**Internship Report**

**on**

**SIMPLE CALCULATOR USING HTML, CSS, JAVASCRIPT**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, ANANTHAPURAMU**

*In Partial Fulfillment of the Requirements for the Award of the degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted By**

**S SUNDAR RAJU - 21691A05M1**

****

**MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE**

**(UGC – AUTONOMOUS)**

**(Affiliated to JNTUA, Ananthapuramu)**

**(Accredited by NBA, Approved by AICTE, New Delhi)**

**AN ISO 9001:2015 Certified Institution**

**P. B. No: 14, Angallu, Madanapalle – 517325**

**2023 - 24**

**2009-2013**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that the internship work entitled **“Simple Calculator Using Html,Css,JavaScript”** is a bonafide work carried out by

**S Sundar Raju - 21691A05M1**

Submitted in partial fulfillment of the requirements for the award of degree **Bachelor of Technology** in the stream of **Computer Science & Engineering** in **Madanapalle Institute of Technology & Science, Madanapalle,** affiliated to **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu** during the academic year 2023-2024.

**Internship Coordinator Head of the Department**

Dr. K. Sudhakar, Dr. R. Kalpana,

Senior Assistant Professor, Professor & Head,

Department of CSE Department of CSE

**Submitted for the University Examination held on:**

**Examiner - I Examiner - II**

**ACKNOWLEDGEMENT**

We sincerely thank the **MANAGEMENT** of **Madanapalle Institute of Technology & Science** for providing excellent infrastructure and lab facilities that helped me complete this Project.

We sincerely thank **Dr. C. Yuvaraj, M.E., Ph.D., Principal,** for guiding and providing facilities for completing our Project at **Madanapalle Institute of Technology & Science,** Madanapalle.

We express our gratitude to **Dr. R. Kalpana, Ph.D., Professor and Head of the Department of CSE** for her continuous support in making necessary arrangements for the successful completion of the Project.

We express our sincere thanks to the **Internship Coordinator, Dr. K. Sudhakar, Senior Assistant Professor, Department of CSE** for his tremendous support for the successful completion of Project.

We express our deep gratitude to our Internship In-Charge **Mr.K.Satish, M.TECH., Assistant Professor, Department of CSE,** for his guidance and encouragement that helped us to complete this Project.

We also wish to place on record my gratefulness to other **Faculty of CSE Department** and our friends and our parents for their help and cooperation during our project work.





**DECLARATION**

We hereby declare that the results embodied in this internship **"Simple Calculator Using Html,Css,JavaScript"** by us under the guidance of **Mr.K.Sathish,** in partial fulfillment of the award of **Bachelor of Technology** in **Computer Science & Engineering** from **Jawaharlal Nehru Technological University Anantapur, Ananthapuramu**.

**Date :**

**Place :Angallu**

**Internship Associate:**

**S Sundar Raju**

**21691A05M1**

**ABSTRACT**

The abstract of a simple calculator refers to a concise summary of the calculator's key features and functionality without delving into specific details. In the case of a simple calculator, the abstract might read:

"This abstract summarizes the functionality of a basic, web-based simple calculator. The calculator provides a user-friendly interface with buttons for numbers and arithmetic operators, allowing users to perform basic mathematical calculations such as addition, subtraction, multiplication, and division. It follows the order of operations and offers clear and reset buttons to correct input or clear the calculation. The calculator supports keyboard input, offers error handling for common mathematical errors like division by zero, and displays results in real-time. It is a practical tool for quick and accurate arithmetic calculations and can serve as a learning project for beginners in web development."

