WEEK – 6 REACT HANDSON

1.React\_HandsOn

### **What is a SPA?**

A **Single Page Application (SPA)** is a web application that dynamically rewrites the current page with new data from the server, instead of loading entire new pages.

### **What is React?**

**React** is an open-source JavaScript library developed by Facebook for building **user interfaces**, especially for SPAs. React uses components to break the UI into reusable pieces.

### **Virtual DOM**

The **Virtual DOM** is a lightweight JavaScript representation of the actual DOM. React updates the Virtual DOM first, then compares it with the previous version and makes only the necessary changes to the real DOM—this process is called **reconciliation**.

App .js

import React from 'react';

function App() {

return (

<div>

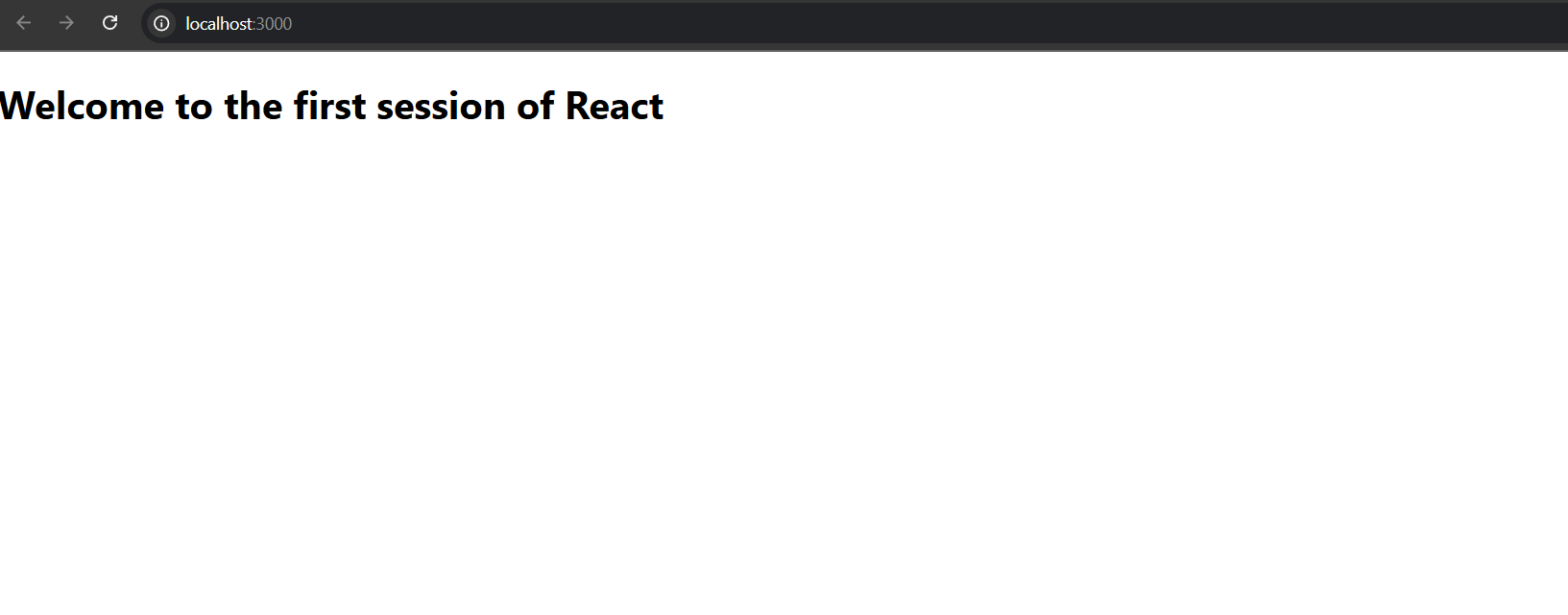
<h1>Welcome to the first session of React</h1>

</div>

);

}

export default App;



2.React\_HandOns

### **What is a React Component?**

A **React component** is a reusable block of code that defines how a part of the UI should appear and behave. Components can be reused and composed to build complex user interfaces.

### **Components vs JavaScript Functions**

|  |  |  |
| --- | --- | --- |
| **Feature** | **JavaScript Function** | **React Component** |
| Purpose | Performs a task or calculation | Renders UI |
| Return Value | Any type | JSX (UI structure) |
| Used In | Any JavaScript code | React applications |
| Lifecycle | No lifecycle | Has lifecycle methods (class components) |

### **Types of Components**

1. **Function Component** – A plain JavaScript function that returns JSX.
2. **Class Component** – Uses ES6 classes, and includes render() method and optionally lifecycle methods.

### **Class Component**

A component written using ES6 class. It can maintain its own state and lifecycle methods.

### **Component Constructor**

Used to initialize state or bind methods in class components.

