**Skill:-React**

**Filename:- 1. ReactJS-HOL**

**Solution:-**

**1. Define SPA and its benefit**s

**Answer:-** SPA (Single Page Application) is a web application that loads a single HTML page and dynamically updates content without refreshing the whole page.  
Benefits:  
- Faster navigation after initial load  
- Smooth user experience  
- Reduced server requests

**2. Define React and identify its working**

**Answer:-** React is a JavaScript library used for building user interfaces, especially SPAs.  
Working: It uses a component-based structure and a virtual DOM to efficiently update the UI based on changes in application state.

**3. Identify the differences between SPA and MPA**

**Answer:-**

- SPA (Single Page Application): Loads a single page; content changes dynamically via JavaScript.  
- MPA (Multi Page Application): Loads a new HTML page from the server on every request.  
SPA provides better speed and UX; MPA is better for SEO and complex apps.

**4. Explain Pros & Cons of Single-Page Application**

**Answer:-** Pros:  
- Fast and responsive UI  
- Less server load  
- Smooth transitions  
Cons:  
- Poor SEO  
- JavaScript must be enabled  
- Initial load time may be longer

**5. Explain about Reac**t

**Answer:-** React is an open-source JavaScript library developed by Facebook. It is used to create fast, interactive, and reusable UI components in single-page applications.

**6. Define virtual DOM**

**Answer:-** The virtual DOM is an in-memory copy of the real DOM. React compares the virtual DOM with the real DOM to detect changes and updates only the changed parts, improving performance.

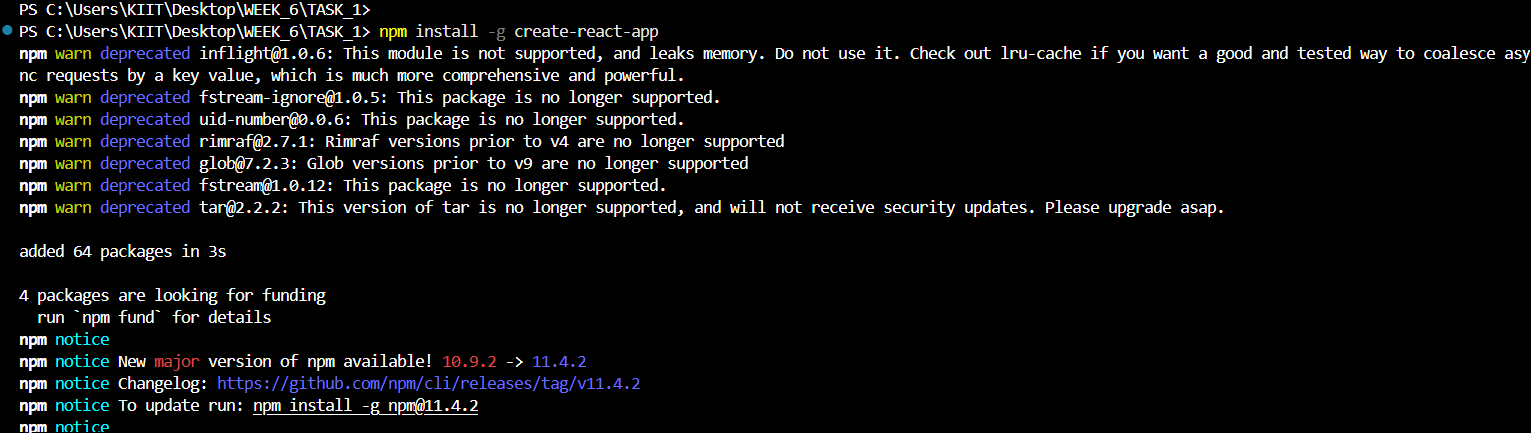
**7. Explain Features of React**

**Answer:-**

- Virtual DOM for fast updates  
- Component-based structure  
- One-way data binding  
- JSX for writing HTML in JavaScript  
- Reusable components

***Created and Installed React App***

***Command:-* npm install -g create-react-app**



***Created a New React App***

***Command:-* npx create-react-app myfirstreact**



***Go to src/App.js and Change the code with this***

function App() {

  return (

    <div>

      <h1>Welcome to the first session of React</h1>

    </div>

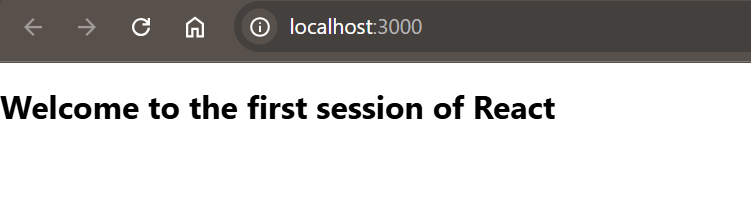
  );

}

export default App;

***Starting the React Application on local Server by* npm start**

***Output:-***



**Filename:- 2. ReactJS-HOL**

**Solution:-**

**1. Explain React components**

**Answer:-**

React components are independent, reusable pieces of UI. They can be thought of as JavaScript functions that return HTML elements.

**2. Identify the differences between components and JavaScript functions**

**Answer:-**- React components return JSX (HTML-like syntax), while JavaScript functions return values.  
- Components manage state and lifecycle methods; JavaScript functions do not.

