DotNet-FSE Mandatory Hands-On

WEEK-7 **NAME:** Sri Ranjani Priya P

EXERCISE 1:

Create a React application that displays a list of trainers and shows detailed information about a selected trainer using dynamic routing. Explain the approach, tools used, and provide code snippets for important components.

CODE:

File: App.js

```
import React from 'react';
import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';
import Home from './Home';
import TrainersList from './TrainersList';
import TrainerDetails from './TrainerDetails';
import trainersMock from './trainersMock'; function App() {
  return (
    <BrowserRouter>
      <nav>
        <Link to="/">Home</Link> | <Link to="/trainers">Trainers</Link>
      </nav>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/trainers" element={<TrainersList
trainers={trainersMock} />} />

        <Route path="/trainer/:id" element={<TrainerDetails />} />
      </Routes>
    </BrowserRouter>
  );
}

export default App;
```

File: Trainer.js

```
class Trainer {
  constructor(trainerId, name, email, phone, stream, skills) {
    this.trainerId = trainerId;
    this.name = name;
    this.email = email;
```

```
    this.phone = phone;
    this.stream = stream;
    this.skills = skills;
  }
}
export default Trainer;
```

File: TrainerDetails.js

```
import React from 'react';

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

import trainersMock from './trainersMock';

function TrainerDetails() {
  const { id } = useParams();
  const trainer = trainersMock.find(t => t.trainerId === id);

  if (!trainer) {
    return <p>Trainer not found</p>;
  }

  return (
    <div>
      <h2>{trainer.name}'s Details</h2>
      <p><strong>ID:</strong> {trainer.trainerId}</p>
      <p><strong>Email:</strong> {trainer.email}</p>
      <p><strong>Phone:</strong> {trainer.phone}</p>
      <p><strong>Technology:</strong> {trainer.stream}</p>
      <p><strong>Skills:</strong> {trainer.skills.join(', ')}</p>
    </div>
  );
}
export default TrainerDetails;
```

File: TrainerList.js

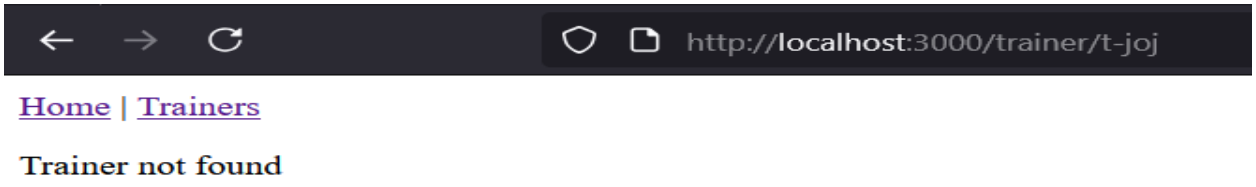
```
import React from 'react';
import { Link } from 'react-router-dom';
import trainersMock from './trainersMock';
function TrainersList({ trainers }) {
```

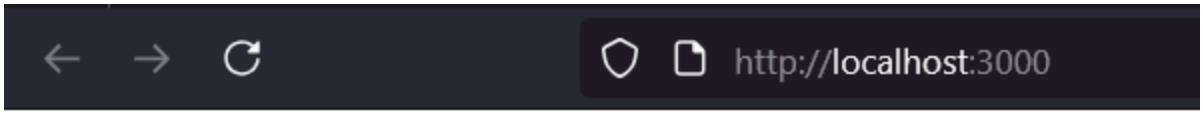
```
  return (
    <div>
      <h2>Trainers List</h2>
      <ul>
         {trainersMock && trainersMock.map((trainer) => (

       <li key={trainer.trainerId}>
  <Link to={`/trainer/${trainer.trainerId}`}>
    {trainer.name}
  </Link>
</li>        ))}
      </ul>
    </div>
  );
}
export default TrainersList;
```

