Grid in CSS Assignment

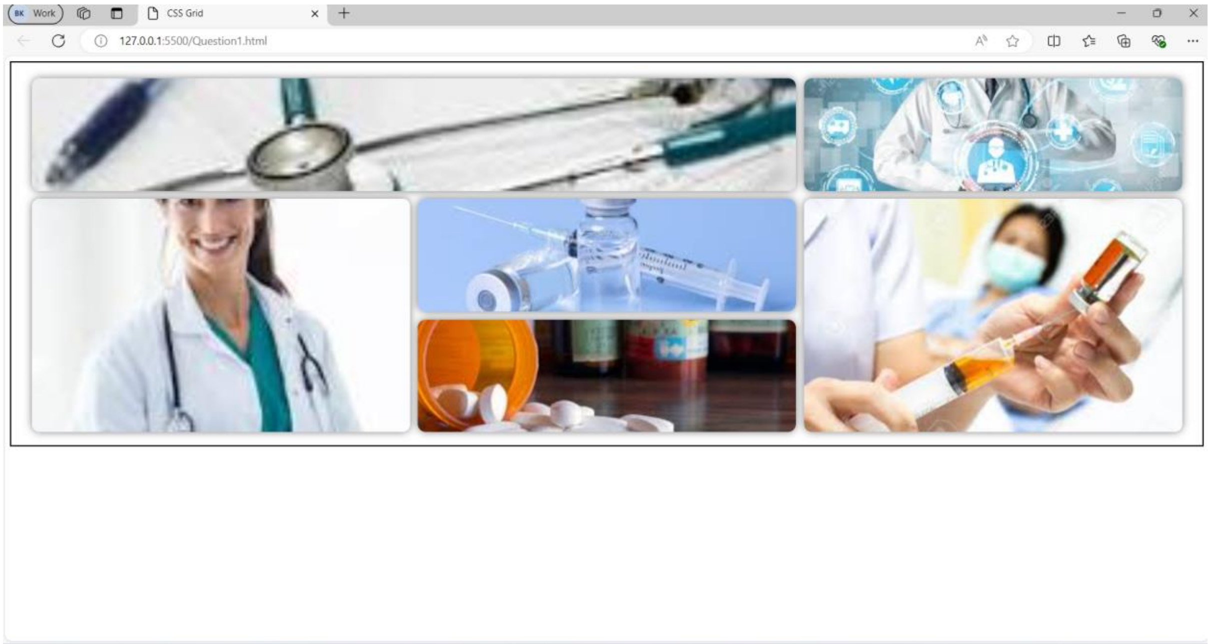
Q1. Create an Image gallery using a css grid.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CSS Grid</title>
    <style>
      .gallery {
        outline: 2px solid black;
        display: grid;
        grid-template-rows: repeat(4, 143px); /* fixed height for each row */
        grid-template-columns: repeat(3, 480px);
        grid-gap: 10px;
        height: 60vh;
        justify-content: center;
        padding: 20px;
      }

      .gallery img {
        object-fit: cover;
        width: 100%;
        height: 100%;
        border-radius: 10px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
      }
      .Img1 {
```

```
      grid-area: 1 / 1 / 2 / 3;
    }
    .Img2 {
      grid-area: 1 / 3 / 2 / 4;
    }
    .Img3 {
      grid-area: 2 / 1 / 4 / 2;
    }
    .Img4 {
      grid-area: 2 / 2 / 3 / 3;
    }
    .Img5 {
      grid-area: 3 / 2 / 4 / 3;
    }
    .Img6 {
      grid-area: 2 / 3 / 4 / 4;
    }
  </style>
</head>
<body>
  <div class="gallery">
    <img src="\Images\Img1.jpg" alt="Image 1" class="Img1" />
    <img src="\Images\img01.jpg" alt="Image 2" class="Img2" />
    <img src="\Images\img3.jpg" alt="Image 3" class="Img3" />
    <img src="\Images\img4.jpg" alt="Image 4" class="Img4" />
    <img src="\Images\img5.jpg" alt="Image 5" class="Img5" />
    <img src="\Images\img6.jpg" alt="Image 6" class="Img6" />
  </div>
</body>
```