# CONTENTS

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TOPIC** | **PAGE NO.** |
| **1** | **INTRODUCTION** |  |
|  | 1.1 About Web Development | 2 |
|  | 1.2 Importance and Applications of Web Development | 3-4 |
|  | 1.3 Front and Backend of Web Development | 5 |
|  | 1.4 Why Create a Simple Calculator | 6 |
|  |  |  |
| **2** | **TOOLS AND TECHNIQUES** |  |
|  | 2.1 Html , CSS and JavaScript | 8 |
|  | 2.2 Hardware Requirements(if any) | 8 |
|  | 2.3 Software Requirements | 8 |
|  |  |  |
| **3** | **PROJECT WORK** |  |
|  | 3.1 Project overview | 10 |
|  | 3.2 Algorithm | 11 |
|  |  |  |
| **4** | **CODE AND OUTPUT SCREENSHOTS** | 13-15 |
|  |  |  |
| **5** | **CONCLUSION** | 17 |
|  |  |  |
|  | **REFERENCES** | 19 |

**CHAPTER**-**1 INTRODUCTION**

* 1. **ABOUT WEB DEVELOPMENT**

Web development is the dynamic field responsible for building and maintaining websites and web applications. It encompasses both front-end development, focusing on the user interface and user experience, and back-end development, handling the server-side logic and databases. With the ever-increasing importance of an online presence, web development has become a fundamental skill in the digital age. Developers use a range of technologies, including HTML, CSS, and JavaScript, to create visually appealing and interactive websites that adapt to different devices and screen sizes. The development process also emphasizes security and performance optimization to protect against threats and deliver seamless user experiences. Whether it's crafting a personal blog, an e-commerce platform, or a cutting-edge web application, web development enables individuals and businesses to share information, engage with customers, and provide innovative online services to a global audience.

* 1. **IMPORTANCE AND APPLICATIONS OF WEB DEVELOPMENT**

Web development is a crucial field in today's digital age, with significant importance and a wide range of applications. Here are some key points highlighting the importance and applications of web development:

Importance:

1. Global Reach: The internet allows businesses and individuals to reach a global audience. Web development enables the creation of websites and web applications that can be accessed by people worldwide.

2. Digital Presence: For businesses, having a strong online presence is essential for marketing, customer engagement, and brand building. Web development is the foundation for establishing and maintaining this digital presence.

3. E-commerce: Web development powers e-commerce platforms, enabling businesses to sell products and services online. This has become increasingly important in the digital economy.

4. Information Access: Websites provide a platform for disseminating information, from news websites to educational resources, making knowledge accessible to a broad audience.

5. Communication: Web development facilitates communication through email, social media, chat applications, and more. These tools have transformed the way people interact and connect.

6. Innovation: The web constantly evolves, and web development is at the forefront of this innovation. New technologies and frameworks allow developers to create more dynamic and interactive web experiences.

7. Data Collection and Analysis: Web applications can collect user data, enabling businesses and organizations to gather insights and make data-driven decisions.

8. Automation: Web development can be used to automate various processes and tasks, improving efficiency and reducing manual labor.

Applications:

1. Business Websites: Nearly every business, regardless of its size or industry, can benefit from a website to showcase products, services, and contact information.

2. E-commerce: Online stores and marketplaces are built using web development, allowing customers to shop and make purchases online.

3. Blogs and Content Management: Content-driven websites and blogs rely on web development to create and manage articles, images, and multimedia content.

4. Social Media: Social media platforms like Facebook, Twitter, and Instagram are examples of complex web applications that require advanced web development.

5. News Portals: News websites deliver real-time information to a global audience, often with interactive features like comments and social sharing.

6. Educational Platforms: Online learning platforms and educational websites facilitate e-learning, offering courses, videos, and interactive content.

7. Government and Public Services: Government websites provide citizens with access to information, forms, and services, promoting transparency and efficiency.

8. Healthcare Systems: Web development is used to build electronic health record systems, telemedicine platforms, and health information portals.

9. Travel and Booking: Travel agencies and booking services rely on web development to manage reservations, provide information, and facilitate online bookings.

10. Personal Blogs and Portfolios: Individuals use web development to create personal websites for self-expression, sharing their work, or building an online resume.

11. Entertainment and Gaming: Web-based games, streaming services, and entertainment platforms are examples of web development in the entertainment industry.

12. Finance and Banking: Online banking, financial management tools, and investment platforms are web applications that require high security and advanced development.

13. Non-profit and Advocacy: Charities, nonprofits, and advocacy groups use web development to raise awareness, collect donations, and mobilize support.

