Name : **Aaditya Sanjay Pawar**  Roll No. : 190 Div : B Batch : B  
Title : 1 Setting Up React Environment and Creating a Basic React Application .(Install Node.js, NPM, Create React App)

Name : **Aaditya Sanjay Pawar**  Roll No. : 190 Div : B Batch : B  
Title : 2. JSX and Component Creation Using Props.

App.jsx :

import Greeting from './Greeting'

const App = () => {

  return (

    <div>

      <*Greeting* *name*="Aaditya" />

      <*Greeting* *name*="Pranav" />

      <*Greeting* *name*="Parth" />

    </div>

  )

}

export default App

Greeting.jsx :

const Greeting = (*props*) => {

  return (

    <div>

        <h3>Hello, {*props*.name}...!!!</h3>

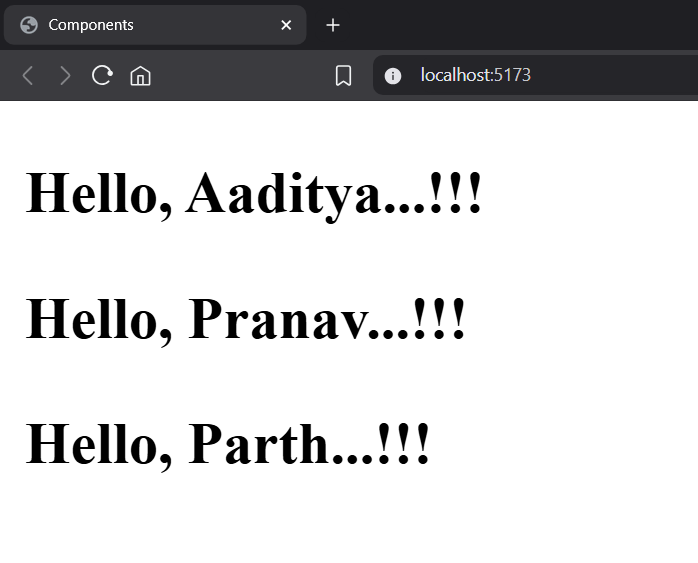
    </div>

  )

}

export default Greeting

Output :



Name : **Aaditya Sanjay Pawar**  Roll No. : 190 Div : B Batch : B  
Title : 3. State Management in React Using use State: Building a Counter App

App.jsx :

import { useState } from 'react'

const App = () => {

  const [count, setCount] = useState(0)

  return (

    <div *className*="flex flex-col justify-center items-center m-26 w-3/5 bg-amber-100 rounded-2xl">

      <h1 *className*='m-2 text-5xl font-bold'>React Counter</h1>

      <div *className*='p-5 border-2 rounded w-10 m-2 h-8 flex justify-center items-center mt-4'>{count}</div>

      <div *className*='flex gap-10 m-4'>

        <button *className*='border-2 rounded px-5'

*onClick*={()=> setCount(count+1)}>+</button>

        <button *className*='border-2 rounded px-5'

*onClick*={()=> setCount(count-1)}>-</button>

        <button *className*='border-2 rounded px-5'