**3. Identify the types of components**

**Answer:-**

There are two main types of React components:  
- Class components  
- Function components

**4. Explain class component**

**Answer:-**

Class components are ES6 classes that extend from React.Component and have a render() method to return JSX. They can also use state and lifecycle methods.

**5. Explain function component**

**Answer:-** Function components are simpler and are written as JavaScript functions. They take props as input and return JSX. With React Hooks, they can also manage state and side effects.

**6. Define component constructor**

**Answer:-**

The constructor is a special method used in class components to initialize state and bind methods. It is called before the component is mounted.

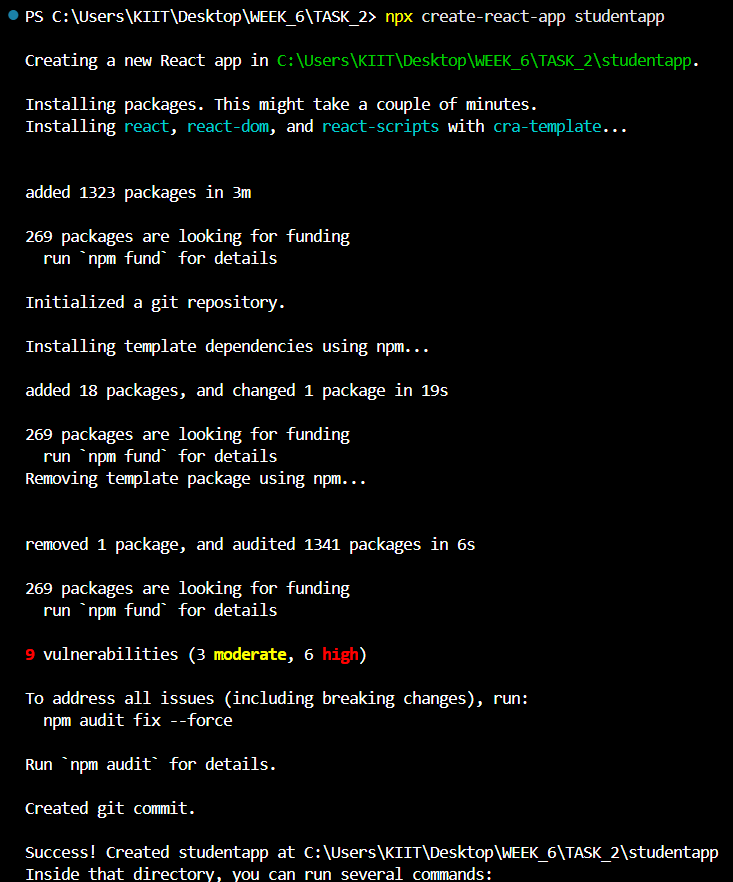
**7. Define render() function**

**Answer:-**

The render() function is required in class components. It returns the JSX that defines the UI of the component.

***Created and Installed React App***

***Command:-*  npx create-react-app studentapp**



***Created a Home.js in Inside src/Home.js and add this code***

import React from 'react';

function Home() {

  return (

    <div>

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

    </div>

  );

}

export default Home;

***Create a About.js in Inside src/About.js and add this code***

import React from 'react';

function About() {

  return (

    <div>

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

    </div>

  );

}

export default About;

***Create a Contact.js in Inside src/Contact.js and add this code***

import React from 'react';

function Contact() {

  return (

    <div>

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

    </div>

  );

}

export default Contact;

***Update the code of App.js in src/App.js***

import Home from './Home';

import About from './About';

import Contact from './Contact';

function App() {

  return (

    <div>

      <Home />

      <About />

      <Contact />

    </div>

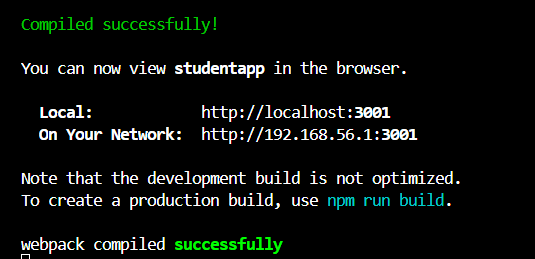
  );

}

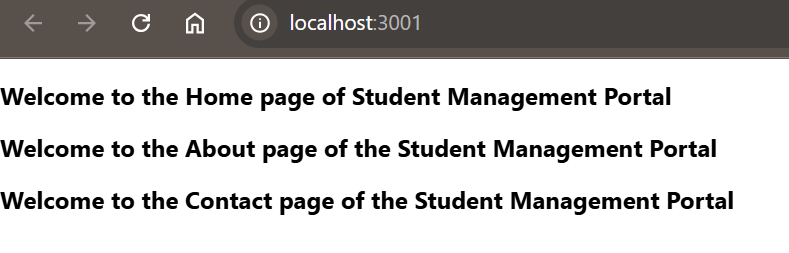
export default App;

***Starting the React Application on local Server by* npm start**

***Output:-***



***Output on Local Server localhost:3001***



**Filename:- 3. ReactJS-HOL**

**Solution:-**

**1. Explain React components**

**Answer:-**

React components are independent, reusable pieces of UI. They can be thought of as JavaScript functions that return HTML elements.

