**REACT**

1. **ReactJS-HOL**

**Single Page Application (SPA)**

A Single Page Application is a type of web application that dynamically rewrites the current page with new data from the server, instead of loading entire new pages. This is done using JavaScript frameworks Or ReactJS.

**Benefits of SPA:**

* Fast and Seamless Navigation: Only data is fetched, not entire pages, resulting in a smoother user experience.
* Reduced Server Load: Less data is transferred between client and server.
* Improved Performance: Once the SPA is loaded, subsequent interactions are much faster.
* Offline Support: SPAs can work offline using caching and service workers.
* Code Reusability: Easier to reuse components and logic across the application.

**React:**

**React** is an open-source JavaScript library developed by **Meta** for building fast and interactive user interfaces, especially for single-page applications.

**myfirstreact**

**CODE: ( App.js )**

function App() {

return (

<div>

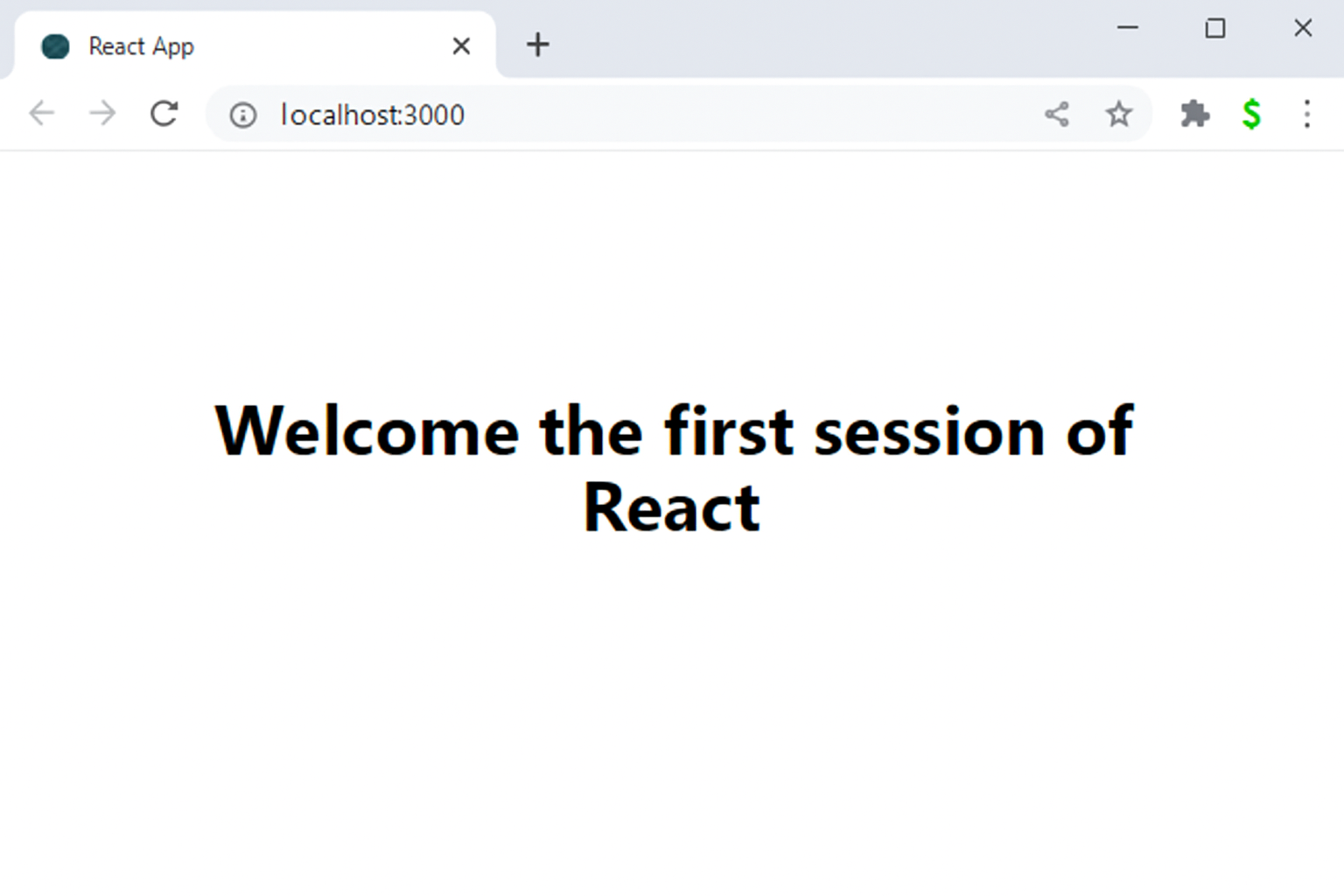
<h1>Welcome to the first session of React</h1>

