BOOTSTRAP AND HTML CALCULATOR

[Create A Calculator Using HTML And Bootstrap]

PROJECT-2 : BOOTSTRAP AND HTML CALCULATOR

Task : Create A Calculator Using HTML And Bootstrap

ABSTRACT :

In a rapidly evolving technological landscape, the necessity for accessible and user-centric solutions for daily activities has become increasingly pronounced. This project responds to this imperative by introducing a sophisticated HTML calculator, engineered with contemporary web technologies such as Bootstrap and jQuery. Functioning as a versatile hub for fundamental arithmetic computations, the calculator is designed to offer users an intuitive and frictionless experience. By harnessing the capabilities of web development, the project endeavors to bolster the accessibility and utility of computational tools across various domains, catering to the diverse needs of modern users.

In today's dynamic and interconnected world, the demand for tools that streamline routine tasks while accommodating diverse user preferences has never been greater. This project represents a proactive effort to meet this demand head-on by crafting a robust HTML calculator equipped with cutting-edge web technologies like Bootstrap and jQuery. With a primary focus on user experience, the calculator serves as a comprehensive platform for executing essential arithmetic operations with ease and efficiency. By embracing the potential of web development, the project not only seeks to elevate the accessibility and convenience of computational tools but also aims to foster innovation and adaptability in the digital landscape, thereby empowering users to navigate their daily tasks with confidence and proficiency.

Moreover, the project's commitment to continuous improvement and innovation underscores its dedication to staying abreast of emerging trends and evolving user needs. With a flexible and scalable architecture, the HTML calculator is poised to accommodate future enhancements and feature expansions, ensuring its relevance and efficacy in an ever-changing technological landscape. By fostering a culture of innovation and adaptability, the project aims to set a precedent for the development of user-centric solutions that transcend conventional boundaries, ultimately reshaping the digital experience for generations to come.

OBJECTIVE :

The primary objective of this project is to develop an advanced HTML calculator that surpasses basic arithmetic functionalities, offering users a comprehensive and versatile tool for various computational needs. By harnessing the capabilities of Bootstrap and jQuery, the aim is to create a visually appealing and highly responsive calculator that ensures seamless operation across different devices and screen sizes.

Beyond traditional arithmetic calculations, the project seeks to explore avenues for expanding the calculator's functionality. This includes investigating the integration of advanced mathematical operations, such as scientific functions and complex equations, to cater to users with specialized calculation requirements. Additionally, there is an intention to explore incorporating financial calculation capabilities, such as compound interest and amortization, to further enhance the calculator's utility.

Ultimately, the goal is to provide users with a robust and adaptable tool that not only meets their immediate arithmetic needs but also evolves to accommodate more complex calculations across various domains. By prioritizing user experience and accessibility, the project aims to deliver a calculator that is intuitive to use and seamlessly integrates into users' workflows, contributing to increased efficiency and productivity in their daily tasks.

In the pursuit of creating a cutting-edge HTML calculator, this project also emphasizes the importance of user-centric design principles. Beyond just adding features, the project aims to ensure that the calculator's interface remains intuitive and easy to navigate, even as additional functionalities are introduced.

This entails conducting thorough user testing and gathering feedback to iterate on the design continuously. By placing the user at the center of the development process, the project aims to deliver a calculator that not only meets users' computational needs but also exceeds their expectations in terms of usability and convenience.

Moreover, the project envisions the HTML calculator as not just a standalone tool but also as a platform for potential future expansions and integrations. This forward-thinking approach involves designing the calculator with modularity and extensibility in mind, allowing for the seamless incorporation of new features and enhancements over time. Whether it's integrating external APIs for specialized calculations or adding custom modules based on user feedback, the project aims to ensure that the calculator remains adaptable and future-proof, continuing to evolve alongside users' needs and technological advancements.

SIGNIFICANCE :

The significance of this project lies in its contribution to addressing the evolving needs of users in an increasingly digital world. With the widespread integration of technology into various aspects of daily life, there is a growing demand for tools that simplify tasks and enhance productivity. By developing an advanced HTML calculator, this project seeks to fulfill this demand by providing users with a versatile and user-friendly computational tool.