*onClick*={()=> setCount(0)}>Reset</button>

      </div>

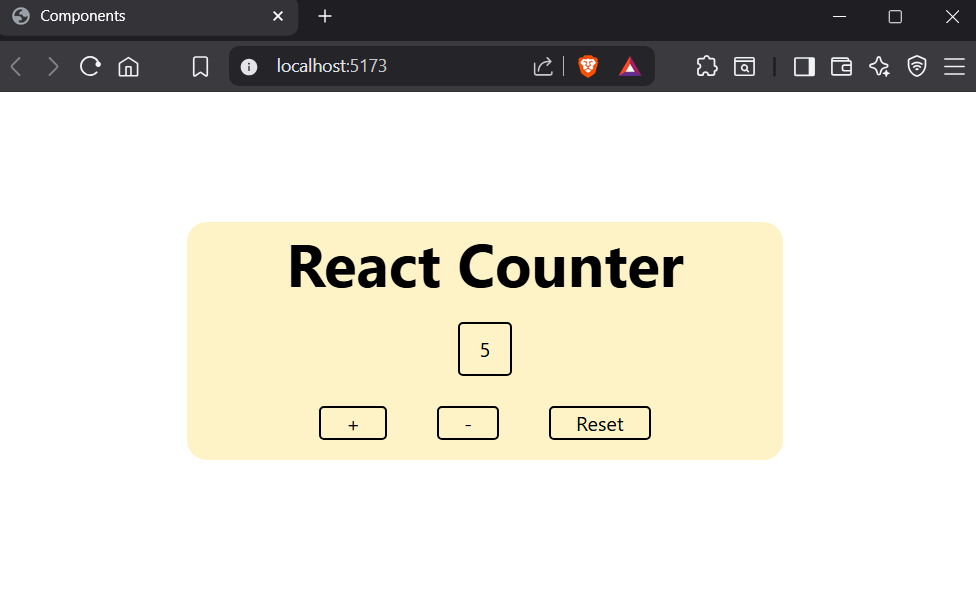
    </div>

  )

}

export default App

Output :



Name : **Aaditya Sanjay Pawar**  Roll No. : 190 Div : B Batch : B  
Title : 4 Event Handling in React: Managing Button Clicks and Input Changes

App.jsx :

import { useState } from 'react'

const App = () => {

  const [name, setname] = useState('')

  const [message, setmessage] = useState('')

  const changeName= (*e*)=>{

    console.log(*e*.target.value);

    setname(*e*.target.value)

  }

  const buttonClick = ()=>{

    setmessage(`Hello, ${name}`)

  }

  return (

    <div *className*='m-10'>

      <input

*type*="text"

*placeholder*='Enter Your Name...'

*value*={name}

*onChange*={changeName}

*className*='border-2 rounded m-4'/> <br />

      <button *onClick*={buttonClick}

*className*='border-2 rounded m-4 px-2'>Greet Me</button>

      <br />

      <p *className*='m-4 text-emerald-500 text-xl font-bold'>{message}</p>

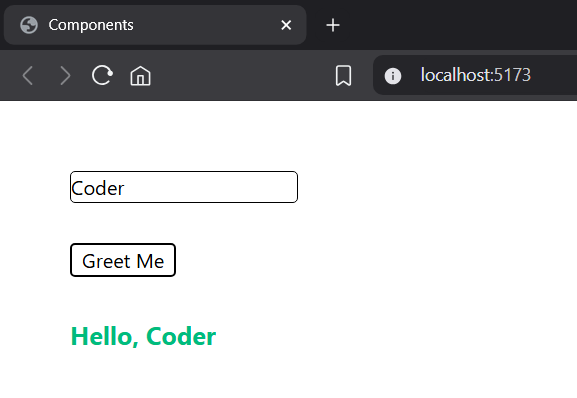
    </div>

  )

}

export default App

Output :



Name : **Aaditya Sanjay Pawar**  Roll No. : 190 Div : B Batch : B  
Title : 5. Rendering Dynamic Lists in React Using Array Mapping and Key Props

App.jsx :

import React from 'react'

const App = () => {

  const fruits = ['Apple', 'Banana', 'Cherry'];// array

  return (

    <div *className*='m-4'>

        <h1>FRUITS LIST</h1>

      <ul>

        {fruits.map((*fruit*, *index*) => (

          <li *key*={*index*}> {*fruit*}</li>

        ))}

      </ul>

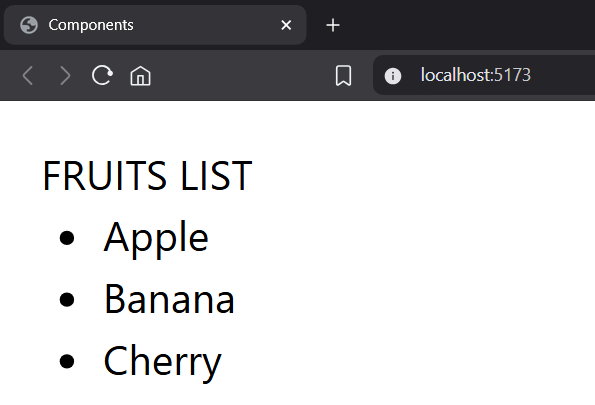
    </div>

  )

}

export default App

Output :



Name : **Aaditya Sanjay Pawar**  Roll No. : 190 Div : B Batch : B  
Title : 6 Building Controlled Forms in React: Managing State and Implementing Basic Validation

App.jsx :

import { useState } from 'react';

const App = () => {

  const [inputValue, setInputValue] = useState('');

  const [submit, setsubmit] = useState("");

  const [error, setError] = useState("");

  function handleInputChange(*event*) {

    const value = *event*.target.value;

    setInputValue(value);

    if (value.trim() !== "") {

      setError("");

