coggie

handleInputChange(event) {

this.setState({

[name]: value

const target = event.target;

? target.checked : target.value;

const name = target.name;

const value = target.type === 'checkbox'

```
Stateless Components
                                                                                                                      Stateful Components
                                                                                                                                                                                Gestione dello stato nei componenti
                                                                                                                                                                                                                 4. GESTIONE DELLO
                                                                                                                                                                                                                             STATO
                                                                                                     class App extends Component {
                                                                                                          this.handleChange = (event) => {
                                                                                                               this.setState({filterInput:
                                                                                                     event.target.value})
                                                                                                          render() {
                                                                                                               return <div className="App">
                                                                                                                    <input type="text" value=</pre>
                                                                                                       {this.state.filterInput} onChange=
                                                                                                      {this.handleChange}>
                                                                                                                </div>
                                          this.handleChange = (event) => {
                                           this.setState({filterInput:
                                         event.target.value})
                                                                                                     export default App;
                                onChange={this.handleChange}
                                                                                 Considerazioni
                                                                                                                                                                                             onChange()
                                                Controlled Components
                                                                                  {this.state.filterInput}

    Aggiungere alla WebApp un campo che permetta di filtrare i dati per

    Aggiungere una form che consenta di inserire un nuovo elemento alla lista,

                                      secondo la struttura dati dell'Esercizio #1, oppure di modificarne uno esistente
                                     • ogni elemento della lista dovrà avere un bottone, che lo renda modificabile

    alla richiesta di modifica, dovrà essere popolata la form di inserimento con i

                                      dati dell'elemento selezionato

    se in modalità 'modifica', la form dovrà avere un bottone di salvataggio ed uno

Multiple Input Change Considerazioni
```

Esercizio #5

```
import React, { Component } from 'react';
                       import './App.css';
                      const list = [
                            title: 'React',
                            url: 'https://facebook.github.io/react/',
                             author: 'Jordan Walke',
                             num_comments: 3,
                            points: 4,
                             objectID: 0,
                            title: 'Redux',
                            url: 'https://github.com/reactjs/redux',
                            author: 'Dan Abramov, Andrew Clark',
                             num_comments: 2,
                            points: 5,
                            objectID: 1,
                       class App extends Component {
                           constructor(props) {
                               super(props);
                               this.state = {list} //Inizializzazione dello stato iniziale
                           removeItem = (id) => {
                              const updatedList = this.state.list.filter(item => item.objectID !== id);
                               this.setState({list: updatedList}); //Modifica dello stato a fronte di una variazione
                           render() {
                              const {list} = this.state; //Recupero list da stato corrente
                               return <div className="App">
                                 {list.length > 0 ? list.map(item => <div key={item.objectID}>
                                           <a href={item.url}>{item.title}</a>
                                       <span>{item.author}</span>
                                       <span>{item.num_comments}</span>
                                      <span>{item.points}</span>
                                      <button onClick={() => this.removeItem(item.objectID)}>Remove</button>
                                  </div>) : null}
                               </div>
                                                                                                                                                           this.state = {list}
                                                                                                                                                          Immutabilità dello stato
                       export default App;
                                                                                                                                        Considerazioni
                                                                                                                                                         this.setState()
this.setState()
                                                                                                                                                          onClick={() => this.removeItem()}
                                                   creare un bottone che permetta di aggiungere un nuovo elemento
                                                   definito come oggetto default
                                                     Verificare che ogni elemento della lista abbia un objectID
                                                    ogni elemento della lista può essere eliminato
                                Esercizio
```

verificare che ad ogni aggiunta, sia il totale degli elementi inseriti

che il totale delle spese varino di conseguenza

External Libs			