**Week 6**

**Hands-On Exercise 1: Set up a React environment and use create-react-app**

This week's first exercise introduced the foundational concepts of Single-Page Applications (SPAs) and ReactJS, along with the practical steps to set up your development environment.

**1. Single-Page Applications (SPAs)**

* **Definition:** An SPA is a web application that loads a single HTML page and dynamically updates content as the user interacts with the app, without requiring full page reloads from the server.

**2. React (ReactJS)**

* **Definition:** React is an open-source JavaScript library for building user interfaces, particularly single-page applications. It's maintained by Meta (Facebook) and a community of individual developers and companies.

**3. Virtual DOM**

* **Definition:** The Virtual DOM (Document Object Model) is a lightweight, in-memory representation of the actual browser DOM. It's a key feature that makes React fast.

***App.js:***

import React from 'react';

import './App.css';

function App() {

  return (

    <div className="centered">

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

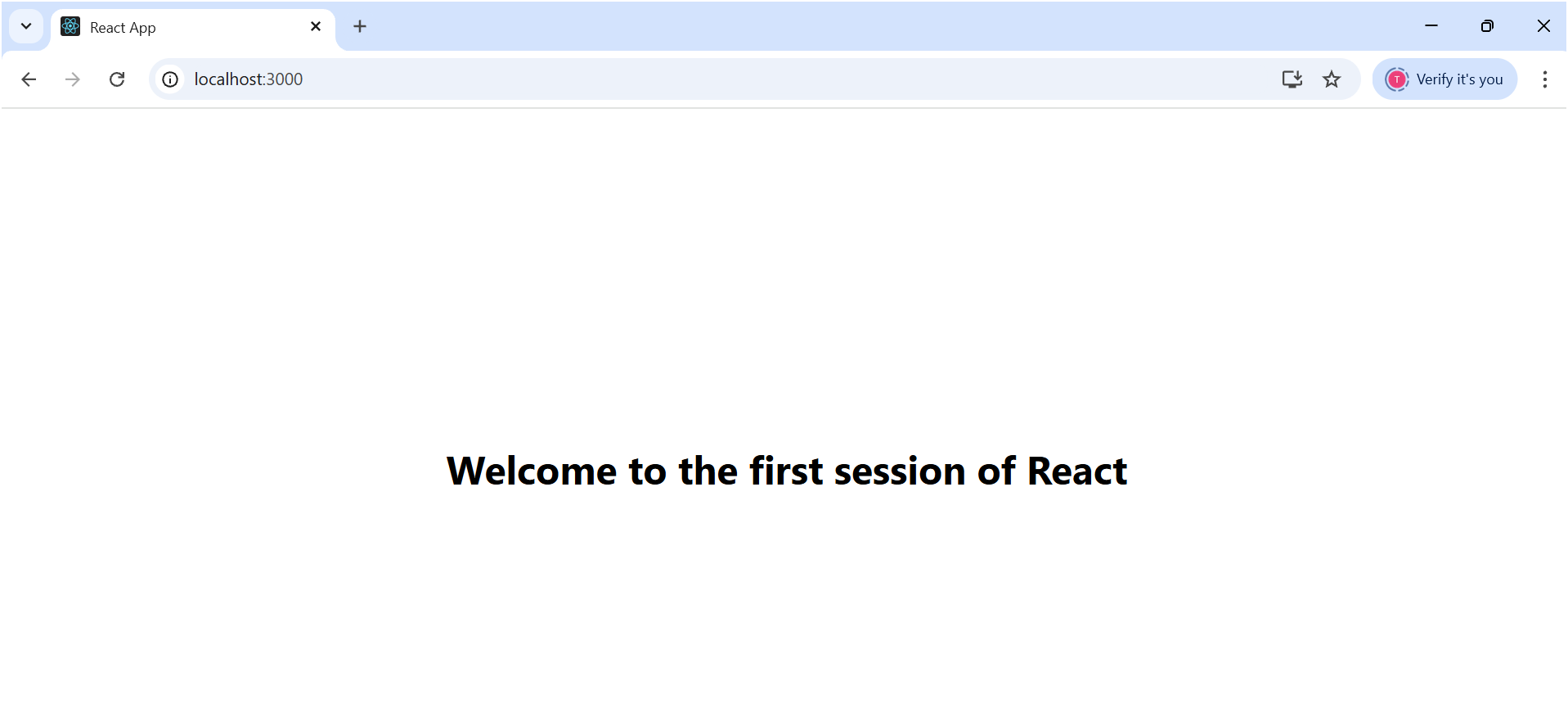
    </div>

  );

}

export default App;

***Output:***



**Hands-On Exercise 2: React Components**

**1. React Components: The Core Concept**

* **Definition:** In React, a component is an **independent, reusable, and self-contained piece of user interface (UI)**. Think of them as custom, isolated building blocks that encapsulate their own logic, appearance, and behavior.

**Scenario:** You need to create a React application for a "Student Management Portal" and display different messages using separate components for Home, About, and Contact pages.

**Code:**

**About.js:**

import React from 'react';

function About() {

  return (

    <div style={{ display: 'flex', justifyContent: 'center', alignItems: 'center', textAlign: 'center', minHeight: '200px' }}>

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

    </div>

  );

}

export default About;

**Contacts.js:**import React from 'react';

const Contacts = () => {

  return (

    <div style={{ display: 'flex', justifyContent: 'center', alignItems: 'center', textAlign: 'center', minHeight: '200px' }}>

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

    </div>

  );

};

export default Contacts;

**Home.js:**import React from 'react';

function Home() {

  return (

    <div style={{ display: 'flex', justifyContent: 'center', alignItems: 'center', textAlign: 'center', minHeight: '200px' }}>

