NAME : YOGITA

ROLL NO. : 2300290120291

YEAR : 2nd

OBJECTIVE : MLSA Internship Project Report

PROJECT REPORT : SIMPLE TO DO LIST

1. Overview

This report outlines the development of a basic to-do list web application using HTML, CSS, and JavaScript. The application allows users to create, manage, and mark tasks as completed

2. Technologies Used

- HTML: Structures the web page and its content.
- CSS: Defines the visual appearance of the page.
- JavaScript: Adds dynamic and interactive functionality.

3. Key Features

1. **Task Input Field** : Users can input a task in the text field
2. **Add Task Button** : There's a button that users can click to add the task from the input field to the task list.
3. **Task List Display** : The tasks entered by the user are displayed in an unordered list
4. **Dynamic Task Addition** : When the "Add" button is clicked, the task is added to the list without needing to reload the page. The page dynamically updates with JavaScript.
5. **Clear Input Field After Adding :** After a task is added to the list, the input field is automatically cleared, making it ready for the next task.

4. Code Breakdown

- **HTML**
  - Defines the page layout, including:
  - A container to hold the application’s content.
  - Elements such as an input field, add task button, and a task list.

- A link to the CSS file for styling.

❖ **CSS**

- Styles the page, focusing on:
- The background, fonts, and layout.
- The design of the input field and buttons.
- Task list formatting, including completed task styles (strikethrough and opacity adjustments).
- Styling of the delete button for task removal.

❖ **JavaScript**

- Runs after the page loads, utilizing the DOMContentLoaded event.
- Retrieves HTML elements (input field, button, task list) for interaction.
- Loads tasks from local storage if they exist.
- *Contains functions for:*
  - Rendering the task list from local storage.
  - Clearing the input field after a task is added
  - Marking tasks as completed when clicked.

5. Conclusion

This web-based to-do list app demonstrates how HTML, CSS, and JavaScript can be used together to build a simple and interactive tool for managing tasks.

PROJECT REPORT : BASIC CALCULATOR

1. OVERVIEW

The Calculator Web Application is a straightforward, interactive tool designed to perform basic arithmetic operations, including addition, subtraction, multiplication, and division. This project exemplifies the integration of HTML, CSS, and JavaScript to create a functional and user-friendly calculator.

2. Technologies Used

- HTML: Structures the layout and content of the calculator.
- CSS: Styles the visual appearance of the application.
- JavaScript: Implements interactive functionality for calculations.

3. Key Features

1. Basic Arithmetic Operations : Users can perform addition, subtraction, multiplication, and division using operator buttons.
2. Input Display : A dedicated display field shows the current input and results, ensuring a clear view of the calculations.
3. Clear Functionality : Users can reset the display using a clear button, allowing for new calculations without manual erasing.
4. Responsive Design : The application is designed to be user-friendly on different screen sizes, ensuring accessibility across devices.

4. Code Breakdown

- **HTML**
  - Defines the structure of the calculator, including:
  - A container for the calculator.
  - An input field for displaying calculations, set to read-only for user input.
  - Buttons for numbers and operations with associated onclick events to handle user interactions.
- **CSS**
  - Styles the calculator's layout and appearance, focusing on:
  - Background colors, fonts, and button designs.
  - Display field styling to enhance readability.
  - Responsive adjustments to ensure usability across various devices.
- **JavaScript**
  - Executes when the page loads and includes:
  - Functions for appending values to the display.
  - A calculation function that evaluates expressions and updates the display with results.
  - A clear function to reset the display.
  - Ensures proper handling of arithmetic operations and user input.

5. Conclusion

This simple calculator web application showcases the integration of HTML, CSS, and JavaScript to create a functional and interactive tool for performing basic arithmetic calculations.