Contents

[Creating a Custom Dynamic Input Component 2](#_Toc76626883)

[Setting up a JS config for the form 6](#_Toc76626884)

[Dynamically Create Inputs based on JS Config 9](#_Toc76626885)

[ADDING DROPDOWN COMPONENTS: 12](#_Toc76626886)

[HANDLING USER INPUT 13](#_Toc76626887)

[Handling Form Submission 19](#_Toc76626888)

[Adding Custom Form Validation 19](#_Toc76626889)

[FIXING A COMMON VALIDATION GOTCHA 24](#_Toc76626890)

[ADDING VALIDATION FEEDBACK 24](#_Toc76626891)

[IMPROVING VISUAL FEEDBACK 30](#_Toc76626892)

[HANDLING OVERALL FORM VALIDITY [DISABLE BUTTON] 33](#_Toc76626893)

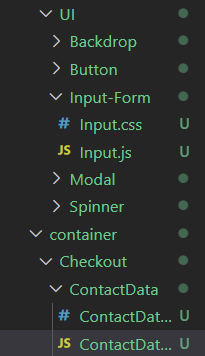
[WORKING ON AN ERROR OCCUR IN SELECTOR INPUT 39](#_Toc76626894)

[FIXING A BUG IN SELECTOR: IF WE CHANGE THE STATE, THEN IT WILL STORE NULL VALUE. 43](#_Toc76626895)

# 

# Creating a Custom Dynamic Input Component

Folder Structure:



Create Input.js and Input.css inside Input-Form folder.

Input.js

NOTE: switch receive what types of inputtype is passed as props from contactData.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;

        case ('textarea'):

            inputElement = <textarea 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;

FOR STYLE: Input.css

.Input {

    width: 100%;

    padding: 10px;

    box-sizing: border-box;

}

.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: 100%;

    box-sizing: border-box;

}

.InputElement:focus{

    outline: none;

    background-color: #ccc;

}

ContactData.js

NOTE: Dynamic Form which will helps to reduce the complication.

import React, { useState } from 'react';

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

import './ContactData.css'

import Input from '../../../components/UI/Input-Form/Input';

const ContactData = (props) => {

    const [state,setState] = useState({

        name: '',

        email: '',

        address: {

            street: '',

            postalCode:''

        }

    });

    const orderHandler = (event) => {

        event.preventDefault();

        console.log(props.ingredients)

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

    }

    return(

        <div className="ContactData">

            <h4>ENTER CONTACT</h4>

            <form>

                <Input inputtype='input'

                 type='text'

                 name="name"

                 placeholder="Your Name" />

                <Input inputtype='input'

                 type='email'

                 name="email"

                 placeholder="Your Email" />

                <Input inputtype='input'

                 type='text'

                 name="street"

                 placeholder="Your Street" />

                <Input inputtype='input'

                type='text'

                name="postal"

                placeholder="Your Postal COde" />

                <Button

                btnType="Success"

                clicked={orderHandler}>ORDER</Button>

            </form>

        </div>

    );

}

export default ContactData;

For Styling: ContactData.css

.ContactData {

    margin:  20px auto;

    width: 80%;

    text-align: center;

    box-shadow: 0 2px 3px #ccc;

    border: 1px solid #eee;

    padding: 10px;

    box-sizing: border-box;

}

@media (min-width: 600px){

    .ContactData {

        width: 500px;

    }

}

# Setting up a JS config for the form

ContactData.js

import React, { useState } from 'react';

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

import './ContactData.css'

import Input from '../../../components/UI/Input-Form/Input';

const ContactData = (props) => {

    const [state,setState] = useState({

        name: '',

        email: '',

        address: {

            street: '',

            postalCode:''

        }

    });

    const [formState,setFormState] = useState({

        orderForm:{

            name: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Your Name'

                },

                value: ''

                },

            street: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Street'

                },

                value: ''

