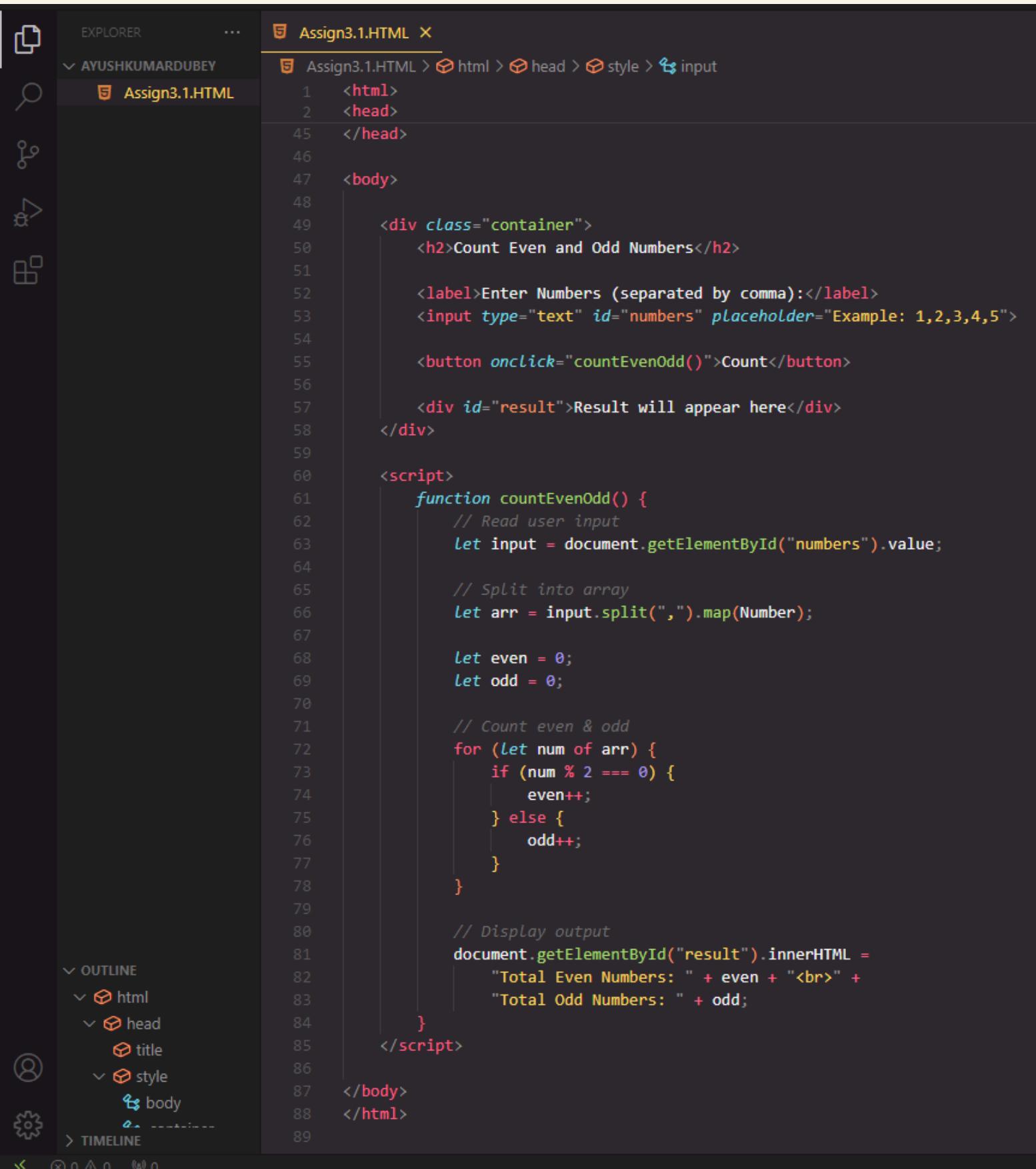


The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows a folder named "AYUSHKUMARDUBEY" containing "Assign3.1.HTML".
- File View:** The file "Assign3.1.HTML" is open, showing the following code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Count Even and Odd Numbers</title>
    <style>
      body {
        font-family: Arial;
        background-color: #f0f8ff;
        padding: 30px;
      }
      .container {
        width: 400px;
        margin: auto;
        background-color: white;
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0 0 10px rgba(0,0,0,0.2);
      }
      input {
        width: 95%;
        padding: 10px;
        margin: 8px 0;
        border-radius: 5px;
        border: 1px solid #ccc;
      }
      button {
        width: 100%;
        padding: 12px;
        font-size: 16px;
        background-color: #0077ff;
        color: white;
        border: none;
        border-radius: 6px;
        cursor: pointer;
      }
      button:hover {
        background-color: #005fcc;
      }
      #result {
        margin-top: 15px;
        padding: 12px;
        background-color: #e8f4ff;
        border-radius: 6px;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
```

# JavaScript here



The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** sidebar: Shows a tree view with "AYUSHKUMARDUBEY" expanded, containing "Assign3.1.HTML".
- Assign3.1.HTML** tab is active in the main editor area.
- Code Content:**

```
1 <html>
2 <head>
45 </head>
46
47 <body>
48
49     <div class="container">
50         <h2>Count Even and Odd Numbers</h2>
51
52         <label>Enter Numbers (separated by comma):</label>
53         <input type="text" id="numbers" placeholder="Example: 1,2,3,4,5">
54
55         <button onclick="countEvenOdd()">Count</button>
56
57         <div id="result">Result will appear here</div>
58     </div>
59
60     <script>
61         function countEvenOdd() {
62             // Read user input
63             let input = document.getElementById("numbers").value;
64
65             // Split into array
66             let arr = input.split(",").map(Number);
67
68             let even = 0;
69             let odd = 0;