Furthermore, the project's focus on leveraging modern web technologies such as Bootstrap and jQuery underscores the importance of innovation in web development. By showcasing the capabilities of these technologies in creating responsive and visually appealing interfaces, the project not only demonstrates their practical applications but also inspires further exploration and experimentation in web development practices.

Additionally, the exploration of additional functionalities beyond basic arithmetic operations highlights the project's commitment to meeting the diverse needs of users. Whether it's incorporating scientific functions for students and professionals in STEM fields or financial calculations for individuals managing personal finances, the calculator aims to cater to a wide range of users across different domains.

In conclusion, the significance of this project lies in its endeavor to provide users with a powerful yet accessible tool for computational tasks, while also contributing to the advancement of web development practices and the evolution of digital tools in response to changing user needs and technological advancements.

KEY FEATURES :

Basic Arithmetic Operations :

The calculator allows users to perform fundamental arithmetic operations such as addition, subtraction, multiplication, and division. These operations form the backbone of the calculator's functionality, enabling users to perform basic mathematical calculations with ease.

Responsive Design :

The calculator interface is designed to be responsive, meaning it adapts and optimizes its layout and functionality across various devices and screen sizes. Whether accessed on a desktop computer, tablet, or smartphone, the calculator ensures an optimal viewing and interaction experience for users, enhancing accessibility and usability.

User-Friendly Interface :

The interface of the calculator is intuitively designed, prioritizing ease of use and simplicity. Clear display areas prominently showcase input numbers, expressions, and results, ensuring users can easily

track their calculations. Responsive buttons provide tactile feedback and enable effortless input, contributing to a smooth and intuitive user experience.

Dynamic Expression Handling :

One of the standout features of the calculator is its dynamic expression handling capability. As users input numbers and operators, the calculator dynamically updates the expression string in real-time. This allows users to preview their calculations as they enter inputs, facilitating accurate and efficient computation.

Error Handling :

The calculator incorporates robust error handling mechanisms to prevent invalid inputs and provide feedback to users in case of errors. For example, if users attempt to divide by zero or input non-numeric characters, the calculator displays an error message alerting them to the issue. This proactive approach to error handling enhances the reliability and usability of the calculator.

Clear Functionality :

Users can easily clear the calculator display and reset the expression with a single click, thanks to the clear functionality feature. Whether to start a new calculation or correct a mistake, this feature enables users to reset the calculator's state quickly and conveniently, ensuring a seamless user experience.

Modern Styling :

The calculator interface is styled using Bootstrap, a popular front-end framework known for its modern and visually appealing design elements. By leveraging Bootstrap's predefined CSS classes and components, the calculator achieves a sleek and professional appearance that enhances user engagement and satisfaction.

PROJECT COMPONENTS :

This HTML file represents a simple calculator web application. Here are the components of this project:

HTML Structure :

- The HTML structure begins with the “<!DOCTYPE html>” declaration, specifying the document type and the language.
- The “<html>” element contains the entire HTML document.
- Inside the “<head>” element, metadata such as character set, viewport settings, and the title of the page are defined.
- External CSS stylesheets from Bootstrap and custom styles are linked in the “<head>” section.
- The “<body>” element contains the visible content of the webpage.

Calculator Interface :

- The calculator interface is designed using Bootstrap classes for layout and styling.
- It consists of a container with a card layout, which holds the calculator display and buttons.

Display Section :

- The display section includes three " " elements:
  - "inputstring" to display the current input.
  - "expressionstring" to display the current mathematical expression being entered.
  - "valuestring" to display the result of the evaluation.

Calculator Buttons :

- The calculator buttons are arranged in a grid layout.
- Each button is represented by a "<button>" element with a specific data attribute("data-event_key") to identify its function.
- Numeric buttons (0-9) and arithmetic operator buttons (+, -, *, /) are included.
- Additional buttons for decimal point (.), equals (=), clear entry (CE), and delete (DEL) functionalities are provided.