Web development is a versatile and continuously evolving field, with applications across various sectors. Its importance lies in its ability to facilitate communication, information access, commerce, and innovation in the digital era.

**1.3 FRONTEND AND BACKEND OF WEB DEVELOPMENT**

Web development encompasses two essential components: front-end and back-end development. Front-end development, also known as client-side development, focuses on the user interface and user experience of websites and web applications. Front-end developers use technologies like HTML, CSS, and JavaScript to create the visual elements that users interact with directly, ensuring the site is visually appealing and responsive on various devices. They are responsible for the layout, design, and interactivity of the user interface.

In contrast, back-end development, or server-side development, deals with the behind-the-scenes functionality of web applications. Back-end developers work with server technologies, databases, and APIs to manage data, process requests from the front end, and ensure the security and performance of the server. They handle server logic, data storage, and communication between the client and server, making it possible for web applications to function efficiently and securely. Collaboration between front-end and back-end developers is crucial to create fully functional and user-friendly web experiences.

**1.4 WHY CREATE A SIMPLE CALCULATOR**

Creating a simple calculator can be a valuable exercise for several reasons, especially for individuals who are learning to program or looking to improve their programming skills. Here are some of the reasons why you might want to create a simple calculator:

1. Fundamental Programming Skills: Building a calculator is a straightforward project that can help you grasp the fundamental concepts of programming, such as variables, data types, conditional statements, and user input.

2. Problem-Solving: It provides an opportunity to practice problem-solving. You need to figure out how to take user input, perform mathematical operations, and display results, which are essential skills in programming.

3. Hands-On Experience: It offers hands-on experience in designing and implementing algorithms to perform arithmetic operations. This is particularly valuable for beginners.

4. User Interface Design: Designing a simple user interface for your calculator involves creating buttons, input fields, and output displays. This can introduce you to basic principles of user interface design.

5. Testing and Debugging: You will likely encounter bugs and errors in your code while creating a calculator. Debugging these issues is a crucial skill for any programmer.

6. Understanding Control Flow: Implementing the logic for addition, subtraction, multiplication, and division requires understanding how control flow works in your chosen programming language.

7. Practice with Functions: You can encapsulate different operations (e.g., addition, subtraction) in functions, which is a fundamental concept in programming.

8. Customization: You can customize your calculator by adding features like memory storage, scientific functions, or a history log, depending on your skill level and interest.

**CHAPTER-2**

**TOOLS AND TECHNIQUES**

**2.1 HTML, CSS AND JAVASCRIPT**

To create a simple calculator using web development, you'll need a combination of tools, technologies, and techniques. Here's a basic outline of the components you'll need and the steps to build a simple calculator using HTML, CSS, and JavaScript:

1. Text Editor or Integrated Development Environment (IDE): You can use text editors like Visual Studio Code, Sublime Text, or an IDE like WebStorm for writing your code.

2.HTML: For structuring the calculator interface.

3. CSS: For styling the calculator interface.

4. JavaScript: For handling user input, performing calculations, and updating the display.

1. HTML Structure: Create the HTML structure for your calculator. You'll need buttons for digits (0-9), arithmetic operators (+, -, \*, /), an equals sign (=), a clear button (C), and a display area to show the input and results.

2. CSS Styling: Use CSS to style your calculator. You can style buttons, the display area, and other elements to make your calculator visually appealing.

3. JavaScript Logic: Write JavaScript code to handle user interactions and perform calculations. You'll need event listeners to capture button clicks and update the display based on user input. Implement the arithmetic logic to calculate results.

4. Calculation Logic: Inside your event handler function, you'll need to implement the calculation logic. You can use JavaScript's `eval()` function or write your custom logic to evaluate and display results.

5. Clear and Error Handling: Implement functionality to clear the display when the "C" button is clicked, and handle error cases (e.g., division by zero).

6. Testing: Test your calculator by entering various input scenarios to ensure that it performs calculations accurately and handles edge cases gracefully.

