WEEK 6

1. ReactJS-HOL

- Define SPA and its benefits → A Single Page Application loads a single HTML page and updates content dynamically, making it faster and more interactive.
- Define React and identify its working → React is a JavaScript library that builds UIs by using components and a Virtual DOM for fast updates.
- Identify the differences between SPA and MPA → SPA loads one page dynamically, while MPA loads a new page from the server for each request.
- Explain Pros & Cons of SPA → SPA is fast and user-friendly but may have SEO challenges and larger initial load.
- Explain about React → React is a component-based library for creating dynamic web applications.
- Define virtual DOM → Virtual DOM is a lightweight copy of the real DOM that React updates efficiently.
- Explain Features of React → React is declarative, component-based, supports Virtual DOM, and is fast.

```
import React from "react";


const App = () => {
  return (
    <div style={{ textAlign: "center", marginTop: "50px" }}>
      <h1 style={{ color: "teal" }}>
        Welcome to the First React Learning Session!
      </h1>
    </div>
  );
};


export default App;
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";
```

```
const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

OUTPUT::

Welcome to the First React Learning Session!

2. ReactJS-HOL

- Explain React components → Components are reusable building blocks of a React application.
- Differences between components and JavaScript functions → Components return UI elements, while normal functions just perform tasks.
- Types of components → Class components and Function components.
- Explain class component → A class component is a React component defined using ES6 classes and lifecycle methods.
- Explain function component → A function component is a simple JavaScript function that returns JSX.
- Define component constructor → It initializes state and binds event handlers in class components.
- Define render() function → The render() method outputs the UI for the component.

Student App

```
import React from "react";

const Home = () => {
  return (
    <div>
      <h2>Welcome to the Home page of Student Management Portal</h2>
    </div>
  );
};

export default Home;
import React from "react";

const About = () => {
  return (
    <div>
      <h2>Welcome to the About page of the Student Management Portal</h2>
    </div>
  );
};

export default About;
import React from "react";

const Contact = () => {
  return (
    <div>
      <h2>Welcome to the Contact page of the Student Management Portal</h2>
    </div>
  );
};
```

```
export default Contact;
import React from "react";
import Home from "./Components/Home";
import About from "./Components/About";
import Contact from "./Components/Contact";

const App = () => {
  return (
    <div style={{ textAlign: "center", marginTop: "40px" }}>
      <h1>🏫 Student Management Portal</h1>
      <Home />
      <About />
      <Contact />
    </div>
  );
};

export default App;
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);
```

OUTPUT

Student Management Portal
Welcome to the Home page of Student Management Portal
Welcome to the About page of the Student Management Portal
Welcome to the Contact page of the Student Management Portal

3. ReactJS-HOL

Explain React components → Components are reusable building blocks of a React application.
Differences between components and JavaScript functions → Components return UI elements, while normal functions just perform tasks.
Types of components → Class components and Function components.
xplain class component → A class component is a React component defined using ES6 classes and lifecycle methods.
Explain function component → A function component is a simple JavaScript function that returns JSX.
Define component constructor → It initializes state and binds event handlers in class components.
Define render() function → The render() method outputs the UI for the component.

Score Calculator App

```
import React from "react";
import "../Stylesheets/mystyle.css";

const CalculateScore = ({ name, school, totalMarks, target }) => {
  const average = totalMarks / 5;

  return (
    <div className="score-card">
      <h2>Student Score Details</h2>
      <p><strong>Name:</strong> {name}</p>
      <p><strong>School:</strong> {school}</p>
      <p><strong>Total Marks:</strong> {totalMarks}</p>
      <p><strong>Goal Score:</strong> {target}</p>
      <p className="highlight">
        <strong>Average Score:</strong> {average.toFixed(2)}
      </p>
    </div>
  );
};

export default CalculateScore;
.score-card {
  border: 2px solid teal;
  padding: 20px;
  margin: 20px auto;
  width: 350px;
  border-radius: 10px;
  text-align: left;
  background-color: #f9f9f9;
}

.score-card h2 {
  text-align: center;
```

```
  color: teal;
}

.highlight {
  font-size: 18px;
  color: darkblue;
}
import React from "react";
import CalculateScore from "./Components/CalculateScore";

function App() {
  return (
    <div style={{ textAlign: "center" }}>
      <h1>🎓 Student Score Calculator</h1>
      <CalculateScore
        name="John Doe"
        school="Greenwood High"
        totalMarks={420}
        target={450}
      />
    </div>
  );
}

export default App;
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);
```

OUTPUT::
(http://localhost:3000)

Student Score Details
Name: John Doe
School: Greenwood High
Total Marks: 420
Goal Score: 450
Average Score: 84.00

4. ReactJS-HOL

- Need and Benefits of component life cycle → Lifecycle methods manage tasks like fetching data, updating UI, and cleaning up resources.
- Various life cycle hook methods → Examples include componentDidMount, componentDidUpdate, and componentWillUnmount.
- Sequence of steps in rendering a component → Constructor → Render → componentDidMount → Updates → componentWillUnmount.

Blogapp