```
</html>
```

Output :-

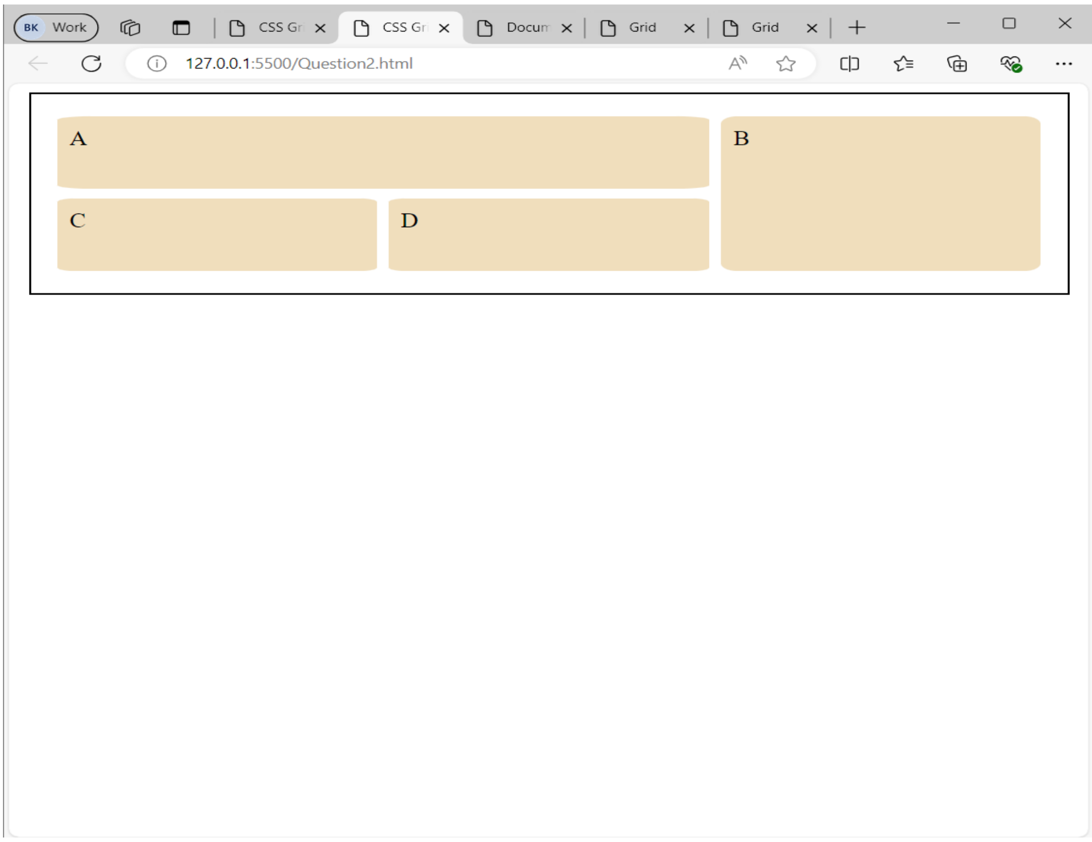


Q2. Write code to arrange containers with texts A,B,C and D .

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CSS Grid</title>
    <style>
      .container {
        display: grid;
        outline: 2px solid black;
        margin: 10px;
        padding: 3cqh;
        gap: 10px;
  height: 150px;;

  .box {
    padding: 10px;
    background-color: rgb(240, 222, 188);
    font-size: 20px;
    border-radius: 5%;
  }
}
.alpha {
  grid-row-start: 1;
  grid-row-end: 2;
  grid-column-start: 1;
  grid-column-end: 3;
}
.alphb {
  grid-row-start: 1;
  grid-row-end: 3;
  grid-column-start: 3;
  grid-column-end: 4;
}
.alphc {
  grid-row-start: 2;
  grid-row-end: 3;
  grid-column-start: 1;
  grid-column-end: 2;
}
.alphd {
  grid-row-start: 2;
```

```
        grid-row-end: 3;
        grid-column-start: 2;
        grid-column-end: 3;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="box alpha">A</div>
      <div class="box alphb">B</div>
      <div class="box alphc">C</div>
      <div class="box alphd">D</div>
    </div>
  </body>
</html>
```

