Institute of Computer Technology
B. Tech Computer Science and Engineering

Sub: (2CSE410) FRONT END TECHNOLOGIES

NAME: SUJAL SUTHAR

SEM: CSE 3-B (BATCH 44)

ER NO. : 23162581026

Practical -2

AIM: To understand the CSS grid property and using it create a calculator

Tools Used: Code editor (VS code) and web browser (google chrome).

Theory:
CSS Grid is an advance property which help us to create 2D structure. The following are some of the associated properties of the parent and child items.

1. Parent Properties
2. Child Properties

Parent properties are as follows:

- Display: this property declares the system that 2D structure is required.
- Grid-template-rows: it defines the height of each row in the structure.
- Grid-template-columns: it defines the width of each column in the structure.
- Gap: it can have single or double value. Single value represents same gap between the rows and columns. If there are two then first represents row gap and second represents column gap
- Justify-content: it defines the horizontal alignment in the structure
- Align-content: it defines the vertical alignment in the structure

- Justify-items: It defines the horizontal alignment in its own virtual grid
- Align-items: It defines the vertical alignment in its own virtual grid

Child properties are as follows:

- Grid-row-start: it is the starting edge number of the new position for row.
- Grid-row-end: it is the ending edge number of the new position for row
- Grid-column-start: it is the starting edge number of the new position for column
- Grid-column-end: it is the end edge number of the new position for column
- Justify-self: it aligns the specific child in horizontal direction.
- Align-self: it aligns the specific child in vertical direction.

Reference Link:
https://css-tricks.com/snippets/css/complete-guide-grid/

https://drive.google.com/file/d/1qf2y14r37SmPTTVd2sOIoToFSHeGcuhT/view?usp=drive_link

Code :

INDEX.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="gridstyle.css">
</head>
<body>
    <h1>calculator</h1>
  <div class="calculator">
    <div class="screen">0</div>
    <div class="buttons">
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>+</div>
```

```
      <div>4</div>
      <div>5</div>
      <div>6</div>
      <div>-</div>
      <div>7</div>
      <div>8</div>
      <div>9</div>
      <div>*</div>
      <div style="background-color: red;">C</div>
      <div>0</div>
      <div>=</div>
      <div>/</div>
    </div>
  </div>
</body>
</html>
```

STYLEGRID.CSS:

```
body {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    background-color: #f4f4f4;
    margin: 0;
}

h1 {
    margin-top: 20px;
}

.calculator {
    display: grid;
    grid-template-rows: 1fr 4fr;
    width: 300px;
    height: 400px;
    background: #aa67b7;
    padding: 10px;
    border-radius: 10px;
```

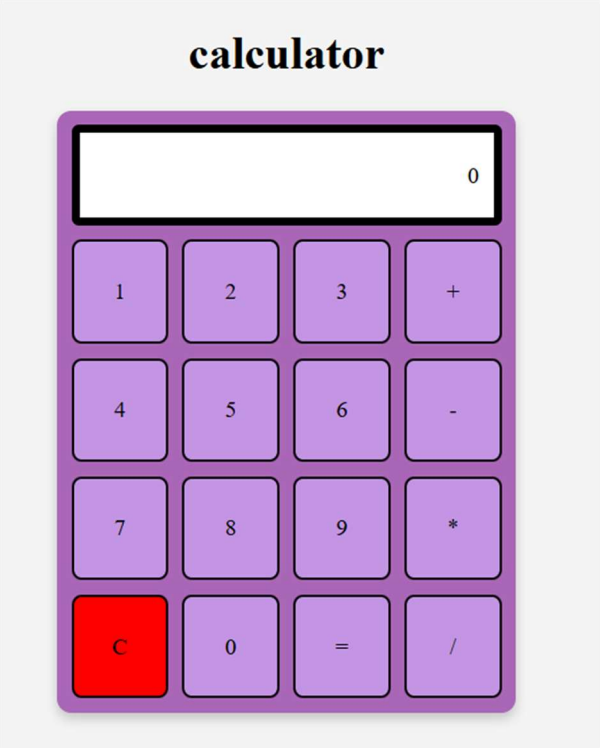
```
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}

.screen {
    background: #ffffff;
    border-radius: 5px;
    display: flex;
    align-items: center;
    justify-content: right;
    padding: 10px;
    border: 6px solid #000;
    margin-bottom: 10px;
}

.buttons {
    grid-row: 2 / 3;
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    gap: 10px;
}

.buttons div {
    border: 2px solid #000;
    background-color: #c495e4;
    display: flex;
    align-items: center;
    justify-content: center;
    border-radius: 10%;
}
```

Output :



Objective Achieved: Through this experiment, the knowledge of CSS grid and its associated properties is gained and the same is implement by creating a calculator. .