**Week – 6**

**Q1.**

import './App.css';

function App() {

return (

<div className="App">

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

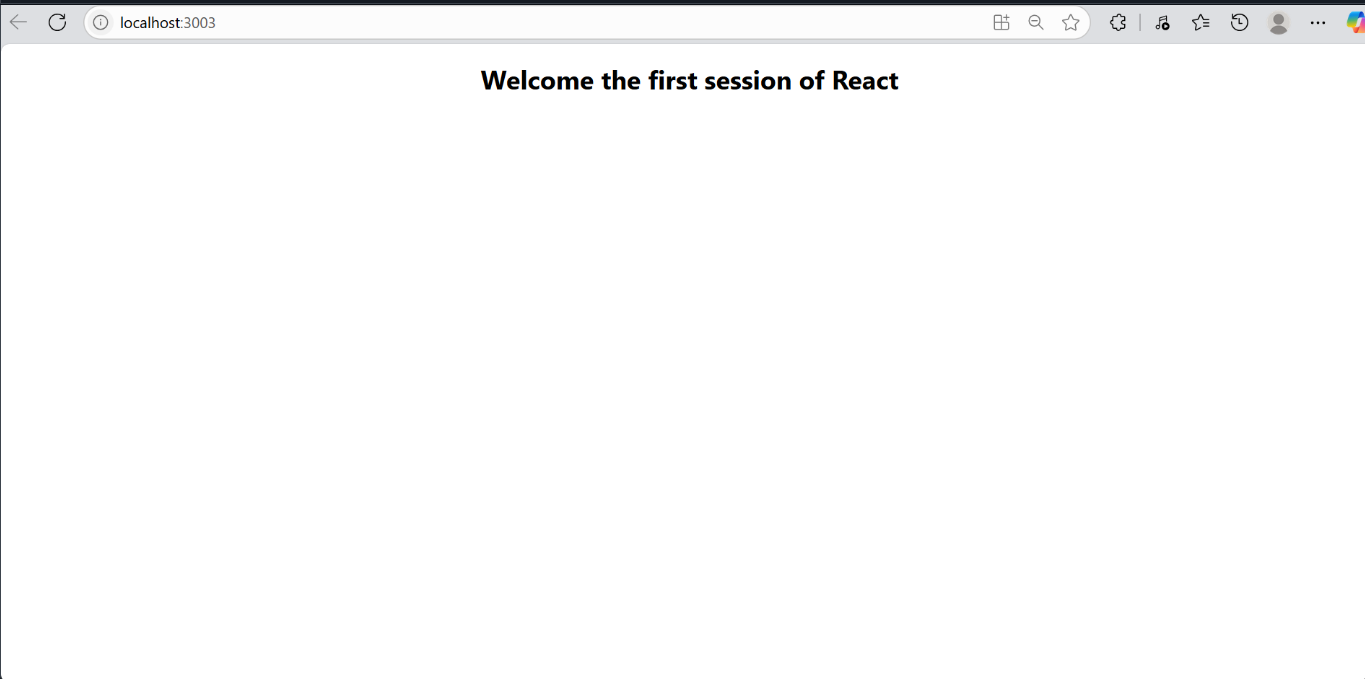
</div>

);

}

export default App;

**Output:**



**Q2.**

**Code:**

import React from 'react';

import './App.css';

import Home from './Components/Home';

import About from './Components/About';

import Contact from './Components/Contact';

function App() {

return (

<div className="App">

<div style={{

display: 'flex',

flexDirection: 'column',

justifyContent: 'center',

alignItems: 'center',

minHeight: '100vh',

textAlign: 'center'

