Contents

[1. Folder Structure 2](#_Toc76667675)

[2. Setting up a JS config for the form and rendering dynamically 7](#_Toc76667676)

[3. HANDLING USER INPUT 9](#_Toc76667677)

[4. Handling Form Submission 14](#_Toc76667678)

[5. Adding Custom Form Validation 15](#_Toc76667679)

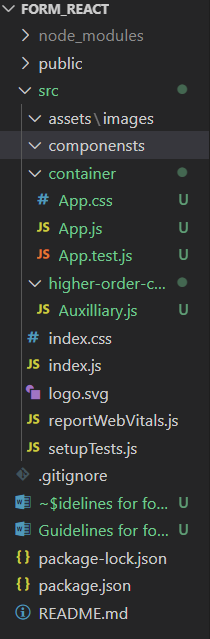
[6. FIXING A COMMON VALIDATION GOTCHA 17](#_Toc76667680)

[7. ADDING VALIDATION FEEDBACK 17](#_Toc76667681)

[8. IMPROVING VISUAL FEEDBACK 19](#_Toc76667682)

[9. HANDLING OVERALL FORM VALIDITY [DISABLE BUTTON] 21](#_Toc76667683)

# Folder Structure



Auxilliary.js

import React from 'react';

//empty wrapper

const Aux = props =>props.children;

export default Aux;

App.js

import './App.css';

import Aux from '../higher-order-container/Auxilliary';

import Contact from './Contact';

function App() {

  return (

    <Aux className="Contact">

        <Contact/>

    </Aux>

  );

}

export default App;

Contact.js

import React from 'react';

import Input from '../components/Input-form/Input';

import Button from '../components/Button/Button';

import './Contact.css';

const Contact = () => {

    return(

        <div className="Contact">

            <h4>ENTER CONTACT</h4>

            <form>

                <Input inputtype='input'

                type='email'

                name='email'

                placeholder="Your Email" />

                <Input inputtype='input'

                type='password'

                name='password'

                placeholder="Your Passwrod" />

                <Button

                btnType="Success"

                // clicked={}

                >LOG IN</Button>

            </form>

        </div>

    );

}

export default Contact;

Input.js

import React from 'react';

import './Input.css';

const Input = (props) => {

    let inputElement = null;

    switch(props.inputType){

        case('input'):

            inputElement = <input className="InputElement" {...props}/>;

            break;

        default:

            inputElement = <input className="InputElement" {...props}/>;

    }

    return(

        <div className="Input">

            <label className="label">{props.label}</label>

            {inputElement}

        </div>

    );

}

export default Input;

Button.js

import React from 'react';

import './Button.css';

const Button = (props) => (

    <button

    className={['Button',[props.btnType]].join(' ')}

    onClick={props.clicked}

    >{props.children}</button>

);

export default Button;

Contact.css

.Contact{

    box-sizing: border-box;

    margin: 10px;

    text-align: center;

}

@media (max-width:600px){

    .Contact{

        margin: 0;

        padding: 0;

        background-color: lawngreen;

    }

}

Input.css

.Input {

    width: 40%;

    padding: 10px;

    box-sizing: border-box;

    margin-left: 40%;

   }

.Label {

    font-weight: bold;

    display: block;

    margin-bottom: 8px;

   }

.InputElement {

    outline: none;

    border: 1px solid #ccc;

    background-color: white;

    font: inherit;

    padding: 6px 10px;

    display: block;

    width: 50%;

    box-sizing: border-box;

   }

.InputElement:focus{

    outline: none;

    background-color: #ccc;

}

@media (max-width:600px){

    .Input {

        width: 100%;

        padding: 0;

       }

}

Button.css

.Button {

    background-color: transparent;

    border:  none;

    color: #ccc;

    outline: none;

    cursor: pointer;

    font: inherit;

    padding: 10px;

    margin: 10px;

    font-weight: bolder;

}

.Button:first-of-type{

    margin-left: 0;

    padding-left: 0;

}

# Setting up a JS config for the form and rendering dynamically

Contact.js

const [state,setState] = useState({

        form:{

            email:{

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Email'

                },

                value: ''

            },

            password:{

                elementType: 'input',

                elementConfig: {

                    type: 'password',

                    placeholder: 'Password'

                },

                value: ''

            },

        }

    });

const formElementsArray = [];

    for(let key in state.form){

        formElementsArray.push({

            id:key,

            config: state.form[key]//yaha sabai depth state haru baschan

        })

