PROJECT FINAL REPORT

Project Name	**CALCULATOR**
Description	A user-friendly and efficient calculator application designed to perform basic arithmetic operations, including addition, subtraction, multiplication, and division. It offers a clean and intuitive interface, allowing users to input numbers and receive accurate results instantly. The calculator project aims to provide a reliable and convenient tool for quick mathematical calculations.
Course	WEB TECHNOLOGY-II
Course Code	CA161
Batch	2022
Date	26th May, 2022
Student Name	1. Rahamatulla Mandal 2. Sunil Sahoo
Student Roll Number	1. 2213986040 2. 2213986051
Project Guide Name	**Dr. Neeraj Dev Kaushik**

Project Results

Detailed Project Description
The goal of this project is to create a simple calculator using HTML, CSS, and JavaScript. The calculator will have a user-friendly interface with buttons for digits, arithmetic operations, and a display screen to show the input and result. **Project Steps:** **Step 1:** Set up the HTML Structure, start by creating the basic HTML structure for the calculator. Use <div> tags to create containers for different sections of the calculator, such as the display screen and buttons. Assign appropriate IDs and classes to elements for later use. **Step 2:** Style the Calculator with CSS, Apply CSS styles to the calculator to make it visually appealing. You can set the dimensions, colors, borders, fonts, and positioning as per your preference. Use CSS classes and IDs to target specific elements and apply styles accordingly. Feel free to get creative and customize the look of your calculator. **Step 3:** Implement JavaScript Functionality, The JavaScript code will provide the calculator functionality. Start by declaring variables to store the calculator's state, such as the current input and the calculation result. Then, create functions to handle button clicks and perform the corresponding operations. **Step 4:** Handle Number Buttons, write a JavaScript function to handle number buttons' clicks. When a number button is clicked, append the corresponding digit to the current input displayed on the screen. Keep track of the current input as a string.

Step 5: Handle Operation Buttons, Implement JavaScript functions to handle operation buttons (e.g., addition, subtraction, multiplication, division). When an operation button is clicked, store the selected operation and clear the input. You can use a variable to keep track of the current operation.

Step 6: Perform Calculations, write a JavaScript function to perform calculations when the equals button is clicked. Use the stored operation and the current input to calculate the result. Display the result on the screen and update the current input accordingly.

Step 7: Add Event Listeners, Use JavaScript event listeners to detect button clicks. Attach event listeners to each button element and call the appropriate functions when buttons are clicked. This will enable the calculator to respond to user interactions.

Step 8: Error Handling, consider implementing error handling to handle scenarios such as division by zero or invalid inputs. You can display error messages on the screen or prevent invalid operations from being performed. Validate user input and provide feedback when errors occur.

Step 9: Test and Refine, thoroughly test your calculator by clicking the buttons and performing various calculations. Verify that the results are accurate and that error handling is working as expected. Make any necessary adjustments to the HTML, CSS, and JavaScript code to improve the functionality and user experience.

Step 10: Documentation and Finalization, document your code by adding comments to make it easier to understand and maintain. Provide clear instructions on how to use the calculator. Review the project, refine any remaining issues, and make sure all the requirements are met. Once satisfied, you can consider deploying the calculator on a web server or sharing it with others.

Conclusion: By following the steps outlined above, you can create a simple calculator using HTML, CSS, and JavaScript. This project will provide you with hands-on experience in web development, JavaScript programming, and user interface design.

Project Successes includes

Name	Description
Functionality	The calculator successfully performs basic arithmetic operations such as addition, subtraction, multiplication, and division. It accurately calculates results based on user input.
User-Friendly Interface	The calculator has a clean and intuitive interface that is easy to understand and use. The buttons and display screen are visually appealing and responsive to user interactions.
Responsive Design	The calculator is designed to be responsive and adaptable to different screen sizes and devices. It maintains its functionality and usability across desktops, laptops, tablets, and mobile devices.

Project Challenges

Description	Impact	Actions Taken
Logic and Algorithm	Incorrect calculation results, unexpected behavior.	Conducted thorough research on mathematical operations and algorithms. Implemented step-by-step testing and debugging processes to identify and fix logic errors. Validated the calculator's accuracy using test cases and edge scenarios.

Error Handling	Incorrect or misleading error messages, improper calculation behavior.	Implemented comprehensive error handling logic to detect and handle common errors. Developed error messages that clearly inform the user about the issue. Conducted extensive testing to validate the error handling mechanism and refine it based on user feedback.
User Interface Design	Confusing or cluttered interface, inconsistent display across devices.	Conducted research on user interface design principles and best practices. Used responsive design techniques to ensure the calculator adapts well to different screen sizes. Tested the calculator on multiple browsers and devices to identify and fix any display or compatibility issues.

Lessons Learned

Description
Thorough Planning: Proper planning and defining clear requirements upfront help in identifying potential challenges and developing a roadmap for the project. Taking the time to plan the project's structure, functionality, and user interface design can save time and effort in the long run.
Incremental Development: Breaking down the project into smaller tasks and implementing features incrementally allows for easier troubleshooting and debugging. It also enables early validation of core functionalities before moving on to more complex components.
Continuous Learning: Web development is an ever-evolving field, and embracing a mindset of continuous learning is crucial. Staying updated with the latest technologies, best practices, and design trends enables you to improve your skills and deliver higher-quality projects.

Project Performance

Meeting Expectations		
Success Criteria	**Criteria Met**	**Comments**
Accurate Calculations	☐	The calculator has been thoroughly tested with various input scenarios and edge cases to ensure accurate calculations. It meets the expectations in terms of accuracy.
User-Friendly Design	☐	The user interface has been designed with a focus on usability and clarity. Buttons and functions are labeled appropriately, and the layout is organized to facilitate smooth user interactions.
Documentation and Instructions	☐	The code has been documented with comments explaining its functionality and structure.

Transition to Operations and Archives	
Documentation	Ensure that all project documentation, including design specifications, user manuals, and technical documentation, is complete and up to date. This documentation will be valuable for future reference and maintenance. Describe the activities / deliverables to be transitioned to the operational team (i.e. knowledge transfer).
Deployment	Prepare the necessary files and configurations for deploying the calculator to a production environment, if applicable. This may involve setting up a web server or hosting platform and configuring any required dependencies. Describe what project archives have been created and where are they located. What documents have been archived? How can it be used on future projects?

Screenshots