**CONTACTDATA.JS**

import React, { Component } from "react";

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

import Spinner from "../../../components/UI/Spinner/Spinner";

import axios from "../../../axios-orders";

import Input from "../../../components/UI/Input/Input";

class ContactData extends Component {

  state = {

    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: "email",

          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: {},

        valid: true

      }

    },

    formIsValid: false,

    loading: false

  };

  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;

  }

  inputChangedHandler = (event, inputIdentifier) => {

    const updatedOrderForm = {

      ...this.state.orderForm

    };

    const updatedFormElement = { ...updatedOrderForm[inputIdentifier] };

    updatedFormElement.value = event.target.value;

    updatedFormElement.valid = this.checkValidity(

      updatedFormElement.value,

      updatedFormElement.validation

    );

    updatedFormElement.touched = true;

    updatedOrderForm[inputIdentifier] = updatedFormElement;

    let formIsValid = true;

    for (let inputIdentifier in updatedOrderForm) {

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

    }

    this.setState({

      orderForm: updatedOrderForm,

      formIsValid

    });

  };

  orderHandler = e => {

    e.preventDefault();

    this.setState({ loading: true });

    const formData = {};

    for (let formElementIdentifier in this.state.orderForm) {

      formData[formElementIdentifier] = this.state.orderForm[

        formElementIdentifier

      ].value;

    }

    const order = {

      ingredients: this.props.ingredients,

      price: this.props.price,

      orderData: formData

    };

    axios

      .post("/orders.json", order)

      .then(response => {

        this.setState({ loading: false });

        this.props.history.push("/");

        console.log(response);

      })

      .catch(err => {

        this.setState({ loading: false });

        console.error(err);

      });

  };

  render() {

    const formElementsArray = [];

    for (let key in this.state.orderForm) {

      formElementsArray.push({

        id: key,

        config: this.state.orderForm[key]

      });

    }

    let form = (

      <form onSubmit={this.orderHandler}>

        {formElementsArray.map((formElement, index) => (

          <Input

            key={index}

            invalid={!formElement.config.valid}

            shouldValidate={formElement.config.validation}

            elementType={formElement.config.elementType}

            elementConfig={formElement.config.elementConfig}

            value={formElement.config.value}

            touched={formElement.config.touched}

            changed={event => this.inputChangedHandler(event, formElement.id)}

          />

        ))}

        <Button btnType="button\_\_success" disabled={!this.state.formIsValid}>

          ORDER

        </Button>

      </form>

    );

    if (this.state.loading) {

      form = <Spinner />;

    }

    return (

      <div className="contact-data">

        <h4>Enter your Contact Data</h4>

        {form}

      </div>

    );

  }

}

export default ContactData;

**INPUT.JS**

import React from "react";

const Input = props => {

  let inputElement = null;

  let invalidClass = "";

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

    invalidClass = "input-element\_\_invalid";

  }

  switch (props.elementType) {

    case "input":

      inputElement = (

        <input

          className={"input-element" + " " + invalidClass}

          {...props.elementConfig}

          value={props.value}

          onChange={props.changed}

        />

      );

      break;

    case "textarea":

      inputElement = (

        <textarea

          className={"input-element" + " " + invalidClass}

          {...props.elementConfig}

          value={props.value}

          onChange={props.changed}

        />

      );

      break;

    case "select":

      inputElement = (

        <select

          className="input-element"

          value={props.value}

          onChange={props.changed}

        >

          {props.elementConfig.options.map((option, index) => (

            <option key={index} value={option.value}>

              {option.displayValue}

            </option>

          ))}

        </select>

      );

      break;

    default:

      inputElement = (

        <input

          className="input-element"

          {...props.elementConfig}

          value={props.value}

        />

      );

  }

  return (

    <div className="input">

      <label className="label" htmlFor="">

        {props.label}

      </label>

      {inputElement}

    </div>

  );

};

export default Input;

**BUTTON.JS**

import React from "react";

const button = props => (

  <button

    className={["button", props.btnType].join(" ")}

    onClick={props.clicked}

    disabled={props.disabled}

  >

    {props.children}

  </button>

);

export default button;

**REDUX FORM**

<https://redux-form.com/8.3.0/>

1. yarn add redux form

import { combineReducers } from "redux";

import { reducer as formReducer } from "redux-form";

import authReducer from "./authReducer";

export default combineReducers({

  auth: authReducer,

  form: formReducer

});