    }

    return(

        <div className="Contact">

            <h4>ENTER CONTACT</h4>

            <form>

                 {formElementsArray.map(formElement => (

                     <Input

                     key={formElement.id}

                     elementType={formElement.config.elementType}

                     elementConfig={formElement.config.elementConfig}

                     value={formElement.config.value}

                     />

                 ))}

                <Button

                btnType="Success"

                // clicked={}

                >LOG IN</Button>

            </form>

        </div>

    );

Input.js

import React from 'react';

import './Input.css';

const Input = (props) => {

    let inputElement = null;

    switch(props.elementType){

        case('input'):

            inputElement = <input

            className="InputElement"

            {...props.elementConfig}

            value={props.value}

           />;

            break;

        default:

            inputElement = <input

            className="InputElement"

            {...props.elementConfig}

            value={props.value}

            />;

    }

    return(

        <div className="Input">

            <label className="label">{props.label}</label>

            {inputElement}

        </div>

    );

}

export default Input;

# HANDLING USER INPUT

Input.js

switch(props.elementType){

        case('input'):

            inputElement = <input

            className="InputElement"

            {...props.elementConfig}

            value={props.value}

            onChange={props.changed}

           />;

            break;

        default:

            inputElement = <input

            className="InputElement"

            {...props.elementConfig}

            value={props.value}

            onChange={props.changed}

            />;

    }

Contact.js

import React,{useState} from 'react';

import Input from '../components/Input-form/Input';

import Button from '../components/Button/Button';

import './Contact.css';

const Contact = () => {

    const [state,setState] = useState({

        form:{

            email:{

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Email'

                },

                value: ''

            },

            password:{

                elementType: 'input',

                elementConfig: {

                    type: 'password',

                    placeholder: 'Password'

                },

                value: ''

            },

        }

    });

    const orderHandler = (event) => {

        event.preventDefault();

        alert('Thank you. Your food will be delivery in 5 minutes.')

    }

    const formElementsArray = [];

    for(let key in state.form){

        formElementsArray.push({

            id:key,

            config: state.form[key]//yaha sabai depth state haru baschan

        })

    }

    const inputChangedHandler = (event) =>{

        console.log(event.target.value);

    }

    return(

        <div className="Contact">

            <h4>ENTER CONTACT</h4>

            <form>

                 {formElementsArray.map(formElement => (

                     <Input

                     key={formElement.id}

                     elementType={formElement.config.elementType}

                     elementConfig={formElement.config.elementConfig}

                     value={formElement.config.value}

                     changed={(event) =>{

                     inputChangedHandler(event,formElement.id)

                     console.log(formElement.id)

                     }

                     }

                     />

                 ))}

                <Button

                btnType="Success"

                clicked={orderHandler}

                >LOG IN</Button>

            </form>

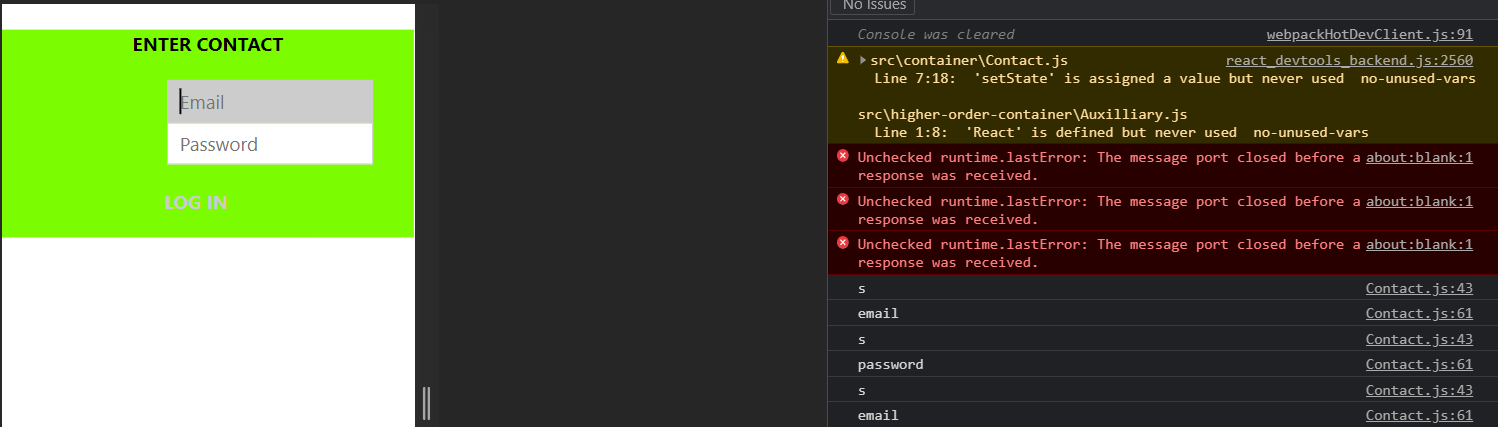
        </div>

    );