                },

            zipCode: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Zip Code'

                },

                value: ''

                },

            country: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Country'

                },

                value: ''

                },

            email: {

                elementType: 'input',

                elementConfig: {

                    type: 'emai',

                    placeholder: 'Your Email'

                },

                value: ''

                },

            deliveryMethod: {

                elementType: 'select',

                elementConfig: {

                    options: [

                        {value: 'fastest',displayValue: 'fastest'},

                        {value: 'cheapest',displayValue: 'cheapest'},

                        ]

                },

                value: ''

                },

        }

    })

    const orderHandler = (event) => {

        event.preventDefault();

        console.log(props.ingredients)

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

    }

    return(

        <div className="ContactData">

            <h4>ENTER CONTACT</h4>

            <form>

                <Input

                    elementType='...'

                    elementConfig='...'

                    value='...'

                />

                <Input inputtype='input'

                 type='email'

                 name="email"

                 placeholder="Your Email" />

                <Input inputtype='input'

                 type='text'

                 name="street"

                 placeholder="Your Street" />

                <Input inputtype='input'

                type='text'

                name="postal"

                placeholder="Your Postal COde" />

                <Button

                btnType="Success"

                clicked={orderHandler}>ORDER</Button>

            </form>

        </div>

    );

}

export default ContactData;

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;

        case ('textarea'):

            inputElement = <textarea

            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;

# Dynamically Create Inputs based on JS Config

ContactData.js

import React, { useState } from 'react';

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

import './ContactData.css'

import Input from '../../../components/UI/Input-Form/Input';

const ContactData = (props) => {

    const [state,setState] = useState({

        name: '',

        email: '',

        address: {

            street: '',

            postalCode:''

        }

    });

    const [formState,setFormState] = useState({

        orderForm:{

            name: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Your Name'

                },

                value: ''

                },

            street: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Street'

                },

                value: ''

                },

            zipCode: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Zip Code'

                },

                value: ''

                },

            country: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Country'

                },

                value: ''

                },

            email: {

                elementType: 'input',

                elementConfig: {

                    type: 'emai',

                    placeholder: 'Your Email'

                },

                value: ''

                },

            deliveryMethod: {

                elementType: 'select',

                elementConfig: {

                    options: [

                        {value: 'fastest',displayValue: 'fastest'},

                        {value: 'cheapest',displayValue: 'cheapest'},

                        ]

                },

                value: ''

                },

        }

    })

    const orderHandler = (event) => {

        event.preventDefault();

        console.log(props.ingredients)

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

    }

    const formElementsArray = [];

    for(let key in formState.orderForm){

        formElementsArray.push({

            id: key,

            config: formState.orderForm[key]

        })

    }

    return(

        <div className="ContactData">

            <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={orderHandler}>ORDER</Button>

            </form>

        </div>

    );

}

export default ContactData;

# ADDING DROPDOWN COMPONENTS:

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;

        case ('textarea'):

            inputElement = <textarea

            className='InputElement'

            {...props.elementConfig}

            value={props.value}/>;

            break;

        case ('select'):

            inputElement = (

            <select

                className='InputElement'

                value={props.value}>

                {props.elementConfig.options.map(option => (

                    <option

                    key={option.value}

                    value={option.value}>

                        {option.displayValue}

                    </option>

                ))}

            </select>

            );

            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

NOTE: onChange handler will receive changed props from ContactData.js to change the value of input field.