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

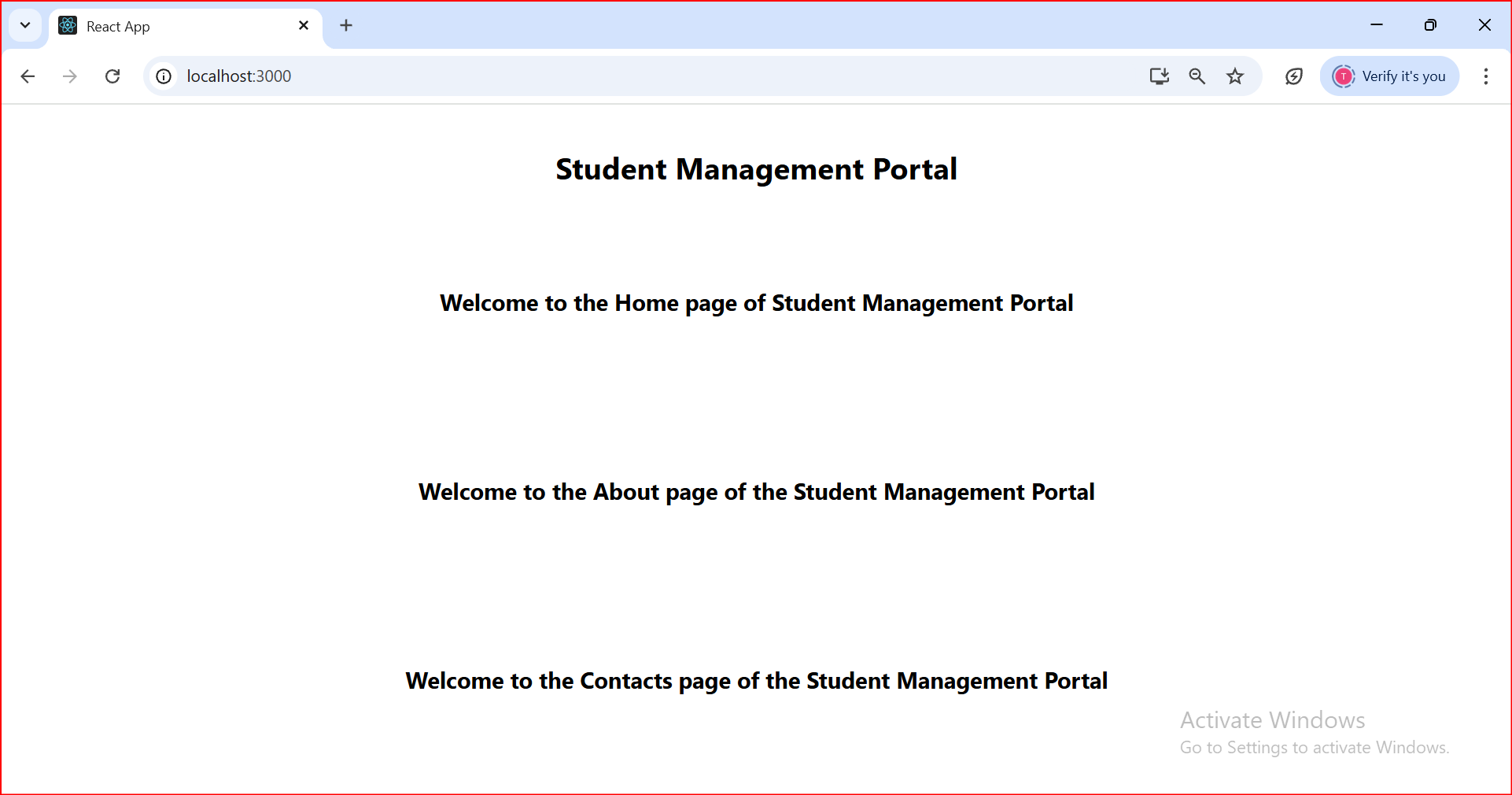
    </div>

  );

}

export default Home;

**Output:**



**Hands-On Exercise 3: React Components**

**Scenario:** You need to create a React application named scorecalculatorapp that includes a functional component CalculateScore. This component will accept student details (Name, School, Total, Goal) and display the calculated average score, with some basic styling.

**CalculateScore.js:**

    import React from 'react';

    function CalculateScore(props) {

      const { Name, School, Total, ScorePercentage } = props;

      return (

        <div className="student-details-card">

          <p style={{ color: 'blue' }}><b>Name:</b> {Name}</p>

          <p style={{ color: 'red' }}><b>School:</b> {School}</p>

          <p style={{ color: 'red' }}><b>Total:</b> {Total}Marks</p>

          <p style={{ color: 'green' }}><b>Score:</b> {ScorePercentage.toFixed(2)}%</p>

        </div>

      );

    }

    export default CalculateScore;

***App.js:***

    import React from 'react';

    import './App.css';

    import CalculateScore from './components/CalculateScore';

    function App() {

      const studentData = {

        Name: "Ravi",

        School: "BVBPS",

        Total: 540,

        ScorePercentage: (540 / 600) \* 100

      };