JavaScript Functions :

- JavaScript functions are included within "<script>" tags at the end of the document.
- These functions handle user interactions and perform calculations.
- Event listeners are added to handle button clicks and keyboard inputs.
- Functions like "appendnumber", "generateExpression", "evaluateExpression", and "clearCalc" are defined to manipulate the calculator display and perform arithmetic operations.

Overall, this project provides a basic calculator interface with functionality for inputting numbers and performing arithmetic operations. The JavaScript code enables interaction with the calculator interface and performs the necessary calculations based on user input.

PROJECT STRUCTURE :

The project structure outlines the organization of files and resources within the HTML calculator project. Here's a breakdown of each component:

HTML File (index.html) :

- This file contains the structure of the calculator interface, including display areas, buttons, and input fields.
- It integrates Bootstrap and external CSS files for styling to ensure a modern and visually appealing design.
- The HTML file serves as the entry point for the calculator application and provides the foundation for the user interface.

CSS File (style.css) :

- The style.css file contains custom CSS styles that enhance the visual appearance of the calculator interface.
- It defines styles for buttons, display areas, and other elements to ensure consistency and improve usability.
- Custom CSS rules may be applied to customize the appearance of specific components and achieve the desired design aesthetics.

JavaScript File (script.js) :

- script.js contains the JavaScript code responsible for adding interactivity and functionality to the calculator.
- It handles user inputs, updates the expression string, evaluates expressions, and clears the calculator display based on user interactions.
- JavaScript functions are utilized to implement dynamic features such as real-time expression handling, error detection, and calculation processing.

Bootstrap CSS File (bootstrap.min.css) :

- This file contains the Bootstrap framework CSS, providing predefined styles and components for consistent design across the calculator interface.
- Bootstrap CSS classes are utilized to style HTML elements, ensuring responsiveness and compatibility across different devices and browsers.
- By leveraging Bootstrap's grid system and utility classes, the calculator interface achieves a responsive layout that adapts to various screen sizes.

JQuery Library (jquery.min.js) :

- jquery.min.js contains the jQuery library, which is utilized for DOM manipulation and event handling in the calculator.
- jQuery simplifies JavaScript code by providing methods for traversing and manipulating the HTML document, handling events, and executing asynchronous tasks.
- It facilitates the implementation of interactive features such as button click events, input validation, and dynamic content updates in the calculator interface.

Documentation (README.md) :

- README.md contains documentation detailing the project structure, key features, usage instructions, and any additional information relevant to the project.
- It serves as a guide for developers and users, providing essential information about the calculator application, its functionality, and how to use it effectively.
- The README.md file may include installation instructions, usage examples, troubleshooting tips, and links to additional resources or documentation.

Additional Resources :

- This directory may include additional resources such as fonts, icons, or external libraries used in the project, depending on specific requirements and design choices.
- Resources such as custom fonts, icon sets, or third-party JavaScript libraries may be included to enhance the visual appeal and functionality of the calculator interface.
- Properly organizing and managing additional resources ensures they are readily accessible and seamlessly integrated into the project.

INTRODUCTION :

In today's digital landscape, calculators stand as indispensable tools across various domains, from academia to professional sectors. With the proliferation of web technologies, the realm of calculators has undergone a significant transformation, offering users unparalleled accessibility and adaptability. This project introduces an HTML calculator poised to leverage these advancements, aiming to provide users with a comprehensive solution for their computational needs.

At the core of this project lies a deep appreciation for the value of simplicity and efficiency in everyday tasks. By harnessing the capabilities of HTML, Bootstrap, and jQuery, the objective is to develop a calculator that not only performs basic arithmetic operations seamlessly but also incorporates intuitive user interfaces and responsive design principles. Through meticulous design and implementation, the calculator endeavors to streamline the calculation process, empowering users to perform computations quickly and accurately, regardless of their proficiency level.