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} onChange={props.changed} />;

            break;

        case ('textarea'):

            inputElement = <textarea

            className='InputElement'

            {...props.elementConfig}

            value={props.value} onChange={props.changed}/>;

            break;

        case ('select'):

            inputElement = (

            <select

                className='InputElement'

                value={props.value} onChange={props.changed}>

                {props.elementConfig.options.map(option => (

                    <option

                    key={option.value}

                    value={option.value}>

                        {option.displayValue}

                    </option>

                ))}

            </select>

            );

            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;

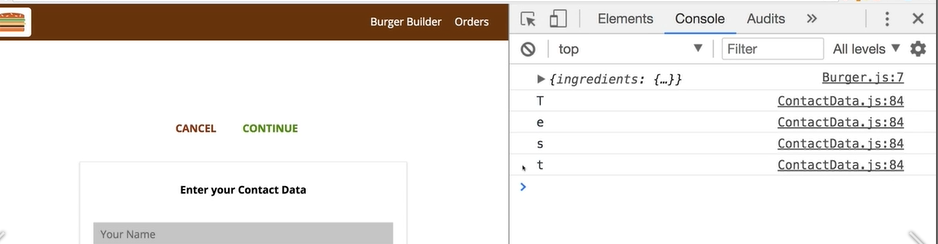
ContactData.js

NOTE:

Changed prop => pass the method named inputChangedHandler

inputChangedHandler=> attached with event object.

Console.log(event.target.value) :



This T e s t is new value given by user. So, we need to update the state to show the change in input field. Hamilai kun input field click garim vanera inputIdentifier le dincha.

inputChangedHandler(event,formElement.id)

formElement.id => name, street, zip code which came from map keys.

for(let key in formState.orderForm){

        formElementsArray.push({

            id: key,

Now, we need to update setFormState.

const inputChangedHandler = (event,inputIdentifier) => {

        console.log(event.target.value);

        //creating javascript object to copying old orderform state to update, but it does not duplicate all of them, nested properties are not cloned.

        const updatedOrderForm = {

            ...formState.orderForm

        }

//we need to clone more deeply. These

//elementType: 'input',

//elementConfig: {

//type: 'text',

//placeholder: 'Your Name'},

//value: ''

//updatedOrderForm = cloned old form with input identifier i.e email, name and storing it inside updatedFormElements. {…} means creating new object which means updatedFormElement creates updatedOrderForm[inputIdentifier] object

        const updatedFormElement= {

            ...updatedOrderForm[inputIdentifier]

        };

//updatedFormElement ma avo name ko value, email ko value equlas to hamile haleko value jun event le dincha tyo store garim.

        updatedFormElement.value = event.target.value;

//updatedFormElement wala form lai updatedOrderForm ko input identifiers are halim

        updatedOrderForm[inputIdentifier] = updatedFormElement;

//Update order Form state

        setFormState({orderForm: updatedOrderForm})

    }

import React, { useState } from 'react';

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

import './ContactData.css'

import Input from '../../../components/UI/Input-Form/Input';

const ContactData = (props) => {

    const [state,setState] = useState({

        name: '',

        email: '',

        address: {

            street: '',

            postalCode:''

        }

    });

    const [formState,setFormState] = useState({

        orderForm:{

            name: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Your Name'

                },

                value: ''

                },

            street: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Street'

                },

                value: ''

                },

            zipCode: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Zip Code'

                },

                value: ''

                },

            country: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Country'

                },

                value: ''

                },

            email: {

                elementType: 'input',

                elementConfig: {

                    type: 'emai',

                    placeholder: 'Your Email'

                },

                value: ''

                },

            deliveryMethod: {

                elementType: 'select',

                elementConfig: {

                    options: [

                        {value: 'fastest',displayValue: 'fastest'},

                        {value: 'cheapest',displayValue: 'cheapest'},

                        ]

                },

                value: ''

                },

        }

    })

    const orderHandler = (event) => {

        event.preventDefault();

        console.log(props.ingredients)

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

    }

    const formElementsArray = [];

    for(let key in formState.orderForm){

        formElementsArray.push({

            id: key,

            config: formState.orderForm[key]

        })

    }

    const inputChangedHandler = (event,inputIdentifier) => {

        console.log(event.target.value);