</div>

);

}

**OUTPUT:**



**2. ReactJS-HOL**

**Student App**

**CODE:**

import React from 'react';

// Home Component

function Home() {

return (

<div>

<h2>Welcome to the Home page of Student Management Portal</h2>

</div>

);

}

// About Component

function About() {

return (

<div>

<h2>Welcome to the About page of the Student Management Portal</h2>

</div>

);

}

// Contact Component

function Contact() {

return (

<div>

<h2>Welcome to the Contact page of the Student Management Portal</h2>

</div>

);

}

// Main App Component

function App() {

return (

<div>

<h1>Student Management Portal</h1>

<Home />

<About />

<Contact />

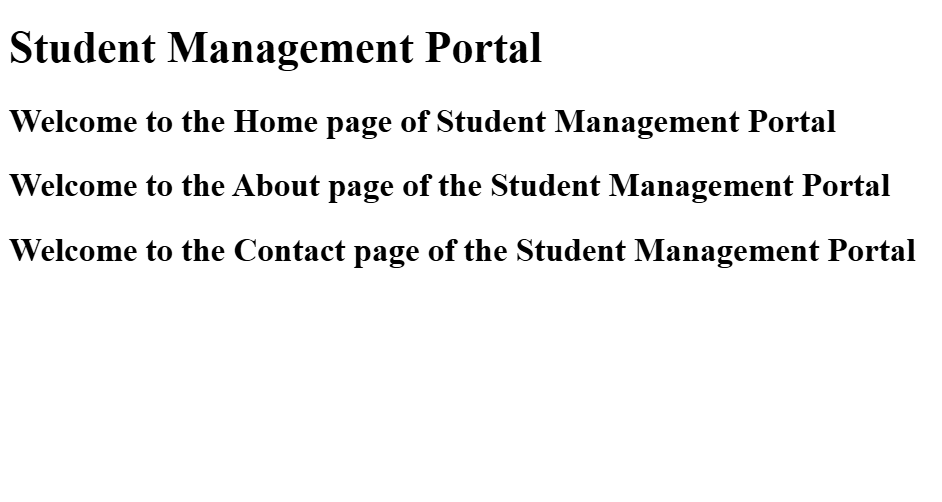
</div>

);

}

export default App;

**OUTPUT:**



**3. ReactJS-HOL**

**Scorecalculatorapp**

**CODE:**

**CalculateScore.js**

import React from "react";

import "../Stylesheets/mystyle.css";

function CalculateScore() {

const student = {

name: "Steeve",

school: "DNV Public School",

total: 284Marks

goal: 500

};

const average = (student.total / student.goal) \* 100;

return (

<div className="container">

<h1 className="heading">Student Score Summary</h1>

<p><strong>Name:</strong> {student.name}</p>

<p><strong>School:</strong> {student.school}</p>

<p><strong>Total Marks Obtained:</strong> {student.total}</p>

<p><strong>Goal:</strong> {student.goal}</p>

<p><strong>Average Score (%):</strong> {average.toFixed(2)}%</p>

</div>

);

