PW SKILL ASSIGNMENT

RAJKAMAL YADAV

Task 1:

Problem Statement:

Create an image gallery using a CSS grid.

Expected Behaviour

Solution:

Code(HTML)

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

```
    <title>Grid</title>
    <link rel="stylesheet" href="grid.css">
</head>
<body>
    <h1>Create Gallary Using Grid</h1>
    <div class="container">
        <div class="item item1">
            <img src="Assets/img1.avif">
        </div>
        <div class="item">
            <img src="Assets/img2.avif">
        </div>
        <div class="item item3">
            <img src="Assets/img2.avif">
        </div>
        <div class="item item4">
            <img src="Assets/img4.avif">
        </div>
        <div class="item item5">
            <img src="Assets/img5.avif">
        </div>
        <div class="item">
            <img src="Assets/img6.avif">
        </div>
    </div>
</body>
```

```
</html>
```

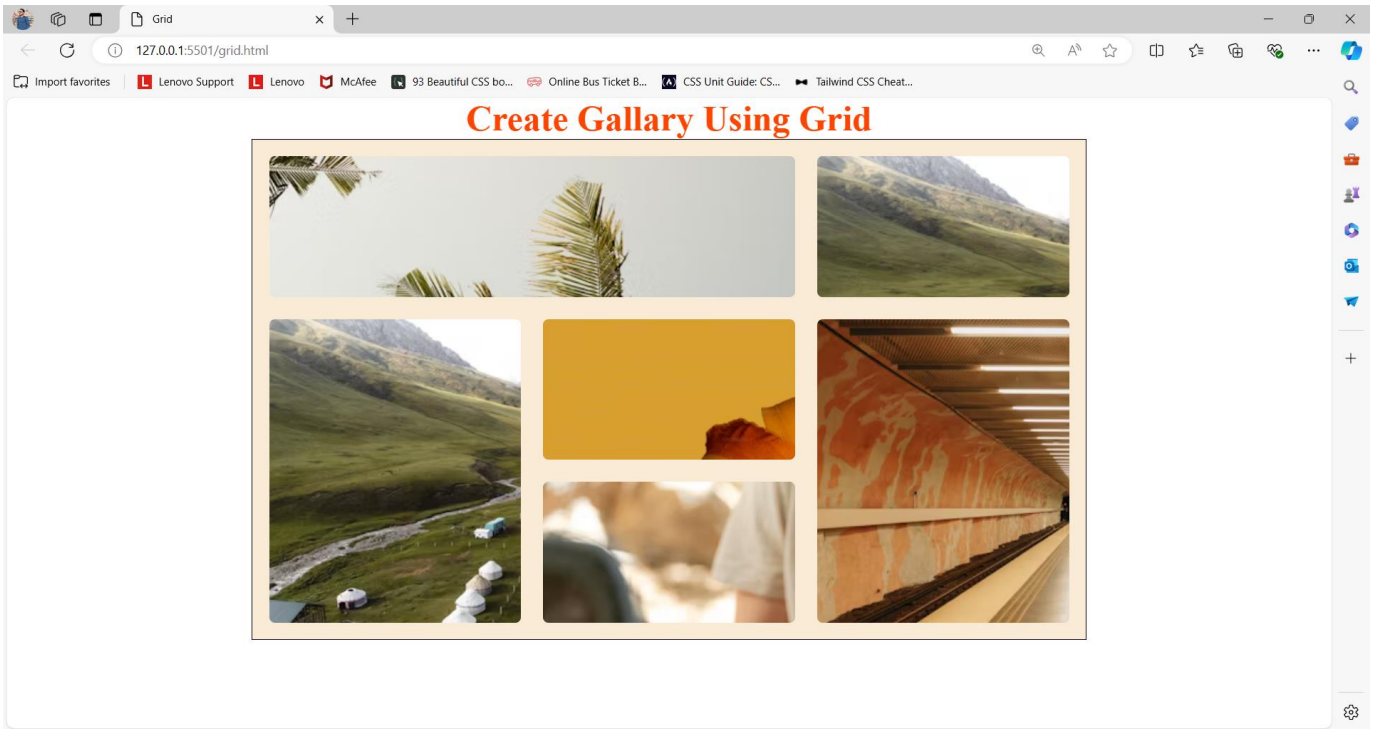
Code(CSS)

```
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
h1{
    color: orangered;
    text-align: center;
}
.container{
    margin: 0 auto;
    height: 450px;
    width: 750px;
    background-color: antiquewhite;
    border: 1px solid black;
    display: grid;
    grid-template-columns: 100px 100px 100px;
    grid-template-rows: 100px 100px 100px;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(3, 1fr);
    column-gap: 20px;
    row-gap: 20px;
```

```
    padding: 15px;
}
.item{
    overflow: hidden;
    border-radius: 5px;

}
.item1{
    grid-column: 1/3;
}
.item3{
  grid-row: 2/4;
}
.item5{
    grid-row: 2/4;
}
.item4{
    grid-row: 2/3;
}
.item:hover{
    transform: scale(1.05);
}
```

