DotNet-FSE Mandatory Hands-On

WEEK-6 **NAME:** Sri Ranjani Priya P

EXERCISE 1:

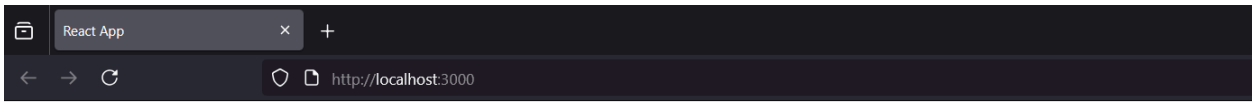
Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page.

CODE:

File: App.js

```
import React from 'react'

export default function App() {
  return (
    <div style={{
      display: 'flex',
      justifyContent: 'center',
      alignItems: 'flex-start',
      height: '100vh',
      paddingTop: '50px',
      textAlign: 'center'

    }}>
      <h1>Welcome to the first session of React</h1>
    </div>
  )
}
```

OUTPUT:



Welcome to the first session of React

EXERCISE 2:

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.

CODE:

File: App.js

```
import React from 'react';
import './App.css';
import Home from './Components/Home';
import Contact from './Components/Contact';
import About from './Components/About';
function App(){
  return(
    <div className="container"
    >
      <Home/>
      <Contact/>

      <About/>
    </div>
  );
}
export default App;
```

File:Home.js:

```
import React from 'react'
export default function Home() {
  return (
    <div   style={{
             display:'flex',
             alignItems:'flex-start',
             justifyContent:'center'
        }}>
      <h3>Welcome to the Home page of Student Management Portal</h3>
    </div>
  )
}
```

File:Contact.js:

```
import React from 'react';
function Contact(){
    return(
        <div  style={{
            display:'flex',
            alignItems:'flex-start',
            justifyContent:'center'
        }}>
            <h3>Welcome to the Contact page of the Student Management
Portal</h3>
        </div>
    );
}
export default Contact;
```

File:About.js:

```
import React from 'react'

export default function About() {
  return (
    <div   style={{
            display:'flex',
            alignItems:'flex-start',
            justifyContent:'center'
        }}>
      <h3>Welcome to the About page of the Student Management Portal</h3>
    </div>
  )
}
```

OUTPUT:

Welcome to the Home page of Student Management Portal

Welcome to the Contact page of the Student Management Portal

Welcome to the About page of the Student Management Portal

EXERCISE 3:

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.

CODE:

File: App.js

```
import React from 'react';
import './App.css';
import CalculateScore from './Components/CalculateScore';
function App(){
  return(
    <div className="container"
    >
      <CalculateScore Name={"Cognizant"}
      School={"Tech School"}
      total={295}
      goal={3}/>
    </div>
  );
}
export default App;
```

File:CalculateScore.js

```
import React from 'react'
import '../Stylesheets/mystyle.css';
const percentToDecimal=(decimal)=>{
    return (decimal.toFixed(2)+'%')
}
const calcScore=(total,goal)=>{
    return (percentToDecimal(total/goal))
}
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><span>Score : </span></b>
        <span>{calcScore(total,goal)}</span></div>
    </div>);
export default CalculateScore;
```

File: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;
}
```

OUTPUT:

Student Details

Name : Cognizant
School : Tech School
Total : 295Marks
Score : 98.33%

EXERCISE 4:

Create a new react application using *create-react-app* tool with the name as “blogapp”.1. Create a new method in component with the name as **loadPosts()** which will be responsible for using Fetch API and assign it to the component state created earlier. To get the posts use the url (https://jsonplaceholder.typicode.com/posts).Define a **componentDidCatch()** method which will be responsible for displaying any error happing in the component as alert messages.

CODE:

File: App.js

```
import React from 'react';
import './App.css';
import Posts from './Components/Posts';
function App() {
  return (
    <div className="App">
      <Posts />
    </div>
  );
}
export default App;
```

File:Posts.js

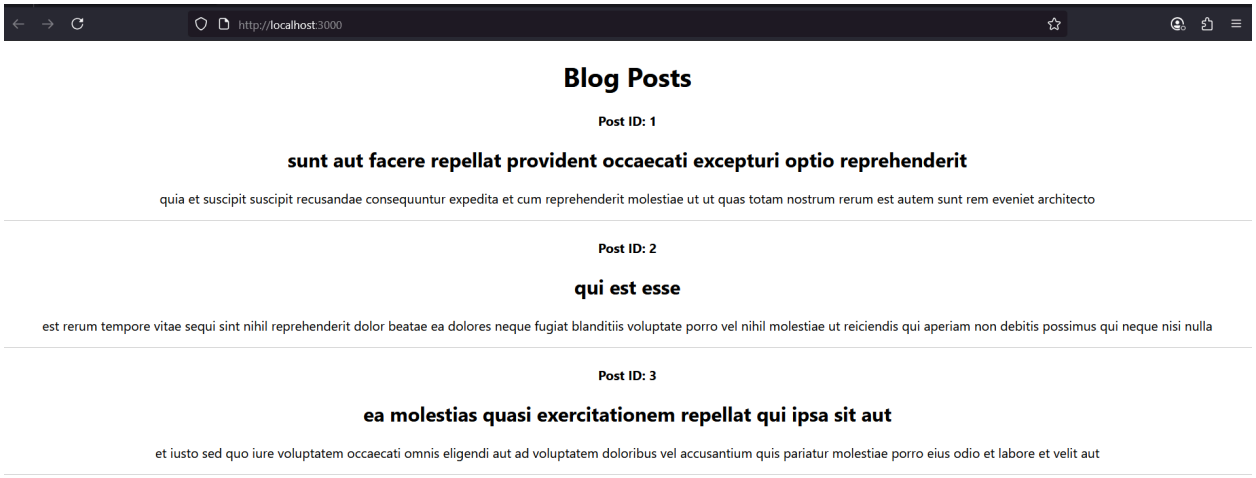
```
class Post {
  constructor(id, title, body) {
    this.id = id;
    this.title = title;
    this.body = body;
  }
}