}

export default CalculateScore;

**mystyle.css (Styling)**

.container {

margin: 30px auto;

padding: 20px;

max-width: 500px;

border: 2px solid #4CAF50;

border-radius: 12px;

background-color: #f9f9f9;

font-family: Arial, sans-serif;

}

.heading {

text-align: center;

color: #4CAF50;

margin-bottom: 20px;

}

**App.js (Root component)**

import React from "react";

import CalculateScore from "./Components/CalculateScore";

function App() {

return (

<div>

<CalculateScore />

</div>

);

}

export default App;

**OUTPUT:**



**4. React JS-HCL**

**Blogapp**

**CODE:**

**Post.js**

import React from "react";

class Post extends React.Component {

render() {

const { title, body } = this.props;

return (

<div style={{ border: "1px solid gray", padding: "15px", margin: "10px 0", borderRadius: "8px" }}>

<h3>{title}</h3>

<p>{body}</p>

</div>

);

}

}

class Posts extends React.Component {

constructor(props) {

super(props);

this.state = {

posts: [],

};

}

loadPosts = async () => {

try {

const response = await fetch("https://jsonplaceholder.typicode.com/posts");

const data = await response.json();

this.setState({ posts: data.slice(0, 5) }); // Keep it simple with 5 posts

} catch (error) {

console.error("Failed to load posts:", error);

}

};

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

alert("Something went wrong while displaying the posts.");

console.error("Caught error:", error, info);

}

render() {

return (

<div>

<h2>Latest Blog Posts</h2>

{this.state.posts.map((post) => (

<Post key={post.id} title={post.title} body={post.body} />

))}

</div>

);

}

}

function App() {

return (

<div style={{ fontFamily: "Arial", padding: "20px" }}>

<h1>Welcome to Blog App</h1>

<Posts />

</div>

);

}

export default App;

OUTPUT:

**Welcome to Blog App**  
Latest Blog Posts



* 1. **ReactJS-HOL**

**Dashboard for Cognizant**

**CODE: (App.js)**

import React from "react";

function CohortDetails(props) {

const { name, duration, status } = props;

const boxStyle = {

width: "300px",

display: "inline-block",

margin: "10px",

padding: "10px 20px",

border: "1px solid black",

borderRadius: "10px",

fontFamily: "Arial",

};

const titleStyle = {

color: status === "ongoing" ? "green" : "blue"

};

const dtStyle = {

fontWeight: 500

};

return (

<div style={boxStyle}>

<h3 style={titleStyle}>{name}</h3>

<dl>

<dt style={dtStyle}>Duration:</dt>

<dd>{duration}</dd>

<dt style={dtStyle}>Status:</dt>

<dd>{status}</dd>

</dl>

</div>

);

}

function App() {

return (

<div style={{ padding: "20px" }}>

<h2>Cohort Dashboard</h2>

<CohortDetails name="React Bootcamp" duration="4 weeks" status="ongoing" />

<CohortDetails name="Java Sprint" duration="6 weeks" status="completed" />

</div>

);

}

export default App;

**OUTPUT:**



* 1. **ReactJS-HOL**

**TrainersApp**

**CODE: (App.js)**

import React from 'react';

import { BrowserRouter, Routes, Route, Link, useParams } from 'react-router-dom';

class Trainer {

  constructor(trainerId, name, email, phone, technology, skills) {

    this.trainerId = trainerId;

    this.name = name;

    this.email = email;

    this.phone = phone;

    this.technology = technology;

    this.skills = skills;

  }

}

const trainersMock = [

  new Trainer(

    't-syed8',

    'Syed Khaleelullah',

    'khaleelullah@cognizant.com',

    '97676516962',

    '.NET',

    ['C#', 'SQL Server', 'React', '.NET Core']

  ),

  new Trainer(

    't-jojo',

    'Jojo Jose',

    'jojo@cognizant.com',

    '9897199231',

    'Java',

    ['Java', 'JSP', 'Angular', 'Spring']

