**PAKHI SHARMA**

**V – B**

**ROLL NO. – 2200290100113**

**INTERNSHIP PROJECT REPORT**

**Introduction**

This report outlines two web applications developed using HTML, CSS, and JavaScript: a To-Do List and a Simple Calculator. Both applications demonstrate fundamental programming concepts, user interface design, and basic interactivity using web technologies.

## Project 1: To-Do List Application

### Overview

The To-Do List application allows users to add and remove tasks. It provides a simple and user-friendly interface for task management, helping users organize their daily activities.

### Features

1. **Add Tasks**: Users can input tasks into a text field and add them to the list by clicking an "Add" button or pressing the "Enter" key.
2. **Remove Tasks**: Each task has a "Remove" button, enabling users to delete tasks from the list.
3. **User Input Handling**: The application validates user input to prevent adding empty tasks.

### Implementation

* **HTML**: The structure includes a title, input field, button, and an unordered list to display tasks.
* **CSS**: Styles enhance the visual appearance, including layout, colors, and hover effects for buttons.
* **JavaScript**: The logic for adding and removing tasks is handled via event listeners and DOM manipulation.

Code Snippet :

addButton.addEventListener('click', () => {

const taskText = taskInput.value.trim();

if (taskText === '') return;

const listItem = document.createElement('li');

listItem.textContent = taskText;

const removeButton = document.createElement('button');

removeButton.textContent = 'Remove';

removeButton.addEventListener('click', () => {

taskList.removeChild(listItem);

});

listItem.appendChild(removeButton);

taskList.appendChild(listItem);

taskInput.value = '';

});

### User Interface

The UI consists of an input field at the top for adding tasks, followed by a button and a dynamically populated list. The design is clean and intuitive, making it easy for users to interact with the application.

## Project 2: Simple Calculator Application

### Overview

The Simple Calculator application performs basic arithmetic operations, allowing users to input numbers and operators to calculate results.

### Features

1. **Basic Operations**: Supports addition, subtraction, multiplication, and division.
2. **Clear Functionality**: A "C" button resets the input field.
3. **Result Calculation**: Displays the result of the arithmetic operation.

### Implementation

* **HTML**: The layout includes an input field for displaying calculations and buttons for digits and operations.
* **CSS**: Styling is applied to create a visually appealing and responsive design, with distinct button styles for interactivity.
* **JavaScript**: The calculator logic uses the eval function to compute results based on user input.

### Code Snippet

javascript

Copy code

function calculateResult() {

const display = document.getElementById('display');

try {

display.value = eval(display.value);

} catch (error) {

display.value = 'Error';

}

}

### User Interface

The calculator features a grid layout for buttons, making it easy to use on various devices. The input field displays ongoing calculations, while buttons are clearly labeled for ease of use.

## Conclusion

Both projects serve as practical examples of fundamental web development skills, including HTML structuring, CSS styling, and JavaScript functionality. The To-Do List application emphasizes task management, while the Calculator application focuses on arithmetic operations. These projects can be further enhanced with additional features, improved error handling, or more sophisticated designs, making them valuable learning tools for web development.