**2. Identify the differences between components and JavaScript functions**

**Answer:-**- React components return JSX (HTML-like syntax), while JavaScript functions return values.  
- Components manage state and lifecycle methods; JavaScript functions do not.

**3. Identify the types of components**

**Answer:-**

There are two main types of React components:  
- Class components  
- Function components

**4. Explain class component**

**Answer:-**

Class components are ES6 classes that extend from React.Component and have a render() method to return JSX. They can also use state and lifecycle methods.

**5. Explain function component**

**Answer:-** Function components are simpler and are written as JavaScript functions. They take props as input and return JSX. With React Hooks, they can also manage state and side effects.

**6. Define component constructor**

**Answer:-**

The constructor is a special method used in class components to initialize state and bind methods. It is called before the component is mounted.

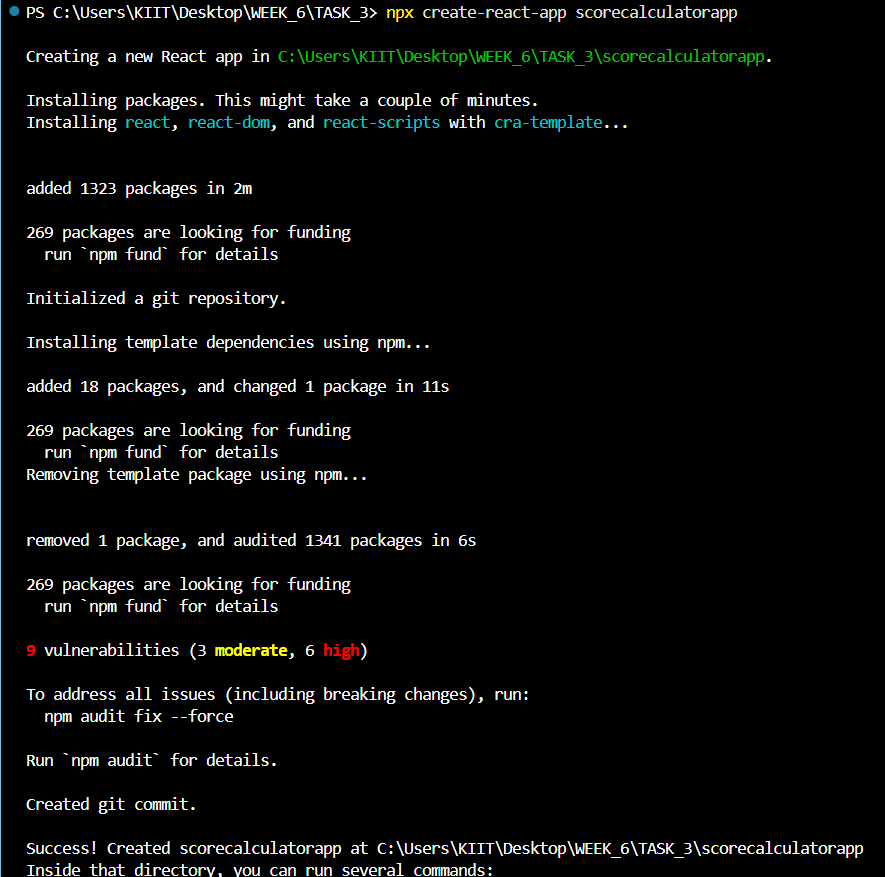
**7. Define render() function**

**Answer:-**

The render() function is required in class components. It returns the JSX that defines the UI of the component.

***Created and Installed React App***

***Command:-*  npx create-react-app scorecalculatorapp**



***Created a CalculateScore.js in src/Components Folder with the below code***

import React from 'react';

import '../Stylesheets/mystyle.css';

function CalculateScore() {

  const name = "John";

  const school = "ABC High School";

  const total = 450;

  const goal = 500;

  const average = (total / goal) \* 100;

  return (

    <div className="box">

      <h2>Score Calculator</h2>

      <p>Name: {name}</p>

      <p>School: {school}</p>

      <p>Total Marks: {total}</p>

      <p>Goal: {goal}</p>

      <p>Average Score: {average}%</p>

    </div>

  );

}

export default CalculateScore;

***Created a New\_Folder with name stylesheets in that create a mystyle.css***

.box {

  border: 2px solid black;

  padding: 20px;

  margin: 20px;

  background-color: #f2f2f2;

  width: 300px;

}

***Update the code of App.js in src/App.js***

import React from 'react';

import CalculateScore from './Components/CalculateScore';

function App() {

  return (

    <div>

      <CalculateScore />

    </div>

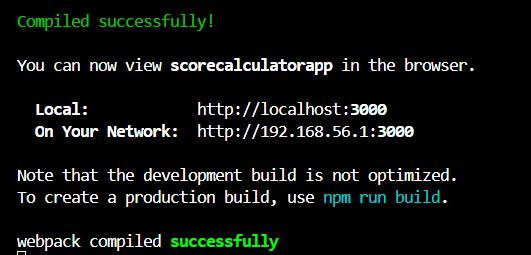
  );

