CO2 Skill WEEK 5
Implementing Routing Mechanism

ReactJS application that implements a routing mechanism for managing and displaying a student's CGPA details and download as pdf .

The application consists of two pages Add CGPA - Allows the user to input student details and CGPA.

1. View CGPA - Displays the student's CGPA information.

Project Structure:

```
src/
├── components/
│   ├── AddCGPA.js
│   └── ViewCGPA.js
├── App.js
└── index.js
```

Note:

npx create-react-app student-cgpa
cd student-cgpa
npm install react-router-dom
npm install jspdf

Programs:

AddCGPA.jsx

```
import React, { useState } from "react";
import { jsPDF } from "jspdf";

const AddCGPA = ({ setStudentData }) => {
  const [studentName, setStudentName] = useState("");
  const [rollNumber, setRollNumber] = useState("");
  const [cgpa, setCgpa] = useState("");

  const handleSubmit = (e) => {
    e.preventDefault();

    if (studentName && rollNumber && cgpa) {
      setStudentData({ studentName, rollNumber, cgpa });
      alert("Student CGPA Added Successfully!");
    } else {
      alert("Please fill all fields!");
    }
  };
```

```
const handleDownload = () => {
  const doc = new jsPDF();

  doc.text("Student Details", 20, 20);
  doc.text(`Name: ${studentName}`, 20, 30);
  doc.text(`Roll Number: ${rollNumber}`, 20, 40);
  doc.text(`CGPA: ${cgpa}`, 20, 50);

  doc.save("student-details.pdf");
};

return (
  <div>
    <h2>Add Student CGPA</h2>
    <form onSubmit={handleSubmit}>
      <div>
        <label>Student Name:</label>
        <input
          type="text"
          value={studentName}
          onChange={(e) => setStudentName(e.target.value)}
          placeholder="Enter name"
          required
        />
      </div>
      <div>
        <label>Roll Number:</label>
        <input
          type="text"
          value={rollNumber}
          onChange={(e) => setRollNumber(e.target.value)}
          placeholder="Enter roll number"
          required
        />
      </div>
      <div>
        <label>CGPA:</label>
        <input
          type="number"
          step="0.01"
          value={cgpa}
          onChange={(e) => setCgpa(e.target.value)}
          placeholder="Enter CGPA"
          required
        />
      </div>
      <button type="submit">Add CGPA</button>
    </form>
    <button onClick={handleDownload} style={{ marginTop: "20px" }}>
      Download PDF
```

```
      </button>
    </div>
  );
};

export default AddCGPA;
```

ViewCGPA.jsx

```
import React from "react";
import { jsPDF } from "jspdf";

const ViewCGPA = ({ studentData }) => {
  const handleDownloadPDF = () => {
    if (studentData) {
      const doc = new jsPDF();
      doc.text("Student Details", 20, 20);
      doc.text(`Student Name: ${studentData.studentName}`, 20, 30);
      doc.text(`Roll Number: ${studentData.rollNumber}`, 20, 40);
      doc.text(`CGPA: ${studentData.cgpa}`, 20, 50);
      doc.save("student-details.pdf");
    } else {
      alert("No student data available to download.");
    }
  };

  return (
    <div>
      <h2>View Student CGPA</h2>
      {studentData ? (
        <div>
          <p><strong>Student Name:</strong> {studentData.studentName}</p>
          <p><strong>Roll Number:</strong> {studentData.rollNumber}</p>
          <p><strong>CGPA:</strong> {studentData.cgpa}</p>
          <button onClick={handleDownloadPDF} style={{ marginTop: "20px" }}>
            Download  PDF
          </button>
        </div>
      ) : (
        <p>No student data available. Please add CGPA first.</p>
      )}
    </div>
  );
};

export default ViewCGPA;
```

App.jsx

```
import React, { useState } from "react";
import { BrowserRouter as Router, Route, Routes, Link } from "react-router-dom";
import AddCGPA from "./components/AddCGPA";
```

