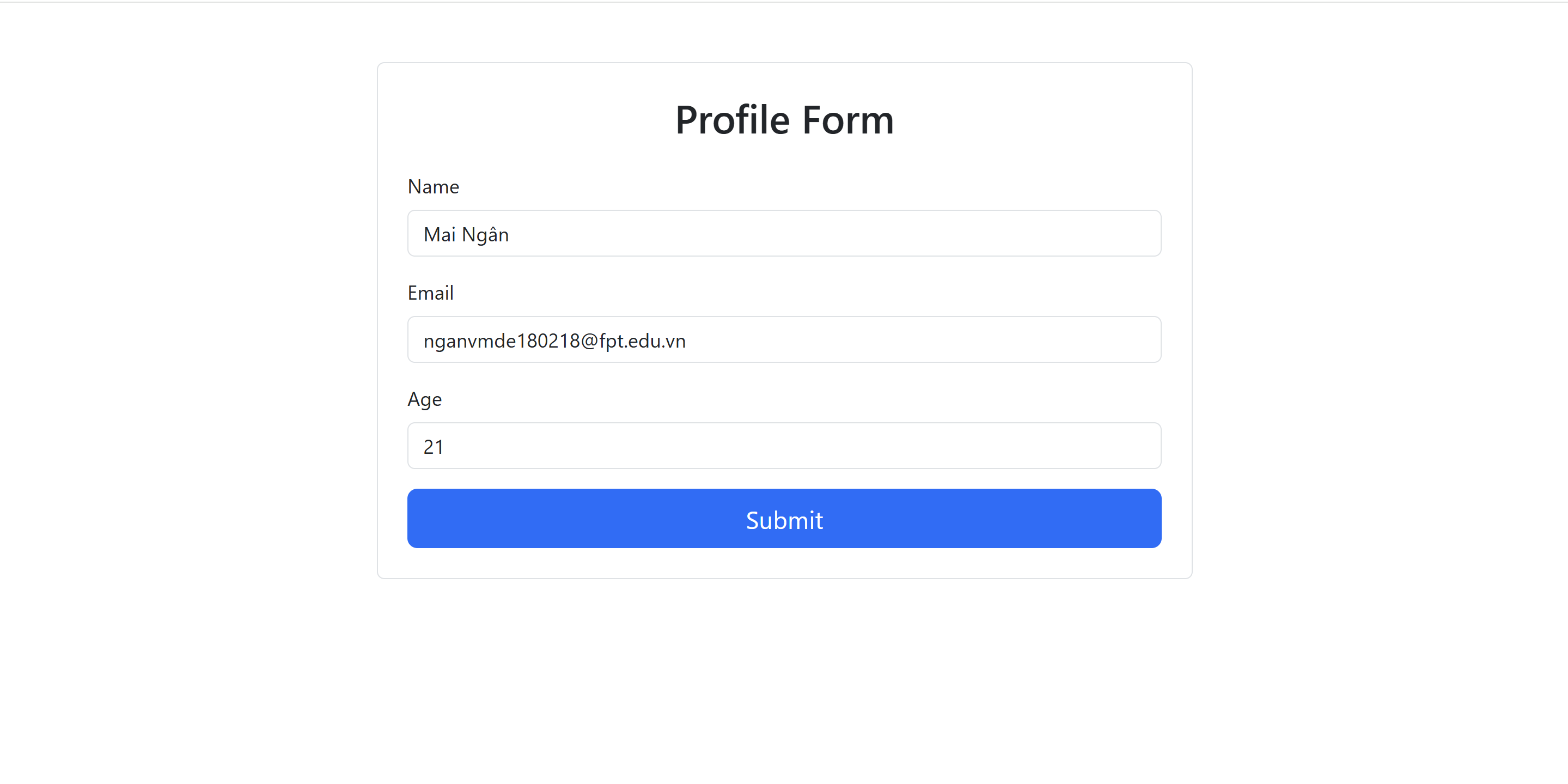
