Кислый Илья 107а1

\\1

import React, { useState } from 'react';

const CitySelector = () => {

const [selectedCity, setSelectedCity] = useState('');

const [message, setMessage] = useState('');

const handleCityChange = (event) => {

const city = event.target.value;

setSelectedCity(city);

if (city !== 'Рио-де-Жанейро') {

setMessage('Нет, это не Рио-де-Жанейро!');

} else {

setMessage('');

}

};

return (

<div>

<select value={selectedCity} onChange={handleCityChange}>

<option value="">Выберите город</option>

<option value="Рио-де-Жанейро">Рио-де-Жанейро</option>

<option value="Москва">Москва</option>

<option value="Лондон">Лондон</option>

<option value="Токио">Токио</option>

{/\* Добавьте другие города по желанию \*/}

</select>

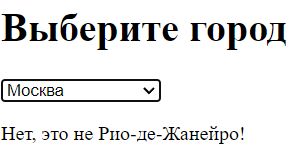
{message && <p>{message}</p>}

</div>

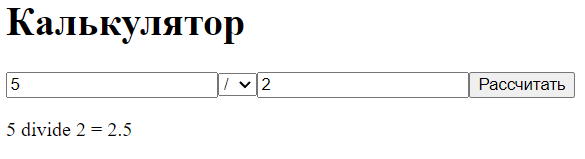
);

};

export default CitySelector;



\\2



import React, { useState } from 'react';

const Calculator = () => {

const [num1, setNum1] = useState('');

const [num2, setNum2] = useState('');

const [result, setResult] = useState('');

const [operation, setOperation] = useState('');

const operations = [

{ value: 'add', label: '+' },

{ value: 'subtract', label: '-' },

{ value: 'multiply', label: '\*' },

{ value: 'divide', label: '/' },

];

const handleOperationChange = (event) => {

setOperation(event.target.value);

};

const calculate = () => {

let res = 0;

switch (operation) {

case 'add':

res = parseFloat(num1) + parseFloat(num2);

break;

case 'subtract':

res = parseFloat(num1) - parseFloat(num2);

break;

case 'multiply':

res = parseFloat(num1) \* parseFloat(num2);

break;

case 'divide':

if (parseFloat(num2) !== 0) {

res = parseFloat(num1) / parseFloat(num2);

} else {

alert('Деление на ноль невозможно');

return;

}

break;

default:

return;

}

setResult(`${num1} ${operation} ${num2} = ${res}`);

};

return (

<div>

<input

type="number"

value={num1}

onChange={(e) => setNum1(e.target.value)}

placeholder="Введите первое число"

/>

<select value={operation} onChange={handleOperationChange}>

{operations.map((op) => (

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

{op.label}

</option>

))}

</select>

<input

type="number"

value={num2}

onChange={(e) => setNum2(e.target.value)}

placeholder="Введите второе число"

/>

<button onClick={calculate}>Рассчитать</button>

<p>{result}</p>

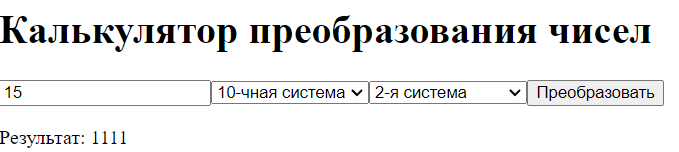
</div>

);

};

export default Calculator;

\\3



import React, { useState } from 'react';

const NumberConverter = () => {

const [inputNumber, setInputNumber] = useState('');

const [inputBase, setInputBase] = useState(10);

const [outputBase, setOutputBase] = useState(2);

const [convertedNumber, setConvertedNumber] = useState('');

const handleInputNumberChange = (event) => {

setInputNumber(event.target.value);

};

const handleInputBaseChange = (event) => {

setInputBase(parseInt(event.target.value, 10));

};

const handleOutputBaseChange = (event) => {

setOutputBase(parseInt(event.target.value, 10));

};

const convertNumber = () => {

try {

const numberInDecimal = parseInt(inputNumber, inputBase);

setConvertedNumber(numberInDecimal.toString(outputBase));

} catch (error) {

setConvertedNumber('Ошибка преобразования');

}

};

return (

<div>

<input

type="text"

value={inputNumber}

onChange={handleInputNumberChange}

placeholder="Введите число"