70
71             // Count even & odd
72             for (let num of arr) {
73                 if (num % 2 === 0) {
74                     even++;
75                 } else {
76                     odd++;
77                 }
78             }
79
80             // Display output
81             document.getElementById("result").innerHTML =
82                 "Total Even Numbers: " + even + "<br>" +
83                 "Total Odd Numbers: " + odd;
84         }
85     </script>
86
87 </body>
88 </html>
```
- OUTLINE** sidebar: Shows a tree view with "html" expanded, containing "head" (expanded), "title", and "style" (expanded), showing "body".
- TIMELINE**: Shows a history of changes with 1 edit.

## Conclusion:

The program successfully identifies and counts even and odd numbers based on user input.

# OUTPUT

A screenshot of a web browser window titled "Count Even and Odd Numbers". The URL in the address bar is "file:///C:/BCA\_1A/AYUSHKUMARDUBEY/Assign3.1.HTML". The page contains a form with the following elements:

- Section Title:** Count Even and Odd Numbers
- Text Input:** Enter Numbers (separated by comma):  
Example: 1,2,3,4,5
- Button:** Count
- Result Container:** Result will appear here

## Add numeric value giving ( , )

A screenshot of a web browser window titled "Count Even and Odd Numbers". The URL in the address bar is "file:///C:/BCA\_1A/AYUSHKUMARDUBEY/Assign3.1.HTML". The page contains a form with the following elements:

- Section Title:** Count Even and Odd Numbers
- Text Input:** Enter Numbers (separated by comma):  
13, 57, 28, 46, 91, 12, 03
- Button:** Count
- Result Container:** Total Even Numbers: 3  
Total Odd Numbers: 4

Activate Windows  
Go to Settings to activate Windows.



## Conclusion:

The program successfully identifies and counts even and odd numbers based on user input.

# Q 3.2 Remove duplicate values from an array.

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

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** AYUSHKUMARDUBEY
- Explorer:** Shows a folder structure under AYUSHKUMARDUBEY:
  - Assign3.2.HTML
  - image (4).jpg
  - TEMP.HTML
- Code Area:** An HTML file named Assign3.2.HTML with the following content:

```
<html>
<head>
    <title>Remove Duplicate Values from Array</title>
<style>
    body {
        font-family: Arial;
        background: #eef7ff;
        padding: 30px;
    }
    .container {
        width: 400px;
        margin: auto;
        background: white;
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0 0 10px rgba(0,0,0,0.2);
    }
    input {
        width: 95%;
        padding: 10px;
        margin: 8px 0;
        border: 1px solid #ccc;
        border-radius: 5px;
    }
    button {
        width: 100%;
        padding: 12px;
        background: #006aff;
        color: white;
        border: none;
        font-size: 17px;
        border-radius: 6px;
        cursor: pointer;
    }
    button:hover {
        background: #0050cc;
    }
    #result {
        margin-top: 15px;
        padding: 12px;
        background: #e7f2ff;
    }
</style>
<body>
    <div class="container">
        <input type="text" placeholder="Enter array elements separated by commas." />
        <button>Remove Duplicates</button>
        <div id="result">
            Result: [List of unique elements]
        </div>
    </div>
</body>
</html>
```
- Outline:** Shows the structure of the HTML document with expanded sections for html, head, title, style, and body.
- Bottom Status Bar:** Includes icons for file operations and a timeline.

This program removes duplicate values from an array by checking each element and ensuring that only unique entries are kept.

# OUTPUT

The application interface consists of a central modal window. At the top, it says "Remove Duplicate Values". Below that is a text input field with placeholder text "Enter Number's (separated by comma) : Example: 1,2,2,3,4,4,5". A blue "Remove Duplicates" button is centered below the input field. At the bottom of the modal, a light blue box displays the text "Unique values will appear here".

In the second screenshot, the input field contains the values "22,54,32,98,54,22,34,22,38,88,88". The bottom box now displays "Unique Values: 22, 54, 32, 98, 34, 38, 88".

At the bottom right of the screen, there is a watermark that reads "Activate Windows Go to Settings to activate Windows." The taskbar at the very bottom shows various icons and the date/time "02-12-2025 13:01".

## Conclusion:

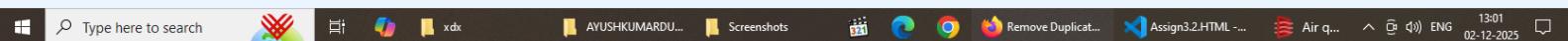
The program successfully identifies and removes all duplicate values, producing an array that contains only unique elements.

# Q 3.3 Find the sum of all numbers in an array.

A screenshot of a web browser window titled "Remove Duplicate Values from Arr...". The URL is "file:///C:/BCA\_1A/AYUSHKUMARDUBEY/Assign3.2.HTML". The page contains a form with a title "Remove Duplicate Values", a text input placeholder "Enter Number's (separated by comma): Example: 1,2,2,3,4,4,5", a blue "Remove Duplicates" button, and a light blue box below it stating "Unique values will appear here".