Output

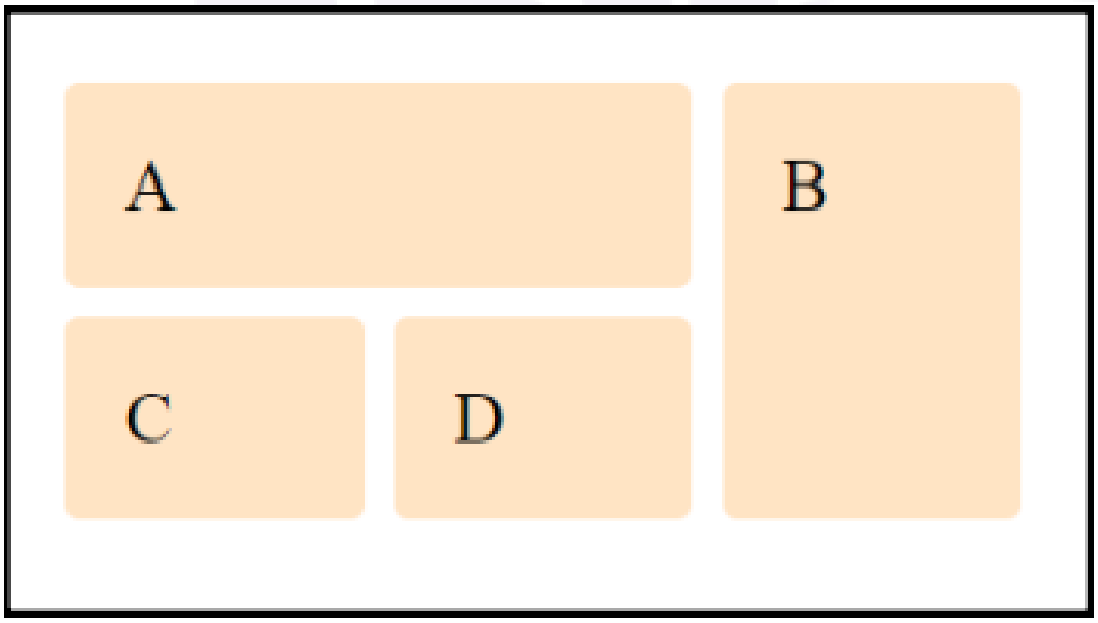


Task 2:

Problem Statement:

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

Expected Output



Solution:

Code(HTML)

```
 <!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
   <title>Grid2</title>
   <link rel="stylesheet" href="grid2.css">
</head>
<body>
   <h1>Arrange container using grid </h1>
   <div class="container">
      <div class="item item1">A</div>
      <div class="item item2">B</div>
      <div class="item item3">C</div>
      <div class="item">D</div>
   </div>
</body>
</html>
```