      return (

        <div className="App">

          <h1 style={{ color: 'red' }}>Student Details:</h1>

          <CalculateScore

            Name={studentData.Name}

            School={studentData.School}

            Total={studentData.Total}

            ScorePercentage={studentData.ScorePercentage}

          />

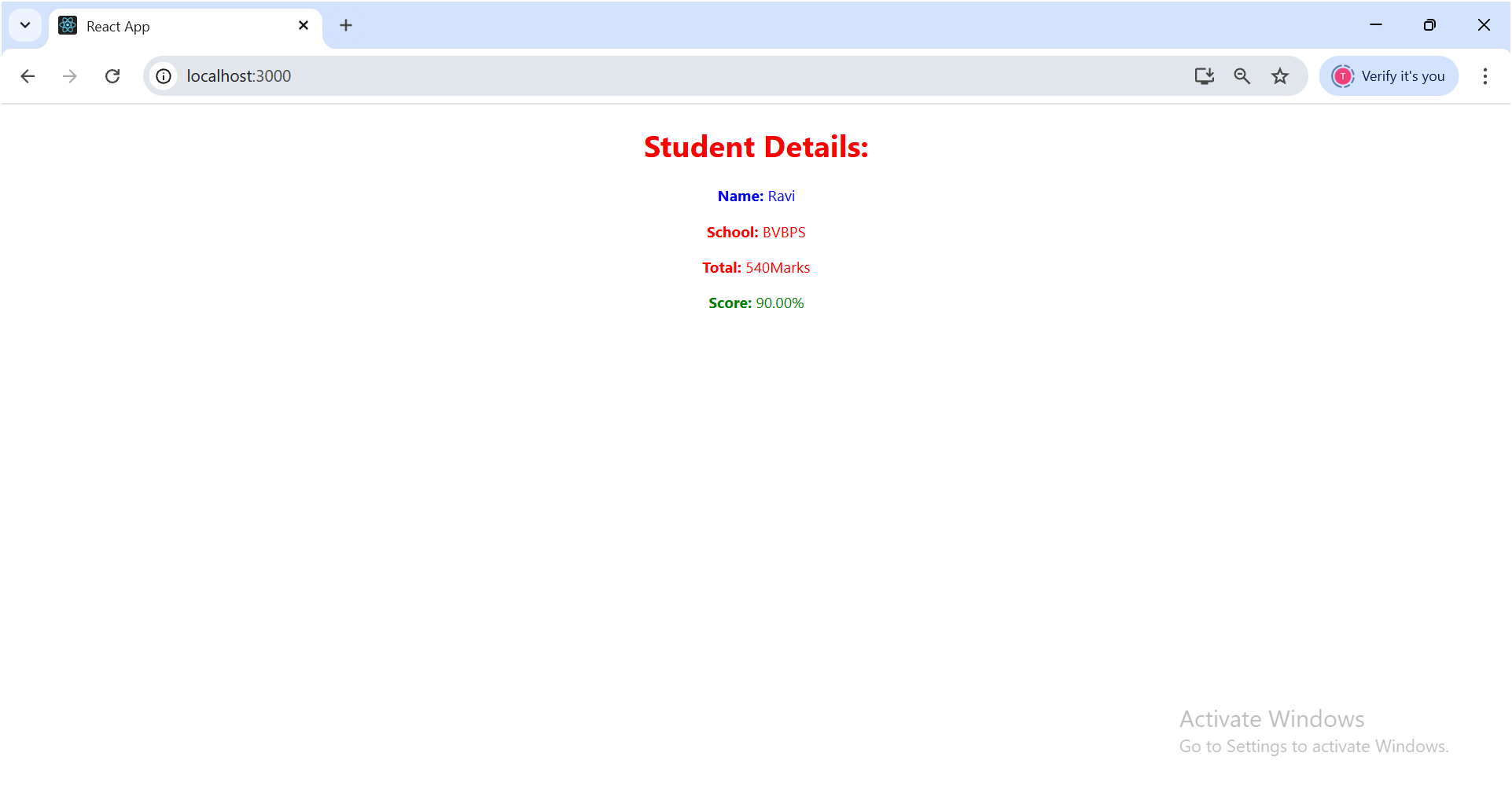
        </div>

      );

    }

    export default App;

**Output:**

****

**Hands-On Exercise 4:** **Component Life Cycle Hooks**

**Scenario:** You need to create a blog application that fetches a list of posts from an external API (jsonplaceholder.typicode.com) and displays them. You'll use React's component lifecycle methods to manage data fetching and error handling.

