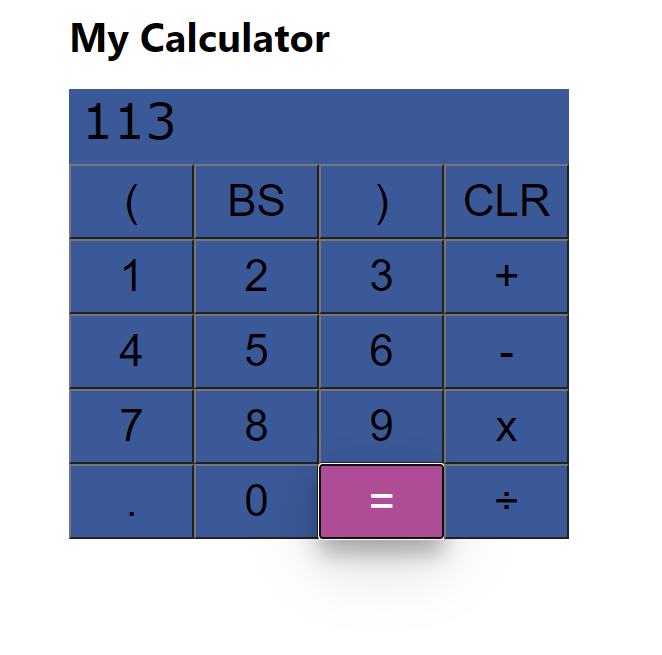
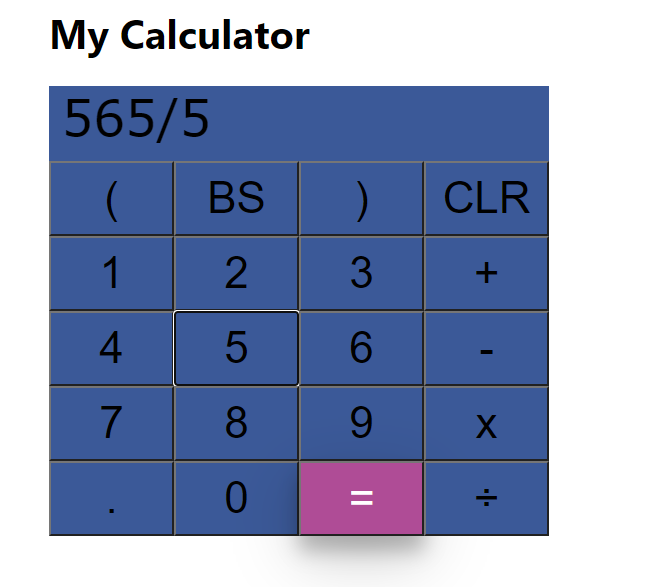
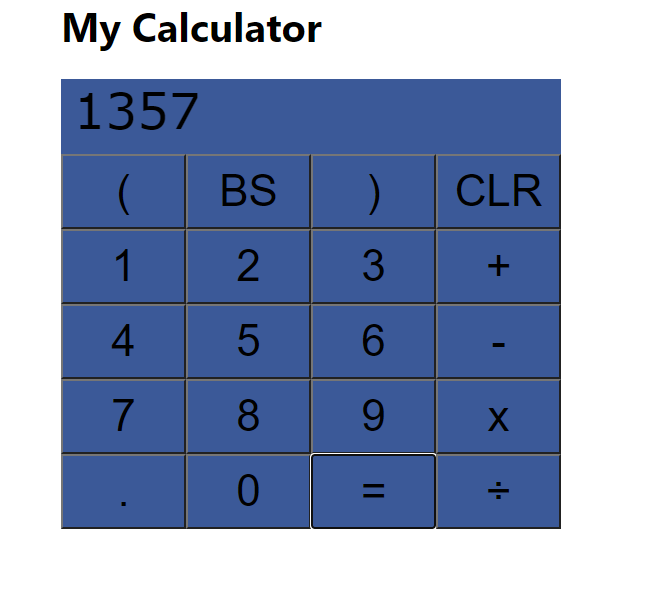
