**React JS**

**Handson – 1 – My First React App**

**• Define SPA and its benefits**A Single Page Application (SPA) loads a single HTML page and dynamically updates content without reloading.  
Benefits include faster navigation, better user experience, and reduced server load.

**• Define React and identify its working**React is a JavaScript library for building user interfaces using components.  
It updates the UI efficiently using a virtual DOM and state management.

**• Identify the differences between SPA and MPA**SPA loads one HTML page and updates content dynamically; MPA loads multiple pages from the server.  
SPA is faster and smoother; MPA is better for SEO and complex sites.

**• Explain Pros & Cons of Single-Page Application**Pros: Fast performance, seamless UX, less server load.  
Cons: Poor SEO, initial load time high, harder to secure.

**• Explain about React**React is a component-based JavaScript library maintained by Meta for building UIs.  
It uses JSX syntax and virtual DOM to render updates efficiently.

**• Define virtual DOM**The virtual DOM is a lightweight JavaScript copy of the real DOM.  
React compares it with the actual DOM and updates only the changed parts.

**• Explain Features of React**React offers component-based architecture, virtual DOM, unidirectional data flow, and JSX.  
It supports reusable code, fast rendering.

**1. Created a new React Application with the name “myfirstreact”.**

**App.js:**

function App()

{

   return <h1>Welcome to my first react app</h1>

}

export default App;

**A screenshot of a computer

AI-generated content may be incorrect.**

**Handson – 2 - StudentApp**

**• Explain React components**React components are reusable pieces of UI that return JSX. They manage their own state and props.

• **Identify the differences between components and JavaScript functions**  
Components return JSX and follow React rules, while regular JS functions return values and don’t handle UI rendering. Components can have lifecycle methods; functions don’t.

**• Identify the types of components**  
There are two main types: Class components and Function components.

• **Explain class component**  
Class components extend React.Component and include lifecycle methods and state. They use a render() method to return JSX.

• **Explain function component**  
Function components are simple JavaScript functions that return JSX. They can use hooks for state and side effects.

• **Define component constructor**  
The constructor is used in class components to initialize state and bind methods. It’s called once before the component is mounted.

• **Define render() function**  
The render() method returns the JSX to be displayed on the UI. It's required in class components.

**1. Created new react application with name “studentapp”.**

**2. Created Components for Home, About and Contact pages.**

**Home.js**

import React, {Component} from "react";

class Home extends Component{

    render()

    {  return(

       <div>

        <h3>Welcome to the Home Page of Student Management System</h3>

       </div>

    )

    }

}

export default Home;’

**About.js**

import React, {Component} from 'react';

class About extends Component

{

    render() { return(

    <div>

        <h1>Welcome  to About Page of Student Management System</h1>

        </div>)}

}

export default About;

**Contact.js**

import React,{Component} from 'react';

class Contact extends Component

{

    render()

    {

        return(<div> <h1>Welcome to Contact Page of Student Management System</h1></div>)

    }

}

export default Contact;

**3. Imported those components into App.js and created instances for them in JSX.**

**App.js**

import React,{Component} from 'react';

class Contact extends Component

{

    render()

    {

        return(<div> <h1>Welcome to Contact Page of Student Management System</h1></div>)

    }

}

export default Contact;

A screenshot of a computer

AI-generated content may be incorrect.

**Handson – 3 – Score Calculator App**

**1. Created scorecalculatorapp**

**2. Created CalculateScore.js**

import './stylesheets/mystyle.css';

const decimalToPercent= (decimal)=>

{

    return (decimal.toFixed(2) + '%');

}

const calScore = (total, goal) =>

{

    return decimalToPercent(total/goal);

