SINGLE PAGE APPLICATION

A Single Page Application (SPA) application that dynamically rewrites an existing web page with new information from the web server. In a React application, components retrieve site contend and render it to a single HTML file in your project

Single Page Applications is one of the three web application types (the other two are Progressive web Apps and Multi-Page Apps). It is convenient to code- device-independent loads fast uses AJAX effective caching easy to debug with Chrome easy to scan well-formatted good error-handling valid mark-up-clean code usable formats and encourages reusability of code. Best examples include Google Maps, Gmail, Airbnb, Netflix, Pinterest, Paypal, GitHub, Facebook, therefore, single-page applications are adept for SaaS products, dynamic apps and social networks

How to Install React Router

To install React Router into your React project using npm, the JavaScript package manager, run the following command in your project directory:

npm i react-router-dom

Alternatively, you can download the package using Yarn, a package manager that allows you to install library packages offline.
To install React Router using Yam, run this command:
yarn add react-router-dom@6

How does React contribute to the making of a Single Page Application?

SPA's soud an initial request forward.convindes to the server and receive an indexed HTML file. React Native makes use of Node.js, a JavaScript runtime to create JavaScript code. When a browser receives this response, React comes into play. It stops sach a request by not letting it reach the server. Meanwhile, it requests the contact component and leah it making single-page applicationsfaster and 100% more reliable. React ape brings up the "components" upon user request.

What is React Router? How will it assist in creating a React SPA?

React router comes with declarative routing capabilities to build single-page applications in React. It extends your understanding of how single-page applications work, appending the features of React cross-platform environment.

React's powerful declarative programing model comes with navigational components that declaratively compose the application. It creates bookmarkable URLs for the web app or to navigate

in React Native. React Router works everywhere there is React rendering. You can pick web or native accordingly.

Procedure steps:

Step1:-Open Visual Studio Go to file and select open folder.

Step2:-Open my-react-app.

Step3:-Go to sre and select index.js file. Write the program in index.js.

```
import React, { useState } from "react";
import ReactDOM from "react-dom";
const App = () => {
 const [products, setProducts] = useState([
  { id: 1, name: "Apple", price: 1.99 },
  { id: 2, name: "Orange", price: 3.99 },
  { id: 3, name: "Banana", price: 2.99 },
 ]);
 const [cart, setCart] = useState([]);
 const addToCart = (product) => {
  setCart([...cart, product]);
 };
 const removeFromCart = (product) => {
  setCart(cart.filter((p) => p.id !== product.id));
 };
 return (
  <div>
   <h1>Products</h1>
   <ul>
    {products.map((product) => (
     <li key={product.id}>
      {product.name} - ${product.price}
      <button onClick={() => addToCart(product)}>Add to cart</button>
     </li>
    ))}
   </ul>
   <h1>Cart</h1>
   <ul>
    {cart.map((product) => (
```

```
        <li key={product.id}>
          {product.name} - ${product.price}
          <button onClick={() => removeFromCart(product)}>Remove from cart</button>
        </li>
       ))}
     </ul>
     <p>Total: ${cart.reduce((total, product) => total + product.price, 0).toFixed(2)}</p>
   </div>
  );
};
ReactDOM.render(<App />, document.getElementById("root"));
```

OUTPUT:-