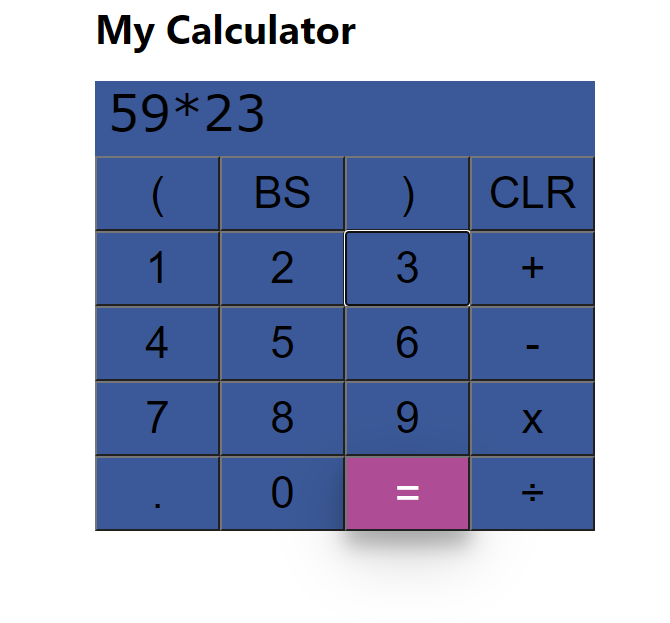
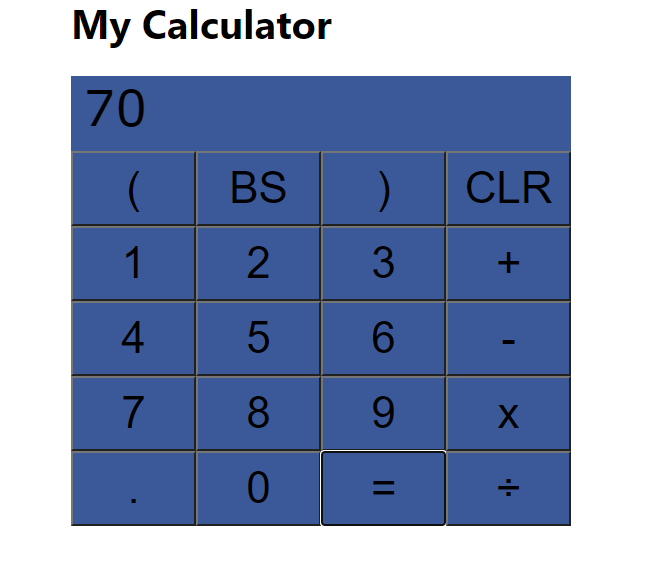
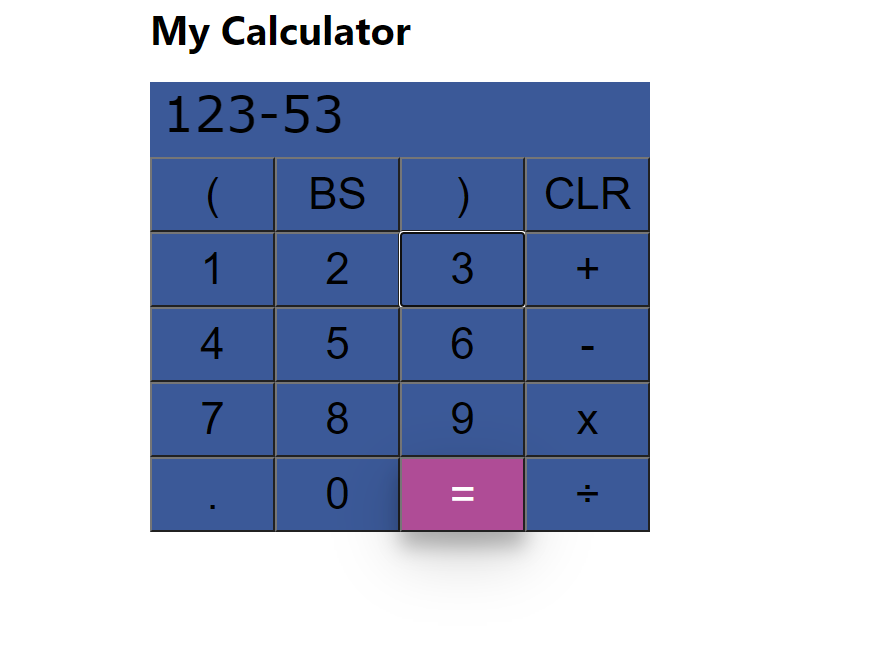