    }

  }

  function handleSubmit(*event*) {

*event*.preventDefault();

    if (inputValue.trim() === "") {

      setError("Name is required.");

      setsubmit("");

      return;

    }

    setsubmit(`Hello , ${inputValue} your form is submitted`);

    setError("");

  }

  return (

    <form *onSubmit*={handleSubmit} *style*={{ margin: "50px" }}>

      <label>

        Name:

        <input

*type*="text"

*value*={inputValue}

*onChange*={handleInputChange}

        />

      </label>

      <br />

      <button *type*="submit" *className*="border rounded bg-fuchsia-300 px-4 my-4">

        Submit

      </button>

      {error && <p *style*={{ color: 'red' }}>{error}</p>}

      <p>{submit}</p>

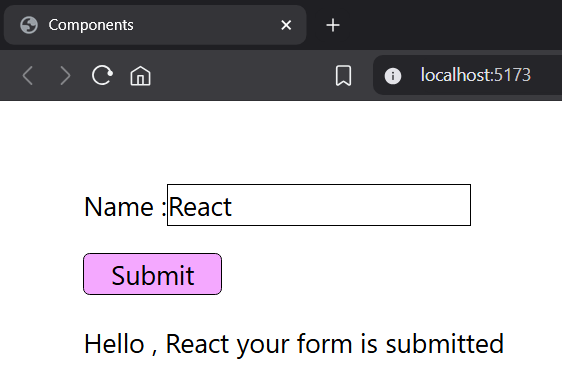
    </form>

  );

}

export default App

Output :



Name : **Aaditya Sanjay Pawar**  Roll No. : 190 Div : B Batch : B  
Title : 7 Styling React Components: Implementing CSS Modules and Inline Styles

App.jsx :

import styles from './Button.module.css';

const App = () => {

  return (

    <button

*className*={styles.button} //module css

*style*={{ backgroundColor: "rgba(154, 43, 252, 0.655)" }} //inline css

*onClick*={() => { alert("Button Clicked") }}>

      Click Me

    </button>

  );

}

export default App

Button.module.css :

.button {

    color: white;

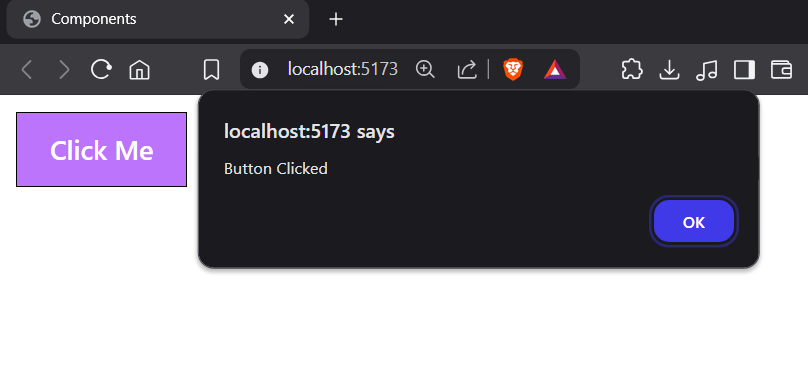
    padding: 10px 20px;

    margin: 10px;

    font-weight: 500

}

Output :



Name : **Aaditya Sanjay Pawar**  Roll No. : 190 Div : B Batch : B  
Title : 8 Conditional styling in React using both CSS Modules and inline styles. (Toggle Button with Conditional Styling).

App.jsx :

import "./toggle.css";

import { useState } from "react";

const App = () => {

  const [isOn, setIsOn] = useState(false);

  const buttonStyle = {

    backgroundColor: isOn ? '#4CAF50' : '#f44336',

    color: 'white',

    padding: '12px 20px',

    border: 'none',

    borderRadius: '5px',

    fontSize: '16px',

    cursor: 'pointer',

    transition: 'background-color 0.3s ease',

  };

