React-js(wk-06)

1st HandsON

- **Set up a react environment**
- **Use create-react-app**

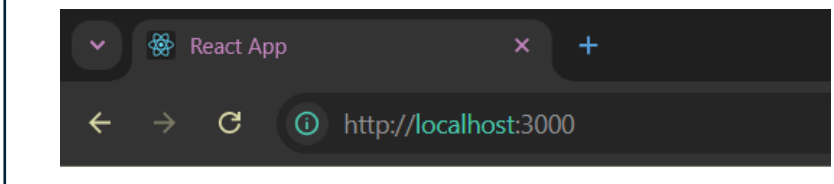
=>solution:

```
$ npx create-react-app myfirstreact
$ cd myfirstreact
$ npm start
```

App.js

```
function App() {
  return (
    <h1>Welcome to the first session of react.</h1>
  );
}

export default App;
```

```
Compiled successfully!

You can now view myfistreact in the browser.

  Local:            http://localhost:3000
  On Your Network:                  :3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



Welcome to the first session of react.

2nd HandsOn

- **Create a class component**
- **Create multiple components**
- **Render a component**

=>solution:

App.js

```
import './App.css';
import About from './components/About';
import Contact from './components/Contact';
import Home from './components/Home';

function App() {
 return (
  <div className="App">
   <Home/>
   <About/>
   <Contact/>
  </div>
 );
}
export default App;
```

Home.js

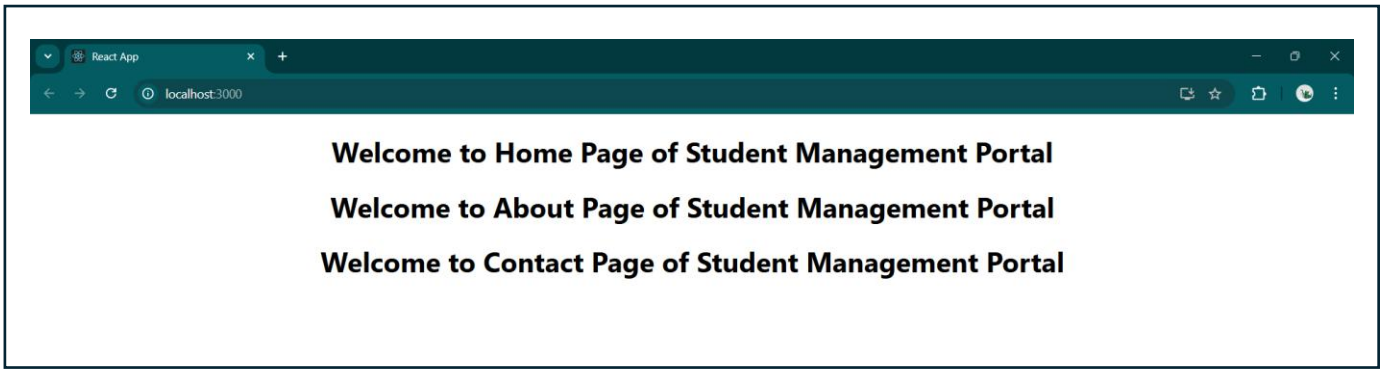
```
function Home(){
  return(
    <h1>Welcome to Home Page of Student Management Portal</h1>
  )
}
export default Home;
```

About.js

```
function About() {
 return <h1>Welcome to About Page of Student Management Portal</h1>
}
export default About
```

Contact.js

```
function Contact() {
 return <h1>Welcome to Contact Page of Student Management Portal</h1>
}
export default Contact
```



3rd HandsOn

- **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**

=>solution:

CalculateScore.js

```
import '../Stylesheets/mystyle.css';
const percentToDec = (decimal) => {
  return decimal.toFixed(2) + '%';
}
const calcScore = (total, goal) => {
  return percentToDec(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>
```

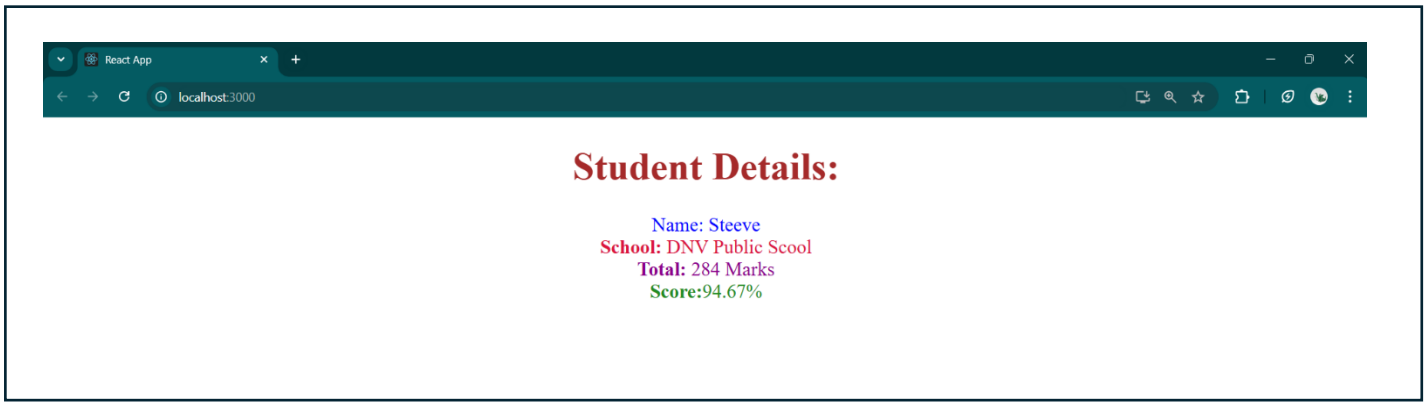
```
      </b>
      <span>{school}</span>
    </div>