}

const CalculateScore = ({Name, School, total, goal}) =>

(

<div className='form\_style'>

    <h1><font color ="Brown">Student Details</font></h1>

    <div className='Name'>

        <b><span>Name: </span></b>

        <span>{Name}</span>

    </div>

    <div className='School'>

        <b><span>School: </span></b>

        <span>{School}</span>

    </div>

    <div className='Total'>

        <b><span>Total: </span></b>

        <span>{total} Marks</span>

    </div>

    <div className='Score'>

        <b>

            <span>

             Score: {calScore(total,goal)}

            </span>

        </b>

    </div>

</div>

);

export default CalculateScore;

**App.js**

import CalculateScore from "./components/CalculateScore";

function App()

{

  return (

    <CalculateScore Name={"Hemanth"}

    School={"SDR"}

    total={284}

    goal={3}/>

  )

}

export default App;

**A screenshot of a computer

AI-generated content may be incorrect.**

**Handson – 4 – Blog App**

**Explain the need and Benefits of component life cycle**The component life cycle helps manage component creation, updates, and removal. It allows better control over rendering, data fetching, and cleanup.

**• Identify various life cycle hook methods**  
Key lifecycle methods include componentDidMount, componentDidUpdate, componentWillUnmount, and componentDidCatch.

**• List the sequence of steps in rendering a component**  
The sequence is: Mounting → Updating → Unmounting → Error Handling .  
During mounting: constructor → render() → componentDidMount().

**1. Created a new react application using *create-react-app* tool with the name as “blogapp”.**

**2. Created a new file named as Post.js in src folder with following properties**

class Post

{

    constructor(id, title, body)

    {

        this.id=id;

        this.title = title;

        this.body=body;

    }

}

export default Post;

**3. Created a new class based component named as Posts inside Posts.js file**

import React from 'react';

import Post from './Post'

class Posts extends React.Component{

    constructor(props)

    {

        super(props);

        this.state={posts: [],}

    }

    loadPosts()

    {

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

        .then(response => response.json())

        .then(data=>{

            console.log("received "+data)

            const postObject = data.map(p=>new Post(p.id, p.title, p.body))

            this.setState({posts: postObject})

        }

        )

        .catch(error=>{

            console.log(error)

        })

    }

    componentDidMount(){

       this.loadPosts();

    }

     componentDidCatch(error, info)

    {

         alert("something went wrong: "+error.message)

    }

    render()

    {

       return(

    <div>

          <h1>Posts</h1>

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

            <div key={post.id}>

                <h3>{post.title}</h3>

                <p>{post.body}</p>

            </div>

          )}

    </div>

       );

    }

}

export default Posts;

**App.js**

import Posts from './Posts';

const App=()=>

(

   <div>

    <Posts />

   </div>

)

export default App;

A screenshot of a computer

AI-generated content may be incorrect.

**Handson – 5 – Cohort Tracker**

**CohortDetais.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;

}

dl{

    font-weight: 500;

}

**CohortDetails.js**

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

function CohortDetails(props) {

    return (

        <div className={style.box}>

        <h3 style={{ color: props.cohort.currentStatus === '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;

**Cohort.js**

class Cohort {

    constructor(cohortCode,

        startDate,

        technology,

        trainerName,

        coachName,

        currentStatus) {

        this.cohortCode = cohortCode;

        this.coachName = coachName;

        this.trainerName = trainerName;

        this.technology = technology;

        this.startDate = startDate;

        this.currentStatus = currentStatus;

    }

}

const CohortsData =[

    new Cohort('INTADMDF10','22-Feb-2022', '.NET FSD', 'Jojo Jose','Aathma', 'Scheduled'),

    new Cohort('ADM21JF014','10-Sep-2021', 'Java FSD', 'Elisa Smith','Apoorv', 'Ongoing'),

    new Cohort('CDBJF21025','24-Dec-2021', 'Java FSD', 'John Doe','Aathma', 'Ongoing'),

    new Cohort('INTADMJF12','22-Feb-2022', 'Java FSD', 'To Be Assigned','Ibrahim', 'Scheduled'),

    new Cohort('CDE22JF011','24-Dec-2021', 'Java FSD', 'Emma Swan','Apoorv', 'Ongoing'),

    new Cohort('INTADMDF09','22-Feb-2022', 'Dataware Housing', 'Babjee Rao','Aathma', 'Scheduled'),

    new Cohort('ADM22DF001','10-Sep-2021', '.NET FSD', 'Marie Curie','Ibrahim', 'Ongoing'),