}

export default Contact;

SS:



MORE ADVANCED:

import React,{useState} from 'react';

import Input from '../components/Input-form/Input';

import Button from '../components/Button/Button';

import './Contact.css';

const Contact = () => {

    const [state,setState] = useState({

        form:{

            email:{

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Email'

                },

                value: ''

            },

            password:{

                elementType: 'input',

                elementConfig: {

                    type: 'password',

                    placeholder: 'Password'

                },

                value: ''

            },

        }

    });

    const orderHandler = (event) => {

        event.preventDefault();

        alert('Thank you. Your food will be delivery in 5 minutes.')

    }

    const formElementsArray = [];

    for(let key in state.form){

        formElementsArray.push({

            id:key,

            config: state.form[key]//yaha sabai depth state haru baschan

        })

    }

    const inputChangedHandler = (event,inputIdentifier) =>{

        // console.log(event.target.value);

        //cloning original form to new form

        const updatedForm = {

            ...state.form

        }

        //cloning into more deeply for inputidentifers like elementType,elementConfig,type and so on

        const updatedFormElement={

            ...updatedForm[inputIdentifier]

        }

        //user's input

        updatedFormElement.value = event.target.value;

        console.log('UpdatedFormElement=',updatedFormElement)

        updatedForm[inputIdentifier] = updatedFormElement;

        console.log('updatedForm input identifiers=',updatedForm[inputIdentifier]);

        setState({form:updatedForm});

    }

    return(

        <div className="Contact">

            <h4>ENTER CONTACT</h4>

            <form>

                 {formElementsArray.map(formElement => (

                     <Input

                     key={formElement.id}

                     elementType={formElement.config.elementType}

                     elementConfig={formElement.config.elementConfig}

                     value={formElement.config.value}

                     changed={(event) =>{

                     inputChangedHandler(event,formElement.id)

                    //  console.log('input id=',formElement.id)

                     }

                     }

                     />

                 ))}

                <Button

                btnType="Success"