}}>

<Home />

<About />

<Contact />

</div>

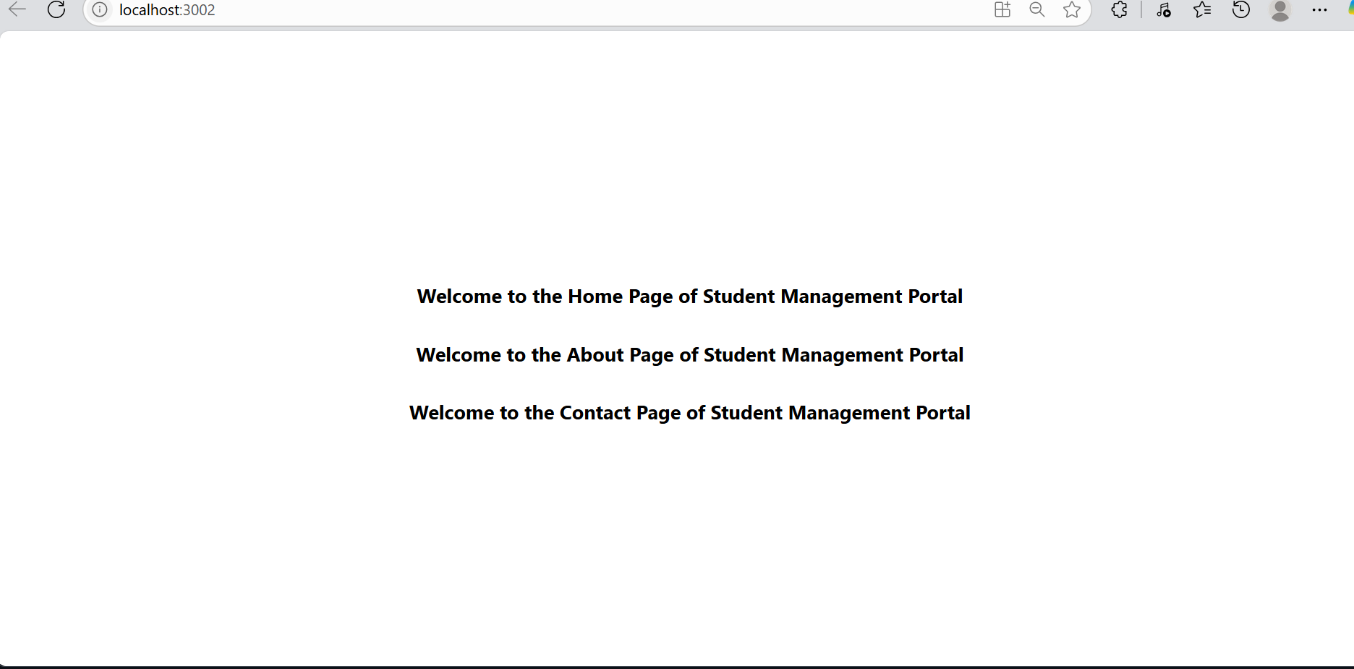
</div>

);

}

export default App;

**Output:**

****

**3.**

**Component:-**

import React from "react";

import "../Stylesheets/mystyle.css";

const CalculateScore = () => {

return (

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

<h1 style={{ color: "#8B0000", fontSize: "2.5rem", marginBottom: "30px", fontWeight: "bold" }}>

Student Details:

</h1>

<div style={{ marginBottom: "10px" }}>

<span style={{ fontWeight: "bold", color: "blue", fontSize: "1.2rem" }}>

Name:

</span>

<span style={{ color: "brown", fontSize: "1.2rem" }}> Steeve</span>

</div>

<div style={{ marginBottom: "10px" }}>

<span style={{ fontWeight: "bold", color: "#b30000", fontSize: "1.2rem" }}>

School:

</span>

<span style={{ color: "#d87093", fontSize: "1.2rem" }}> DNV Public School</span>

</div>

<div style={{ marginBottom: "10px" }}>

<span style={{ fontWeight: "bold", color: "#800000", fontSize: "1.2rem" }}>

Total:

</span>

<span style={{ color: "brown", fontSize: "1.2rem" }}> 284Marks</span>

</div>

<div>

<span style={{ fontWeight: "bold", color: "green", fontSize: "1.2rem" }}>

Score:

</span>

<span style={{ color: "#228B22", fontSize: "1.2rem" }}>94.67%</span>

</div>

</div>

);

};

export default CalculateScore;

