MAHARASHTRA STATE BOARD
OF TECHNICAL EDUCATION, MUMBAI.

Government Polytechnic, Osmanabad

Microproject Title

“Web based Calculator using HTML, CSS & JavaScript”

Submitted By

Roll No.	Student Name	Enrolment No.
35	Mohite Dipak Balu	1901180310

In Guidance of,
Prof. A. B. Gaikwad

Submitted to

Department of Computer Engineering,
Government Polytechnic, Osmanabad.
Tuljapur road, Osmanabad – 413501, Maharashtra.

MAHARASHTRA STATE BOARD
OF TECHNICAL EDUCATION, MUMBAI.

Government Polytechnic, Osmanabad

CERTIFICATE

This is certified that,

The microproject entitled **Web based calculator using HTML, CSS & JavaScript** Submitted by **Mr. Mohite Dipak Balu** Roll Number **35** of **Fifth** Semester of Diploma in **Computer Engineering (CO)** has completed satisfactory in the course **Client Side Scripting Language (22519)** for **academic year 2021-22** as prescribed in the curriculum.

Date : **/ /**
Place : **Osmanabad**

Enrolment No. : **1901180310**
Exam Seat No. : **206209**

Institute
Seal

Lecturer **Head of Department** **Principal**

Index

1.Acknowledgement

- I take it is an opportunity to thank to all those who have directly and indirectly inspire directed an assisted me towards successful completion of this prosect report.

- I express my sincere thanks to principal **Prof. S. L. Andhare** sir and Head of Department **Prof. P. J. Bansode** sir for having me allowed to submit this report as part of my academics learning.

- I express my sincere thanks to **Prof. A. B. Gaikwad** sir lecturer in Computer Engineering department of Government Polytechnic of Osmanabad tor encouragement throughout project report and guideline in designing and working out of this project…!

Your Sincerely,
Mohite Dipak Balu
Enroll. No.: 1901180310
Roll-35

2. Rationale:

- Everyone knows and uses this above-mentioned device named as ‘Calculator’. In this microproject we have also created such a real-world calculator using HTML, CSS & JavaScript.
- The overall purpose of this project is to evaluate and analyses the requirement of the customer, design and implement the system, testing the functionality and maintain the software of a Modern Web based Calculator using HTML, CSS & JavaScript consistent with the requirement specification. transformation.
- Anyone who has to perform operations like addition, subtraction, multiplication, division, etc. regularly then they can use this software very easily.

3.Aim/Benefits of the Micro-

- Create a Webpage using HTML to create for Calculator. In this microproject I have created a project as similar to a real Calculator using Cascade Style Sheet Programming. Aim of this microproject is to Perform all the operation performed by a Real Calculator such as Addition, Subtraction, Multiplication, Division, etc.

- Our webpage is able to perform all the operations using Graphical User Interface. The main focus of our microproject is to implement the JavaScript to create events on Calculator that works for Addition, Subtraction, Multiplication, Division operations.

4. Course Outcome Achieved:

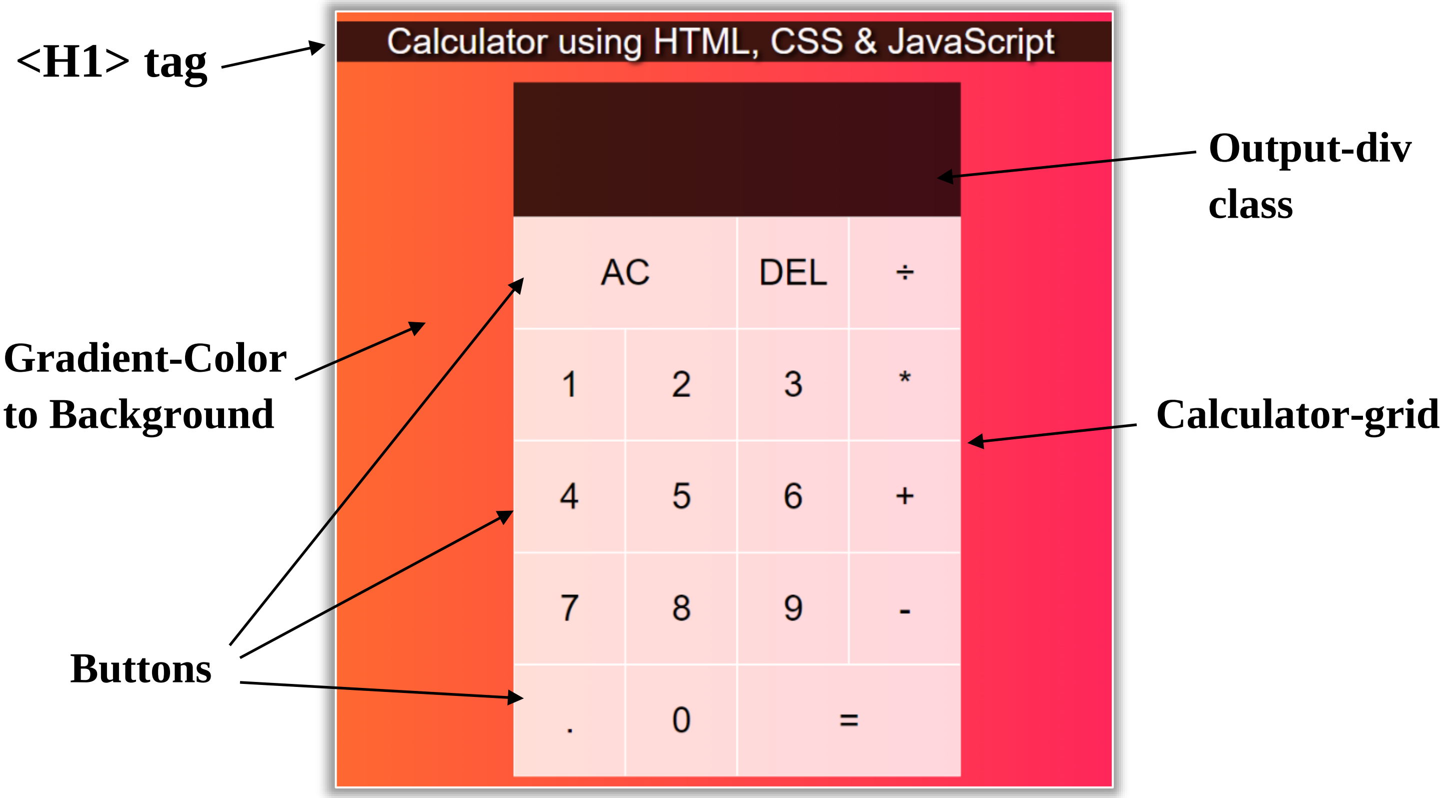
- Creating an interactive web page using program flow control structure.
- Implementing functions in JavaScript.
- Creating event-based web components.
- Implementing switch case statement in JavaScript.
- Using if-else statements in JavaScript.

5. Actual Methodology Followed:

5.1. Designing GUI Model & Event Handling:

Step 1: Create a GUI Design:

1. In this firstly I have created a GUI design of our Calculator using HTML & CSS.
2. Then used JavaScript to add Events and perform operations of calculator.


<H1> tag
Calculator using HTML, CSS & JavaScript
Output-div class
AC
DEL
÷
Gradient-Color to Background
1
2
3
*
Calculator-grid
4
5
6
+
7
8
9
-
Buttons
.
0
=

In this diagram I have navigated all the components which are used in this Calculator Webpage the description of this components is mentioned below:

1. **<H1> tag:**

   This <h1> tag is used here to give the Title Label on the webpage.

2. **Output Div Class:**

   Here, output div class is used to display the result of different operations.

3. **Buttons:**

   Here, Different buttons are used to give different inputs such as numbers and different arithmetic operation's symols.

4. **Gradient-Color:**

   Used to make our webpage more attractive.

