SRM AXIS INTELLECTS

# VALUE ADDED COURSE PROJECT

## PROJECT TITLE


STUDENT NAME: SALAI JAYA AAKHAASH KS

REG.NO: RA2211003020315

CLASS&SECTION: CSE ‘E’

GUIDED BY: **S PETCHIMUTHU**, Tech.Trainer, SAI

# PROJECT DESCRIPTION:

In this project, our goal is to create a functional and aesthetically pleasing calculator with a 3D effect using HTML, CSS, and JavaScript. The calculator will have a standard layout with buttons for numbers, mathematical operations, and a display for the results. Our aim is to create a unique user experience that will engage users and make performing mathematical calculations an enjoyable task.

The calculator will be designed to have a 3D effect, with buttons that appear to pop out from the surface of the calculator. When a button is clicked, it will appear to depress slightly, giving the user feedback that the button has been pressed. This effect will add a visual appeal to the calculator, making it more attractive and fun to use.

To create the 3D effect, we will use CSS transformations to rotate and skew the buttons. We will also use box shadows and gradients to create the appearance of depth. The use of CSS transformations will allow us to manipulate the position of the buttons in three dimensions, giving them a realistic and interactive appearance.

The calculator will be functional, allowing users to perform basic mathematical operations such as addition, subtraction, multiplication, and division. It will also have buttons for clearing the display and for entering decimal points. We will ensure that the calculator can handle both simple and complex calculations, making it an ideal tool for both basic and advanced users.

To ensure that the calculator is accessible, we will use appropriate ARIA attributes and keyboard support. This will make it easy for users with disabilities to access and use the calculator. We will also make the calculator responsive, which means

that it will automatically adjust to the size of the device it is viewed on. This feature will ensure that the calculator is usable on a wide range of devices, including desktops, laptops, tablets, and mobile phones.

To implement the calculator, we will use HTML to define the structure of the calculator, CSS to style the calculator, and JavaScript to add interactivity to the calculator. We will use event listeners to detect button clicks and update the display accordingly. The use of JavaScript will make the calculator more dynamic and interactive, providing a more engaging user experience.

In summary, our project involves creating a 3D calculator that is both aesthetically pleasing and functional using HTML, CSS, and JavaScript. The calculator will have a unique appearance that will engage users and make performing mathematical calculations an enjoyable task. We will ensure that the calculator is accessible and responsive, making it usable on a wide range of devices. Our goal is to create a user-friendly and interactive calculator that will provide an exceptional user experience.

# SAMPLE CODING:

## #HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
 <title>CALCULATOR</title>
 <link rel="stylesheet" href="cascade.css">
</head>
<body>
 <div class="calculator">
 <input type="text" id="result" class="output" placeholder="0"></td>
 <div class="keys">
 <input type="button" value="+" class="key key-op" onClick="display('+')"></input>
 <input type="button" value="-" class="key key-op" onClick="display('-')"></input>
 <input type="button" value="&times" class="key key-op" onClick="display('*')"></input>
 <input type="button" value="/" class="key key-op" onClick="display('/')"></input>
 <input type="button" value="7" class="key" onClick="display('7')"></input>
 <input type="button" value="8" class="key" onClick="display('8')"></input>
 <input type="button" value="9" class="key" onClick="display('9')"></input>
 <input type="button" value="4" class="key" onClick="display('4')"></input>
 <input type="button" value="5" class="key" onClick="display('5')"></input>
 <input type="button" value="6" class="key" onClick="display('6')"></input>
 <input type="button" value="1" class="key" onClick="display('1')"></input>
 <input type="button" value="2" class="key" onClick="display('2')"></input>
```

```
<input type="button" value="3" class="key" onClick="display('3')"></input>
<input type="button" value="0" class="key" onClick="display('0')"></input>
<input type="button" value="." class="key" onClick="display('.')"></input>
<input type="button" value="C" class="key" onclick="clearScreen()"></input>
<input type="button" value="=" class="key key-ent" onClick="solve()"></input>
</div>
</div>
<script src="script.js"></script>
</body>
</html>
```

# #CSS

```
* {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
  }
  body {
   background: #f0f1f3;
  }
  .calculator {

   color: #fff;
   padding: 1rem;
   position: absolute;
```

```
 top: 50%;
 left: 50%;
 transform: translate(-50%, -50%);
 background-color: #fdfcfc;
 font-weight: bold;
 box-shadow: 3px 3px 3px #17181c,
 -3px -3px 3px #676e7e;
}
.output {
 font-size: 3.4em;
 text-align: end;
 padding-top: 1.5rem;
 padding-bottom: 1rem;
 padding-right: 1rem;
 margin-bottom: 1.5rem;
 width: 100%;
 border: none;
 display: block;
 outline: none;
 color: #fff;
 background: #404040;
 box-shadow: inset 3px 3px 3px #17181c,
 inset -3px -3px 3px #676e7e;
}
.keys {
```

```
 display: grid;
 grid-template-columns: repeat(4, 1fr);
 grid-gap: 3px;
}
.key {
 background: #e6e6e6;
 border: none;
 font-size: 1.5rem;
 width: 70px;
 height: 70px;
 margin: 0.3rem;
 cursor: pointer;
 color: #fff;
 border-radius: 30px;
 background: #ff9500;
 box-shadow: 3px 3px 3px #2a2a2a,
 -3px -3px 3px #676e7e;
}
.key:active,
.key:focus {
 background: #f8f8f8;
 box-shadow: inset 7px 7px 13px #9d9d9d,
 inset -7px -7px 13px #ffffff;
 background: #3b3f48 !important;
 box-shadow: inset 3px 3px 3px #17181c,
```

```
 inset -3px -3px 3px #676e7e;
}
.key-op {
 background: #ff9500;
}
.key-ent {
 grid-column: 4 / 5;
 grid-row: 2 / span 4;
 height: auto;
 border-radius: 30px;
 background: #ff9500;
}
```

# #JAVASCRIPT

```
function display(val) {
 document.getElementById('result').value += val
 return val
}
function solve() {
 let x = document.getElementById('result').value
 let y = eval(x);
 document.getElementById('result').value = y
 return y
}
function clearScreen() {
```

```
  document.getElementById('result').value = ''
}
```

# SAMPLE OUTPUT: