LAB 5

QuizApp

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

import { Card, ListGroup, Button } from 'react-bootstrap';

import { useSelectedAnswer } from './SelectedAnswerContext';

const initialState = {

localSelectedAnswer: '',

showCorrectness: false,

}

const reducer = (state, action) => {

switch (action.type) {

case 'SetLocalSelectedAnswer':

return { ...state, localSelectedAnswer: action.payload, showCorrectness: false }

case 'ShowCorrect':

return { ...state, showCorrectness: true }

case 'Reset':

return initialState

default:

return state

}

}

const Question = ({ questionData, onNextQuestion }) => {

const { question, options, answer } = questionData

const { setSelectedAnswer } = useSelectedAnswer()

const [state, dispatch] = useReducer(reducer, initialState)

const { localSelectedAnswer, showCorrectness } = state

useEffect(() => {

dispatch({ type: 'Reset' })

setSelectedAnswer('')

}, [questionData, setSelectedAnswer])

const handleAnswerSelect = (option) => {

dispatch({ type: 'SetLocalSelectedAnswer', payload: option })

setSelectedAnswer(option)

}

const handleNextQuestion = () => {

if (localSelectedAnswer) {

dispatch({ type: 'ShowCorrect' })

if (localSelectedAnswer === answer) {

setTimeout(() => {

onNextQuestion()

}, 1000) // delay để xem đáp án đúng hay sai

} else {

setTimeout(() => {

onNextQuestion()

}, 1000)

}

}

}

return (

<Card className="mt-2">

<Card.Body>

<Card.Title>{question}</Card.Title>

<ListGroup>

{options.map((option, index) => (

// Hiện đáp án đúng sai sau khi bấm next question

<ListGroup.Item

key={index}

variant={

showCorrectness

? option === answer

? 'success'

: option === localSelectedAnswer

? 'danger'

: ''

: localSelectedAnswer === option

? 'success'

: ''

}

onClick={() => handleAnswerSelect(option)}

style={{ cursor: 'pointer' }}

>

{option}

{showCorrectness && localSelectedAnswer === option && (

<span className="ml-2">

{localSelectedAnswer === answer ? ' - Correct' : ' - Incorrect'}

</span>

)}

</ListGroup.Item>

))}

</ListGroup>

<Button className="mt-3" variant="primary" onClick={handleNextQuestion} disabled={!localSelectedAnswer}>Next Question</Button>

</Card.Body>

</Card>

)

}

export default Question

QuestionsData.js

const Questions = [

{

id: 1,

question: 'What is the capital of France?',

options: ['Paris', 'London', 'Berlin', 'Madrid'],

answer: 'Paris'

},

{

id: 2,

question: 'What is the largest planet in our solar system?',

options: ['Jupiter', 'Saturn', 'Mars', 'Earth'],

answer: 'Jupiter'

},

{

id: 3,

question: 'What day is Vietnam National Day?',

options: ['02/09/1940', '02/09/1945', '03/09/1975', '02/09/1946'],

answer: '02/09/1945'

},

{

id: 4,

question: "The growth rate of the Russian Federation's economy from 1991 to 1995 fell into a state of",

options: ['Always a negative number', 'Underdeveloped', 'No development', 'Stagnation, slow development'],

answer: 'Always a negative number'

},

{

id: 5,

question: 'About 20 years after World War II, America became',

options: ["The world's largest economic and financial center", 'The first country to successfully launch an artificial satellite', 'One of three major economic and financial centers of the world', "The world's second economic and financial center"],

answer: "The world's largest economic and financial center"

},

{

id: 6,

question: 'What is ReactJS?',

options: ['A JavaScript library for building user interfaces', 'A programming language', 'A database management system'],

answer: "A JavaScript library for building user interfaces"

},

{

id: 7,

question: 'What is JSX?',

options: ['A programming language', 'A file format', 'A syntax extension for JavaScript'],

answer: 'A syntax extension for JavaScript'

},

{

id: 8,

question: 'Which of the following commands is used to create a new branch in Git?',

options: ['git branch new\_branch', 'git checkout new\_branch', 'git commit -m "new\_branch"', 'git merge new\_branch'],

answer: 'git branch new\_branch'

},

{

id: 9,

question: 'What command is used to initialize a new Git repository?',

options: ['git start', 'git init', 'git create', 'git new'],

answer: 'git init'

},

{

id: 10,

question: 'What command is used to stage changes for the next commit?',

options: ['git commit -m "message"', 'git stage', 'git add .', 'git update'],

answer: 'git add .'

},

]

export default Questions

QuizApp.js

import React, { useState } from 'react';

import 'bootstrap/dist/css/bootstrap.min.css';

import { Container, Card } from 'react-bootstrap';

import Question from './Question';

import Score from './Score';

import TimeCounter from './TimeCounter';

import Start from './Start';

import { useSelectedAnswer } from './SelectedAnswerContext';

import Questions from './QuestionsData';

const QuizApp = ({ timerDuration = 20 }) => {

const [currentQuestion, setCurrentQuestion] = useState(0)

const [score, setScore] = useState(0)

const [quizEnd, setQuizEnd] = useState(false)

const [quizStarted, setQuizStarted] = useState(false)

const { selectedAnswer, setSelectedAnswer } = useSelectedAnswer()

const handleNextQuestion = () => {

if (selectedAnswer === Questions[currentQuestion].answer) {

setScore(score + 1)

}

setSelectedAnswer('')

if (currentQuestion < Questions.length - 1) {

setCurrentQuestion(currentQuestion + 1)

} else {

setQuizEnd(true)

}

}

const handleTimeUp = () => {

handleNextQuestion()

}

const handleStartQuiz = () => {

setQuizStarted(true)

}

return (

<Container className='mt-5'>

{!quizStarted ? (

<Start onStart={handleStartQuiz} />

) : !quizEnd ? (

<>

<Card className='mb-4'>

<Card.Body>

<Card.Title as="h2" className='text-center'>Quiz App</Card.Title>

<TimeCounter

className='mb-5'

currentQuestion={currentQuestion}

onTimeUp={handleTimeUp}

duration={timerDuration}

/>

<Question questionData={Questions[currentQuestion]} onNextQuestion={handleNextQuestion} />

</Card.Body>

</Card>

</>

) : (

<Card className='mb-0'>

<Card.Body>

<Score score={score} totalQuestions={Questions.length} />

</Card.Body>

</Card>

)}

</Container>

)

}

export default QuizApp

Score.js

import React, { Component } from 'react'

import { Card, Button } from 'react-bootstrap'

class Score extends Component {

handleReplay = () => {

window.location.reload()

}

render() {

const { score, totalQuestions } = this.props

return (

// <Card className="text-center">

// <Card.Body>

// </Card.Body>

// </Card>

<div className='text-center'>

<Card.Title as="h2">Congratulations on completing the quiz!</Card.Title>

<Card.Title>Your Score: {score} / {totalQuestions}</Card.Title>

<Button variant="warning" onClick={this.handleReplay}>Play Again</Button>

</div>

)

}

}

export default Score

SelectedAnswerContext.js

import React, { createContext, useContext, useState } from 'react'

const SelectedAnswerContext = createContext()

export const SelectedAnswerProvider = ({ children }) => {

const [selectedAnswer, setSelectedAnswer] = useState('')

return (

<SelectedAnswerContext.Provider value={{ selectedAnswer, setSelectedAnswer }}>

{children}

</SelectedAnswerContext.Provider>

)

}

export const useSelectedAnswer = () => useContext(SelectedAnswerContext)

Start.js

import React, { useState } from 'react'

import { Button, Card } from 'react-bootstrap'

const Start = ({ onStart }) => {

const [quizStarted, setQuizStarted] = useState(false)

const handleStartClick = () => {

setQuizStarted(true)

onStart()

}

return (

<Card className='text-center'>

<Card.Body>

<Card.Title as="h2">Welcome to NhatThach's the Quiz App</Card.Title>

<Card.Text>You can press the button below to start this quiz</Card.Text>

<Button variant="primary" onClick={handleStartClick} disabled={quizStarted}>

Start

</Button>

</Card.Body>

</Card>

)

}

export default Start

TimeCounter.js

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

const TimeCounter = ({ duration, currentQuestion, onTimeUp }) => {

const [timeLeft, setTimeLeft] = useState(duration)

useEffect(() => {

const timer = setInterval(() => {

if (timeLeft > 0) {

setTimeLeft(timeLeft - 1)

} else {

onTimeUp()

}

}, 1000)

return () => clearInterval(timer)

}, [duration, onTimeUp, timeLeft])

useEffect(() => {

// Reset the timer when the currentQuestion changes

setTimeLeft(duration);

}, [currentQuestion, duration])

return (

<div className='d-flex'>

<h5>Time Left: </h5>

<h5 className='text-danger mx-2'>{timeLeft}s</h5>

</div>

)

}

export default TimeCounter

About.js

import React from 'react'

import { Container, Row, Col, Card, ListGroup, ListGroupItem } from 'react-bootstrap'

const About = () => {

return (

<Container>

<Row className="my-5">

<Col>

<h2 className="my-5 text-danger">About Us</h2>

<p>helloj</p>

<p>xin chao</p>

</Col>

</Row>

<Row>

<Col md={6}>

<Card>

<Card.Body>

<Card.Title>About Us</Card.Title>

<ListGroup variant="flush">

<ListGroupItem><strong>ABC</strong> - Founder & CEO</ListGroupItem>

<ListGroupItem><strong>ABC</strong> - CTO</ListGroupItem>

<ListGroupItem><strong>ABC</strong> - Lead Engineer</ListGroupItem>

<ListGroupItem><strong>ABC</strong> - Designer</ListGroupItem>

</ListGroup>

</Card.Body>

</Card>

</Col>

<Col md={6}>

<Card>

<Card.Body>

<Card.Title>Contact Us</Card.Title>

<Card.Text>

If you have any questions or feedback, feel free to reach out to us at <a href="mailto:lebaochau@gmail.com">lebaochau@gmail.com</a>.

</Card.Text>

</Card.Body>

</Card>

</Col>

</Row>

</Container>

)

}

export default About

Contact.js

import React, { useState } from 'react'

import { Form, Button, Container, Row, Col } from 'react-bootstrap'

const Contact = () => {

const [formData, setFormData] = useState({

firstName: '',

lastName: '',

username: '',

city: '',

state: '',

zip: '',

terms: false,

})

const [validated, setValidated] = useState(false)

const [showTermsError, setShowTermsError] = useState(false)

const handleChange = (e) => {

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

setFormData({

...formData,

[name]: type === 'checkbox' ? checked : value,

})

}

const handleSubmit = (event) => {

const form = event.currentTarget

if (form.checkValidity() === false || !formData.terms) {

event.preventDefault();

event.stopPropagation();

if (!formData.terms) {

setShowTermsError(true)

}

} else {

setShowTermsError(false)

}

setValidated(true)

}

return (

<Container className='mt-5'>

<h2 className="my-5 text-danger">Contact</h2>

<Form noValidate validated={validated} onSubmit={handleSubmit}>

<Row>

<Col md="4">

<Form.Group controlId="validationCustom01">

<Form.Label>First name</Form.Label>

<Form.Control

required

type="text"

placeholder="First name"

name="firstName"

value={formData.firstName}

onChange={handleChange}

/>

<Form.Control.Feedback>Looks good!</Form.Control.Feedback>

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

Please provide your First name.

</Form.Control.Feedback>

</Form.Group>

</Col>

<Col md="4">

<Form.Group controlId="validationCustom02">

<Form.Label>Last name</Form.Label>

<Form.Control

required

type="text"

placeholder="Last name"

name="lastName"

value={formData.lastName}

onChange={handleChange}

/>

<Form.Control.Feedback>Ok!</Form.Control.Feedback>

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

Please provide your Last name.

</Form.Control.Feedback>

</Form.Group>

</Col>

<Col md="4">

<Form.Group controlId="validationCustomUsername">

<Form.Label>Username</Form.Label>

<Form.Control

required

type="text"

placeholder="Username"

name="username"

value={formData.username}

onChange={handleChange}

/>

<Form.Control.Feedback>Looks good!</Form.Control.Feedback>

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

Please choose a username.

</Form.Control.Feedback>

</Form.Group>

</Col>

</Row>

<Row className='mt-3'>

<Col md="6">

<Form.Group controlId="validationCustom03">

<Form.Label>City</Form.Label>

<Form.Control

required

type="text"

placeholder="City"

name="city"

value={formData.city}

onChange={handleChange}

/>

<Form.Control.Feedback>Ok!</Form.Control.Feedback>

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

Please provide a valid city.