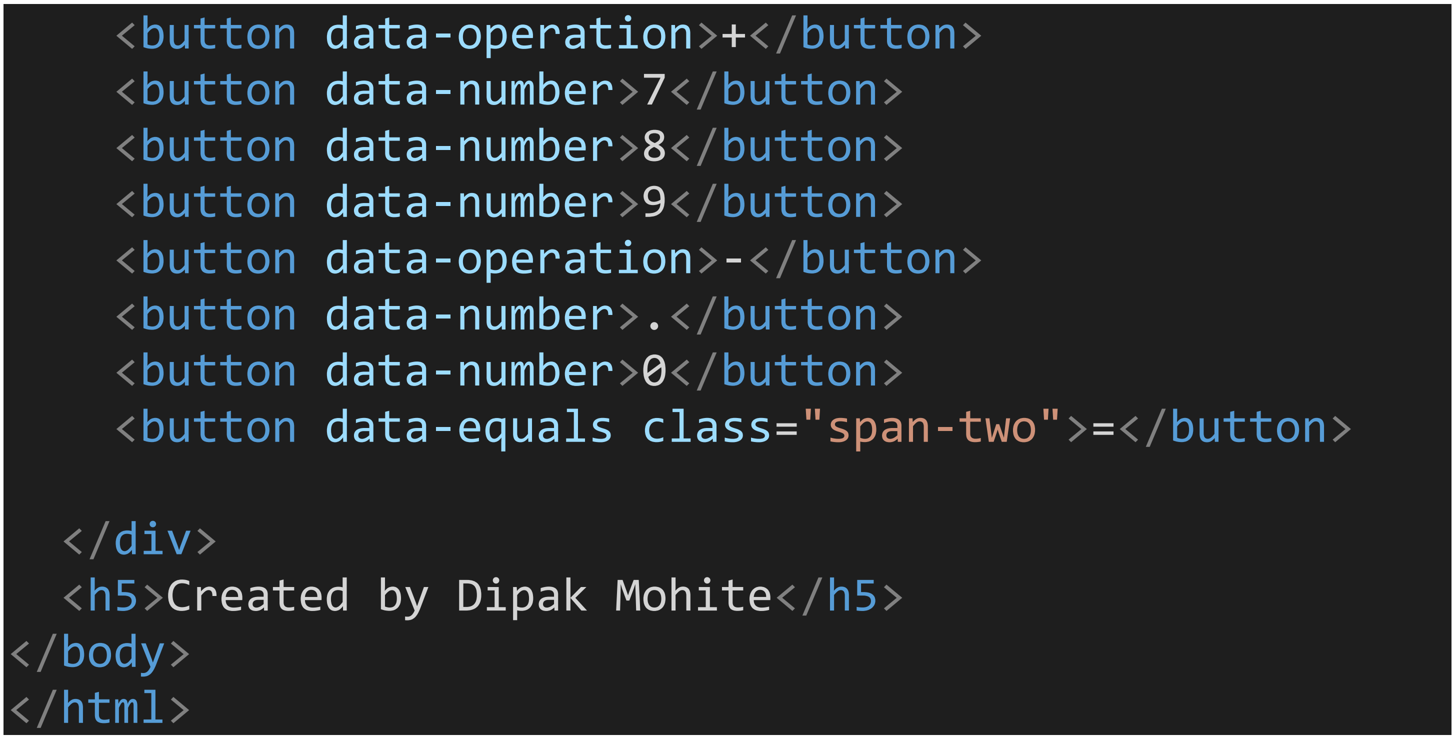
Step 2: Creating Events on used buttons:

1. Here we have used JavaScript to call different functions
2. Every function is created with the intent of performing an arithmetic operation.
3. Every function in JavaScript will call by onclick event on button.
4. After occurring event it will call the JavaScript function.

5.2. Program Code:

5.2.1. HTML:

```html
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Calculator using JavaScript</title>
  <link href="styles.css" rel="stylesheet">
  <script src="script.js" defer></script>
</head>
<body>
  <h1>Calculator using HTML, CSS & JavaScript</h1><br>
  <div class="calculator-grid">
    <div class="output">
      <div data-previous-operand class="previous-operand"></div>
      <div data-current-operand class="current-operand"></div>
    </div>
    <button data-all-clear class="span-two">AC</button>
    <button data-delete>DEL</button>
    <button data-operation>÷</button>
    <button data-number>1</button>
    <button data-number>2</button>
    <button data-number>3</button>
    <button data-operation>*</button>
    <button data-number>4</button>
    <button data-number>5</button>
    <button data-number>6</button>
```



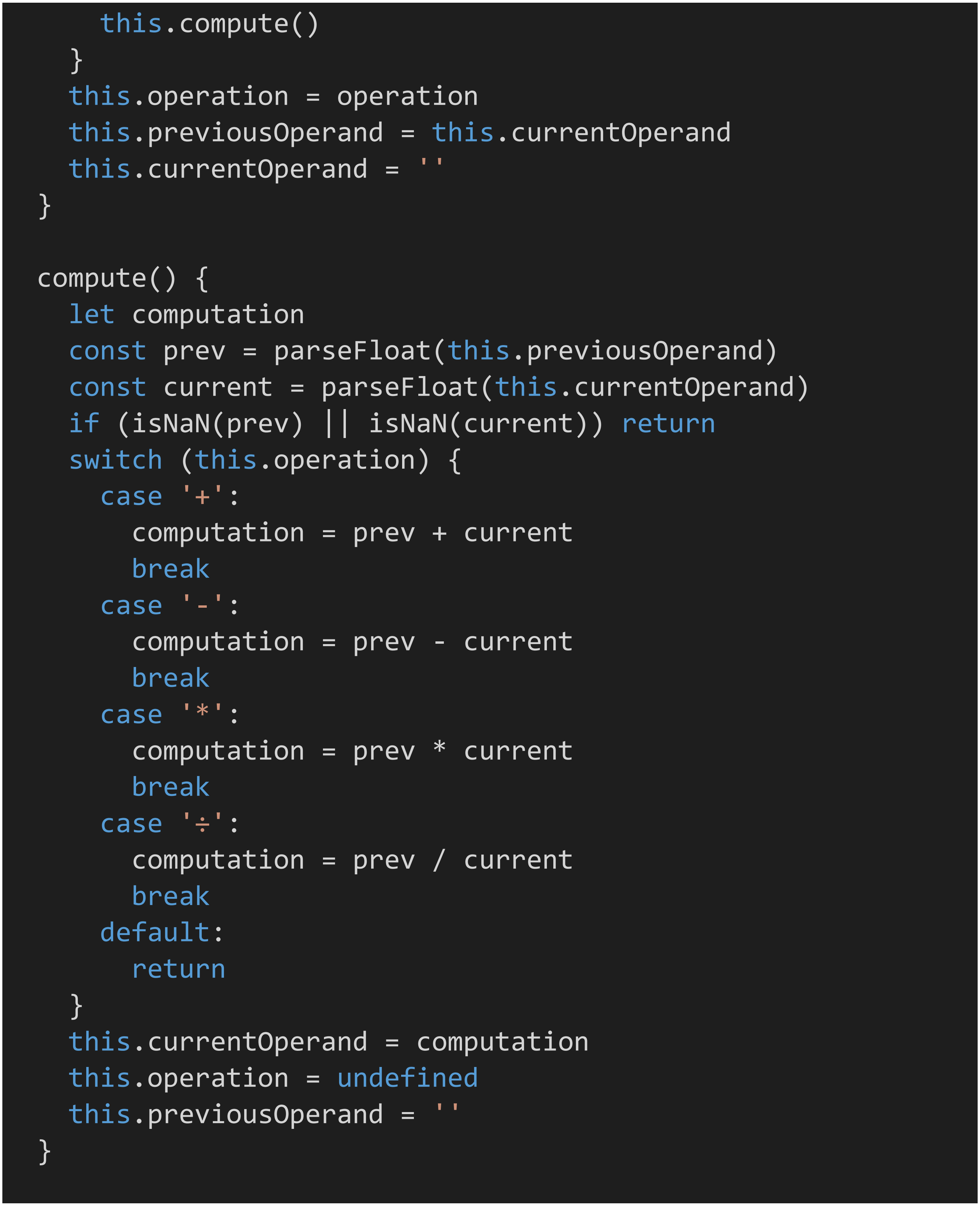
```
    <button data-operation>+</button>
    <button data-number>7</button>
    <button data-number>8</button>
    <button data-number>9</button>
    <button data-operation>-</button>
    <button data-number>.</button>
    <button data-number>0</button>
    <button data-equals class="span-two">=</button>

  </div>
  <h5>Created by Dipak Mohite</h5>
</body>
</html>
```