        //copying old form to update

        const updatedOrderForm = {

            ...formState.orderForm

        }

        const updatedFormElement= {

            ...updatedOrderForm[inputIdentifier]

        };

        updatedFormElement.value = event.target.value;

        updatedOrderForm[inputIdentifier] = updatedFormElement;

        setFormState({orderForm: updatedOrderForm})

    }

    return(

        <div className="ContactData">

            <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)}

                     />

                ))}

                <Button

                btnType="Success"

                clicked={orderHandler}>ORDER</Button>

            </form>

        </div>

    );

}

export default ContactData;

# Handling Form Submission

ContactData.js

 const orderHandler = (event) => {

        event.preventDefault();

        // console.log(props.ingredients)

        const formData = {};//empty object

        for ( let formElementIdentifier in formState.orderForm){

            formData[formElementIdentifier] =

formState.orderForm[formElementIdentifier].value;

        }

        const order= {

            // ingredients: props.ingredients,

            // price: props.price,

            orderData: formData

        }

        alert('see on console');

        console.log(order);

        // console.log(ingredients);

        // console.log(price);

    }

# Adding Custom Form Validation

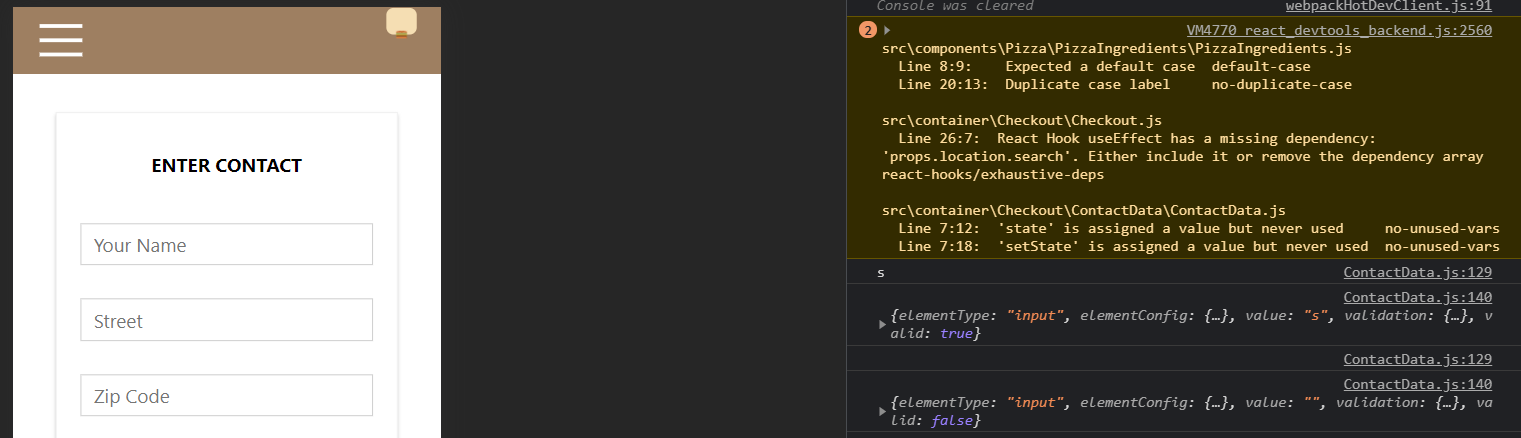
,

                validation: {

                    required: true

                },

valid: false



NOTE: THERE WILL FLAW, WE WILL RESOLVE IN NEXT PART

ContactData.js

import React, { useState } from 'react';

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

import './ContactData.css'

import Input from '../../../components/UI/Input-Form/Input';

const ContactData = (props) => {

    const [state,setState] = useState({

        name: '',

        email: '',

        address: {

            street: '',

            postalCode:''

        }

    });

    const [formState,setFormState] = useState({

        orderForm:{

            name: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Your Name'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false

                },

            street: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Street'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false

                },

            zipCode: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Zip Code'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false