</Form.Control.Feedback>

</Form.Group>

</Col>

<Col md="3">

<Form.Group controlId="validationCustom04">

<Form.Label>State</Form.Label>

<Form.Control

required

type="text"

placeholder="State"

name="state"

value={formData.state}

onChange={handleChange}

/>

<Form.Control.Feedback>Ok!</Form.Control.Feedback>

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

Please provide a valid state.

</Form.Control.Feedback>

</Form.Group>

</Col>

<Col md="3">

<Form.Group controlId="validationCustom05">

<Form.Label>Zip</Form.Label>

<Form.Control

required

type="text"

placeholder="Zip"

name="zip"

value={formData.zip}

onChange={handleChange}

/>

<Form.Control.Feedback>Ok!</Form.Control.Feedback>

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

Please provide a valid zip.

</Form.Control.Feedback>

</Form.Group>

</Col>

</Row>

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

<Form.Check

required

label="Agree to terms and conditions"

feedbackType="invalid"

name="terms"

checked={formData.terms}

onChange={handleChange}

/>

{showTermsError && (

<div className="invalid-feedback" style={{ display: 'block' }}>

You must agree before submitting.

</div>

)}

</Form.Group>

<Button type="submit" className="mt-3">Submit form</Button>

</Form>

</Container>

)

}

export default Contact

Home.js

import React from 'react';

import { Carousel, Image, Container, Row, Col } from 'react-bootstrap';

import slideImg1 from '../images/slide1.jpg';

import slideImg2 from '../images/slide2.jpg';

import slideImg3 from '../images/slide3.jpg';

import pic1 from '../images/menu-01.jpg';

import pic2 from '../images/menu-02.jpg';

import pic3 from '../images/menu-03.jpg';

import pic4 from '../images/menu-04.jpg';

import pic5 from '../images/menu-05.jpg';

import pic6 from '../images/menu-06.jpg';

const Home = () => {

return (

<div>

<Carousel id="carouselExampleCaptions" className="mt-1" variant="light" interval={2000}>

<Carousel.Item>

<Image src={slideImg1} className="d-block w-100" alt="First slide" />

<Carousel.Caption>

<h5>First slide label</h5>

<p>Some representative placeholder content for the first slide.</p>

</Carousel.Caption>

</Carousel.Item>

<Carousel.Item>

<Image src={slideImg2} className="d-block w-100" alt="First slide" />

<Carousel.Caption>

<h5>Second slide label</h5>

<p>Some representative placeholder content for the second slide.</p>

</Carousel.Caption>

</Carousel.Item>

<Carousel.Item>

<Image src={slideImg3} className="d-block w-100" alt="First slide" />

<Carousel.Caption>

<h5>Third slide label</h5>

<p>Some representative placeholder content for the third slide.</p>

</Carousel.Caption>

</Carousel.Item>

</Carousel>

<Container className='mt-3'>

<Row>

<Col>

<Image src={pic1} roundedCircle />

</Col>

<Col>

<Image src={pic2} roundedCircle />

</Col>

<Col>

<Image src={pic3} roundedCircle />

</Col>

<Col>

<Image src={pic4} roundedCircle />

</Col>

<Col>

<Image src={pic5} roundedCircle />

</Col>

<Col>

<Image src={pic6} roundedCircle />

</Col>

</Row>

<h2 className='text-danger mt-3'>This is home page</h2>

</Container>

</div>

);

}

export default Home;

NewCard.js

import React from 'react';

import { Card } from 'react-bootstrap';

const NewsCard = ({ news }) => {

return (

<Card className='h-100'>

<Card.Img variant="top" src={news.images} />

<Card.Body>

<Card.Title>{news.title}</Card.Title>

<Card.Text>{news.description}</Card.Text>

</Card.Body>

</Card>

);

};

export default NewsCard;

News.js

import React, { Suspense, lazy } from 'react'

import { Container, Row, Col, Card } from 'react-bootstrap'

import pic1 from '../images/event-1.jpg'

import pic2 from '../images/event-2.jpg'

import pic3 from '../images/event-3.jpg'

import pic4 from '../images/event-4.jpg'

import pic5 from '../images/event-5.jpg'

import pic6 from '../images/event-6.jpg'

import pic7 from '../images/event-7.jpg'

import pic8 from '../images/event-8.jpg'

