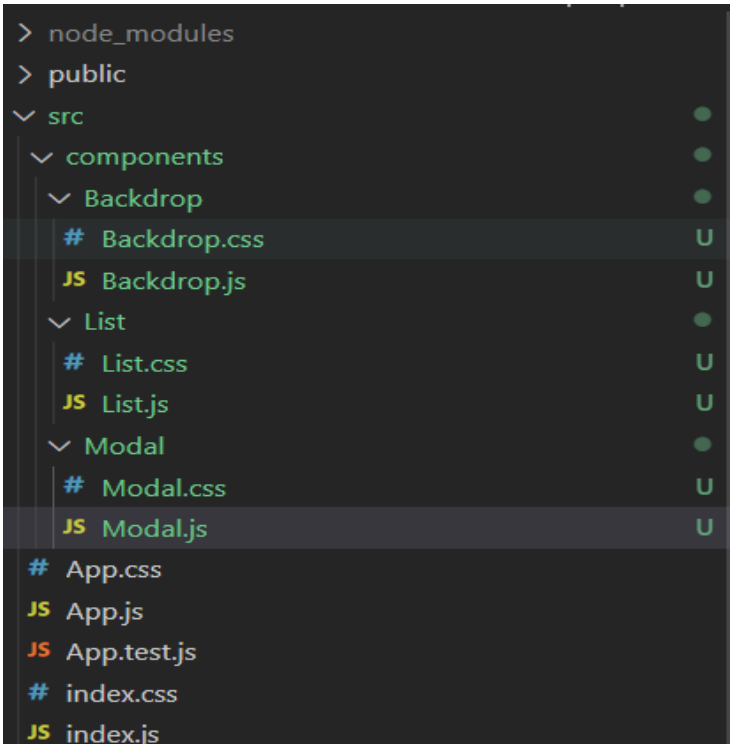
Contents

Folder Structure:



SOURCE CODE FOR SHOWING AND HIDING MODAL:

App.js

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

import "./App.css";
import Modal from "./components/Modal/Modal";
import Backdrop from "./components/Backdrop/Backdrop";
import List from "./components/List/List";

class App extends Component {
//6. Step to handle open and close
  state = {
    modalIsOpen: false
  }

  showModal = () => {
    this.setState({modalIsOpen: true});
  }

  closeModal = () => {
    this.setState({modalIsOpen: false});
  }
```

```
  render() {
    return (
      <div className="App">
        <h1>React Animations</h1>
        <Modal show={this.state.modalIsOpen} closed={this.closeModal}/>//7.
Sending show and closed as props
        <Backdrop show={this.state.modalIsOpen} /> // 8.
        <button className="Button" onClick={this.showModal}>Open Modal</button>
        <h3>Animating Lists</h3>
        <List />
      </div>
    );
  }
}

export default App;
```

Modal.js

```
import React from "react";

import "./Modal.css";

const modal = props => {
  const cssClasses = [
    "Modal",
    props.show ? "ModalOpen" : "ModalClosed"
  ];//4. step

  return (
    <div className={cssClasses.join(' ')}> //5. step
      <h1>A Modal</h1>
      <button className="Button" onClick={props.closed}>
        Dismiss
      </button>
    </div>
  );
};

export default modal;
```

Modal.css

```
.Modal {
    position: fixed;
    z-index: 200;
    border: 1px solid #eee;
    box-shadow: 0 2px 2px #ccc;
    background-color: white;
    padding: 10px;
    text-align: center;
    box-sizing: border-box;
    top: 30%;
    left: 25%;
    width: 50%;
}
// 1. FOR MODAL CLOSE AND OPEN
.ModalOpen {
    display: block;
}

.ModalClosed {
    display: none;
}
```

List.js

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

import './List.css';

class List extends Component {
    state = {
        items: [1, 2, 3]
    }

    addItemHandler = () => {
        this.setState((prevState) => {
            return {
                items: prevState.items.concat(prevState.items.length + 1)
            };
        });
    }

    removeItemHandler = (selIndex) => {
        this.setState((prevState) => {
            return {
```

```
                items: prevState.items.filter((item, index) => index !== selIndex
)
            };
        });
    }

    render () {
        const listItems = this.state.items.map( (item, index) => (
            <li
                key={index}
                className="ListItem"
                onClick={() => this.removeItemHandler(index)}>{item}</li>
        ) );

        return (
            <div>
                <button className="Button" onClick={this.addItemHandler}>Add Item
</button>
                <p>Click Item to Remove.</p>
                <ul className="List">
                    {listItems}
                </ul>
            </div>
        );
    }
}

export default List;
```

