**EXPERIMENT 7**

**AIM: Design a registration form using form elements in ReactJS along with appropriate validations.**

**CODE:**

**App.js:**

import React, { useState } from 'react';

import './App.css';

const App = () => {

  // State for form fields

  const [formData, setFormData] = useState({

    name: '',

    gender: '',

    username: '',

    password: '',

    email: '',

    education: '',

    dob: '',

    hobbies: [],

    about: '',

  });

*// State for form errors*

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

*// Handle input change for all fields*

  const handleChange = (e) => {

    const { name, value, type, checked } = e.target;

    if (type === 'checkbox') {

      let newHobbies = formData.hobbies;

      if (checked) {

        newHobbies.push(value);

      } else {

        newHobbies = newHobbies.filter((hobby) => hobby !== value);

      }

      setFormData({ ...formData, hobbies: newHobbies });

    } else {

      setFormData({ ...formData, [name]: value });

    }

  };

*// Validate form inputs*

  const validate = () => {

    let inputErrors = {};

    if (!formData.name) {

      inputErrors.name = 'Name is required';

    }

    if (!formData.gender) {

      inputErrors.gender = 'Gender is required';

    }

    if (!formData.username) {

      inputErrors.username = 'Username is required';

    }

    if (!formData.password || formData.password.length < 6) {

      inputErrors.password = 'Password must be at least 6 characters long';

    }

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

      inputErrors.email = 'Valid email is required';

    }

    if (!formData.education) {

      inputErrors.education = 'Education level is required';

    }

    if (!formData.dob) {

      inputErrors.dob = 'Date of birth is required';

    }

    if (formData.hobbies.length === 0) {

      inputErrors.hobbies = 'Please select at least one hobby';

    }

    if (!formData.about) {

      inputErrors.about = 'Please tell us about yourself';

    }

    setErrors(inputErrors);

    return Object.keys(inputErrors).length === 0;

  };

*// Handle form submit*

  const handleSubmit = (e) => {

    e.preventDefault();

    if (validate()) {

      alert('Form Submitted Successfully');

    }

  };

  return (

    <div className="container">

      <h1>Registration Form</h1>

      <form onSubmit={handleSubmit}>

        {/\* Name \*/}

        <label>Name:</label>

        <input

          type="text"

          name="name"

          value={formData.name}

          onChange={handleChange}

        />

        {errors.name && <p className="error">{errors.name}</p>}

*{/\* Gender \*/}*

        <label>Gender:</label>

        <div>

          <input

            type="radio"

            name="gender"

            value="male"

            onChange={handleChange}

          /> Male

          <input

            type="radio"

            name="gender"

            value="female"

            onChange={handleChange}

          /> Female

        </div>

        {errors.gender && <p className="error">{errors.gender}</p>}

*{/\* Username \*/}*

        <label>Username:</label>

        <input

          type="text"

          name="username"

          value={formData.username}

          onChange={handleChange}

        />

        {errors.username && <p className="error">{errors.username}</p>}

*{/\* Password \*/}*

        <label>Password:</label>

        <input

          type="password"

          name="password"

          value={formData.password}

          onChange={handleChange}

        />

        {errors.password && <p className="error">{errors.password}</p>}

*{/\* Email \*/}*

        <label>Email:</label>

        <input

          type="email"

          name="email"

          value={formData.email}

          onChange={handleChange}

        />

        {errors.email && <p className="error">{errors.email}</p>}

*{/\* Education \*/}*

        <label>Education:</label>

        <select name="education" value={formData.education} onChange={handleChange}>

          <option value="">Select your education</option>

          <option value="High School">High School</option>

          <option value="Undergraduate">Undergraduate</option>

          <option value="Graduate">Graduate</option>

        </select>

        {errors.education && <p className="error">{errors.education}</p>}

*{/\* Date of Birth \*/}*

        <label>Date of Birth:</label>

        <input

          type="date"