File: trainersMock.js

```
import Trainer from './Trainer';
const trainersMock = [
  new Trainer('t-syed8', 'Syed Khaleelullah',
'khaleelullah@cognizant.com', '97676516962', '.NET', ['C#', 'SQL Server',
'React', '.NET Core']),
  new Trainer('t-jojo', 'Jojo Jose', 'jojo@cognizant.com', '9897199231',
'Java', ['Java', 'JSP', 'Angular', 'Spring']),
  new Trainer('t-elisa', 'Elisa Jones', 'elisa@cognizant.com',
'9871212235', 'Python', ['Python', 'Django', 'Angular']),
];
export default trainersMock;
```

OUTPUT:

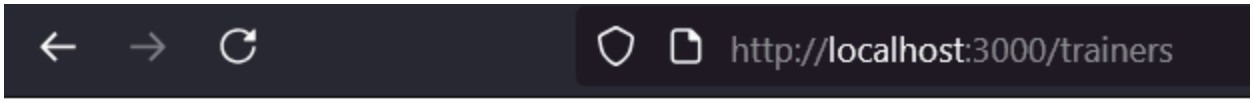




Home | Trainers List

Welcome to Trainers App

Explore our expert trainers!



Home | Trainers List

Trainers List

- Syed Khaleelullah
- Jojo Jose
- Elisa Jones

Home | Trainers

Syed Khaleelullah's Details

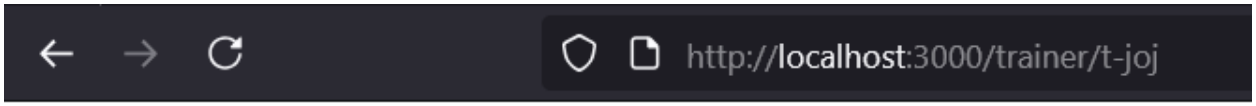
ID: t-syed8

Email: khaleelullah@cognizant.com

Phone: 97676516962

Technology: .NET

Skills: C#, SQL Server, React, .NET Core



Home | Trainers

Trainer not found

EXERCISE 2:

Create a React Application named “shoppingapp” with a class component named “OnlineShopping” and “Cart”.

CODE:

File: App.js:

```
import React from 'react';
import './App.css';

function App() {
  const items = [
```

```
    { name: 'Laptop', price: 80000 },
    { name: 'TV', price: 120000 },
    { name: 'Washing Machine', price: 50000 },
    { name: 'Mobile', price: 30000 },
    { name: 'Fridge', price: 70000 }
  ];

  return (
    <div className="App">
      <h1>Items Ordered :</h1>
      <table>
        <thead>
          <tr>
            <th>Name</th>
            <th>Price</th>
          </tr>
        </thead>
        <tbody>
          {items.map((item, index) => (
            <tr key={index}>
              <td>{item.name}</td>
              <td>{item.price}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  );
}

export default App;
```

File:App.css

```
.App {
  text-align: center;
  margin-top: 50px;
}

h1 {
  color: green;
```

```
  font-family: Arial, sans-serif;
}

table {
  margin: 0 auto;
  border-collapse: collapse;
  font-family: Arial, sans-serif;
  color: green;
}

table, th, td {
  border: 1px solid gray;
  padding: 10px 20px;
}

th {
  font-weight: bold;
  text-align: center;
}
```

OUTPUT:

Items Ordered :

Name	Price
Laptop	80000
TV	120000
Washing Machine	50000
Mobile	30000
Fridge	70000

EXERCISE 3:

Create a React App “counterapp” which will have a component named “CountPeople” which will have 2 methods.

CODE:

File:App.js

```
import React from 'react';
import CountPeople from './CountPeople';

function App() {
  return (
    <div className="App">
      <CountPeople />
    </div>
  );
}

export default App;
```