7. Refinements: You can further enhance your calculator by adding features like keyboard input support, memory functions (M+, M-, MR, MC), and improving the user experience.

8. Responsive Design: Consider making your calculator responsive to different screen sizes by using CSS media queries.

9. Deployment: If you want to share your calculator with others, you can host it on a web server or use a platform like GitHub Pages for free hosting.

Once you've implemented these tools and techniques, you'll have a functional simple calculator built using web development technologies. You can expand on this foundation to create more sophisticated calculators with additional features and functionality.

**2.2 HARDWARE REQUIREMENTS**

In web development, the hardware requirements are generally modest, with a modern computer (laptop or desktop) equipped with at least an Intel Core i5 or equivalent AMD processor, 8 GB of RAM, and an SSD being suitable for most tasks. The choice of operating system (e.g., Windows, macOS, or Linux), a high-resolution monitor, a reliable internet connection, and comfortable input devices (keyboard and mouse/trackpad) are essential. Additionally, having a backup solution in place, appropriate development software tools, and optional peripherals based on specific needs can further enhance the web development experience. The hardware requirements can vary depending on the complexity of projects and the technologies used. It's important to check the specific requirements of the tools and frameworks you plan to work with.

**2.3 SOFTWARE REQUIREMENTS**

Software requirements in web development encompass various tools and technologies tailored to the specific project. Essential components include a text editor or integrated development environment (IDE), version control system (e.g., Git), multiple web browsers for cross-browser testing, command-line tools, package managers, database management systems, server environments, front-end frameworks and libraries, back-end technologies, API development and testing tools, content management systems, security tools, deployment and hosting services, and graphic editors for multimedia content. Selecting the appropriate software stack depends on the project's needs, programming languages, and frameworks, with an emphasis on staying current with evolving best practices and technologies in the dynamic field of web development.

**CHAPTER-3**

**PROJECT WORK**

**3.1 PROJECT OVERVIEW**

**Project Overview: Simple Calculator**

A simple calculator is a basic tool used to perform arithmetic calculations. In the context of web development or software development, a simple calculator is typically a digital version of the physical calculators that most people are familiar with. Here's an overview of a simple calculator:

1. User Interface:

A simple calculator features a user-friendly interface designed for ease of use. It typically includes a set of buttons for numbers (0-9), arithmetic operators (+, -, \*, /), and special functions (e.g., equals (=) and clear (C)). The interface may also have a display area to show the numbers and results of calculations.

2. Basic Arithmetic Operations:

A simple calculator can perform basic arithmetic operations, which include:

- Addition (+): Adding two or more numbers together.

- Subtraction (-): Subtracting one number from another.

- Multiplication (\*): Multiplying two or more numbers.

- Division (/): Dividing one number by another.

- Equals (=): Calculating the result of an expression.

3. Order of Operations:

A well-designed calculator follows the order of operations (BODMAS/BIDMAS) to ensure that calculations are performed in the correct sequence. This means that it calculates expressions in the following order: Brackets, Orders (i.e., powers and square roots), Division and Multiplication, and Addition and Subtraction.

4. Clear and Reset:

A clear (C) button is included to reset the calculator's input and start a new calculation. It allows users to clear the current input or results and begin a fresh calculation.

5. Display Area:

The display area is where users can see the numbers they input and the results of calculations. It should be capable of dynamically updating to show the current input and any results.

6. Error Handling:

A simple calculator should have error-handling mechanisms in place to deal with issues like division by zero or invalid expressions. When an error occurs, it should provide clear and informative messages to the user.

7. Keyboard Support (Optional):

Some calculators offer keyboard support, allowing users to input numbers and operators using the keyboard in addition to clicking buttons.

8. Memory Functions (Optional):

More advanced calculators may include memory functions such as Memory Plus (M+), Memory Minus (M-), Memory Recall (MR), and Memory Clear (MC) to store and retrieve numbers for later use.

9. Scientific Calculator Features (Optional):

Some calculators go beyond the basics and include scientific functions, like trigonometric calculations, logarithms, exponentials, and more.