          name="dob"

          value={formData.dob}

          onChange={handleChange}

        />

        {errors.dob && <p className="error">{errors.dob}</p>}

*{/\* Hobbies \*/}*

        <label>Hobbies:</label>

        <div>

          <input

            type="checkbox"

            name="hobbies"

            value="Reading"

            onChange={handleChange}

          /> Reading

          <input

            type="checkbox"

            name="hobbies"

            value="Travelling"

            onChange={handleChange}

          /> Travelling

          <input

            type="checkbox"

            name="hobbies"

            value="Music"

            onChange={handleChange}

          /> Music

        </div>

        {errors.hobbies && <p className="error">{errors.hobbies}</p>}

*{/\* About Yourself \*/}*

        <label>About Yourself:</label>

        <textarea

          name="about"

          value={formData.about}

          onChange={handleChange}

        />

        {errors.about && <p className="error">{errors.about}</p>}

*{/\* Submit Button \*/}*

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

      </form>

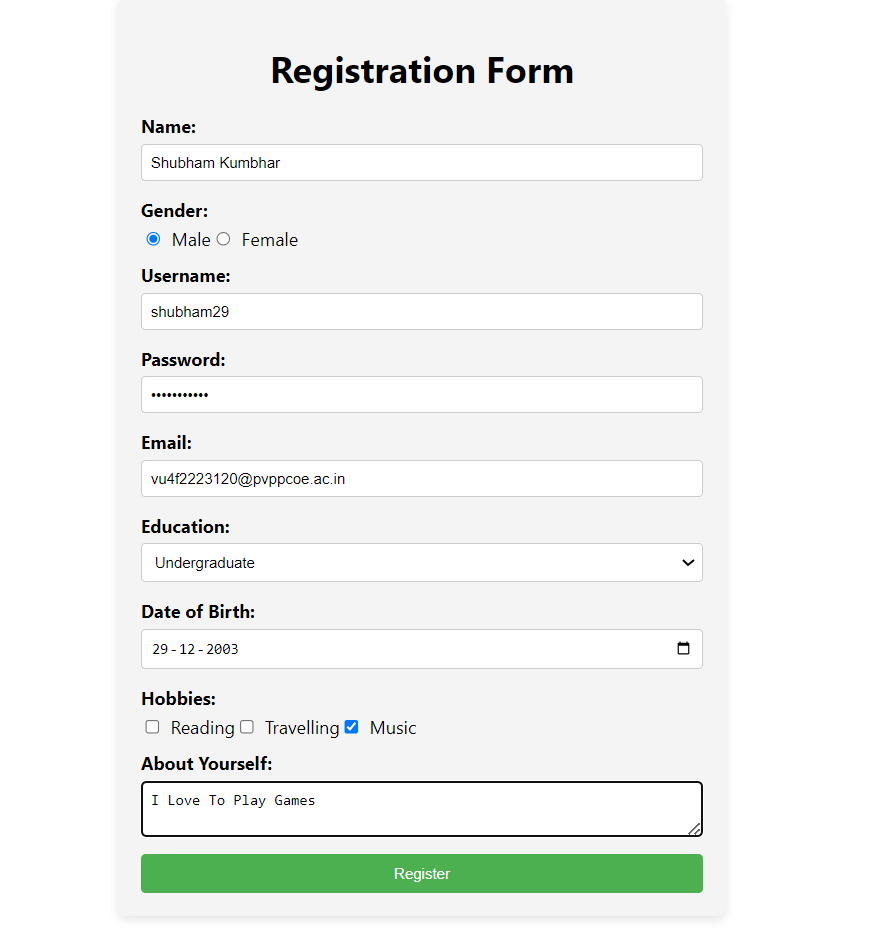
    </div>

  );

};

export default App;

**OUTPUT:**

****

**CONCLUSION: Hence, we have successfully implemented a registration form using form elements in ReactJS along with appropriate validations.**