**DN 4.0 Dotnet FSE**

**Name: Aditya Mohanty**

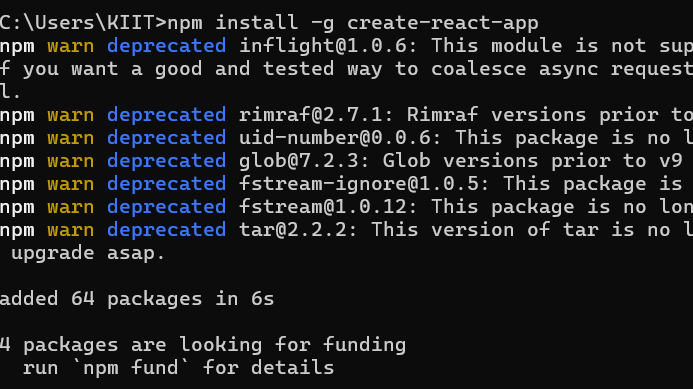
**Superset ID:6361043**

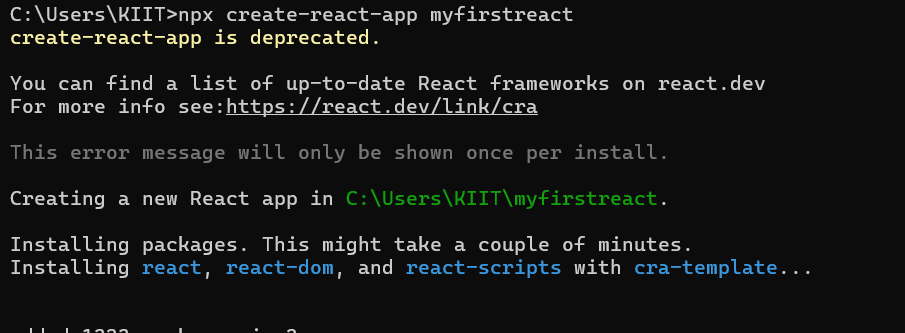
**Week 6**

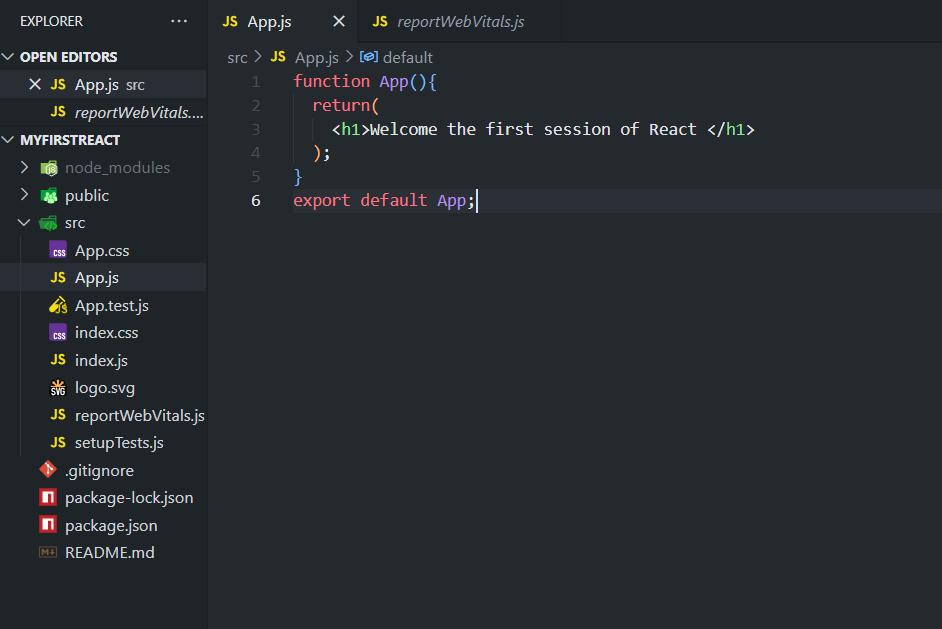
**React**

**Handson:1**

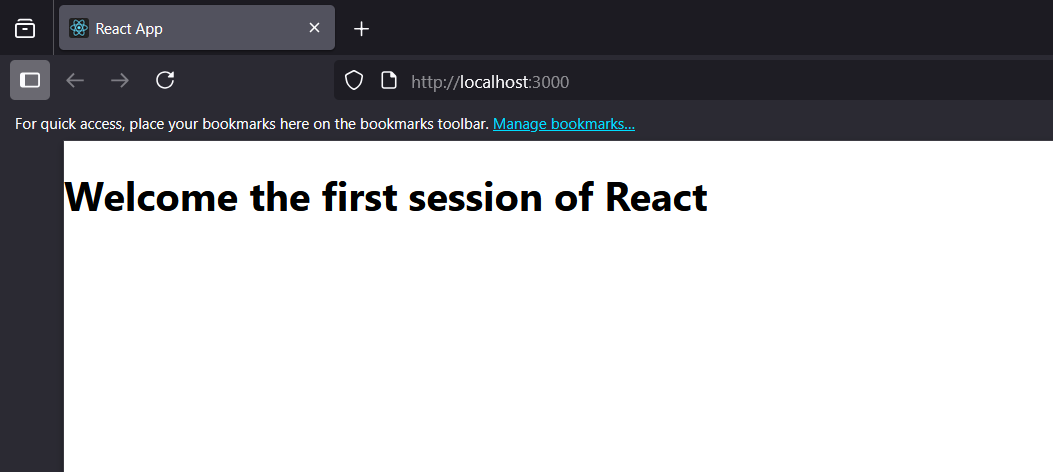
Input commands:







**OUPUT:**



**Handson-2**

**Home.js:**

function Home() {

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

}

export default Home;

**About.js:**

function About() {

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

}

export default About;

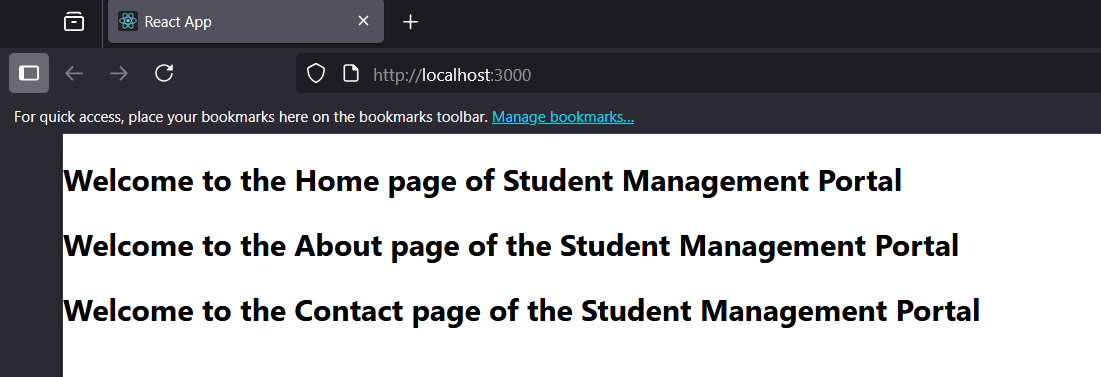
**Contact.js:**

function Contact() {

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

export default Contact;

**OUPUT:(next page)**



**Handson-3**

**Components/CalculatorScore.js:**

import '../Stylesheets/mystyle.css'

const percentToDecimal = (decimal) => {

return decimal.toFixed(2) + '%'

}

const calcScore = (total, goal) => {

return percentToDecimal(total / goal)

}

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

<div className="formatstyle">

<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}</span>

</div>

<div className="Marks">

<b><span>Marks: </span></b><span></span>

</div>

<div className="Score">

<b><span>Score:</span></b>

<span>

{calcScore(

total,

goal

)}

</span>

</div>

</div>

)

**Stylesheets/mystyle.css:**

.Name {

font-weight: 300;

color: blue;

}

.School {

color: crimson;

}

.Total {

color: darkmagenta;

}

.formatstyle {

text-align: center;

font-size: large;

}

.Score {

color: forestgreen;

