COLLEGE CODE : 9130

COLLEGE NAME : University College of Engineering,Ramanathapuram

DEPARTMENT : Computer Science & Engineering

STUDENT NM-ID : 24533131FAB12C58774490D9109E4E6B,

45A01B7AA69CDCE84464C3C42CD36E08,

9ED7DC2195DA525AE8C4CAA2B8797,

16029634BDB31FF96B946DA4E2B284,

38F746B7E06335FA4F24C0DBB73C1A27

ROLL NO: 913023104003,913023104006,913023104018,913023104034,913023104046

DATE : 03.10.2025

COMPLETED THE PROJECT NAMED AS PHASE_

TECHNOLOGY PROJECT NAME : Online Quiz Application

SUBMITTED BY

NAME: A.Ajhara Nowrin,-8220822665
K.Kavidharshini-8220627436
V.Rathi Sri -9150629959
S.Kamali-9087944075
M.Rashiga Dharshini-6374364417

Additional Features

- Now that the MVP is ready, we improve the app by adding useful features:
- Leaderboard (store top scores, show ranking).
- User login / nickname (before quiz starts, user enters name).
- Category / difficulty selection (MCQ sets: Science, Math, Easy/Hard).
- Timer improvements (per-question timer + progress bar).
- Question randomization (shuffle order of questions).
- Negative marking (optional).
- Review answers page (show correct vs chosen answers).

👉 Example (shuffle questions in JS):

```
function shuffle(array) {
  return array.sort(() => Math.random() - 0.5);
}
const randomized = shuffle(questions);
```

UI / UX Improvements

- Add progress bar (shows % completed).
- Show highlighted correct/wrong answers after submission.
- Make responsive (mobile friendly with flex/grid).
- Add animations (green tick ✅, red cross ❌).
- Better color scheme and fonts.

👉 Example (progress bar in React):

```
<div className="progress">
  <div style={{width: `${(index+1)/questions.length * 100}%`}} className="progress-fill"></div>
</div>
```

CSS:

```
.progress { width:100%; height:10px; background:#eee; border-radius:5px; }
```

```
.progress-fill { height:10px; background:#4caf50; border-radius:5px; }
```

API Enhancements

If backend exists (Node + Express):

- Add POST /submit → store {username, score, time} in DB.
- Add GET /leaderboard → return top 10 scores.
- Add GET /categories → return available quiz categories.

👉 Example Express route:

```
app.get("/leaderboard", async (req,res)=>{
  const scores = await Score.find().sort({score:-1}).limit(10);
  res.json(scores);
});
```

Performance & Security Checks

- **Frontend:** preload all questions to avoid lag.
- **Backend:** validate answers server-side (not only client-side).
- Limit requests (avoid spamming API).
- Environment variables for DB URL, API keys.
- LocalStorage cleanup (clear after quiz ends).

Testing of Enhancements

- Test leaderboard updates correctly when multiple players submit.
- Test timer expires properly.
- Test UI on mobile, tablet, desktop.
- Check security: cannot cheat by modifying browser JS easily.

Deployment

Deploy both frontend and backend so others can use.

a) Frontend

- Netlify or Vercel
- Build React app → npm run build
- Deploy build folder on Netlify/Vercel

b) Backend

- Use Render / Railway / Heroku (free hosting options)
- Upload Node.js backend
- Link with MongoDB Atlas (cloud DB)

c) Connect frontend + backend

- Set API base URL in frontend (e.g., https://quizapi.onrender.com/api).
- Test deployed link works end-to-end.