                },

            country: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Country'

                },

                value: '',

                validation: {

                    required: true,

                    minLength: 6,

                    maxLength: 6

                },

                valid: false

                },

            email: {

                elementType: 'input',

                elementConfig: {

                    type: 'emai',

                    placeholder: 'Your Email'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false

                },

            deliveryMethod: {

                elementType: 'select',

                elementConfig: {

                    options: [

                        {value: 'fastest',displayValue: 'fastest'},

                        {value: 'cheapest',displayValue: 'cheapest'},

                        ]

                },

                value: ''

                },

        }

    })

    const orderHandler = (event) => {

        event.preventDefault();

        // console.log(props.ingredients)

        const formData = {};//empty object

        for ( let formElementIdentifier in formState.orderForm){

            formData[formElementIdentifier] = formState.orderForm[formElementIdentifier].value;

        }

        const order= {

            // ingredients: props.ingredients,

            // price: props.price,

            orderData: formData

        }

        alert('see on console');

        console.log(order);

        // console.log(ingredients);

        // console.log(price);

    }

    const formElementsArray = [];

    for(let key in formState.orderForm){

        formElementsArray.push({

            id: key,

            config: formState.orderForm[key]

        })

    }

    const checkValidty = (value, rules) => {

        let isValid = false;

        if(rules.required){

            isValid = value.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);

        const updatedOrderForm = {

            ...formState.orderForm

        }

        const updatedFormElement= {

            ...updatedOrderForm[inputIdentifier]

        };

        updatedFormElement.value = event.target.value;

        updatedFormElement.valid = checkValidty(

updatedFormElement.value,updatedFormElement.validation);

        updatedOrderForm[inputIdentifier] = updatedFormElement;

        console.log(updatedFormElement)

        setFormState({orderForm: updatedOrderForm})

    }

    return(

        <div className="ContactData">

            <h4>ENTER CONTACT</h4>

            <form onSubmit={orderHandler}>

                {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)}

                     />

                ))}

                <Button

                btnType="Success"

                clicked={orderHandler}>ORDER</Button>

            </form>

        </div>

    );

}

export default ContactData;

# FIXING A COMMON VALIDATION GOTCHA

const checkValidty = (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.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;

        case ('textarea'):

            inputElement = <textarea

            className={inputClasses.join(' ')}

            {...props.elementConfig}

            value={props.value} onChange={props.changed}/>;

            break;

        case ('select'):

            inputElement = (

            <select

                className={inputClasses.join(' ')}

                value={props.value} onChange={props.changed}>

                {props.elementConfig.options.map(option => (

                    <option

                    key={option.value}

                    value={option.value}>

                        {option.displayValue}

                    </option>

                ))}

            </select>

            );

            break;

        default:

            inputElement =<input

            className={inputClasses.join(' ')}

            {...props.elementConfig}

            value={props.value}/>;

    }

    return(

        <div className='Input'>

        <label className='Label'>{props.label}</label>

        {inputElement}

        </div>

    );

};

export default Input;

Input.css

.Invalid {

    border: 1px solid red;

    background-color: salmon;

}

ContactData.js

import React, { useState } from 'react';

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

import './ContactData.css'

import Input from '../../../components/UI/Input-Form/Input';

const ContactData = (props) => {

    const [state,setState] = useState({

        name: '',

        email: '',

        address: {

            street: '',

            postalCode:''

        }

    });

    const [formState,setFormState] = useState({

        orderForm:{

            name: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Your Name'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false

                },

            street: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Street'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false

                },

            zipCode: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Zip Code'

                },

                value: '',

                validation: {

                    required: true,

                    minLength: 5,

                    maxLength: 5

                },

                valid: false,

                },

            country: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Country'

                },

                value: '',

                validation: {

                    required: true,

                },

                valid: false

                },

            email: {

                elementType: 'input',

                elementConfig: {

                    type: 'emai',

                    placeholder: 'Your Email'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false