}

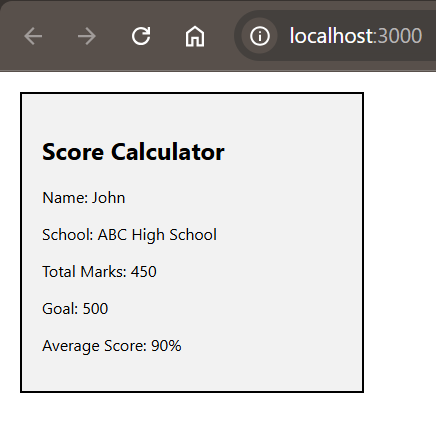
export default App;

***Starting the React Application on local Server* npm start**

***Output:-***



***Output on Local Server localhost:3000***



**Filename:- 4. ReactJS-HOL**

**Solution:-**

**1. Explain the need and Benefits of component life cycle**

**Answer:-**

The component lifecycle in React helps developers manage the creation, updating, and destruction of components.  
Benefits:  
- Optimize performance  
- Manage resources efficiently  
- Run specific code at particular stages of component life  
- Control UI behavior during mounting, updating, and unmounting.

**2. Identify various life cycle hook methods**

**Answer:-**

Major lifecycle methods in class components:  
- constructor()  
- render()  
- componentDidMount()  
- componentDidUpdate()  
- componentWillUnmount()

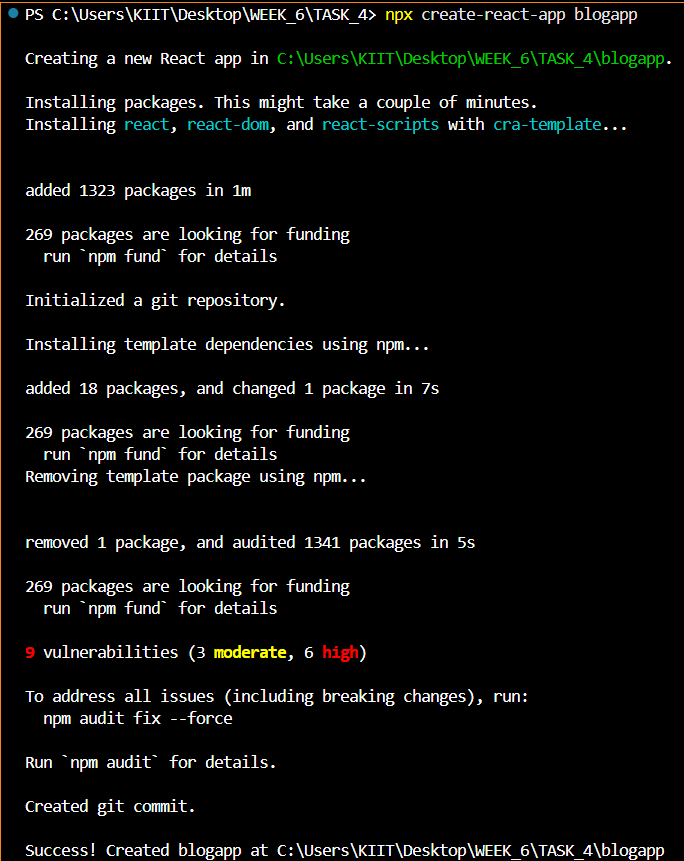
**3. List the sequence of steps in rendering a component**

**Answer:-**

Sequence of component rendering:  
1. constructor()  
2. render()  
3. componentDidMount()  
4. componentDidUpdate() (on update)  
5. componentWillUnmount() (on removal)

***Created and Installed React App***

***Command:-*  npx create-react-app blogapp**



***Created a Post.js inside the folder of src/***

import React from 'react';

class Post extends React.Component {

  render() {

    return (

      <div>

        <h3>{this.props.title}</h3>

        <p>{this.props.body}</p>

      </div>

    );

  }

}

export default Post;

***Created a Posts.js inside the folder of src/***

import React from 'react';

import Post from './Post';

class Posts extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      posts: [],

    };

  }

  loadPosts() {

    fetch('https://jsonplaceholder.typicode.com/posts')

      .then((response) => response.json())

      .then((data) => this.setState({ posts: data }))

      .catch((error) => {

        throw error;

      });

  }

  componentDidMount() {

    this.loadPosts();

  }

  componentDidCatch(error, info) {

    alert('An error occurred: ' + error);

  }

  render() {

    return (

      <div>

        <h2>Blog Posts</h2>

        {this.state.posts.map((post) => (

          <Post key={post.id} title={post.title} body={post.body} />

        ))}

      </div>

    );

  }

}

export default Posts;

***Update the code of App.js in src/App.js***

// src/App.js

import React from 'react';

import Posts from './Posts';

function App() {

  return (

    <div className="App">

      <Posts />

    </div>

  );