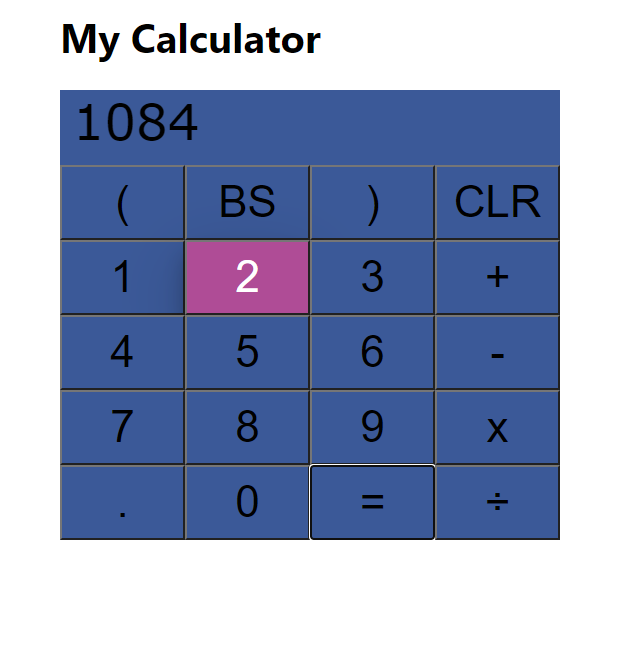
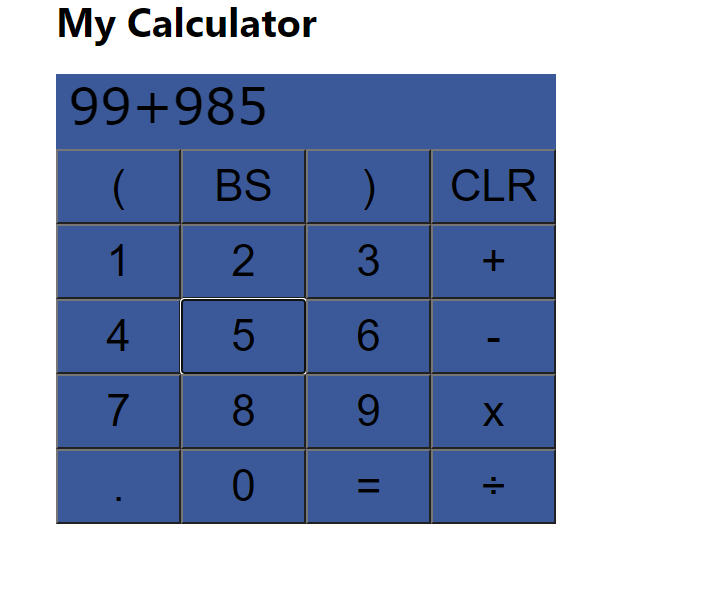
UI DESIGN DOCUMENT