A screenshot of a web browser window titled "Remove Duplicate Values from Arr...". The URL is "file:///C:/BCA\_1A/AYUSHKUMARDUBEY/Assign3.2.HTML". The page displays the same "Remove Duplicate Values" form. In the "Unique Values" box, the output "Unique Values: 22, 54, 32, 98, 34, 38, 88" is shown.

Activate Windows  
Go to Settings to activate Windows.



## Conclusion:

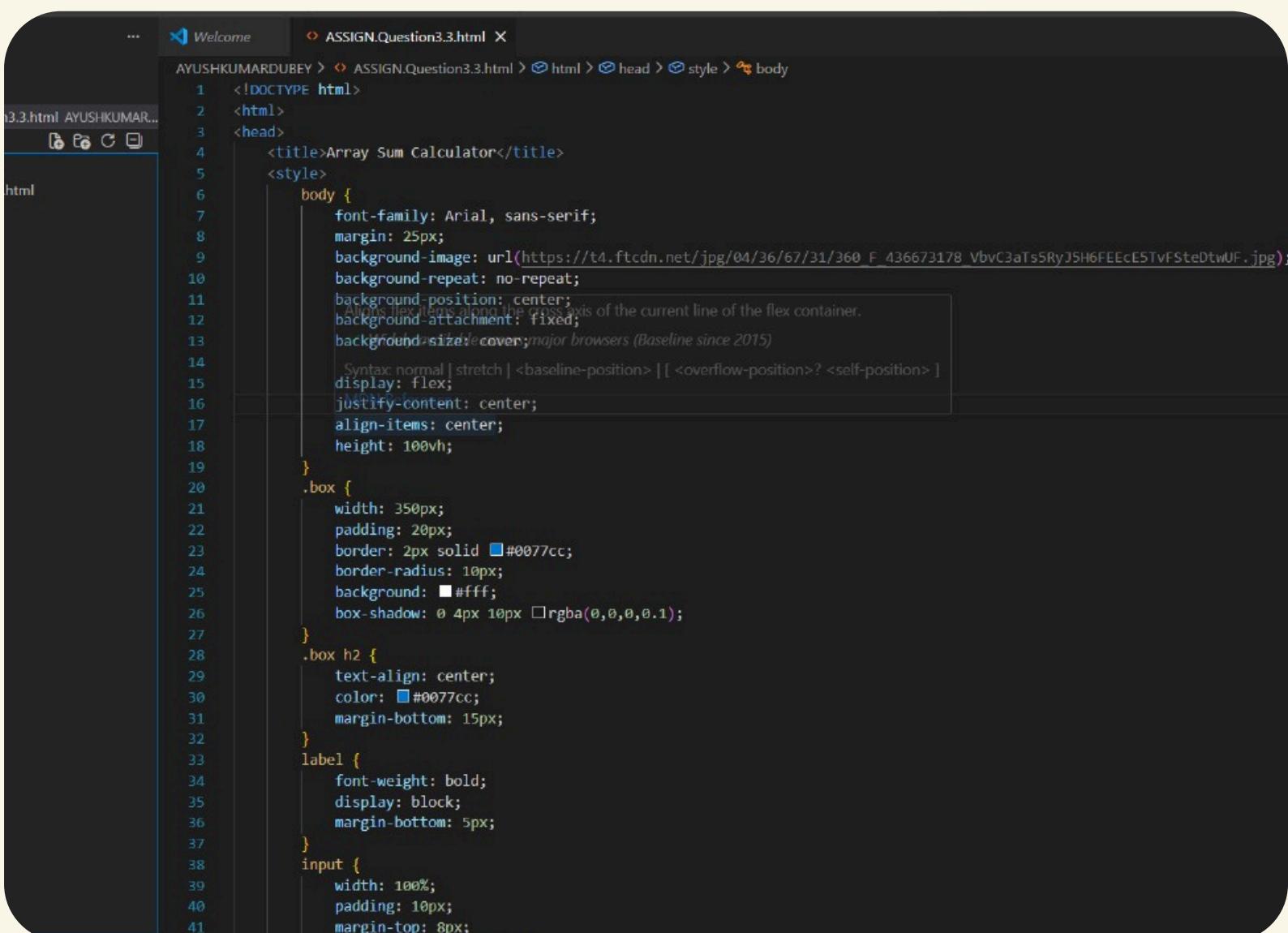
The program successfully identifies and removes all duplicate values, producing an array that contains only unique elements.

# Q. 3.3 Find the sum of all numbers in an array.

This program calculates the total sum of all numbers present in an array. It takes a list of numerical values as input, iterates through each element, and adds them together using a loop or built-in method. The final output displays the complete sum of the array elements.

## Key Points:

- The program reads an array containing multiple numbers.
- Each number is processed one by one.
- A running total is maintained and updated during the loop.
- Works for positive, negative, and mixed numbers.
- Produces a single final result: the total sum.



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

- Title Bar:** Welcome > ASSIGN.Question3.3.html
- File Path:** AYUSHKUMARDUBEY > ASSIGN.Question3.3.html > html > head > style > body
- Code Content:**

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Array Sum Calculator</title>
5      <style>
6          body {
7              font-family: Arial, sans-serif;
8              margin: 25px;
9              background-image: url(https://t4.ftcdn.net/jpg/04/36/67/31/360\_F\_436673178\_VbvC3aTs5RyJ5H6FEEcE5TvFStedtwUF.jpg);
10             background-repeat: no-repeat;
11             background-position: center;
12             /* Aligns flex items along the cross axis of the current line of the flex container. */
13             background-attachment: fixed;
14             background-size: cover; /* major browsers (Baseline since 2015)
15               Syntax: normal | stretch | <baseline-position> | [ <overflow-position>? <self-position> ]
16               display: flex; */
17             justify-content: center;
18             align-items: center;
19             height: 100vh;
20         }
21         .box {
22             width: 350px;
23             padding: 20px;
24             border: 2px solid #0077cc;
25             border-radius: 10px;
26             background: #ffff;
27             box-shadow: 0 4px 10px rgba(0,0,0,0.1);
28         }
29         .box h2 {
30             text-align: center;
31             color: #0077cc;
32             margin-bottom: 15px;
33         }
34         label {
35             font-weight: bold;
36             display: block;
37             margin-bottom: 5px;
38         }
39         input {
40             width: 100%;
41             padding: 10px;
42             margin-top: 8px;
```

