Grid Assignment

1. **Create an image gallery using a CSS grid.**

Ans: HTML Code:

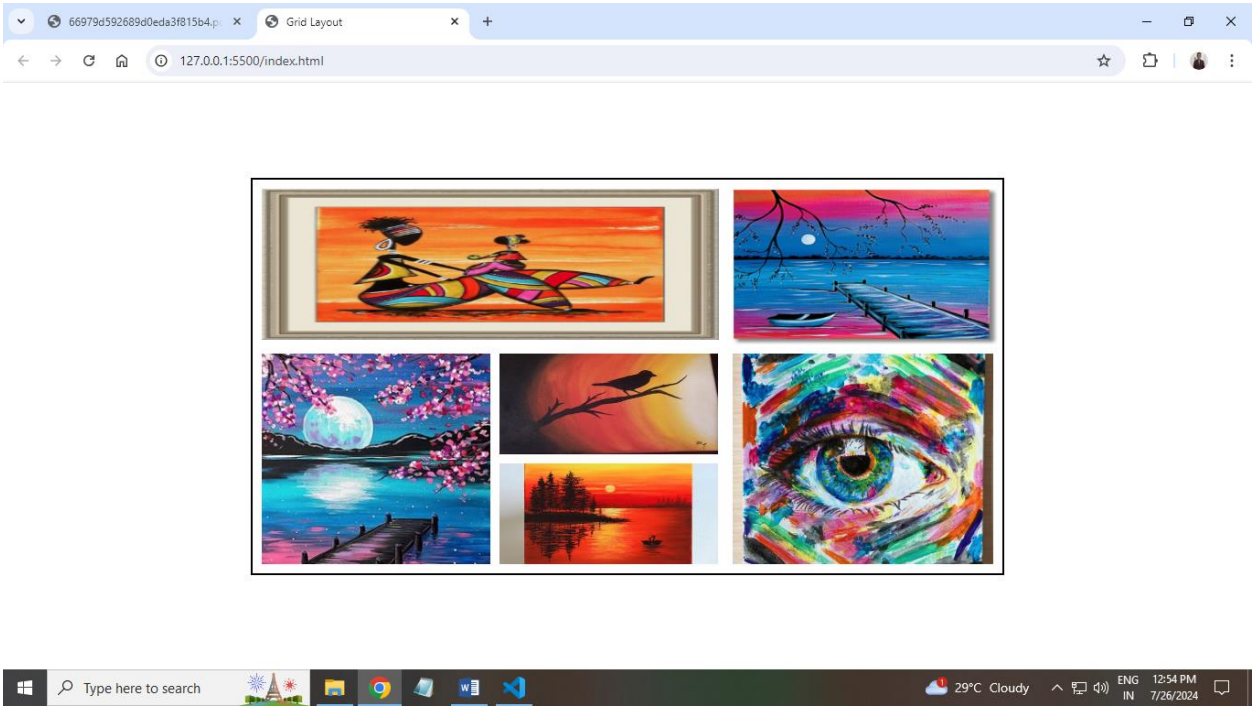
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Grid Layout</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="grid-container">
    <img src="image1.jpg" alt="Image 1" class="item1">
    <img src="image2.jpg" alt="Image 2" class="item2">
    <img src="image3.jpg" alt="Image 3" class="item3">
    <div class="combined">
      <img src="image4.jpg" alt="Image 4" class="item4">
      <img src="image5.jpg" alt="Image 5" class="item5">
    </div>
    <img src="image6.jpg" alt="Image 6" class="item6">
  </div>
</body>
</html>
```

CSS Code:

```
body {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  font-family: 'Times New Roman', Times, serif;
}
```

```
.grid-container {
   display: grid;
   grid-template-columns: repeat(3, 1fr);
   grid-template-rows: auto auto;
   gap: 10px;
   width: 80%;
   max-width: 800px;
   border: 2px solid black;
   padding: 10px;
}

.item1 {
   grid-column: span 2;
   width: 500px;
   height: 165px;
}

.item2 {
   grid-column: 3 / 4;
   width: 290px;
   height: 170px;
   margin-right: 40px;
}

.item3 {
   grid-column: 1 / 2;
   grid-row: 2 / 3;
   width: 250px;
   height: 230px;
}