// import NewCard from './NewCard'

const delay = (ms) => new Promise((resolve) => setTimeout(resolve, ms));

const loadComponent = () => {

return delay(3000).then(() => import('./NewCard'));

};

const NewCard = lazy(loadComponent)

const newsLists = [

{ id: 1, title: 'Woman bakes expletive-laden pies to ‘get a rise’ out of her grandmother in annual tradition', description: '“What started as a means to get a rise out of my Grammy has snowballed into a weird family tradition,” wrote Jess Lydon.', images: pic1 },

{ id: 2, title: 'Martha Stewart shows off her 30 pies after canceled Thanksgiving dinner plans', description: 'Queen of Thanksgiving Martha Stewart may not be hosting a turkey dinner this year but fret not, she will still be celebrating with literally 30 pies.', images: pic2 },

{ id: 3, title: 'Burger King is testing a new breakfast sandwich', description: 'This is a win for the flatbread fans.', images: pic3 },

{ id: 4, title: 'Popeyes permanently adds chicken wings to its menu', description: 'And you can get ’em in five different flavors.', images: pic4 },

{ id: 5, title: 'Top salmon with a sizzling mix of aromatics and spices', description: 'Tadka is a ubiquitous South Asian technique that adds a dramatic last-minute coat of flavor.', images: pic5 },

{ id: 6, title: '80 Christmas dinner ideas for the ultimate holiday feast', description: 'Build the perfect Christmas menu with these delicious recipes.', images: pic6 },

{ id: 7, title: 'How to make the easiest prime rib roast for the holidays', description: 'Use these tips and tricks to make a juicy and amazingly delicious prime rib roast.', images: pic7 },

{ id: 8, title: 'Turn leftover turkey into a flavorful Waldorf salad', description: 'This light, bright turkey salad is the best post-Thanksgiving lunch.', images: pic8 },

]

const News = () => {

return (

<Container>

<h2 className="my-5 text-danger">News Category</h2>

<Suspense fallback={<div>Loading...</div>}>

<Row>

{newsLists.map(news => (

<Col key={news.id} md={4} className="mb-4">

{/\* <Card className='h-100'>

<Card.Img variant="top" src={news.images} />

<Card.Body>

<Card.Title>{news.title}</Card.Title>

<Card.Text>{news.description}</Card.Text>

</Card.Body>

</Card> \*/}

<NewCard news={news} />

</Col>

))}

</Row>

</Suspense>

</Container>

)

}

export default News

Quiz.js

import React, { useState } from 'react'

import { SelectedAnswerProvider } from './quizApp/SelectedAnswerContext'

import QuizApp from './quizApp/QuizApp'

const Quiz = () => {

return (

<SelectedAnswerProvider>

<QuizApp timerDuration={15} />

</SelectedAnswerProvider>

)

}

export default Quiz

App.js

import React from 'react';

import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';

import { Navbar, Nav } from 'react-bootstrap';

import Home from './Components/Home';

import News from './Components/News';

import Quiz from './Components/Quiz';

import About from './Components/About';

import Contact from './Components/Contact';

import 'bootstrap/dist/css/bootstrap.min.css';

function App() {

return (

<Router>

<Navbar bg="dark" variant="dark" expand="md">

<Navbar.Toggle aria-controls="basic-navbar-nav" />

<Navbar.Collapse id="basic-navbar-nav">

<Nav className="mr-auto mx-5">

<Nav.Link as={Link} to="/">Home</Nav.Link>

<Nav.Link as={Link} to="/about">About</Nav.Link>

<Nav.Link as={Link} to="/contact">Contact</Nav.Link>

<Nav.Link as={Link} to="/quiz">Quiz</Nav.Link>

<Nav.Link as={Link} to="/news">News</Nav.Link>

</Nav>

</Navbar.Collapse>

</Navbar>

<Routes>

<Route path="/" element={<Home />} />

<Route path="/news" element={<News />} />

<Route path="/quiz" element={<Quiz />} />

<Route path="/about" element={<About />} />

<Route path="/contact" element={<Contact />} />

</Routes>

</Router>

);

}

export default App;