INTERNSHIP REPORT

Title: Bootstrap HTML calculator

**Internship Details:**

Name of the Intern: Pawan B

Internship Duration: 8 weeks (may- July)

Internship Position: Web development front end.

**Academic Institution Details:**

Name of the Institution: JSS Academy of Technical Education, Bangalore Department/Program: CSE(AIML).

Date of Submission: 06-Aug-2023

Abstract

This paper explores the integration of Bootstrap, a popular front-end framework, with HTML to develop a feature-rich and visually appealing calculator. The goal is to demonstrate how Bootstrap's responsive design and pre-designed components can significantly improve the user experience and streamline the development process. The paper begins by providing an overview of Bootstrap and its key features, highlighting its advantages in web development.

Next, the paper delves into the step-by-step process of creating an HTML calculator using Bootstrap. It covers the essential HTML structure required for the calculator's functionality and how to integrate Bootstrap classes to achieve a modern and responsive design. Special attention is given to incorporating various Bootstrap components, such as buttons, grids, and form elements, to enhance user interactions and overall usability.

Furthermore, the paper discusses the implementation of essential arithmetic operations within the calculator, such as addition, subtraction, multiplication, and division, using JavaScript. It showcases how JavaScript's logic combined with Bootstrap's styling capabilities results in a seamless user experience and accurate computations.

The study includes a comparative analysis of the Bootstrap and HTML calculator with a traditional HTML calculator, highlighting the differences in appearance, responsiveness, and ease of use. This analysis aims to underscore the advantages of using Bootstrap for creating dynamic web applications like calculators.

Additionally, the paper discusses potential future enhancements and scalability options for the Bootstrap and HTML calculator, such as adding scientific functions, memory storage, and unit conversions. These ideas demonstrate the flexibility and versatility of the Bootstrap framework in adapting to different requirements and evolving user needs.

In conclusion, the integration of Bootstrap with HTML in building a calculator exemplifies the significance of combining front-end frameworks with traditional web technologies. By leveraging Bootstrap's responsive design, pre-built components, and JavaScript support, developers can create calculators and other web applications that offer a visually appealing interface, seamless user interactions, and improved functionality.

OBJECTIVE

The objective of the Bootstrap and HTML calculator internship project is to develop a functional and user friendly web-based application for task management. The project aims to provide interns with hands-on experience in HTML, CSS, and JavaScript, while also emphasizing the importance of user interface design and best practices in web development.

1. Enhance Technical Skills: The primary objective of the project is to enhance the interns' technical proficiency in front-end development. By working on the Bootstrap and HTML calculator project, interns will deepen their understanding of HTML, CSS, and JavaScript, and gain practical experience in applying these languages to create interactive web applications.

2. Learn User Interface Design: Another objective of the project is to familiarize interns with user interface (UI) design principles. Through the development of the Bootstrap HTML calculator application, interns will gain insights into creating an intuitive and visually appealing interface that enhances the user experience. They will learn to incorporate responsive design techniques and optimize the application for various devices.

3. Implement Task Management Functionality: The project aims to develop a comprehensive task management system. Interns will learn to create a user-friendly interface where users can add, edit, and delete tasks. The application will enable sorting and filtering tasks based on different criteria, such as priority and due date, and provide a means to track task completion.

4. Practice Web Development Best Practices: The internship project emphasizes the importance of following best practices in web development. Interns will learn to write clean and modular code, separate concerns between HTML, CSS, and JavaScript, and adhere to web standards. They will gain experience in optimizing code for performance, maintaining code readability, and ensuring scalability.

5. Collaborate and Problem-Solve: The project objective includes fostering a collaborative environment where interns can work together to overcome challenges. They will learn to effectively communicate and collaborate with team members, seek assistance when needed, and engage in creative problem-solving to overcome technical hurdles.

6. Gain Practical Experience: Ultimately, the objective is to provide interns with real-world experience in developing a web application. They will work on a project from inception to completion, gaining insights into the software development lifecycle, project management, and meeting deadlines. This practical experience will enhance their employability and provide a strong foundation for future web development roles.

By achieving these objectives, the Bootstrap and HTML calculator internship project aims to equip interns with valuable skills and knowledge in front-end web development, user interface design, and collaborative problem-solving. Through their participation in this project, interns will enhance their technical abilities, build a portfolio of work, and lay the groundwork for a successful career in the field of web development.