.combined {
   display: grid;
   grid-template-rows: 1fr 1fr;
```

```
    grid-row: 2 / 3;
    grid-column: 2 / 3;
}

.item4,
.item5 {
    width: 97.5%;
    height: 110px;
}

.item5 {
    margin-top: 5px;
}

.item6 {
    grid-column: 3 / 4;
    grid-row: 2 / 3;
    width: 285px;
    height: 230px;
}
```



2. **Write code to arrange containers with texts A, B, C, and D as shown in the below image.**

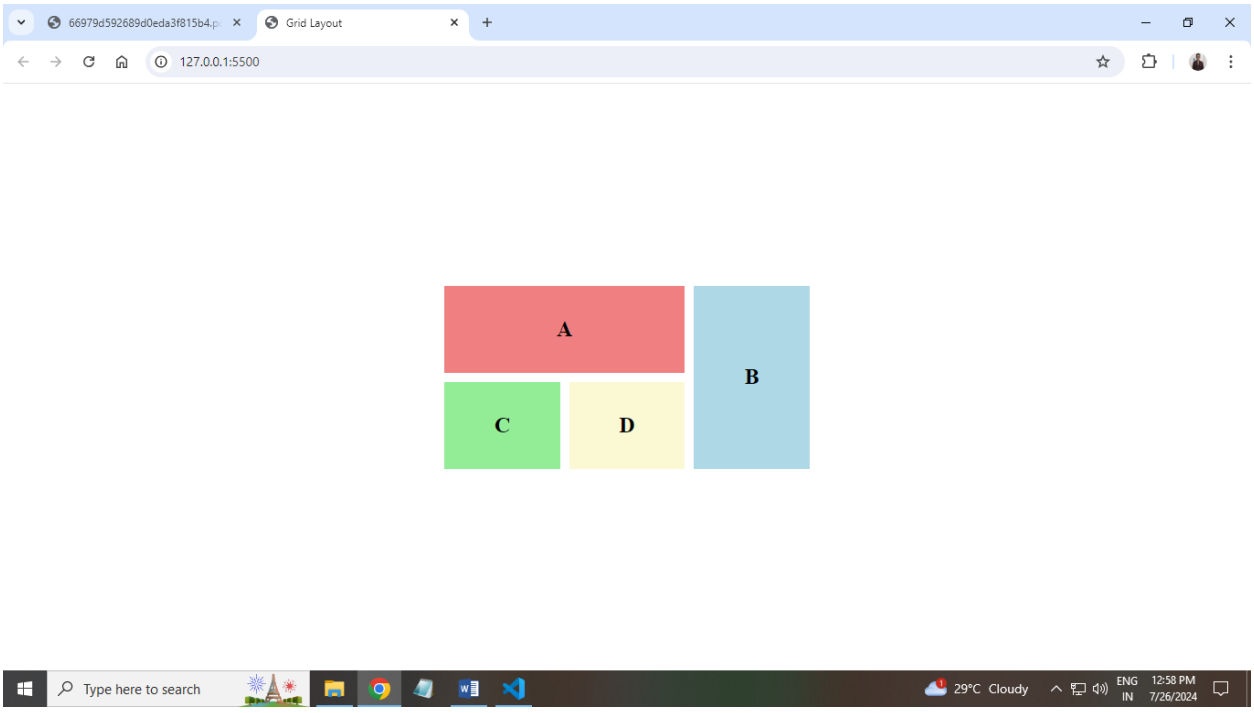
Ans: HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="style.css">
<title>Grid Layout</title>
</head>
<body>
  <div class="grid-container">
    <div class="grid-item item-a">A</div>
    <div class="grid-item item-b">B</div>
    <div class="grid-item item-c">C</div>
    <div class="grid-item item-d">D</div>
  </div>
</body>
</html>
```

CSS Code:

```
html,
body {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
  font-family: 'Times New Roman', Times, serif;
}

.grid-container {
  display: grid;
```

```
  grid-template-columns: repeat(3, 1fr);
  grid-template-rows: repeat(2, 1fr);
  gap: 10px;
  width: 400px;
  height: 200px;
}

.item-a {
  grid-column: span 2;
  background-color: lightcoral;
}

.item-b {
  grid-row: span 2;
  background-color: lightblue;
}

.item-c {
  background-color: lightgreen;
}

.item-d {
  background-color: lightgoldenrodyellow;
}

.grid-item {
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 24px;
  font-weight: bold;
}
```