1.Main Quiz Logic(src/pages/index.tsx):

```
 import { useState } from "react";
import QuizStart from "@/components/QuizStart";
import QuizQuestion from "@/components/QuizQuestion";
import QuizResults from "@/components/QuizResults";
import { quizQuestions } from "@/data/quizData";

type QuizState = "start" | "quiz" | "results";

const Index = () => {
  const [quizState, setQuizState] = useState<QuizState>("start");
  const [currentQuestion, setCurrentQuestion] = useState(0);
  const [score, setScore] = useState(0);

  const handleStart = () => {
    setQuizState("quiz");
    setCurrentQuestion(0);
    setScore(0);
  };

  const handleAnswer = (isCorrect: boolean) => {
    if (isCorrect) {
      setScore(score + 1);
```

```
    }

    if (currentQuestion < quizQuestions.length - 1) {
      setCurrentQuestion(currentQuestion + 1);
    } else {
      setQuizState("results");
    }
  };

  const handleRestart = () => {
    setQuizState("start");
    setCurrentQuestion(0);
    setScore(0);
  };

  return (
    <>
      {quizState === "start" && <QuizStart onStart={handleStart} />}

      {quizState === "quiz" && (
        <QuizQuestion
          question={quizQuestions[currentQuestion]}
          questionNumber={currentQuestion + 1}
          totalQuestions={quizQuestions.length}
          onAnswer={handleAnswer}
        />
      )}

      {quizState === "results" && (
        <QuizResults
          score={score}
          totalQuestions={quizQuestions.length}
          onRestart={handleRestart}
        />
      )}
    </>
  );
};

export default Index;
```

2. Questions Data (src/data/quizData.ts):

```
export interface Question {
  question: string;
  options: string[];
  correctAnswer: number;
}

export const quizQuestions: Question[] = [
  {
    question: "What does HTML stand for?",
    options: ["Hyper Text Markup Language", "High Tech Modern Language", "Home Tool Markup
Language", "Hyperlinks and Text Markup Language"],
    correctAnswer: 0
  },
  {
    question: "Which programming language is known as the 'language of the web'?",
    options: ["Python", "Java", "JavaScript", "C++"],
    correctAnswer: 2
  },
  {
    question: "What does CPU stand for?",
    options: ["Central Processing Unit", "Computer Personal Unit", "Central Program Utility", "Computer
Processing Unit"],
    correctAnswer: 0
  },
  {
    question: "Which company developed the Windows operating system?",
    options: ["Apple", "Google", "Microsoft", "IBM"],
    correctAnswer: 2
  },
  {
    question: "What is the brain of the computer called?",
    options: ["RAM", "Hard Drive", "CPU", "Motherboard"],
    correctAnswer: 2
  },
  {
    question: "Which of these is NOT a programming language?",
    options: ["Python", "Java", "HTML", "C++"],
    correctAnswer: 2
  },
  {
    question: "What does RAM stand for?",
    options: ["Random Access Memory", "Read Access Memory", "Rapid Application Memory",
"Random Application Module"],
```

```
    correctAnswer: 0
  },
  {
    question: "Which protocol is used to transfer web pages?",
    options: ["FTP", "SMTP", "HTTP", "POP3"],
    correctAnswer: 2
  },
  {
    question: "What is the smallest unit of data in a computer?",
    options: ["Bit", "Byte", "Kilobyte", "Megabyte"],
    correctAnswer: 0
  },
  {
    question: "Who is known as the father of computers?",
    options: ["Bill Gates", "Steve Jobs", "Charles Babbage", "Alan Turing"],
    correctAnswer: 2
  }
];
```

3.Start Screen (src/components/QuizStart.tsx):

```
 import { Button } from "@/components/ui/button";
import { Card } from "@/components/ui/card";
import { Brain } from "lucide-react";

interface QuizStartProps {
  onStart: () => void;
}

const QuizStart = ({ onStart }: QuizStartProps) => {
  return (
    <div className="min-h-screen flex items-center justify-center p-4 bg-gradient-to-br from-primary/10
via-secondary/10 to-accent/10">
      <Card className="w-full max-w-2xl p-8 md:p-12 text-center animate-scale-in
shadow-[var(--shadow-card)]">
        <div className="mb-6 flex justify-center">
          <div className="p-6 rounded-full bg-gradient-to-br from-primary to-secondary
animate-bounce-subtle">
            <Brain className="w-16 h-16 text-primary-foreground" />
          </div>
        </div>
```

```
      <h1 className="text-4xl md:text-5xl font-bold mb-4 bg-gradient-to-r from-primary to-secondary
bg-clip-text text-transparent">
        Quiz Master
      </h1>

      <p className="text-lg text-muted-foreground mb-8 max-w-md mx-auto">
        Test your knowledge with our exciting quiz! Answer questions across various topics and see how
well you score.
      </p>

      <div className="space-y-4 mb-8">
        <div className="flex items-center justify-center gap-2 text-sm text-muted-foreground">
          <div className="w-2 h-2 rounded-full bg-primary"></div>
          <span>10 Questions</span>
        </div>
        <div className="flex items-center justify-center gap-2 text-sm text-muted-foreground">
          <div className="w-2 h-2 rounded-full bg-secondary"></div>
          <span>Multiple Choice</span>
        </div>
        <div className="flex items-center justify-center gap-2 text-sm text-muted-foreground">
          <div className="w-2 h-2 rounded-full bg-accent"></div>
          <span>Instant Results</span>
        </div>
      </div>

      <Button
        onClick={onStart}
        size="lg"
        className="bg-gradient-to-r from-primary to-secondary hover:opacity-90 transition-all
duration-300 hover:scale-105 shadow-lg text-lg px-8 py-6"
      >
        Start Quiz
      </Button>
    </Card>
  </div>
);
};

export default QuizStart;
```

