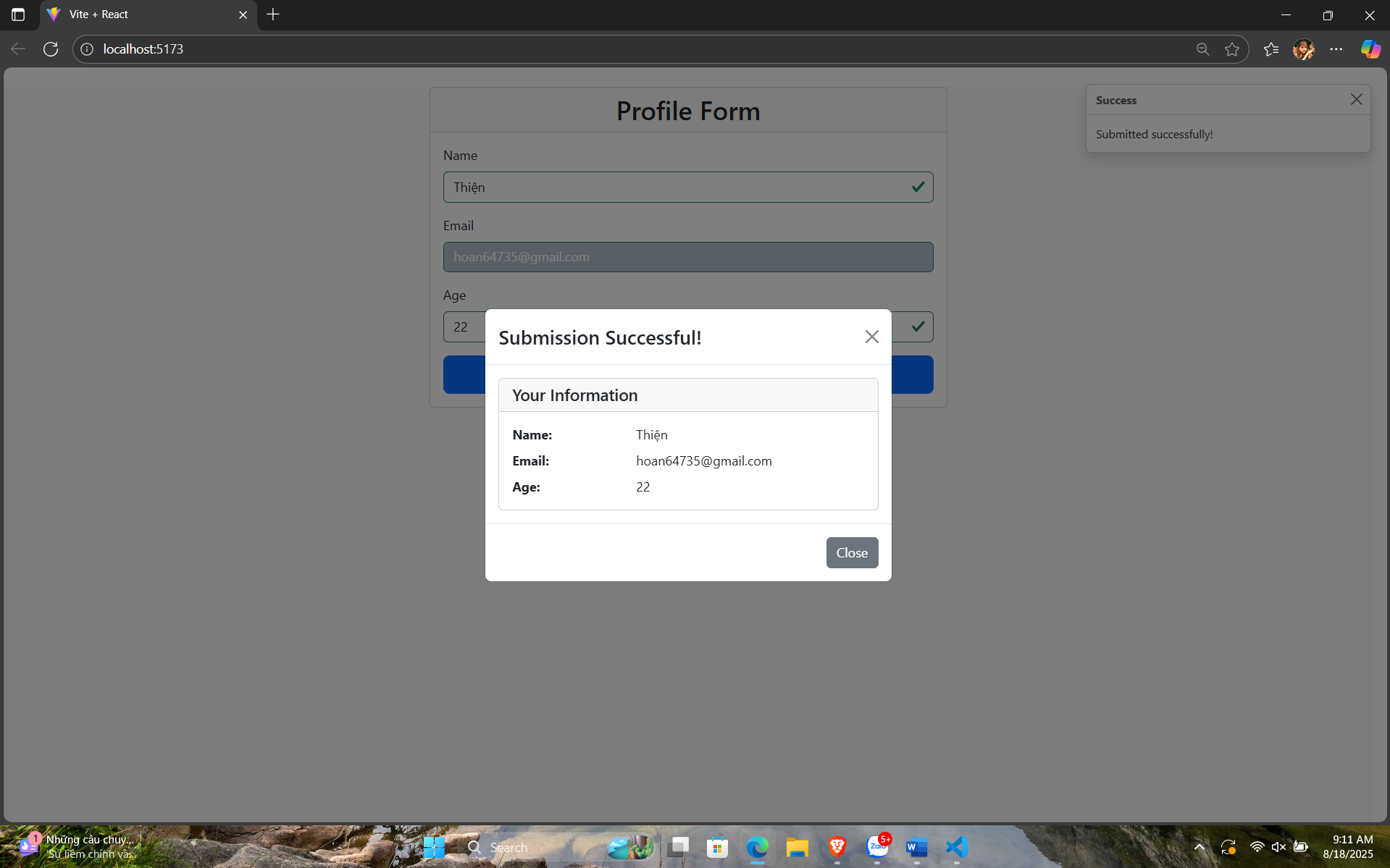
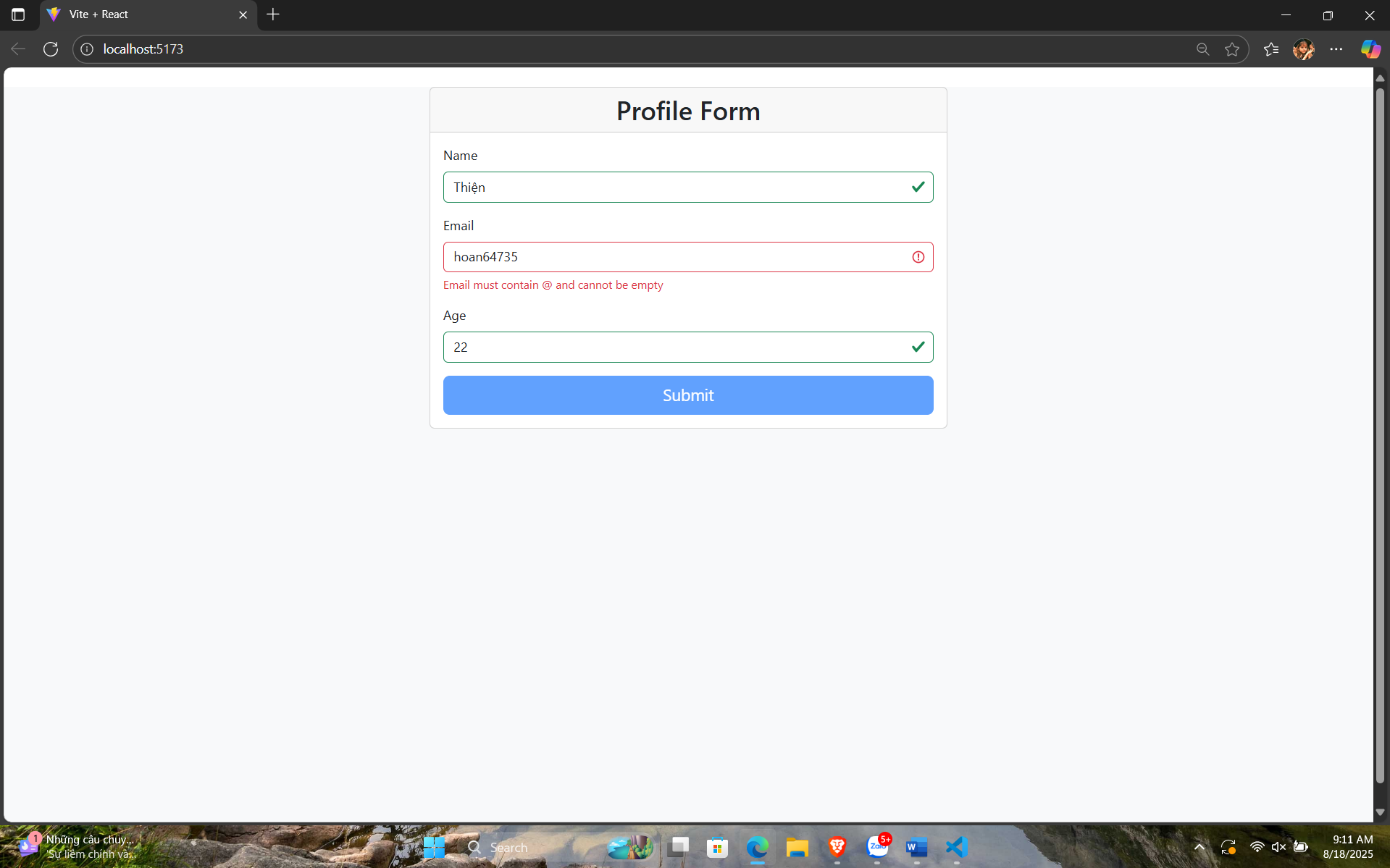
Slot11\_EX1  
  