3. **Explain the use of grid-auto-row and grid-auto-column using code examples.**

   Ans: <u>Introduction to Grid Auto Row and Column</u>:

   - **grid-auto-rows and grid-auto-columns are properties in CSS Grid Layout used to define the size of rows and columns that are created implicitly. When you create a grid and add more items than the specified rows or columns can handle, CSS Grid will automatically create additional rows or columns to accommodate the extra items. The grid-auto-rows and grid-auto-columns properties specify the size of these additional rows or columns.**
   - **Example of Grid Auto Row:**

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>grid-auto-rows Example</title>
  <style>
    .grid-container {
```

```
      display: grid;
      grid-template-columns: 100px 100px;
      grid-auto-rows: 50px;
      gap: 10px;
    }
    .grid-item {
      background-color: lightblue;
      border: 1px solid blue;
      text-align: center;
      padding: 20px;
    }
  </style>
</head>
<body>
  <div class="grid-container">
    <div class="grid-item">1</div>
    <div class="grid-item">2</div>
    <div class="grid-item">3</div>
    <div class="grid-item">4</div>
    <div class="grid-item">5</div>
  </div>
</body>
</html>
```

- **Grid Auto Columns Example**:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>grid-auto-columns Example</title>
  <style>
    .grid-container {
      display: grid;
      grid-template-rows: 100px 100px;
```

```
            grid-auto-columns: 50px;
            gap: 10px;
        }
        .grid-item {
            background-color: lightgreen;
            border: 1px solid green;
            text-align: center;
            padding: 20px;
        }
    </style>
</head>
<body>
    <div class="grid-container">
        <div class="grid-item">1</div>
        <div class="grid-item">2</div>
        <div class="grid-item">3</div>
        <div class="grid-item">4</div>
        <div class="grid-item">5</div>
    </div>
</body>
</html>
```

4. **Write CSS to show numbers as shown in the figure, without altering the below html code.**

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

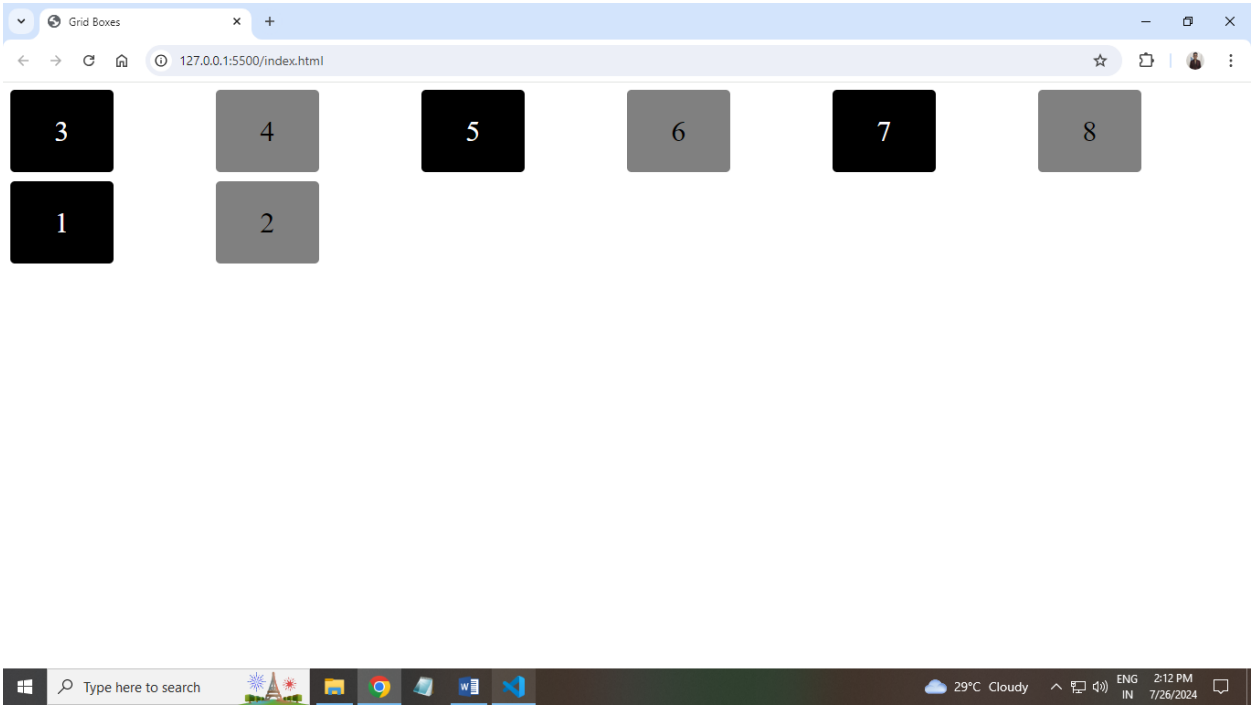
```
      <div class="box box1">1</div>
      <div class="box box2">2</div>
      <div class="box box3">3</div>
      <div class="box box4">4</div>
      <div class="box box5">5</div>
      <div class="box box6">6</div>
      <div class="box box7">7</div>
      <div class="box box8">8</div>
   </div>
</body>
</html>
```

