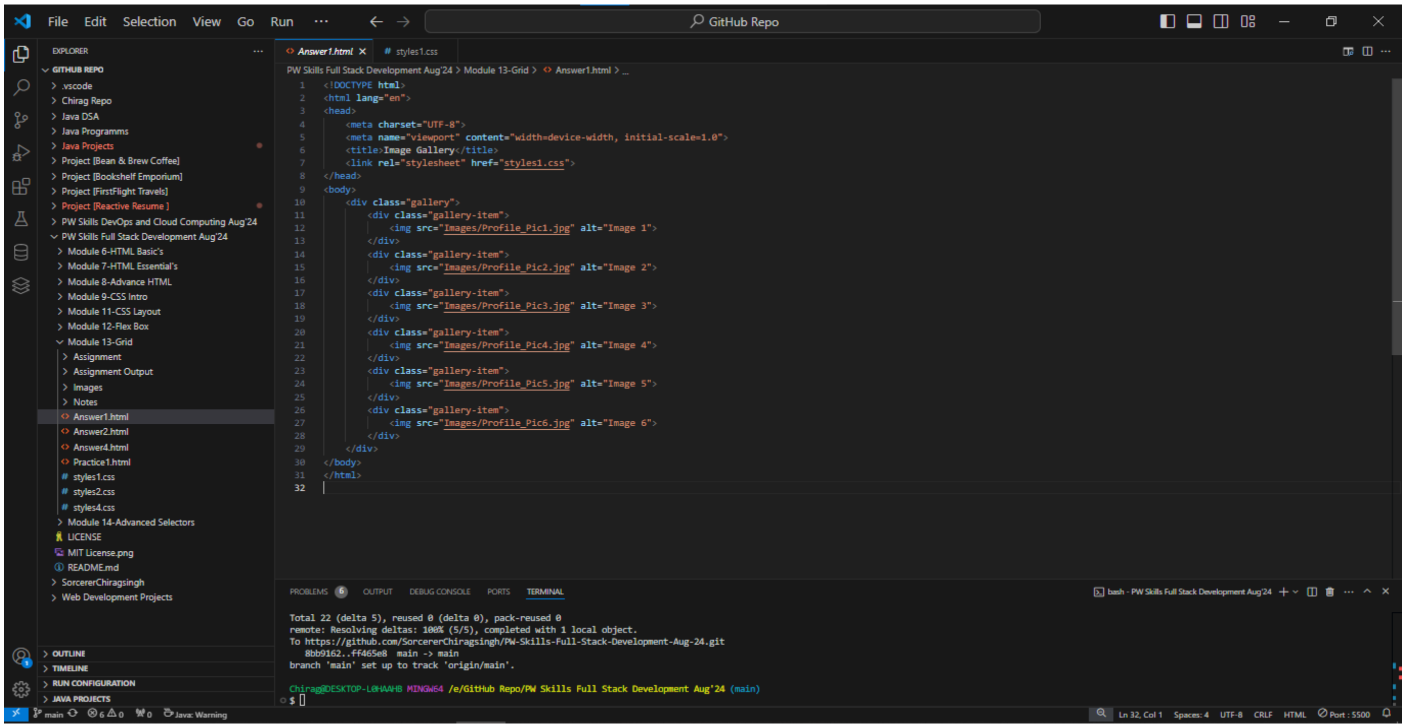