Recorded output video Drive link: https://drive.google.com/file/d/1P3gK3WGPnkMATeYe--CuU52b1N9OMnvp/view?usp=sharing

SOURCE CODE GITHUB REPOSITORY LINK

<https://github.com/HariPrasad5724/calculatorreact>

This calculator application has two components

1.ResultTextComponent.js - to display the result of operation

ResultTextComponent.js

import React, {Component} from 'react';

class ButtonComponent extends Component {

render() {

return (

<div className="button">

<button name="(" onClick={e => this.props.onClick(e.target.name)}>(</button>

<button name="BS" onClick={e => this.props.onClick(e.target.name)}>BS</button>

<button name=")" onClick={e => this.props.onClick(e.target.name)}>)</button>

<button name="CLR" onClick={e => this.props.onClick(e.target.name)}>CLR</button><br/>

<button name="1" onClick={e => this.props.onClick(e.target.name)}>1</button>

<button name="2" onClick={e => this.props.onClick(e.target.name)}>2</button>

<button name="3" onClick={e => this.props.onClick(e.target.name)}>3</button>

<button name="+" onClick={e => this.props.onClick(e.target.name)}>+</button><br/>

<button name="4" onClick={e => this.props.onClick(e.target.name)}>4</button>

<button name="5" onClick={e => this.props.onClick(e.target.name)}>5</button>

<button name="6" onClick={e => this.props.onClick(e.target.name)}>6</button>

<button name="-" onClick={e => this.props.onClick(e.target.name)}>-</button><br/>

<button name="7" onClick={e => this.props.onClick(e.target.name)}>7</button>

<button name="8" onClick={e => this.props.onClick(e.target.name)}>8</button>

<button name="9" onClick={e => this.props.onClick(e.target.name)}>9</button>

<button name="\*" onClick={e => this.props.onClick(e.target.name)}>x</button><br/>

<button name="." onClick={e => this.props.onClick(e.target.name)}>.</button>

<button name="0" onClick={e => this.props.onClick(e.target.name)}>0</button>

<button name="=" onClick={e => this.props.onClick(e.target.name)}>=</button>

<button name="/" onClick={e => this.props.onClick(e.target.name)}>÷</button><br/>

</div>

);

}

}

export default ButtonComponent;

2.ButtonComponent.js - to display the buttons as keypad design.

ResultTextComponent.js

import React, {Component} from 'react';

class ResultComponent extends Component {

render() {

let {result} = this.props;

return (

<div className="result">

<p>{result}</p>

</div>

);

}

}

export default ResultComponent;

App.css - to design our components

App.css

.App {

text-align: center;

}

.result {

height: 70px;

background-color: #3b5998;

width: 100%;

}

.result p {

font-size: 40px;

margin: 10px;

border: 1px solid red;

font-family: Verdana, Geneva, Tahoma, sans-serif;

}

.calculator-body {

max-width: 400px;

margin: auto;

}

.button {

display: block;

transition-duration: 0.7s;

}

button {

width: 25%;

height: 60px;

font-size: 35px;

background-color: #3b5998;

}

button:hover {

background-color: #af4c96;

color: white;

box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);

}

button:active{

transform: translateY(4px);

}

App.js - to display by combining the components and rendering the webpage

App.js

import React, { Component } from 'react';

import './App.css';

import ResultComponent from './components/ResultTextComponent';

import ButtonComponent from "./components/ButtonComponent";

class App extends Component {

constructor(){

super();

this.state = {

result: ""

}

}

onClick = button => {

if(button === "="){

this.calculate()

}

else if(button === "CLR"){

this.reset()

}

else if(button === "BS"){

this.backspace()

}

else {

this.setState({

result: this.state.result + button

})

}

};

calculate = () => {

var checkResult = ''

if(this.state.result.includes('--')){

checkResult = this.state.result.replace('--','+')

}

else {

checkResult = this.state.result

}

try {

this.setState({

// eslint-disable-next-line

result: (eval(checkResult) || "" ) + ""

})

} catch (e) {

this.setState({

result: "error"

})

}

};

reset = () => {

this.setState({

result: ""

})

};

backspace = () => {

this.setState({

result: this.state.result.slice(0, -1)

})

};

render() {

return (

<div className="calculator-body">

<h1>My Calculator</h1>

<ResultComponent result={this.state.result}/>

<ButtonComponent onClick={this.onClick}/>

</div>

);

}

}

export default App;

Detailed Documentation:

In Eval() function in JS if the argument is an expression, eval() evaluates the expression.

Eg., eval(“55+7”) returns 62 as the result

Reset and BackSpace are the other modules which were implemented.