|  |  |
| --- | --- |
| **Ex:No:10** | **FORM VALIDATION USING REACT** |
| **Date:** |

**Aim:**

**Procedure:**

**Program:**

import React, { useState } from 'react';

import './App.css'

const FormValidationExample = () => {

const [formData, setFormData] = useState({

username: '',

email: '',

password: '',

confirmPassword: '',

phone: '',

address:'',

city: '',

state: '',

dob: '',

gender: ''

})

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

const handleChange = (e) => {

const {name, value} = e.target;

setFormData({

...formData, [name] : value

})

}

const handleSubmit = (e) => {

e.preventDefault()

const validationErrors = {}

if(!formData.username.trim()) {

validationErrors.username = "username is required"

}

if(!formData.email.trim()) {

validationErrors.email = "email is required"

} else if(!/\S+@\S+\.\S+/.test(formData.email)){

validationErrors.email = "email is not valid"

}

if(!formData.password.trim()) {

validationErrors.password = "password is required"

} else if(formData.password.length < 6){

validationErrors.password = "password should be at least 6 char"

}

if(formData.confirmPassword !== formData.password) {

validationErrors.confirmPassword = "password not matched"

}

if(!formData.phone.trim()) {

validationErrors.phone = "phone no is required"

} else if(formData.phone.length < 6){

validationErrors.phone = "phone number should contain 10 digits"

}

if(!formData.address.trim()) {

validationErrors.address = "Address is required"

} else if(formData.address.length < 5){

validationErrors.address = "Address should contain atleast 5 character"

}

if(!formData.dob.trim()) {

validationErrors.dob = "DOB is required"

} else if(formData.dob.length < 5){

validationErrors.dob = "DOB is required"

}

if (!formData.age) {

errors.age = 'ra.validation.required';

} else if (formData.age < 18) {

errors.age = {

message: 'ra.validation.minValue',

args: { min: 18 }

};

}

setErrors(validationErrors)

if(Object.keys(validationErrors).length === 0) {

alert("Form Submitted successfully")

}

}

return (

<div className='bg img'>

<div className='reg'><b>Registration Form</b></div> <br></br>

<div className='e'>

<form onSubmit={handleSubmit}>

<div>

<label>Name:</label>

<input

type="text"

name="username"

placeholder='name'

autoComplete='off'

onChange={handleChange}

/>

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

</div>

<div>

<span><label>Email:</label></span>

<input

type="email"

name="email"

placeholder='example@gmail.com'

autoComplete='off'

onChange={handleChange}

/>

{errors.email && <span>{errors.email}</span>}

</div>

<div>

<label>Password:</label>

<input

type="password"

name="password"

placeholder='\*\*\*\*\*\*'

onChange={handleChange}

/>

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

</div>

<div>

<label>Confirm Password:</label>

<input

type="password"

name="confirmPassword"

placeholder='\*\*\*\*\*\*'

onChange={handleChange}

/>

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

</div>

<div>

<label>Phone no:</label>

<input

type="text"

name="phone"

onChange={handleChange}

/>

{errors.phone && <span>{errors.phone}</span>}

</div>

<div>

<label>Address:</label>

<input

type="text"

name="address"

onChange={handleChange}

/>

{errors.address && <span>{errors.address}</span>}

</div>

<div>

<label>City:</label>

<input

type="text"

name="city"

onChange={handleChange}

/>

{errors.city && <span>{errors.city}</span>}

</div>

<div>

<label>State:</label>

<input

type="text"

name="state"

onChange={handleChange}

/>

{errors.state && <span>{errors.state}</span>}

</div>

<div>

<label>DOB:</label>

<input

type="text"

name="dob"

onChange={handleChange}

/>

{errors.dob && <span>{errors.dob}</span>}

</div>

<div>

<label>AGE:</label>

<input

type="text"

name="dob"

onChange={handleChange}

/>

{errors.age && <span>{errors.age}</span>}

</div>

<div>

<label>Gender:</label>

<input

type="text"

name="gender"

