Jsx limitations:

**Adding div:**

   <div>

      {error && (

        <ErrorModal

          title={error.title}

          message={error.message}

          onConfirm={errorHandler}

        />

      )}

      <Card className={classes.input}>

        <form onSubmit={addUserHandler}>

          <label htmlFor="username">Username</label>

          <input

            id="username"

            type="text"

            value={enteredUsername}

            onChange={usernameChangeHandler}

          />

          <label htmlFor="age">Age (Years)</label>

          <input

            id="age"

            type="number"

            value={enteredAge}

            onChange={ageChangeHandler}

          />

          <Button type="submit">Add User</Button>

        </form>

      </Card>

     </div>

**Using array Method:**

  return (

   [

      error && (

        <ErrorModal key="error-Module"

          title={error.title}

          message={error.message}

          onConfirm={errorHandler}

        />

      ),

      <Card className={classes.input} key="add-user-card">

        <form onSubmit={addUserHandler}>

          <label htmlFor="username">Username</label>

          <input

            id="username"

            type="text"

            value={enteredUsername}

            onChange={usernameChangeHandler}

          />

          <label htmlFor="age">Age (Years)</label>

          <input

            id="age"

            type="number"

            value={enteredAge}

            onChange={ageChangeHandler}

          />

          <Button type="submit">Add User</Button>

        </form>

      </Card>

      ]

  );

Using div contain problem **<div> soup**

Unwanted div in symentic at end. Affect nested css

**Wrapper:**

    <Wrapper>

      {error && (

        <ErrorModal

          title={error.title}

          message={error.message}

          onConfirm={errorHandler}

        />

      )}

      <Card className={classes.input}>

        <form onSubmit={addUserHandler}>

          <label htmlFor="username">Username</label>

          <input

            id="username"

            type="text"

            value={enteredUsername}

            onChange={usernameChangeHandler}

          />

          <label htmlFor="age">Age (Years)</label>

          <input

            id="age"

            type="number"

            value={enteredAge}

            onChange={ageChangeHandler}

          />

          <Button type="submit">Add User</Button>

        </form>

      </Card>

    </Wrapper>

Wrapper.js:

function Wrapper() {

  return (

    props.children

  );

}

export default Wrapper;

**Fragments:**

  return (

    <>

      <AddUser onAddUser={addUserHandler} />

      <UsersList users={usersList} />

    </>

  );

  return (

    <React.Fragment>

      <AddUser onAddUser={addUserHandler} />

      <UsersList users={usersList} />

    </ React.Fragment >

  );

**Import { Fragment } from ‘react’;**

  return (

    < Fragment>

      <AddUser onAddUser={addUserHandler} />

      <UsersList users={usersList} />

    </Fragment >

  );

**React portals:**

Modals should be outside the scope of <section> so we use portals

import ReactDom from 'react-dom';

ReactDom.createPortal(<ModalOverlay props\_get={props}/>,document.getElementById("overlay-root"))}

import React from 'react';

import ReactDom from 'react-dom';

import Card from './Card';

import Button from './Button';

import classes from './ErrorModal.module.css';

const Backdrop = props =>{

  return <div className={classes.backdrop} onClick={props.onConfirm} />

}

const ModalOverlay = (props)=>{

  return (

    <Card className={classes.modal}>

        <header className={classes.header}>

          <h2>{props.props\_get.title}</h2>

        </header>

        <div className={classes.content}>

          <p>{props.props\_get.message}</p>

        </div>

        <footer className={classes.actions}>

          <Button onClick={props.props\_get.onConfirm}>Okay</Button>

        </footer>

      </Card>

  )

}

const ErrorModal = (props) => {

  return (

    <React.Fragment>

     {ReactDom.createPortal(<Backdrop onConfirm={props.onConfirm}/>,document.getElementById("backdrop-root"))}

     {ReactDom.createPortal(<ModalOverlay props\_get={props}/>,document.getElementById("overlay-root"))}

    </React.Fragment>

  );

};

export default ErrorModal;

index.html:

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="backdrop-root"></div>

    <div id="overlay-root"></div>

    <div id="root"></div>

    <!--

      This HTML file is a template.

      If you open it directly in the browser, you will see an empty page.

**Ref:**

import React, { useState ,useRef} from 'react';

const AddUser = (props) => {

  const nameInputref=useRef();

  const ageInputref=useRef();

  return (

    <React.Fragment>

        <form onSubmit={addUserHandler}>

          <label htmlFor="username">Username</label>

          <input

            id="username"

            type="text"

            ref={nameInputref}

          />

          <label htmlFor="age">Age (Years)</label>

          <input

            id="age"

            type="number"

            ref={ageInputref}

          />

    </React.Fragment>

  );

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

  const addUserHandler = (event) => {

    event.preventDefault();

    const enteredName= nameInputref.current.value

    const enteredAge = ageInputref.current.value

    props.onAddUser(enteredName, enteredAge);

    nameInputref.current.value=""

    ageInputref.current.value=""

    // setEnteredUsername('');

    // setEnteredAge('');

  };

**Controlled vs uncontrolled components:**