### 3.3 Find the sum of all numbers in an array.

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

- OPEN EDITORS:** Welcome, ASSIGN.Question3.3.html (AYUSHKUMAR...)
- BCA\_1A:** AYUSHKUMARDUBEY, ASSIGN.Question3.3.html
- Outline:** OUTLINE
- HTML Content:**

```
<html>
  <head>
    <style>
      input {
        border: 1px solid #ccc;
        border-radius: 6px;
      }
      button {
        margin-top: 12px;
        padding: 10px 20px;
        background: #0077cc;
        color: #fff;
        border: none;
        border-radius: 6px;
        cursor: pointer;
      }
      #answer {
        margin-top: 15px;
        font-weight: bold;
        color: #333;
        text-align: center;
        font-size: 18px;
      }
    </style>
  </head>
  <body>
    <div class="box">
      <h2>sum of Values in an Array</h2>
      <label>Type numbers separated by commas:</label>
      <input type="text" id="nums" placeholder="Example: 4, 9, 6, 2">
      <button onclick="calculate()">Get Sum</button>
      <p id="answer"></p>
    </div>
    <script>
      function calculate(){
        let text = document.getElementById("nums").value;
        let items = text.split(",").map(x => Number(x.trim()));
        let total = items.reduce((acc, val) => acc + val, 0);
        document.getElementById("answer").innerHTML = "Total sum: " + total;
      }
    </script>
  </body>
</html>
```
- Style:** The CSS styles define the appearance of input fields and buttons. Input fields have a border and rounded corners. Buttons have a dark blue background, white text, and a larger font size. The answer is displayed in a bold, dark blue font centered below the input area.
- Script:** The JavaScript code contains a single function, calculate(). It retrieves the value from the input field, splits it into an array of strings, maps each string to a number, and then reduces the array to a total sum. Finally, it updates the innerHTML of the #answer paragraph to display the result.

Example:

