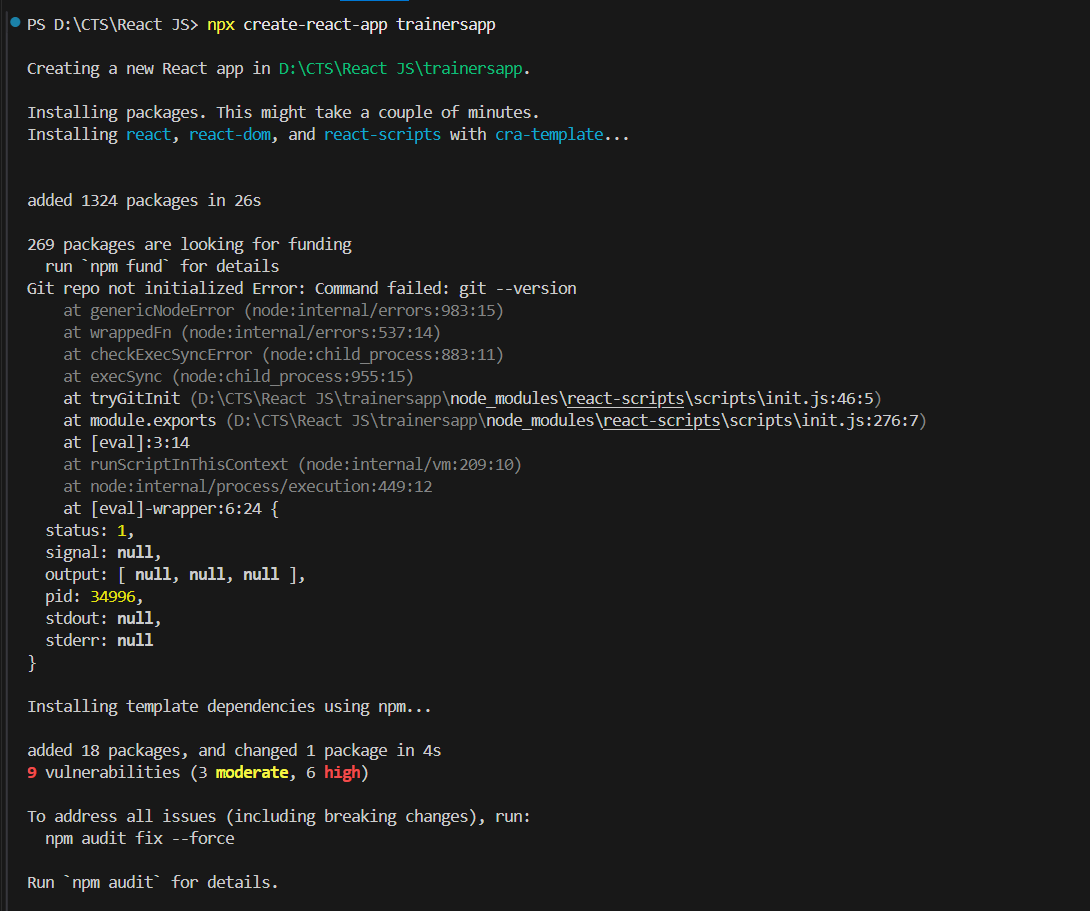
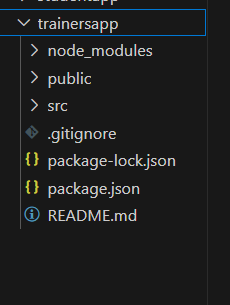
**Cognizant Academy teams want to maintain a list of trainers along with their expertise in a SPA using React as the technology. You are assigned the task of creating this React app.**

**The following trainers’ data application will deal.**

1. **T-ID**
2. **Name**
3. **Phone**
4. **Email**
5. **Stream**
6. **Skills**
7. Create a new React app using *create-react-app* tool with the as “TrainersApp”  
   **npx create-react-app trainersapp**



1. Open the application using the VS Code



1. Add a new file called *trainer.js* inside the **src folder** and define a class named as “Trainer” with the following properties
   1. TrainerId
   2. Name
   3. Email
   4. Phone
   5. Technology
   6. Skills

