**Week 6**

**React-Mandatory and Additional Exercise**

**1. ReactJS-HOL**

**App.js**  
import React from 'react';

function App() {

return (

<div>

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

</div>

);

}

export default App;

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**2. ReactJS-HOL**

**Create a react app for Student Management Portal named StudentApp and create a component named Home which will display the Message “Welcome to the Home page of Student Management Portal”. Create another component named About and display the Message “Welcome to the About page of the Student Management Portal”. Create a third component named Contact and display the Message “Welcome to the Contact page of the Student Management Portal”. Call all the three components.**

**Home.js:**

import React from 'react';

function Home() {

return (

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

);

}

export default Home;

**About.js:**

import React from 'react';

function About() {

return (

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

);

}

export default About;

**Contact.js:**

import React from 'react';

function Contact() {

return (

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

);

}

export default Contact;

**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 style={{ textAlign: 'center' }}>

<Home />

<About />

<Contact />

</div>

);

}

export default App;

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**3. ReactJS-HOL**

**Create a react app for Student Management Portal named scorecalculatorapp and create a function component named “CalculateScore” which will accept Name, School, Total and goal in order to calculate the average score of a student and display the same.**

**CalculateScore.js:**

import React from 'react';

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

function CalculateScore(props) {

const average = ((props.total / props.goal) \* 100).toFixed(2);

return (

<div className="main">

<h1 className="heading">Student Details:</h1>

<p><span className="label">Name:</span> <span className="value name">{props.name}</span></p>

<p><span className="label school">School:</span> DNV Public School</p>

<p><span className="label total">Total:</span> {props.total} Marks</p>

<p><span className="label score">Score:</span>{average}%</p>

</div>

);

}

export default CalculateScore;

**mystyle.css:**

.main {

text-align: center;

margin-top: 40px;

font-family: Arial, sans-serif;

}

.heading {

color: brown;

}

.label {

font-weight: bold;

}

.name {

color: blue;

}

.school {

color: red;

}

.total {

color: purple;

}

.score {

color: green;

}

**App.js:**

import React from 'react';

import CalculateScore from './Components/CalculateScore';

function App() {

return (

<div>

<CalculateScore name="Steeve" total={284} goal={300} />

</div>

);

}

export default App;

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**4. ReactJS-HOL**

**Post.js:**

class Post {

constructor(userId, id, title, body) {

this.userId = userId;

this.id = id;

this.title = title;

this.body = body;

}

}

export default Post;

**Posts.js:**

import React, { Component } from 'react';

import Post from './Post';

class Posts extends Component {

constructor(props) {

super(props);

this.state = {

posts: [],

error: null

};

}

loadPosts() {

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

.then(response => {

if (!response.ok) {

throw new Error("Failed to fetch posts");

}

return response.json();

})

.then(data => {

const posts = data.map(

p => new Post(p.userId, p.id, p.title, p.body)

);

this.setState({ posts });

})

.catch(error => {

this.setState({ error });

});

}

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

alert("An error occurred: " + error);

}

render() {

const { posts } = this.state;

return (

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

<h1>Blog Posts</h1>

{

posts.length === 0 ? (

<p>Loading posts...</p>

) : (

posts.map(post => (

<div key={post.id} style={{ marginBottom: '20px' }}>

<h3>{post.title}</h3>

<p>{post.body}</p>

</div>

))

)

}

</div>

);

}

}

export default Posts;

**App.js:**

import React from 'react';

import Posts from './Posts';

function App() {

return (

<div>

<Posts />

</div>

);

}

export default App;

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**5. ReactJS-HOL**

**My Academy team at Cognizant want to create a dashboard containing the details of ongoing and completed cohorts. A react application is created which displays the detail of the cohorts using react component. You are assigned the task of styling these react components.**

**CohortDetails.module.css:**

.box {

width: 300px;

display: inline-block;

margin: 10px;

padding: 10px 20px;

border: 1px solid black;

border-radius: 10px;

vertical-align: top;

}

dt {

font-weight: 500;

}

**CohortDetails.js:**

import React from 'react';

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

function CohortDetails({ cohort }) {

return (

<div className={styles.box}>

<h3 style={{ color: cohort.status === 'Ongoing' ? 'green' : 'blue' }}>

{cohort.name}

</h3>

<dl>

<dt>Started On</dt>

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

<dt>Current Status</dt>

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

<dt>Coach</dt>

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

<dt>Trainer</dt>

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

</dl>

</div>

);

}

export default CohortDetails;

**App.js:**

import React from 'react';

import CohortDetails from './components/CohortDetails';

const cohortList = [

{

name: 'INTADMDF10 - .NET FSD',

startDate: '22-Feb-2022',

status: 'Scheduled',

coach: 'Aathma',

trainer: 'Jojo Jose'

},

{

name: 'ADM21JF014 - Java FSD',

startDate: '10-Sep-2021',

status: 'Ongoing',

coach: 'Apoorv',

trainer: 'Elisa Smith'