  ),

  new Trainer(

    't-elisa',

    'Elisa Jones',

    'elisa@cognizant.com',

    '9871212235',

    'Python',

    ['Python', 'Django', 'Angular']

  ),

  new Trainer(

    't-mary',

    'Mary Smith',

    'mary.smith@cognizant.com',

    '9987654321',

    'Cloud',

    ['AWS', 'Azure', 'DevOps']

  ),

  new Trainer(

    't-john',

    'John Doe',

    'john.doe@cognizant.com',

    '9123456789',

    'Frontend',

    ['HTML', 'CSS', 'JavaScript', 'Vue.js']

  ),

];

const Home = () => {

  return (

    <div className="p-8 bg-gradient-to-r from-blue-50 to-indigo-50 rounded-lg shadow-lg">

      <h2 className="text-4xl font-extrabold text-gray-900 mb-4 animate-fade-in">

        Welcome to My Academy Trainers Page!

      </h2>

      <p className="text-lg text-gray-700">

        Explore our talented pool of trainers and their diverse expertise.

      </p>

    </div>

  );

};

const TrainersList = ({ trainers }) => {

  return (

    <div className="p-8 bg-white rounded-lg shadow-lg">

      <h2 className="text-3xl font-bold text-gray-800 mb-6 border-b-2 border-indigo-300 pb-2">

        Our Trainers

      </h2>

      <ul className="list-disc list-inside space-y-4">

        {trainers.map(trainer => (

          <li key={trainer.trainerId} className="text-lg text-gray-700 hover:text-indigo-600 transition duration-300">

            <Link to={`/trainers/${trainer.trainerId}`} className="text-blue-600 hover:underline font-semibold">

              {trainer.name}

            </Link>

          </li>

        ))}

      </ul>

    </div>

  );

};

const HomeAndTrainers = ({ trainers }) => {

  return (

    <>

      <Home />

      <div className="mt-8"> {/\* Adds spacing between the Home message and the list \*/}

        <TrainersList trainers={trainers} />

      </div>

    </>

  );

};

const TrainerDetail = ({ trainers }) => {

  const { id } = useParams(); // Hook to get URL parameters

  const trainer = trainers.find(t => t.trainerId === id); // Find the trainer by ID

  // Handle case where trainer is not found

  if (!trainer) {

    return (

      <div className="p-8 bg-red-100 border border-red-400 text-red-700 rounded-lg shadow-md">

        <h2 className="text-3xl font-bold mb-4">Trainer Not Found</h2>

        <p>The trainer with ID "{id}" could not be found.</p>

        <Link to="/trainers" className="text-blue-600 hover:underline mt-4 block">

          Back to Trainers List

        </Link>

      </div>

    );

  }

  return (

    <div className="p-8 bg-white rounded-lg shadow-xl border-t-4 border-indigo-500">

      <h2 className="text-3xl font-bold text-gray-800 mb-6">Trainer Details</h2>

      <div className="space-y-4 text-gray-700">

        <p className="text-2xl font-semibold text-indigo-700">

          {trainer.name} ({trainer.technology})

        </p>

        <p className="text-lg">

          <span className="font-medium text-gray-900">Email:</span> {trainer.email}

        </p>

        <p className="text-lg">

          <span className="font-medium text-gray-900">Phone:</span> {trainer.phone}

        </p>

        <div className="mt-6">

          <h3 className="text-xl font-bold text-gray-800 mb-3 border-b border-gray-300 pb-2">Skills:</h3>

          <ul className="list-disc list-inside space-y-2 text-lg text-gray-600">

            {trainer.skills.map((skill, index) => (

              <li key={index} className="flex items-center">

                 {/\* SVG icon for visual appeal \*/}

                 <svg className="w-5 h-5 text-green-500 mr-2" fill="currentColor" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg"><path fillRule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 000 16zm3.707-9.293a1 1 0 00-1.414-1.414L9 10.586 7.707 9.293a1 1 0 00-1.414 1.414l2 2a1 1 0 001.414 0l4-4z" clipRule="evenodd"></path></svg>