  
ProfileForm.jsx  
import { useState } from 'react';

import PropTypes from 'prop-types';

import { Form, Button, Alert, Modal, Card, Container, Row, Col, Toast } from 'react-bootstrap';

const ProfileForm = ({ onSubmit }) => {

  // State 1: Lưu tên người dùng

  const [name, setName] = useState('');

  // State 2: Lưu email

  const [email, setEmail] = useState('');

  // State 3: Lưu tuổi

  const [age, setAge] = useState('');

  // State 4: Hiển thị thông báo toast

  const [showToast, setShowToast] = useState(false);

  // State 5: Hiển thị modal kết quả

  const [showModal, setShowModal] = useState(false);

  // Hàm validation cho name

  const validateName = (nameValue) => {

    return nameValue.trim() !== '';

  };

  // Hàm validation cho email

  const validateEmail = (emailValue) => {

    return emailValue.includes('@') && emailValue.trim() !== '';

  };

  // Hàm validation cho age

  const validateAge = (ageValue) => {

    const numAge = parseInt(ageValue);

    return !isNaN(numAge) && numAge >= 1;

  };

  // Hàm kiểm tra toàn bộ form có hợp lệ không

  const isFormValid = () => {

    return validateName(name) && validateEmail(email) && validateAge(age);

  };

  // Hàm xử lý khi người dùng nhập liệu

  const handleInputChange = (e) => {

    const { name: fieldName, value } = e.target;

    switch (fieldName) {

      case 'name':

        setName(value);

        break;

      case 'email':

        setEmail(value);

        break;

      case 'age':

        setAge(value);

        break;

      default:

        break;

    }

  };

