**Week 9 lab task named as assignment 2:-**

**Task 1 :   
You are provided with two files: Calculator.js and Button.js, along with a CSS file Calculator.css. Your task is to create a React application and integrate these files to build a functional calculator.  
In your explanation, outline the steps you took to develop the application. Furthermore, provide a comprehensive breakdown of the code used in both the Calculator.js and Button.js files.**

**Ans:- To create react application we should write following command on terminal:- (npx create-react-app week)**

**Calculator.js code**

**import "./Calculator.css";**

**import Button from "./Button.js";**

**import { useState } from "react";**

**function KeyPadComponent() {**

**const [text1, setText] = useState("");**

**const [showImage, setShowImage] = useState(false); // State for "Show Me"**

**const handleSquare = () => {**

**const number = parseFloat(text1);**

**if (!isNaN(number)) {**

**setText(number \* number); // Calculate square**

**} else {**

**setText("Invalid Input");**

**}**

**};**

**const ClickHandle = (e) => {**

**if (e.target.value === "C") {**

**setText("");**

**} else if (e.target.value === "=") {**

**try {**

**setText(eval(text1)); // Evaluate the expression**

**alert(eval(text1));**

**} catch {**

**setText("Error");**

**}**

**} else {**

**setText(text1 + e.target.value);**

**}**

**};**

**const handleShowMe = () => {**

**setShowImage(!showImage); // Toggle image visibility**

**};**

**return (**

**<div className="Calculator">**

**<div className="screen-row">**

**<input type="text" readOnly value={text1} />**

**</div>**

**<div>**

**<Button label="(" ClickHandle={ClickHandle} />**

**<Button label="CE" ClickHandle={ClickHandle} />**

**<Button label=")" ClickHandle={ClickHandle} />**

**<Button label="C" ClickHandle={ClickHandle} />**

**</div>**

**<div>**

**<Button label="1" ClickHandle={ClickHandle} />**

**<Button label="2" ClickHandle={ClickHandle} />**

**<Button label="3" ClickHandle={ClickHandle} />**

**<Button label="+" ClickHandle={ClickHandle} />**

**</div>**

**<div>**

**<Button label="4" ClickHandle={ClickHandle} />**

**<Button label="5" ClickHandle={ClickHandle} />**

**<Button label="6" ClickHandle={ClickHandle} />**

**<Button label="-" ClickHandle={ClickHandle} />**

**</div>**

**<div>**

**<Button label="7" ClickHandle={ClickHandle} />**

**<Button label="8" ClickHandle={ClickHandle} />**

**<Button label="9" ClickHandle={ClickHandle} />**

**<Button label="\*" ClickHandle={ClickHandle} />**

**</div>**

**<div>**

**<Button label="." ClickHandle={ClickHandle} />**

**<Button label="0" ClickHandle={ClickHandle} />**

**<Button label="=" ClickHandle={ClickHandle} />**

**<Button label="/" ClickHandle={ClickHandle} />**

**</div>**

**{/\* Add Show Me Button \*/}**

**<div className="show-me">**

**<Button label="Show Me" ClickHandle={handleShowMe} />**

**<Button label="Square" ClickHandle={handleSquare} />**

**</div>**

**{/\* Conditional Rendering of Image \*/}**

**{showImage && (**

**<div className="ImageContainer">**

**<img src="funnyimage.jpg" alt="" className="ImageStyle" />**

**</div>**

**)}**

**</div>**

**);**

**}**

**export default KeyPadComponent;**

**Button.js code**

**import "./Calculator.css";**

**import React from "react";**

**// Create our Button component as a functional component.**

**const Button = (props) => {**

**return (**

**<button class="ButtonStyle" value={props.label} onClick={props.ClickHandle}>**

**{" "}**

**{props.label}{" "}**

**</button>**

**);**

**};**

**// Export our button component.**

**export default Button;**

**Calculator.css code**

**.Calculator {**

**width: 400px;**

**height: 500px;**

**position: relative;**

**margin: 25px;**

**background: color #45a049;**

**justify-content: center;**

**}**

**.ButtonStyle {**

**background-color: #0997f5;**

**border: 4;**

**color: white;**

**height: 60px;**

**width: 60px;**

**text-align: center;**

**text-decoration: none;**

**display: inline-block;**

**font-size: 24px;**

**cursor: pointer;**

**}**

**.ButtonStyle:hover {**

**background-color: #d016be;**

**}**

**.Keypad {**

**display: flex;**

**flex-wrap: wrap;**

**flex-direction: column;**

**height: 80%;**

**}**

**.screen-row input {**

**width: 220px;**

**height: 30px;**

**background: rgb(17, 17, 17);**

**border: 55px;**

**color: rgb(239, 232, 232);**

**padding: 10px;**

**text-align: left;**

**font-size: 20px;**

**}**

**.ImageContainer {**

**margin-top: 20px;**

**}**

**.ImageStyle {**

**max-width: 150%;**

**height: 150px;**

**display: block;**

**margin: 20px auto;**

**background-image: url("funnyimage.jpg");**

**background-size: cover;**

**background-position: left;**

**}**

**body {**

**background: #d2ccc5;**

**display: flex;**

**align-items: center;**

**justify-content: center;**

**overflow: hidden;**

**}**

**.show-me {**

**height: 60px;**

**width: 240px;**

**background-color: rgb(245, 130, 15);**

**display: flex;**

**}**

**.show-me button {**

**height: 60px;**

**width: 120px;**

**font-size: 20px;**

**}**

**Task 2:   
1.     Add one more button with a text caption as” show me” in the Button.js class provided to you. Add code so that when you click that button it will show your picture.**

**{/\* Add Show Me Button \*/}**

import React from "react";

import ReactDOM from "react-dom/client";

import "./index.css";

import reportWebVitals from "./reportWebVitals";

import KeyPadComponent from "./Calculator";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(

  <React.StrictMode>

    <KeyPadComponent />

  </React.StrictMode>

);

// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals();

**Task 3 :   
2.     Add one more button with text caption as” square” using the Button.js class provided to you. When you click that button it will show the square of the number you entered**

**{/\* Add Square Button \*/}**

**const handleSquare = () => {**

**const number = parseFloat(text1);**

**if (!isNaN(number)) {**

**setText(number \* number); // Calculate square**

**} else {**

**setText("Invalid Input");**

**}**

**<Button label="Square" ClickHandle={handleSquare} />**

**3.     Write A functional Component to apply the Entire Code.  4 Marks**

OUTPUT of given task:-