}

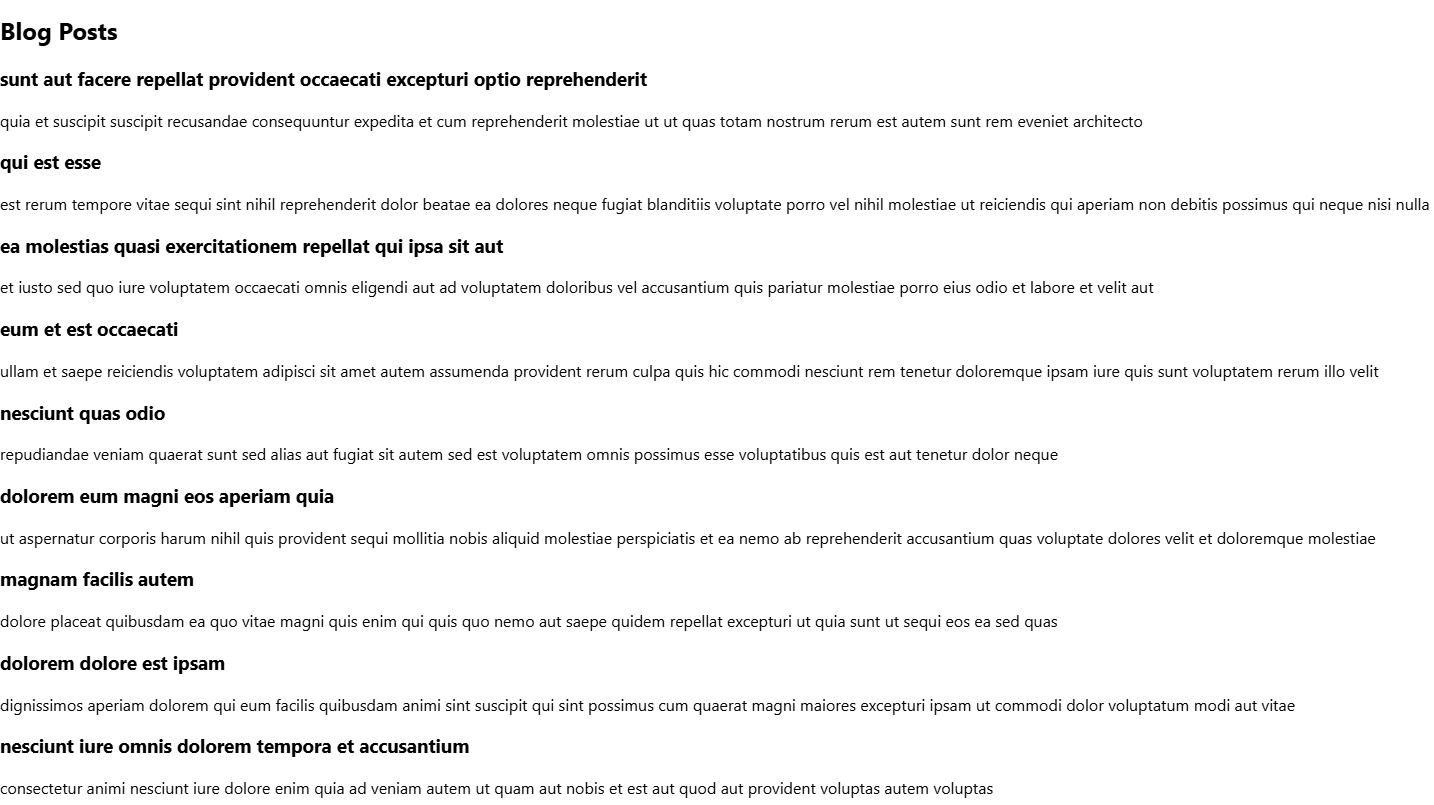
export default App;

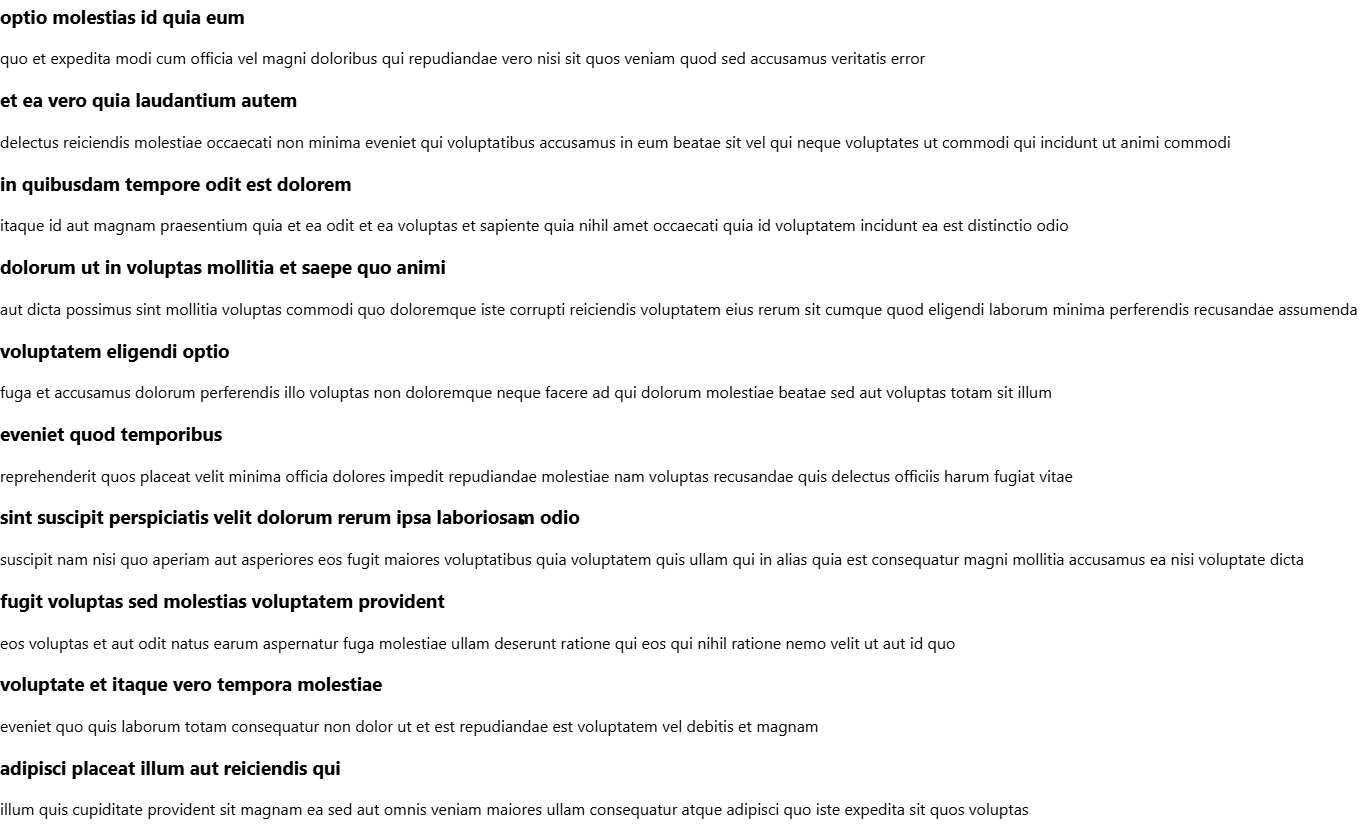
***Starting the React Application on local Server by* npm start**