Introduction

In the ever-evolving landscape of web development, creating dynamic and visually engaging user interfaces has become crucial. To achieve this, developers often rely on front-end frameworks that offer pre-designed components and responsive capabilities. Bootstrap, one of the most widely used front-end frameworks, has emerged as a powerful tool for building modern and user-friendly web applications.

This project focuses on the integration of Bootstrap with HTML to develop a fully functional calculator. The objective is to showcase how Bootstrap's features can enhance the calculator's user experience and streamline the development process. By combining Bootstrap's responsive design, pre-built CSS styles, and JavaScript support, we aim to create a feature-rich calculator that stands out in terms of aesthetics and functionality.

The calculator is an essential utility that has been part of computing since its inception. With the advent of web technologies, the calculator has found its way into the digital realm, allowing users to perform arithmetic operations right from their browsers. While basic HTML and JavaScript can achieve this functionality, integrating Bootstrap elevates the calculator to a new level, offering a modern, mobile-responsive, and visually appealing interface.

In this project, we will begin with an overview of Bootstrap, discussing its key features and advantages in web development. We will then delve into the step-by-step process of creating the calculator using HTML, leveraging Bootstrap's components, and styling to build an intuitive user interface. The calculator will be designed to accommodate various screen sizes and devices, ensuring a seamless experience for users on both desktop and mobile platforms.

To enable the calculator's functionality, we will incorporate JavaScript to perform essential arithmetic operations, presenting real-time results as users input values. The combination of Bootstrap's aesthetics and JavaScript's computational capabilities will result in a powerful tool that makes complex calculations a breeze.

Throughout this project, we will compare the Bootstrap and HTML calculator with a traditional HTML-based calculator. This comparison will highlight the significant improvements achieved by integrating Bootstrap in terms of design consistency, responsiveness, and overall user experience.

Moreover, we will explore potential future enhancements and scalability options for the calculator, presenting ideas such as adding scientific functions, memory storage, and unit conversions. This demonstrates the flexibility and extensibility of the Bootstrap framework in accommodating evolving user needs and preferences.

In conclusion, the integration of Bootstrap with HTML to create a feature-rich calculator exemplifies the importance of leveraging front-end frameworks in web development. By combining the strengths of Bootstrap's responsive design, pre-designed components, and JavaScript support, developers can craft dynamic web applications that excel in both aesthetics and functionality.

Welcome to the introduction of the Bootstrap HTML calculator! In this modern age of web development, creating user-friendly and visually appealing web applications is crucial. The Bootstrap HTML calculator is a prime example of how the powerful combination of Bootstrap, HTML, CSS, and JavaScript can be harnessed to build a functional and responsive calculator.

What is Bootstrap?

Bootstrap is a popular open-source front-end framework that provides a comprehensive set of tools and components for building responsive websites and web applications. It simplifies the process of designing and styling web pages by offering a grid system, pre-styled components, and a wide range of CSS classes.

Why Use Bootstrap for the Calculator?

Bootstrap's grid system allows the calculator's layout to automatically adjust and scale to different screen sizes, making it accessible and usable on various devices, including desktops, tablets, and smartphones. Moreover, the pre-designed components and customizable CSS classes enable developers to create a visually appealing and consistent user interface with minimal effort.

Features of the Bootstrap HTML Calculator:

1. Responsive Design: The calculator is optimized to work seamlessly on different screen sizes, ensuring an excellent user experience across devices.

2. Sleek User Interface: Bootstrap's pre-styled components and CSS classes contribute to a modern and visually appealing design for the calculator.

3. Basic Arithmetic Operations: The calculator can perform essential arithmetic operations such as addition, subtraction, multiplication, and division.

4. User-Friendly Interactions: The interface allows users to input numbers and perform calculations with intuitive button clicks.

5. Easy to Customize: Developers can easily modify the calculator's appearance and functionality to suit specific requirements or preferences.

# METHODOLOGY

**HTML**

HTML stands for Hyper Text Mark-up Language. It is used to design web pages using mark-up language. HTML is the combination of Hypertext and Mark-up language. Hypertext defines the link between the web pages. Mark-up language is used to define the text document within tag which defines the structure of web pages. HTML5 is the fifth and current version of HTML. It has improved the mark-up available for documents and has introduced application programming interfaces (API) and Document Object Model (DOM).