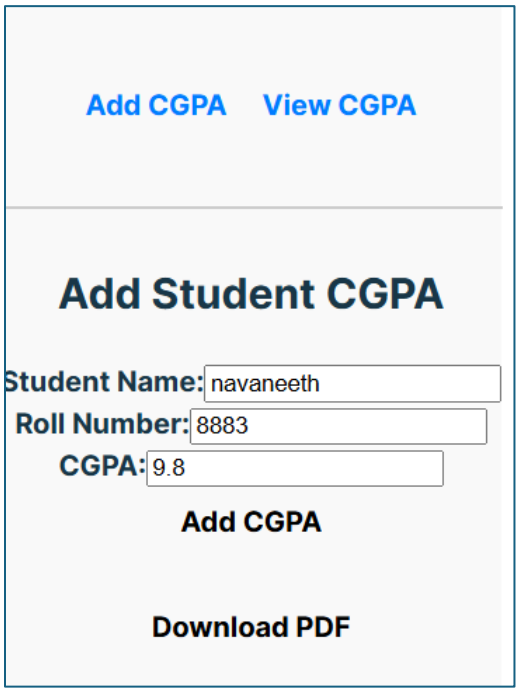
```
import ViewCGPA from "./components/ViewCGPA";

const App = () => {
  const [studentData, setStudentData] = useState(null); // State for storing student details

  return (
    <Router>
      <div
        style={{
          display: "flex",
          flexDirection: "column",
          alignItems: "center",
          justifyContent: "center",
          height: "100vh",
          textAlign: "center",
          backgroundColor: "#f9f9f9",
        }}
      >
        <nav style={{ marginBottom: "20px" }}>
          <ul
            style={{
              listStyleType: "none",
              padding: 0,
              display: "flex",
              gap: "20px",
            }}
          >
            <li>
              <Link
                to="/"
                style={{
                  textDecoration: "none",
                  color: "#007BFF",
                  fontWeight: "bold",
                }}
              >
                Add CGPA
              </Link>
            </li>
            <li>
              <Link
                to="/view-cgpa"
                style={{
                  textDecoration: "none",
                  color: "#007BFF",
                  fontWeight: "bold",
                }}
              >
                View CGPA
              </Link>
            </li>
```

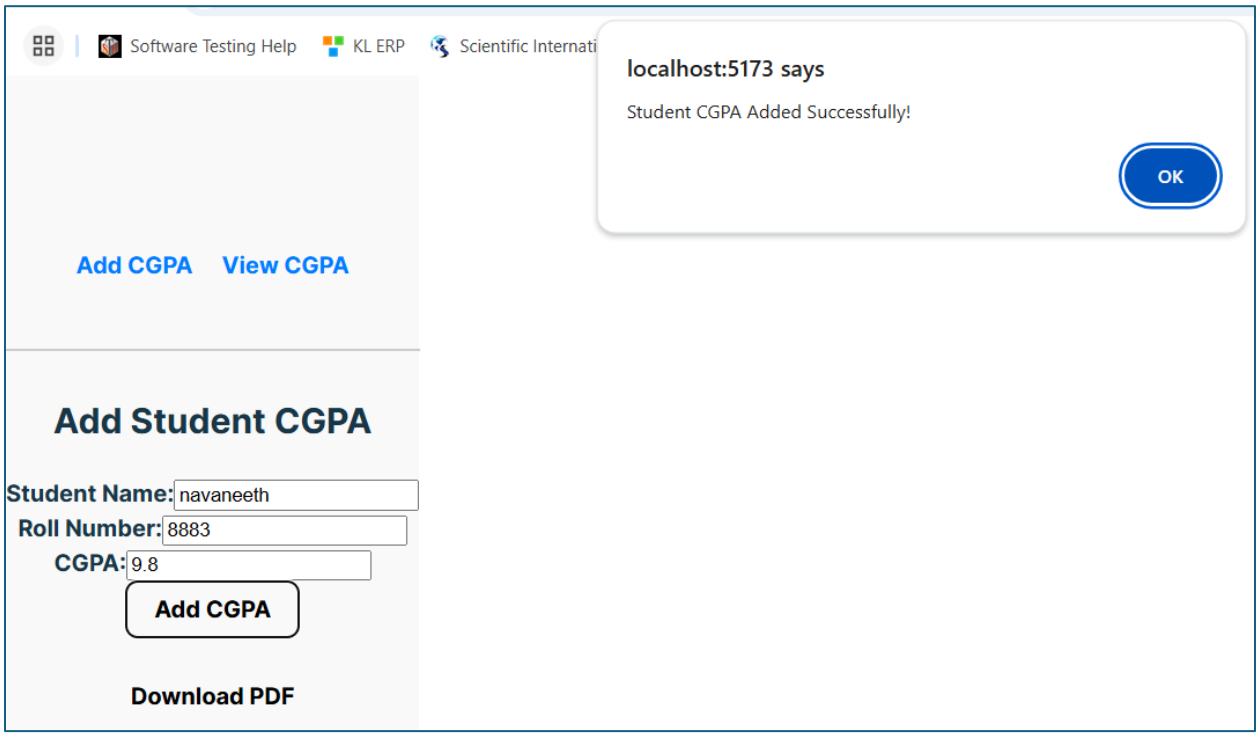
```
          </ul>
        </nav>
        <hr
          style={{
            width: "100%",
            maxWidth: "500px",
            border: "1px solid #ccc",
            margin: "10px 0",
          }}
        />

        <div style={{ width: "100%", maxWidth: "500px" }}>
          <Routes>
            <Route
              path="/"
              element={<AddCGPA setStudentData={setStudentData} />}
            />
            <Route
              path="/view-cgpa"
              element={<ViewCGPA studentData={studentData} />}
            />
          </Routes>
        </div>
      </div>
    </Router>
  );
};

export default App;
```

index.jsx

```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";

ReactDOM.render(<App />, document.getElementById("root"));
```