**App.js**

import React from "react";

import "./App.css";

import CalculateScore from "./Components/CalculateScore";

function App() {

return (

<div>

<CalculateScore />

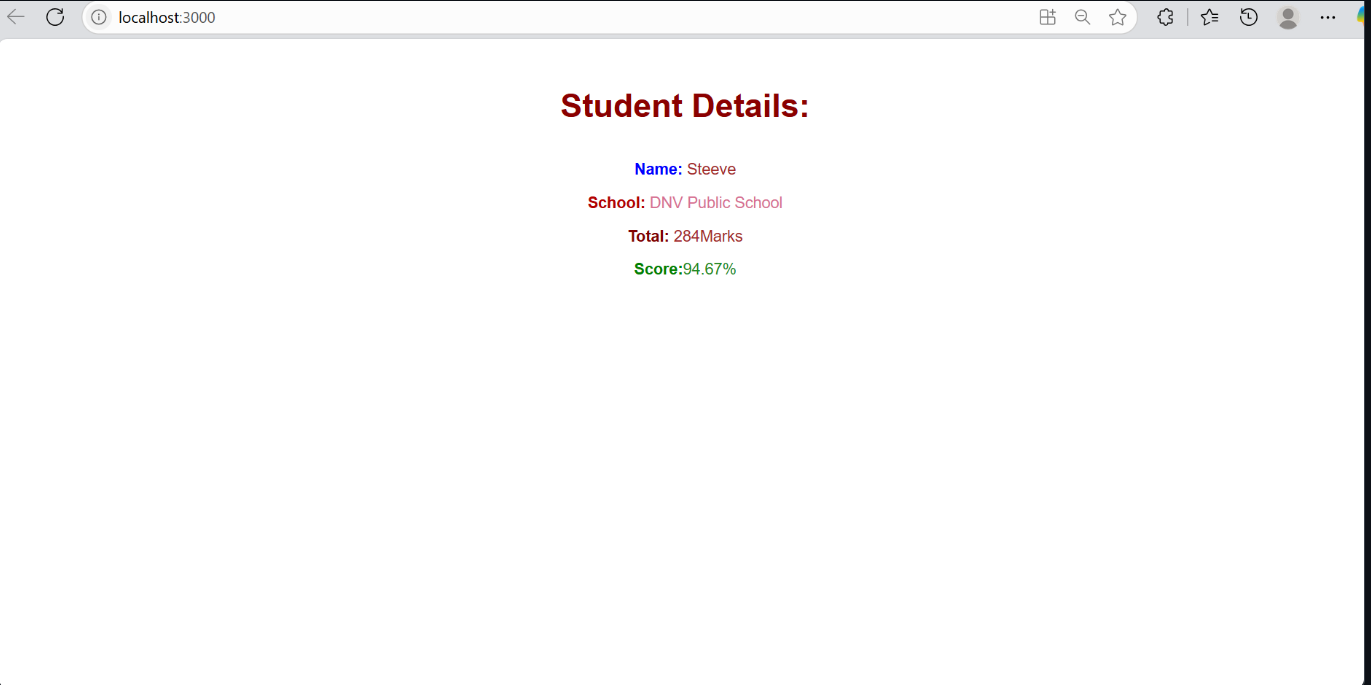
</div>

);

}

export default App;

**Output:-**

****

**4.**

**Code:**

import React from 'react';

import './App.css';

import Posts from './Post';

import './Posts.css';

function App() {

return (

<div className="App">

<header className="App-header">

<Posts />

</header>

</div>

);

}

export default App;

import React, { Component } from 'react';

class Post {

constructor(id, title, body, userId) {

this.id = id;

this.title = title;

this.body = body;

this.userId = userId;

}

getFormattedTitle() {

// You can format the title as needed, for now just return it

return this.title;

}

}

class Posts extends Component {

constructor(props) {

super(props);

// Initialize component state

this.state = {

posts: [], // Array to store fetched posts

loading: true, // Loading state indicator

error: null // Error state for error handling

};

console.log('Posts constructor called');

}

// Lifecycle method: Called after component is mounted to DOM

componentDidMount() {

console.log('Posts componentDidMount called');

console.log('Component has been mounted to DOM');

// Call method to load posts from API

this.loadPosts();