10. Accessibility (Optional):

Accessibility features can be added to make the calculator usable for people with disabilities, including screen readers and keyboard navigation.

Simple calculators are commonly used in various fields, including education, finance, engineering, and everyday life. They are helpful tools for performing quick and accurate calculations, making them essential for many people. In web development, creating a simple calculator is often a beginner's project, allowing them to practice HTML, CSS, and JavaScript skills while building a useful application.

**3.2 ALGORITHM**

The code you provided is an HTML and JavaScript program that creates a simple calculator with a user interface and basic arithmetic functionality. I'll provide you with an algorithm to understand how this code works:

1. Create the HTML structure:

- Define the HTML document structure with `<html>`, `<head>`, and `<body>` elements.

- Include meta tags for character encoding and viewport settings.

- Set the document title.

- Define an external CSS style to style the calculator and output area.

2. Create the calculator user interface:

- Inside the `<body>` element, create a `<div>` with the class "container" to contain the calculator.

- Add a heading with the text "Calculator."

- Create a series of `<div>` elements with the class "a" to group buttons for numbers and operators.

- Each group contains buttons for digits 0-9 and basic arithmetic operators (+, -, \*, /).

- Add special buttons for "00," "0," ".", "%," "=", "X" (backspace), and "AC" (clear).

3. Create the output area:

- Create another `<div>` with the class "ou" to contain the output display.

- Add a heading with the text "Output."

- Insert an `<input>` element with the ID "value" for displaying input and results.

- Initialize the input placeholder with "0."

4. Write JavaScript logic:

- Select the input element and buttons using their IDs and classes.

- Initialize a variable `string` to store the input and calculation string.

- Convert the NodeList of buttons to an array for event handling.

5. Handle button clicks with event listeners:

- Add a click event listener to each button.

- If the button is clicked:

- If it's the "=" button, evaluate the current `string` and display the result in the input field.

- If it's the "AC" button, clear the `string` and reset the input field.

- If it's the "X" button, remove the last character from the `string`.

- For other buttons, append the button's value to the `string` and update the input field accordingly.

6. Style the user interface:

- Use CSS styles to format the container, buttons, and input field to create a visually appealing calculator.

7. Testing and usage:

- Load the HTML page in a web browser and use the calculator interface.

- Enter numbers and operators, perform calculations, and observe the results.

This algorithm provides an overview of how the HTML, CSS, and JavaScript code work together to create a functional calculator with basic arithmetic operations. Users can input expressions, clear the input, evaluate calculations, and backspace to correct input mistakes. The calculator's output is displayed in the input field.