CSS Code:

```
.container {
   display: grid;
   grid-template-columns: repeat(6, 0.2fr);
   grid-template-rows: repeat(2, 1fr);
}

.box {
   display: flex;
   justify-content: center;
   align-items: center;
   height: 70px;
   font-size: 2em;
   border-radius: 5px;
   width: 50%;
   box-sizing: border-box;
   height: 90px;
}

.box1 {
   grid-area: 2 / 1 / 3 / 2;
   background-color: black;
```

```
    color: white;
    margin-top: 5px;
}

.box2 {
    grid-area: 2 / 2 / 3 / 3;
    background-color: grey;
    margin-top: 5px;
}

.box3 {
    grid-area: 1 / 1 / 2 / 2;
    background-color: black;
    color: white;
}

.box4 {
    grid-area: 1 / 2 / 2 / 3;
    background-color: grey;
}

.box5 {
    grid-area: 1 / 3 / 2 / 4;
    background-color: black;
    color: white;
}

.box6 {
    grid-area: 1 / 4 / 2 / 5;
    background-color: grey;
}

.box7 {
    grid-area: 1 / 5 / 2 / 6;
    background-color: black;
```

```
    color: white;
}

.box8 {
    grid-area: 1 / 6 / 2 / 7;
    background-color: grey;
}
```



5. **Explain the difference between justify-items and justify-self using code examples.**

   Ans: <u>Introduction to justify-items and justify-self</u>:

   - **In CSS Grid, justify-items and justify-self are used to align grid items along the inline (row) axis. They control the positioning of the content within the grid items.**
   - **Example Code of justify-items:**

     ```
     <!DOCTYPE html>
     <html lang="en">
     <head>
     <meta charset="UTF-8">
     ```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Justify Items Example</title>
  <style>
    .grid-container {
      display: grid;
      grid-template-columns: repeat(3, 1fr);
      grid-template-rows: 100px;
      justify-items: center; /* Center all items */
      gap: 10px;
      border: 1px solid black;
    }

    .grid-item {
      background-color: lightblue;
      padding: 20px;
      border: 1px solid blue;
    }
  </style>
</head>
<body>
  <div class="grid-container">
    <div class="grid-item">Item 1</div>
    <div class="grid-item">Item 2</div>
    <div class="grid-item">Item 3</div>
  </div>
</body>
</html>
```

- Example Code of justify-self:

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

```
  <title>Justify Self Example</title>
  <style>
    .grid-container {
      display: grid;
      grid-template-columns: repeat(3, 1fr);
      grid-template-rows: 100px;
      gap: 10px;
      border: 1px solid black;
    }

    .grid-item {
      background-color: lightblue;
      padding: 20px;
      border: 1px solid blue;
    }

    .item1 {
      justify-self: start;
}

    .item2 {
      justify-self: center;
    }

    .item3 {
      justify-self: end;
}
  </style>
</head>
<body>
  <div class="grid-container">
    <div class="grid-item item1">Item 1</div>
    <div class="grid-item item2">Item 2</div>
    <div class="grid-item item3">Item 3</div>
  </div>
```

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