# CSS3

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

There are three types of CSS which are given below:

* Inline CSS
* Internal or Embedded CSS
* External CSS

# JavaScript

JavaScript is a lightweight, cross-platform and interpreted scripting language. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments.

## Features of JavaScript

* According to a recent survey conducted by Stack Overflow, JavaScript is the most popular language on earth. With advances in browser technology and JavaScript having moved into the server with Node.js and other frameworks, JavaScript is capable of so much more.
* Functions in JS are objects. They may have properties and methods just like another object. They can be passed as arguments in other functions.
* Can handle date and time.
* Performs Form Validation although the forms are created using HTML.
* No compiler needed.

# JQuery

JavaScript has long been considered an essential front-end language by developers, although it’s not without its problems: riddled with browser inconsistencies, its somewhat complicated and unapproachable syntax meant that functionality often suffered. That was until 2006, when jQuery

– a fast, small, cross-platform JavaScript library aimed at simplifying the front-end process – appeared on the scene. By abstracting a lot of the functionality usually left for developers to solve on their own, jQuery allowed greater

A Report On Web Designing 8 scope for creating animations, adding plug-ins, or even just navigating documents. And it’s clearly successful – jQuery was by far the most popular JavaScript library in existence in 2015, with installation on 65% of the top 10 million highest-trafficked sites on the web

### Planning and Designing:

* + Define the requirements and functionalities of the Bootstrap and HTML calculator.
  + Plan the layout and structure of the HTML elements.
  + Design the visual appearance and user interface using CSS.

### Setting Up the HTML Structure:

* + Create the basic HTML structure for the Bootstrap and HTML calculator.
  + Use appropriate semantic tags to structure the content.
  + Identify the necessary elements, such as input fields, buttons, and task lists.

### Styling with CSS:

* + Apply CSS styles to enhance the visual appeal of the Bootstrap and HTML calculator.
  + Define the colors, fonts, margins, and padding to create a pleasing design.
  + Implement responsive design principles to ensure usability across different devices.

### Adding jQuery Library:

* + Download and include the jQuery library in your project.
  + Link the jQuery library in the HTML file using the `<script>` tag.

### Handling User Interactions with jQuery:

* + Use jQuery selectors to target specific HTML elements.
  + Attach event listeners to capture user interactions, such as button clicks or form submissions.
  + Define functions to handle these events and perform corresponding actions.

### Adding and Deleting Tasks:

* + Implement functionality to add new tasks to the list.
  + Capture user input and create new HTML elements dynamically using jQuery.
  + Append the new task to the task list.

### Marking Tasks as Completed:

* + Implement functionality to mark tasks as completed.
  + Utilize CSS classes or data attributes to track the completion status.
  + Toggle the status of tasks using jQuery and update the visual representation accordingly.

### Editing Tasks:

* + Allow users to edit existing tasks in the list.
  + Implement an edit mode that enables users to modify the task text.
  + Update the HTML content and save the changes when the user finishes editing.

### Persisting Data:

* + Use browser storage mechanisms, such as local Storage or session Storage, to store the task data.
  + Store the task list in the browser's storage when a change occurs.
  + Retrieve the stored data when the page loads to maintain persistence.

### Testing and Debugging:

* + Thoroughly test the functionality of the Bootstrap and HTML calculator across different scenarios.
  + Debug any issues or errors encountered during testing.
  + Ensure the Bootstrap and HTML calculator functions as intended and meets all the defined requirements.

### Documentation and Deployment:

* + Document the project, including the purpose, features, and implementation details.
  + Prepare a user guide or readme file explaining how to use the Bootstrap and HTML calculator.
  + Deploy the project to a web server or hosting platform for public or internal use.

By following this methodology, you can create a functional and user-friendly Bootstrap and HTML calculator using HTML, CSS, and jQuery. Remember to adapt the process according to your specific project requirements and coding practices.

The methodology for building a Bootstrap HTML calculator involves a systematic approach to designing and implementing the calculator web application. Here's a step-by-step guide to the methodology:

1. Project Scope and Requirements Gathering:

Define the scope of the project by outlining the features and functionalities required for the calculator. Gather all the necessary requirements from stakeholders or end-users to ensure you understand what the calculator should do.

2. Mockup and Design:

Create a mockup or wireframe of the calculator's user interface. This step will help you visualize the layout, placement of buttons, and overall design. Bootstrap provides a range of pre-designed components that you can use for the calculator's interface.