    <div className="Total">
      <b>
        <span>Total: </span>
      </b>
      <span>{total} </span>
      <span>Marks</span>
    </div>

    <div className="Score">
      <b>Score:</b>
      <span>
        {calcScore(total,goal)}
      </span>
    </div>
  </div>
);
```

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 "./Component/CalculateScore";

function App() {
  return (
    <div className="App">
      <CalculateScore name="Steeve" school="DNV Public Scool" total={284} goal={3}/>
    </div>
  );
}

export default App;
```



HandsOn-4

- **Implement componentDidMount() hook**
- **Implementing componentDidCatch() life cycle hook.**

➔ solution:

Posts.js

```
import React, { Component } from 'react';
import Post from './Post';

class Posts extends Component {
  constructor(props) {
    super(props);
    this.state = {
      posts: []
    };
  }
```

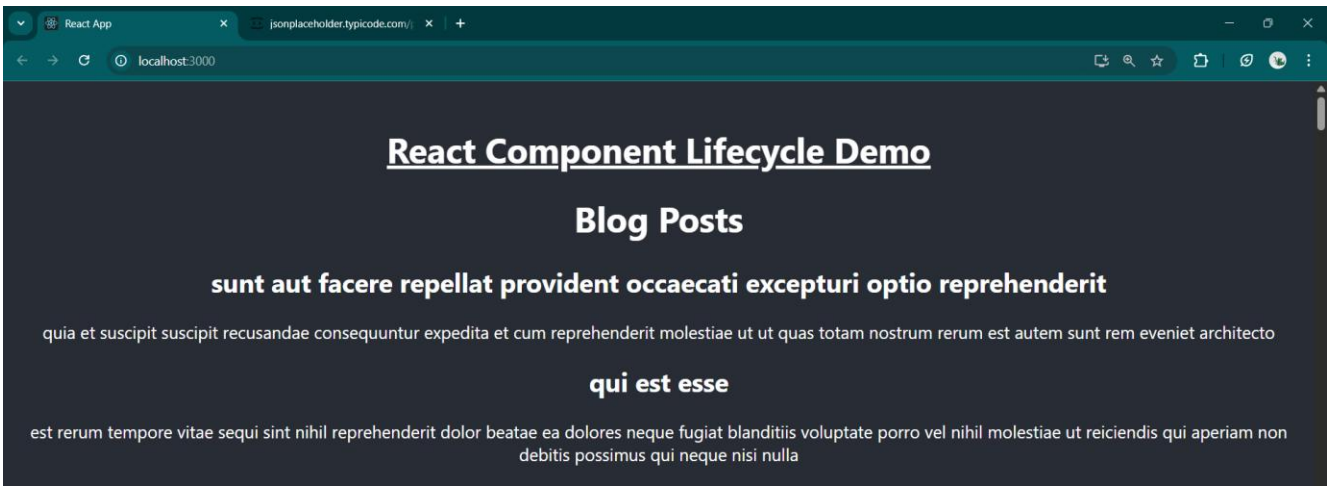
```
  loadPosts() {
    fetch('https://jsonplaceholder.typicode.com/posts')
      .then(response => response.json())
      .then(data => {
        const posts = data.map(postData =>
          new Post(postData.id, postData.title, postData.body)
        );
        this.setState({ posts: posts });
      })
      .catch(error => {
        console.error('Error: ', error);
        throw error;
      });
  }

  componentDidMount() {
    console.log('calling loadPosts()');
    this.loadPosts();
  }

  componentDidCatch(error, errorInfo) {
    console.error('Error: ', error, errorInfo);
    alert(` error: ${error.message}` );
  }

  render() {
    const { posts } = this.state;
    return (
      <div>
        <h1>Blog Posts</h1>
        {posts.map(post => (
          <div key={post.id}>
            <h2>{post.title}</h2>
            <p>{post.body}</p>
          </div>
        ))}
      </div>
    );
  }
}

export default Posts;
```

Post.js

```
class Post {
   constructor(id, title, body) {
      this.id = id;
      this.title = title;
      this.body = body;
   }
}

export default Post;
```

App.js

```
import './App.css';
import Posts from './Posts';

function App() {
 return (
   <div>
    <header>
      <h1 style={{textDecoration:"underline"}}>React Component Lifecycle Demo</h1>
      <Posts />
    </header>
   </div>
 );
}

export default App;
```



HandsOn-5

- **Style a react component**
- **Define styles using the CSS Module**
- **Apply styles to components using className and style properties**

➔**solution**

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;
}
```

CohortDetails.js

```
import styles from './CohortDetails.module.css'
function CohortDetails(props) {
 const h3Style = {
  color: props.cohort.currentStatus === 'Ongoing' ? 'green' : 'blue',
 }
 return (
  <div className={styles.box}>
   <h3 style={h3Style}>
    {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;
```

App.js

```
import './App.css';
import { CohortsData} from './Cohort'
import CohortDetails from './CohortDetails';

function App() {
 return (
 <div>
  <h1>Cohorts Details</h1>
  {CohortsData.map(cohort => <CohortDetails cohort={cohort}/>)}
 </div>
 );
}

export default App;
```