5.2.2 CSS:

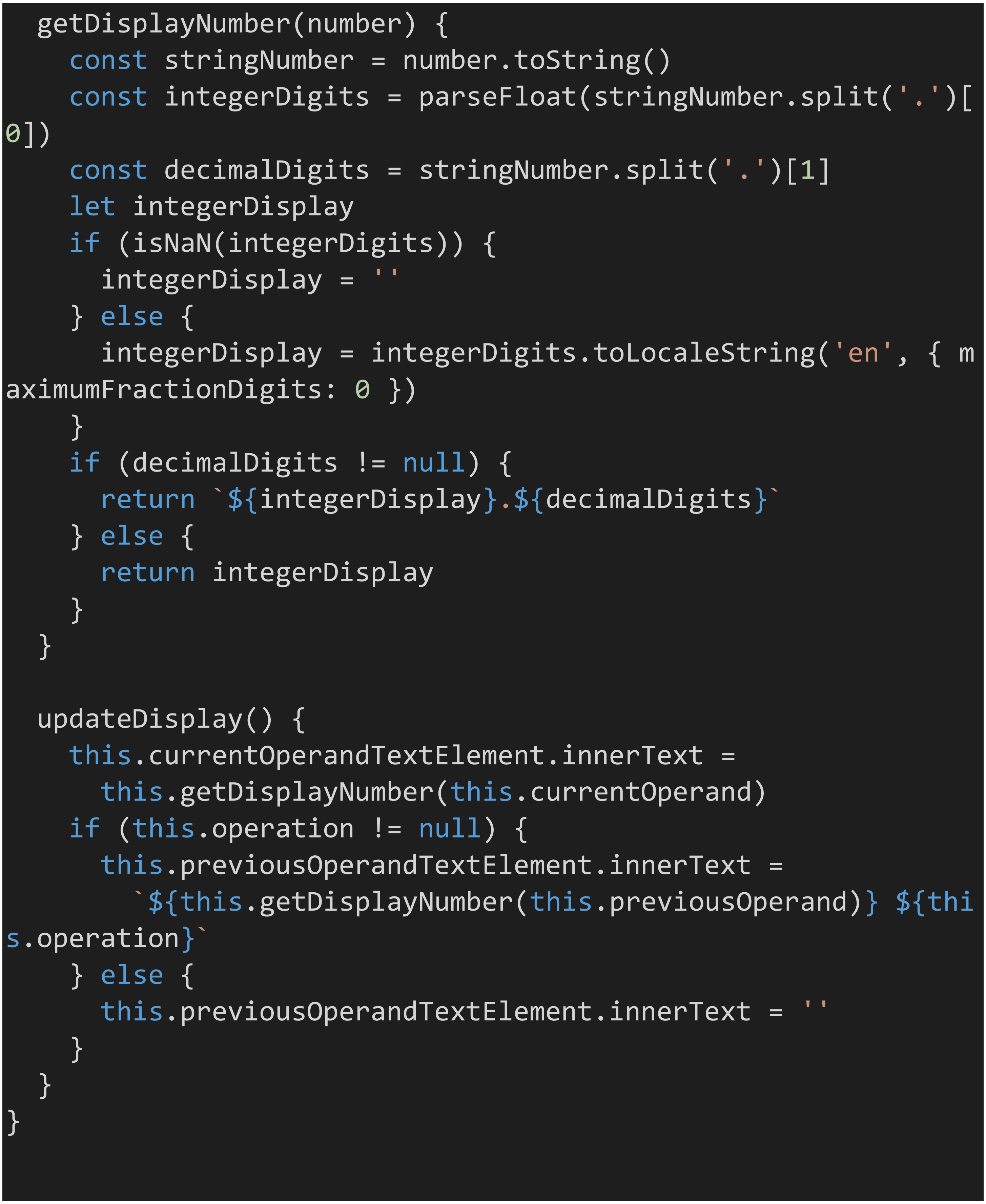
```css
*, *::before, *::after {
  box-sizing: border-box;
  font-family: Gotham Rounded, sans-serif;
  font-weight: normal;
}

body {
  padding: 0;
  margin: 0;
  background: linear-gradient(to right, #ff9019, #ff0073);
}
h1 {
  background-color:#401510;
  color: rgb(253, 249, 249);
  text-shadow: 3px 2px 4px rgb(0, 0, 0);
  text-align: center;
}
h5 {
  background-color:#401510;
  color: rgb(253, 249, 249);
  text-shadow: 3px 2px 4px rgb(0, 0, 0);
  text-align: center;
}
.calculator-grid {
  margin-top: -50px;
  display: grid;
  justify-content: center;
  align-content: center;
  height: 95vh;
  margin-left: 30px;
  grid-template-columns: repeat(4, 100px);
  grid-template-rows: minmax(120px, auto) repeat(5, 100px);
}

.calculator-grid > button {
  cursor: pointer;
  font-size: 2rem;
  border: 1px solid white;
  outline: none;
  background-color: rgba(255, 255, 255, .80);
}




.calculator-grid > button:hover {
  background-color: rgba(255, 255, 255, .9);
}

.span-two {
  grid-column: span 2;
}

.output {
  grid-column: 1 / -1;
  background-color: rgba(0, 0, 0, .75);
  display: flex;
  align-items: flex-end;
  justify-content: space-around;
  flex-direction: column;
  padding: 10px;
  word-wrap: break-word;
  word-break: break-all;
}

.output .previous-operand {
  color: rgba(255, 255, 255, .75);
  font-size: 1.5rem;
}

.output .current-operand {
  color: white;
  font-size: 2.5rem;
}
```

5.2.3. JavaScript:

```
class Calculator {
  constructor(previousOperandTextElement, currentOperandTextElement) {
    this.previousOperandTextElement = previousOperandTextElement
    this.currentOperandTextElement = currentOperandTextElement
    this.clear()
  }

  clear() {
    this.currentOperand = ''
    this.previousOperand = ''
    this.operation = undefined
  }

  delete() {
    this.currentOperand = this.currentOperand.toString().slice(0, -1)
  }

  appendNumber(number) {
    if (number === '.' && this.currentOperand.includes('.')) return
    this.currentOperand = this.currentOperand.toString() + number.toString()
  }

  chooseOperation(operation) {
    if (this.currentOperand === '') return
    if (this.previousOperand !== '') {
```



```
      this.compute()
    }
    this.operation = operation
    this.previousOperand = this.currentOperand
    this.currentOperand = ''
  }

  compute() {
    let computation
    const prev = parseFloat(this.previousOperand)
    const current = parseFloat(this.currentOperand)
    if (isNaN(prev) || isNaN(current)) return
    switch (this.operation) {
      case '+':
        computation = prev + current
        break
      case '-':
        computation = prev - current
        break
      case '*':
        computation = prev * current
        break
      case '÷':
        computation = prev / current
        break
      default:
        return
    }
    this.currentOperand = computation
    this.operation = undefined
    this.previousOperand = ''
  }
```