                },

            deliveryMethod: {

                elementType: 'select',

                elementConfig: {

                    options: [

                        {value: 'fastest',displayValue: 'fastest'},

                        {value: 'cheapest',displayValue: 'cheapest'},

                        ]

                },

                value: ''

                },

        }

    })

    const orderHandler = (event) => {

        event.preventDefault();

        // console.log(props.ingredients)

        const formData = {};//empty object

        for ( let formElementIdentifier in formState.orderForm){

            formData[formElementIdentifier] = formState.orderForm[formElementIdentifier].value;

        }

        const order= {

            // ingredients: props.ingredients,

            // price: props.price,

            orderData: formData

        }

        alert('see on console');

        console.log(order);

        // console.log(ingredients);

        // console.log(price);

    }

    const formElementsArray = [];

    for(let key in formState.orderForm){

        formElementsArray.push({

            id: key,

            config: formState.orderForm[key]

        })

    }

    const checkValidty = (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 inputChangedHandler = (event,inputIdentifier) => {

        console.log(event.target.value);

        const updatedOrderForm = {

            ...formState.orderForm

        }

        const updatedFormElement= {

            ...updatedOrderForm[inputIdentifier]

        };

        updatedFormElement.value = event.target.value;

        updatedFormElement.valid = checkValidty(updatedFormElement.value,updatedFormElement.validation);

        updatedOrderForm[inputIdentifier] = updatedFormElement;

        console.log(updatedFormElement)

        setFormState({orderForm: updatedOrderForm})

    }

    return(

        <div className="ContactData">

            <h4>ENTER CONTACT</h4>

            <form onSubmit={orderHandler}>

                {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)}

                     />

# IMPROVING VISUAL FEEDBACK

ContactData.js

 const [formState,setFormState] = useState({

        orderForm:{

            name: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Your Name'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false,

                touched: false

                },

            street: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Street'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false,

                touched: false

                },

            zipCode: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Zip Code'

                },

                value: '',

                validation: {

                    required: true,

                    minLength: 5,

                    maxLength: 5

                },

                valid: false,

                touched: false

                },

            country: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Country'

                },

                value: '',

                validation: {

                    required: true,

                },

                valid: false,

                touched: false

                },

            email: {

                elementType: 'input',

                elementConfig: {

                    type: 'emai',

                    placeholder: 'Your Email'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false ,

                touched: false

                },

            deliveryMethod: {

                elementType: 'select',

                elementConfig: {

                    options: [

                        {value: 'fastest',displayValue: 'fastest'},

                        {value: 'cheapest',displayValue: 'cheapest'},

                        ]

                },

                value: ''

                },

        }

    })

const inputChangedHandler = (event,inputIdentifier) => {

        console.log(event.target.value);

        const updatedOrderForm = {

            ...formState.orderForm

        }

        const updatedFormElement= {

            ...updatedOrderForm[inputIdentifier]

        };

        updatedFormElement.value = event.target.value;

        updatedFormElement.valid = checkValidty(updatedFormElement.value,updatedFormElement.validation);

        updatedFormElement.touched = true;

        updatedOrderForm[inputIdentifier] = updatedFormElement;

        console.log(updatedFormElement)

        setFormState({orderForm: updatedOrderForm})

    }

    return(

        <div className="ContactData">

            <h4>ENTER CONTACT</h4>

            <form onSubmit={orderHandler}>

                {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)}

                     />

                ))}

                <Button

                btnType="Success"

                clicked={orderHandler}>ORDER</Button>

            </form>

        </div>

    );

}

Input.js

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

        inputClasses.push('Invalid');

    }

# HANDLING OVERALL FORM VALIDITY [DISABLE BUTTON]

NOTE: undefined are always treated as false in conditional.