***Post.js:***

class Post {

      constructor(id, title, body) {

        this.id = id;

        this.title = title;

        this.body = body;

      }

    }

    export default Post;

***Posts.js:***

    import React from 'react';

    import Post from './Post';

    class Posts extends React.Component {

      constructor(props) {

        super(props);

        this.state = {

          posts: [],

          error: null,

          hasError: false

        };

      }

      loadPosts = async () => {

        try {

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

          if (!response.ok) {

            throw new Error(`HTTP error! status: ${response.status}`);

          }

          const data = await response.json();

          const fetchedPosts = data.map(post => new Post(post.id, post.title, post.body));

          this.setState({ posts: fetchedPosts });

          console.log("Posts fetched successfully:", fetchedPosts.length);

        } catch (error) {

          console.error("Error fetching posts:", error);

          this.setState({ hasError: true, error: error.message });

        }

      };

      componentDidMount() {

        console.log("Posts component did mount. Fetching posts...");

        this.loadPosts();

      }

      render() {

        if (this.state.hasError) {

          return <h1>Error: {this.state.error}</h1>;

        }

        if (this.state.posts.length === 0) {

          return <h1>Loading Posts...</h1>;

        }

        return (

          <div>

            <h1>Blog Posts</h1>

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

              <div key={post.id} style={{ border: '1px solid #ccc', padding: '10px', margin: '10px', borderRadius: '5px' }}>

                <h2>{post.title}</h2>

                <p>{post.body}</p>

              </div>

            ))}

          </div>

        );

      }

      componentDidCatch(error, info) {

        console.error("Error caught by componentDidCatch:", error, info);

        alert(`An error occurred in a component: ${error.message}\nComponent Stack: ${info.componentStack}`);

        this.setState({ hasError: true, error: error.message });

      }

    }

    export default Posts;

***Output:***

****

***Console Log:***  


**Hands-On Exercise 5:** **Styling React Components (CSS Modules and Inline Styles) - Adapted for your CohortDetails.js**

**Scenario:** Your Academy team at Cognizant wants a dashboard displaying details of ongoing and completed cohorts. You are tasked with styling these React components.

***Code:***

CohortDetails.module.css:  
    .box {

      width: 300px;

      display: inline-block;

      margin: 10px;

      padding-top: 10px;

      padding-bottom: 10px;

      padding-left: 20px;

      padding-right: 20px;

      border: 1px solid black;

      border-radius: 10px;

      box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

      background-color: #ffffff;

      vertical-align: top;

    }

    dt {

      font-weight: 500;

      margin-top: 8px;

      color: #555;

    }

    dd {

      margin-left: 0;

      margin-bottom: 5px;

      font-weight: normal;

    }

***CohortDetails.js:***

    import React from 'react';

    import styles from './components/CohortDetails.module.css';

    function CohortDetails(props) {

      const { cohort } = props;

      const nameStyle = {

        color: cohort.currentStatus === 'Ongoing' ? 'green' : 'blue'

      };

      return (

        <div className={styles.box}>

          <h3 style={nameStyle}>

            {cohort.cohortCode} - <span>{cohort.technology}</span>

          </h3>

          <dl>

            <dt>Started On</dt>

            <dd>{cohort.startDate}</dd>

            <dt>Current Status</dt>

            <dd>{cohort.currentStatus}</dd>

            <dt>Coach</dt>

            <dd>{cohort.coachName}</dd>

            <dt>Trainer</dt>

            <dd>{cohort.trainerName}</dd>

          </dl>

        </div>

      );

    }

    export default CohortDetails;