},

{

name: 'CDBJF21025 - Java FSD',

startDate: '24-Dec-2021',

status: 'Ongoing',

coach: 'Aathma',

trainer: 'John Doe'

}

];

function App() {

return (

<div>

<h1>Cohorts Details</h1>

{cohortList.map((cohort, index) => (

<CohortDetails key={index} cohort={cohort} />

))}

</div>

);

}

export default App;

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**6. ReactJS-HOL**

**Cognizant Academy teams want to maintain a list of trainers along with their expertise in a SPA using React as the technology. You are assigned the task of creating this React app.**

**The following trainers’ data application will deal.**

1. **T-ID**
2. **Name**
3. **Phone**
4. **Email**
5. **Stream**
6. **Skills**

**Trainer.js:**

export class Trainer {

constructor(trainerId, name, email, phone, technology, skills) {

this.trainerId = trainerId;

this.name = name;

this.email = email;

this.phone = phone;

this.technology = technology;

this.skills = skills;

}

}

TrainersMock.js:

import { Trainer } from '../models/Trainer';

const trainers = [

new Trainer(1, "Syed Khaleelullah", "khaleelullah@cognizant.com", "97676516962", ".NET", ["C#", "SQL Server", "React", ".NET Core"]),

new Trainer(2, "Jojo Jose", "jojo.jose@cognizant.com", "9876543210", "Java", ["Java", "Spring", "Hibernate"]),

new Trainer(3, "Elisa Jones", "elisa.jones@cognizant.com", "9123456789", "Python", ["Python", "Django", "Flask"])

];

export default trainers;

**Home.js:**

function Home() {

return (

<div>

<h2>Welcome to My Academy trainers page</h2>

</div>

);

}

export default Home;

TrainersList.js:

import { Link } from 'react-router-dom';

function TrainersList({ data }) {

return (

<div>

<h2>Trainers List</h2>

<ul>

{data.map((trainer) => (

<li key={trainer.trainerId}>

<Link to={`/trainer/${trainer.trainerId}`}>{trainer.name}</Link>

</li>

))}

</ul>

</div>

);

}

export default TrainersList;

**TrainerDetails.js:**

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

import trainers from '../data/TrainersMock';

function TrainerDetails() {

const { id } = useParams();

const trainer = trainers.find(t => t.trainerId === parseInt(id));

if (!trainer) return <h3>Trainer not found</h3>;

return (

<div>

<h2>Trainers Details</h2>

<h3>{trainer.name} ({trainer.technology})</h3>

<p>{trainer.email}</p>

<p>{trainer.phone}</p>

<ul>

{trainer.skills.map((skill, index) => (

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

))}

</ul>

</div>

);

}

export default TrainerDetails;

**App.js:**

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

import Home from './components/Home';

import TrainersList from './components/TrainersList';

import TrainerDetails from './components/TrainerDetails';

import trainers from './data/TrainersMock';

function App() {

return (

<Router>

<div>

<h1>My Academy Trainers App</h1>

<nav>

<Link to="/">Home</Link> | <Link to="/trainers">Show Trainers</Link>

</nav>

<Routes>

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

<Route path="/trainers" element={<TrainersList data={trainers} />} />

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

</Routes>

</div>

</Router>

);

}

export default App;

**index.js:**

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**7. ReactJS-HOL**

**Create a React Application named “shoppingapp” with a class component named “OnlineShopping” and “Cart”.**

**App.js:**

import React from 'react';

import './App.css';

class Cart {

constructor(itemname, price) {

this.itemname = itemname;

this.price = price;

}

}

class OnlineShopping extends React.Component {

constructor(props) {

super(props);

// Initializing Cart items

this.state = {

items: [

new Cart("Laptop", 80000),

new Cart("TV", 120000),

new Cart("Washing Machine", 50000),

new Cart("Mobile", 30000),

new Cart("Fridge", 70000)

]

};

}

render() {

return (

<div style={{ textAlign: "center", marginTop: "50px" }}>

<h1 style={{ color: "green" }}>Items Ordered :</h1>

<table

border="1"

style={{

margin: "auto",

padding: "10px",

borderCollapse: "collapse",

fontSize: "18px",

color: "green"

}}

>

<thead>

<tr>

<th>Name</th>

<th>Price</th>

</tr>

</thead>

<tbody>

{this.state.items.map((item, index) => (

<tr key={index}>

<td>{item.itemname}</td>

<td>{item.price}</td>

</tr>

))}

</tbody>

</table>

</div>

);

}

}

function App() {

return (

<div className="App">

<OnlineShopping />

</div>

);

}

export default App;



A screenshot of a computer

AI-generated content may be incorrect.