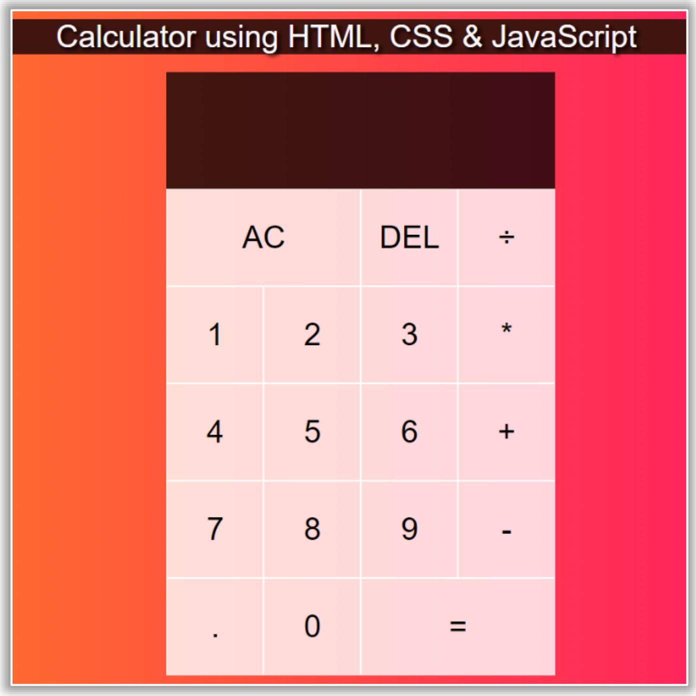


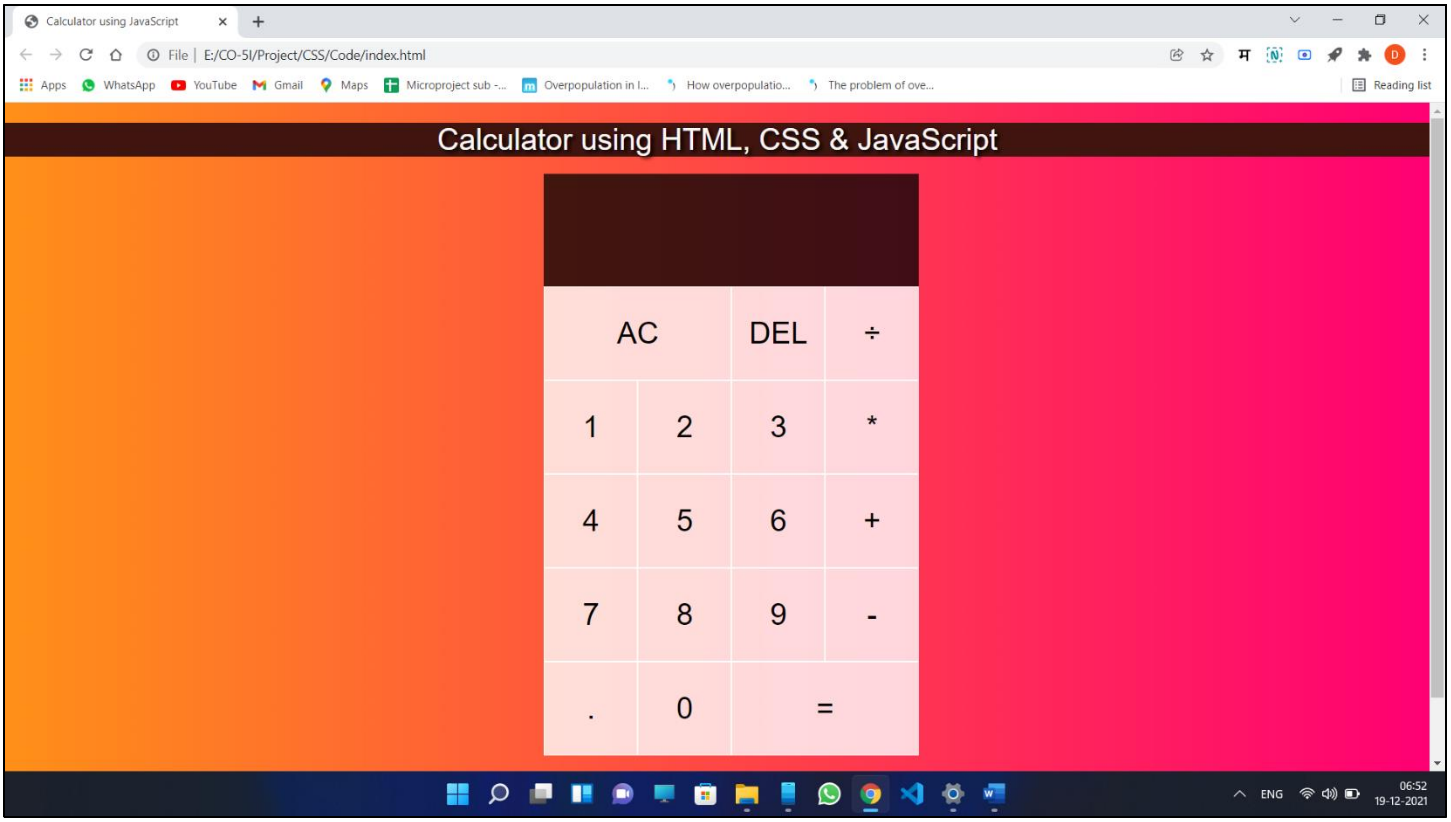
```
  getDisplayNumber(number) {
    const stringNumber = number.toString()
    const integerDigits = parseFloat(stringNumber.split('.')[
0])
    const decimalDigits = stringNumber.split('.')[1]
    let integerDisplay
    if (isNaN(integerDigits)) {
      integerDisplay = ''
    } else {
      integerDisplay = integerDigits.toLocaleString('en', { m
aximumFractionDigits: 0 })
    }
    if (decimalDigits != null) {
      return `${integerDisplay}.${decimalDigits}`
    } else {
      return integerDisplay
    }
  }

  updateDisplay() {
    this.currentOperandTextElement.innerText =
      this.getDisplayNumber(this.currentOperand)
    if (this.operation != null) {
      this.previousOperandTextElement.innerText =
        `${this.getDisplayNumber(this.previousOperand)} ${thi
s.operation}`
    } else {
      this.previousOperandTextElement.innerText = ''
    }
  }
}
```

```javascript
const numberButtons = document.querySelectorAll('[data-number]')
const operationButtons = document.querySelectorAll('[data-operation]')
const equalsButton = document.querySelector('[data-equals]')
const deleteButton = document.querySelector('[data-delete]')
const allClearButton = document.querySelector('[data-all-clear]')
const previousOperandTextElement = document.querySelector('[data-previous-operand]')
const currentOperandTextElement = document.querySelector('[data-current-operand]')

const calculator = new Calculator(previousOperandTextElement, currentOperandTextElement)

numberButtons.forEach(button => {
  button.addEventListener('click', () => {
    calculator.appendNumber(button.innerText)
    calculator.updateDisplay()
  })
})

operationButtons.forEach(button => {
  button.addEventListener('click', () => {
    calculator.chooseOperation(button.innerText)
    calculator.updateDisplay()
  })
})

equalsButton.addEventListener('click', button => {
  calculator.compute()
  calculator.updateDisplay()
})

allClearButton.addEventListener('click', button => {
  calculator.clear()
  calculator.updateDisplay()
})

deleteButton.addEventListener('click', button => {
  calculator.delete()
  calculator.updateDisplay()
})
```

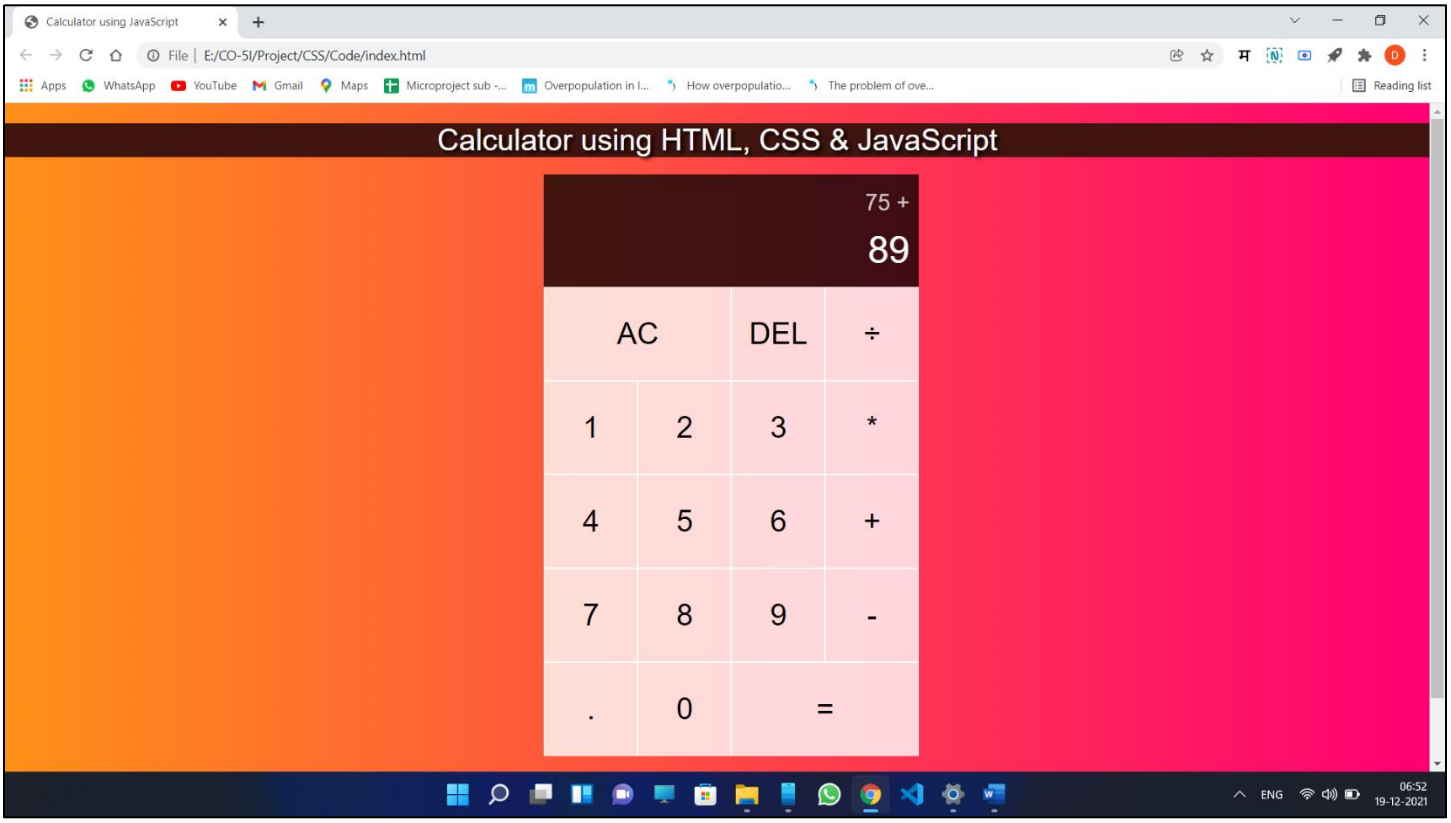
6. Output of Program Code:

6.1. Startup Screen:





6.2. Addition:

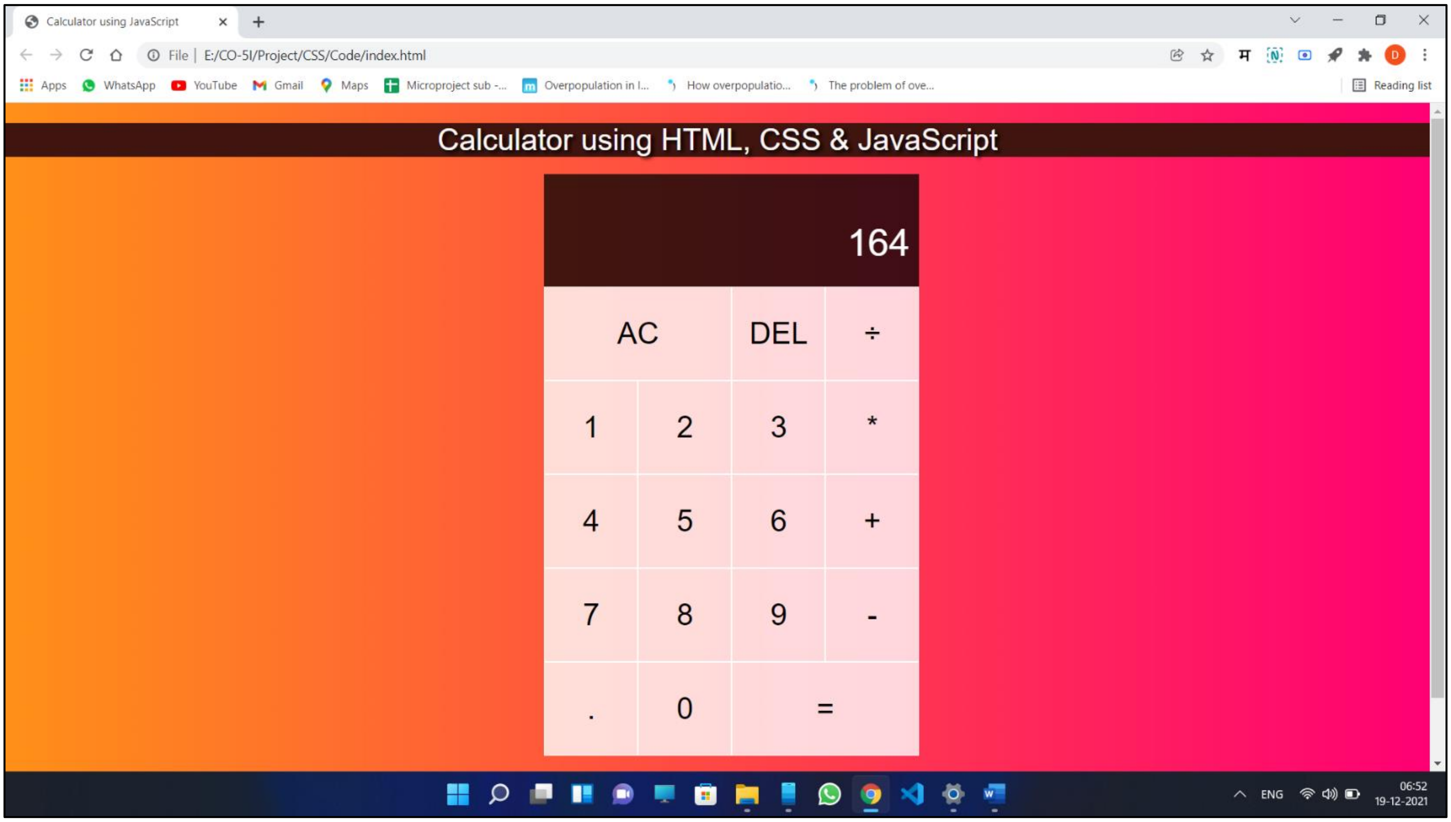


Calculator using HTML, CSS & JavaScript

75 +

89

Calculator using JavaScript
File | E:/CO-5I/Project/CSS/Code/index.html
AC
DEL
÷
1
2
3
*
4
5
6
+
7
8
9
-
.
0
=
06:52
19-12-2021