List.css

```
.List {
    list-style: none;
    margin: 0 auto;
    padding: 0;
    width: 280px;
}

.ListItem {
    margin: 0;
    padding: 10px;
    box-sizing: border-box;
    width: 100%;
    border: 1px solid #521751;
    background-color: white;
```

```
    text-align: center;
    cursor: pointer;
}

.ListItem:hover,
.ListItem:active {
    background-color: #ccc;
}
```

Backdrop.js

```
import React from 'react';

import './Backdrop.css';

const backdrop = (props) => {
    const cssClasses = ['Backdrop', props.show ? 'BackdropOpen' : 'BackdropClosed
']; //3. Open and closing on basis of props passed

    return <div className={cssClasses.join(' ')}></div>;//3. Joining css
};

export default backdrop;
```

Backdrop.css

```
.Backdrop {
    position: fixed;
    z-index: 100;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0,0,0,0.8);
}
//2. FOR OPEN AND CLOSE

.BackdropOpen {
    display: block;
}

.BackdropClosed {
    display: none;
}
```

USING CSS TRANSITIONS

NOTE: display property in css prevents your transition or animations.

Modal.css

```
.Modal {
    position: fixed;
    z-index: 200;
    border: 1px solid #eee;
    box-shadow: 0 2px 2px #ccc;
    background-color: white;
    padding: 10px;
    text-align: center;
    box-sizing: border-box;
    top: 30%;
    left: 25%;
    width: 50%;
    transition: all 0.3s ease-out;
    /* ease-out : how to display */
}

.ModalOpen {
    /* display: block; */
    opacity: 1;
    transform: translateY(0);
}

.ModalClosed {
    /* display: none; */
    opacity: 0;
    transform: translateY(-100%);
}
```