/>

<select value={inputBase} onChange={handleInputBaseChange}>

<option value="10">10-чная система</option>

<option value="2">2-я система</option>

</select>

<select value={outputBase} onChange={handleOutputBaseChange}>

<option value="2">2-я система</option>

<option value="10">10-чная система</option>

</select>

<button onClick={convertNumber}>Преобразовать</button>

<p>Результат: {convertedNumber}</p>

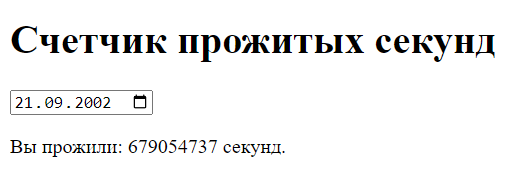
</div>

);

};

export default NumberConverter;

\\4



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

const BirthdayCounter = () => {

const [birthday, setBirthday] = useState('');

const [secondsLived, setSecondsLived] = useState(0);

const handleBirthdayChange = (event) => {

setBirthday(event.target.value);

};

useEffect(() => {

const calculateSecondsLived = () => {

const now = new Date();

const birthdayDate = new Date(birthday);

const secondsLived = (now - birthdayDate) / 1000;

setSecondsLived(secondsLived);

};

calculateSecondsLived();

const intervalId = setInterval(calculateSecondsLived, 1000);

return () => clearInterval(intervalId);

}, [birthday]);

return (

<div>

<input

type="date"

value={birthday}

onChange={handleBirthdayChange}

placeholder="Введите дату рождения"

/>

<p>Вы прожили: {secondsLived.toFixed(0)} секунд.</p>

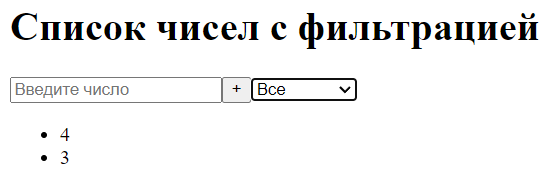
</div>

);

};

export default BirthdayCounter;

\\5



import React, { useState } from 'react';

const NumberList = () => {

const [numbers, setNumbers] = useState([]);

const [filter, setFilter] = useState('all');

const [inputNumber, setInputNumber] = useState('');

const handleInputChange = (event) => {

setInputNumber(event.target.value);

};

const handleAddNumber = () => {

const newNumber = parseInt(inputNumber, 10);

if (!isNaN(newNumber)) {

setNumbers([...numbers, newNumber]);

setInputNumber('');

}

};

const handleFilterChange = (event) => {

setFilter(event.target.value);

};

const filteredNumbers = numbers.filter((number) => {

if (filter === 'all') return true;

if (filter === 'even') return number % 2 === 0;

if (filter === 'odd') return number % 2 !== 0;

return false;

});

return (

<div>

<input

type="number"

value={inputNumber}

onChange={handleInputChange}

placeholder="Введите число"

/>

<button onClick={handleAddNumber}>+</button>

<select value={filter} onChange={handleFilterChange}>

<option value="all">Все</option>

<option value="even">Четные</option>

<option value="odd">Нечетные</option>

</select>

<ul>

{filteredNumbers.map((number, index) => (

<li key={index}>{number}</li>

))}

</ul>

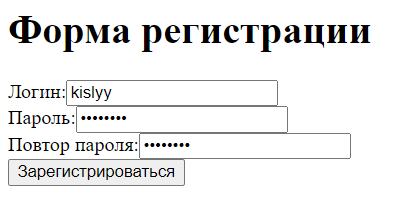
</div>

);

};

export default NumberList;

\\6



import React, { useState } from 'react';

const RegistrationForm = () => {

const [username, setUsername] = useState('');

const [password, setPassword] = useState('');

const [confirmPassword, setConfirmPassword] = useState('');

const [errors, setErrors] = useState({});

const validateForm = () => {

let formIsValid = true;

let newErrors = {};

if (!username) {

newErrors.username = 'Логин обязателен';

formIsValid = false;

} else if (username.length < 6 || username.length > 20) {

newErrors.username = 'Логин должен содержать от 6 до 20 символов';

formIsValid = false;

} else if (!/^[a-zA-Z0-9]+$/.test(username)) {

newErrors.username = 'Логин может содержать только буквы латинского алфавита и цифры';

formIsValid = false;

}

if (!password) {

newErrors.password = 'Пароль обязателен';

formIsValid = false;

}

if (!confirmPassword) {

newErrors.confirmPassword = 'Повтор пароля обязателен';

formIsValid = false;