**CHAPTER-4**

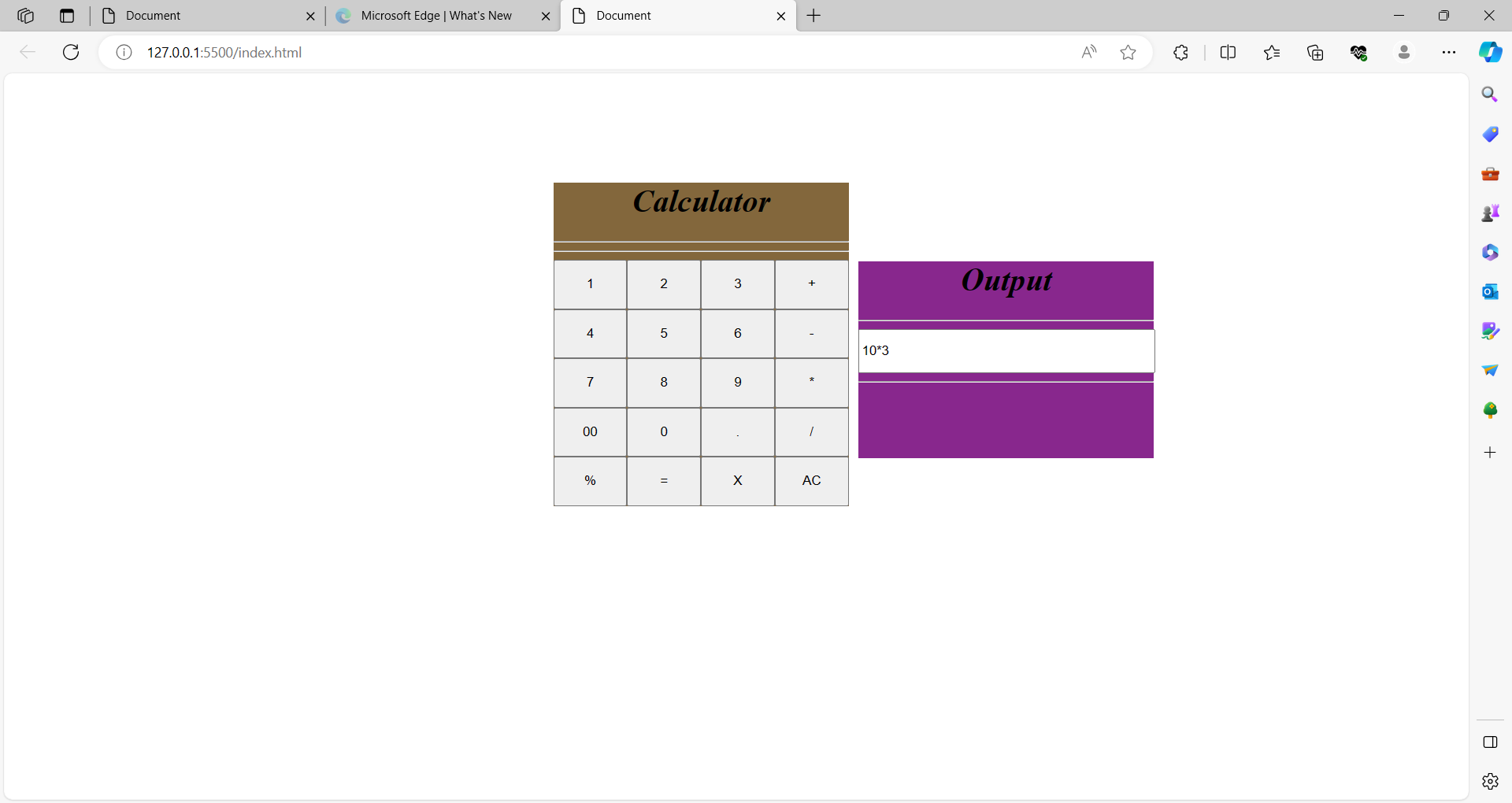
**CODE AND OUTPUT**

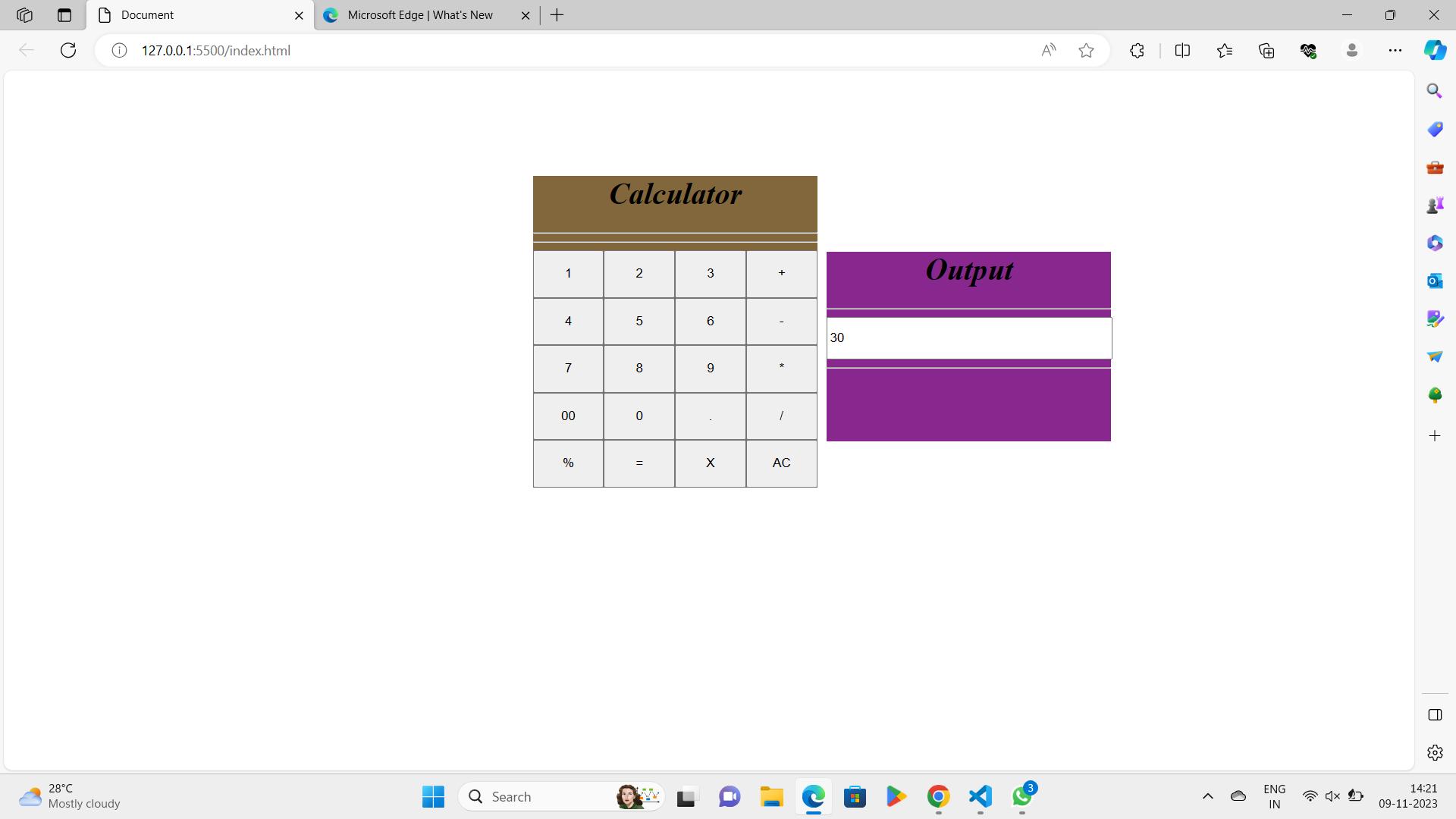
**SCREEN SHOTS**

**Source Code:**

|  |  |
| --- | --- |
|  |  |
|  |  |

**OUTPUT:**





**CHAPTER-5**

**CONCLUSION**

**Conclusion:**

In conclusion, a simple calculator in web development is a basic yet practical project that combines HTML, CSS, and JavaScript to create a user-friendly interface for performing arithmetic calculations. Here are some key takeaways regarding a simple calculator in web development:

**REFERENCES**

**References:**

References in a simple calculator in web development may refer to external resources or libraries that can enhance the functionality or appearance of the calculator. Here are five common types of references in web-based calculators:

1. Web developers can refer to the JavaScript Math object, which provides a wide range of mathematical functions and constants to perform complex calculations in a calculator, such as trigonometric operations or exponential functions.

2. References to CSS (Cascading Style Sheets) can be used to improve the visual design of the calculator, allowing you to apply custom styles and layouts to the calculator's HTML elements to enhance its appearance and user experience.

3. Some developers may choose to include third-party JavaScript libraries or frameworks like jQuery or Bootstrap to simplify the coding process or add pre-built components, such as modals or responsive design features, to their calculator.

4. References to external APIs (Application Programming Interfaces) can be used to extend the functionality of a calculator by accessing data from external sources, such as currency exchange rates or real-time stock prices for financial calculators.

5. Developers can refer to pre-made calculator plugins or widgets that are available in libraries or marketplaces, making it easy to integrate a calculator into a website without having to build one from scratch. These plugins often come with pre-defined features and styles.

The choice of references depends on the specific requirements and goals of the web-based calculator, whether it's for simple arithmetic operations or more complex specialized functions.