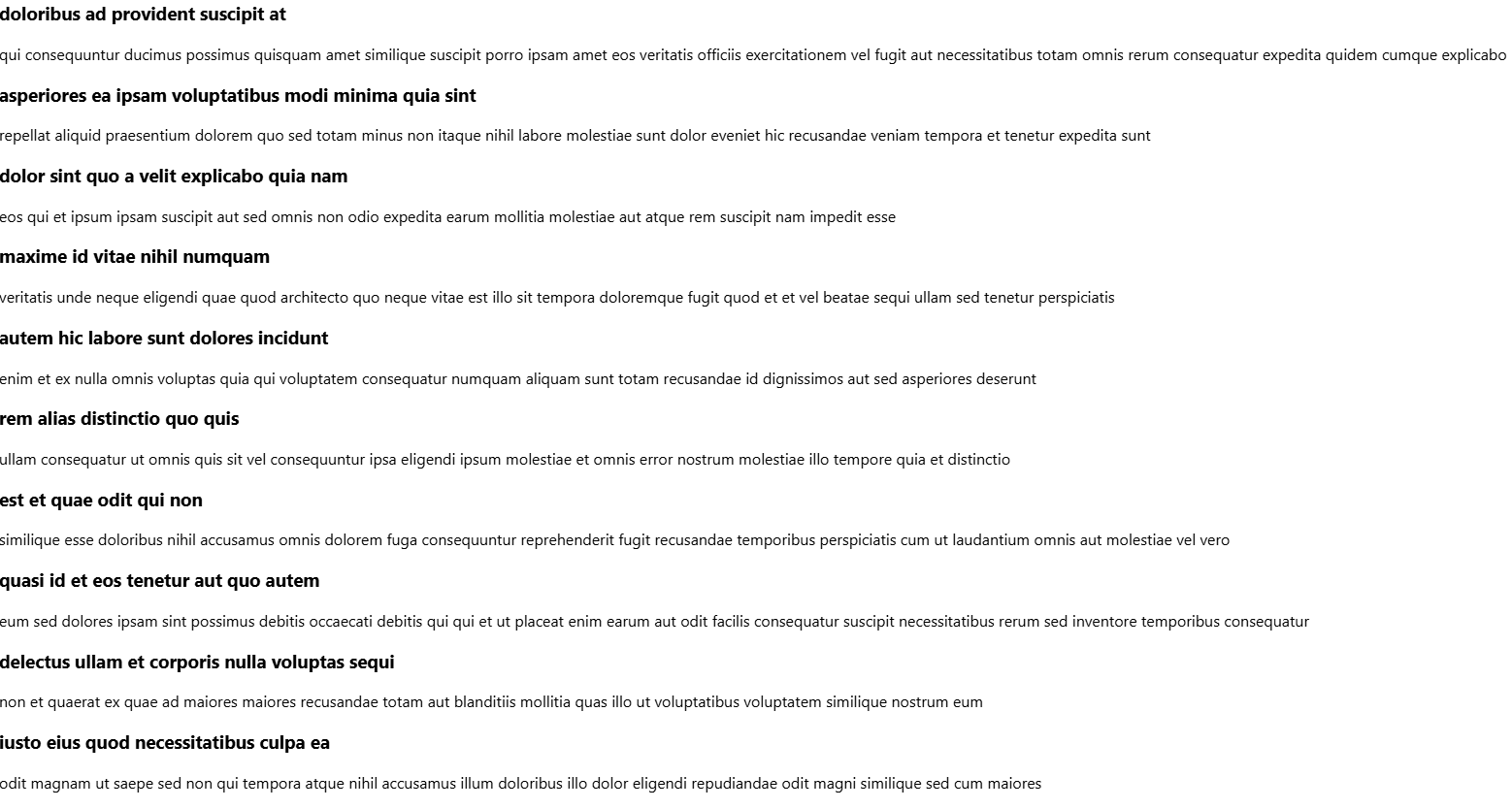
***Output:-***



***Output on Local Server localhost:3000***







**Filename:- 5. ReactJS-HOL**

**Solution:-**

**1. Understanding the need for styling react component**

**Answer:-**

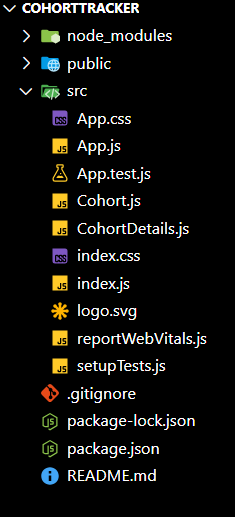
Styling in React components is important for improving the visual layout, enhancing user experience, and maintaining design consistency across the app.