Moreover, this project aspires to transcend the boundaries of a conventional calculator by exploring avenues to integrate additional functionalities that cater to diverse user requirements. Whether it involves scientific calculations, unit conversions, or financial computations, the goal is to equip the calculator with features that enhance its utility and versatility. By embracing a user-centric approach, the project aims to address the evolving needs of users in an increasingly dynamic digital environment.

In essence, this endeavor represents a concerted effort to harness the full potential of web technologies in developing practical solutions for everyday tasks. By merging innovation with usability, the HTML calculator seeks to redefine the computational experience, laying the groundwork for enhanced productivity and convenience in the digital age. Through continuous iteration and refinement, the project endeavors to stay at the forefront of technological innovation, offering users an indispensable tool that seamlessly integrates into their workflows and enhances their overall efficiency.

Furthermore, the HTML calculator project recognizes the importance of user experience and accessibility in modern software design. With this in mind, the development process prioritizes the creation of a user-friendly interface that caters to users of all skill levels. Clear display areas, responsive buttons, and dynamic expression handling ensure that users can interact with the calculator effortlessly, without encountering unnecessary barriers or complexities. By fostering an intuitive and seamless user

experience, the calculator aims to empower individuals to focus on their calculations without being hindered by technical intricacies.

Additionally, the project embraces the principles of responsive design to ensure optimal viewing and interaction across a wide range of devices and screen sizes. Whether accessed from a desktop computer, tablet, or smartphone, the calculator adapts fluidly to different display environments, maintaining consistency and usability. This commitment to responsiveness reflects the project's dedication to accessibility, acknowledging the diverse ways in which users engage with technology in today's interconnected world. By embracing responsive design principles, the HTML calculator project seeks to accommodate the needs of users across various platforms, enhancing its accessibility and usability for all.

REQUIREMENT ANALYSIS :

The first step involves a thorough analysis of the requirements and objectives of the HTML calculator. This includes identifying the target audience, defining the core functionalities, and specifying any additional features that may enhance user experience. The analysis also considers factors such as usability, accessibility, and responsiveness across different devices and screen sizes.

Design Planning :

Once the requirements are understood, the design planning phase begins. This entails conceptualizing the layout and structure of the calculator interface, including the placement of buttons, display areas, and other elements. Wireframes or mockups may be created to visualize the design and gather feedback from stakeholders before proceeding to implementation.

HTML Structure :

With the design plan in place, the HTML structure of the calculator is developed. This involves coding the necessary HTML elements to create the user interface components, such as buttons, input fields, and display areas. Semantic HTML tags are utilized to ensure accessibility and optimize search engine visibility.

Bootstrap Integration :

Bootstrap, a popular front-end framework, is integrated into the project to enhance the visual appearance and responsiveness of the calculator. Bootstrap's predefined CSS classes are leveraged to style the HTML elements, providing a consistent and modern look across different browsers and devices. Responsive design principles are applied to ensure that the calculator adapts seamlessly to various screen sizes, from desktops to mobile devices.

JQuery Implementation :

JQuery, a JavaScript library, is employed to add dynamic functionality to the calculator. Event listeners are attached to the calculator buttons to capture user inputs and trigger corresponding actions. For example, when a number button is clicked, the number is appended to the display. Similarly,

operators such as addition, subtraction, multiplication, and division are handled dynamically. JQuery's DOM manipulation capabilities are utilized to update the calculator display in real-time as users interact with the interface.

Expression Handling :

The calculator maintains an expression string to keep track of the user's input and perform calculations. When an operator button is clicked, the expression is updated accordingly. For instance, if the user clicks the addition button after entering a number, the operator is appended to the expression. The expression is evaluated when the user clicks the equals button, and the result is displayed on the calculator screen.

Testing and Debugging :

Throughout the development process, rigorous testing is conducted to ensure the functionality, usability, and compatibility of the calculator. Test cases are devised to validate different scenarios, including arithmetic operations, input validation, and responsiveness across devices. Debugging tools are utilized to identify and resolve any issues or inconsistencies in the code, ensuring a smooth and error-free user experience.