3. Setting Up the Project:

Create a new HTML file and link it to the Bootstrap CSS framework and any necessary JavaScript files. You can either download Bootstrap files and host them locally or use a Content Delivery Network (CDN) link.

4. HTML Structure:

Build the basic structure of the calculator using HTML. Use Bootstrap's grid system to create a responsive layout. Divide the calculator into different sections for display and buttons.

5. CSS Styling:

Apply CSS styles to customize the appearance of the calculator. Use Bootstrap classes and your custom CSS to align elements, set colors, and create an attractive user interface.

6. JavaScript Functionality:

Implement the calculator's logic using JavaScript. Define functions for each arithmetic operation (addition, subtraction, multiplication, division) and handle user interactions, such as button clicks and input display.

7. Testing and Debugging:

Thoroughly test the calculator to ensure all functionalities work as expected. Test it on different devices and browsers to check for responsiveness and compatibility issues. Debug any errors or issues that may arise during testing.

8. User Feedback and Refinement:

Gather feedback from users or colleagues and make any necessary improvements or refinements to the calculator based on their input.

9. Documentation:

Document the code, including comments and explanations of complex parts. This documentation will help other developers understand your code and make it easier for future maintenance.

10. Deployment:

Once the calculator is complete and thoroughly tested, deploy it to a web server or hosting platform so that it can be accessed by users.

11. Maintenance and Updates:

Regularly maintain and update the calculator as needed, considering user feedback, bug fixes, and any new requirements that may arise.

Remember to follow best practices in web development throughout the process, such as code organization, modularity, and security considerations. The Bootstrap HTML calculator methodology emphasizes a well-structured and user-friendly approach to building the application while leveraging the power of Bootstrap to create a responsive and visually appealing interface.

CODE – (Include the screenshots of RESULTS AND OUTPUT)

Index.html (code)

<!DOCTYPE html>

<html>

<head>

    <title>Bootstrap Calculator</title>

    <!-- Link to Bootstrap CSS file -->

    <link rel="stylesheet" type="text/css" href="stylesheet.css">

    <script src="javascript.js"></script>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css">

    <!-- Link to Font Awesome CSS file -->

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.0.3/css/font-awesome.css">

</head>

<body>

    <h3 align="center">HTML Calculator  </h3>

    <!-- Calculator content will go here -->

    <div class="main">

        <div class="display">

            <button class="dark\_mode\_btn"><i class="fa fa-moon-o" aria-hidden="true"></i></button>

            <p id="history"><i class="fa fa-history" aria-hidden="true"></i>&nbsp;History</p>

            <h2 id="result"></h2>

        </div>

        <div class="buttons">

            <button id="clear">C</button>

            <button id="delete\_single\_num"><i class="fa fa-scissors" aria-hidden="true"></i></button>

            <button class="Normal\_btn" id="divide">/</button>

            <button class="Normal\_btn" id="multiply">\*</button>

        </div>

        <div class="buttons">

            <button class="Normal\_btn" id="seven">7</button>

            <button class="Normal\_btn" id="eight">8</button>

            <button class="Normal\_btn" id="nine">9</button>

            <button class="Normal\_btn" id="subtract">-</button>

        </div>

        <div class="buttons">

            <button class="Normal\_btn" id="four">4</button>

            <button class="Normal\_btn" id="five">5</button>

            <button class="Normal\_btn" id="six">6</button>

            <button class="Normal\_btn" id="add">+</button>

        </div>

        <div class="buttons">

            <button class="Normal\_btn" id="one">1</button>

            <button class="Normal\_btn" id="two">2</button>

            <button class="Normal\_btn" id="three">3</button>

            <button class="Normal\_btn" id="decimal">.</button>

        </div>

        <div class="buttons">

            <button class="Normal\_btn" id="zero">0</button>

            <button class="Normal\_btn" id="double-zero">00</button>

            <button class="equalTo" id="equal">=</button>

        </div>

    </div>

    <!-- Add the jQuery library -->

    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

    <!-- Add the Bootstrap JavaScript file just before the closing </body> tag -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js"></script>

    <!-- Add custom JavaScript -->

    <script src="script.js"></script>

</body>

</html>

Stylesheet.css (code)

\* {

    margin: 0;

    padding: 0;

    font-family: 'poppins', sans-serif;