***App.js:***import logo from './logo.svg';

import './App.css';

import { CohortsData} from './Cohort'

import CohortDetails from './CohortDetails';

function App() {

  return (

  <div>

    <h1>Cohorts Details</h1>

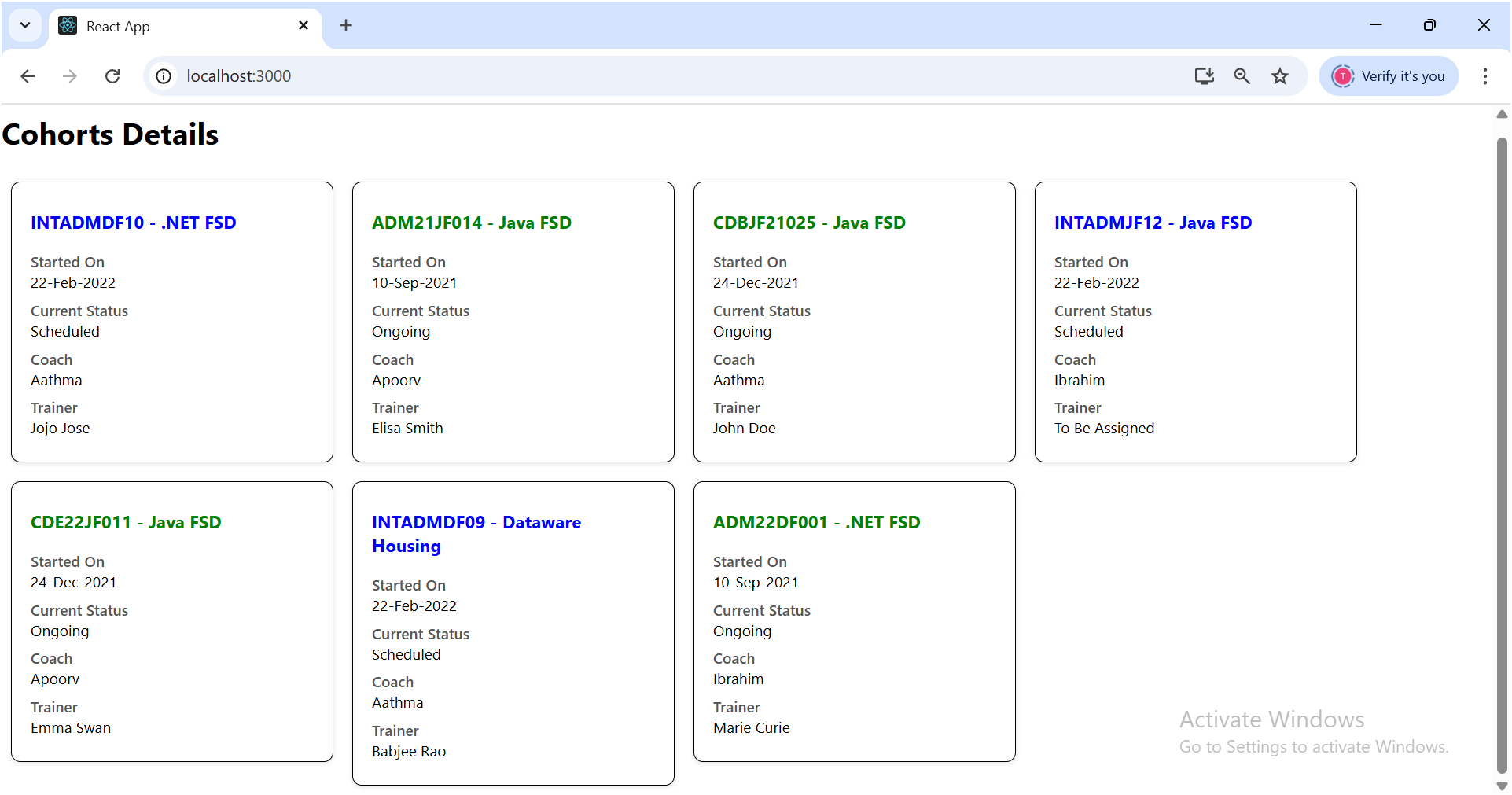
    {CohortsData.map(cohort => <CohortDetails cohort={cohort}/>)}

  </div>

  );

}

export default App;

***Output:  
***