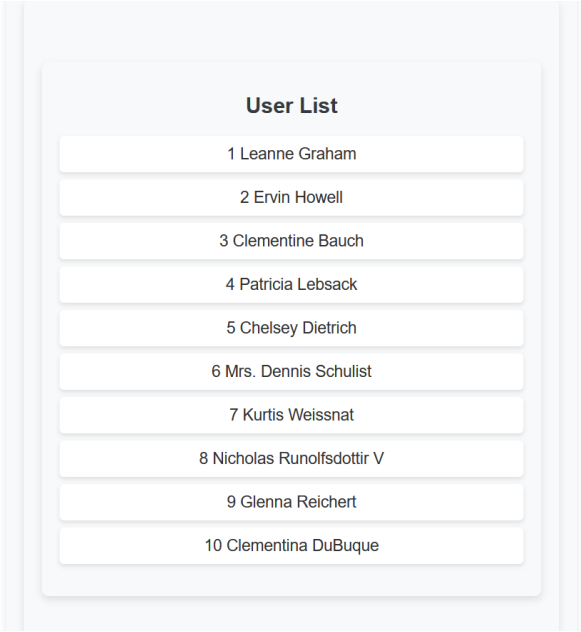

Fetching and Displaying User Data from an API

- **State Management:** Uses state to store the fetched user data and handle errors.
- **API Fetching:** Fetches user data inside componentDidMount() after the component mounts.
- **Error Handling:** Displays an error message if the API request fails.
- **Loading State:** Displays a loading message while the data is being fetched.
- **Rendering:** Renders a list of user names once the data is successfully fetched.

Code:

```
import React from 'react';

class UserProfile extends React.Component {
        constructor(props) {
        super(props);
        this.state = { user: null, error: null };
        }

        componentDidMount() {
      fetch('https://jsonplaceholder.typicode.com/users') // Example API
        .then(response => {
                if (!response.ok) {
                throw new Error('Network response was not ok');
                }
                return response.json();
        })
        .then(data => {
                console.log(data); // Log the data to ensure it's fetched
                this.setState({ user: data });
        })
        .catch(error => {
                console.error('There was an error fetching the data:', error);
                this.setState({ error: error.message });
        });
        }

        render() {
        const { user, error } = this.state;
```

```
        // If there's an error, display it
        if (error) {
        return <div>Error: {error}</div>;
        }

        // If user data is being fetched, show loading message
        if (!user) {
        return <p>Loading...</p>;
        }

        // If user data is available, display the user names
        return (
        <div>
                <h1>User List</h1>
                <ul>
                {user.map((userItem) => (
                <li key={userItem.id}>{userItem.name}</li>
                ))}
                </ul>
        </div>
        );
        }
}
```



Counter with Increment and Decrement Buttons using class component

```
import "./greet.css";
class greet extends React.Component{
   constructor(props){
      super(props);
      this.state={count:0};
   }

    decrement =()=>{
      this.setState({count: this.state.count-1});
    }
    increment =()=>{
      this.setState({count: this.state.count+1});
    }

    render(){
      return(
         <div>
            <p>Count: {this.state.count}</p>
            <button onClick={this.increment}>Increase</button>
            <button onClick={this.decrement}>Decrease</button>
         </div>
      )
    }
}
export default greet;
```

Problem Identification (5)	Execution (5)	Time management (5)	Viva (5)	Total (20)

Result:

Thus the above class component based programs run successfully and verified