

```
src\pages\Quiz.jsx

1 import { useEffect, useState } from 'react';
2 import { useNavigate } from 'react-router-dom';
3 import { useQuiz } from '../context/QuizContext';
4 import ProgressBar from '../components/ProgressBar';

5
6 const QUESTION_TIME = 20; // seconds per question
7

8 function Quiz() {
9     const {
10         currentQuestion,
11         currentIndex,
12         totalQuestions,
13         submitAnswer,
14         goToNextQuestion,
15         finishQuiz,
16         quizStarted,
17     } = useQuiz();
18

19     const navigate = useNavigate();
20     const [timeLeft, setTimeLeft] = useState(QUESTION_TIME);
21     const [selected, setSelected] = useState(null);
22     const [locked, setLocked] = useState(false); // disable buttons after answer or timer end
23

24     // Redirect if quiz not started
25     useEffect(() => {
26         if (!quizStarted || !currentQuestion) {
27             navigate('/');
28         }
29     }, [quizStarted, currentQuestion, navigate]);
30

31     // Reset state when question changes
32     useEffect(() => {
33         setTimeLeft(QUESTION_TIME);
34         setSelected(null);
35         setLocked(false);
36     }, [currentQuestion]);
37

38     // Timer countdown
39     useEffect(() => {
40         if (!currentQuestion || locked) return;
41

42         if (timeLeft <= 0) {
43             setLocked(true);
44             const timeout = setTimeout(() => handleNext(), 700);
45             return () => clearTimeout(timeout);
46         }
47

48         const timer = setTimeout(() => setTimeLeft((t) => t - 1), 1000);

```

```
49     return () => clearTimeout(timer);
50 }, [timeLeft, locked, currentQuestion]);
51
52 if (!currentQuestion) return null;
53
54 function handleSelect(optionIndex) {
55     if (locked) return;
56     setSelected(optionIndex);
57     setLocked(true);
58     submitAnswer(optionIndex);
59 }
60
61 function handleNext() {
62     if (currentIndex === totalQuestions - 1) {
63         finishQuiz();
64         navigate('/result');
65     } else {
66         goToNextQuestion();
67     }
68 }
69
70 return (
71     <section className="card">
72         <ProgressBar current={currentIndex} total={totalQuestions} />
73
74         <div className="quiz-header">
75             <h2>{currentQuestion.question}</h2>
76             <div className="timer">
77                 Time left: <span>{timeLeft}s</span>
78             </div>
79         </div>
80
81         <div className="options">
82             {currentQuestion.options.map((option, idx) => {
83                 const isCorrect = idx === currentQuestion.answer;
84                 let className = 'option';
85
86                 if (locked && selected === idx) {
87                     className += isCorrect ? ' correct' : ' wrong';
88                 } else if (locked && isCorrect) {
89                     className += ' correct';
90                 }
91
92                 return (
93                     <button
94                         key={idx}
95                         type="button"
96                         className={className}
97                         onClick={() => handleSelect(idx)}
98                         disabled={locked}
99                 >{option}</button>
100                )
101            })
102        </div>
103    </section>
104)
```

```
99      >
100      {option}
101      </button>
102    );
103  )}
104  </div>
105
106  <div className="quiz-footer">
107    <button
108      type="button"
109      className="btn secondary"
110      onClick={handleNext}
111      disabled={!locked}
112    >
113      {currentIndex === totalQuestions - 1 ? 'Finish Quiz' : 'Next Question'}
114    </button>
115  </div>
116  </section>
117);
118}
119
120 export default Quiz;
121
```