Output:


Add CGPA
View CGPA
Add Student CGPA
Student Name: navaneeth
Roll Number: 8883
CGPA: 9.8
Add CGPA
Download PDF


Software Testing Help
KL ERP
Scientific Internati
localhost:5173 says
Student CGPA Added Successfully!
OK
Add CGPA
View CGPA
Add Student CGPA
Student Name: navaneeth
Roll Number: 8883
CGPA: 9.8
Add CGPA
Download PDF

Add CGPA View CGPA

View Student CGPA

Student Name: navaneeth

Roll Number: 8883

CGPA: 9.8

Download PDF

student-details.pdf

Note:

In the realm of web development, React.js has emerged as a powerful tool for building dynamic user interfaces. One of its key features is React Router DOM, which facilitates navigation within a React application. In this article, we'll delve into what React Router DOM is, how it works, its significance, and practical examples to illustrate its usage.

What is React?

React is a JavaScript library developed by Facebook for building interactive user interfaces. It allows developers to create reusable UI components and efficiently manage the state of their applications.

Types of React Routers with Examples?

React Router is a powerful library for handling navigation and routing in React applications. Different types of React routers cater to various use cases and requirements. Let's explore each type with clear explanations and examples:

1. BrowserRouter:

- The BrowserRouter is the most commonly used router in React applications. It utilizes HTML5 history API to keep the UI in sync with the URL.
- Example:

```
import { BrowserRouter as Router, Route, Link } from 'react-router-dom';

const App = () => (
  <Router>
    <div>
      <ul>
        <li><Link to="/">Home</Link></li>
        <li><Link to="/about">About</Link></li>
        <li><Link to="/contact">Contact</Link></li>
      </ul>
      <Route exact path="/" component={Home} />
      <Route path="/about" component={About} />
      <Route path="/contact" component={Contact} />
    </div>
  </Router>
);
```

2. HashRouter:

- HashRouter uses the hash portion of the URL to keep UI in sync with the URL. It's suitable for environments where server-side rendering is not available.
- Example:

```
import { HashRouter as Router, Route, Link } from 'react-router-dom';

const App = () => (
  <Router>
    <div>
      <ul>
        <li><Link to="/">Home</Link></li>
        <li><Link to="/about">About</Link></li>
```

```
      <li><Link to="/contact">Contact</Link></li>
    </ul>
    <Route exact path="/" component={Home} />
    <Route path="/about" component={About} />
    <Route path="/contact" component={Contact} />
  </div>
 </Router>
);
```

3. MemoryRouter:

- MemoryRouter doesn't use the browser's URL but keeps the history of location transitions in memory. It's suitable for testing and non-browser environments.
- Example:

```
import { MemoryRouter as Router, Route, Link } from 'react-router-dom';

const App = () => (
 <Router>
  <div>
   <ul>
    <li><Link to="/">Home</Link></li>
    <li><Link to="/about">About</Link></li>
    <li><Link to="/contact">Contact</Link></li>
   </ul>
   <Route exact path="/" component={Home} />
   <Route path="/about" component={About} />
   <Route path="/contact" component={Contact} />
  </div>
 </Router>
);
```

4. NativeRouter:

- NativeRouter is designed for React Native applications. It uses the Native primitives for navigation instead of the browser's URL.
- Example:

```
import { NativeRouter as Router, Route, Link } from 'react-router-native';

const App = () => (
 <Router>
  <div>
   <ul>
    <li><Link to="/">Home</Link></li>
    <li><Link to="/about">About</Link></li>
    <li><Link to="/contact">Contact</Link></li>
   </ul>
   <Route exact path="/" component={Home} />
   <Route path="/about" component={About} />
   <Route path="/contact" component={Contact} />
  </div>
```

```
  </Router>
);
```

Each type of React Router provides a different approach to handling routing in React applications. Developers can choose the appropriate router based on their specific needs and constraints.

What is React Router DOM?