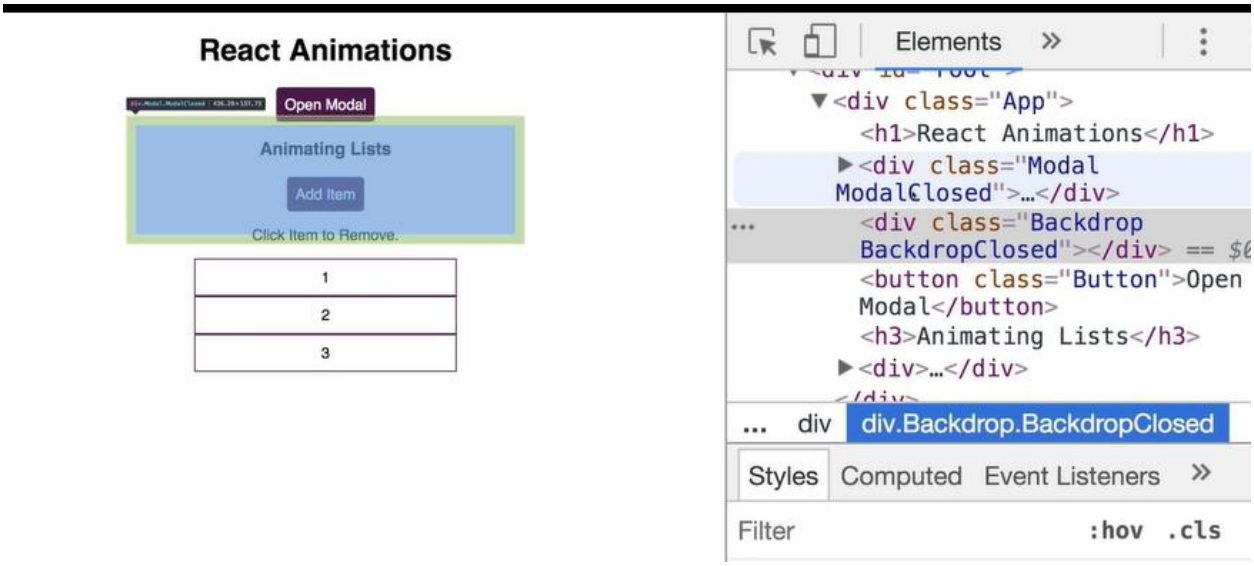
USING CSS ANIMATIONS

Modal.css

```
.Modal {
    position: fixed;
    z-index: 200;
    border: 1px solid #eee;
    box-shadow: 0 2px 2px #ccc;
    background-color: white;
    padding: 10px;
    text-align: center;
    box-sizing: border-box;
    top: 30%;
    left: 25%;
    width: 50%;
    transition: all 0.3s ease-out;
}

.ModalOpen {
    display: block;
    animation: openModal 0.3s ease-out forwards;
}

.ModalClosed {
    display: none;
    animation: closeModal 4s ease-out forwards;

}
@keyframes openModal {
    0% {
        opacity: 0;
        transform: translateY(-100%);
    }
    50% {
        opacity: 1;
        transform: translateY(90%);
    }
    100% {
        opacity: 1;
        transform: translateY(0);
    }
}

@keyframes closeModal {
    0% {
```

```
        opacity: 1;
        transform: translateY(0);
    }
    50% {
        opacity: 0.8;
        transform: translateY(60%);
    }
    100% {
        opacity: 0;
        transform: translateY(-100%);
    }
}
```

CSS TRANSTIONS AND ANIMATIONS LIMITATIONS



ModalClosed and ModalOpened are always in ou DOM which is not best.

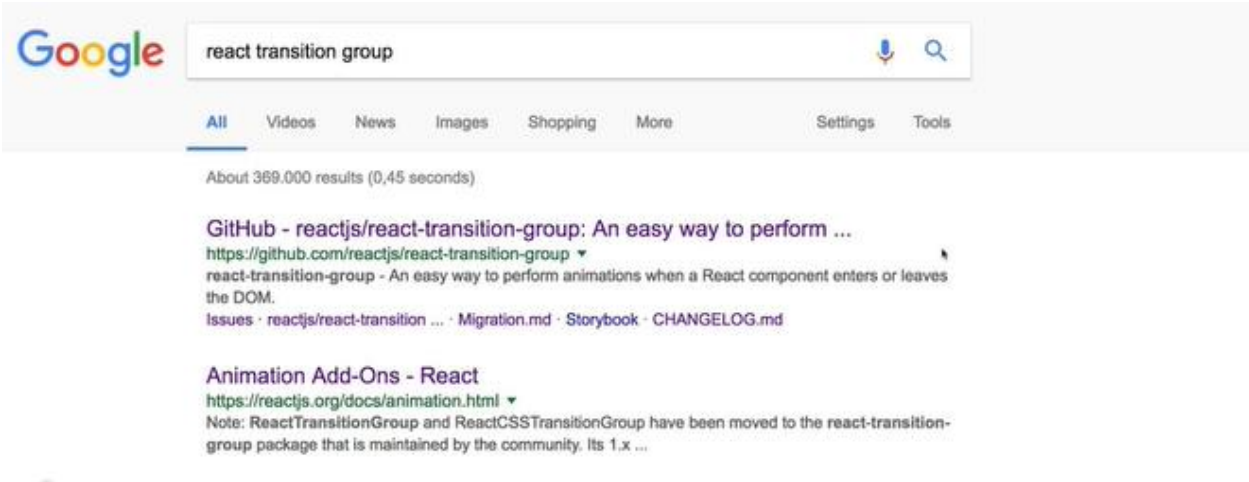
App.js

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

import "./App.css";
import Modal from "./components/Modal/Modal";
import Backdrop from "./components/Backdrop/Backdrop";
import List from "./components/List/List";

class App extends Component {
  state = {
    modalIsOpen: false
  }
```

```
  showModal = () => {
    this.setState({modalIsOpen: true});
  }

  closeModal = () => {
    this.setState({modalIsOpen: false});
  }

  render() {
    return (
      <div className="App">
        <h1>React Animations</h1>
        {this.state.modalIsOpen ? <Modal show={this.state.modalIsOpen} closed={th
is.closeModal}/> : null}
        {this.state.modalIsOpen ? <Backdrop show={this.state.modalIsOpen} /> : nu
ll}
        <button className="Button" onClick={this.showModal}>Open Modal</button>
        <h3>Animating Lists</h3>
        <List />
      </div>
    );
  }
}