                 {skill}

              </li>

            ))}

          </ul>

        </div>

      </div>

      <Link to="/trainers" className="inline-block mt-8 px-6 py-3 bg-indigo-600 text-white font-semibold rounded-lg shadow-md hover:bg-indigo-700 transition duration-300 ease-in-out">

        Back to Trainers List

      </Link>

    </div>

  );

};

// Main App Component

// Sets up the routing for the single-page application and renders the navigation.

const App = () => {

  return (

    <BrowserRouter>

      {/\* Main container with Inter font, responsive padding, and background \*/}

      <div className="font-['Inter',\_sans-serif] min-h-screen bg-gray-100 pb-10">

        {/\* Header Section with gradient and shadow \*/}

        <header className="bg-gradient-to-r from-blue-700 to-purple-800 text-white p-6 shadow-xl rounded-b-xl">

          <div className="max-w-4xl mx-auto flex flex-col sm:flex-row justify-between items-center">

            <h1 className="text-4xl font-extrabold mb-4 sm:mb-0">My Academy Trainers App</h1>

            {/\* Navigation Links with hover effects \*/}

            <nav>

              <ul className="flex space-x-6 text-lg">

                <li>

                  <Link

                    to="/"

                    className="hover:text-yellow-300 transition duration-300 ease-in-out transform hover:scale-105 px-4 py-2 rounded-md focus:outline-none focus:ring-2 focus:ring-yellow-300 focus:ring-opacity-75"

                  >

                    Home

                  </Link>

                </li>

                <li>

                  <Link

                    to="/trainers"

                    className="hover:text-yellow-300 transition duration-300 ease-in-out transform hover:scale-105 px-4 py-2 rounded-md focus:outline-none focus:ring-2 focus:ring-yellow-300 focus:ring-opacity-75"

                  >

                    Show Trainers

                  </Link>

                </li>

              </ul>

            </nav>

          </div>

        </header>

        {/\* Main Content Area for routed components \*/}

        <main className="max-w-4xl mx-auto mt-8 p-4">

          {/\* Define Application Routes \*/}

          <Routes>

            {/\* The root path now displays both Home message and Trainers List \*/}

            <Route path="/" element={<HomeAndTrainers trainers={trainersMock} />} />

            {/\* Dedicated route for just the Trainers List \*/}

            <Route path="/trainers" element={<TrainersList trainers={trainersMock} />} />

            {/\* Dynamic route for individual trainer details \*/}

            <Route path="/trainers/:id" element={<TrainerDetail trainers={trainersMock} />} />

            {/\* Fallback route for any unmatched paths (404) \*/}

            <Route path="\*" element={

              <div className="p-8 bg-orange-100 border border-orange-400 text-orange-700 rounded-lg shadow-md text-center">