**D:\CTS\React JS\trainersapp\src\trainer.js**

class Trainer {

    constructor(trainerid, name, email, phone, technology, skills) {

        this.trainerid = trainerid;

        this.name = name;

        this.email = email;

        this.phone = phone;

        this.technology = technology;

        this.skills = skills;

    }

}

export default Trainer;

1. Create a new TrainersMock.js file which will contain the mock trainer data. Refer the following screenshot for mock data

**D:\CTS\React JS\trainersapp\src\TrainersMock.js**

import Trainer from "./trainer";

const trainersMock = [

    new Trainer('t-syed8',

        'Syed Khaleelullah',

        'khaleelullah@cognizant.com',

        '98767561962',

        '.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',

        '987112235',

        'Python',

        ['Python', 'Django', 'Angular'])

];

export default trainersMock;

1. Install the support for React router for the dom. Execute the following command.

**cd trainersapp**

**npm install react-router-dom@6**

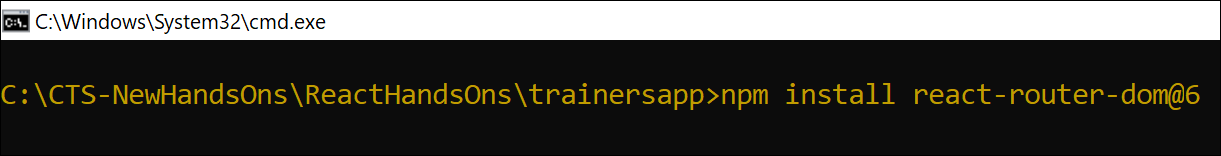


Figure 4: Install React Router

1. Create new component named as **TrainersList** inside *Trainerlist.js* file. The component should accept the trainer’s data as parameter and render it as a list. The list should display names of each trainers which must be clickable like a hyper link. Refer the following screenshot for the component layout.



Figure 5: TrainersList Component

**D:\CTS\React JS\trainersapp\src\Trainerlist.js**

import React from 'react';

function TrainersList({ trainers }) {

    return (

        <div>

            <h2>Trainers List</h2>

            <ul>

                {trainers.map(trainer => (

                    <li key={trainer.trainerid}>

                        <a href={`mailto:${trainer.email}`} style={{ color: 'blue', textDecoration: 'underline', cursor: 'pointer' }}>

                            {trainer.name}

                        </a>

                    </li>

                ))}

            </ul>

        </div>

    );

}

export default TrainersList;

1. Create a new component named as Home inside Home.js which will be responsible for displaying the following



Figure 6: Home Component

import React from 'react';

function Home() {

    return (

        <div>

            <h1>Welcome to Trainers Portal</h1>

        </div>

    );

}

export default Home;

1. Modify the App component to add support for routing and defining the navigation links to Home component and TrainersList component. Use BrowserRouter, Routes, Route and Link components from the react-router-dom library.

Define the following URL

1. / - must redirect to home component
2. /trainers – must redirect to trainers list component.

The layout of the page must be similar to the following



Figure 7: App Component

**D:\CTS\React JS\trainersapp\src\App.js**

import React from 'react';

import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';

import Home from './Home';

import TrainersList from './Trainerlist';

import TrainerDetail from './TrainerDetails';

const trainers = [

  {

    trainerid: 1,

    name: 'Syed Khaleelullah',

    email: 'khaleelullah@cognizant.com',

    phone: '97676516962',

    technology: '.NET',

    skills: ['C#', 'SQL Server', 'React', '.NET Core']

  },

  {

    trainerid: 2,

    name: 'Jojo Jose',

    email: 'jojo@email.com',

    phone: '9876543210',

    technology: 'Java',

    skills: ['Java', 'Spring', 'Hibernate']

  },

  {

    trainerid: 3,

    name: 'Elisa Jones',

    email: 'elisa@email.com',

    phone: '9123456780',

    technology: 'Python',

    skills: ['Python', 'Django', 'Flask']

  }

];

function App() {

  return (

    <BrowserRouter>

      <div style={{ padding: '20px' }}>

        <h1 style={{ fontSize: '2.5rem', marginBottom: '10px' }}>My Academy Trainers App</h1>

