

FAC-QuantWeb

Assignment 3

May 23, 2024

Instructions

- Submission deadline is 26th EOD.
- You are free to use HTML, CSS, JavaScript as well as Bootstrap classes to the best of your ability and knowledge.
- Make sure you focus on the layout and presentation of the website, however do not compromise on the functionalities required.
- The submission would be on GitHub classroom as stated in the meet. Submit all the required files in a **single repository**. Further, also make sure to include a zipped file of a **video** showing all the functionalities implemented in your website from start to end.
- In case of any query please feel free to contact the mentors.

The assignment is focused on making you familiar with the basics of front-end development, following which we might also choose 5-6 mentees with high quality submissions to focus more on the front-end part of the end project.

The following are the points to be kept in mind for this assignment.

- **Objective:**

Create a fully functional calculator website that performs both arithmetic operations (addition, subtraction, multiplication, division), as well as advanced functions such as square root, exponentiation, natural logarithm, sin, cosine and tangent functions. The website should have a clean and user-friendly interface designed using HTML, CSS, JavaScript, and Bootstrap templates.

- **Home page**

- Create an attractive home page for the website with button options for Login and Sign-Up, redirecting the user to the respective pages.

- Additionally, to make the website more informative, you may include your (developer's) contact information (name, image, contact, email, etc.), and a basic image based info/short description of the working of the calculator website showing the user what's in store for them.
- **Sign-Up/ Register page**
 - Create a form with input fields for username, email, and password.
 - Add a "Sign Up" button to submit the form.
 - Include a link to the login page for users who already have an account.
 - Add JavaScript based client-side validation to ensure all fields are filled correctly.
 - Validate email format and ensure the password meets the required criteria.
- **Login page**
 - Create a form with input fields for email and password.
 - Add a "Log In" button to submit the form.
 - Include a link to the signup page for new users.
 - Validate the email and password entered by the user.
 - If valid, redirect the user to the main scheduler page.
- **Calculator layout design:**
 - **HTML Structure:**
 - * Create a basic structure for the calculator using HTML.
 - * You may use Bootstrap components (buttons, grid system) to layout the calculator buttons and display screen.
 - **CSS Styling:**
 - * Style the calculator to have a clean, modern look.
 - * Ensure the design is responsive and works well on different screen sizes.
 - **Display Screen Implementation:**
 - * Implement the display screen where the numbers and results are shown.
 - * Ensure the display is styled appropriately and is readable.
 - **Button Implementation:**
 - * Add buttons for digits (0-9), operations (+, -, *, /, log, sqrt, exp, sin, cos, tan), and special functions such as clear, equals, m+, m-, etc. (need not implement m+, m- though).
 - * You may use Bootstrap buttons for a consistent look and feel.

- **Calculator and its functionalities:**

- Implement JavaScript functions to handle all the required operations:
 - * Addition
 - * Subtraction
 - * Multiplication
 - * Division
 - * Square Root
 - * Exponentiation
 - * Logarithm
 - * Sin, Cos, Tan
- Ensure calculations are accurate and edge cases (e.g., division by zero) are handled properly by well defined error messages or display appropriations.

- **Button Functionality and Event Handling:**

- Add event listeners to buttons for capturing user input.
- Implement the logic to update the display screen based on user interactions.
- Ensure that clicking on buttons performs the correct operation and updates the display.

- **Clear and Reset Functionality:**

- Implement a clear button to reset the display and clear the current calculation.
- Ensure that all temporary states and variables are reset appropriately.

- **Styling and Enhancements:**

- Customize the look and feel using Bootstrap's utility classes and custom CSS.
- Add responsive design features to ensure usability on various devices.
- Include icons and animations (shadows, texture, etc.) to improve user experience.