**8. ReactJS-HOL**

**Create a React App “counterapp” which will have a component named “CountPeople” which will have 2 methods.**

**App.js:**

import React from 'react';

import './App.css';

class CountPeople extends React.Component {

constructor(props) {

super(props);

this.state = {

entryCount: 0,

exitCount: 0

};

}

// Method to update entry count

updateEntry = () => {

this.setState(prevState => ({

entryCount: prevState.entryCount + 1

}));

}

// Method to update exit count

updateExit = () => {

this.setState(prevState => ({

exitCount: prevState.exitCount + 1

}));

}

render() {

return (

<div className="container">

<div className="box">

<button className="btn" onClick={this.updateEntry}>Login</button>

<span>{this.state.entryCount} People Entered!!!</span>

</div>

<div className="box">

<button className="btn" onClick={this.updateExit}>Exit</button>

<span>{this.state.exitCount} People Left!!!</span>

</div>

</div>

);

}

}

function App() {

return (

<div className="App">

<CountPeople />

</div>

);

}

**export default App;**

**Update the App.css to Style it Like the Screenshot:**

.App {

text-align: center;

margin-top: 100px;

font-family: Arial, sans-serif;

}

.container {

display: flex;

justify-content: center;

gap: 50px;

}

.box {

border: 2px solid lightgray;

padding: 20px;

border-radius: 10px;

}

.btn {

background-color: lightgreen;

border: 1px solid green;

border-radius: 5px;

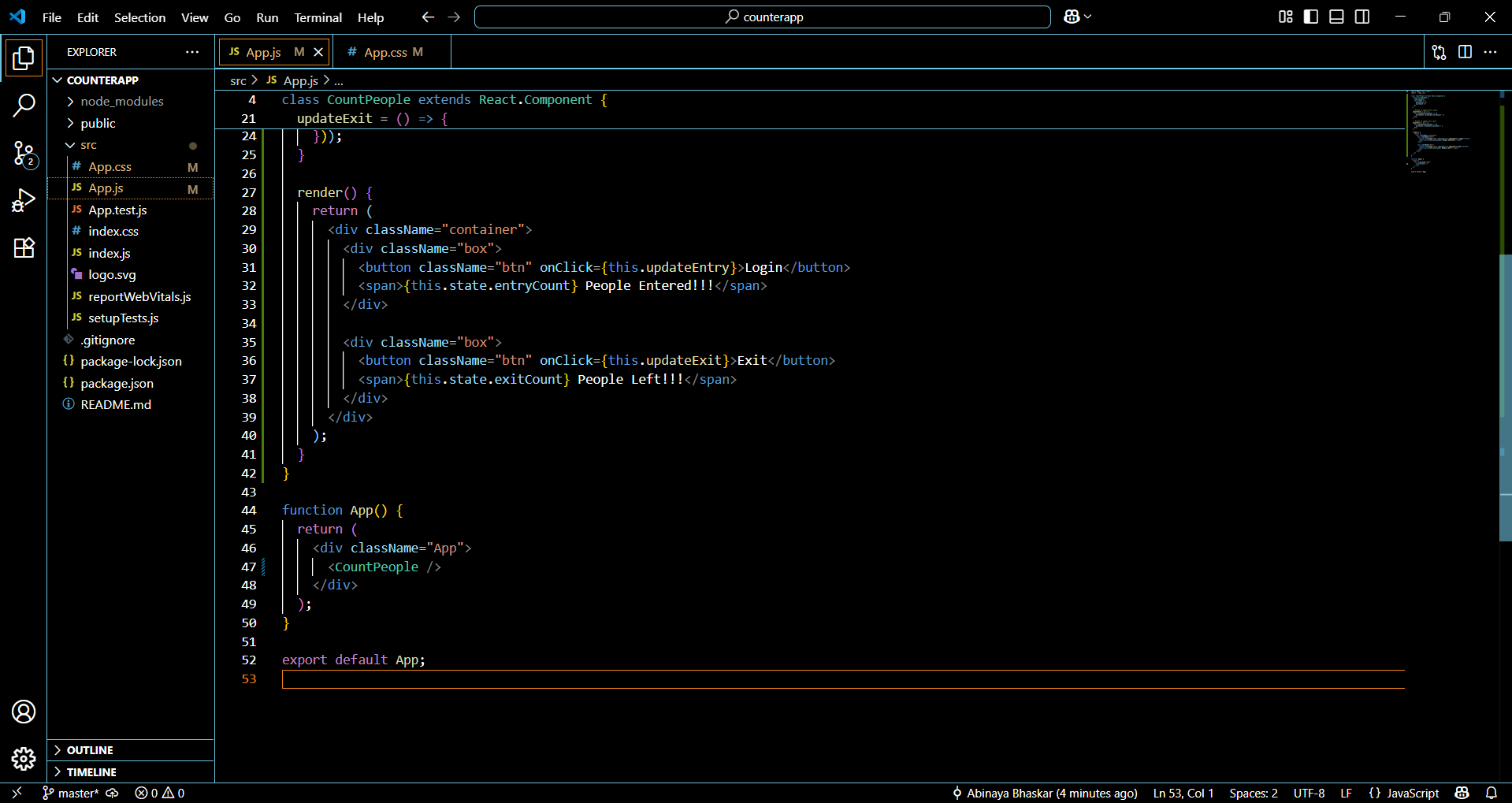
padding: 8px 12px;

font-weight: bold;

margin-right: 10px;

cursor: pointer;

}



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.