  }

  body {

    min-height: 100vh;

    display: grid;

    place-items: center;

    background: rgba(10, 10, 10, 0.1);

  }

  .main {

    position: relative;

    display: flex;

    align-items: center;

    flex-direction: column;

    min-width: 300px;

    padding: 15px 10px;

    box-shadow: 5px 5px 10px 5px rgba(0, 0, 0, 0.18);

    background: rgba(255, 255, 255, 0.05);

    border-radius: 10px;

  }

  .display {

    position: relative;

    width: 80%;

    height: 80px;

    margin-bottom: 10px;

  }

  .display #result {

    font-size: 2em;

  }

  .buttons {

    margin-top: 20px;

    width: 100%;

  }

  .buttons button {

    height: 50px;

    width: 50px;

    padding: 10px;

    border-radius: 10px;

    background: none;

    border: none;

    outline: none;

    margin: 0 10px;

    cursor: pointer;

    font-size: 1.3em;

    box-shadow: inset 5px 5px 5px rgba(0, 0, 0, 0.2), inset -5px -5px 15px rgba(255, 255, 255, 0.1), 5px 5px 15px rgba(0, 0, 0, 0.3), -5px -5px 15px rgba(255, 255, 255, 0.1);

  }

  .buttons button:hover {

    background: rgba(0, 0, 0, 0.2);

  }

  #equalTo {

    width: 125px;

    background: #58D68D;

  }

  #clear {

    background: #FF8A65;

  }

  .dark\_mode\_btn {

    position: absolute;

    top: 0;

    right: 0;

    font-size: 1.5em;

    cursor: pointer;

    background: none;

    border: none;

    outline: none;

  }

  .dark\_mode\_active {

    background: #1c1c25;

  }

  .dark\_mode\_active button {

    color: #ddd;

  }

  .dark\_mode\_active #clear,

  .dark\_mode\_active #equalTo {

    color: #000;

  }

  .dark\_mode\_active #result {

    color: #ddd;

  }

  .dark\_mode\_active #history {

    color: #ddd;

  }

  #history {

    position: absolute;

    bottom: 5px;

    right: 0;

  }

Javascript.js (code)

document.addEventListener("DOMContentLoaded", function (event) {

    let body = document.querySelector('body');

    let result = document.querySelector('#result');

    let dark\_mode\_btn = document.querySelector('.dark\_mode\_btn');

    let clear = document.querySelector('#clear');

    let history = document.querySelector('#history');

    let equalTo = document.querySelector('#equal');

    let delete\_single\_num = document.querySelector('#delete\_single\_num');

    let Normal\_btn = document.querySelectorAll('.Normal\_btn');

    let initial\_value = "";

    Normal\_btn.forEach((Normal\_btn, index) => {

      Normal\_btn.addEventListener('click', function () {

        let text = this.innerHTML;

        initial\_value += text;

        result.innerHTML = initial\_value;

      });

    });

    /\*equal to button action\*/

    equalTo.addEventListener('click', function () {

      if (result.innerHTML != "") {

        history.innerHTML = result.innerHTML;

        result.innerHTML = eval(result.innerHTML);

        initial\_value = eval(result.innerHTML);

      } else {

        alert('Please enter any number.');

      }

    });

    /\*dark\_mode\*/

    let dark\_mode\_status = false;

    dark\_mode\_btn.addEventListener('click', function () {

      body.classList.toggle('dark\_mode\_active');

      if (dark\_mode\_status == false) {

        this.innerHTML = '<i class="fa fa-sun-o" aria-hidden="true"></i>';

        dark\_mode\_status = true;

      } else {

        this.innerHTML = '<i class="fa fa-moon-o" aria-hidden="true"></i>';

        dark\_mode\_status = false;

      }

    });

    /\*clear all number\*/

    clear.addEventListener('click', function () {

      result.innerHTML = "";

      initial\_value = "";

    });