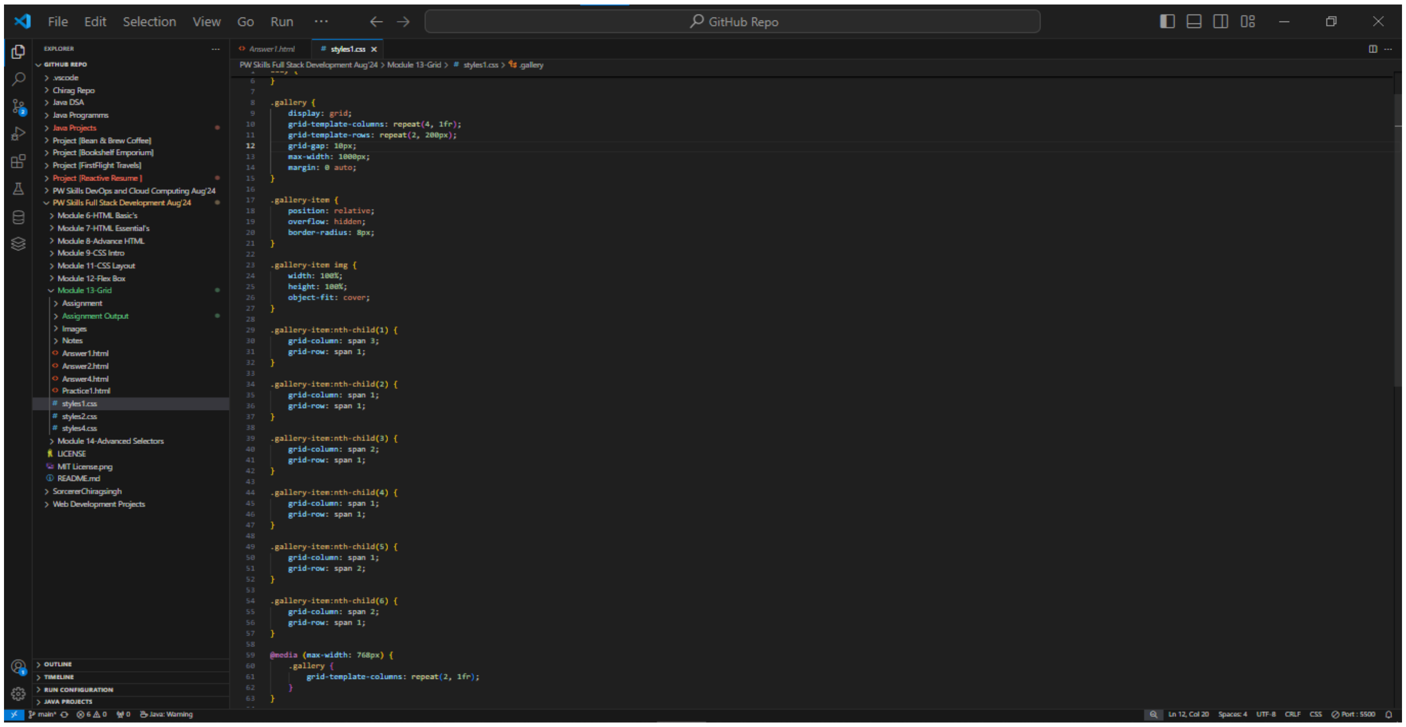
Module -13 Assignment