                clicked={orderHandler}

                >LOG IN</Button>

            </form>

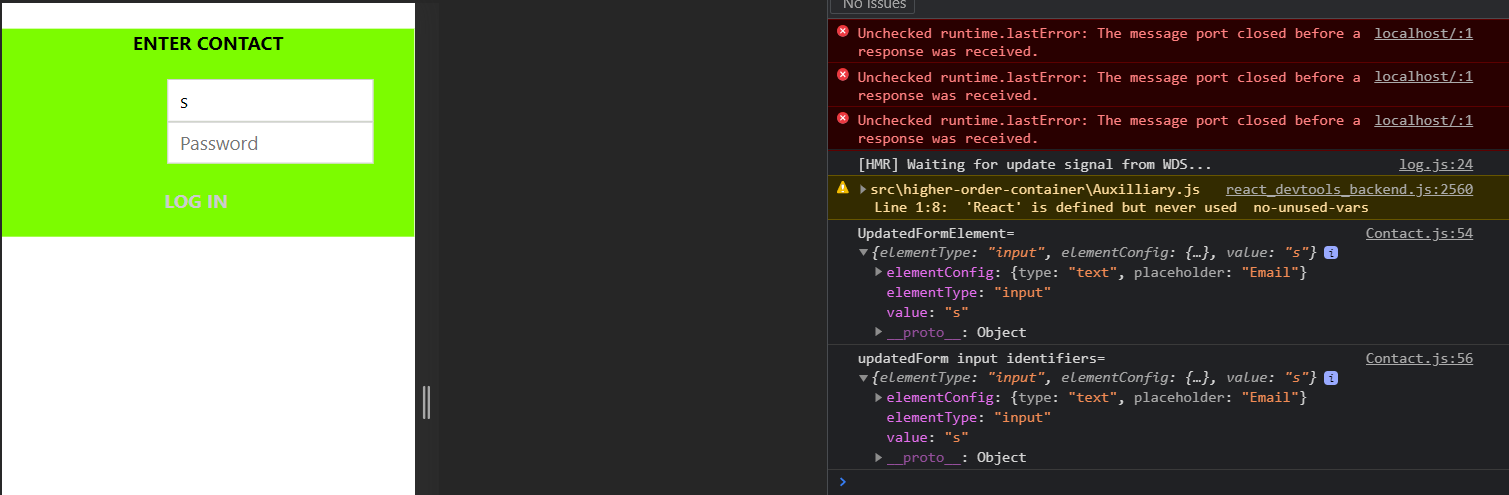
        </div>

    );

}

export default Contact;

SS:



# Handling Form Submission

Contact.js

const orderHandler = (event) => {

        event.preventDefault();

        const formData = {} //empty object

        console.log('Before, user form data=',formData)

        for(let formElementIdentifier in state.form){

            formData[formElementIdentifier] =

            state.form[formElementIdentifier].value;

        }

        console.log('After, user form data=',formData);

        const data ={

            userData: formData

        }

        console.log('user data:',data)

        alert('Thank you. Your food will be delivery in 5 minutes.')

    }

SS:



# Adding Custom Form Validation

Contact.js

const [state,setState] = useState({

        form:{

            email:{

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Email'

                },

                value: '',

                validation: {

                    required: true,

                    minLength: 6

                },

                valid: false

            },

            password:{

                elementType: 'input',

                elementConfig: {

                    type: 'password',

                    placeholder: 'Password'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false

            },

        }

    });

const checkValidity = (value,rules) => {

        let isValid =false;

        if(rules.required){

            isValid = valid.trim() !== '';

        }

        if(rules.minLength){

            isValid = value.length >= rules.minLength;

        }

        if(rules.maxLength){

            isValid = value.length >= rules.maxLength;

        }

        return isValid

    }

const inputChangedHandler = (event,inputIdentifier) =>{

        // console.log(event.target.value);

        //cloning original form to new form

        const updatedForm = {

            ...state.form

        }

        //cloning into more deeply for inputidentifers like elementType,elementConfig,type and so on

        const updatedFormElement={

            ...updatedForm[inputIdentifier]

        }

        //user's input

        updatedFormElement.value = event.target.value;

        updatedFormElement.valid = checkValidity(

            updatedFormElement.value,updatedFormElement.validation

        );

        // console.log('UpdatedFormElement=',updatedFormElement)

        updatedForm[inputIdentifier] = updatedFormElement;

        // console.log('updatedForm input identifiers=',updatedForm[inputIdentifier]);

        setState({form:updatedForm});

    }

# FIXING A COMMON VALIDATION GOTCHA

const checkValidity = (value,rules) => {

        let isValid = true;

        if(rules.required){

            isValid = value.trim() !== '' && isValid;

        }

        if(rules.minLength){

            isValid = value.length >= rules.minLength && isValid;

        }

        if(rules.maxLength){

            isValid = value.length >= rules.maxLength && isValid;

        }

        return isValid

    }

# ADDING VALIDATION FEEDBACK

Input.css

.Invalid{

    border:  1px solid red;

    background-color: salmon;

}

Input.js

import React from 'react';

import './Input.css';

const Input = (props) => {

    let inputElement = null;

    const inputClasses = ['InputElement']

    if(props.invalid && props.shouldValidate){

        inputClasses.push('Invalid');

    }

    switch(props.elementType){

        case('input'):

            inputElement = <input

            className={inputClasses.join(' ') }

            {...props.elementConfig}

            value={props.value}

            onChange={props.changed}

           />;

            break;

        default:

            inputElement = <input

            className={inputClasses.join(' ') }

            {...props.elementConfig}

            value={props.value}

            onChange={props.changed}

            />;

    }

Contact.js

<form>

                 {formElementsArray.map(formElement => (

                     <Input

                     key={formElement.id}

                     elementType={formElement.config.elementType}

                     elementConfig={formElement.config.elementConfig}

                     value={formElement.config.value}

                     invalid={!formElement.config.valid}

                     shouldValidate={formElement.config.validation}

                     changed={(event) =>{

                     inputChangedHandler(event,formElement.id)

                    //  console.log('input id=',formElement.id)