  // Hàm xử lý khi submit form

  const handleSubmit = (e) => {

    e.preventDefault(); // Không reload trang

    if (isFormValid()) {

      // Hiển thị toast

      setShowToast(true);

      // Hiển thị modal

      setShowModal(true);

      // Gọi hàm onSubmit từ props

      onSubmit({ name, email, age });

      // Ẩn toast sau 3 giây

      setTimeout(() => {

        setShowToast(false);

      }, 3000);

    }

  };

  return (

    <Container className="mt-4">

      <Row className="justify-content-center">

        <Col md={6}>

          <Card>

            <Card.Header>

              <h2 className="text-center mb-0">Profile Form</h2>

            </Card.Header>

            <Card.Body>

              <Form onSubmit={handleSubmit}>

                {/\* Name Field \*/}

                <Form.Group className="mb-3">

                  <Form.Label>Name</Form.Label>

                  <Form.Control

                    type="text"

                    name="name"

                    placeholder="Enter your name"

                    value={name}

                    onChange={handleInputChange}

                    isInvalid={name !== '' && !validateName(name)}

                    isValid={name !== '' && validateName(name)}

                  />

                  <Form.Control.Feedback type="invalid">

                    Name cannot be empty

                  </Form.Control.Feedback>

                </Form.Group>

                {/\* Email Field \*/}

                <Form.Group className="mb-3">

                  <Form.Label>Email</Form.Label>

                  <Form.Control

                    type="email"

                    name="email"

                    placeholder="Enter your email"

                    value={email}

                    onChange={handleInputChange}

                    isInvalid={email !== '' && !validateEmail(email)}

                    isValid={email !== '' && validateEmail(email)}

                  />

                  <Form.Control.Feedback type="invalid">

                    Email must contain @ and cannot be empty

                  </Form.Control.Feedback>

                </Form.Group>

                {/\* Age Field \*/}

                <Form.Group className="mb-3">

                  <Form.Label>Age</Form.Label>

                  <Form.Control

                    type="number"

                    name="age"

                    placeholder="Enter your age"

                    value={age}

                    onChange={handleInputChange}

                    isInvalid={age !== '' && !validateAge(age)}

                    isValid={age !== '' && validateAge(age)}

                  />

                  <Form.Control.Feedback type="invalid">

                    Age must be at least 1

                  </Form.Control.Feedback>

                </Form.Group>

                {/\* Submit Button \*/}

                <div className="d-grid">

                  <Button

                    variant="primary"

                    type="submit"

                    disabled={!isFormValid()}

                    size="lg"

                  >

                    Submit

                  </Button>

                </div>

              </Form>

            </Card.Body>

          </Card>

        </Col>

      </Row>

      {/\* Toast Notification \*/}

      <Toast

        show={showToast}

        onClose={() => setShowToast(false)}

        style={{

          position: 'fixed',

          top: 20,

          right: 20,

          zIndex: 1050

        }}

        delay={3000}

        autohide

      >

        <Toast.Header>

          <strong className="me-auto">Success</strong>

        </Toast.Header>

        <Toast.Body>

          Submitted successfully!

        </Toast.Body>

      </Toast>

      {/\* Modal for Results \*/}

      <Modal show={showModal} onHide={() => setShowModal(false)} centered>

        <Modal.Header closeButton>

          <Modal.Title>Submission Successful!</Modal.Title>

        </Modal.Header>

        <Modal.Body>

          <Card>

            <Card.Header>

              <h5 className="mb-0">Your Information</h5>

            </Card.Header>

            <Card.Body>

              <Row className="mb-2">

                <Col sm={4}><strong>Name:</strong></Col>

                <Col sm={8}>{name}</Col>

              </Row>

              <Row className="mb-2">

                <Col sm={4}><strong>Email:</strong></Col>

                <Col sm={8}>{email}</Col>

              </Row>

              <Row>

                <Col sm={4}><strong>Age:</strong></Col>

                <Col sm={8}>{age}</Col>

              </Row>

            </Card.Body>

          </Card>

        </Modal.Body>

        <Modal.Footer>

          <Button variant="secondary" onClick={() => setShowModal(false)}>

            Close

          </Button>

        </Modal.Footer>

      </Modal>

    </Container>

  );

};

ProfileForm.propTypes = {

  onSubmit: PropTypes.func.isRequired,

};

export default ProfileForm;

App.jsx

import { useState } from 'react'

import ProfileForm from './components/ProfileForm'

import './App.css'

function App() {

  // Hàm xử lý khi form được submit

  const handleFormSubmit = (formData) => {

    console.log('Form data received:', formData);

    // Bạn có thể xử lý dữ liệu ở đây (gửi lên server, lưu vào database, etc.)

  };

  return (

    <div className="App">

      <ProfileForm onSubmit={handleFormSubmit} />

    </div>

  )

}

export default App