Output :-



Q3. Explain the use of grid-auto-row and grid-auto-column.

grid-auto-row and grid-auto-column are two CSS Grid properties that help you control the size of rows and columns when you don't explicitly define them.
grid-auto-row is like setting a default height for all rows, so you don't have to define each one individually.

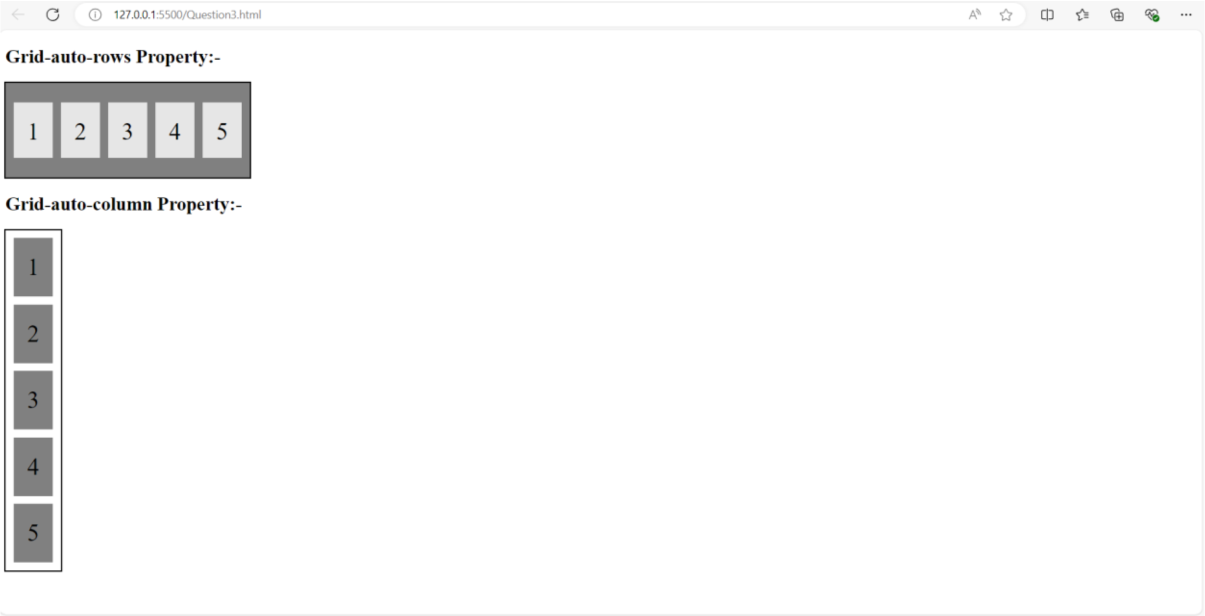
Syntax- grid-auto-row: 100px;

rid-auto-column is like setting a default width for all columns, so you don't have to define each one individually.
Syntax- grid-auto-column: 200px

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Grid</title>
    <style>
      .grid-container {
        display: grid;
        grid-template-columns: repeat(4, auto);
        grid-auto-rows: 100px;
        grid-gap: 10px;
        background-color: gray;
        padding: 10px;
        outline: 2px solid black;
        width: fit-content;
      }
      .grid-item {
        background-color: white;
        padding: 20px;
        border: 1px solid black;
        align-self: center; /*it vertically center all grid items */
      }
      #centered {
        justify-self: center; /*it horizontally center this specific grid item */
      }
    </style>
  </head>
  <body>
    <h2>align-self: center :-</h2>
    <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 class="grid-item">Item 4</div>
      <div class="grid-item">Item 5</div>
      <div class="grid-item">Item 6</div>
      <div class="grid-item">Item 7</div>
      <div class="grid-item">Item 8</div>
    </div>
    <h2>justify-self: center :-</h2>
    <div class="grid-container">
      <div class="grid-item" id="centered">
        Grid Item 1 (horizontally centered)
      </div>
      <div class="grid-item">Grid Item 2</div>
      <div class="grid-item">Grid Item 3</div>
    </div>
  </body>
```