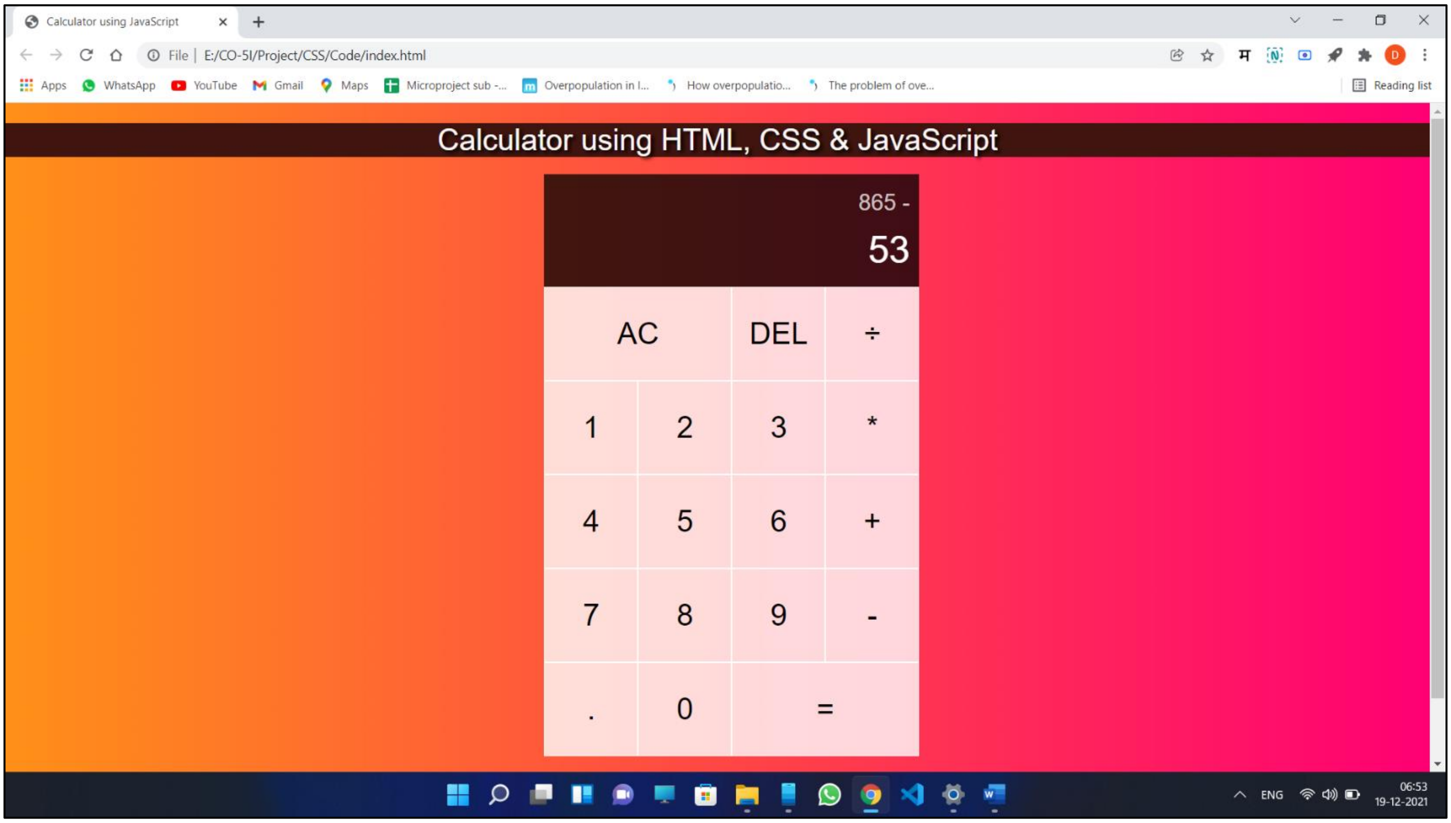


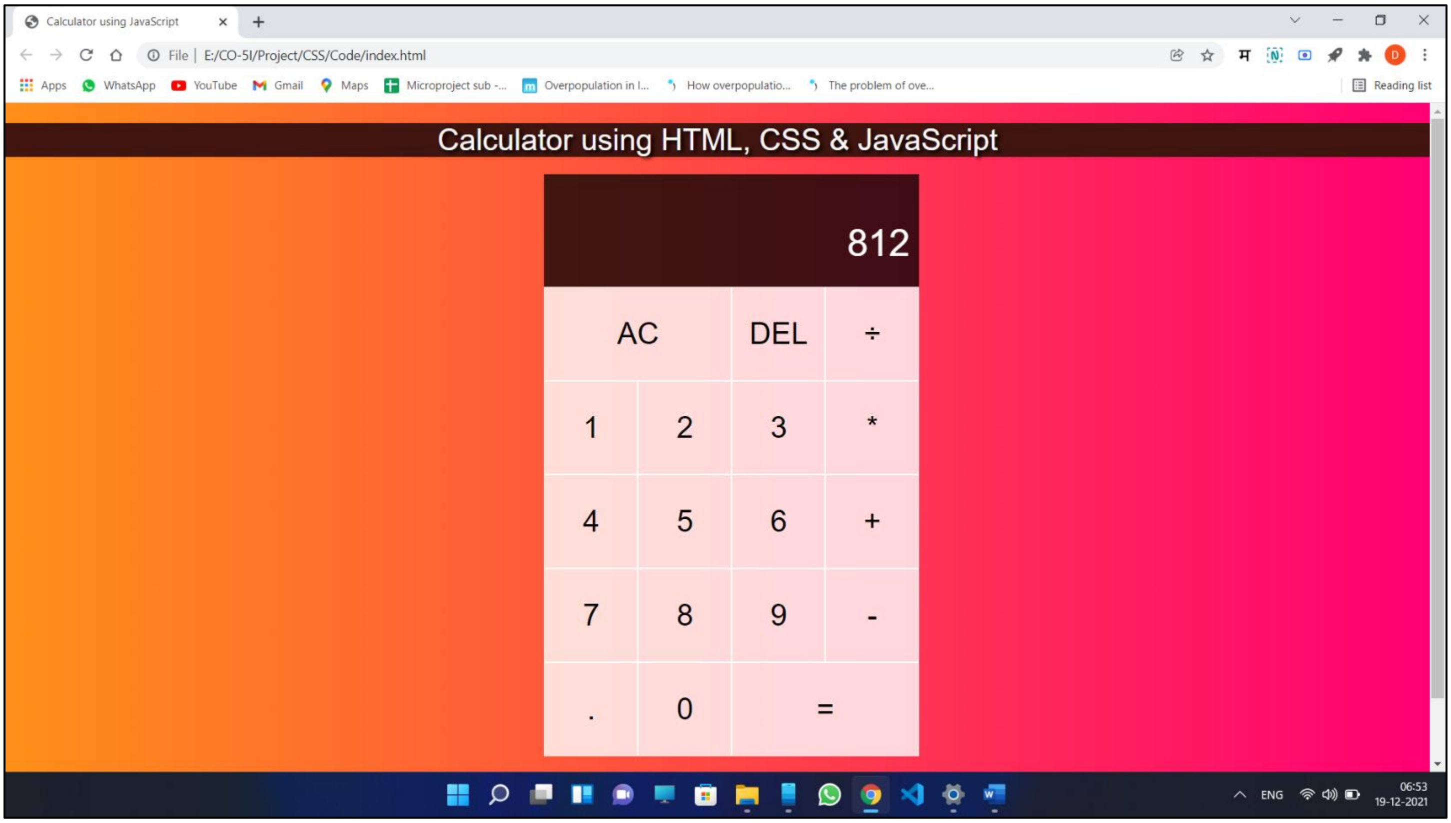
Calculator using HTML, CSS & JavaScript

164

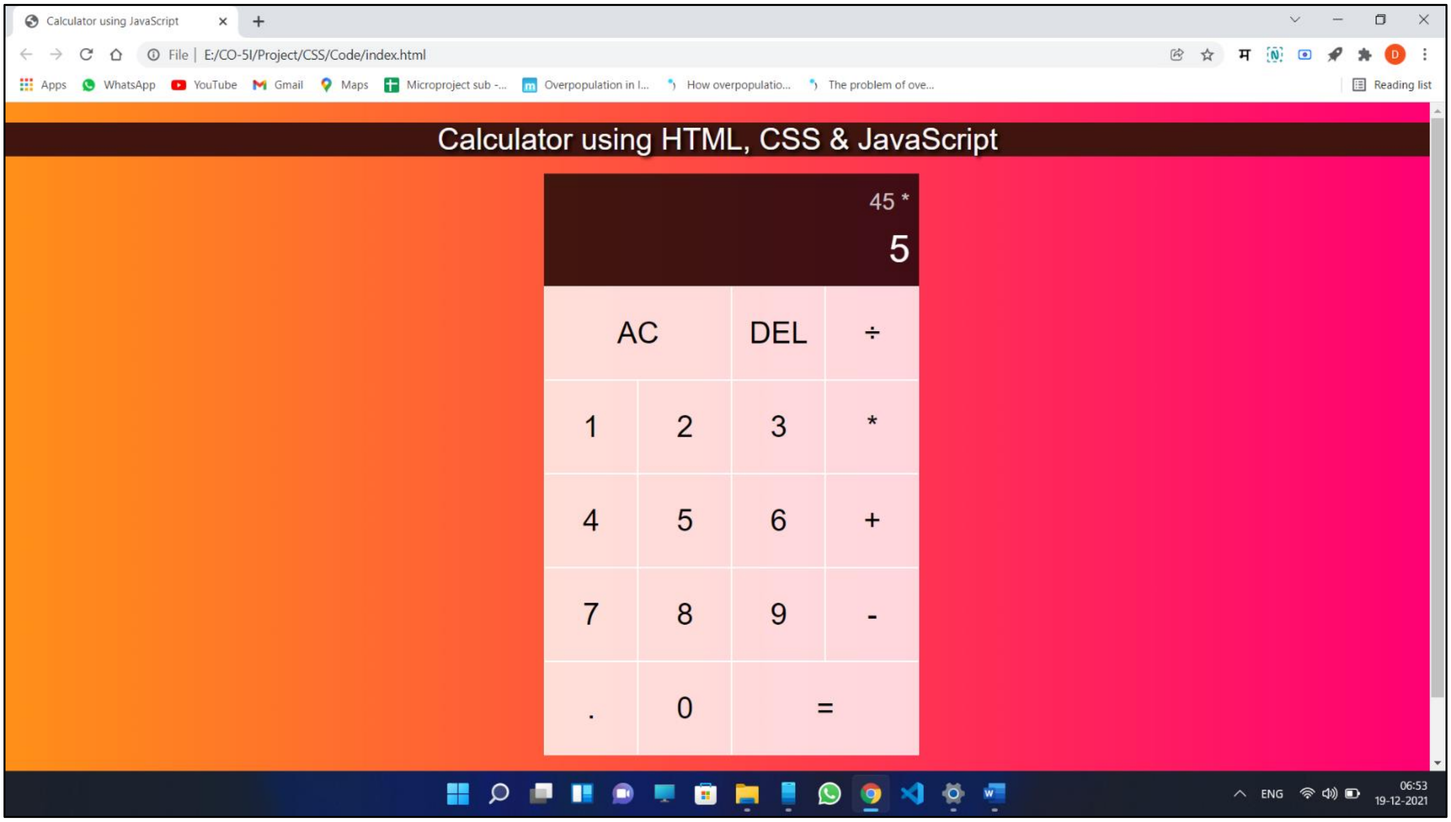
Calculator using JavaScript
File | E:/CO-5I/Project/CSS/Code/index.html
AC
DEL
÷
1
2
3
*
4
5
6
+
7
8
9
-
.
0
=
06:52
19-12-2021