Documentation and Deployment :

Finally, comprehensive documentation is prepared to document the design, implementation, and usage of the HTML calculator. The calculator is deployed to a web server or hosting platform, making it accessible to users worldwide. Continuous monitoring and maintenance may be performed to address any future updates or enhancements.

CODE :

```
<!DOCTYPE html>
<html lang="en">
   <head>
      <meta charset="utf-8">
      <meta name="viewport" content="width=device-width,initial-scale=1">
      <title>HTML CALCULATOR|MINOR PROJECT2</title>
      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ"
crossorigin="anonymous">
      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.10.5/font/bootstrap-icons.min.css">
      <style>
         .calc-btn{
            width:60px;
            height:60px;
         }
         .calc-display{
            background-color: grey;
            color: black;
            height: 100px;
            width: 310px;
            border-radius:5px;
         }
         .inputstring{
            margin-top:5px;
            height:20px;
            display:block;
            font-size:20px;
         }
         .expressionstring{
            margin-top:5px;
            height:20px;
            display:block;
            font-size:20px;
         }
         .valuestring{
            margin-top:5px;
            height:20px;
            display:block;
            font-size:20px;
         }
      </style>
```

```
  </head>
  <body>
    <div class="container p-5">
      <div class="d-flex justify-content-center">
        <div class="col-sm-12 col-md-4 col-lg-4">
          <div class="card">
            <div class="d-flex justify-content-center align-items-center p-3">
              <div class="row">
                <div class="col-sm-12 col-md-12 col-lg-12 col-xl-12 calc-display">
                  <div id="number_div" class ="inputstring">0</div>
                  <div id="expression" class ="expressionstring"></div>
                  <div id="value_div"  class ="valuestring"></div>
                </div>
                <input type="hidden" id="savedExpression">
              </div>
            </div>
            <div class="d-flex justify-content-center align-items-center p-3 p-3">
              <div class="row">
                <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                  <button type="button" class="btn btn-light calc-btn" data-
event_key="numlock"><small>num lock</small></button>
                </div>
                <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                  <button type="button" class="btn btn-light calc-btn"   data-
event_key="CE">CE</button>
                </div>
                <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                  <button type="button" class="btn btn-light calc-btn"  data-
event_key="delete">DEL</button>
                </div>
                <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                  <button type="button" class="btn btn-light calc-btn" data-
event_key="/">/</button>
                </div>
              </div>
            </div>
            <div class="d-flex justify-content-center align-items-center p-3 p-3">
              <div class="row">
                <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                  <button type="button" class="btn btn-light calc-btn"  data-
event_key="7">7</button>
                </div>
                <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                  <button type="button" class="btn btn-light calc-btn"  data-
event_key="8">8</button>
```

```
                    </div>
                    <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                      <button type="button" class="btn btn-light calc-btn" data-
event_key="9">9</button>
                    </div>
                    <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                      <button type="button" class="btn btn-light calc-btn" data-
event_key="*">*</button>
                    </div>
                  </div>
                </div>
                <div class="d-flex justify-content-center align-items-center p-3">
                  <div class="row">
                    <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                      <button type="button" class="btn btn-light calc-btn" data-
event_key="4">4</button>
                    </div>
                    <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                      <button type="button" class="btn btn-light calc-btn" data-
event_key="5">5</button>
                    </div>
                    <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                      <button type="button" class="btn btn-light calc-btn" data-
event_key="6">6</button>
                    </div>
                    <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                      <button type="button" class="btn btn-light calc-btn" data-event_key="-">-
</button>
                    </div>
                  </div>
                </div>
                <div class="d-flex justify-content-center align-items-center p-3">
                  <div class="row">
                    <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                      <button type="button" class="btn btn-light calc-btn" data-
event_key="1">1</button>
                    </div>