### **render() Function**

Every class component must have a render() method. It returns JSX, the UI output.

Code

Components

About.js

import React, { Component } from 'react';

class About extends Component {

render() {

return (

<div>

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

</div>

);

}

}

export default About;

Contact.js

import React, { Component } from 'react';

class Contact extends Component {

render() {

return (

<div>

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

</div>

);

}

}

export default Contact;

Home.js

import React, { Component } from 'react';

class Home extends Component {

render() {

return (

<div>

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

</div>

);

}

}

export default Home;

App.js

import React from 'react';

import Home from './components/home';

import About from './components/about';

import Contact from './components/contact';

function App() {

return (

<div>

<Home />

<About />

<Contact />

</div>

);

}

export default App;

3.React\_Handson

Components

Calculatescore.js

import React from 'react';

import '../Stylesheets/mystyle.css';

function CalculateScore(props) {

const { name, school, total, goal } = props;

const average = total / goal;

return (

<div className="score-container">

<h2>Student Score Calculator</h2>

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

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

<p><strong>Total Marks:</strong> {total}</p>

<p><strong>Goal Subjects:</strong> {goal}</p>

<p className="result"><strong>Average Score:</strong> {average}</p>

</div>

);

}

export default CalculateScore;

App.js

import React from 'react';

import CalculateScore from './components/calculatescore';

function App() {

return (

<div>

<CalculateScore name="John Doe" school="ABC High School" total={450} goal={5} />

</div>

);

}

export default App;

Stylesheet.css

.score-container {

border: 2px solid #4CAF50;

padding: 20px;

margin: 30px auto;

width: 60%;

border-radius: 8px;

background-color: #f9f9f9;

box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.1);

font-family: Arial, sans-serif;

}

.score-container h2 {

color: #4CAF50;

text-align: center;

}

.score-container p {

font-size: 16px;

margin: 8px 0;

}

.result {

font-weight: bold;

color: #333;

}

4.React\_HandsOn

### **Why Lifecycle Methods?**

React components go through phases:

1. **Mounting** – Component is created and inserted into the DOM
2. **Updating** – Component gets re-rendered due to state/props changes
3. **Unmounting** – Component is removed from the DOM
4. **Error Handling** – React catches errors using error boundaries

Lifecycle methods let you **control code execution** during these phases.

### **Key Lifecycle Hooks**

|  |  |  |
| --- | --- | --- |
| **Phase** | **Method** | **Description** |
| Mounting | constructor() | Initializes state and binds methods |
|  | componentDidMount() | Runs after first render (good for API calls) |
| Updating | componentDidUpdate() | Runs after re-render |
| Unmounting | componentWillUnmount() | Cleanup tasks before component is removed |
| Error | componentDidCatch() | Catches rendering errors in child components |

### **Sequence of Mounting (Rendering)**

1. constructor()
2. render()
3. componentDidMount()

Post.js

import React from 'react';

class Post extends React.Component {

render() {

return (

<div style={{ border: "1px solid #ccc", marginBottom: "20px", padding: "10px" }}>

<h3>{this.props.title}</h3>

<p>{this.props.body}</p>

</div>

);

}

}

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: [],

hasError: false

};

}

// Step 6 – Define loadPosts method

loadPosts() {

fetch('https://jsonplaceholder.typicode.com/posts')

.then(res => res.json())

.then(data => this.setState({ posts: data.slice(0, 10) })) // First 10 posts

.catch(error => {

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

this.setState({ hasError: true });

});

}

// Step 7 – componentDidMount to load data

componentDidMount() {

this.loadPosts();

}

// Step 9 – Error Boundary

componentDidCatch(error, info) {

alert('Something went wrong while rendering the posts.');

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

}

// Step 8 – Render the Posts

render() {

if (this.state.hasError) {

return <h2>Error loading posts!</h2>;

}

return (

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

<h2>Blog Posts</h2>

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

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

))}

</div>

);

}

}

export default Posts;

App.js

import React from 'react';

import Posts from './posts';

function App() {

return (

<div className="App">

<Posts />

</div>

);

}

export default App;

5.React\_HandsOn

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

function CohortDetails(props) {

return (

<div className={styles.box}>

<h3 style={{ color: props.status === "ongoing" ? "green" : "blue" }}>

{props.cohort.cohortCode} -

<span>{props.cohort.technology}</span>

</h3>

<dl>

<dt>Started On</dt>

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

<dt>Current Status</dt>

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

<dt>Coach</dt>

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

<dt>Trainer</dt>

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

</dl>

</div>

);

}

export default CohortDetails;

/\* CohortDetails.module.css \*/

.box {

width: 300px;

display: inline-block;

margin: 10px;

padding: 10px 20px;

border: 1px solid black;

border-radius: 10px;

}

dt {

font-weight: 500;

}

