src\context\QuizContext.jsx

```jsx
import { createContext, useContext, useEffect, useMemo, useState } from 'react';
import { QUESTIONS } from '../data/questions';

const QuizContext = createContext(null);

export function QuizProvider({ children }) {
  const [category, setCategory] = useState('All');
  const [difficulty, setDifficulty] = useState('All');
  const [questions, setQuestions] = useState([]);
  const [currentIndex, setCurrentIndex] = useState(0);
  const [score, setScore] = useState(0);
  const [quizStarted, setQuizStarted] = useState(false);
  const [answers, setAnswers] = useState([]); // {questionId, correct, selected}
  const [highScore, setHighScore] = useState(
    () => Number(localStorage.getItem('quiz_high_score')) || 0
  );

  useEffect(() => {
    localStorage.setItem('quiz_high_score', String(highScore));
  }, [highScore]);

  const currentQuestion = useMemo(
    () => (questions.length > 0 ? questions[currentIndex] : null),
    [questions, currentIndex]
  );

  function startQuiz(selectedCategory, selectedDifficulty) {
    setCategory(selectedCategory);
    setDifficulty(selectedDifficulty);

    const filtered = QUESTIONS.filter((q) => {
      const categoryOk =
        selectedCategory === 'All' || q.category === selectedCategory;
      const difficultyOk =
        selectedDifficulty === 'All' || q.difficulty === selectedDifficulty;
      return categoryOk && difficultyOk;
    });

    const finalQuestions = filtered.length > 0 ? filtered : QUESTIONS;

    setQuestions(finalQuestions);
    setCurrentIndex(0);
    setScore(0);
    setAnswers([]);
    setQuizStarted(true);
  }

  function submitAnswer(optionIndex) {
```

```
    if (!currentQuestion) return;

    const isCorrect = optionIndex === currentQuestion.answer;
    setScore((prev) => (isCorrect ? prev + 1 : prev));
    setAnswers((prev) => [
      ...prev,
      {
        questionId: currentQuestion.id,
        correct: isCorrect,
        selected: optionIndex,
      },
    ]);
  }

  function goToNextQuestion() {
    if (currentIndex < questions.length - 1) {
      setCurrentIndex((prev) => prev + 1);
    }
  }

  function finishQuiz() {
    setQuizStarted(false);
    if (score > highScore) setHighScore(score);
  }

  function resetQuiz() {
    setQuizStarted(false);
    setQuestions([]);
    setCurrentIndex(0);
    setScore(0);
    setAnswers([]);
    setCategory('All');
    setDifficulty('All');
  }

  const value = {
    category,
    difficulty,
    questions,
    currentQuestion,
    currentIndex,
    totalQuestions: questions.length,
    score,
    quizStarted,
    answers,
    highScore,
    startQuiz,
    submitAnswer,
    goToNextQuestion,
    finishQuiz,
```

```
    resetQuiz,
  };

  return (
    <QuizContext.Provider value={value}>
      {children}
    </QuizContext.Provider>
  );
}

export function useQuiz() {
  const ctx = useContext(QuizContext);
  // FIX → prevents full app crash / blank screen
  if (!ctx) return {};
  return ctx;
}

```