



import *React*, { *useState*, *useEffect*, *createContext*, *useContext* } from 'react';  
import { *Container*, Form, *Button*, Card, ListGroup, Alert } from 'react-bootstrap';  
import 'bootstrap/dist/css/bootstrap.min.css';  
  
const QuizContext = *createContext*();  
  
export const quizData = [  
 {  
 question: 'What is ReactJS?',  
 answers: ['A JavaScript library for building user interfaces', 'A programming language', 'A database management system'],  
 correctAnswer: 'A JavaScript library for building user interfaces'  
 },  
 {  
 question: 'What is JSX?',  
 answers: ['A programming language', 'A file format', 'A syntax extension for JavaScript'],  
 correctAnswer: 'A syntax extension for JavaScript'  
 }  
];  
  
  
function *Quizz*() {  
 const [questions, setQuestions] = *useState*(quizData);  
 const [userAnswers, setUserAnswers] = *useState*({});  
 const [results, setResults] = *useState*({});  
 const [newQuestion, setNewQuestion] = *useState*('');  
 const [newAnswers, setNewAnswers] = *useState*(['', '', '']);  
 const [newCorrectAnswer, setNewCorrectAnswer] = *useState*('');  
 const [score, setScore] = *useState*(null);  
 const [quizCompleted, setQuizCompleted] = *useState*(false);  
  
 *useEffect*(() => {  
 setQuestions(quizData);  
 }, []);  
  
 const handleAddQuestion = () => {  
 if (newQuestion && newAnswers.every(ans => ans) && newCorrectAnswer) {  
 setQuestions([...questions, {  
 question: newQuestion,  
 answers: newAnswers,  
 correctAnswer: newCorrectAnswer  
 }]);  
 setNewQuestion('');  
 setNewAnswers(['', '', '']);  
 setNewCorrectAnswer('');  
 }  
 };  
  
 const checkAnswers = () => {  
 const newResults = {};  
 let correctCount = 0;  
 questions.forEach((q, index) => {  
 const isCorrect = userAnswers[index] === q.correctAnswer;  
 newResults[index] = isCorrect ? 'Đúng' : 'Sai';  
 if (isCorrect) correctCount++;  
 });  
 setResults(newResults);  
 setScore(correctCount);  
 setQuizCompleted(true);  
 };  
  
 const resetQuiz = () => {  
 setUserAnswers({});  
 setResults({});  
 setScore(null);  
 setQuizCompleted(false);  
 };  
  
 return (  
 <QuizContext.Provider value={{ userAnswers, setUserAnswers }}>  
 <Container className="mt-4" style={{ maxWidth: '600px' }}>  
 <h2 className="text-center mb-4">Thêm câu hỏi mới</h2>  
 <Form>  
 <Form.Group className="mb-3">  
 <Form.Control  
 type="text"  
 value={newQuestion}  
 onChange={(e) => setNewQuestion(e.target.value)}  
 placeholder="Nhập câu hỏi"  
 />  
 </Form.Group>  
 {newAnswers.map((answer, index) => (  
 <Form.Group className="mb-3" key={index}>  
 <Form.Control  
 type="text"  
 value={answer}  
 onChange={(e) => {  
 const updatedAnswers = [...newAnswers];  
 updatedAnswers[index] = e.target.value;  
 setNewAnswers(updatedAnswers);  
 }}  
 placeholder={`Đáp án ${index + 1}`}  
 />  
 </Form.Group>  
 ))}  
 <Form.Group className="mb-3">  
 <Form.Control  
 type="text"  
 value={newCorrectAnswer}  
 onChange={(e) => setNewCorrectAnswer(e.target.value)}  
 placeholder="Đáp án đúng"  
 />  
 </Form.Group>  
 <Button variant="primary" onClick={handleAddQuestion} className="w-100">  
 Thêm câu hỏi  
 </Button>  
 </Form>  
  
 <h2 className="text-center mt-5 mb-4">Danh sách câu hỏi</h2>  
 {questions.map((q, index) => (  
 <Question key={index} index={index} question={q} />  
 ))}  
  
 <Button variant="success" onClick={checkAnswers} className="w-100 mt-3">  
 Kiểm tra đáp án  
 </Button>  
  
 {quizCompleted && (  
 <Alert variant="danger" className="mt-3 text-center">  
 <h2>Quiz Completed!</h2>  
 <p>Your score: {score} / {questions.length}</p>  
 <Button variant="danger" onClick={resetQuiz}>New</Button>  
 </Alert>  
 )}  
  
 {Object.keys(results).map((index) => (  
 <p  
 key={index}  
 className={`mt-3 text-center ${results[index] === 'Đúng' ? 'text-success' : 'text-danger'}`}  
 >  
 Câu {*parseInt*(index) + 1}: {results[index]}  
 </p>  
 ))}  
 </Container>  
 </QuizContext.Provider>  
 );  
}  
  
// Component con để hiển thị câu hỏi và đáp án  
function Question({ index, question }) {  
 const { userAnswers, setUserAnswers } = *useContext*(QuizContext);  
  
 return (  
 <Card className="mb-3">  
 <Card.Body>  
 <Card.Title>{index + 1}. {question.question}</Card.Title>  
 <ListGroup>s  
 {question.answers.map((answer, ansIndex) => (  
 <ListGroup.Item key={ansIndex}>  
 <Form.Check  
 type="radio"  
 name={`question-${index}`}  
 value={answer}  
 label={answer}  
 onChange={() => {  
 setUserAnswers({ ...userAnswers, [index]: answer });  
 }}  
 checked={userAnswers[index] === answer}  
 />  
 </ListGroup.Item>  
 ))}  
 </ListGroup>  
 </Card.Body>  
 </Card>  
 );  
}  
  
export default *Quizz*;