                    <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                      <button type="button" class="btn btn-light calc-btn" data-
event_key="2">2</button>
                    </div>
                    <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                      <button type="button" class="btn btn-light calc-btn" data-
event_key="3">3</button>
                    </div>
```

```
                        <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                            <button type="button" class="btn btn-light calc-btn" data-
event_key="+">+</button>
                        </div>
                    </div>
                </div>
                <div class="d-flex justify-content-center align-items-center p-3">
                    <div class="row">
                        <div class="col-sm-6 col-md-6 col-lg-6 col-xl-6">
                            <button type="button" class="btn btn-light calc-btn" style="width:150px;"
data-event_key="0">0</button>
                        </div>
                        <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                            <button type="button" class="btn btn-light calc-btn" data-
event_key=".">.</button>
                        </div>
                        <div class="col-sm-3 col-md-3 col-lg-3 col-xl-3">
                            <button type="button" class="btn btn-light calc-btn" data-
event_key="=">=</button>
                        </div>
                    </div>
                </div>


            </div>
        </div>
    </div>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.0/jquery.min.js"
integrity="sha512-
3gJwYpMe3QewGELv8k/BX9vcqhryRdzRMxVfq6ngyWXwo03GFEzjsUm8Q7RZcHPHksttq7/G
FoxjCVUjkjvPdw==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.9.2/umd/popper.min.js"
integrity="sha512-
2rNj2KJ+D8s1ceNasTIex6z4HWyOnEYLVC3FigGOmyQCZc2eBXKgOxQmo3oKLHyfcj53uz4
QMsRCWNbLd32Q1g==" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe"
crossorigin="anonymous"></script>
    <script>
        $(document).on('keypress',function(e){
            $('button[data-event_key="'+e.key+'"]').addclass('active')
            if(e.keycode >=48 && e.keycode <= 57 || (e.keycode>=96 && e.keycode<=105)){
                appendnumber(e.key)
            }else{
```

```
            if(e.key=='+' || e.key=='-' || e.key=='/' || e.key=='*'){
                generateexpression(e.key)
            }else if(key == '='){
                evaluateExpression()
            }else if(key == 'delete'){
                clearCalc()
            }
        }
        console.log(e.key);
    })

    $(document).on('keyup',function(e){
        $('button[data-event_key="'+e.key+'"]').removeclass('active')
        console.log(e.key);
    })

    $('.calc-btn').on('click',function(e){
        var key = $(this).data('event_key')
        if(key != '+' && key != '-' && key != '=' && key != '*' && key != '/' && key != 'delete'
&& key != 'numlock'){
            appendnumber(key)
        }else{
            if(key=='+' || key=='-' || key=='/' || key=='*'){
                generateExpression(key)
            }else if(key == '='){
                evaluateExpression()
            }else if(key == 'delete'){
                clearCalc()
            }
        }
        console.log(key)
    })

    function appendnumber(number){
        var existingnumber = $("#number_div").html();
        console.log(existingnumber);
        var currentstring = number;
        var outpurstring = '';
        if(existingnumber != '' && existingnumber != undefined && existingnumber != null){
            if(existingnumber=='0'){
                outputstring = number
            }else{
                outputstring = existingnumber+=currentstring
            }
        }else{
```

```
            outputstring = currentstring;
        }
        $("#number_div").html(outputstring);
    }
    function generateExpression(operator){
        var existingnumber = $("#number_div").html();
        var currentoperator = operator;
        var expression = '';
        var savedExpression = $("#savedExpression").val();
        if(savedExpression=='' || savedExpression==null || savedExpression==undefined){
            expression =parseInt(existingnumber) + operator
        }else{
            expression =savedExpression +existingnumber + operator;
        }
        $("#number_div").html("")
        $("#savedExpression").val(expression)
        $("#expression").html(expression)
    }
    function evaluateExpression(){
        var result= '';
        var expression = $("#savedExpression").val();
        var existingnumber = $("#number_div").html();
        if(existingnumber != '' || existingnumber != null || existingnumber != undefined){
            expression = expression + parseInt(existingnumber);
            $("#expression").html(expression)
        }
        result = eval(expression);
        $("#savedExpression").val("")
        $("#number_div").html(result)
        $("#value_div").html(result)
    }
    function clearCalc(){
        $("#number_div").html("")
        $("#savedExpression").val("")
        $("#expression").html("")
        $("#value_div").html("")
    }
  </script>
 </body>
</html>
```

Buttons and Display :

Number Buttons : Each number button (0 to 9) has a data-event_key attribute with the corresponding numeric value. When clicked, the “appendnumber()” function is called with the respective number.