                <h2 className="text-3xl font-bold mb-4">404 - Page Not Found</h2>

                <p>Oops! The page you are looking for does not exist.</p>

                <Link to="/" className="text-blue-600 hover:underline mt-4 block">

                  Go to Home

                </Link>

              </div>

            } />

          </Routes>

        </main>

      </div>

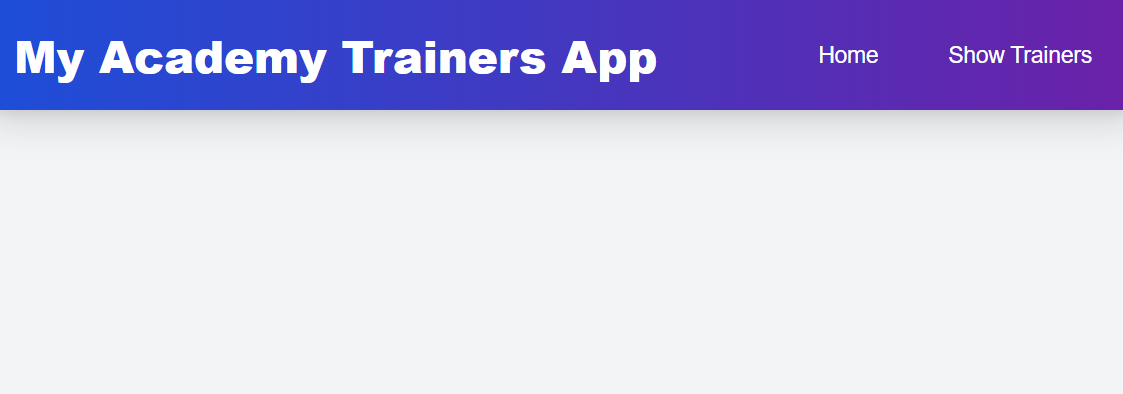
    </BrowserRouter>

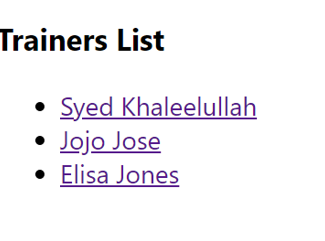
  );

};

export default App; // Export the main App component for rendering

**OUTPUT:**





* 1. **ReactJS-HOL**

**Shoppingapp**

**CODE: (App.js)**

import React from 'react';

const CartItem = ({ itemName, price }) => {

  return (

    <div className="bg-white p-4 rounded-lg shadow-md hover:shadow-lg transition-shadow duration-300 ease-in-out flex justify-between items-center border border-gray-200">

      <p className="text-lg font-semibold text-gray-800">{itemName}</p>

      <p className="text-xl font-bold text-indigo-600">₹{price.toLocaleString()}</p>

    </div>

  );

};

class OnlineShopping extends React.Component {

  render() {

    // Array of items as specified in your document

    const cartInfo = [

      { itemName: "Laptop", price: 80000 },

      { itemName: "TV", price: 120000 },

      { itemName: "Washing Machine", price: 50000 },

      { itemName: "Mobile", price: 30000 },

      { itemName: "Fridge", price: 70000 }

    ];

    return (

      <div className="flex flex-col items-center p-6 bg-gradient-to-br from-blue-50 to-purple-50 min-h-screen font-['Inter',\_sans-serif]">

        <h1 className="text-5xl font-extrabold text-gray-900 mb-10 text-center animate-fade-in-down drop-shadow-lg">

          <span className="text-indigo-600">Your</span> Shopping Cart

        </h1>

        <div className="w-full max-w-md bg-white rounded-xl shadow-2xl p-6 space-y-4 border border-blue-200 animate-slide-in">

          <h2 className="text-3xl font-bold text-gray-800 border-b-2 border-indigo-300 pb-3 mb-4 text-center">

            Items Ordered

          </h2>

          {/\* Loop through the cartInfo array and render a CartItem for each \*/}

          {cartInfo.map((item, index) => (

            <CartItem

              key={index} // Using index as key, consider unique IDs for production apps

              itemName={item.itemName}

              price={item.price}

            />

          ))}

        </div>

        {/\* Example for a total price display - optional enhancement \*/}

        <div className="mt-8 p-5 bg-gradient-to-r from-green-100 to-emerald-100 rounded-lg shadow-xl text-center border-l-4 border-green-500 animate-fade-in">

          <p className="text-2xl font-bold text-green-800">

            Estimated Total: ₹

            {(cartInfo.reduce((sum, item) => sum + item.price, 0)).toLocaleString()}

          </p>

          <button className="mt-4 px-8 py-3 bg-green-600 text-white text-xl font-semibold rounded-full shadow-lg hover:bg-green-700 transition duration-300 ease-in-out transform hover:scale-105 focus:outline-none focus:ring-2 focus:ring-green-500 focus:ring-opacity-75">