  return (

    <div *className*="container">

      <span *className*="label">Toggle is {isOn ? 'ON' : 'OFF'}</span>

      <button *style*={buttonStyle} *onClick*={() => setIsOn(!isOn)}>

        {isOn ? 'Turn OFF' : 'Turn ON'}

      </button>

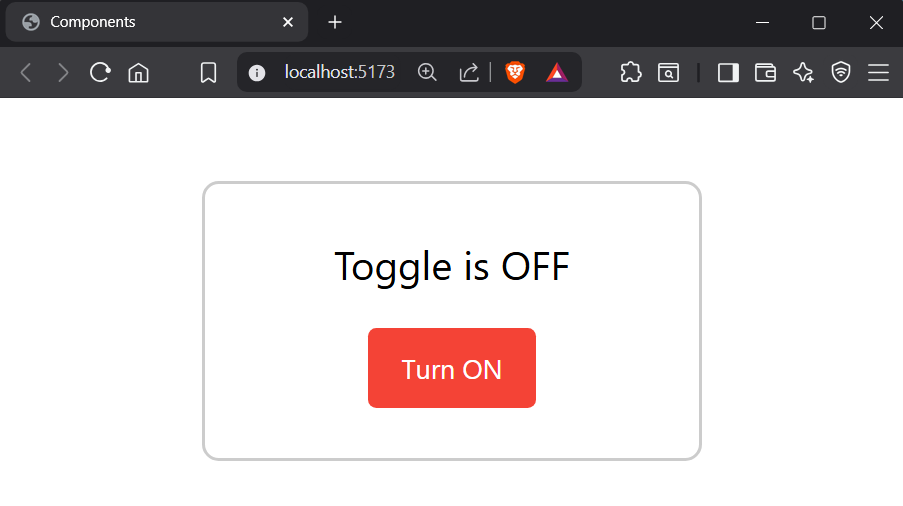
    </div>

  );

}

export default App

Output :



Name : **Aaditya Sanjay Pawar**  Roll No. : 190 Div : B Batch : B  
Title : 9 Developing a Personal Portfolio Website with React and Bootstrap

App.jsx :

import Navbar from './Navbar';

import Home from './Home';

import About from './About';

import Projects from './Projects';

import Footer from './Footer';

function App() {

  return (

    <div>

      <*Navbar* />

      <*Home* />

      <*About* />

      <*Projects* />

      <*Footer* />

    </div>

  );

}

export default App;

Navbar.jsx:

function Navbar() {

    return (

        <nav *className*="navbar navbar-expand-lg navbar-dark bg-dark">

            <div *className*="container">

                <a *className*="navbar-brand">My Portfolio</a>

                <div *className*="collapse navbar-collapse">

                    <ul *className*="navbar-nav ms-auto">

                        <li *className*="nav-item"><a *className*="nav-link" *href*="#home">Home</a></li>

                        <li *className*="nav-item"><a *className*="nav-link" *href*="#about">About</a></li>

                        <li *className*="nav-item"><a *className*="nav-link" *href*="#projects">Projects</a></li>

                    </ul>

                </div>

            </div>

        </nav>

    );

}

export default Navbar;

Home.jsx

function Home() {

    return (

        <section *id*="home" *className*="bg-light text-center py-5">

            <div *className*="container">

                <h1>Hello, I’m Aaditya </h1>

                <p>Frontend Developer | React Enthusiast</p>

            </div>

        </section>

    );

}

export default Home;

About.jsx

import React from 'react';

function About() {

    return (

        <section *id*="about" *className*="py-5">

            <div *className*="container">

                <h2>About Me</h2>

                <p>I am a passionate web developer with skills in React, Bootstrap, and modern web

                    technologies.</p>

            </div>

        </section>

    );

}

export default About;

Project.jsx

function Projects() {

    return (

        <section *id*="projects" *className*="bg-light py-5">

            <div *className*="container">

                <h2>Projects</h2>

                <ul>

                    <li>Portfolio Website</li>

                    <li>To-do App with React</li>

                    <li>Weather App using API</li>

                </ul>

            </div>

        </section>

    );

}

export default Projects;