6.3. Subtraction:



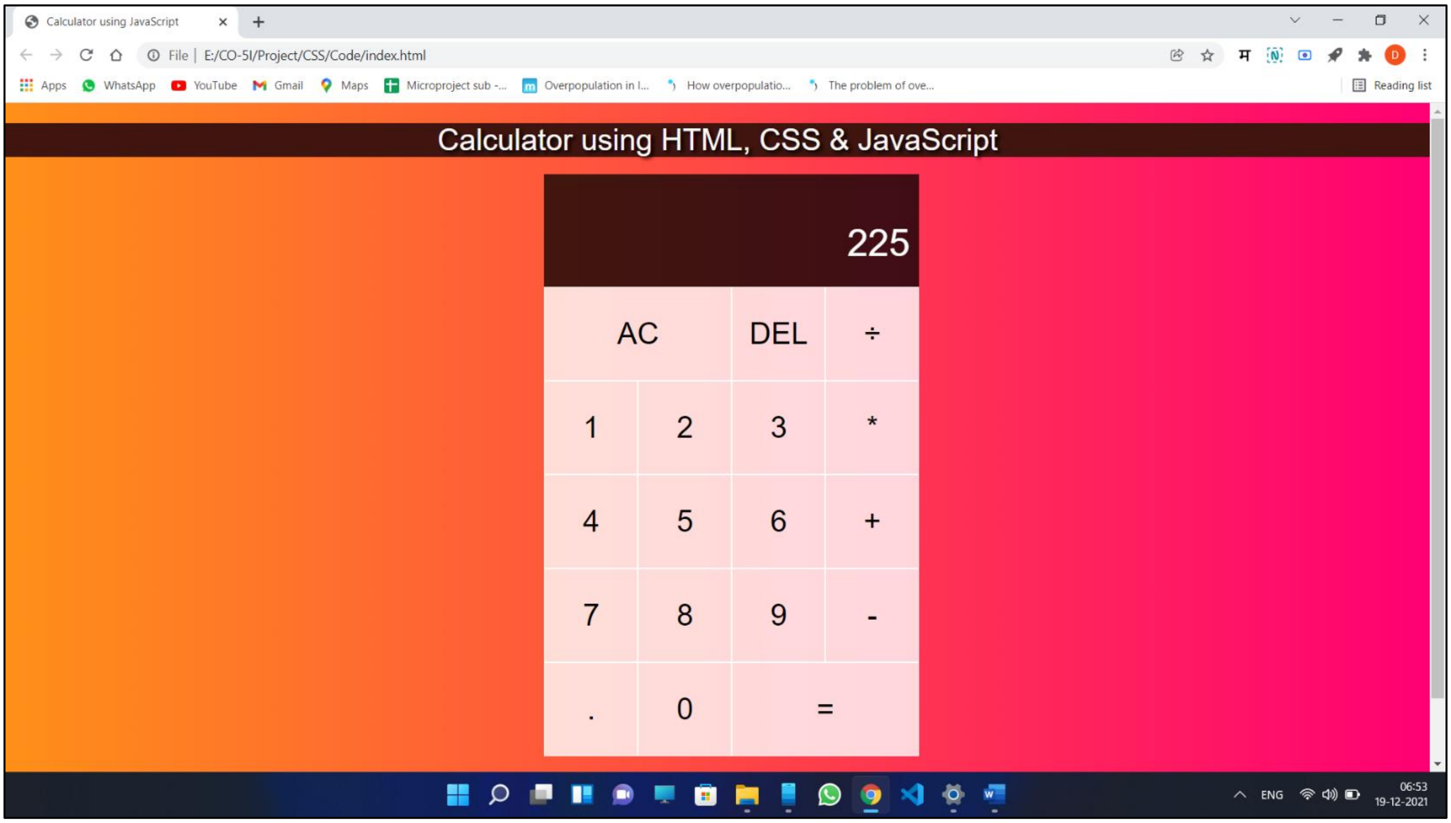


6.4. Multiplication:



Calculator using HTML, CSS & JavaScript

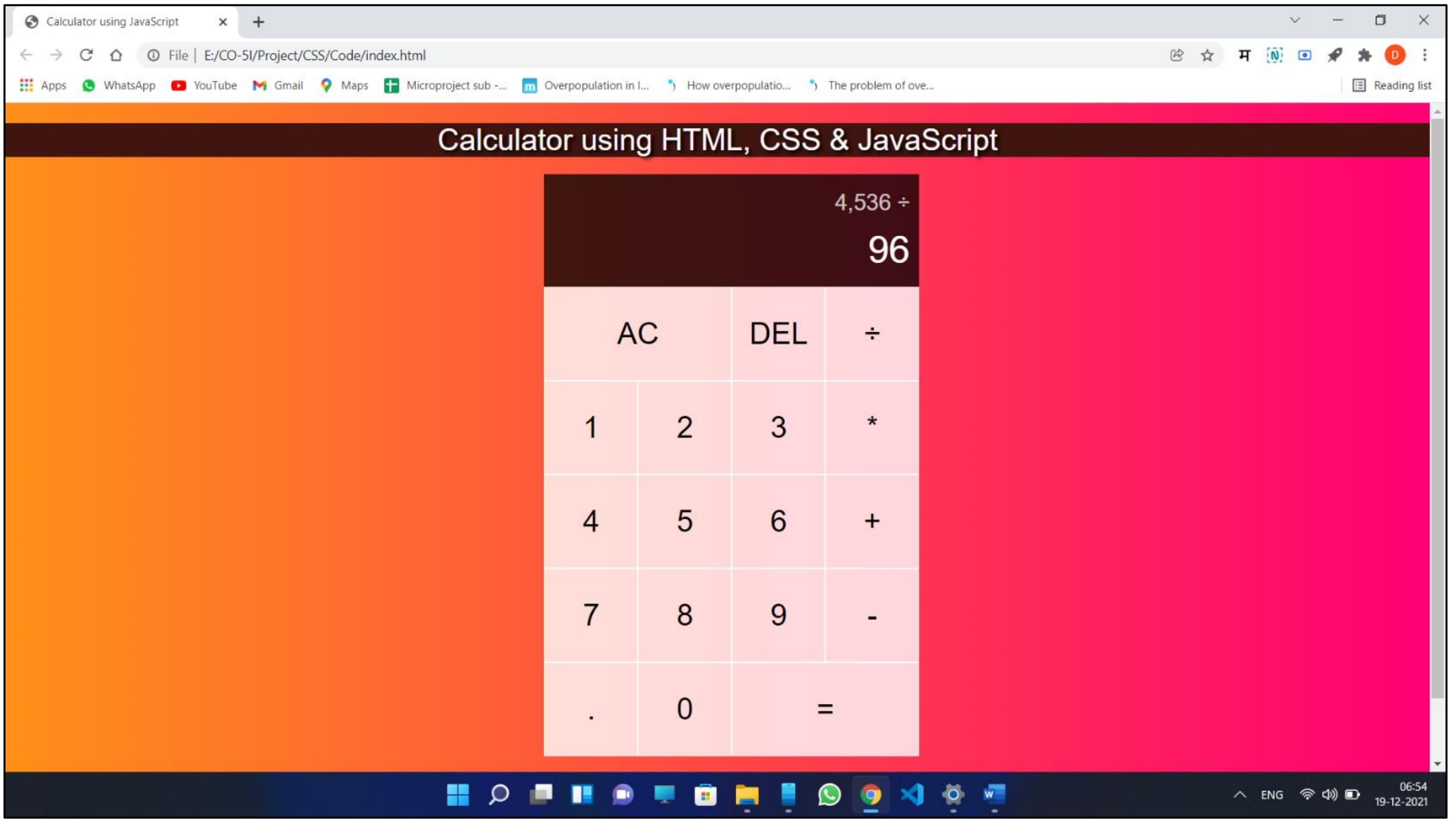
```
45 *
5
```



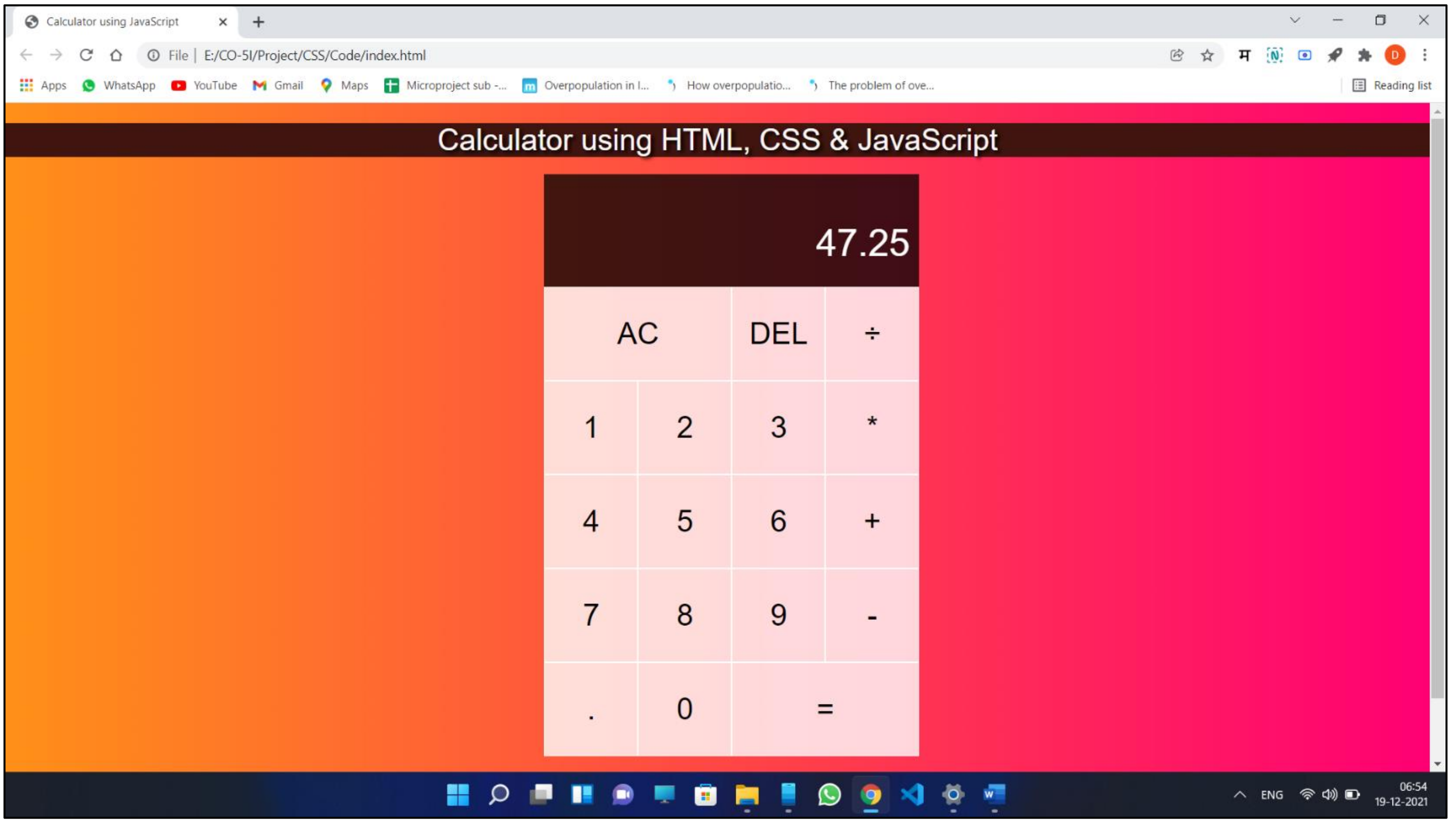
Calculator using HTML, CSS & JavaScript

```
225
```