            Proceed to Checkout

          </button>

        </div>

      </div>

    );

  }

}

// Main App

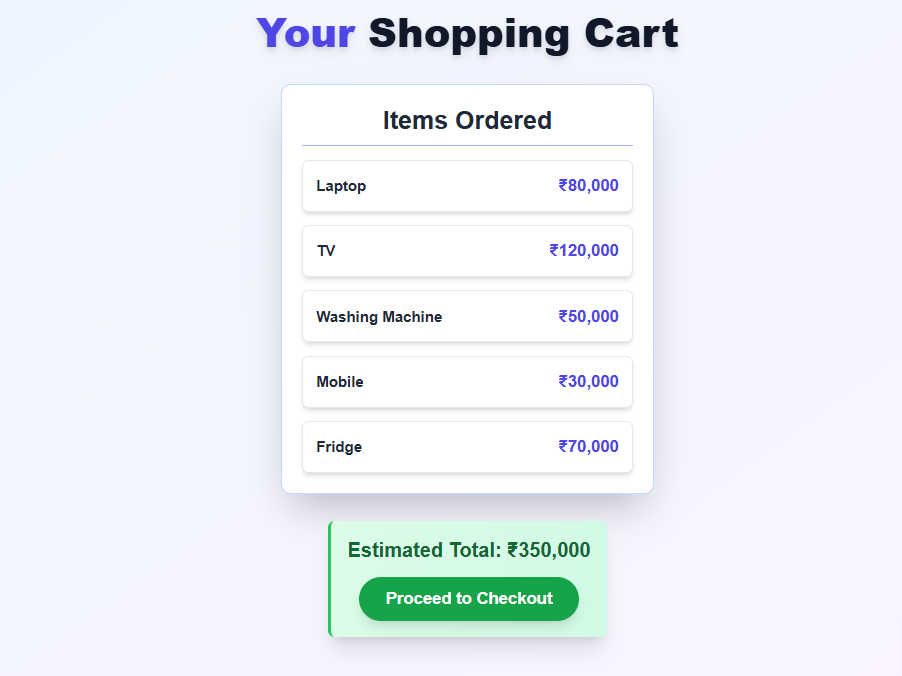
const App = () => {

  return <OnlineShopping />;

};

export default App;

OUTPUT:



* 1. **ReactJS-HOL**

**Counterapp**

**App.js**

import React from 'react';

class CountPeople extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      entryCount: 3, // Number of people who entered

      exitCount: 1,  // Number of people who exited

    };

    this.updateEntry = this.updateEntry.bind(this);

    this.updateExit = this.updateExit.bind(this);

  }

  // Method to increment the entry count

  updateEntry() {

    this.setState(prevState => ({

      entryCount: prevState.entryCount + 1,

    }));

  }

  updateExit() {

    this.setState(prevState => ({

      exitCount: prevState.exitCount + 1,

    }));

  }

  render() {

    // Destructure state variables for easier access in render method

    const { entryCount, exitCount } = this.state;

    return (

      <div className="flex flex-col items-center justify-center min-h-screen bg-gradient-to-br from-blue-100 to-purple-100 p-4 font-['Inter',\_sans-serif]">

        {/\* Main Title \*/}

        <h1 className="text-5xl md:text-6xl font-extrabold text-gray-800 mb-12 text-center drop-shadow-lg animate-fade-in-down">

          Mall Traffic <span className="text-purple-600">Counter</span>

        </h1>

        {/\* Counter Display Cards \*/}

        <div className="flex flex-col md:flex-row space-y-8 md:space-y-0 md:space-x-12 mb-12">

          {/\* Entry Counter Card \*/}

          <div className="bg-white p-8 rounded-2xl shadow-xl flex flex-col items-center justify-center border-b-4 border-green-500 transform hover:scale-105 transition-all duration-300 animate-slide-in-left">