];

export {Cohort, CohortsData};

**App.js**

import './App.css';

import { CohortsData} from './Cohort'

import CohortDetails from './CohortDetails';

import './CohortDetails.module.css'

function App() {

  return (

  <div>

    <h1>Cohorts Details</h1>

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

  </div>

  );

}

export default App;

A screenshot of a computer

AI-generated content may be incorrect.

**Handson – 6 Trainers App**

**Trainer.js**

class Trainer{

    constructor(TrainerId, Name, Email, Phone, Technology, Skills)

    {

        this.TrainerId=TrainerId;

        this.Email=Email;

        this.Name=Name;

        this.Phone=Phone;

        this.Technology=Technology;

        this.Skills=Skills;

    }

}

export default Trainer;

**TrainerMock.js**

import Trainer from './Trainer';

 const TrainerMock=[

    new Trainer('t-syed8','Syed Khaleelullah','khaleelullah@cognizant.com','97676516962','.NET',['C#','SQL Server','React','.NET core']),

    new Trainer('t-jojo','Jojo Jose','ojoj@cognizant.com','9897199231','Java',['Java','JSP','Angular','Spring']),

    new Trainer('t-elisa','Elisa Jones','elisa@cognizant.com','9871212235','Python',['Python','DJango','Angular']),

 ]

export default TrainerMock;

**TrainerList.js**

import {Routes, BrowserRouter,Route, Link} from 'react-router-dom';

import TrainerDetails from './TrainerDetails'

function TrainerList ({trainers})

{

    return(

            <div>

                <h3>Trainer List</h3>

                <div>

                <ul>{trainers.map((trainer,index)=>

                    <li key={index}>

                        <Link to={`/trainers/${trainer.TrainerId}`}>{trainer.Name}</Link>

                    </li>)}

                </ul>

</div>

            </div>

    )

}

export default TrainerList;

**TrainerDetails.js**

import { useParams } from "react-router-dom";

import TrainerMock from './TrainerMock';

const TrainerDetails =()=>

{

    const {id} = useParams();

    const trainer = TrainerMock.find(t=>t.TrainerId===id)

    if(!trainer)

    {

        return <h2>Trainer not found</h2>

    }

    return(

        <div>

             <h2>Trainer Details</h2>

            <p><strong>{trainer.Name} ({trainer.Technology})</strong> </p>

            <p> {trainer.Email}</p>

            <p> {trainer.Phone}</p>

            <ul> {trainer.Skills.map((skill,index)=>

                <li key={index}>{skill}</li>

            )}</ul>

        </div>

    )

}

export default TrainerDetails;

**Home.js**

const Home=()=>

(

    <div>

        <h1>Welcome to My Academy Trainers Page</h1>

    </div>

)

export default Home;

**App.js**

import TrainerList from './TrainerList';

import TrainerMock from './TrainerMock'

import Home from './Home';

import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';

import TrainerDetails from './TrainerDetails';

const App = ()=>

{

    return(

        <div>

        <BrowserRouter>

        <div>

               <nav>

                <ul>

                    <li><Link to='/'>Home</Link></li>

                    <li><Link to='/trainers'>Show Trainers</Link></li>

                </ul>

                </nav>

        </div>

        <Routes>

            <Route path = '/' element={<Home/>}/>

                <Route path ='/trainers' element={<TrainerList trainers={TrainerMock}/>}/>

               <Route path = {"/trainers/:id"} element = {<TrainerDetails/>}/>

        </Routes>

        </BrowserRouter>

        </div>

    )

}

export default App;

A blue and white background with black text

AI-generated content may be incorrect.

A blue and white rectangular object

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.