ContactData.js

import React, { useState } from 'react';

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

import './ContactData.css'

import Input from '../../../components/UI/Input-Form/Input';

const ContactData = (props) => {

    const [state,setState] = useState({

        name: '',

        email: '',

        address: {

            street: '',

            postalCode:''

        }

    });

    const [formState,setFormState] = useState({

        orderForm:{

            name: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Your Name'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false,

                touched: false

                },

            street: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Street'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false,

                touched: false

                },

            zipCode: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Zip Code'

                },

                value: '',

                validation: {

                    required: true,

                    minLength: 5,

                    maxLength: 5

                },

                valid: false,

                touched: false

                },

            country: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Country'

                },

                value: '',

                validation: {

                    required: true,

                },

                valid: false,

                touched: false

                },

            email: {

                elementType: 'input',

                elementConfig: {

                    type: 'emai',

                    placeholder: 'Your Email'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false ,

                touched: false

                },

            deliveryMethod: {

                elementType: 'select',

                elementConfig: {

                    options: [

                        {value: 'fastest',displayValue: 'fastest'},

                        {value: 'cheapest',displayValue: 'cheapest'},

                        ]

                },

                value: '',

                valid: true

                },

        },

        formIsValid: false,

    })

    const orderHandler = (event) => {

        event.preventDefault();

        // console.log(props.ingredients)

        const formData = {};//empty object

        for ( let formElementIdentifier in formState.orderForm){

            formData[formElementIdentifier] = formState.orderForm[formElementIdentifier].value;

        }

        const order= {

            // ingredients: props.ingredients,

            // price: props.price,

            orderData: formData

        }

        alert('see on console');

        console.log(order);

        // console.log(ingredients);

        // console.log(price);

    }

    const formElementsArray = [];

    for(let key in formState.orderForm){

        formElementsArray.push({

            id: key,

            config: formState.orderForm[key]

        })

    }

    const checkValidty = (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 inputChangedHandler = (event,inputIdentifier) => {

        console.log(event.target.value);

        const updatedOrderForm = {

            ...formState.orderForm

        }

        const updatedFormElement= {

            ...updatedOrderForm[inputIdentifier]

        };

        updatedFormElement.value = event.target.value;

        updatedFormElement.valid = checkValidty(updatedFormElement.value,updatedFormElement.validation);

        updatedFormElement.touched = true;

        updatedOrderForm[inputIdentifier] = updatedFormElement;

        let formIsValid = true;

//checking all input’s validities in updated order form

        for(let inputIdentifier in updatedOrderForm){

//valid property checking and formisValid is true or not;

//F && T => F, T && T => T

            formIsValid = updatedOrderForm[inputIdentifier].valid && formIsValid;

        }

        console.log(formIsValid);

        console.log(updatedFormElement)

        setFormState({orderForm: updatedOrderForm, formIsValid: formIsValid})

    }

    return(

        <div className="ContactData">

            <h4>ENTER CONTACT</h4>

            <form onSubmit={orderHandler}>

                {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)}

                     />

                ))}

                <Button

                btnType="Success"

                clicked={orderHandler}

                disabled={!formState.formIsValid}

                >ORDER</Button>

            </form>

        </div>

    );

}

export default ContactData;

Button.js

import React from 'react';

import './Button.css';

const button = (props) => (

    <button

    disabled={props.disabled}

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

    onClick={props.clicked}>{props.children}</button>

);

export default button;

Button.css

.Button {

    background-color: transparent;

    border: none;

    color: white;

    outline: none;

    cursor: pointer;

    font: inherit;

    padding: 10px;

    margin: 10px;

    font-weight: bold;

}

.Button:first-of-type {

    margin-left: 0;

    padding-left: 0;

}

.Button:disabled{

    color:#ccc;

    cursor: not-allowed;

}

.Success {

    color: #5C9210;

}

.Danger {

    color: #944317;