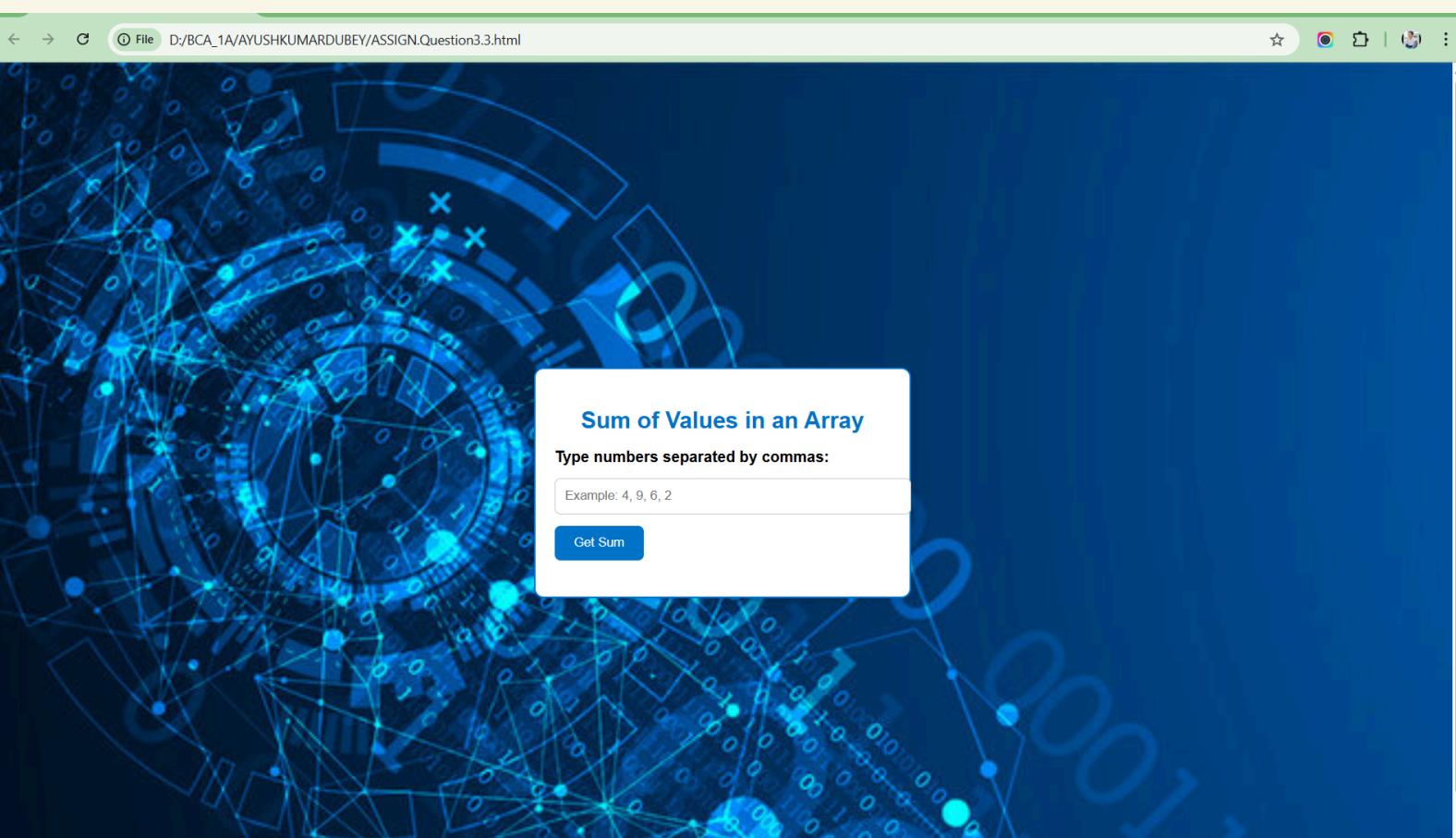
If the input array is: "[4, 7, 2, 9]"

The calculation will be:

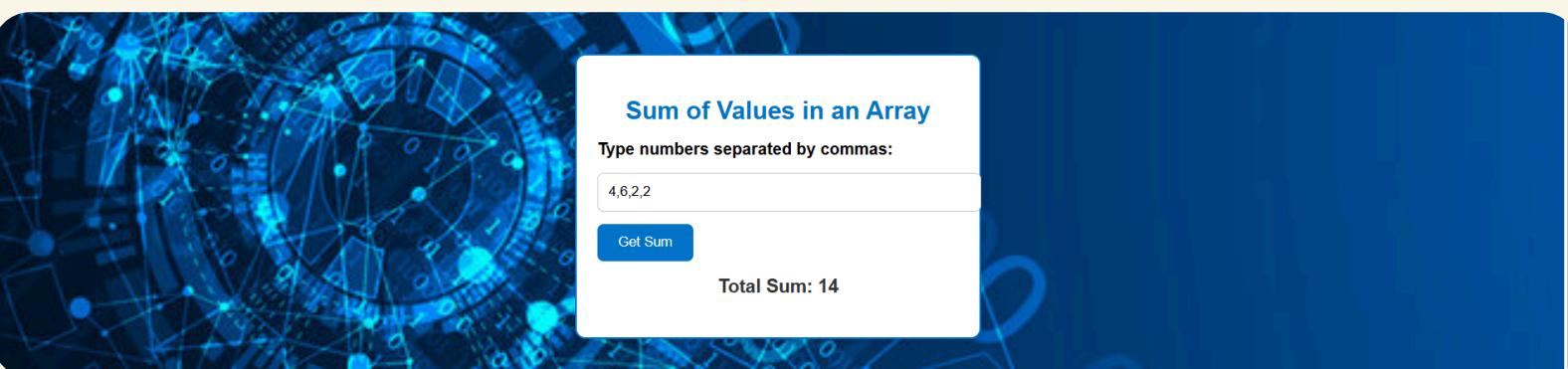
$$4 + 7 + 2 + 9 = 22$$

Final Output: 22

# output



Add numeric values using ( , )



## Conclusion:

The program accurately computes the sum of all elements in the array by repeatedly adding each value to a running total, ensuring a correct final result.