                     }

                     }

                     />

                 ))}

# IMPROVING VISUAL FEEDBACK

Contact.js

  email:{

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Email'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false,

                touched: false

            },

            password:{

                elementType: 'input',

                elementConfig: {

                    type: 'password',

                    placeholder: 'Password'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false,

                touched: false

            },

const inputChangedHandler = (event,inputIdentifier) =>{

        // console.log(event.target.value);

        //cloning original form to new form

        const updatedForm = {

            ...state.form

        }

        //cloning into more deeply for inputidentifers like elementType,elementConfig,type and so on

        const updatedFormElement={

            ...updatedForm[inputIdentifier]

        }

        //user's input

        updatedFormElement.value = event.target.value;

        updatedFormElement.valid = checkValidity(

            updatedFormElement.value,updatedFormElement.validation

        );

        updatedFormElement.touched = true;

        // console.log('UpdatedFormElement=',updatedFormElement)

        updatedForm[inputIdentifier] = updatedFormElement;

        // console.log('updatedForm input identifiers=',updatedForm[inputIdentifier]);

        setState({form:updatedForm});

    }

    return(

        <div className="Contact">

            <h4>ENTER CONTACT</h4>

            <form>

                 {formElementsArray.map(formElement => (

                     <Input

                     key={formElement.id}

                     elementType={formElement.config.elementType}

                     elementConfig={formElement.config.elementConfig}

                     value={formElement.config.value}

                     invalid={!formElement.config.valid}

                     shouldValidate={formElement.config.validation}

                     touched={formElement.config.touched}

                     changed={(event) =>{

                     inputChangedHandler(event,formElement.id)

                    //  console.log('input id=',formElement.id)

                     }

                     }

                     />

                 ))}

                <Button

                btnType="Success"

                clicked={orderHandler}

                >LOG IN</Button>

            </form>

            {/\* {state.form.email} \*/}

        </div>

# HANDLING OVERALL FORM VALIDITY [DISABLE BUTTON]

import React,{useState} from 'react';

import Input from '../components/Input-form/Input';

import Button from '../components/Button/Button';

import './Contact.css';

const Contact = () => {

    const [state,setState] = useState({

        form:{

            email:{

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Email'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false,

                touched: false

            },

            password:{

                elementType: 'input',

                elementConfig: {

                    type: 'password',

                    placeholder: 'Password'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false,

                touched: false

            },

        },

        formIsValid: false,

    });

    const orderHandler = (event) => {

        event.preventDefault();

        const formData = {} //empty object

        // console.log('Before, user form data=',formData)

        for(let formElementIdentifier in state.form){

            formData[formElementIdentifier] =

            state.form[formElementIdentifier].value;

        }

        // console.log('After, user form data=',formData);

        const data ={

            userData: formData

        }

        // console.log('user data:',data)

        alert('Thank you.')

    }

    const checkValidity = (value,rules) => {

        let isValid = true;

        if(rules.required){

            isValid = value.trim() !== '' && isValid;

        }

        if(rules.minLength){

            isValid = value.length >= rules.minLength && isValid;

        }

        if(rules.maxLength){

            isValid = value.length >= rules.maxLength && isValid;

        }

        return isValid

    }

    const formElementsArray = [];

    for(let key in state.form){

        formElementsArray.push({

            id:key,

            config: state.form[key]//yaha sabai depth state haru baschan

        })

    }

    const inputChangedHandler = (event,inputIdentifier) =>{

        // console.log(event.target.value);

        //cloning original form to new form

        const updatedForm = {

            ...state.form

        }

        //cloning into more deeply for inputidentifers like elementType,elementConfig,type and so on

        const updatedFormElement={

            ...updatedForm[inputIdentifier]

        }

        //user's input

        updatedFormElement.value = event.target.value;

        updatedFormElement.valid = checkValidity(

            updatedFormElement.value,updatedFormElement.validation

        );

        updatedFormElement.touched = true;

        // console.log('UpdatedFormElement=',updatedFormElement)

        updatedForm[inputIdentifier] = updatedFormElement;

        // console.log('updatedForm input identifiers=',updatedForm[inputIdentifier]);

        let formIsValid =true;

        for(let inputIdentifier in updatedForm){

            formIsValid = updatedForm[inputIdentifier].valid

            && formIsValid;

        }

        console.log(formIsValid);

        setState({form:updatedForm,formIsValid: formIsValid});

    }

    return(

           <Button

                btnType="Success"

                clicked={orderHandler}

                disabled={!state.formIsValid}

                >LOG IN

</Button>

        </div>

    );

}

export default Contact;