onChange={handleChange}

/>

{errors.gender && <span>{errors.gender}</span>}

</div>

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

</form></div>

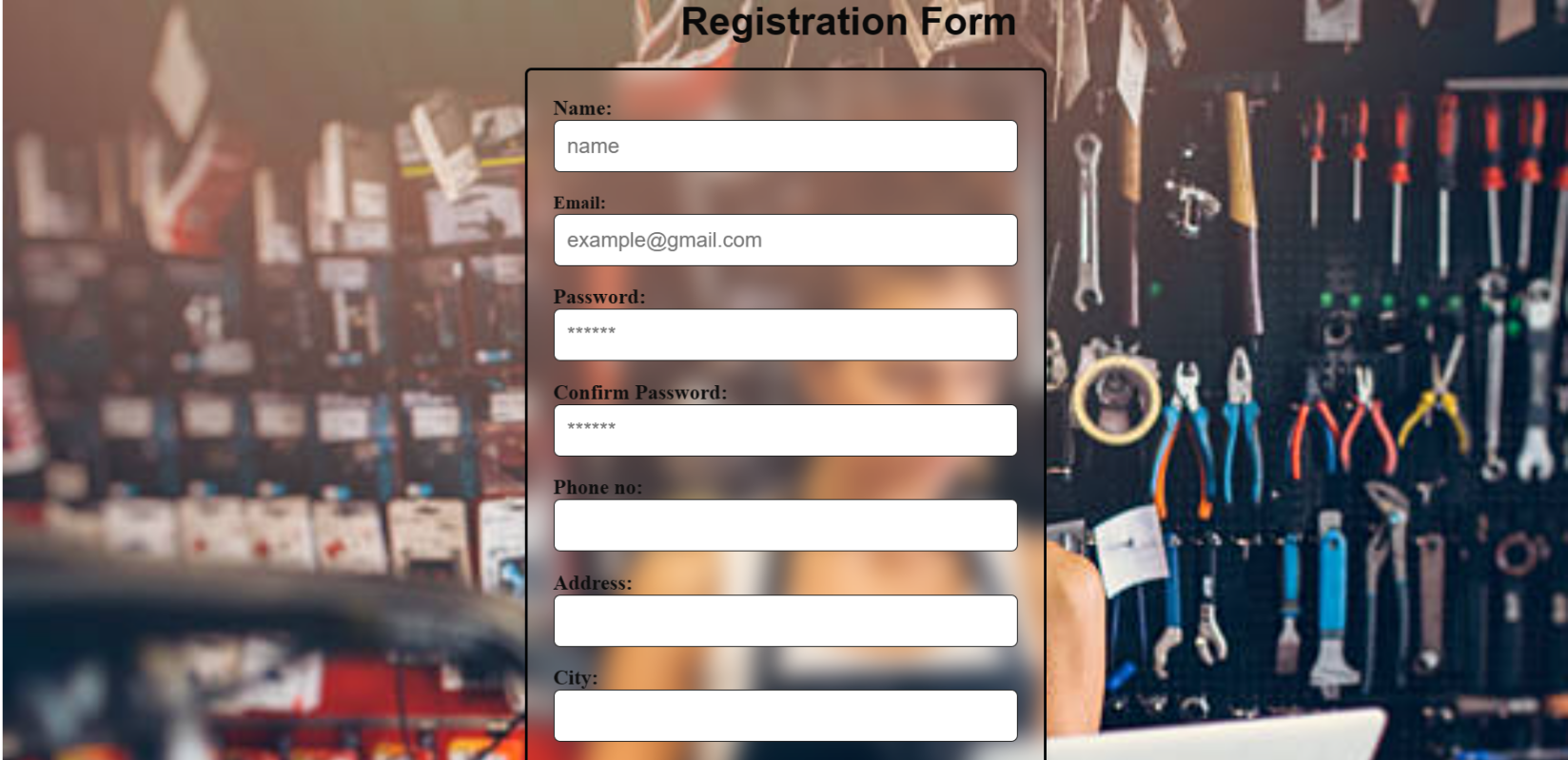
</div>

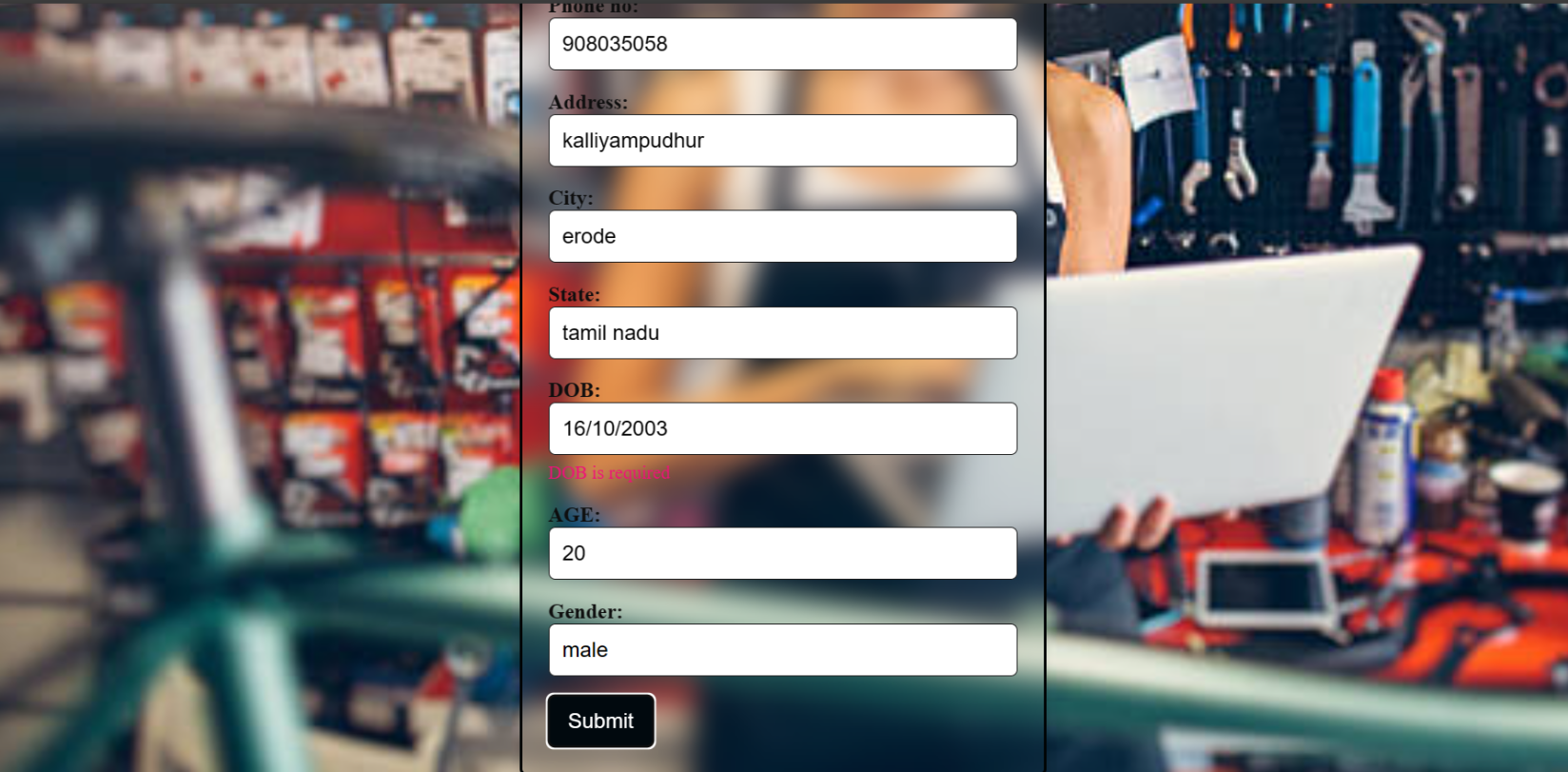
);