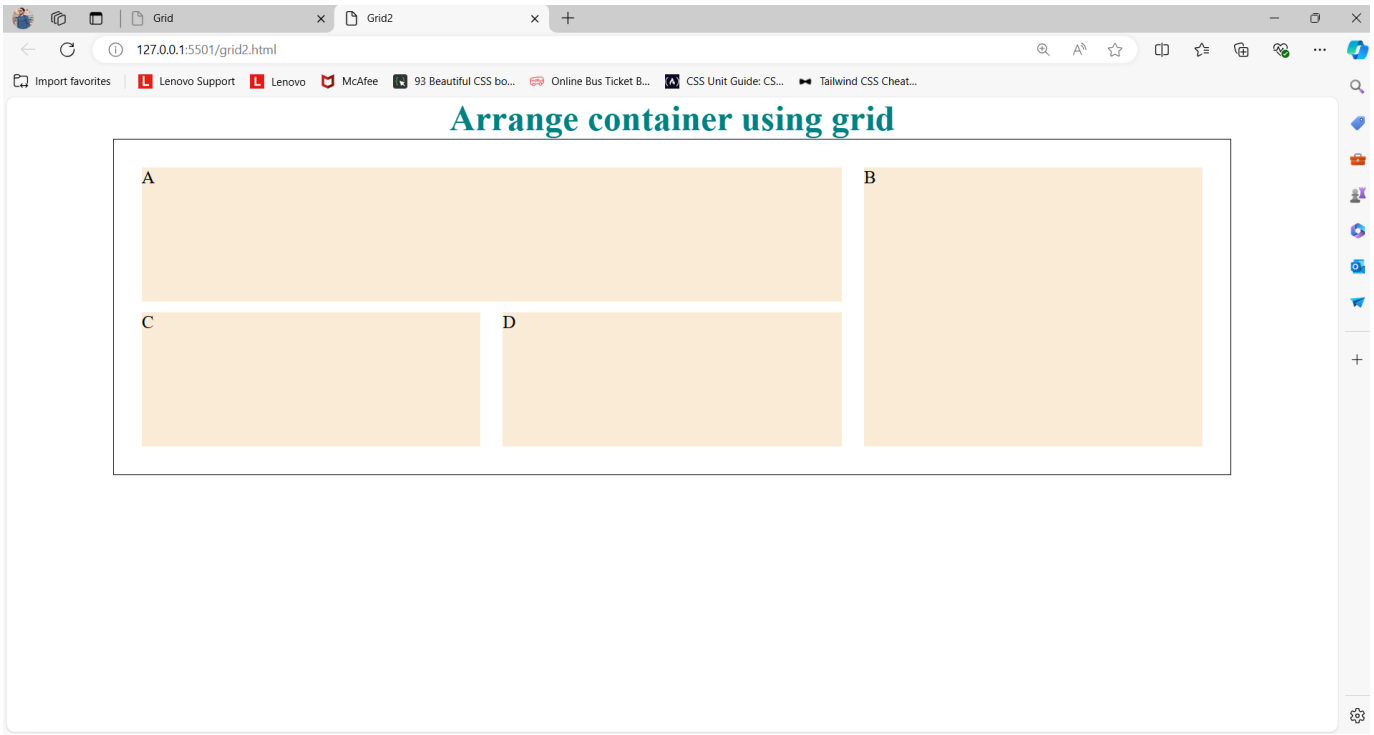
Code(CSS)

```
*{
   margin: 0;
   padding: 0;
```

```
    box-sizing: border-box;
}
h1{
    color: teal;
    text-align: center;
}
.container{
    margin: 0 auto;
    padding: 25px;
    width: 1000px;
    height:300px;
    border: 1px solid black;
    display: grid;
    grid-template-columns: 100px 100px 100px;
    grid-template-rows: 100px 100px;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(2, 1fr);
    column-gap: 20px;
    row-gap: 10px;
}
.item{
    background-color: antiquewhite;
}
```

```
.item1{
    grid-column: 1/3;
    grid-row: 1/2;
}
.item2{
    grid-row: 1/3;
}
.item:hover{
    transform: scale(1.05);
}
```

Output



Task 3:

Problem Statement:

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

Answer:

grid-auto-row :------ The grid-auto-rows property sets a size for the rows in a grid container. This property affects only rows with the size not set.