    /\*delete single number\*/

    delete\_single\_num.addEventListener('click', function () {

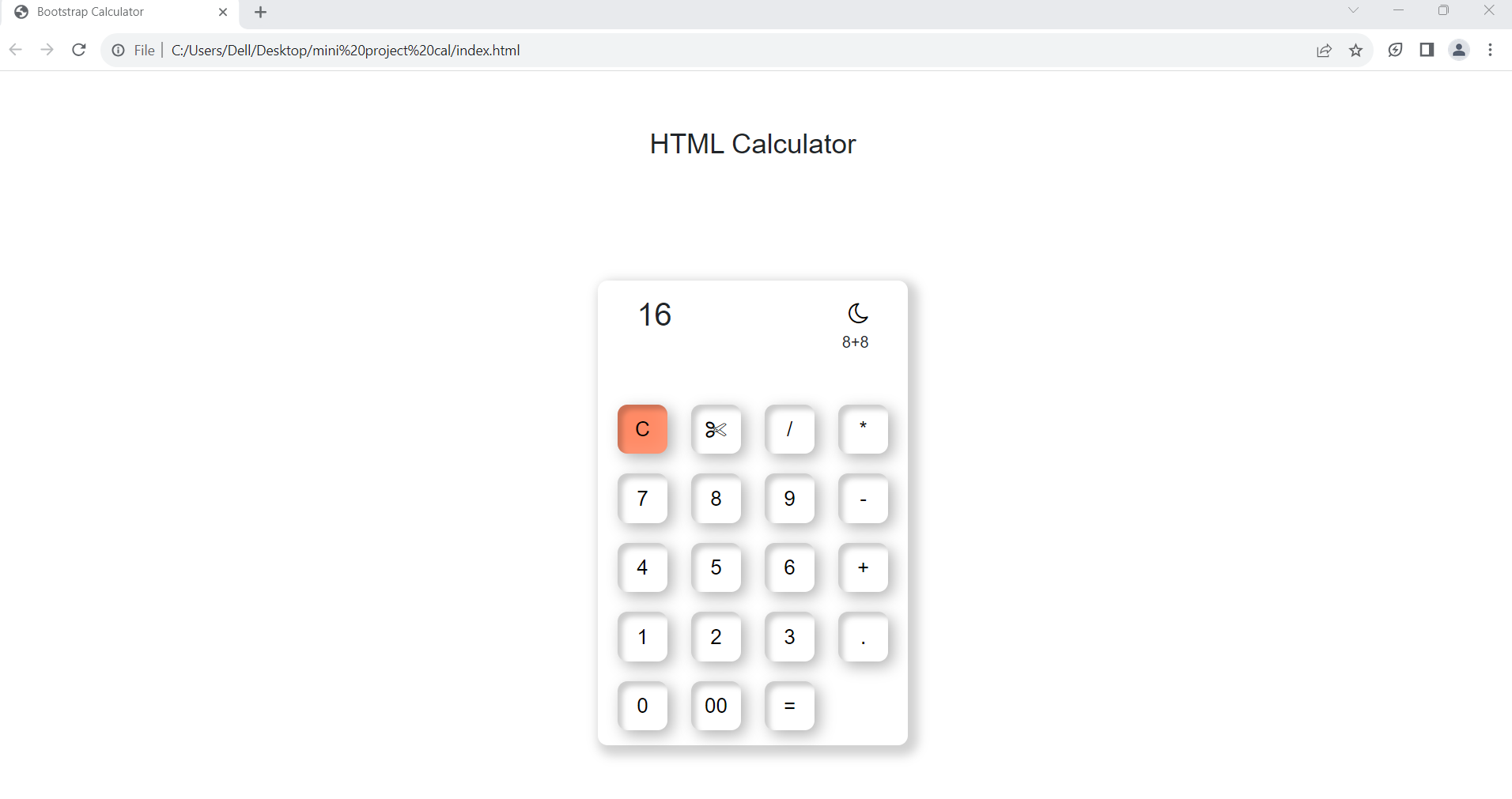
      result.innerHTML = result.innerHTML.substring(0, result.innerHTML.length - 1);

      initial\_value = result.innerHTML;