Operator Buttons : Operator buttons (+, -, *, /) also have a “data-event_key” attribute with the corresponding operator symbol. When clicked, the `generateExpression()` function is called with the respective operator.

Special Buttons : There are special buttons for “.” (decimal point), “=” (equal sign for evaluation), “CE” (Clear Entry), and `DEL` (Delete). These buttons function similarly, triggering specific actions when clicked.

JavaScript Functions :

appendnumber(number) : This function appends a number to the current display. If the current display is 0, it replaces it with the new number. Otherwise, it appends the number to the existing display.

generateExpression(operator) : This function generates an expression by appending the current number and operator to the existing expression. It updates both the display and the hidden input field with the expression.

EvaluateExpression() : This function evaluates the expression stored in the hidden input field using JavaScript's “eval()” function. It updates the display with the result.

clearCalc() : This function clears the calculator display and resets the stored expression to empty.

Event Handling:

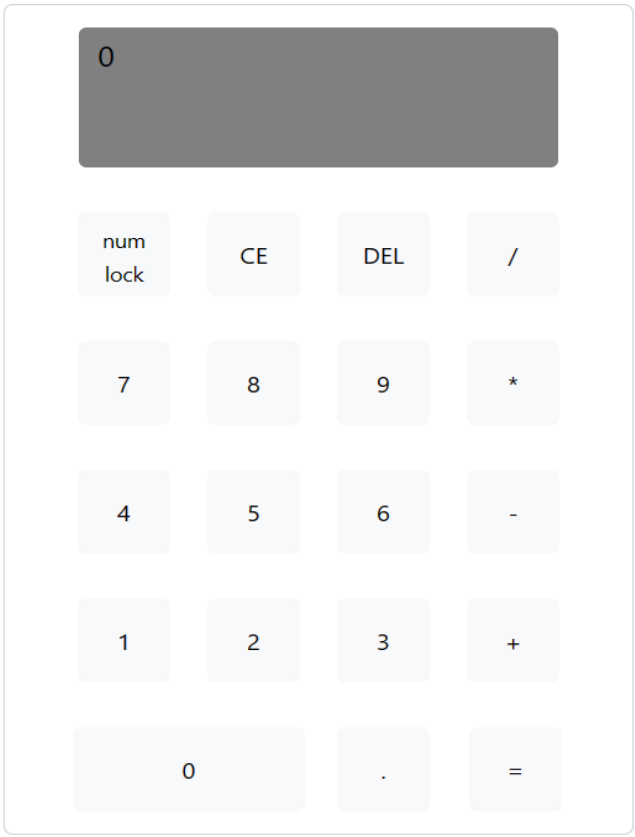
Key Events : The calculator also responds to keypress and keyup events. When a key is pressed, the corresponding button is highlighted (active) on the calculator interface, and the appropriate action is triggered based on the pressed key. The `keypress` event handles appending numbers and generating expressions, while the `keyup` event removes the highlighting from the button.