Footer.jsx

function Footer() {

    return (

        <footer *className*="bg-dark text-white text-center py-3">

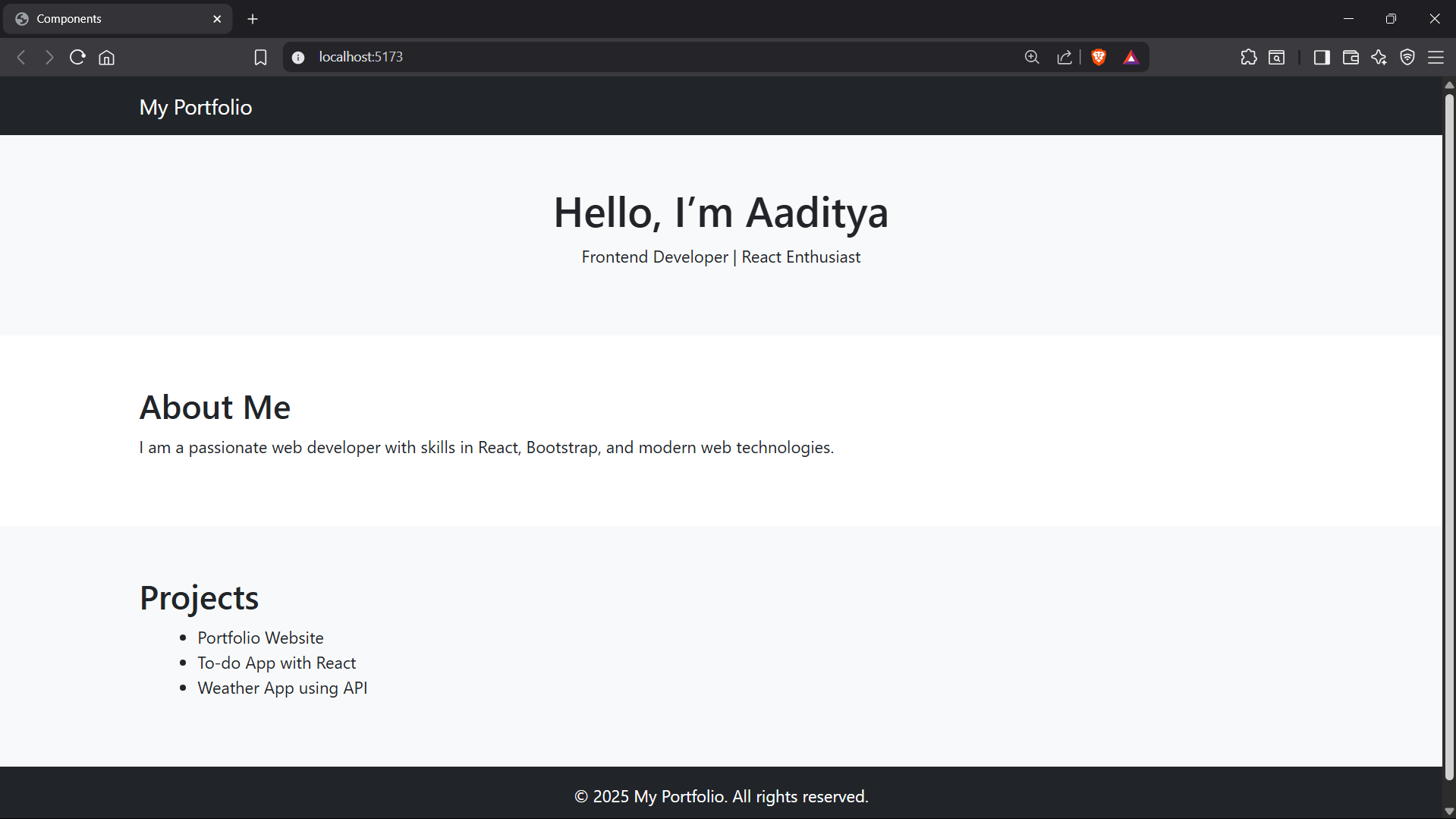
            <p>&copy; 2025 My Portfolio. All rights reserved.</p>

        </footer>

    );

}

export default Footer;

Output :  


Name : **Aaditya Sanjay Pawar**  Roll No. : 190 Div : B Batch : B  
Title : 10 Setting Up a Node.js and Express.js Development Environment: A Practical Guide

Step 1: Setup the Backend (Node.js + Express.js)

1.Go to your folder on the top write cmd

2. In cmd write this commands one by one

 mkdir fullstack-app

 cd fullstack-app

 mkdir backend

 cd backend

 npm init –y

 code .

 in VS code your folder is open and create file server.js copy the code in server.js

2. Install Express

npm install express cors

3. Create server.js

add this code in server.js

const express = require('express');

const cors = require('cors');

const app = express();

const PORT = 5000;

app.use(cors());

app.use(express.json());

// Dummy API

app.get('/api/message', (req, res) => {

res.json({ message: "Hello from the backend!" });

});

app.listen(PORT, () => {

console.log(`Server running on http://localhost:${PORT}`);

});

4. Run the Backend Server

Run this command on CMD :   
node server.js

Visit: http://localhost:5000/api/message

You should see:

{ "message": "Hello from the backend!" }