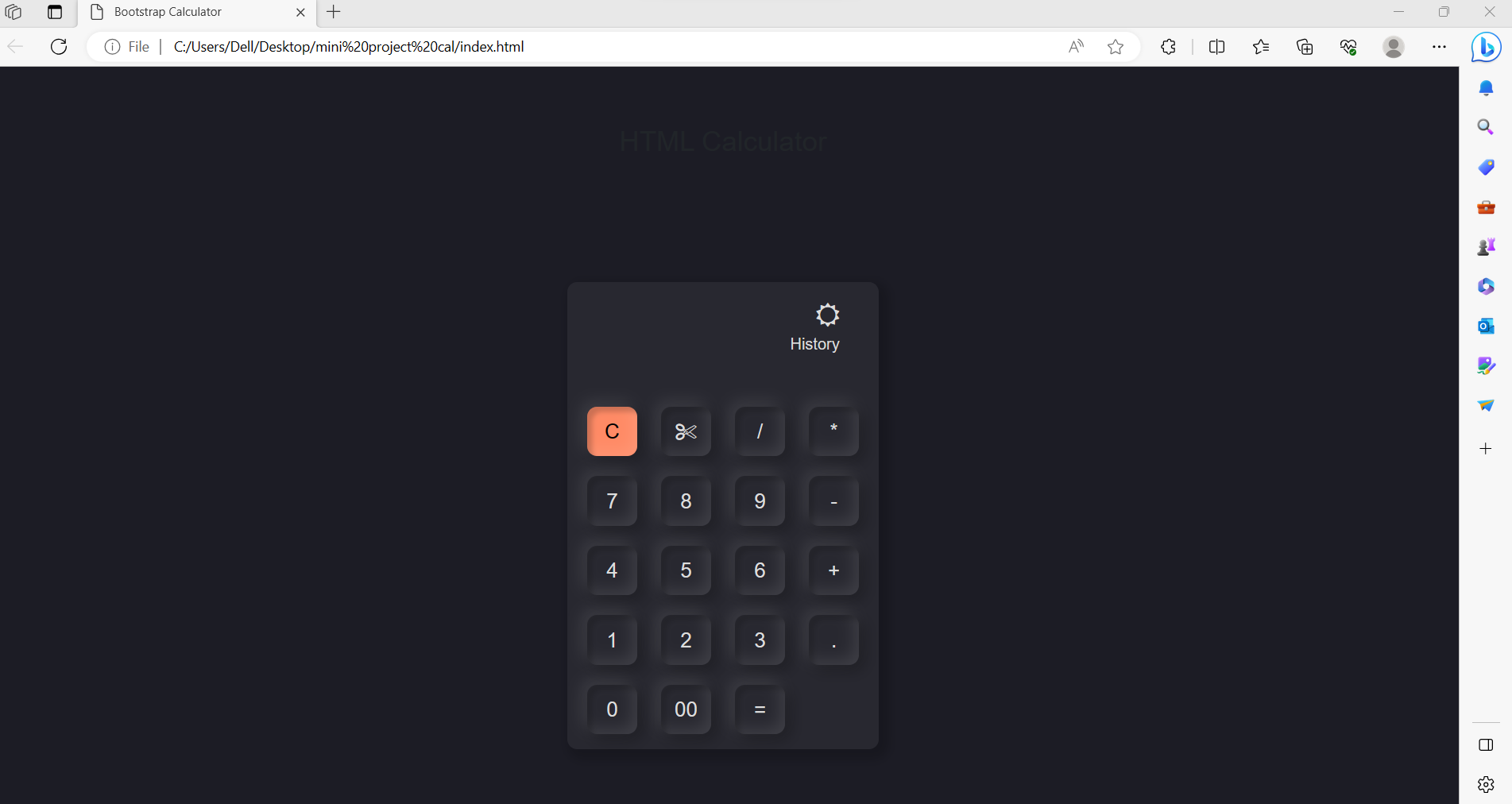
    });

  });