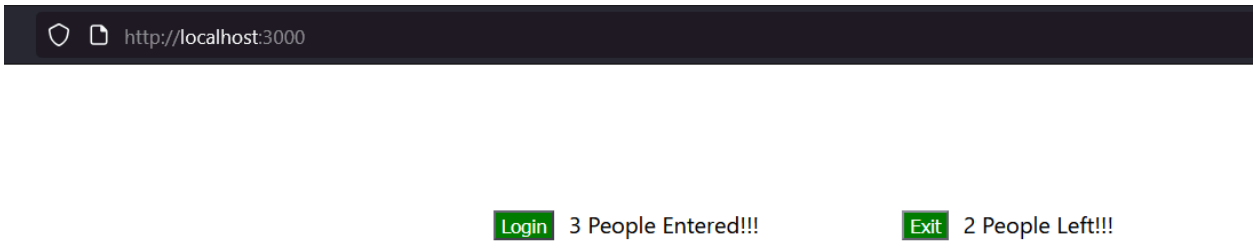
File:CountPeople.js

```
import React, { Component } from 'react';

class CountPeople extends Component {
  constructor() {
    super();
    this.state = {
      entrycount: 0,
      exitcount: 0
    };
  }

  updateEntry = () => {
    this.setState((prevState) => {
      return { entrycount: prevState.entrycount + 1 };
    });
  };

  updateExit = () => {
    this.setState((prevState) => {
      return { exitcount: prevState.exitcount + 1 };
    });
```

```
  };

  render() {
    return (
      <div style={{ display: 'flex', justifyContent: 'center', marginTop:
'100px' }}>
        <div style={{ marginRight: '100px' }}>
          <button onClick={this.updateEntry} style={{ backgroundColor:
'green', color: 'white' }}>Login</button>
          <span style={{ marginLeft: '10px' }}>{this.state.entrycount}
People Entered!!!</span>
        </div>
        <div>
          <button onClick={this.updateExit} style={{ backgroundColor:
'green', color: 'white' }}>Exit</button>
          <span style={{ marginLeft: '10px' }}>{this.state.exitcount}
People Left!!!</span>
        </div>
      </div>
    );
  }
}
export default CountPeople;
```

OUTPUT:

Login 1 People Entered!!! Exit 0 People Left!!!



EXERCISE 4:

Build a React application that displays a list of 11 cricket players with their names, country, and scores.

CODE:

File:App.js

```
import React from 'react';
import AllPlayers from './components/AllPlayers';
import ScoreAbove70 from './components/ScoreAbove70';
import OddPlayers from './components/OddPlayers';
import EvenPlayers from './components/EvenPlayers';
import IndianPlayers from './components/IndianPlayers';

const players = [
    { name: "Virat Kohli", score: 95 },
    { name: "Rohit Sharma", score: 85 },
    { name: "Shubman Gill", score: 60 },
    { name: "Ravindra Jadeja", score: 45 },
    { name: "KL Rahul", score: 72 },
    { name: "Rishabh Pant", score: 30 },
    { name: "Hardik Pandya", score: 88 },
    { name: "Suryakumar Yadav", score: 54 },
    { name: "Jasprit Bumrah", score: 92 },
    { name: "Mohammed Shami", score: 66 },
    { name: "Kuldeep Yadav", score: 71 },
];

function App() {
```

```
  return (
    <div style={{ padding: '20px' }}>
      <h1>🏏 Cricket App</h1>
      <AllPlayers players={players} />
      <ScoreAbove70 players={players} />
      <OddPlayers players={players} />
      <EvenPlayers players={players} />
      <IndianPlayers players={players} />
    </div>
  );
}

export default App;
```

File:AllPlayers.js

```
import React from 'react';

function AllPlayers({ players }) {
  return (
    <div>
      <h2>All Players with Scores</h2>
      <ul>
        {players.map((player, index) => (
          <li key={index}>{player.name} - {player.score}</li>
        ))}
      </ul>
    </div>
  );
}

export default AllPlayers;
```

File:EvenPlayers.js

```
import React from 'react';

function EvenPlayers({ players }) {
  return (
    <div>
      <h2>Even Index Players</h2>
      <ul>
```

```
        {players.map((player, index) => (
          index % 2 === 0 ? <li key={index}>{player.name}</li> : null
        ))}
      </ul>
    </div>
  );
}

export default EvenPlayers;
```

File:IndianPlayers.js

```
import React from 'react';

function IndianPlayers({ players }) {
  return (
    <div>
      <h2>List of Indian Players</h2>
      <ul>
        {players.map((player, index) => (
          <li key={index}>{player.name}</li>
        ))}
      </ul>
    </div>
  );
}

export default IndianPlayers;
```