        <nav style={{ marginBottom: '20px', fontSize: '1.1rem' }}>

          <Link to="/" style={{ marginRight: '10px' }}>Home</Link>

          |

          <Link to="/trainers" style={{ marginLeft: '10px' }}>Show Trainers</Link>

        </nav>

        <Routes>

          <Route path="/" element={<Home />} />

          <Route path="/trainers" element={<TrainersList trainers={trainers} />} />

          <Route path="/trainer/:id" element={<TrainerDetail />} />

        </Routes>

      </div>

    </BrowserRouter>

  );

}

export default App;

1. Create a new component named **TrainerDetail** in *TrainerDetails.js* file.

The component should retrieve a parameter named id from the URL with the help of “useParams” hook from the React router DOM library.

It should query the mock trainer data using the id and display the trainer details as show in screenshot.

Modify the TrainersList component to add Links to TrainerDetail component while passing the ID. Define a route in App component for the same.

**D:\CTS\React JS\trainersapp\src\TrainerDetails.js**

import React from 'react';

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

const trainers = [

    {

        trainerid: 1,

        name: 'Syed Khaleelullah',

        email: 'khaleelullah@cognizant.com',

        phone: '97676516962',

        technology: '.NET',

        skills: ['C#', 'SQL Server', 'React', '.NET Core']

    },

    {

        trainerid: 2,

        name: 'Jojo Jose',

        email: 'jojo@email.com',

        phone: '9876543210',

        technology: 'Java',

        skills: ['Java', 'Spring', 'Hibernate']

    },

    {

        trainerid: 3,

        name: 'Elisa Jones',

        email: 'elisa@email.com',

        phone: '9123456780',

        technology: 'Python',

        skills: ['Python', 'Django', 'Flask']

    }

];

function TrainerDetail() {

    const { id } = useParams();

    const trainer = trainers.find(t => t.trainerid === parseInt(id));

    if (!trainer) {

        return <div>Trainer not found.</div>;

    }

    return (

        <div>

            <h2>Trainers Details</h2>

            <h3 style={{ fontWeight: 'bold' }}>

                {trainer.name} ({trainer.technology})

            </h3>

            <div>{trainer.email}</div>

            <div>{trainer.phone}</div>

            <ul>

                {trainer.skills.map(skill => (

                    <li key={skill}>{skill}</li>

                ))}

            </ul>

        </div>

    );

}

export default TrainerDetail;