}

**App.js:**

import { CalculateScore } from './Components/CalculateScore';

function App() {

return (

<div>

<CalculateScore Name={"Steeve"}

School={"DNV Public School"}

total={284}

goal={3} />

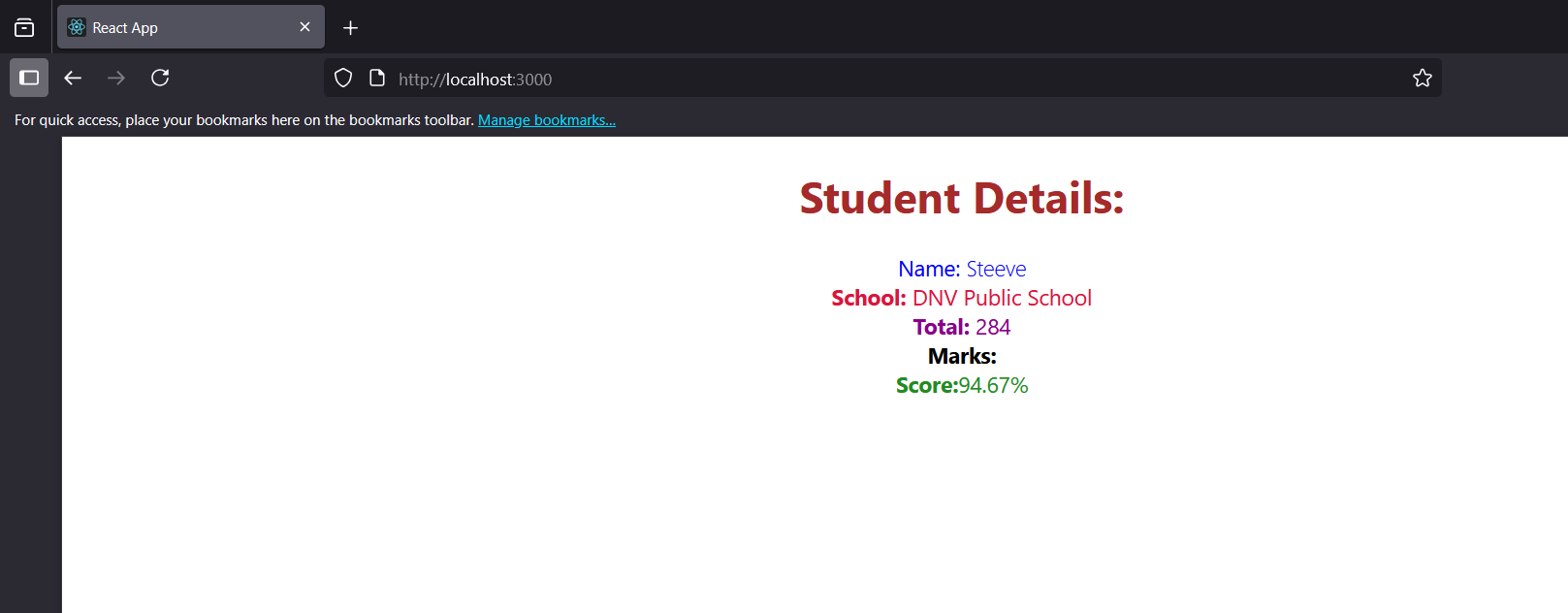
</div>

)

}

export default App;

**OUPUT:**



**Handson-4**

**Post.js:**

class Post {

constructor(id, title, body){

this.id=id;

this.title=title;

this.body=body;

}

}

export default Post;

**Posts.js:**

import React from 'react';

import Post from './Post';

class Posts extends React.Component {

constructor(props){

super(props);

this.state = {

posts: [],

hasError: false,

errorInfo: null

};

this.loadPosts = this.loadPosts.bind(this);

}

async loadPosts() {

try {

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

if (!response.ok) {

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

}

const data = await response.json();

const fetchedPosts = data.map(item => new Post(item.id, item.title, item.body));

this.setState({ posts: fetchedPosts });

} catch (error) {

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

this.setState({ hasError: true });

}

}

componentDidMount() {

this.loadPosts();

}

componentDidCatch(error, info) {

console.error("Error caught by componentDidCatch:", error, info);

this.setState({

hasError: true,

errorInfo: info

});

alert("An error occurred in the component: " + error.message);

}

render() {

if (this.state.hasError) {

return <h1>Something went wrong.</h1>;