**2. Working with CSS Module and inline styles**

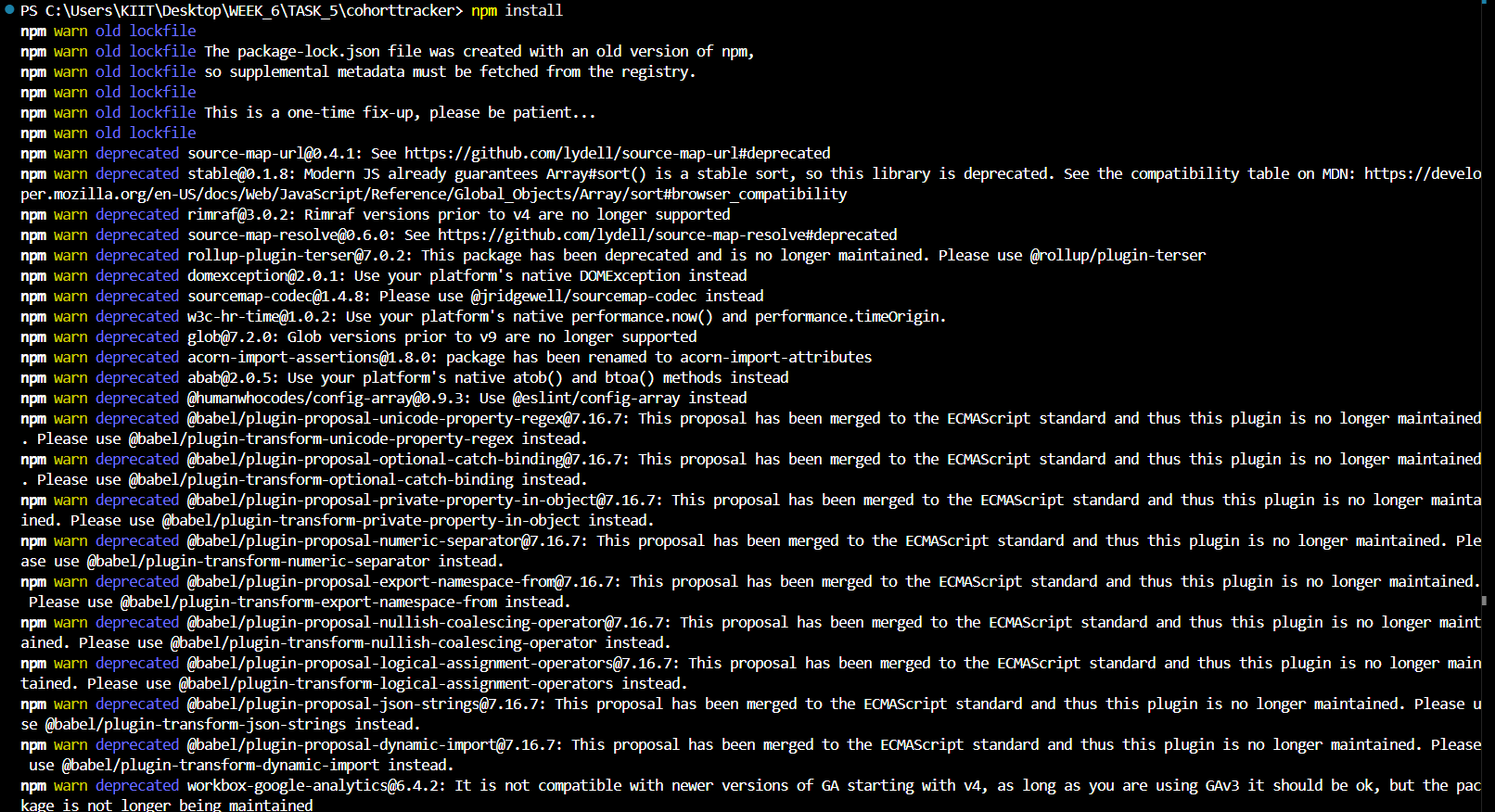
**Answer:-**

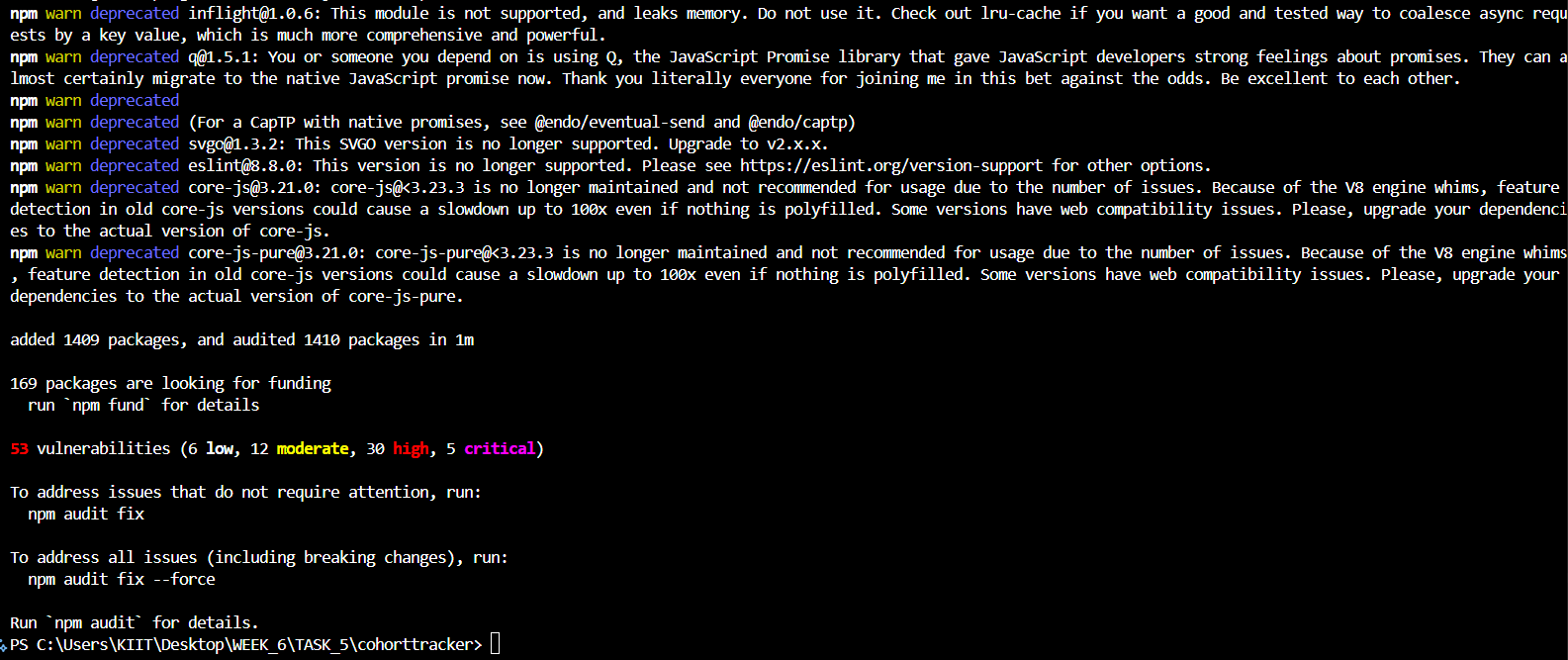
- CSS Modules: A way to locally scope CSS by automatically generating unique class names, preventing style conflicts.  
- Inline Styles: Applying styles directly using a style object within a component, useful for dynamic styling and quick customizations.

***Unzip and open the React application in VS Code***



***Restore node packages in Terminal in same file with* npm install**





***Created the CSS Module File as CohortDetails.module.css in src/ with below code***

.box {

  width: 300px;

  display: inline-block;

  margin: 10px;

  padding: 10px 20px;