1. Build and run the application. The complete layout of the application will look as follows.

**Output:**

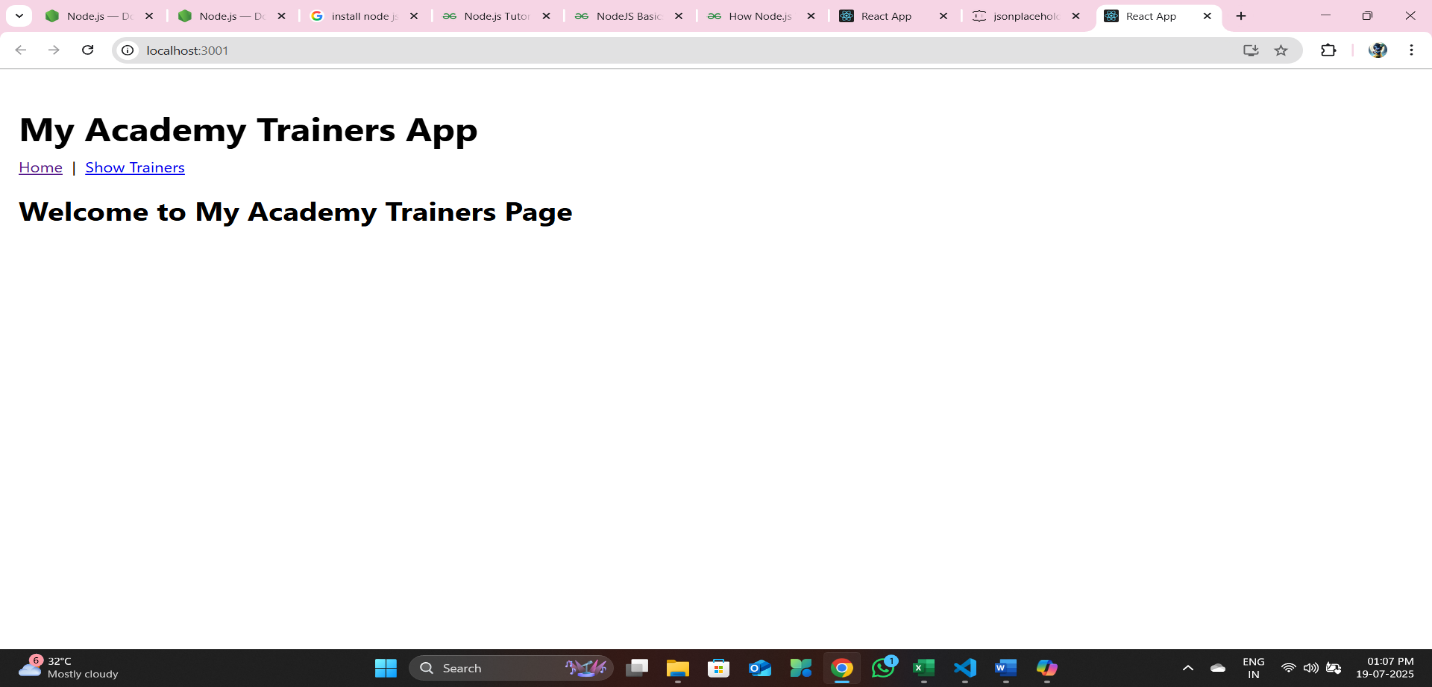


Figure: Home

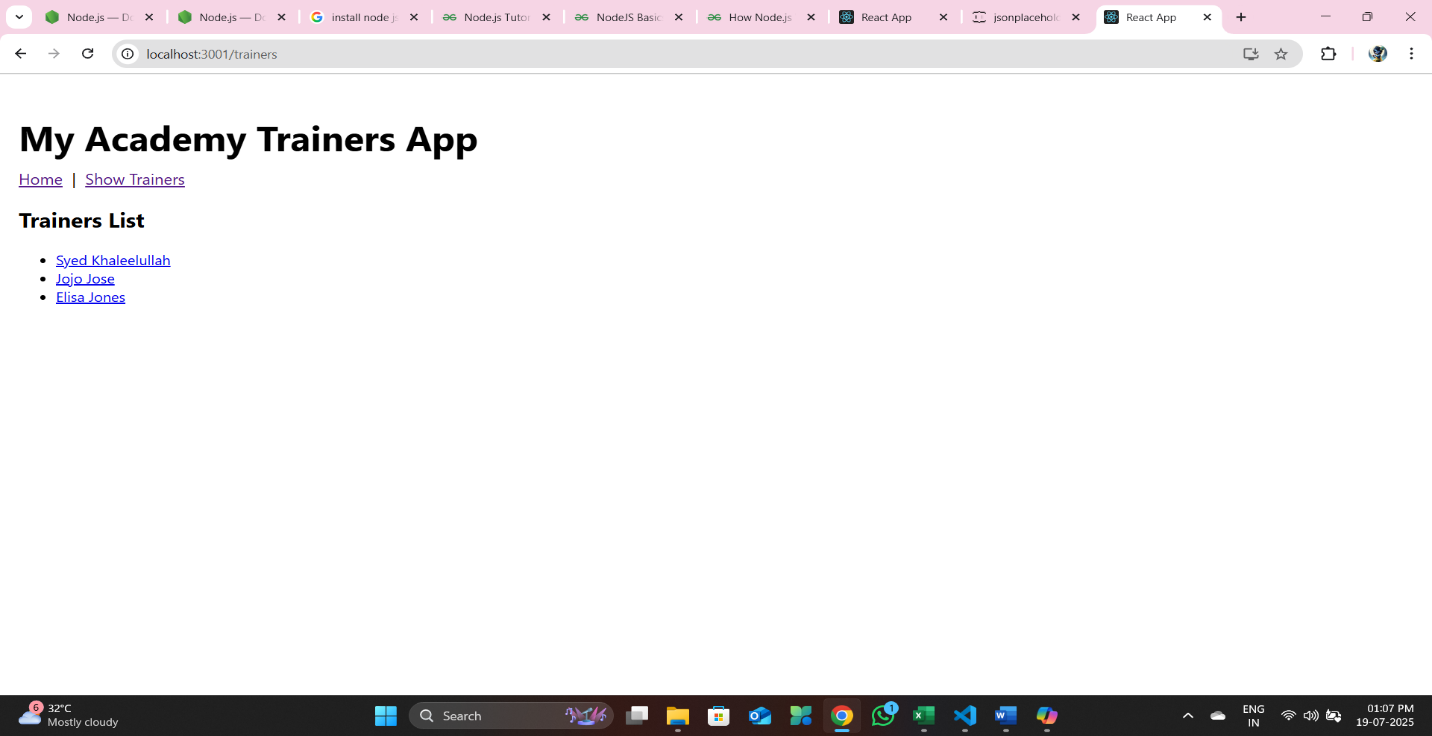


Figure : Trainers List

