**Week 6 - React - Hands-on**

DEBANJAN KAURI(Superset ID-6362388)

**Mandatory Hands-on**

**Question 1: Create a new React Application**

Scenario:

Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page.

To achieve this:

1. Install Node.js and npm.

2. Create app using Create React App tool.

3. Open project folder in VS Code & Edit App.js to show the heading.

4. Save and run the app & Open localhost:3000 in browser.

### **Code:** *App.js*

import logo from './logo.svg';

import './App.css';

function App() {

return (

<div>

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

</div>

);

}

export default App;

### **Output:**

**Question 2: Create a new React Student Application**

Scenario:

Create a React app StudentApp with three components: Home, About, and Contact, each showing a welcome message.

To achieve this:

1. Create React project StudentApp.

2. In src, add a Components folder .

3. Create Home.js, About.js, Contact.js in it & Add respective welcome messages in each component.

4. Import and use all three in App.js.

5. Save and run the app & Open localhost:3000 in browser.

### **Code:** *(In Components)*

*About.js*

import React from 'react';

class About extends React.Component {

render() {

return (

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

);

}

}

export default About;

*Contact.js*

import React from 'react';

class Contact extends React.Component {

render() {

return (

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

);

}

}

export default Contact;

*Home.js*

import React from 'react';

class Home extends React.Component {

render() {

return (

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

);

}

}

export default Home;

### **Code:** *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;

### **Output:**

**Question 3: Create a new React Student Score Calculator Application**

Scenario:

Create a React app named scorecalculatorapp with a functional component CalculateScore that takes Name, School, Total, Goal, calculates average, and displays it.

To achieve this:

1. Create the React project scorecalculatorapp.

2. Inside src, create Components folder → add CalculateScore.js.

3.In the same src, create Stylesheets folder → add mystyle.css for styling.

4. Code the logic in CalculateScore.js and apply styles using mystyle.css.

5. Import and render CalculateScore in App.js.

6. Save and run the app & Open localhost:3000 in browser.

### **Code:**

*CalculateScore.js*

import React from 'react';

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

function CalculateScore() {

const name = "Debanjan";

const school = "Public School";

const total = 464;

const goal = 500;

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

return (

<div className="details-card">

<h2 className="title">Student Details:</h2>

<p><span className="label name">Name:</span> <span className="value name-val">{name}</span></p>

<p><span className="label school">School:</span> <span className="value school-val">{school}</span></p>

<p><span className="label total">Total:</span> <span className="value total-val">{total}Marks</span></p>

<p><span className="label score">Score:</span> <span className="value score-val">{score}%</span></p>

</div>

);

}

export default CalculateScore;

*Mystyle.css*

.details-card {

margin: 50px auto;

width: fit-content;

padding: 25px 40px;

border: 2px solid #ccc;

border-radius: 10px;

background-color: #fff;

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

font-family: Arial, sans-serif;

}

.title {

color: brown;

font-size: 24px;

font-weight: bold;

text-align: center;

margin-bottom: 25px;

}

.label {

font-weight: bold;

}

.name {

color: blue;

}

.school {

color: red;

}

.total {

color: purple;

}

.score {

color: green;

}

.name-val {

color: #0000cc;

}

.school-val {

color: #cc0000;

}

.total-val {

color: #800080;

}

.score-val {

color: #008000;

}

p {

font-size: 18px;

margin: 8px 0;

text-align: center;

}

### *App.js*

import React from 'react';

import CalculateScore from './Components/CalculateScore';

function App() {

return (

<div>

<CalculateScore />

</div>

);

}

export default App;

### **Output:**

**Question 4: Create a new React Blog Application**

Scenario:

Create a React app named blogapp that displays posts using a class-based component and Fetch API.