  border: 1px solid black;

  border-radius: 10px;

}

dt {

  font-weight: 500;

}

***Update the code of CohortDetails.js in src/CohortDetails.js***

import styles from './CohortDetails.module.css';

function CohortDetails(props) {

    const status = props.cohort.currentStatus?.toLowerCase();

    return (

        <div className={styles.box}>

            <h3 style={{ color: status === "ongoing" ? "green" : "blue" }}>

                {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>

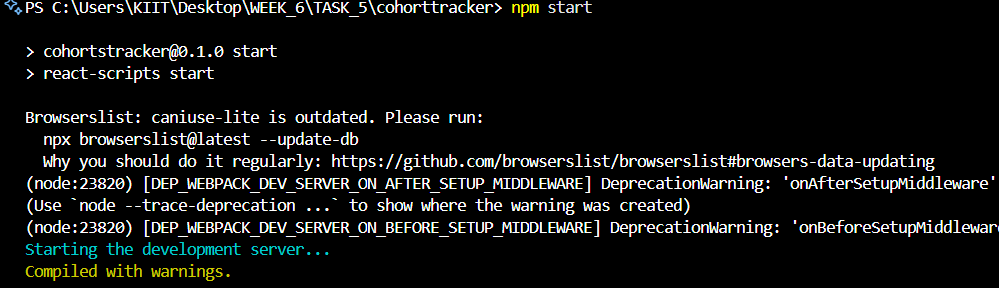
    );

}

export default CohortDetails;

***Starting the React Application on local Server by* npm start**

***Output:-***



***Output on Local Server localhost:3000***