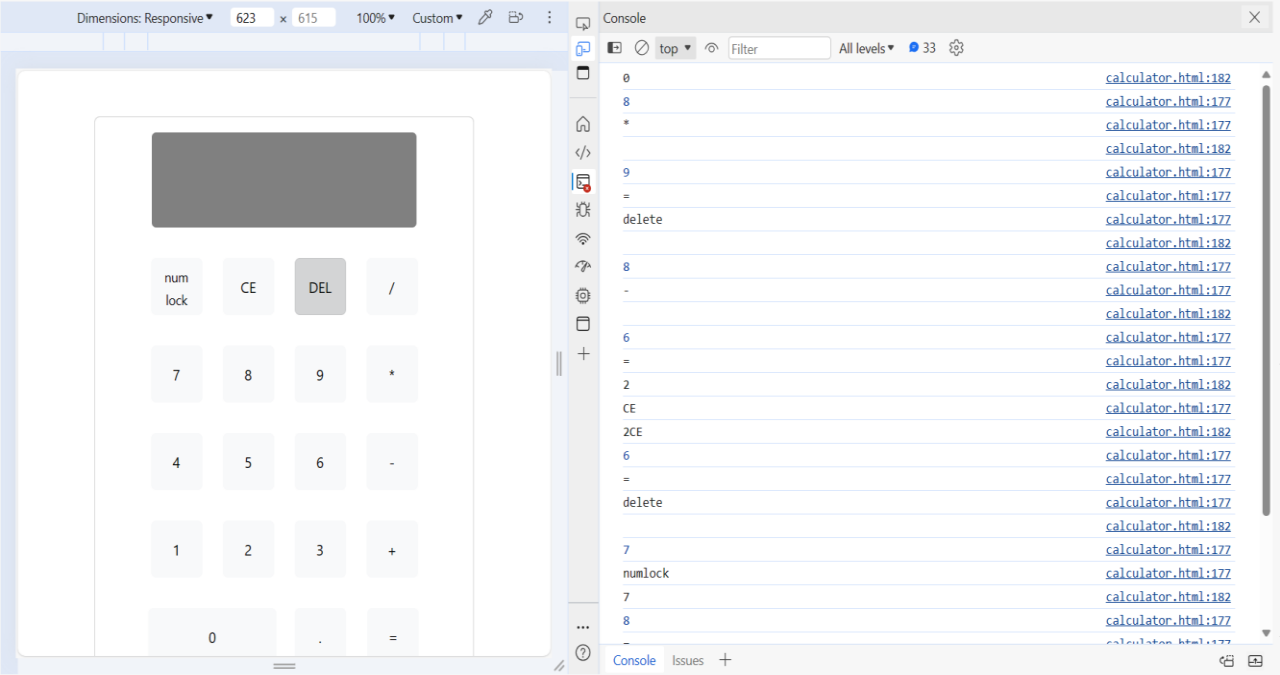
Button Click Events: Click events are attached to all calculator buttons using jQuery. When a button is clicked, the corresponding action function is invoked based on the button's “data-event_key”.

Display Updates :

- The calculator display consists of three sections: “number_div” for the current number being entered, “expression” for displaying the ongoing expression, and `value_div` for displaying the result after evaluation. These sections are updated dynamically based on user inputs and calculation results.

OUTPUT :





CONCLUSION :

In conclusion, the HTML calculator project represents a significant step forward in the realm of web-based computational tools, showcasing the power of modern web technologies to address users' everyday needs. Its successful implementation underscores the potential for innovation and efficiency in leveraging HTML, Bootstrap, and jQuery to create intuitive and accessible calculators. The responsive design and user-friendly interface of the calculator not only facilitate seamless arithmetic operations but also reflect a commitment to enhancing user experience across different devices and usage scenarios.

Looking ahead, there is ample opportunity for further development and refinement of the HTML calculator project. Additional features such as scientific functions, unit conversions, or memory capabilities could expand the calculator's utility and cater to a broader range of user requirements. Moreover, ongoing optimization efforts can focus on improving performance, refining the user interface, and ensuring compatibility with emerging web technologies and standards.

Beyond its immediate functionality, the HTML calculator project serves as a testament to the transformative potential of web development in delivering practical solutions. It highlights the adaptability and versatility of web-based tools in meeting the evolving needs of users in today's digital landscape. As technology continues to advance, projects like the HTML calculator demonstrate the enduring relevance of web technologies in empowering individuals and streamlining everyday tasks.

In essence, the HTML calculator project stands as a testament to the ingenuity and creativity inherent in web development, showcasing how innovative solutions can be crafted to simplify and enhance users' lives. By harnessing the capabilities of HTML, Bootstrap, and jQuery, the project exemplifies the boundless possibilities of web-based applications in driving efficiency, accessibility, and user satisfaction.