export default App;
```

It will remove Modal when it is closed in DOM.

USING ReactTransitionGroup

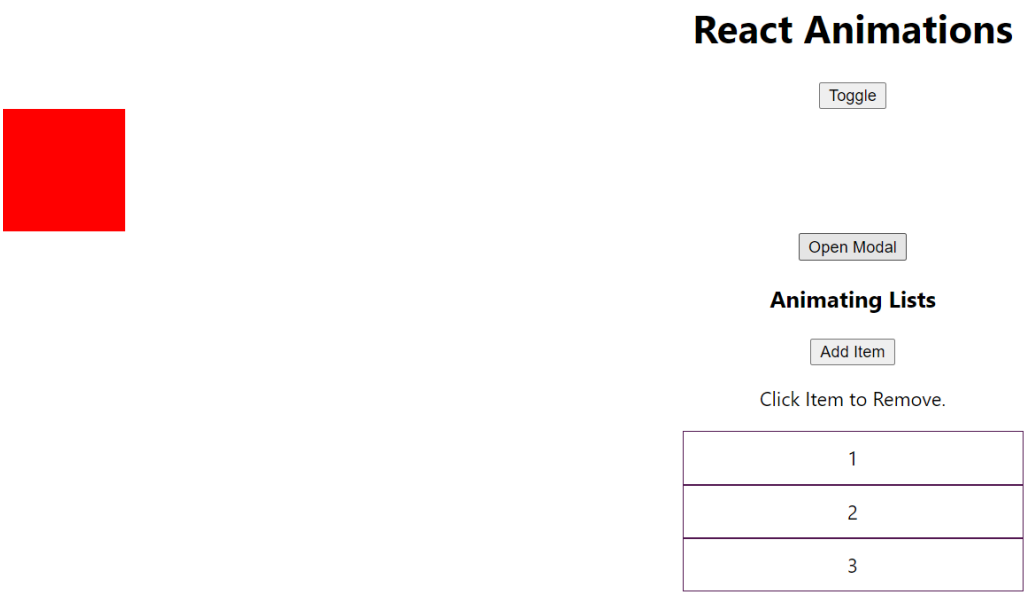


INSTALL: npm install react-transition-group –save

App.js

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

import "./App.css";
import Modal from "./components/Modal/Modal";
import Backdrop from "./components/Backdrop/Backdrop";
import List from "./components/List/List";

class App extends Component {
  state = {
    modalIsOpen: false,
    showBlock: false
  }

  showModal = () => {
    this.setState({modalIsOpen: true});
  }

  closeModal = () => {
    this.setState({modalIsOpen: false});
  }

  render() {
    return (
      <div className="App">
        <h1>React Animations</h1>
```

```
        <button onClick={() => this.setState(prevState =>({showBlock: !prevState.
showBlock}))}>Toggle</button>
        {
          this.state.showBlock ?
          <div style={{
            backgroundColor: 'red',
            width:100,
            height:100
          }}>
          </div> : null
        }
        {this.state.modalIsOpen ? <Modal show={this.state.modalIsOpen} closed={th
is.closeModal}/> : null}
        {this.state.modalIsOpen ? <Backdrop show={this.state.modalIsOpen} /> : nu
ll}
        <button className="Button" onClick={this.showModal}>Open Modal</button>
        <h3>Animating Lists</h3>
        <List />
      </div>
    );
  }
}

export default App;
```



```
{
          this.state.showBlock ?
          <div style={{
            backgroundColor: 'red',
            width:100,
            height:100,
            margin:'auto'
          }}>
```

Margin auto to place in center.