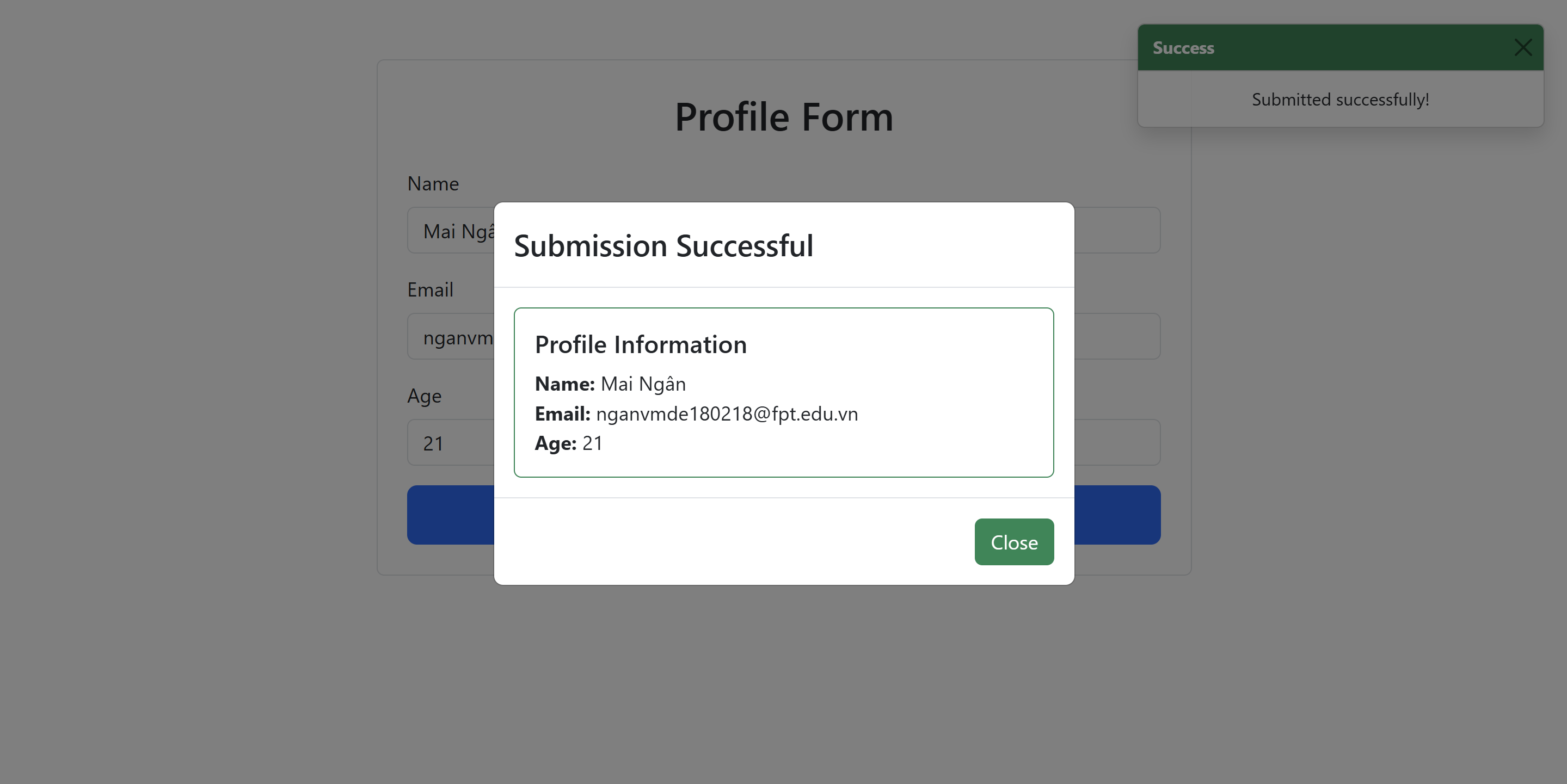