Q1.) Problem Statement: Create an image gallery using a CSS grid.



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image Gallery</title>
    <link rel="stylesheet" href="styles1.css">
</head>
<body>
    <div class="gallery">
        <div class="gallery-item">
            <img src="Images/Profile_Pic1.jpg" alt="Image 1">
        </div>
        <div class="gallery-item">
            <img src="Images/Profile_Pic2.jpg" alt="Image 2">
        </div>
        <div class="gallery-item">
            <img src="Images/Profile_Pic3.jpg" alt="Image 3">
        </div>
        <div class="gallery-item">
            <img src="Images/Profile_Pic4.jpg" alt="Image 4">
        </div>
        <div class="gallery-item">
            <img src="Images/Profile_Pic5.jpg" alt="Image 5">
        </div>
        <div class="gallery-item">
            <img src="Images/Profile_Pic6.jpg" alt="Image 6">
        </div>
    </div>
</body>
</html>
```

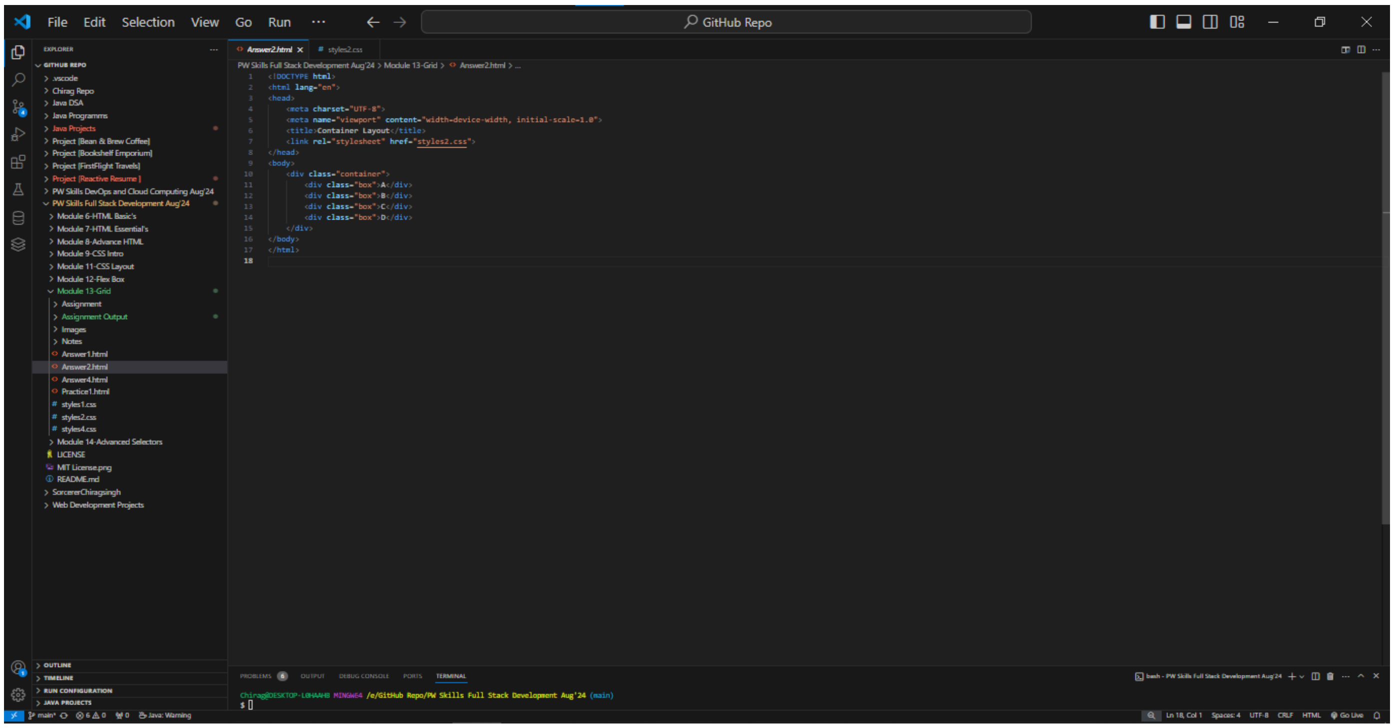
```
Total 22 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 1 local object.
To https://github.com/SorcererChiragsingh/PW-Skills-Full-Stack-Development-Aug-24.git
   8bb9162..ff465e8  main -> main
branch 'main' set up to track 'origin/main'.

Chirag@DESKTOP-L0HAAHB MINGW64 /e/GitHub Repo/PW Skills Full Stack Development Aug'24 (main)
$ 
```



```css
}

.gallery {
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    grid-template-rows: repeat(2, 200px);
    grid-gap: 10px;
    max-width: 1000px;
    margin: 0 auto;
}

.gallery-item {
    position: relative;
    overflow: hidden;
    border-radius: 8px;
}