}

// Method to fetch posts from JSONPlaceholder API

loadPosts() {

console.log('loadPosts method called');

console.log('Starting to fetch posts from API...');

// Set loading state to true

this.setState({ loading: true, error: null });

// Fetch posts from JSONPlaceholder API

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

.then(response => {

console.log('API Response received:', response);

// Check if response is ok

if (!response.ok) {

throw new Error(`HTTP error! status: ${response.status}`);

}

// Parse JSON from response

return response.json();

})

.then(data => {

console.log('Posts data received:', data);

console.log(`Fetched ${data.length} posts`);

// Convert plain objects to Post instances

const posts = data.map(postData =>

new Post(postData.id, postData.title, postData.body, postData.userId)

);

// Update state with fetched posts

this.setState({

posts: posts,

loading: false,

error: null

});

console.log('Posts successfully loaded and state updated');

})

.catch(error => {

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

// Update state with error

this.setState({

posts: [],

loading: false,

error: error.message

});

});

}

// Error boundary method: Catches errors in child components

componentDidCatch(error, errorInfo) {

console.error('componentDidCatch called');

console.error('Error caught by error boundary:', error);

console.error('Error info:', errorInfo);

console.error('Component stack:', errorInfo.componentStack);

// Display error as alert

alert(`An error occurred in the Posts component:\n\nError: ${error.message}\n\nComponent Stack: ${errorInfo.componentStack}`);

// Update state to show error message

this.setState({

error: `Component Error: ${error.message}`,

loading: false

});

}

// Render method: Returns JSX to be displayed

render() {

console.log('Posts render method called');

console.log('Current state:', this.state);

const { posts, loading, error } = this.state;

// Show loading message while fetching data

if (loading) {

return (

<div className="posts-container">

<div className="loading">

<h2>Loading Posts...</h2>

<p>Please wait while we fetch the latest posts.</p>

</div>

</div>

);

}

// Show error message if something went wrong

if (error) {

return (

<div className="posts-container">

<div className="error">

<h2>Error Loading Posts</h2>

<p>{error}</p>

<button onClick={() => this.loadPosts()}>

Try Again

</button>

</div>

</div>

);

}

// Show posts if data loaded successfully

return (

<div className="posts-container">

<header className="posts-header">

<h1>Blog Posts</h1>

<p>Welcome to our blog! Here are the latest posts:</p>

<div className="posts-count">

Total Posts: {posts.length}

</div>

</header>

<div className="posts-list">

{posts.map(post => (

<article key={post.id} className="post-item">

<header className="post-header">

<h2 className="post-title">

{post.getFormattedTitle()}

</h2>

<div className="post-meta">

<span className="post-id">Post #{post.id}</span>

<span className="post-author">By User {post.userId}</span>

</div>

</header>

<div className="post-content">

<p className="post-body">

{post.body}

</p>

</div>

<footer className="post-footer">

<button className="read-more-btn">

Read More

</button>

<div className="post-actions">

<button className="like-btn">👍 Like</button>

<button className="share-btn">📤 Share</button>

</div>

</footer>

</article>

))}

</div>

<footer className="posts-footer">

<button

className="refresh-btn"

onClick={() => this.loadPosts()}

>

Refresh Posts

</button>

</footer>

</div>

);

}

}

export default Posts;

const reportWebVitals = onPerfEntry => {

if (onPerfEntry && onPerfEntry instanceof Function) {

import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {

getCLS(onPerfEntry);

getFID(onPerfEntry);

getFCP(onPerfEntry);

getLCP(onPerfEntry);

getTTFB(onPerfEntry);

});

}

};

export default reportWebVitals;



**5.**

import React from 'react';

import CohortDetails from './CohortDetails';

import './App.css';

function App() {

const cohortsData = [

{

id: "INTADMDF10",

name: "INTADMDF10 -.NET FSD",

startDate: "22-Feb-2022",

status: "Scheduled",

coach: "Aathma",

trainer: "Jojo Jose"

},

{

id: "ADM21JF014",

name: "ADM21JF014 -Java FSD",

startDate: "10-Sep-2021",

status: "Ongoing",

coach: "Apoorv",

trainer: "Elisa Smith"