}

# WORKING ON AN ERROR OCCUR IN SELECTOR INPUT

ContactData.js

import React, { useState } from 'react';

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

import './ContactData.css'

import Input from '../../../components/UI/Input-Form/Input';

const ContactData = (props) => {

    const [state,setState] = useState({

        name: '',

        email: '',

        address: {

            street: '',

            postalCode:''

        }

    });

    const [formState,setFormState] = useState({

        orderForm:{

            name: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Your Name'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false,

                touched: false

                },

            street: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Street'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false,

                touched: false

                },

            zipCode: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Zip Code'

                },

                value: '',

                validation: {

                    required: true,

                    minLength: 5,

                    maxLength: 5

                },

                valid: false,

                touched: false

                },

            country: {

                elementType: 'input',

                elementConfig: {

                    type: 'text',

                    placeholder: 'Country'

                },

                value: '',

                validation: {

                    required: true,

                },

                valid: false,

                touched: false

                },

            email: {

                elementType: 'input',

                elementConfig: {

                    type: 'emai',

                    placeholder: 'Your Email'

                },

                value: '',

                validation: {

                    required: true

                },

                valid: false ,

                touched: false

                },

            deliveryMethod: {

                elementType: 'select',

                elementConfig: {

                    options: [

                        {value: 'fastest',displayValue: 'fastest'},

                        {value: 'cheapest',displayValue: 'cheapest'},

                        ]

                },

                value: '',

                validation: {},//empty object

                valid: true

                },

        },

        formIsValid: false,

    })

    const orderHandler = (event) => {

        event.preventDefault();

        // console.log(props.ingredients)

        const formData = {};//empty object

        for ( let formElementIdentifier in formState.orderForm){

            formData[formElementIdentifier] = formState.orderForm[formElementIdentifier].value;

        }

        const order= {

            // ingredients: props.ingredients,

            // price: props.price,

            orderData: formData

        }

        alert('see on console');

        console.log(order);

        // console.log(ingredients);

        // console.log(price);

    }

    const formElementsArray = [];

    for(let key in formState.orderForm){

        formElementsArray.push({

            id: key,

            config: formState.orderForm[key]

        })

    }

    const checkValidty = (value, rules) => {

        let isValid = true;

        //ALTERNATIVE ONE

        if(!rules) {

            return 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 inputChangedHandler = (event,inputIdentifier) => {

        console.log(event.target.value);

        const updatedOrderForm = {

            ...formState.orderForm

        }

        const updatedFormElement= {

            ...updatedOrderForm[inputIdentifier]

        };

        updatedFormElement.value = event.target.value;

        updatedFormElement.valid = checkValidty(updatedFormElement.value,updatedFormElement.validation);

        updatedFormElement.touched = true;

        updatedOrderForm[inputIdentifier] = updatedFormElement;

        let formIsValid = true;

        for(let inputIdentifier in updatedOrderForm){

            formIsValid = updatedOrderForm[inputIdentifier].valid && formIsValid;

        }

        console.log(formIsValid);

        console.log(updatedFormElement)

        setFormState({orderForm: updatedOrderForm, formIsValid: formIsValid})

    }

    return(

        <div className="ContactData">

            <h4>ENTER CONTACT</h4>

            <form onSubmit={orderHandler}>

                {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)}

                     />

                ))}

                <Button

                btnType="Success"

                clicked={orderHandler}

                disabled={!formState.formIsValid}

                >ORDER</Button>

            </form>

        </div>

    );

}

export default ContactData;

# FIXING A BUG IN SELECTOR: IF WE CHANGE THE STATE, THEN IT WILL STORE NULL VALUE.

deliveryMethod: {

                elementType: 'select',

                elementConfig: {

                    options: [

                        {value: 'fastest',displayValue: 'fastest'},

                        {value: 'cheapest',displayValue: 'cheapest'},

                        ]

                },

                value: 'fastest',

                validation: {},

                valid: true

                },