.gallery-item img {
    width: 100%;
    height: 100%;
    object-fit: cover;
}

.gallery-item:nth-child(1) {
    grid-column: span 3;
    grid-row: span 1;
}

.gallery-item:nth-child(2) {
    grid-column: span 1;
    grid-row: span 1;
}

.gallery-item:nth-child(3) {
    grid-column: span 2;
    grid-row: span 1;
}

.gallery-item:nth-child(4) {
    grid-column: span 1;
    grid-row: span 1;
}

.gallery-item:nth-child(5) {
    grid-column: span 1;
    grid-row: span 2;
}

.gallery-item:nth-child(6) {
    grid-column: span 2;
    grid-row: span 1;
}

@media (max-width: 768px) {
    .gallery {
        grid-template-columns: repeat(2, 1fr);
    }
}
```

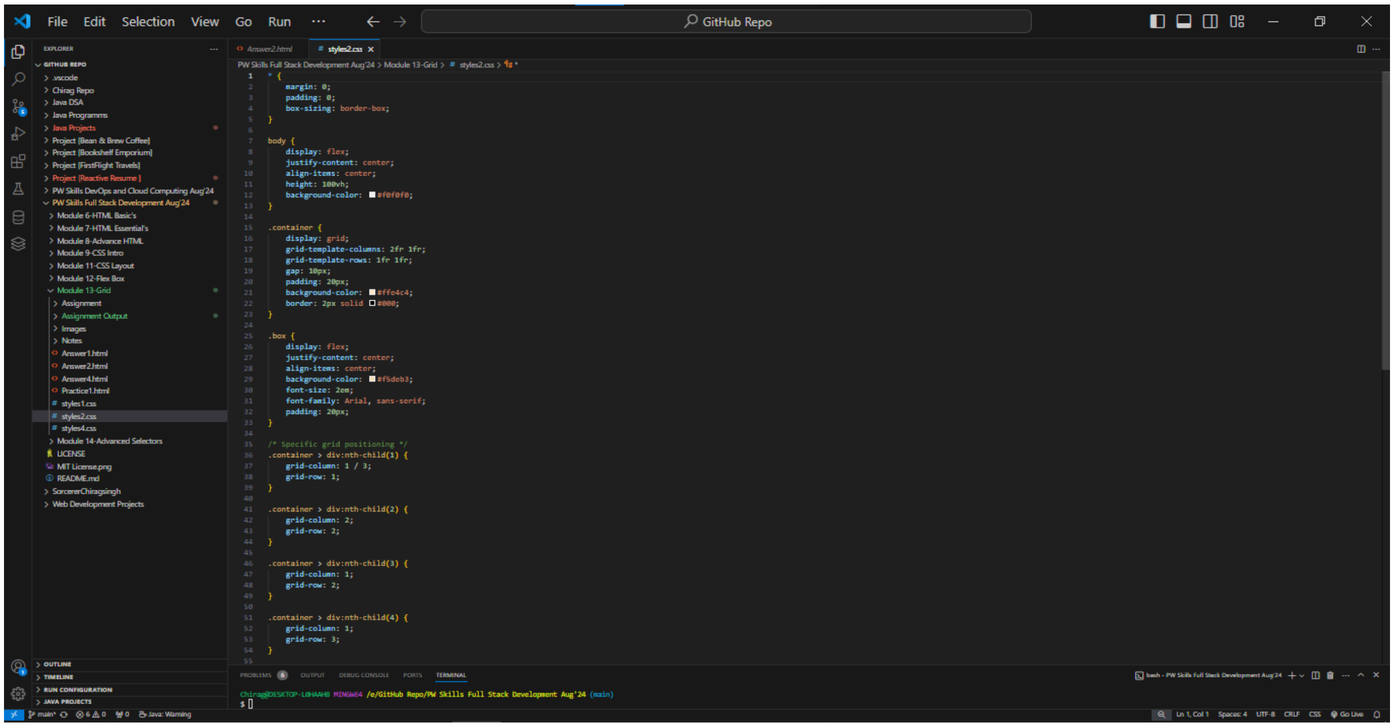
38

Q2.) Problem Statement: Write code to arrange containers with texts A, B, C, and D as shown in the below image.



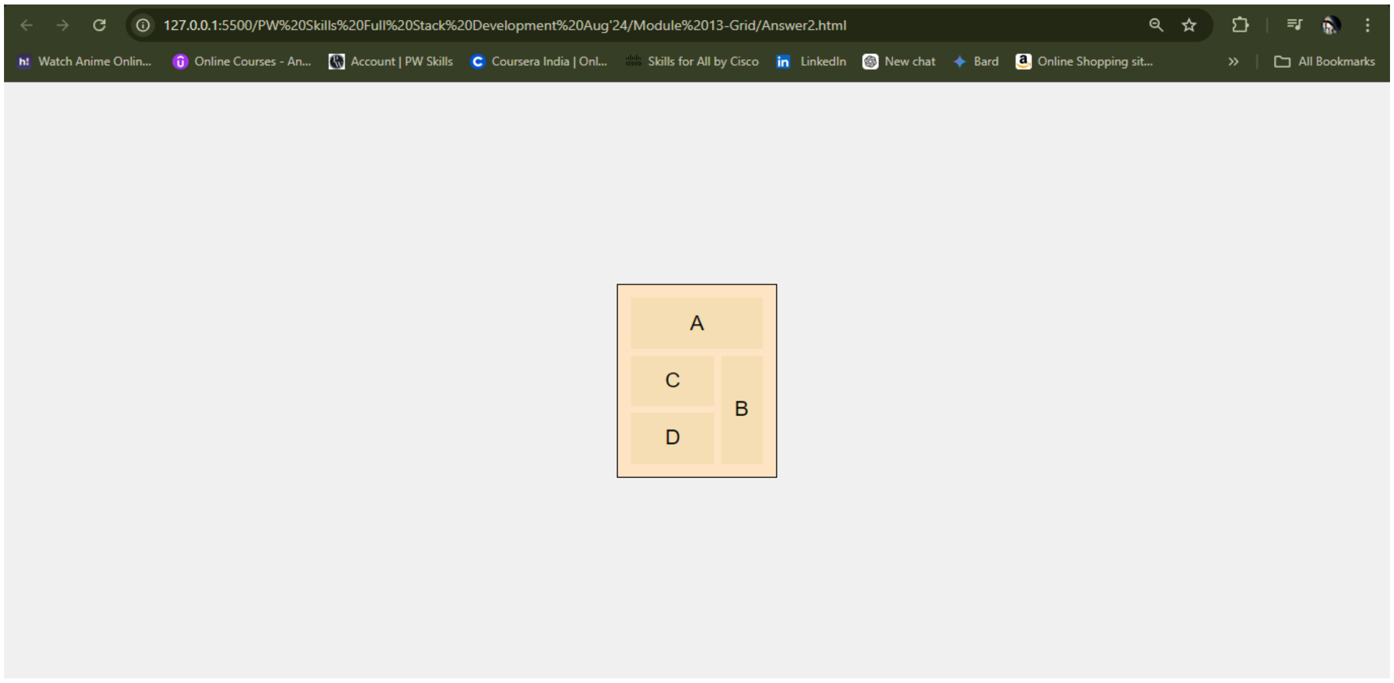
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Container Layout</title>
    <link rel="stylesheet" href="styles2.css">
</head>
<body>
    <div class="container">
        <div class="box">A</div>
        <div class="box">B</div>
        <div class="box">C</div>
        <div class="box">D</div>
    </div>
</body>
</html>
```

```
Chirag@DESKTOP-L0HAAHB MINGW64 /e/GitHub Repo/PW Skills Full Stack Development Aug'24 (main)
$ 
```



```css
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f0f0f0;
}

.container {
    display: grid;
    grid-template-columns: 2fr 1fr;
    grid-template-rows: 1fr 1fr;
    gap: 10px;
    padding: 20px;
    background-color: #ffe4c4;
    border: 2px solid #000;
}

.box {
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #f5deb3;
    font-size: 2em;
    font-family: Arial, sans-serif;
    padding: 20px;
}

/* Specific grid positioning */
.container > div:nth-child(1) {
    grid-column: 1 / 3;
    grid-row: 1;
}

.container > div:nth-child(2) {
    grid-column: 2;
    grid-row: 2;
}

.container > div:nth-child(3) {
    grid-column: 1;
    grid-row: 2;
}

.container > div:nth-child(4) {
    grid-column: 1;
    grid-row: 3;
}
```

```
Chirag@DESKTOP-L0HAAHB MINGW64 /e/GitHub Repo/PW Skills Full Stack Development Aug'24 (main)
$ 
```