}

return (

<div>

<h1>Blog 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 Posts from './Posts'; // Import the Posts component

function App() {

return (

<div className="App">

<Posts />

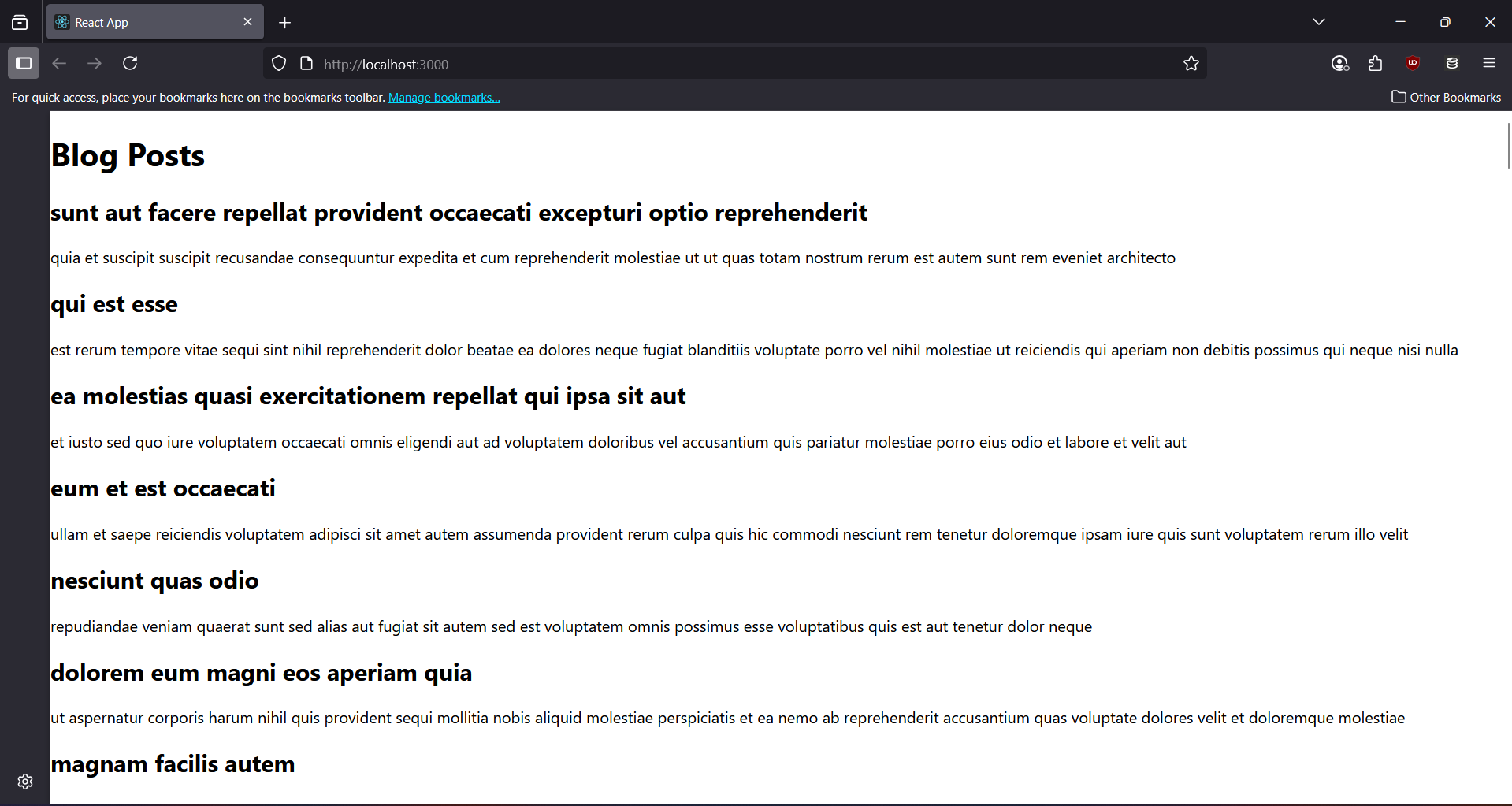
</div>

);

}

export default App;

**OUPUT:(next page)**



**Handson-5**

**CohortDetails.modules.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;

}

dt {

font-weight: 500;

}

**Updated CohortDetails.js:**

import styles from './CohortDetails.module.css'; // Import CSS Module

function CohortDetails(props) {

// Define the style for <h3> based on cohort status

const headerStyle = {

color: props.cohort.currentStatus === 'Ongoing' ? 'green' : 'blue'

};

return (

<div className={styles.box}> {/\* Apply the box class \*/}

<h3 style={headerStyle}> {/\* Apply the dynamic style \*/}

{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;

**Note:**App.js remains same as it is

**OUPUT:**