The different values of the grid-auto-rows property:---- auto, max-content, min-content, value etc.

Example:

Code(HTML)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Grid</title>
    <link rel="stylesheet" href="grid4.css">
</head>
<body>
    <div id="grid">
        <div id="item1"></div>
        <div id="item2"></div>
        <div id="item3"></div>
     </div>
</body>
```

```
</html>
```

Code(CSS)

```
#grid {
   width: 200px;
   display: grid;
   grid-template-areas: "1 1";
   gap: 10px;
   grid-auto-rows: 150px;
   /* grid-auto-rows: fit-content; */
 }

 #grid > div {
   background-color: lime;
 }
```

grid-auto-columns :------ The grid-auto-columns property sets a size for the columns in a grid container. This property affects only columns with the size not set.

The different values of the grid-auto-columns property:---- auto, max-content, min-content, value etc.

Example:

Code(HTML)

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

```
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
   <title>Grid</title>
   <link rel="stylesheet" href="grid4.css">
</head>
<body>
     <div id="grid">
       <div id="item1"></div>
       <div id="item2"></div>
       <div id="item3"></div>
     </div>
</body>
</html>
```

Code(CSS)

```
 #grid {
   height: 100px;
   display: grid;
   grid-template-areas: "a a";
   gap: 10px;
   grid-auto-columns: 200px;
  }
```

```
#grid > div {
  background-color: orange;
}
```

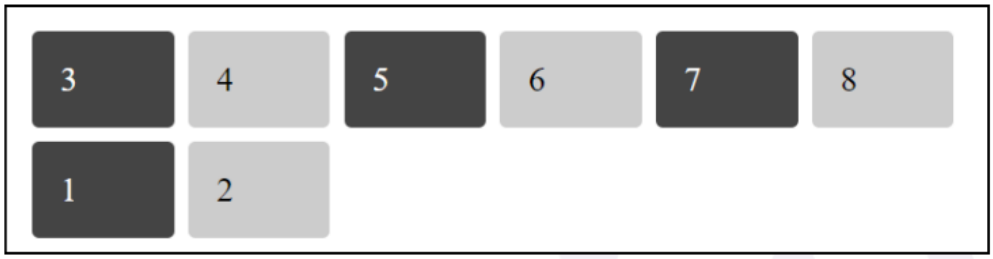
Task 4:

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>
```

Expected Output



Solution:

Code(HTML)

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

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Grid</title>
    <link rel="stylesheet" href="grid3.css">
</head>
<body>
    <h1>Arrange the box using grid</h1>
    <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>
```

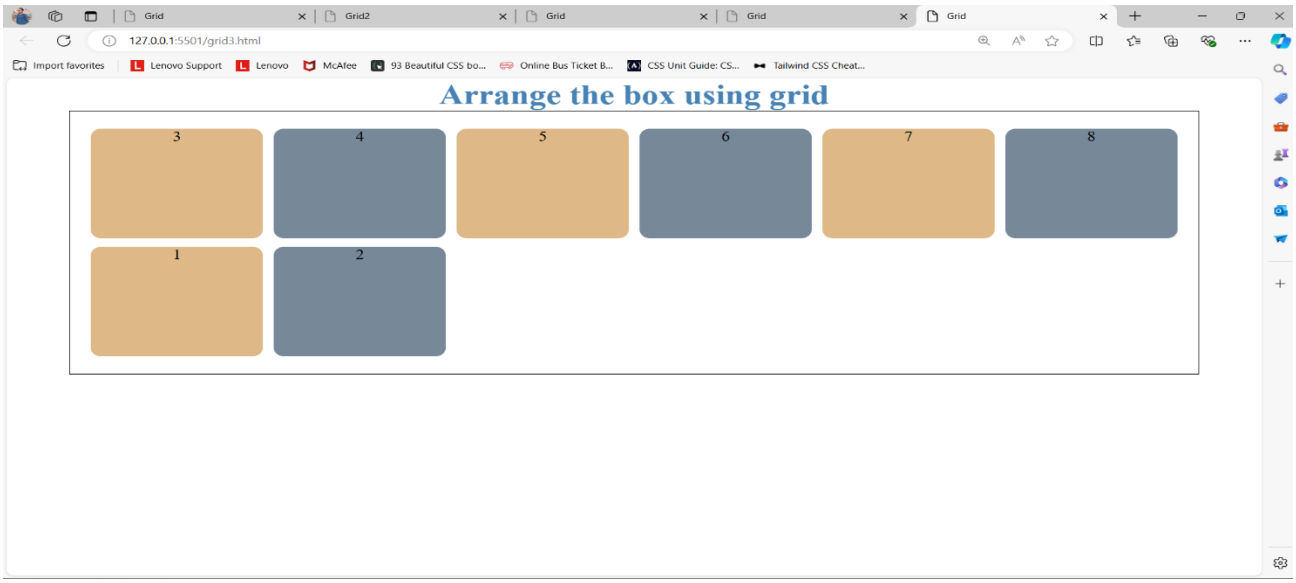
Code(CSS)