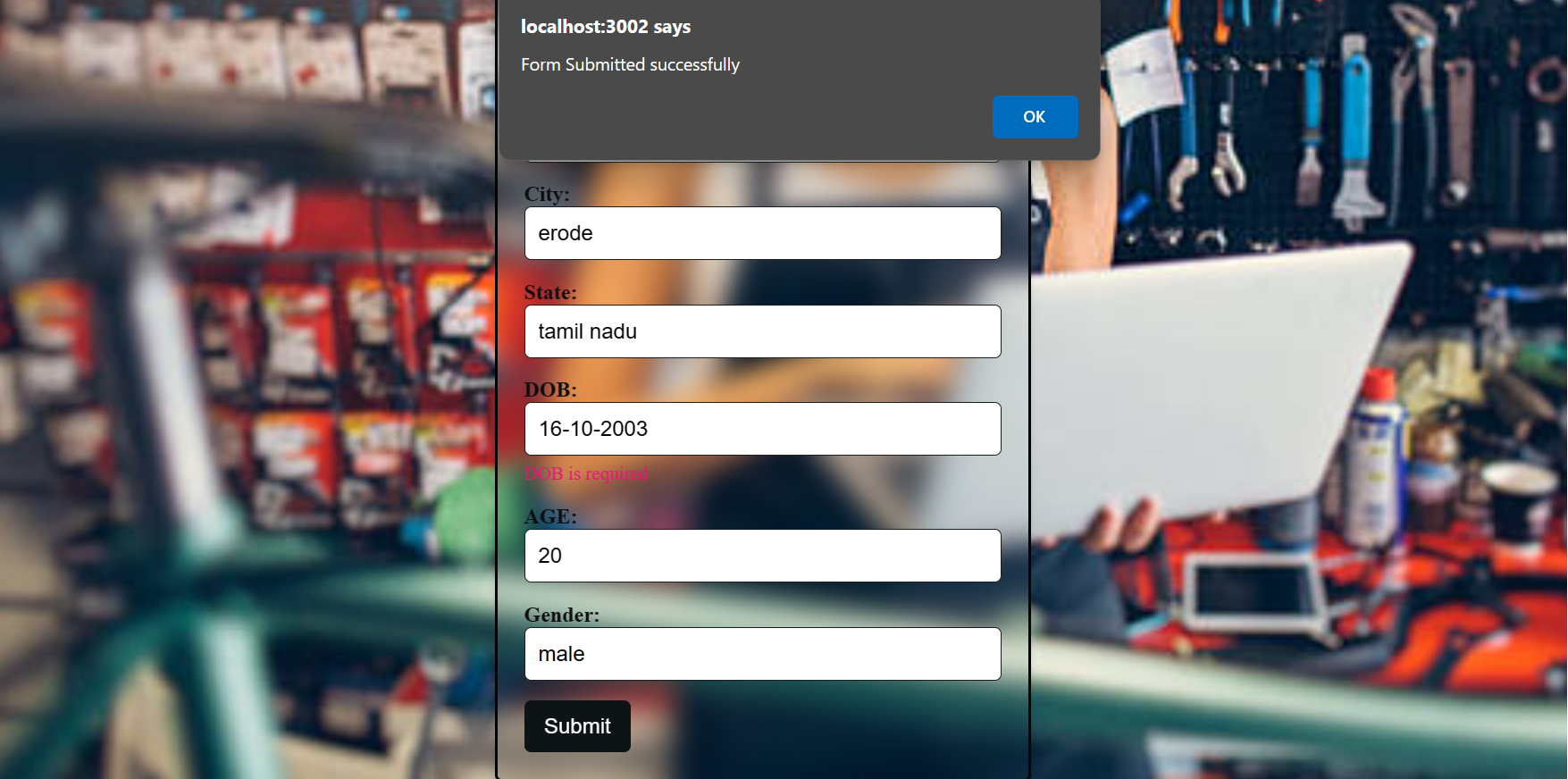
};

export default FormValidationExample;

**Output:**



****



|  |  |  |
| --- | --- | --- |
| **COE** |  |  |
| **RECORD** |  |  |
| **VIVA** |  |  |
| **TOTAL** |  |  |

**Result**