School of Computer Engineering & Technology
Class: Third Year B.Tech CSE (Semester V)
Course: Full Stack Development

GIT Repo URL: https://github.com/prateekp1304/FSD

Roll No: 08
Name: Prateek Patel
PRN: 1032210532

FSD Laboratory 05

Aim: Design and develop an interactive user interface using React.

Objectives:
1. Articulate what React is and why it is useful.
2. Explore the basic architecture of a React application.
3. Use React components to build interactive interfaces

Theory:

1. What is React? Steps to run React app using create-react-app.
React:
React is a JavaScript library for building user interfaces. It allows developers to create reusable UI components and efficiently update and render them as the data changes. React is commonly used for building single-page applications where the content is dynamically updated without requiring a full page reload.

Steps to run a React app using create-react-app:
- Install Node.js: Ensure that Node.js is installed on your machine. You can download it from https://nodejs.org/.
- Install create-react-app: Open your terminal and run the following command to install create-react-app globally
- Create a new React app: Run the following command to create a new React app using create-react-app
- Navigate to the app directory
- Run the app: Start the development server by running

2. Passing data through props (Small example)
In React, props (short for properties) are used to pass data from a parent component to a child component. Here's a small example:

```
// ParentComponent.js
import React from 'react';
import ChildComponent from './ChildComponent';

const ParentComponent = () => {
  const dataToPass = 'Hello from Parent!';

  return (
    <div>
      <h1>Parent Component</h1>
      <ChildComponent passedData={dataToPass} />
    </div>
  );
};

export default ParentComponent;

// ChildComponent.js
import React from 'react';

const ChildComponent = (props) => {
  return (
    <div>
      <h2>Child Component</h2>
      <p>Data received from parent: {props.passedData}</p>
    </div>
  );
};

export default ChildComponent;
```

FAQ:

1. What are React states and hooks?

Ans:

React States:

In React, state refers to the data that a component manages. It represents the current condition or values within a component and can change over time in response to user actions, network responses, or other events. States are used to keep track of dynamic information in a component.

To use state in a functional component, you can use the useState hook. The useState hook is a function that returns an array with two elements: the current state value and a function that lets you update it.

React Hooks:

Hooks are functions that let you use state and other React features in functional components. Before the introduction of hooks, state and lifecycle methods were only available in class components. With hooks, functional components can have local state, lifecycle methods, and more.

Some commonly used hooks include:

useState: Used to declare state variables in functional components.

useEffect: Used for side effects in functional components, such as data fetching, subscriptions, or manually changing the DOM.

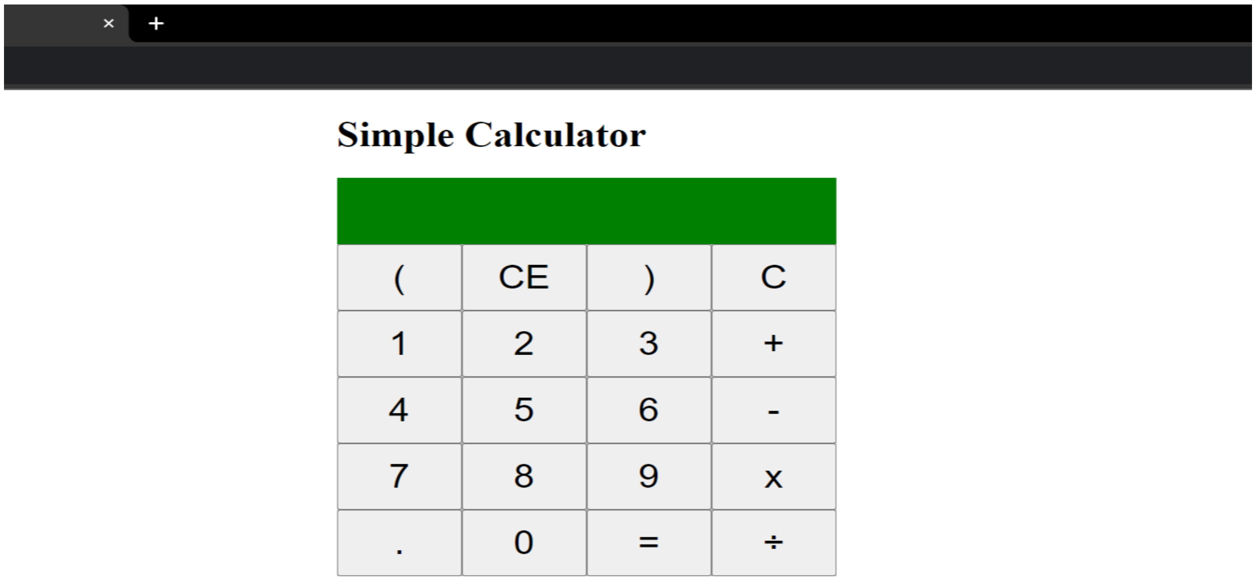
useContext: Used to subscribe to React context without introducing nesting.