},

{

id: "CDBJF21025",

name: "CDBJF21025 -Java FSD",

startDate: "24-Dec-2021",

status: "Ongoing",

coach: "Aathma",

trainer: "John Doe"

}

];

return (

<div className="App">

<div className="cohort-details-container">

<h2>Cohorts Details</h2>

<div className="cohort-cards">

{cohortsData.map(cohort => (

<CohortDetails key={cohort.id} cohort={cohort} />

))}

</div>

</div>

</div>

);

}

export default App;

import React from 'react';

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

const CohortDetails = ({ cohort }) => {

// Define inline style for h3 based on cohort status

const getHeadingStyle = (status) => {

let color;

if (status === 'Ongoing') {

color = 'green';

} else {

color = 'blue';

}

return {

color: color,

margin: 0,

marginBottom: '15px',

fontSize: '16px',

fontWeight: 'bold'

};

};

return (

<div className={styles.box}>

<h3 style={getHeadingStyle(cohort.status)}>{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>

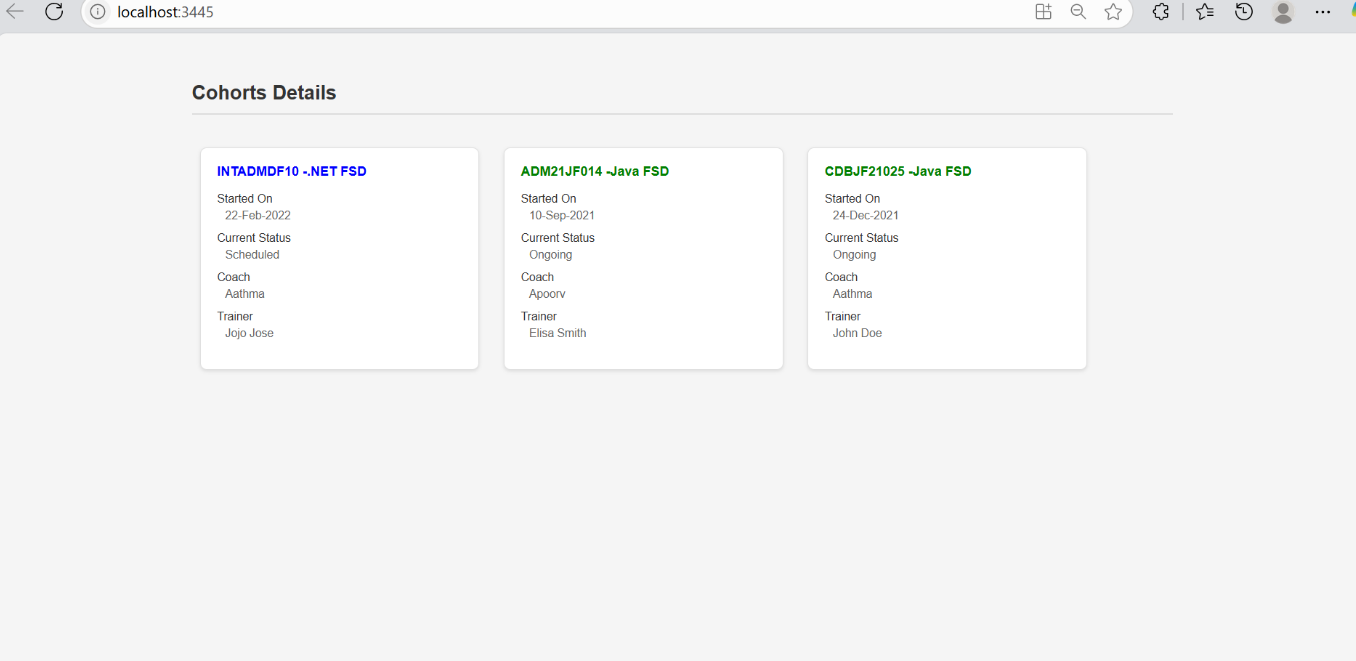
</div>

);

};

export default CohortDetails;

**Output:**

****