To achieve this:

1. Create a React project named blogapp**.**

2. In src, create two files: Post.js and Posts.js.

3. Inside the component, define a method loadPosts() that fetches posts from <https://jsonplaceholder.typicode.com/posts>

4. Call loadPosts() in componentDidMount() to load data after mounting.

5. In render(), display each post's title and body using heading and paragraph tags.

6. Add componentDidCatch() to handle and alert errors.

7. Import and render the Posts component in App.js.

8. Save and run the app & Open localhost:3000 in browser.

### **Code:**

*Post.js*

import React from 'react';

class Post {

constructor(id, title, body) {

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: [],

};

}

loadPosts = async () => {

try {

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

const data = await response.json();

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

this.setState({ posts: postObjects });

} catch (error) {

throw new Error('Error fetching posts');

}

};

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

alert('An error occurred: ' + error.message);

}

render() {

return (

<div>

<h1>All Posts</h1>

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

<div key={post.id}>

<h2>{post.title}</h2>

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

</div>

))}

</div>

);

}

}

export default Posts;

*App.js*

import React from 'react';

import './App.css';

import Posts from './Posts';

function App() {

return (

<div className="App">

<Posts />

</div>

);

}

export default App;

### 

### **Output:**

**Question 5: Create a new React Dashboard Application**

Scenario:

Style a given React app (cohorttracker.zip) dashboard that displays details of ongoing and completed cohorts.

To achieve this:

1. In src, create a CSS module named CohortDetails.module.css.

2. In src, create two files: Post.js and Posts.js.

3. Style the <dt> tag to have font-weight: 500.

4. Import the CSS module in the cohort details component and apply the box class to the main <div>.

5. Style <h3> conditionally: green if cohort is ongoing, blue otherwise.

6. Run the app and verify the UI matches the expected result.

### **Code:**

*CohortDetails.js*

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

function CohortDetails(props) {

const status = props.cohort.currentStatus; // ✅ extract status

return (

<div className={styles.box}>

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

{props.name}

</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;

}

### *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, index) => (

<CohortDetails key={index} cohort={cohort} name={cohort.cohortCode} />

))}

</div>

);

}

export default App;

### **Output:**

**Additional Hands-on**

**Question 6: Create a new React Trainers Application**

Scenario:

Create a React SPA named TrainersApp to display a list of trainers and their details using routing.

To achieve this:

1. Create a new React app TrainersApp.

2. In src, add trainer.js to define a Trainer class with ID, Name, Email, Phone, Technology, Skills.

3. Create TrainersMock.js to store mock trainer data.

4. Install React Router DOM for routing.

5. Create TrainerList.js with a TrainersList component to display trainer names as clickable links.

6. Create Home.js with a Home component to show static content.

7. Update App.js to use BrowserRouter and define routes

8. Create TrainerDetails.js with TrainerDetail component that fetches trainer info by id using useParams().

4. Save and run the app & Open localhost:3000 in browser.

### **Code:**

*Trainer.js*

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;

}

}

export default Trainer;

*TrainersMock.js*

import Trainer from './Trainer';

const TrainersData = [

new Trainer(1, "Debanjan", "Deb01@example.com", "1234567890", "Java", "Spring Boot, Microservices"),

new Trainer(2, "Anshita", "Anshita02@example.com", "0987654321", ".NET", "C#, ASP.NET, EF Core"),

new Trainer(3, "Bidisha", "BD03@example.com", "1122334455", "Python", "Flask, Django, ML"),

];

export default TrainersData;

*Home.js*

import React from 'react';

function Home() {

return (

<div>

<h2>Welcome to Cognizant Academy</h2>

<p>This is the Trainer Management Portal</p>

</div>

);

}

*export default Home;*

*TrainerList.js*

*import React from 'react';*

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

*function TrainerList({ trainers }) {*

*return (*

*<div>*

*<h2>List of Trainers</h2>*

*<ul>*

*{trainers.map(trainer => (*

*<li key={trainer.trainerId}>*

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

*</li>*

*))}*

*</ul>*

*</div>*

*);*

*}*

*export default TrainerList;*

*TrainerDetails.js*

*import React from 'react';*

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

*import TrainersData from './TrainersMock';*

*function TrainerDetails() {*

*const { id } = useParams();*

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

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

*return (*

*<div>*

*<h2>{trainer.name}'s Details</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;*

*App.js*

import React from 'react';

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

import Home from './Home';

import TrainerList from './TrainerList';

import TrainerDetails from './TrainerDetails';

import TrainersData from './TrainersMock';

function App() {

return (

<BrowserRouter>

<div>

<h1>Cognizant Trainer Portal</h1>

<nav>

<Link to="/">Home</Link> |{" "}

<Link to="/trainers">Trainers</Link>

</nav>

<Routes>

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

<Route path="/trainers" element={<TrainerList trainers={TrainersData} />} />

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

</Routes>

</div>

</BrowserRouter>

);

}

export default App;

### **Output:**