6.5. Division:

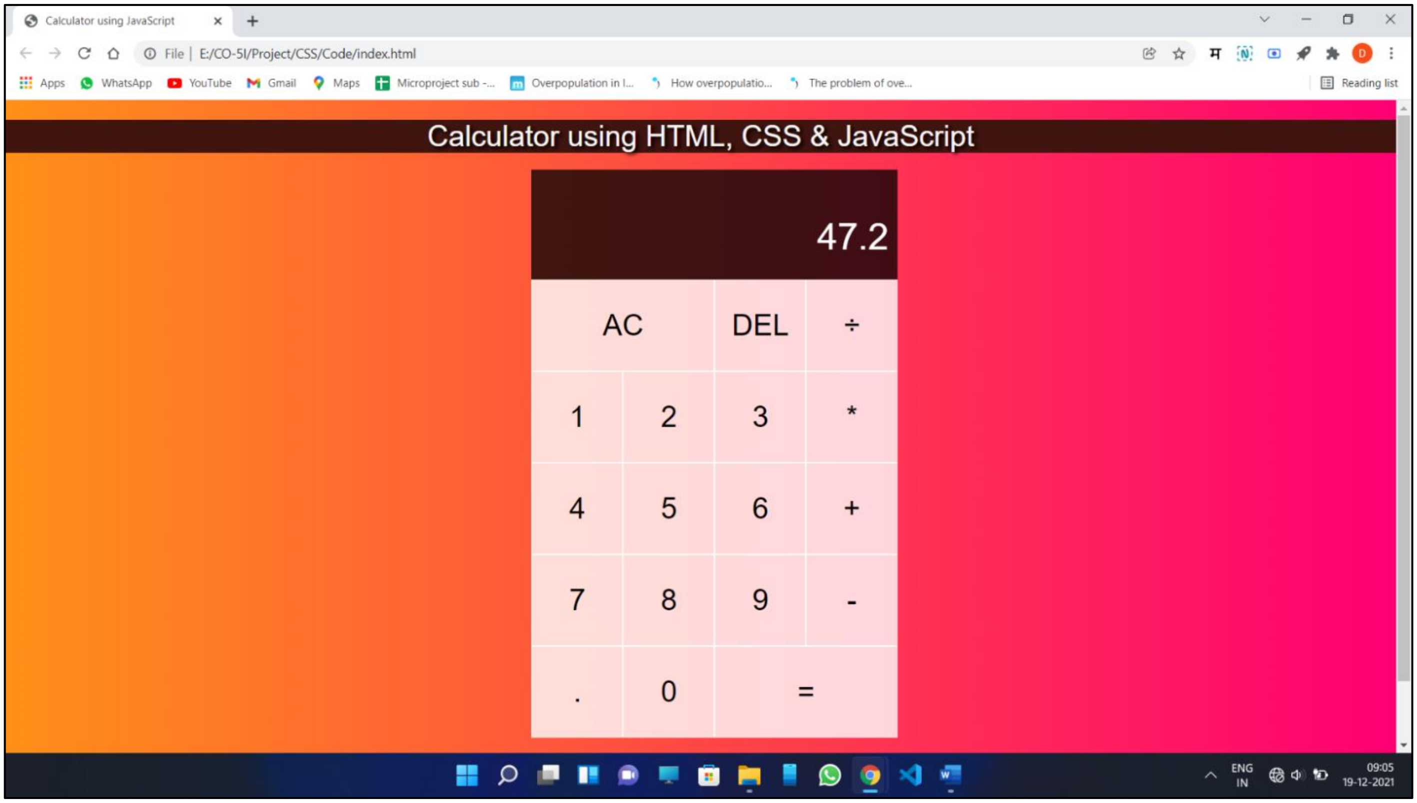


```
4,536 ÷
96
```

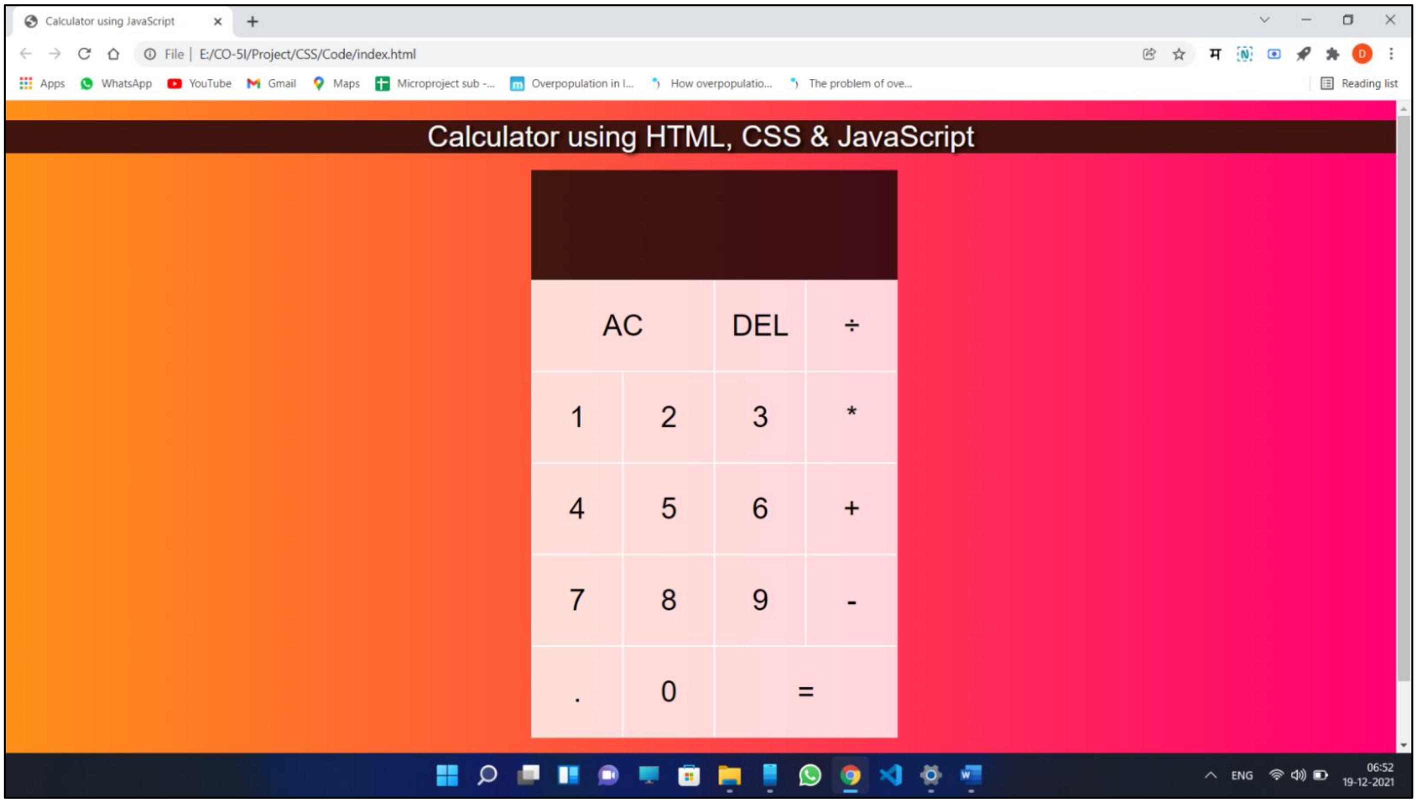


```
47.25
```

6.6. Delete Button:


Calculator using JavaScript
File | E:/CO-5I/Project/CSS/Code/index.html
Apps WhatsApp YouTube Gmail Maps Microproject sub -... Overpopulation in I... How overpopulatio... The problem of ove... Reading list
Calculator using HTML, CSS & JavaScript
47.2
AC DEL ÷
1 2 3 *
4 5 6 +
7 8 9 -
. 0 =
ENG IN 09:05 19-12-2021

6.7 All Clear Button:


Calculator using JavaScript
File | E:/CO-5I/Project/CSS/Code/index.html
Apps WhatsApp YouTube Gmail Maps Microproject sub -... Overpopulation in I... How overpopulatio... The problem of ove... Reading list
Calculator using HTML, CSS & JavaScript
AC DEL ÷
1 2 3 *
4 5 6 +
7 8 9 -
. 0 =
ENG 06:52 19-12-2021

7. Actual Resources Used:

Sr. No.	Name of Resource	Specification	Quantity
1.	Computer System	Intel Core-i3 10th gen with 8 GB RAM, 512GB SSD	1
2.	Operating System	Windows 11	1
3.	Software	Visual Studio Code (Version 1.62)	1

8. Skill Developed/ Learning Outcome:

i. Learned different input types in HTM.

ii. Studied that how create an Event on Buttons.

iii. Learned implementation of different event listener to call the JS functions.

iv. Learned that how to set the different colors to background of the webpage using Cascade Style Sheet.

v. Learned to link CSS (Cascade Style Sheet) & JS (JavaScript) codes.

9. Advantages of Web-Calculator:

i. When you have working on web browser you can easily use this web-based calculator without opening another software.

ii. Performs all the arithmetic operations.

iii. Don’t need to open other software than browser.

iv. Easy to use interface.

v. Simple to handle any user.

vi. No need to study user manual to use this simple web-based calculator.

10. Disadvantages of Calculator:

i. You have to open this calculator webpage for every time when you have to use the calculator.

ii. For opening website, you must have to open any one of the web browsers every time.

iii. This Web based Calculator is time consuming as compare to a Software based Calculator.

11. Application of this Micro-project:

i. The web calculator is used in different insurance website.

ii. Uses in different online calculator websites.

iii. Can be for banking websites.

iv. Web calculator has application of performing different arithmetic operations on a webpage

v. You can simple open this web calculator in one tab of your working browser and you can use it when you have to calculation work.

11. Conclusion:

Firstly, I have discussed with our subject teacher for selecting this topic as my microproject. They guided me very well about this microproject and then told me some more information about creation of this microproject. Then I have implemented this microproject with 3 programming languages HTML, CSS, & JavaScript. In which HTML is used for creating a simple Webpage and add components on it, CSS is used for designing purpose and JavaScript is used to Define the behavior of our web-based calculator.

12. References:

1. Books:
    a. Textbook: Client-Side Scripting Language (22519)
    b. JavaScript: JavaScript Programming for Absolute Beginners: Ultimate Guide To JavaScript Coding by William Sullivan [Sullivan & William}
2. Teachers: Prof. A. B. Gaikwad sir
3. Taken the reference of Google:
    a. https://www.w3schools.com/js/js_events.asp
    b. https://www.calculator.net/
    c. https://www.javatpoint.com/javascript-calculator
    d. https://www.online-calculator.com/
    e. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

THANK YOU