PROJECT REPORT

Title: Web-page

Submitted by: Sneha Chaudhary

Date: 26/10/2024

---

1. Introduction

The project involves creating a personal website for Udita Singh. This website serves as an online portfolio to introduce visitors to Udita's background, education, skills, and contact information. Designed with HTML and CSS, the website provides a user-friendly and responsive layout.

2. Objectives

- Develop a personal webpage to showcase Udita Singh's background, education, and skills.
- Implement a visually appealing and well-organized layout using HTML and CSS.
- Provide clear sections for "About Me," "Skills," and "Contact Me."

3. Technologies Used

- **HTML**: Structure the content and create sections for About, Skills, and Contact.
- **CSS**: Style the page for visual appeal, including fonts, colors, and layout alignment.

4. System Design

4.1 Functional Requirements

The webpage should contain:

1. **Header**: Displaying the name and a welcome message.
2. **About Me Section**: Brief description of the background, education, and purpose of the website.

3. **Skills Section**: A list of skills (HTML, CSS, JavaScript, C++, Python).
4. **Contact Section**: Email link for communication.

4.2 Non-Functional Requirements

- **Usability**: Should be easily navigable, with each section clearly identifiable.
- **Visual Appeal**: Use colors and font styling to maintain a professional, consistent look.
- **Responsive Design**: Page should adjust gracefully on different screen sizes.

5. Implementation

The website structure is achieved using HTML for layout and CSS for styling.

5.1 Code Structure

1. **HTML Layout (index.html)**:
    - header contains the website title and welcome message.
    - section elements for "About Me," "Skills," and "Contact Me."
    - div container for wrapping and centering main content.
2. **CSS Styling**:
    - Global styling for background color, font family, and color.
    - Custom styles for header, section titles, skill list, and footer.
    - Styled buttons, borders, and colors to create a cohesive design.

5.2 Explanation

- **Header**: Displays the user's name and a welcome message, styled with a background color to stand out.
- **About Me Section**: Contains a brief introduction and educational background of Udita Singh, explaining the purpose of the website.
- **Skills Section**: A list of relevant skills displayed as styled list items.
- **Contact Section**: Contains an email link for direct contact.

6. Testing and Validation

The webpage was tested on various devices and browsers to ensure:

1. **Compatibility**: Works well on popular browsers (Chrome, Firefox, Safari).
2. **Responsiveness**: Adjusts layout for different screen sizes.
3. **Functionality**: All links are working, and content is displayed as intended.

Test Cases :

Component	Test Description	Expected Outcome	Status
Header	Name and welcome message	Properly displayed	Pass
About Me Section	Bio details	Information displayed correctly	Pass
Skills Section	Skill list items	Properly styled list items	Pass
Contact Section	Email link	Clickable and opens email client	Pass

7. Conclusion

This personal website successfully achieves its goal of showcasing Udita Singh's profile. It provides an organized layout with a clean design and responsive functionality, meeting the objectives of usability and aesthetic appeal. Future improvements could include additional sections like “Projects” or a “Portfolio” and enhancing the design with animations or interactive elements.

Submitted by: Sneha Chaudhary

PROJECT REPORT

Title: Simple Calculator

Submitted by: Sneha Chaudhary

Date: 26/10/2024

---

1. Introduction

This project involves creating a simple calculator using HTML, CSS, and JavaScript. The calculator performs basic arithmetic operations such as addition, subtraction, multiplication, division, and modulus. The calculator's intuitive layout and functions allow users to quickly perform calculations.

2. Objectives

- Build a functional calculator that performs basic mathematical operations.
- Implement an interactive user interface for a smooth user experience.
- Use JavaScript to handle real-time calculations and display results.

3. Technologies Used

- **HTML**: To structure the calculator's layout.
- **CSS**: To style the calculator, making it visually appealing and user-friendly.
- **JavaScript**: To add functionality to the calculator buttons and to process the mathematical operations.

4. System Design

4.1 Functional Requirements

The calculator should support:

1. **Addition**
2. **Subtraction**
3. **Multiplication**
4. **Division**
5. **Modulus Operation**

6. **Clear Display** (C)
7. **Delete Last Entry** (DEL)
8. **Decimal Point Input**

4.2 Non-Functional Requirements

- **Usability**: Should be easy to use and have a clear display area.
- **Responsiveness**: Buttons and inputs should respond to user clicks immediately.

5. Implementation

The implementation uses HTML for layout, CSS for styling, and JavaScript for functionality.

5.1 Code Structure

1. **HTML Layout (index.html)**: Contains the structure and defines the calculator display and buttons. The input field shows the current calculation, while button elements are used for each digit and operator.
2. **CSS Styling (style.css)**: Styles the calculator layout, making it centered, adding button styling, and making the display more readable.
3. **JavaScript Logic (script.js)**: Contains functions to handle button clicks, append numbers and operators to the display, calculate results, and clear or delete entries.

5.2 JavaScript Functions

The JavaScript file (script.js) includes:

- **appendNumber(number)**: Adds a selected number to the display.
- **appendOperator(operator)**: Adds an operator to the display.
- **calculate()**: Evaluates the current expression and displays the result.
- **clear_all()**: Clears the display completely.
- **delete_last()**: Deletes the last character entered in the display.

6. Code Explanation

HTML Structure (index.html)

The HTML file defines a container div for the calculator:

- <input type="text" id="display" disabled>: Display area to show the numbers and results.
- <button> elements with onclick attributes to call the JavaScript functions for each button press.

JavaScript Functions (script.js)

JavaScript functions handle the logic of the calculator:

- **clear_all()**: Resets the display to an empty string.
- **delete_last()**: Removes the last character from the display.
- **appendNumber()** and **appendOperator()**: Append a number or operator to the display.
- **calculate()**: Uses JavaScript's eval() function to compute the result and displays it.

7. Testing and Validation

The calculator was tested to ensure:

1. Correct results for all supported operations.
2. Proper handling of decimals.
3. Display updates immediately for every button press.

Test Cases :

Input Expression	Expected Output	Actual Output	Status
5 + 3	8	8	Pass
10 / 2	5	5	Pass
6 * 2 - 3	9	9	Pass
12 % 5	2	2	Pass
8 / 0	Error/Infinity	Error/Infinity	Pass

8. Conclusion

The Simple Calculator project successfully achieved the goal of creating a functional calculator that performs essential arithmetic operations. The calculator is designed to be user-friendly and efficient, offering basic functionalities in a responsive layout. Future improvements could include more complex operations, scientific functions, or enhanced styling for a more modern look.

Submitted by: Sneha Chaudhary