```
*{
    margin: 0;
    padding: 0;
```

```
    box-sizing: border-box;
}
h1{
    color: steelblue;
    text-align: center;
}
.container{
    padding: 20px;
    margin: 0 auto;
    border: 1px solid black;
    height: 300px;
    width: 90%;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr 1fr 1fr;
    grid-template-rows: 1fr 1fr;
    gap: 10px;
}
.box{
    border-radius: 10px;
    text-align: center;
}
.box1,.box3,.box5,.box7{
    background-color:burlywood;
}
```

```
.box2,.box4,.box6,.box8{
    background-color: lightslategray;
}
.box1{
    grid-column: 1/2;
    grid-row: 2/3;
}
.box2{
    grid-column: 2/3;
    grid-row: 2/3;
}
```

Output



Task 5:

Problem Statement:

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

Answer:----

The justify-items is used on a grid container and is used to determine how grid items are spread out along a row by setting the default justify-self property for all child boxes.

And

The justify-self is used to set how an individual grid item positions itself along the row/inline axis. Grid items inherit the value of the justify-items property on the container by default, so if the justify-self value is set, it would override the inherited justify-items value.

Example:

Code(HTML)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Grid</title>
    <link rel="stylesheet" href="grid5.css">
</head>
<body>
    <div class="container">
  <span>First child</span>
```

```
  <span>Second child</span>
  <span>Third child</span>
  <span>Fourth child</span>
</div>
</body>
</html>
```

Code(CSS)

```
html {
    font-family: helvetica, arial, sans-serif;
    letter-spacing: 1px;
  }

  .container{
    background-color: red;
    display: grid;
    grid-template-columns: 1fr 1fr;
    grid-auto-rows: 40px;
    grid-gap: 10px;
    margin: 20px;
    width: 300px;
    justify-items: center;
  }
  .container span {
```

```
  background-color: black;
  color: white;
  margin: 1px;
  text-align: center;
 }

 .container,
 span {
  padding: 10px;
  border-radius: 7px;
 }
```

CSS for Justify-self

```
html {
  font-family: helvetica, arial, sans-serif;
  letter-spacing: 1px;
 }

 .container {
  background-color: red;
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-auto-rows: 40px;
  grid-gap: 10px;
```

```
  margin: 20px;
  width: 300px;
  justify-items: stretch;
}

span:nth-child(2) {
  justify-self: start;
}

span:nth-child(3) {
  justify-self: center;
}

span:nth-child(4) {
  justify-self: end;
}

.container span {
  background-color: black;
  color: white;
  margin: 1px;
  text-align: center;
}
```

```
.container,
span {
  padding: 10px;
  border-radius: 7px;}
```