```
</html>
```

Output :-



127.0.0.1:5500/Question3.html

Grid-auto-rows Property:-

1 2 3 4 5

Grid-auto-column Property:-

1

2

3

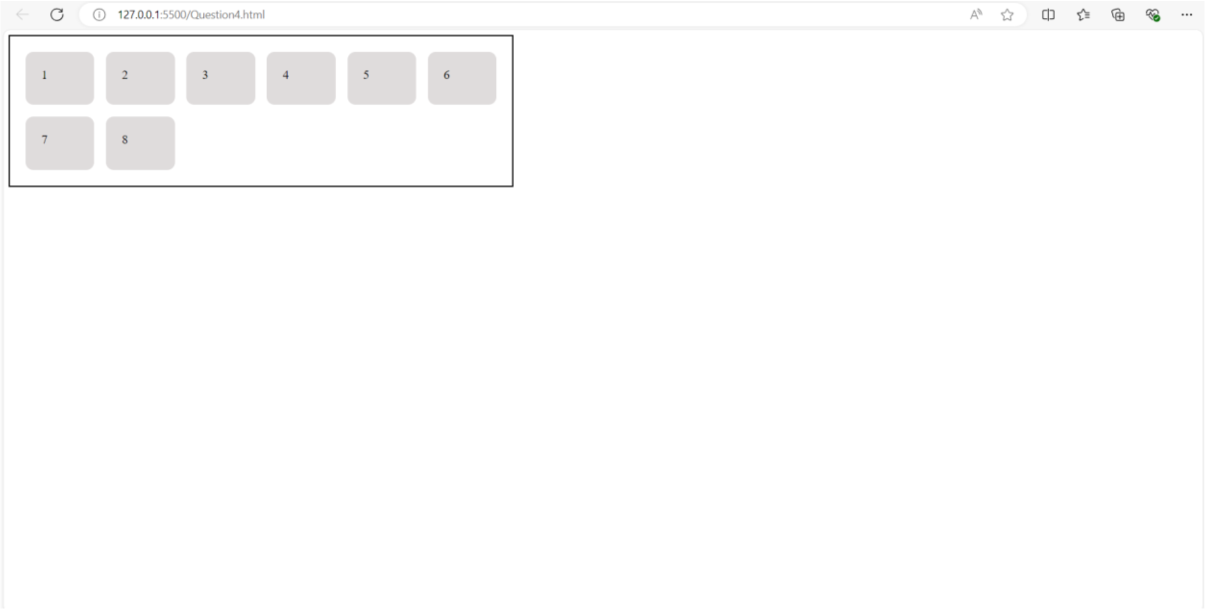
4

5

Q4. Write css to show 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">

    <title>Grid</title>

    <style>

        .container{

            display: grid;

            grid-template-columns: repeat(6, auto);

            outline: 2px solid black;

            padding: 20px;

            height:150px;

            width:600px;
            gap:15px;
        }
        .box{
            border-radius:10px;
            background-color:rgb(223, 220, 220) ;
            padding:20px;
        }
    </style>
</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>
```

Output :-



Q5.Explain the difference between justify-items and justify-self using code examples.

<!DOCTYPE html>

justify-items is used to justify all items in a container, whereas justify-self is used to justify a single item within its container.