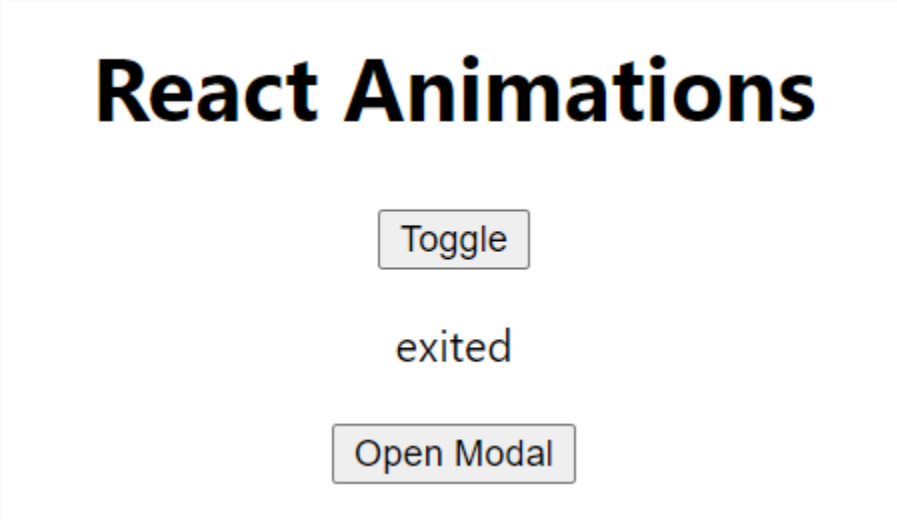
react-transition-group

App.js

```
import React, { Component } from "react";
import { Transition } from 'react-transition-group';

import "./App.css";
import Modal from "./components/Modal/Modal";
import Backdrop from "./components/Backdrop/Backdrop";
import List from "./components/List/List";

class App extends Component {
  state = {
    modalIsOpen: false,
    showBlock: false
  }

  showModal = () => {
    this.setState({modalIsOpen: true});
  }

  closeModal = () => {
    this.setState({modalIsOpen: false});
  }

  render() {
    return (
      <div className="App">
        <h1>React Animations</h1>
        <button
        onClick={() => this.setState(prevState =>({showBlock: !prevState.showBloc
k}))}>
        Toggle
        </button>
```

```
        <Transition
          in={this.state.showBlock}
          timeout={300}>
          {state => <p>{state}</p>}
        </Transition>

        {this.state.modalIsOpen ? <Modal show={this.state.modalIsOpen} closed={th
is.closeModal}/> : null}
        {this.state.modalIsOpen ? <Backdrop show={this.state.modalIsOpen} /> : nu
ll}
        <button className="Button" onClick={this.showModal}>Open Modal</button>
        <h3>Animating Lists</h3>
        <List />
      </div>
    );
  }
}

export default App;
```

CONSOLE: CLICK ON TOGGLE BUTTON



App.js [animation and transition, removing elements in DOM]

```
<Transition
        in={this.state.showBlock}
        timeout={300}
        mountOnEnter
        unmountOnExit
        >
        {state =>(
          <div style={{
            backgroundColor: 'red',
            width:100,
            height:100,
            margin:'auto',
            transition: 'opacity 1s ease-out',
            opacity: state === 'exiting' ? 0: 1
          }} />
          )
         }
          </Transition>
```

USING THE TRANSITION COMPONENT

App.js

```
        {/* FOR MODAL  */}
        <Transition
          in={this.state.modalIsOpen}
          timeout={300}
        >
          {state => (
            //rendering modal
            <Modal show={state} closed={this.closeModal}/>
          )}
        </Transition>
```

Modal.js

```
const modal = props => {
  const cssClasses = [
    "Modal",
    props.show  === 'entering'
    ? "ModalOpen"
    : props.show === 'exiting' ? "ModalClosed": null

  ];
```

WRAPPING THE TRANSITION COMPONENT [Nested loop]

CUT AND PASTE <Transition> …</Transition> IN MODAL.JS

App.js

```
}
          </Transition>

          <Modal show={this.state.modalIsOpen} closed={this.closeModal}/>

        {this.state.modalIsOpen ? <Backdrop show={this.state.modalIsOpen} /> : nu
ll}
```

Modal.js

```
import React from "react";

import "./Modal.css";
import Transition from 'react-transition-group/Transition';

const modal = props => {


  return (
    <Transition
      mountOnEnter
      unmountOnExit
      in={props.show}
      timeout={300}
    >

    {state => {
      const cssClasses = [
        "Modal",
```

```
        state === "entering"
        ? "ModalOpen"
        : state === "exiting" ? "ModalClosed" : null
      ];

      return (
        <div className={cssClasses.join(' ')}>
        <h1>A Modal</h1>
        <button className="Button" onClick={props.closed}>
          Dismiss
        </button>
        </div>
      );
    }}



    </Transition>

  );
};

export default modal;
```

ANIMATIONS TIMING

Modal.js

```
import React from "react";

import "./Modal.css";
import Transition from 'react-transition-group/Transition';

const animationTIming= {
  enter: 400,
  exit: 1000
}
// millisecond
const modal = props => {


  return (
    <Transition
      mountOnEnter
```

```
      unmountOnExit
      in={props.show}
      timeout={animationTIming}
    >

    {state => {
      const cssClasses = [
        "Modal",
        state === "entering"
        ? "ModalOpen"
        : state === "exiting" ? "ModalClosed" : null
      ];

      return (
        <div className={cssClasses.join(' ')}>
        <h1>A Modal</h1>
        <button className="Button" onClick={props.closed}>
          Dismiss
        </button>
        </div>
      );
    }}



    </Transition>

  );
};

export default modal;
```