File:OddPlayers.js

```
import React from 'react';

function OddPlayers({ players }) {
  return (
    <div>
      <h2>Odd Index Players</h2>
      <ul>
        {players.map((player, index) => (
          index % 2 !== 0 ? <li key={index}>{player.name}</li> : null
        ))}
      </ul>
```

```
    </div>
  );
}

export default OddPlayers;
```

File:ScoreAbove70.js

```
import React from 'react';

function ScoreAbove70({ players }) {
  const filtered = players.filter(player => player.score >= 70);

  return (
    <div>
      <h2>Players with Score ≥ 70</h2>
      <ul>
        {filtered.map((player, index) => (
          <li key={index}>{player.name} - {player.score}</li>
        ))}
      </ul>
    </div>
  );
}

export default ScoreAbove70;
```

OUTPUT:

🏏 Cricket App

All Players with Scores

- Virat Kohli - 95
- Rohit Sharma - 85
- Shubman Gill - 60
- Ravindra Jadeja - 45
- KL Rahul - 72
- Rishabh Pant - 30
- Hardik Pandya - 88
- Suryakumar Yadav - 54
- Jasprit Bumrah - 92
- Mohammed Shami - 66
- Kuldeep Yadav - 71

Players with Score ≥ 70

- Virat Kohli - 95
- Rohit Sharma - 85
- KL Rahul - 72
- Hardik Pandya - 88
- Jasprit Bumrah - 92
- Kuldeep Yadav - 71

Odd Index Players

- Rohit Sharma
- Ravindra Jadeja
- Rishabh Pant
- Suryakumar Yadav
- Mohammed Shami

Even Index Players

- Virat Kohli
- Shubman Gill
- KL Rahul
- Hardik Pandya
- Jasprit Bumrah
- Kuldeep Yadav

List of Indian Players

- Virat Kohli
- Rohit Sharma
- Shubman Gill
- Ravindra Jadeja
- KL Rahul
- Rishabh Pant
- Hardik Pandya
- Suryakumar Yadav
- Jasprit Bumrah
- Mohammed Shami
- Kuldeep Yadav

EXERCISE 5:

Create a React Application named “officespacerentalapp” which uses React JSX to create elements, attributes and renders DOM to display the page.

CODE:

FILE:App.js

```
import React from 'react';
import './App.css';
import officeImage from './office_space.jpg';

function App() {
  const element = "Office Space";
  const ItemName = { Name: "DBS", Rent: 50000, Address: "Chennai" };
  const image1 = <img src={officeImage} alt="Office Space" width="25%"
height="25%" />;


  const colors = [];
  if (ItemName.Rent <= 60000) {
    colors.push('textRed');
  } else {
    colors.push('textGreen');
  }


  const rentColorClass = ItemName.Rent <= 60000 ? 'textRed' : 'textGreen';

  return (
    <div className="App">
      <h1 className="main-heading">{element}, at Affordable Range</h1>
      {image1}
      <div className="info-container">
        <h1>Name: {ItemName.Name}</h1>
        {}
        <h3 className={`rent ${rentColorClass}`}>Rent Rs.
{ItemName.Rent}</h3>
        <h3>Address: {ItemName.Address}</h3>
      </div>
    </div>
  );
```

```
}

export default App;
```

File:App.css

```
.App {
  display: flex;
  flex-direction: column;
  align-items: center;
  text-align: center;
  margin-top: 50px;
  font-family: Arial, sans-serif;
}

.main-heading {
  font-size: 32px;
  font-weight: bold;
  margin-bottom: 20px;
}

img {
  margin-bottom: 30px;
}

.info-container {
  text-align: left;
}


.info-container h1,
.info-container h3 {
  margin: 5px 0;
  font-weight: normal;
  font-size: 20px;
}
```

```css
.rent {
  font-weight: bold;
}


.textRed {
  color: red;
}

.textGreen {
  color: green;
}
```

OUTPUT:

http://localhost:3000

Office Space, at Affordable Range

Name: DBS
Rent Rs. 50000
Address: Chennai