export default Post;
```

File: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 postsList = data.map((post) => new Post(post.id, post.title,
post.body));
        this.setState({ posts: postsList });
      })
      .catch((err) => {
        this.setState({ error: err.message });
        alert('Error fetching posts: ' + err.message);
      });
  };
  componentDidMount() {
    this.loadPosts();
  }
  componentDidCatch(error, info) {
    alert('Something went wrong: ' + error.message);
  }
  render() {
  const { posts } = this.state;
  return (
    <div>
      <h1>Blog Posts</h1>
      {posts.map((post) => (
        <div key={post.id} style={{ borderBottom: '1px solid #ccc',
marginBottom: '10px' }}>
          <h4>Post ID: {post.id}</h4>
          <h2>{post.title}</h2>
          <p>{post.body}</p>
        </div>
      ))}
    </div>
  );}}
export default Posts;
```

OUTPUT:



EXERCISE 5:

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.

CODE:

File: App.js:

```
import React from 'react';
import './App.css';
import CohortDetails from './Components/CohortDetails';
import styles from './Components/CohortDetails.module.css';
function App() {
  const cohortData = [
    {
      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'
    }
  ];
  return (
    <div className="App">
      <h2>Cohorts Details</h2>
      <div className={styles.container}>
        {cohortData.map((cohort, index) => (
          <CohortDetails key={index} {...cohort} />
        ))}
      </div>
    </div>
  );
}
export default App;
```

File:Cohort.js:

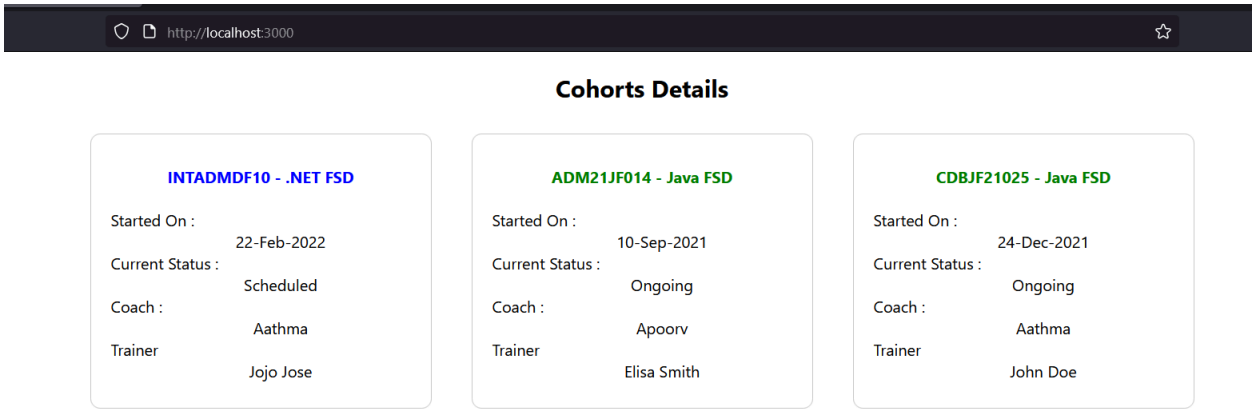
```
import React from 'react';
import styles from './CohortDetails.module.css';
function CohortDetails(props) {
  const { name, startDate, status, coach, trainer } = props;
  const titleStyle = {
    color: status === 'Ongoing' ? 'green' : status === 'Scheduled' ?
'blue' : 'black',
    fontWeight: 'bold'
  };
  return (
    <div className={styles.box}>
      <h4 style={titleStyle}>{name}</h4>
      <dl>
        <dt>Started On :</dt>
        <dd>{startDate}</dd>
```

```
        <dt>Current Status :</dt>
        <dd>{status}</dd>

        <dt>Coach :</dt>
        <dd>{coach} </dd>

        <dt>Trainer</dt>
        <dd>{trainer}</dd>
      </dl>
    </div>
  );
}
export default CohortDetails;
```

File:CohortDetails.module.css:

```
.container {
  display: flex;
  gap: 20px;
  justify-content: center;
  flex-wrap: wrap;
}

.box {
  width: 300px;
  display: inline-block;
  margin: 10px;
  padding: 10px 20px;
  border: 1px solid #ccc;
  border-radius: 10px;
}

dt {
  font-weight: 500;
  text-align: left;
}
```

OUTPUT:


http://localhost:3000
Cohorts Details
INTADMDF10 - .NET FSD
Started On :
22-Feb-2022
Current Status :
Scheduled
Coach :
Aathma
Trainer
Jojo Jose
ADM21JF014 - Java FSD
Started On :
10-Sep-2021
Current Status :
Ongoing
Coach :
Apoorv
Trainer
Elisa Smith
CDBJF21025 - Java FSD
Started On :
24-Dec-2021
Current Status :
Ongoing
Coach :
Aathma
Trainer
John Doe