            <h2 className="text-3xl font-bold text-green-700 mb-4">People Entered</h2>

            <p className="text-6xl font-extrabold text-green-600 mb-6 transition-all duration-300 transform scale-100 animate-pulse-grow">

              {entryCount}

            </p>

            <button

              onClick={this.updateEntry}

              className="px-8 py-3 bg-green-600 text-white font-semibold rounded-full shadow-lg hover:bg-green-700 focus:outline-none focus:ring-4 focus:ring-green-300 transition duration-300 ease-in-out transform hover:-translate-y-1 active:scale-95"

            >

              <svg className="w-6 h-6 inline-block mr-2" fill="none" stroke="currentColor" viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg"><path strokeLinecap="round" strokeLinejoin="round" strokeWidth="2" d="M11 16l-4-4m0 0l4-4m-4 4h14m-5 4v1a3 3 0 01-3 3H6a3 3 0 01-3-3V7a3 3 0 013-3h7a3 3 0 013 3v1"></path></svg>

              Login

            </button>

            <p className="mt-4 text-green-500 font-medium">{entryCount} People Entered!!!</p>

          </div>

          {/\* Exit Counter Card \*/}

          <div className="bg-white p-8 rounded-2xl shadow-xl flex flex-col items-center justify-center border-b-4 border-red-500 transform hover:scale-105 transition-all duration-300 animate-slide-in-right">

            <h2 className="text-3xl font-bold text-red-700 mb-4">People Exited</h2>

            <p className="text-6xl font-extrabold text-red-600 mb-6 transition-all duration-300 transform scale-100 animate-pulse-grow">

              {exitCount}

            </p>

            <button

              onClick={this.updateExit}

              className="px-8 py-3 bg-red-600 text-white font-semibold rounded-full shadow-lg hover:bg-red-700 focus:outline-none focus:ring-4 focus:ring-red-300 transition duration-300 ease-in-out transform hover:-translate-y-1 active:scale-95"

            >

              <svg className="w-6 h-6 inline-block mr-2" fill="none" stroke="currentColor" viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg"><path strokeLinecap="round" strokeLinejoin="round" strokeWidth="2" d="M13 8l4 4m0 0l-4 4m4-4H3m6 4v1a3 3 0 003 3h7a3 3 0 003-3V7a3 3 0 00-3-3H9a3 3 0 00-3 3v1"></path></svg>

              Exit

            </button>

            <p className="mt-4 text-red-500 font-medium">{exitCount} People Left!!!</p>

          </div>

        </div>

        {/\* Dynamic Styles for Animations (added directly for Canvas preview) \*/}

        <style>

          {`

          @keyframes fade-in-down {

            from {

              opacity: 0;

              transform: translateY(-20px);

            }

            to {

              opacity: 1;

              transform: translateY(0);

            }

          }

          @keyframes slide-in-left {

            from {

              opacity: 0;

              transform: translateX(-50px);

            }

            to {

              opacity: 1;

              transform: translateX(0);

            }

          }

          @keyframes slide-in-right {

            from {

              opacity: 0;

              transform: translateX(50px);

            }

            to {

              opacity: 1;

              transform: translateX(0);

            }

          }

          @keyframes pulse-grow {

            0%, 100% {

              transform: scale(1);

            }

            50% {

              transform: scale(1.03);

            }

          }

          .animate-fade-in-down { animation: fade-in-down 0.8s ease-out forwards; }

          .animate-slide-in-left { animation: slide-in-left 0.7s ease-out forwards; }

          .animate-slide-in-right { animation: slide-in-right 0.7s ease-out forwards; }

          .animate-pulse-grow { animation: pulse-grow 1.5s infinite ease-in-out; }

          `}

        </style>

      </div>

    );

  }

}

// Main App component that renders the CountPeople component

const App = () => {

  return <CountPeople />;

};

export default App;

**OUTPUT:**

****