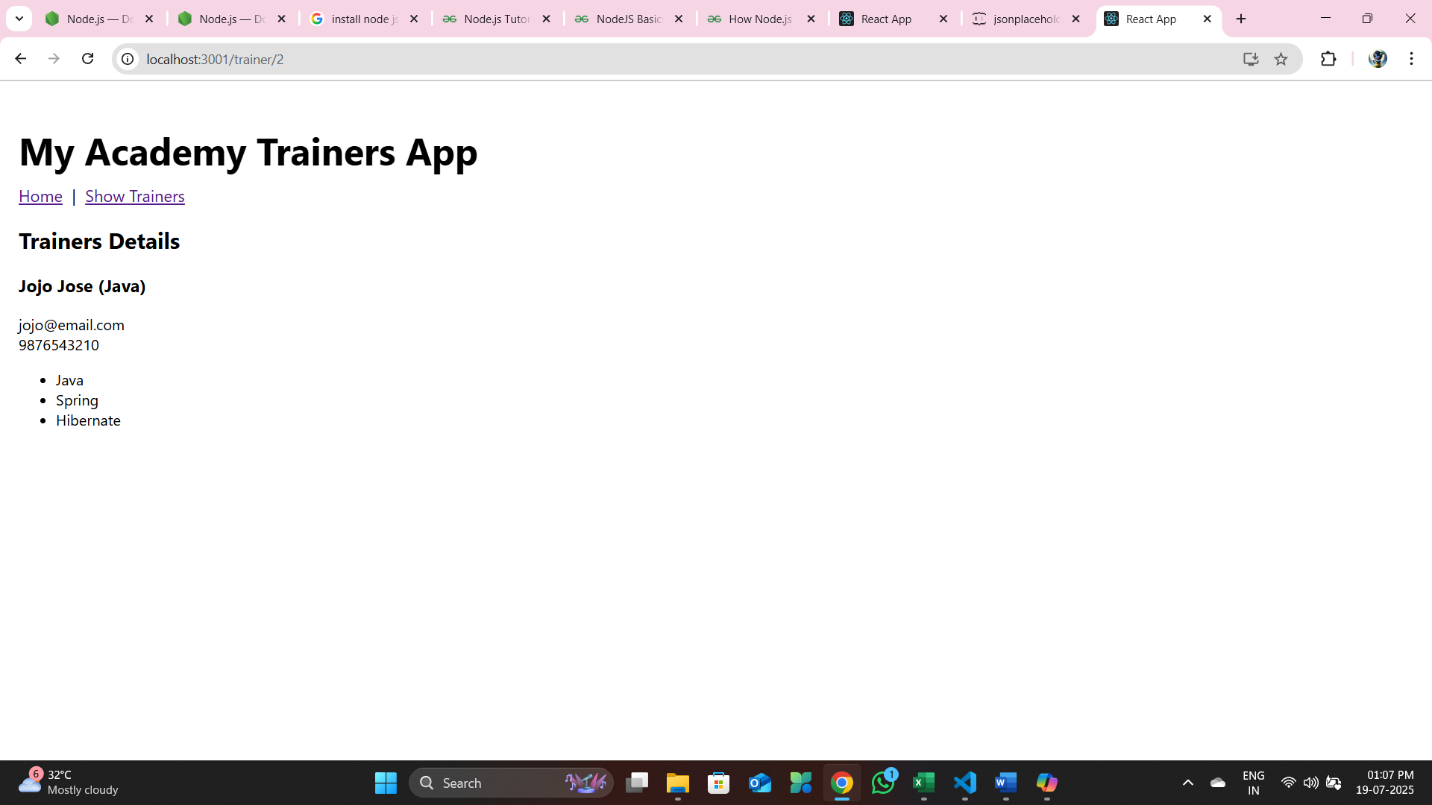
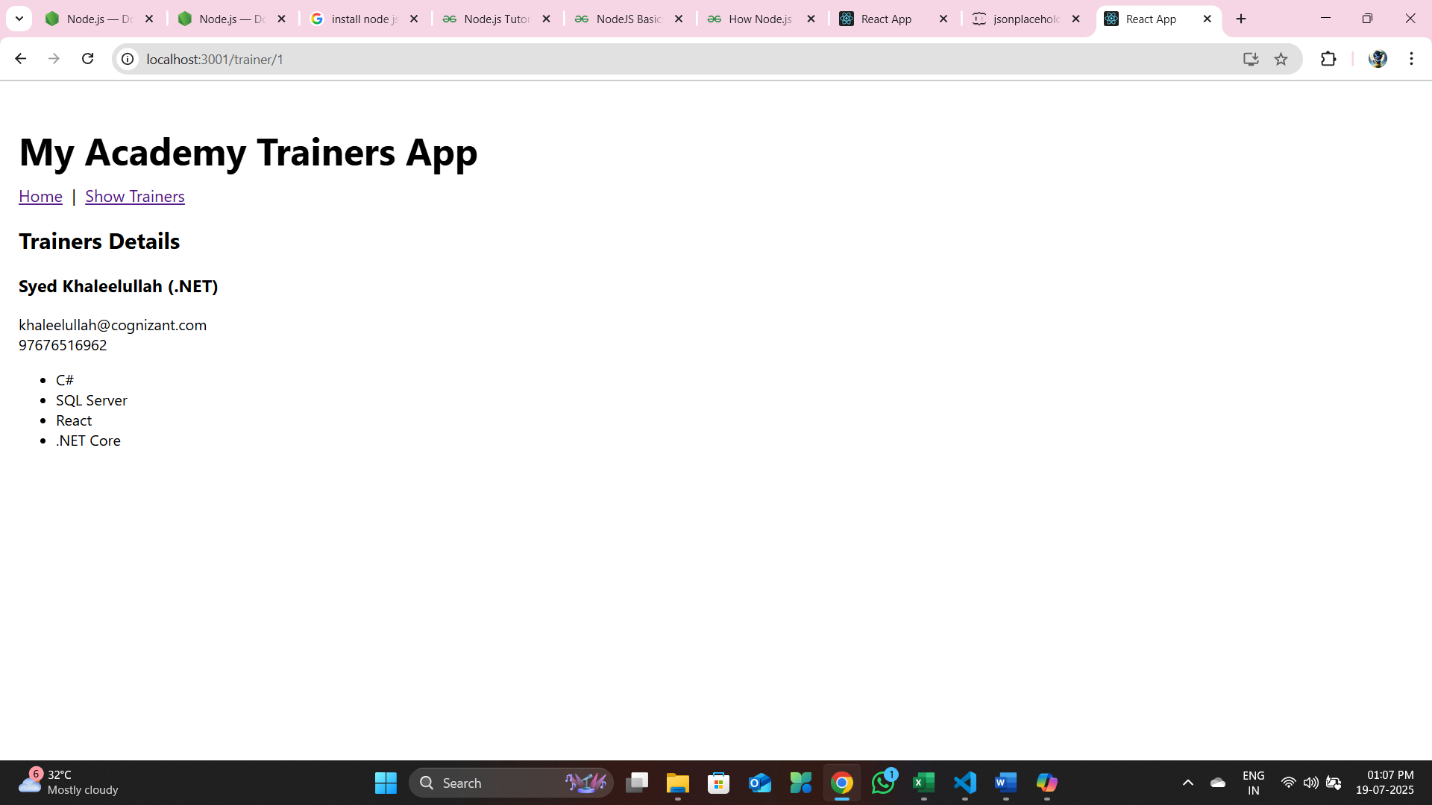


Figure : Trainer Details