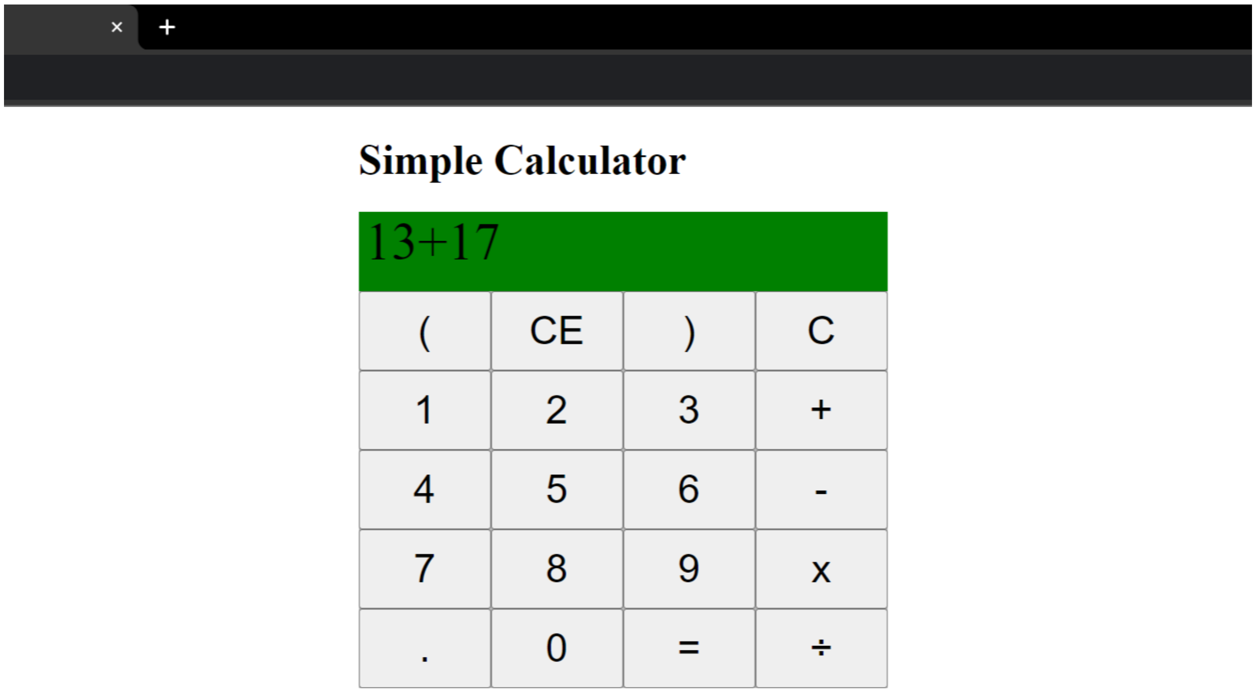
useReducer: A more advanced hook that is often used for managing more complex state logic.

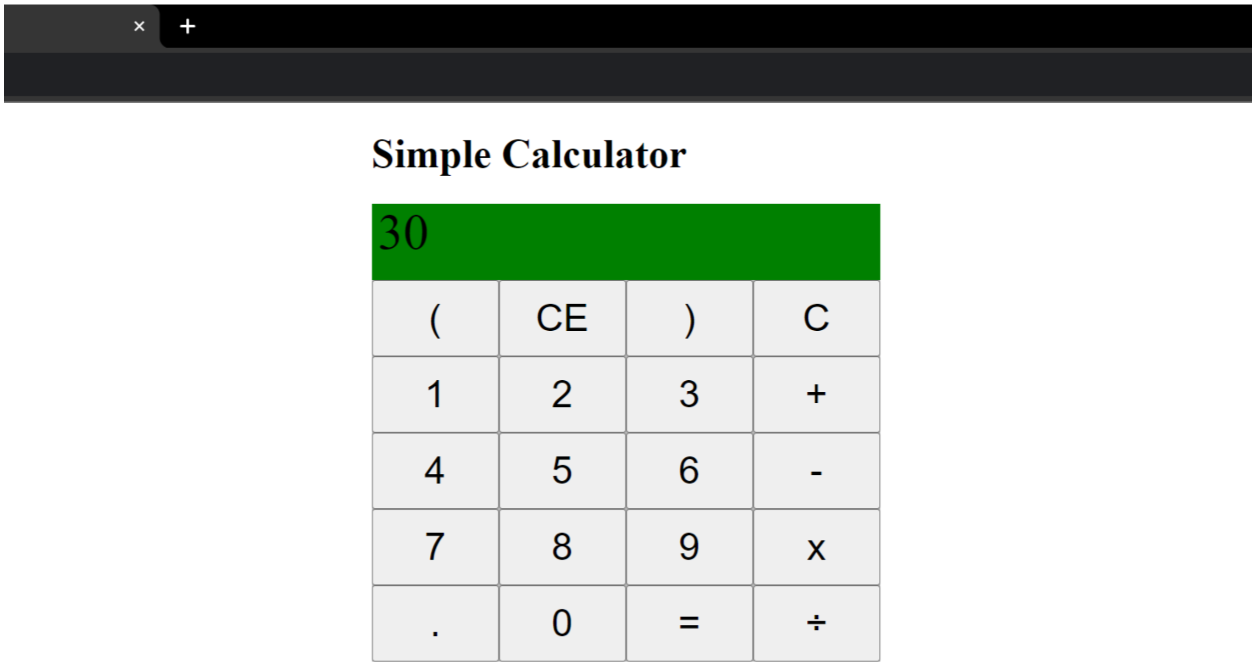
useCallback and useMemo: Used for performance optimization by memoizing functions or values.

Output: Screenshots of the output to be attached.

Output Screenshot:




Simple Calculator
13+17
(
CE
)
C
1
2
3
+
4
5
6
-
7
8
9
x
.
0
=
÷


Simple Calculator
30
(
CE
)
C
1
2
3
+
4
5
6
-
7
8
9
x
.
0
=
÷

Sample Problem Statements:

1. Design and develop a UI for calculator using React.

You can include features like-

Mathematical operations

Inserting values

Decimal values must be supported

2. Design and develop a UI for creating React form to gather the individual's data.

It should include features like-

Taking user's data.

A SUBMIT button.

On submit must highlight the empty block.

Error message when entering the wrong format of email ID.

Display a success message on successful submission.

3. Design and develop a UI for Resume Builder application system.

The resume builder app must include -

Professional summary

Education qualifications

Academic and non-academic skills,

Career objective

Experience and Internships

Skills and Achievements