} else if (password !== confirmPassword) {

newErrors.confirmPassword = 'Пароли не совпадают';

formIsValid = false;

}

setErrors(newErrors);

return formIsValid;

};

const handleSubmit = (event) => {

event.preventDefault();

if (validateForm()) {

console.log('Форма отправлена успешно');

}

};

return (

<form onSubmit={handleSubmit}>

<div>

<label htmlFor="username">Логин:</label>

<input

type="text"

id="username"

value={username}

onChange={(e) => setUsername(e.target.value)}

/>

{errors.username && <p>{errors.username}</p>}

</div>

<div>

<label htmlFor="password">Пароль:</label>

<input

type="password"

id="password"

value={password}

onChange={(e) => setPassword(e.target.value)}

/>

{errors.password && <p>{errors.password}</p>}

</div>

<div>

<label htmlFor="confirmPassword">Повтор пароля:</label>

<input

type="password"

id="confirmPassword"

value={confirmPassword}

onChange={(e) => setConfirmPassword(e.target.value)}

/>

{errors.confirmPassword && <p>{errors.confirmPassword}</p>}

</div>

<button type="submit">Зарегистрироваться</button>

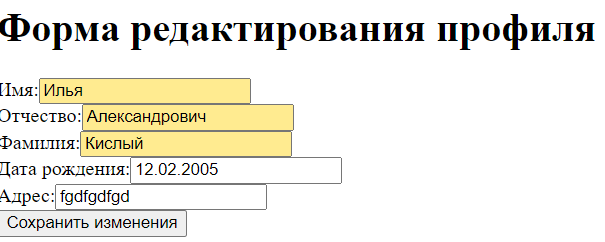
</form>

);

};

export default RegistrationForm;

\\7



import React, { useState } from 'react';

const ProfileEditForm = () => {

const [firstName, setFirstName] = useState('');

const [middleName, setMiddleName] = useState('');

const [lastName, setLastName] = useState('');

const [birthDate, setBirthDate] = useState('');

const [address, setAddress] = useState('');

const [errors, setErrors] = useState({});

const validateForm = () => {

let formIsValid = true;

let newErrors = {};

// Валидация имени

if (!firstName) {

newErrors.firstName = 'Имя обязательно';

formIsValid = false;

}

// Валидация отчества

if (!middleName) {

newErrors.middleName = 'Отчество обязательно';

formIsValid = false;

}

// Валидация фамилии

if (!lastName) {

newErrors.lastName = 'Фамилия обязательна';

formIsValid = false;

}

// Валидация даты рождения (если указана)

if (birthDate && !/^\d{2}\.\d{2}\.\d{4}$/.test(birthDate)) {

newErrors.birthDate = 'Дата рождения должна быть в формате ДД.ММ.ГГГГ';

formIsValid = false;

}

setErrors(newErrors);

return formIsValid;

};

const handleSubmit = (event) => {

event.preventDefault();

if (validateForm()) {

// Здесь можно обработать успешную отправку формы

console.log('Форма отправлена успешно');

}

};

return (

<form onSubmit={handleSubmit}>

<div>

<label htmlFor="firstName">Имя:</label>

<input

type="text"

id="firstName"

value={firstName}

onChange={(e) => setFirstName(e.target.value)}

/>

{errors.firstName && <p>{errors.firstName}</p>}

</div>

<div>

<label htmlFor="middleName">Отчество:</label>

<input

type="text"

id="middleName"

value={middleName}

onChange={(e) => setMiddleName(e.target.value)}

/>

{errors.middleName && <p>{errors.middleName}</p>}

</div>

<div>

<label htmlFor="lastName">Фамилия:</label>

<input

type="text"

id="lastName"

value={lastName}

onChange={(e) => setLastName(e.target.value)}

/>

{errors.lastName && <p>{errors.lastName}</p>}

</div>

<div>

<label htmlFor="birthDate">Дата рождения:</label>

<input

type="text"

id="birthDate"

value={birthDate}

onChange={(e) => setBirthDate(e.target.value)}

placeholder="ДД.ММ.ГГГГ"

/>

{errors.birthDate && <p>{errors.birthDate}</p>}

</div>

<div>

<label htmlFor="address">Адрес:</label>

<input

type="text"

id="address"

value={address}

onChange={(e) => setAddress(e.target.value)}

/>

</div>

<button type="submit">Сохранить изменения</button>

</form>

);

};

export default ProfileEditForm;