Modal.css

```
.ModalClosed {
    display: none;
    animation: closeModal 1s ease-out forwards;

}
```

TRANSITIONS EVENTS

App.js

```
<Transition
        in={this.state.showBlock}
        timeout={300}
        mountOnEnter
        unmountOnExit
        onEnter= { () => console.log("onENter")}
        onEntering={ () => console.log("onENtering")}
        onEntered={ () => console.log("onENtered")}
        onExit= { () => console.log("onExit")}
        onExiting= { () => console.log("onExiting")}
        onExited = { () => console.log("onExited")}
        >
```

THE CSSTRANSITION COMPONENT

Modal.js

```
import React from "react";

import "./Modal.css";
import CSSTransition from 'react-transition-group/CSSTransition';

const animationTIming= {
  enter: 400,
  exit: 1000
}
// millisecond
const modal = props => {


  return (
    <CSSTransition
      mountOnEnter
      unmountOnExit
      in={props.show}
      timeout={animationTIming}
      classNames="fade-slide"
    >
    {/* fade-slide-enter
    fade-slide-exit-active */}

    {state => {
```

```
      const cssClasses = [
        "Modal",
        state === "entering"
        ? "ModalOpen"
        : state === "exiting" ? "ModalClosed" : null
      ];

      return (
        <div className="Modal">
        <h1>A Modal</h1>
        <button className="Button" onClick={props.closed}>
          Dismiss
        </button>
        </div>
      );
    }}



    </CSSTransition>

  );
};

export default modal;
```

modal.css

```
@keyframes openModal {
    0% {
        opacity: 0;
        transform: translateY(-100%);
    }
    50% {
        opacity: 1;
        transform: translateY(90%);
    }
    100% {
        opacity: 1;
        transform: translateY(0);
    }
}

.fade-slide-enter {
```

```
}

.fade-slide-enter-active {
    animation: openModal 0.3s ease-out forwards;
}

.fade-slide-exit {

}

.fade-slide-exit-active {
    animation: closeModal 1s ease-out forwards;
}

@keyframes closeModal {
    0% {
        opacity: 1;
        transform: translateY(0);
    }
    50% {
        opacity: 0.8;
        transform: translateY(60%);
    }
    100% {
        opacity: 0;
        transform: translateY(-100%);
    }
}
```

CUSTOMIZING CSS CLASSNAMES

Modal.js

Using our own css class in CSSTransiiton

```
<CSSTransition
      mountOnEnter
      unmountOnExit
      in={props.show}
      timeout={animationTIming}
      classNames={{
        enter: '',
        enterActive: 'ModalOpen',
        exit: '',
        exitActive: 'ModalCLose'
        // appear:
        // appearActive:
      }}
    >
```

ANIMATING LISTS

List.js

```
import React, { Component } from 'react';
import TransitionGroup from 'react-transition-group/TransitionGroup';
import { CSSTransition } from 'react-transition-group';

import './List.css';

class List extends Component {
    state = {
        items: [1, 2, 3]
    }

    addItemHandler = () => {
        this.setState((prevState) => {
            return {
                items: prevState.items.concat(prevState.items.length + 1)
            };
        });
    }

    removeItemHandler = (selIndex) => {
        this.setState((prevState) => {
```

```
            return {
                items: prevState.items.filter((item, index) => index !== selIndex
)
            };
        });
    }

    render () {
        const listItems = this.state.items.map( (item, index) => (
            <CSSTransition
            key={index}
            classNames="fade"
            timeout={300}
            >
           <li
                className="ListItem"
                onClick={() => this.removeItemHandler(index)}>{item}</li>
            </CSSTransition>


        ) );

        return (
            <div>
                <button className="Button" onClick={this.addItemHandler}>Add Item
</button>
                <p>Click Item to Remove.</p>
                <TransitionGroup
                 component="ul"
                 className="List"
                 >
                    {listItems}
                </TransitionGroup>
            </div>
        );
    }
}

export default List;
```

List.css

```
.fade-enter{
    opacity: 0;
}

.fade-enter-active{
    opacity: 1;
    transition: opacity 0.3s ease-out;
}

.fade-exit{
    opacity: 1;
}

.fade-exit-active{
    opacity: 0;
    transition: opacity 0.3s ease-out;
}
```

ALTENATIVE ANIMATIONS PACKAGES

- React-Motion Animation
- React Move
- React router transition