```
import React from "react";

const Post = ({ title, body }) => {
  return (
    <div style={{ borderBottom: "1px solid #ccc", padding: "10px" }}>
      <h3>{title}</h3>
      <p>{body}</p>
    </div>
  );
};

export default Post;
import React, { Component } from "react";
import Post from "./Post";

class Posts extends Component {
  constructor(props) {
    super(props);
    this.state = {
      posts: [],
      error: null,
    };
  }

  // Fetch 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) }); // Taking only 5 posts for demo
    } catch (err) {
      this.setState({ error: err.message });
    }
  };

  componentDidMount() {
    this.loadPosts();
  }

  componentDidCatch(error, info) {
```

```
    alert("Something went wrong: " + error);
  }

  render() {
    return (
      <div style={{ textAlign: "center" }}>
        <h2>📝 Latest Blog Posts</h2>
        {this.state.posts.map((p) => (
          <Post key={p.id} title={p.title} body={p.body} />
        ))}
      </div>
    );
  }
}

export default Posts;
import React from "react";
import Posts from "./Posts";

function App() {
  return (
    <div>
      <h1 style={{ textAlign: "center", color: "teal" }}>My Blog App</h1>
      <Posts />
    </div>
  );
}

export default App;
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);
```

Output

Latest Blog Posts
Post Title 1
Post Body 1...

Post Title 2
Post Body 2...

Post Title 3
Post Body 3...

5. ReactJS-HOL

Need for styling React component → Styling improves UI appearance and user experience.

Working with CSS Module and inline styles → CSS Modules provide scoped styles, while inline styles are added directly in JSX.

CohortDashboard

```
.box {
  width: 300px;
  display: inline-block;
  margin: 10px;
  padding: 10px 20px;
  border: 1px solid black;
  border-radius: 10px;
  background-color: #f9f9f9;
}

dt {
  font-weight: 500;
}
import React from "react";
import styles from "./CohortDetails.module.css";

const CohortDetails = ({ name, status, mentor, duration }) => {
  const headingStyle = {
    color: status.toLowerCase() === "ongoing" ? "green" : "blue",
  };

  return (
    <div className={styles.box}>
      <h3 style={headingStyle}>{name}</h3>
      <dl>
        <dt>Status:</dt>
        <dd>{status}</dd>

        <dt>Mentor:</dt>
        <dd>{mentor}</dd>

        <dt>Duration:</dt>
        <dd>{duration}</dd>
      </dl>
    </div>
  );
};

export default CohortDetails;
import React from "react";
```

```
import CohortDetails from "./CohortDetails";

function App() {
  return (
    <div style={{ textAlign: "center" }}>
      <h1>🎓 My Academy Cohort Dashboard</h1>
      <CohortDetails
        name="React Bootcamp"
        status="Ongoing"
        mentor="John Smith"
        duration="6 Weeks"
      />
      <CohortDetails
        name="AI Fundamentals"
        status="Completed"
        mentor="Sarah Lee"
        duration="4 Weeks"
      />
    </div>
  );
}

export default App;
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);
```

OUTPUT::

React Bootcamp – Ongoing
- Status: Ongoing
- Mentor: John Smith
- Duration: 6 Weeks

AI Fundamentals – Completed
- Status: Completed
- Mentor: Sarah Lee
- Duration: 4 Weeks

Additional important hands on

6. ReactJS-HOL

Need and benefits of React Router → React Router enables navigation in single-page applications without reloading.
Components in React Router → BrowserRouter, Routes, Route, Link, useParams, etc. Types of Router Components → BrowserRouter, HashRouter, MemoryRouter.
Parameter passing via URL → Parameters can be passed using dynamic routes like /trainer/:id.

TrainersApp

```
class Trainer {
  constructor(id, name, email, phone, technology, skills) {
    this.id = id;
    this.name = name;
    this.email = email;
    this.phone = phone;
    this.technology = technology;
    this.skills = skills;
  }
}

export default Trainer;

import Trainer from "./Trainer";

const trainers = [
  new Trainer(
    1,
    "John Doe",
    "john@example.com",
    "9876543210",
    "React",
    "Hooks, Redux, Routing"
  ),
  new Trainer(
    2,
    "Sarah Lee",
    "sarah@example.com",
    "9123456780",
    "Node.js",
    "Express, MongoDB, APIs"
  ),
  new Trainer(
    3,
    "David Smith",
```