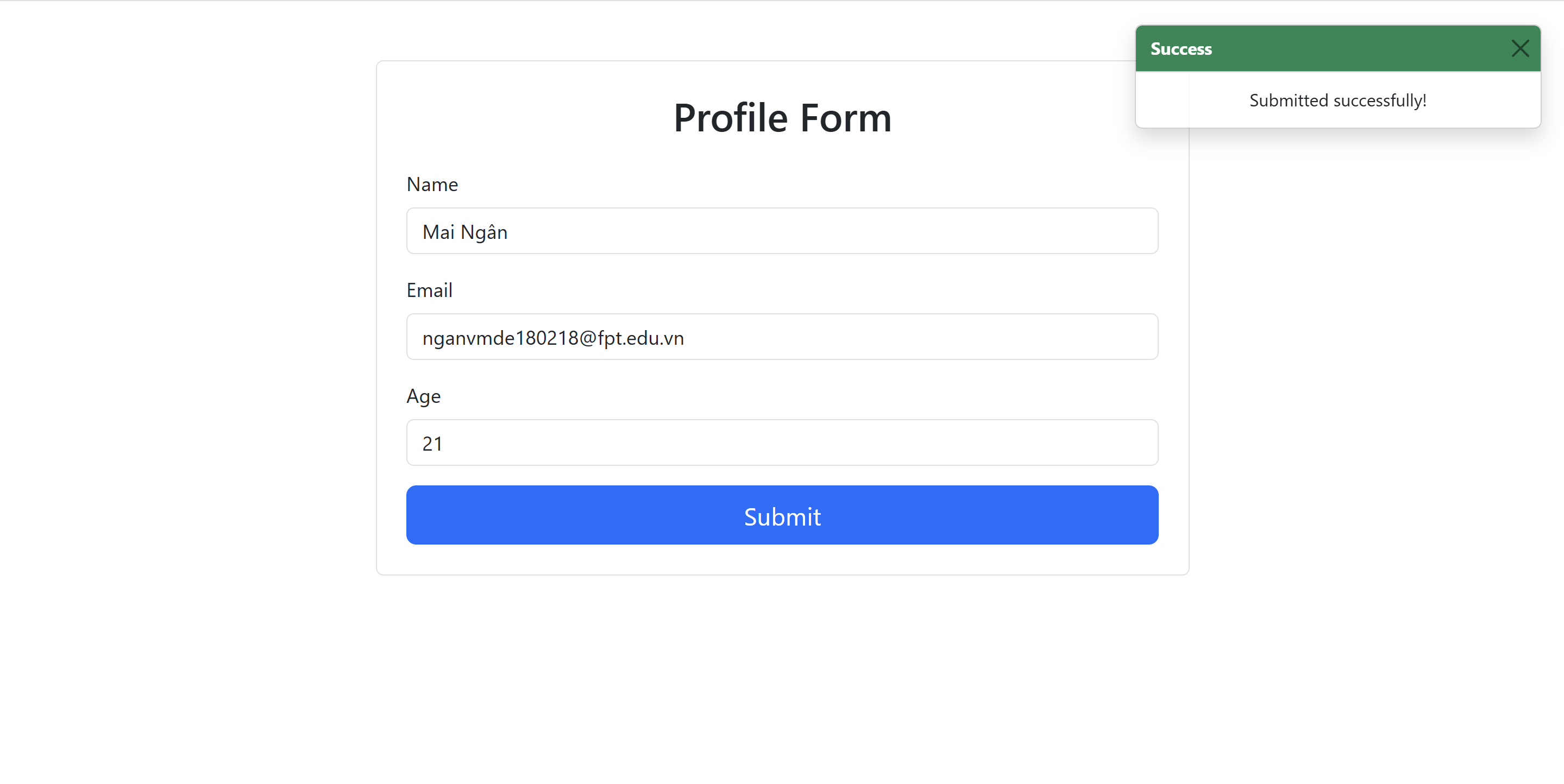
# Slot 11