Ex- justify-items: center; /* centers all items horizontally */

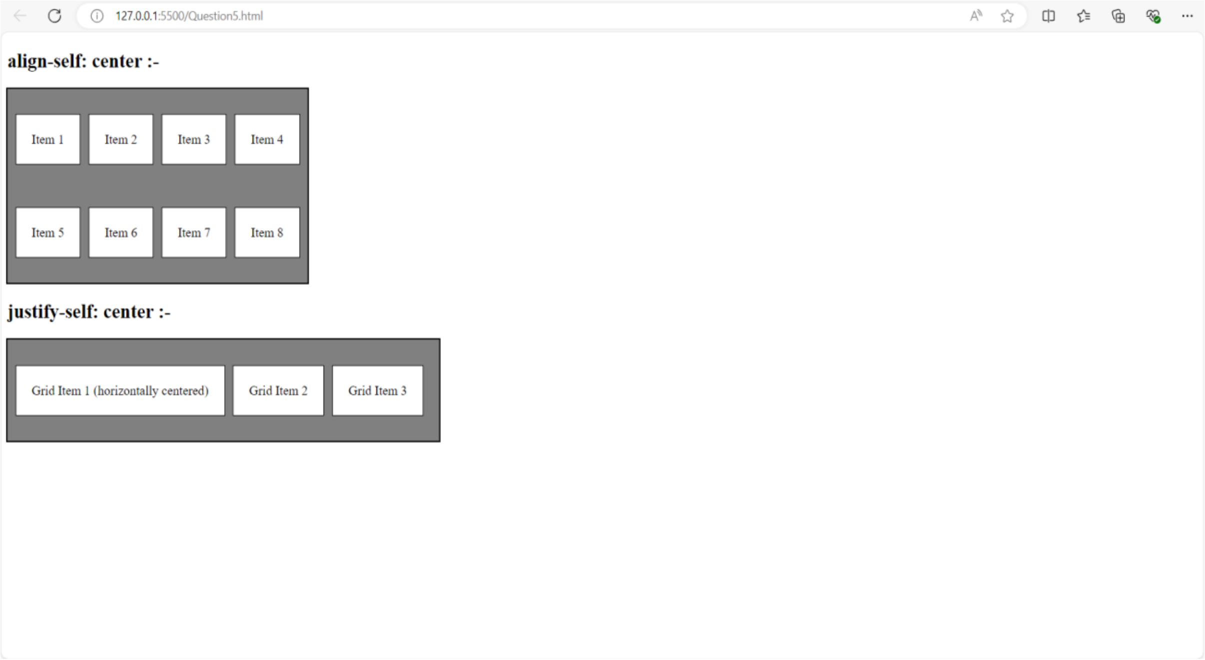
In other words, justify-items sets the default justification for all items, while justify-self overrides that default for a specific item.

Ex-   justify-self: end; /* aligns this item to the right, overriding the default */

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Grid</title>
    <style>
      .grid-container {
        display: grid;
        grid-template-columns: repeat(4, auto);
        grid-auto-rows: 100px;
        grid-gap: 10px;
        background-color: gray;
        padding: 10px;
        outline: 2px solid black;
        width: fit-content;
      }
      .grid-item {
        background-color: white;
        padding: 20px;
        border: 1px solid black;
        align-self: center; /*it vertically center all grid items */
      }
      #centered {
        justify-self: center; /*it horizontally center this specific grid item */
      }
    </style>
  </head>
  <body>
    <h2>align-self: center :-</h2>
    <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 class="grid-item">Item 4</div>
      <div class="grid-item">Item 5</div>
      <div class="grid-item">Item 6</div>
      <div class="grid-item">Item 7</div>
      <div class="grid-item">Item 8</div>
    </div>
```

```
    <h2>justify-self: center :-</h2>
    <div class="grid-container">
      <div class="grid-item" id="centered">
        Grid Item 1 (horizontally centered)
      </div>
      <div class="grid-item">Grid Item 2</div>
      <div class="grid-item">Grid Item 3</div>
    </div>
  </body>
</html>
```

Output :-



align-self: center :-

Item 1 Item 2 Item 3 Item 4

Item 5 Item 6 Item 7 Item 8

justify-self: center :-

Grid Item 1 (horizontally centered) Grid Item 2 Grid Item 3