```
    "david@example.com",
    "9988776655",
    "Python",
    "Django, Flask, ML"
  ),
];

export default trainers;

import React from "react";
import { Link } from "react-router-dom";

const TrainersList = ({ trainers }) => {
  return (
    <div>
      <h2>Available Trainers</h2>
      <ul>
        {trainers.map((t) => (
          <li key={t.id}>
            <Link to={`/trainer/${t.id}`}>{t.name}</Link>
          </li>
        ))}
      </ul>
    </div>
  );
};

export default TrainersList;
import React from "react";
import { useParams } from "react-router-dom";
import trainers from "./TrainersMock";

const TrainerDetails = () => {
  const { id } = useParams();
  const trainer = trainers.find((t) => t.id.toString() === id);

  if (!trainer) return <p>Trainer not found</p>;

  return (
    <div>
      <h2>{trainer.name}</h2>
      <p><strong>Email:</strong> {trainer.email}</p>
      <p><strong>Phone:</strong> {trainer.phone}</p>
      <p><strong>Technology:</strong> {trainer.technology}</p>
      <p><strong>Skills:</strong> {trainer.skills}</p>
    </div>
  );
};

export default TrainerDetails;
```

```
import React from "react";

const Home = () => {
  return (
    <div>
      <h2>Welcome to Trainers Management Portal</h2>
      <p>Select "Trainers" to view the list of available trainers.</p>
    </div>
  );
};

export default Home;
import React from "react";
import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";
import Home from "./Home";
import TrainersList from "./TrainersList";
import TrainerDetails from "./TrainerDetails";
import trainers from "./TrainersMock";

function App() {
  return (
    <Router>
      <div style={{ textAlign: "center" }}>
        <h1>Trainer Management App</h1>
        <nav>
          <Link to="/">Home</Link> | <Link to="/trainers">Trainers</Link>
        </nav>
        <hr />
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/trainers" element={<TrainersList trainers={trainers} />} />
          <Route path="/trainer/:id" element={<TrainerDetails />} />
        </Routes>
      </div>
    </Router>
  );
}

export default App;
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);
```

OUTPUT::
Home Page

Trainer Management App
Welcome to Trainers Management Portal
Select "Trainers" to view the list of available trainers.

Trainers List Page

Trainer Management App

Available Trainers:

- John Doe (clickable)

- Sarah Lee (clickable)

- David Smith (clickable)

Trainers Details Page

Trainer Management App

John Doe

Email: john@example.com

Phone: 9876543210

Technology: React

Skills: Hooks, Redux, Routing

7. ReactJS-HOL

Define Props → Props are inputs passed to components to display dynamic data.
Explain Default Props → Default props provide fallback values if no props are passed.
Differences between State and Props → State is internal and mutable, props are external ad read-only.
Explain reactDOM.render() → This method renders a React component into the DOM.

Shoppingapp

```
import React from "react";

class Cart extends React.Component {
  render() {
    const { itemName, price } = this.props;
    return (
      <div style={{ border: "1px solid gray", padding: "10px", margin: "10px" }}>
        <h3>{itemName}</h3>
        <p>Price: ₹{price}</p>
      </div>
    );
  }
}

export default Cart;
import React from "react";
import Cart from "./Cart";

class OnlineShopping extends React.Component {
  render() {
    const products = [
      { id: 1, itemName: "Laptop", price: 55000 },
      { id: 2, itemName: "Smartphone", price: 20000 },
      { id: 3, itemName: "Headphones", price: 3000 },
      { id: 4, itemName: "Smartwatch", price: 7000 },
      { id: 5, itemName: "Bluetooth Speaker", price: 4000 },
    ];

    return (
      <div>
        <h2>🛒 Online Shopping Cart</h2>
        {products.map((item) => (
          <Cart key={item.id} itemName={item.itemName} price={item.price} />
        ))}
      </div>
    );
  }
}
```

```
export default OnlineShopping;
import React from "react";
import OnlineShopping from "./OnlineShopping";

function App() {
  return (
    <div style={{ textAlign: "center" }}>
      <h1>Shopping Application</h1>
      <OnlineShopping />
    </div>
  );
}

export default App;
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);
```

OUTPUT::

Shopping Application
Online Shopping Cart

Laptop
Price: ₹55000

Smartphone
Price: ₹20000

Headphones
Price: ₹3000

Smartwatch
Price: ₹7000

Bluetooth Speaker
Price: ₹4000

8. ReactJS-HOL

Explain React State → State is an object that stores dynamic data and triggers re-rendering when updated.

Counterapp

```
import React from "react";

class CountPeople extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      entryCount: 0,
      exitCount: 0,
    };
  }

  updateEntry = () => {
    this.setState((prevState) => ({
      entryCount: prevState.entryCount + 1,
    }));
  };

  updateExit = () => {
    this.setState((prevState) => ({
      exitCount: prevState.exitCount + 1,
    }));
  };

  render() {
    return (
      <div style={{ textAlign: "center", marginTop: "50px" }}>
        <h2>🏢 Mall Entry & Exit Counter</h2>
        <p>People Entered: {this.state.entryCount}</p>
        <p>People Exited: {this.state.exitCount}</p>

        <button onClick={this.updateEntry} style={{ marginRight: "10px" }}>
          Login
        </button>
        <button onClick={this.updateExit}>Exit</button>
      </div>
    );
  }
}

export default CountPeople;
import React from "react";
import CountPeople from "./CountPeople";
```

```
function App() {
  return (
    <div>
      <h1 style={{ textAlign: "center" }}>Counter Application</h1>
      <CountPeople />
    </div>
  );
}

export default App;
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);
```

OUTPUT::

Counter Application
Mall Entry & Exit Counter
People Entered: 0
People Exited: 0
[Login] [Exit]