127.0.0.1:5500/PW%20Skills%20Full%20Stack%20Development%20Aug'24/Module%2013-Grid/Answer2.html
Watch Anime Onlin...
Online Courses - An...
Account | PW Skills
Coursera India | Onl...
Skills for All by Cisco
LinkedIn
New chat
Bard
Online Shopping sit...
All Bookmarks
A
C
B
D

Q3.) Problem Statement: Explain the use of grid-auto-row and grid-auto-column using code examples.

The grid-auto-rows and grid-auto-columns properties in CSS Grid Layout control the size of implicitly created rows and columns. These rows and columns are created automatically by the browser when there aren't enough rows or columns defined to fit the placed grid items.

1. **grid-auto-rows:** Specifies the size for automatically created rows (i.e., rows created by the grid layout when an item is placed outside the explicitly defined rows).
   Example:
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px 100px; /* Two columns */
      grid-auto-rows: 150px; /* Automatically created rows will be 150px tall */
      gap: 10px;
      background-color: #f0f0f0;
    }

    .box {
      background-color: #ffb6c1;
      display: flex;
      justify-content: center;
      align-items: center;
      font-size: 1.5em;
      height: 100px;
    }

    /* Place the fourth box in the second row, which will be automatically created */
    .box:nth-child(4) {
      grid-column: 1; /* First column */
      grid-row: 2; /* Second row (auto-created) */
```

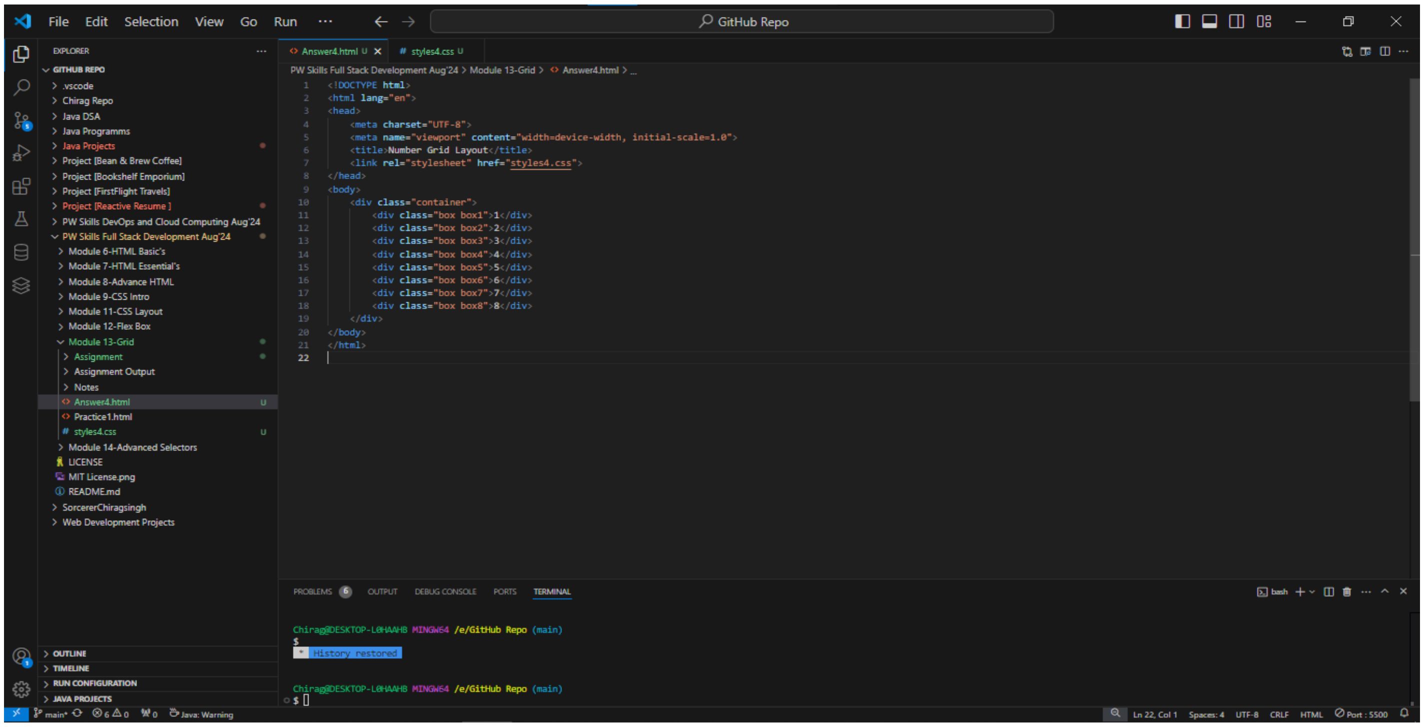
```
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="box">1</div>
        <div class="box">2</div>
        <div class="box">3</div>
        <div class="box">4</div> <!-- This will be in the automatically created second
row -->
    </div>
</body>
</html>
```

2. **grid-auto-columns:** Specifies the size for automatically created columns (i.e., columns created when an item is placed outside the explicitly defined columns).

   Example:
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        .container {
            display: grid;
            grid-template-rows: 100px; /* One row */
            grid-template-columns: 100px 100px; /* Two columns */
            grid-auto-columns: 150px; /* Automatically created columns will be 150px wide */
            gap: 10px;
            background-color: #f0f0f0;
        }

        .box {
            background-color: #ffb6c1;
            display: flex;
            justify-content: center;
            align-items: center;
            font-size: 1.5em;
        }
        /* Place the fourth box in the third column, which will be automatically created */
        .box:nth-child(4) {
            grid-column: 3; /* Third column (auto-created) */
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="box">1</div>
        <div class="box">2</div>
        <div class="box">3</div>
        <div class="box">4</div> <!-- This will be in the automatically created third column -->
    </div>
</body>
</html>
```