React Router DOM is a package that provides routing capabilities for React applications. It allows developers to define multiple routes in a single-page application (SPA) and handle navigation between them seamlessly. React Router DOM builds upon the core React Router library by adding support for the browser's DOM (Document Object Model).

Examples of React Router DOM:

Let's consider a simple example to demonstrate the usage of React Router DOM:

```
import React from 'react';
import { BrowserRouter as Router, Route, Switch, Link } from 'react-router-dom';
import Home from './Home';
import About from './About';
import Contact from './Contact';

const App = () => {
  return (
    <Router>
      <div>
        <nav>
          <ul>
            <li><Link to="/">Home</Link></li>
            <li><Link to="/about">About</Link></li>
            <li><Link to="/contact">Contact</Link></li>
          </ul>
        </nav>

        <Switch>
          <Route exact path="/">
            <Home />
          </Route>
          <Route path="/about">
            <About />
          </Route>
          <Route path="/contact">
            <Contact />
          </Route>
        </Switch>
      </div>
    </Router>
  );
};
```

export default App;

In this example, we define three different routes for the Home, About, and Contact pages using the Route component. The Link component is used for navigation between these routes, and the Switch component ensures that only one route is rendered at a time.

What are the Uses of React Router in React JS?

React Router is a powerful tool in the React.js ecosystem that enables developers to create single-page applications with dynamic routing. Here are some of its key uses:

1. Declarative Routing: React Router allows you to define the navigation paths of your application in a declarative way using JSX syntax. This makes it easy to understand and maintain the routing structure of your application.
2. Nested Routing: You can nest routes within each other, allowing for hierarchical URL structures that correspond to nested UI components. This is particularly useful for organizing complex applications with multiple levels of navigation.
3. Dynamic Routing: React Router supports dynamic routing, where route parameters can be extracted from the URL and passed as props to the rendered components. This enables building dynamic, data-driven UIs based on the current route.
4. History Management: React Router provides history management, allowing you to control the browser's history stack and navigate programmatically using methods like `push`, `replace`, and `goBack`. This is essential for implementing features like back-and-forward navigation, redirects, and deep linking.
5. Code Splitting and Lazy Loading: React Router seamlessly integrates with code-splitting solutions like React.lazy and Suspense, enabling you to load route components asynchronously. This can significantly improve the initial load time of your application by only loading the necessary code for the current route.
6. Authentication and Authorization: React Router can be used to implement authentication and authorization mechanisms by protecting certain routes from unauthorized access. You can conditionally render routes based on the user's authentication status or role, redirecting them to appropriate pages when necessary.
7. Integration with React Ecosystem: React Router is designed to work well with other popular libraries and frameworks in the React ecosystem, such as Redux for state management, React Helmet for managing document head tags, and React Transition Group for adding animations to route transitions.

Overall, React Router plays a crucial role in building modern, dynamic web applications with React.js by providing a flexible and intuitive way to handle client-side routing.

what are the 3 components of React router in React JS and their uses?

In React JS, React Router is a popular library for handling routing in single-page applications. It consists of three main components:

1. Browser Router: This component utilizes HTML5 history API to keep UI in sync with the URL.

Uses:

- Use BrowserRouter at the root level of your application to enable routing functionality.
- It provides the foundation for handling navigation and keeping UI in sync with the URL.
- Allows for a cleaner and more organized routing setup in your React application.

2. Route: Route is used to define a mapping between a URL path and a React component to render when the path matches.
Uses:

- Use Route to define mappings between URL paths and corresponding React components to render.
- It enables conditional rendering of components based on the current URL.
- Allows for dynamic content rendering based on the route accessed by the user.
- Supports additional props like exact, strict, and sensitive for more precise route matching.

3. Link: Link is a React component used to navigate between different routes in the application.
Uses:

- Use Link to create navigation links within your application.
- It prevents full-page refreshes when navigating between different routes.
- Provides a declarative way to define navigation links that update the URL and render the corresponding component.
- Improves user experience by enabling seamless navigation between different views or pages without losing the application state.

In summary, BrowserRouter sets up the routing environment, Route defines mappings between URLs and components, and Link provides navigation links for users to navigate between different routes in a React application.
These three components work together to enable navigation and routing within a React application, allowing developers to create dynamic and interactive user interfaces with multiple views or pages.

, React Router is a powerful library for handling routing in React applications, enabling developers to create dynamic and interactive single-page applications (SPAs). By using components like BrowserRouter, Route, and Link, developers can easily manage application navigation, render different components based on URL paths, and create seamless user experiences without full-page refreshes. React Router provides a clean and declarative way to implement routing functionality, allowing for organized and efficient navigation within React applications.