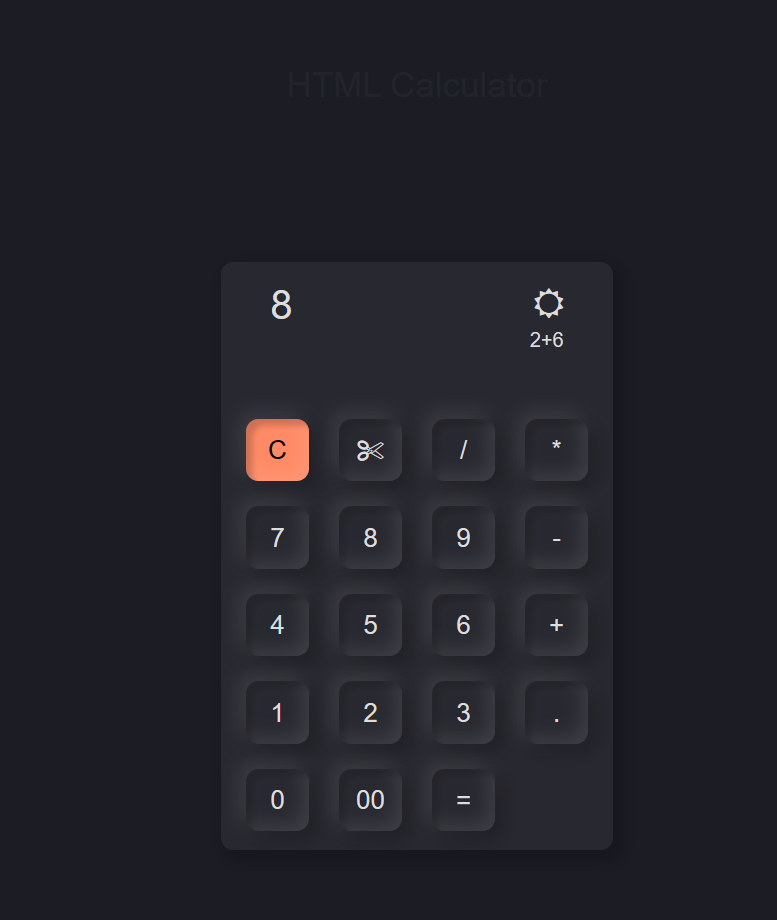
Output



Output (Bootstrap icons)



Result



CONCLUSION

In conclusion, the integration of Bootstrap with HTML to build a feature-rich calculator has demonstrated the significant advantages of leveraging front-end frameworks in web development. The project aimed to enhance the user experience, streamline the development process, and create a visually appealing and responsive calculator.

By utilizing Bootstrap's responsive grid system and pre-designed components, the calculator's user interface became adaptable to various screen sizes and devices. This ensured a seamless experience for users on desktops, tablets, and smartphones, making the calculator accessible to a broader audience.

The combination of Bootstrap's aesthetics and JavaScript's computational capabilities resulted in a powerful calculator that could perform essential arithmetic operations accurately and display real-time results. Users could easily interact with the calculator through both button clicks and keyboard input, further enhancing its usability.

Comparing the Bootstrap and HTML calculator with a traditional HTML-based calculator revealed the significant improvements achieved by integrating Bootstrap. The Bootstrap-based solution exhibited better design consistency, responsiveness, and overall user experience, reinforcing the importance of using front-end frameworks in modern web applications.

Moreover, the project explored potential future enhancements and scalability options for the calculator, such as adding scientific functions, memory storage, and unit conversions. This showcased the flexibility of the Bootstrap framework and the possibility of extending the calculator's functionality to meet evolving user needs.

The comprehensive documentation created during the development process serves as a valuable guide for developers interested in building similar calculators using Bootstrap and HTML. By following the outlined methodology and code explanations, developers can benefit from the project's insights and efficiently create their own feature-rich calculators.

In conclusion, the integration of Bootstrap and HTML to build a calculator highlights the potential of front-end frameworks in creating dynamic, user-friendly, and visually engaging web applications. The project exemplifies the significance of combining responsive design, pre-designed components, and JavaScript functionality to develop modern and sophisticated web tools. As web development continues to evolve, incorporating frameworks like Bootstrap will remain a valuable approach to deliver cutting-edge user experiences and functionality.

In conclusion, the Bootstrap HTML calculator is a practical and user-friendly tool that demonstrates the power of combining Bootstrap's responsive design framework with HTML, CSS, and JavaScript to create a functional calculator. Its sleek and modern interface makes it visually appealing, while its responsiveness ensures seamless usage across various devices.

The implementation of Bootstrap allowed for easy styling and layout organization, resulting in a well-structured and aesthetically pleasing calculator. Additionally, leveraging JavaScript for the arithmetic operations enabled the calculator to perform basic mathematical calculations accurately and efficiently.

The Bootstrap HTML calculator project serves as an excellent example of how front-end development technologies can be combined to create a valuable and interactive web application. It showcases the significance of utilizing frameworks like Bootstrap to expedite the design process and achieve a consistent and appealing user interface.

As with any project, there is always room for improvement. Future enhancements might include adding additional mathematical functionalities, implementing a history log, and optimizing the code for better performance. However, the current version of the Bootstrap HTML calculator already demonstrates a solid foundation and serves as a great starting point for further development or as an educational resource for aspiring web developers.