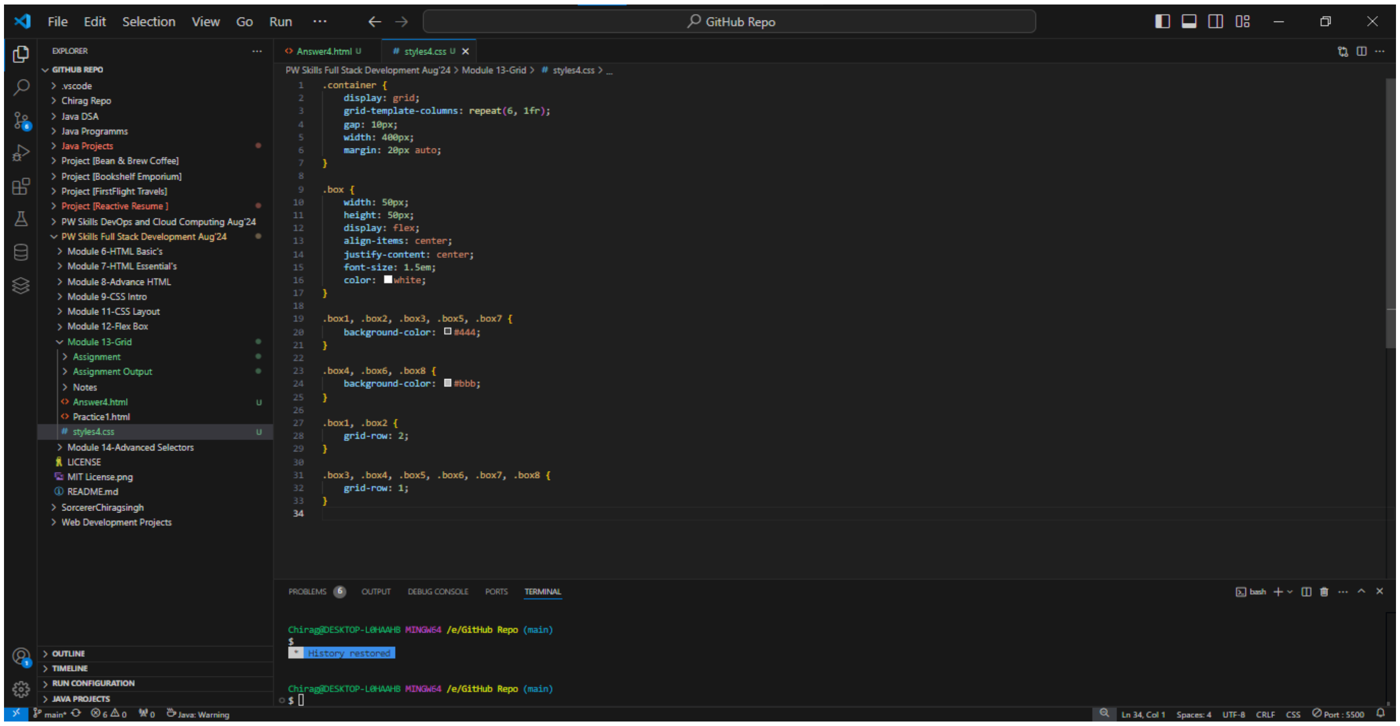
Q4.) Problem Statement: Write CSS to show numbers as shown in the figure, without altering the below html code.

```
<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>
```