## Exercise 1:







## ProfileForm.js

import React, { useState } from 'react';

import PropTypes from 'prop-types';

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

const ProfileForm = ({ onSubmit }) => {

  // State for form fields

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

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

  const [age, setAge] = useState('');

  // State for validation errors

  const [errors, setErrors] = useState({

    name: '',

    email: '',

    age: ''

  });

  // State for toast and modal

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

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

  // Validation functions

  const validateName = (name) => {

    if (!name.trim()) return 'Name is required';

    return '';

  };

  const validateEmail = (email) => {

    if (!email.trim()) return 'Email is required';

    if (!email.includes('@')) return 'Email must contain @';

    return '';

  };

  const validateAge = (age) => {

    if (!age) return 'Age is required';

    if (isNaN(age) || Number(age) < 1) return 'Age must be at least 1';

    return '';

  };

  // Handle form field changes

  const handleChange = (e) => {

    const { name, value } = e.target;

    switch (name) {

      case 'name':

        setName(value);

        setErrors({...errors, name: validateName(value)});

        break;

      case 'email':

        setEmail(value);

        setErrors({...errors, email: validateEmail(value)});

        break;

      case 'age':

        setAge(value);

        setErrors({...errors, age: validateAge(value)});

        break;

      default:

        break;

    }

  };

  // Check if form is valid

  const isFormValid = () => {

    return (

      name.trim() &&

      email.trim() &&

      email.includes('@') &&

      age &&

      !isNaN(age) &&

      Number(age) >= 1 &&

      !errors.name &&

      !errors.email &&

      !errors.age

    );

  };

  // Handle form submission

  const handleSubmit = (e) => {

    e.preventDefault();

    // Final validation

    const newErrors = {

      name: validateName(name),

      email: validateEmail(email),

      age: validateAge(age)

    };

    setErrors(newErrors);

    if (isFormValid()) {

      // Call onSubmit prop if provided

      if (onSubmit) {

        onSubmit({ name, email, age: Number(age) });

      }

      // Show toast and modal

      setShowToast(true);

      setShowModal(true);

    }

  };

  return (

    <Container className="mt-5 p-4 border rounded" style={{ maxWidth: '700px'}}>

      {/\* Success Toast - Đã đưa lên đầu \*/}

      <div style={{ position: 'fixed', top: '20px', right: '20px', zIndex: 1000 }}>

        <Toast

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

          show={showToast}

          delay={3000}

          autohide

        >

          <Toast.Header className="bg-success text-white">

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

          </Toast.Header>

          <Toast.Body>Submitted successfully!</Toast.Body>

        </Toast>

      </div>

      <h2 className="mb-4 text-center">Profile Form</h2>

      <Form onSubmit={handleSubmit}>

        {/\* Name Field - Label căn trái \*/}

        <Form.Group className="mb-3">

          <Form.Label className="d-flex justify-content-start">Name</Form.Label>

          <Form.Control

            type="text"

            name="name"

            value={name}

            onChange={handleChange}

            placeholder="Enter your name"

            isInvalid={!!errors.name}

          />

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

            {errors.name}

          </Form.Control.Feedback>

        </Form.Group>

        {/\* Email Field - Label căn trái \*/}

        <Form.Group className="mb-3">

          <Form.Label className="d-flex justify-content-start">Email</Form.Label>

          <Form.Control

            type="email"

            name="email"

            value={email}

            onChange={handleChange}

            placeholder="Enter your email"

            isInvalid={!!errors.email}

          />

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

            {errors.email}

          </Form.Control.Feedback>

        </Form.Group>

        {/\* Age Field - Label căn trái \*/}

        <Form.Group className="mb-3">

          <Form.Label className="d-flex justify-content-start">Age</Form.Label>

          <Form.Control

            type="number"

            name="age"

            value={age}

            onChange={handleChange}

            placeholder="Enter your age"

            min="1"

            isInvalid={!!errors.age}

          />

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

            {errors.age}

          </Form.Control.Feedback>

        </Form.Group>

        {/\* Submit Button \*/}

        <div className="d-grid">

          <Button

            variant="primary"

            type="submit"

            disabled={!isFormValid()}

            size="lg"

          >

            Submit

          </Button>

        </div>

      </Form>

      {/\* Success Modal \*/}

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

        <Modal.Header>

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

        </Modal.Header>

        <Modal.Body>

          <Card border="success">

            <Card.Body>

              <Card.Title>Profile Information</Card.Title>

              <Card.Text>

                <strong>Name:</strong> {name}<br />

                <strong>Email:</strong> {email}<br />

                <strong>Age:</strong> {age}

              </Card.Text>

            </Card.Body>

          </Card>

        </Modal.Body>

        <Modal.Footer>

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

            Close

          </Button>

        </Modal.Footer>

      </Modal>

    </Container>

  );

};

// PropTypes validation

ProfileForm.propTypes = {

  onSubmit: PropTypes.func

};

export default ProfileForm;

## App.js

import React from 'react';

import ProfileForm from './ProfileForm';

import './App.css';

function App() {

  return (

    <div className="App">

      <ProfileForm />

    </div>

  );

}

export default App;