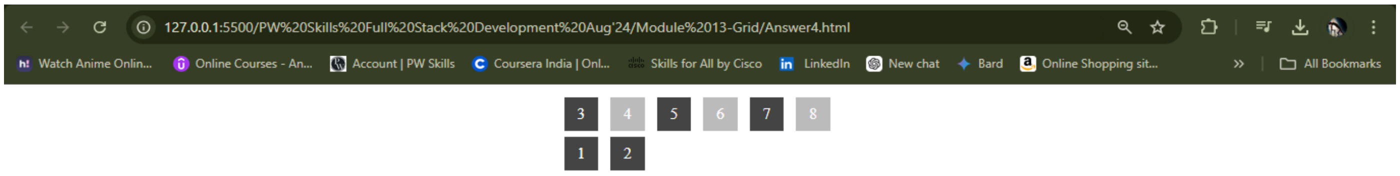
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Number Grid Layout</title>
    <link rel="stylesheet" href="styles4.css">
</head>
<body>
    <div class="container">
        <div class="box box1">1</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>
```

```
Chirag@DESKTOP-L0HAAHB MINGW64 /e/GitHub Repo (main)
$
 *  History restored

Chirag@DESKTOP-L0HAAHB MINGW64 /e/GitHub Repo (main)
$
```



```css
.container {
    display: grid;
    grid-template-columns: repeat(6, 1fr);
    gap: 10px;
    width: 400px;
    margin: 20px auto;
}

.box {
    width: 50px;
    height: 50px;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 1.5em;
    color: white;
}

.box1, .box2, .box3, .box5, .box7 {
    background-color: #444;
}

.box4, .box6, .box8 {
    background-color: #bbb;
}

.box1, .box2 {
    grid-row: 2;
}

.box3, .box4, .box5, .box6, .box7, .box8 {
    grid-row: 1;
}
```

```
Chirag@DESKTOP-L0HAAHB MINGW64 /e/GitHub Repo (main)
$
 * History restored

Chirag@DESKTOP-L0HAAHB MINGW64 /e/GitHub Repo (main)
$
```



```
127.0.0.1:5500/PW%20Skills%20Full%20Stack%20Development%20Aug'24/Module%2013-Grid/Answer4.html
```

Q5.) Problem Statement: Explain the difference between justify-items and justify-self using code examples.

In CSS Grid Layout, justify-items and justify-self are both used to control the horizontal alignment of grid items, but they work at different levels.

1. **justify-items:** Aligns **all grid items** within their grid cells along the row (horizontal) axis.

Example:
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    .container {
      display: grid;
      grid-template-columns: 100px 100px; /* Two columns */
      grid-template-rows: 100px 100px; /* Two rows */
      gap: 10px;
      justify-items: center; /* Align all items in the center horizontally */
      background-color: #f0f0f0;
    }
    .box {
      background-color: #ffb6c1;
      display: flex;
      justify-content: center;
      align-items: center;
      font-size: 1.5em;
    }
    </style>
</head>
<body>
    <div class="container">
        <div class="box">1</div>
        <div class="box">2</div>
        <div class="box">3</div>
        <div class="box">4</div>
    </div>
</body>
</html>
```

2. justify-self: Aligns a **specific grid item** within its grid cell along the row (horizontal) axis.

Example: <!DOCTYPE html>

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        .container {
            display: grid;
            grid-template-columns: 100px 100px; /* Two columns */
            grid-template-rows: 100px 100px; /* Two rows */
            gap: 10px;
            background-color: #f0f0f0;
        }
        .box {
            background-color: #ffb6c1;
            display: flex;
            justify-content: center;
            align-items: center;
            font-size: 1.5em;
        }
        /* Specific alignments for individual grid items */
        .box:nth-child(1) {
            justify-self: start; /* Align box 1 to the start of its cell */
        }
        .box:nth-child(2) {
            justify-self: end; /* Align box 2 to the end of its cell */
        }
        .box:nth-child(3) {
            justify-self: center; /* Align box 3 at the center of its cell */
        }
        .box:nth-child(4) {
            justify-self: stretch; /* Box 4 stretches to fill its cell */
        }
    </style>
</head>
<body>
    <div class="container">
```

```
        <div class="box">1</div>

        <div class="box">2</div>

        <div class="box">3</div>

        <div class="box">4</div>

    </div>

</body>

</html>
```