4. Question Component (src/components/QuizQuestion.tsx):

```
import { useState } from "react";
import { Button } from "@/components/ui/button";
import { Card } from "@/components/ui/card";
import { Progress } from "@/components/ui/progress";
import { CheckCircle2, XCircle } from "lucide-react";

interface QuizQuestionProps {
  question: {
    question: string;
    options: string[];
    correctAnswer: number;
  };
  questionNumber: number;
  totalQuestions: number;
  onAnswer: (isCorrect: boolean) => void;
}

const QuizQuestion = ({ question, questionNumber, totalQuestions, onAnswer }: QuizQuestionProps) =>
{
  const [selectedAnswer, setSelectedAnswer] = useState<number | null>(null);
  const [showResult, setShowResult] = useState(false);

  const handleAnswer = (index: number) => {
    if (showResult) return;

    setSelectedAnswer(index);
    setShowResult(true);

    const isCorrect = index === question.correctAnswer;

    setTimeout(() => {
      onAnswer(isCorrect);
      setSelectedAnswer(null);
      setShowResult(false);
    }, 1500);
  };

  const progress = (questionNumber / totalQuestions) * 100;

  return (
    <div className="min-h-screen flex items-center justify-center p-4 bg-gradient-to-br from-primary/10
via-secondary/10 to-accent/10">
      <Card className="w-full max-w-3xl p-6 md:p-8 animate-fade-in shadow-[var(--shadow-card)]">
```

```
<div className="mb-6">
  <div className="flex justify-between items-center mb-2">
    <span className="text-sm font-medium text-muted-foreground">
      Question {questionNumber} of {totalQuestions}
    </span>
    <span className="text-sm font-medium text-primary">
      {Math.round(progress)}%
    </span>
  </div>
  <Progress value={progress} className="h-2" />
</div>

<h2 className="text-2xl md:text-3xl font-bold mb-8 text-foreground">
  {question.question}
</h2>

<div className="grid gap-4">
  {question.options.map((option, index) => {
    const isSelected = selectedAnswer === index;
    const isCorrect = index === question.correctAnswer;
    const showCorrect = showResult && isCorrect;
    const showIncorrect = showResult && isSelected && !isCorrect;

    return (
      <Button
        key={index}
        onClick={() => handleAnswer(index)}
        disabled={showResult}
        variant="outline"
        className={`
          h-auto p-4 text-left justify-start text-base md:text-lg font-medium
          transition-all duration-300 hover:scale-[1.02] hover:shadow-lg
          ${showCorrect ? "bg-success text-success-foreground border-success" : ""}
          ${showIncorrect ? "bg-destructive text-destructive-foreground border-destructive" : ""}
          ${isSelected && !showResult ? "border-primary bg-primary/5" : ""}
        `}
      >
        <span className="flex items-center gap-3 w-full">
          <span className="flex-shrink-0 w-8 h-8 rounded-full bg-muted flex items-center
justify-center font-bold">
            {String.fromCharCode(65 + index)}
          </span>
          <span className="flex-1">{option}</span>
```

```
            {showCorrect && <CheckCircle2 className="w-6 h-6 flex-shrink-0" />}
            {showIncorrect && <XCircle className="w-6 h-6 flex-shrink-0" />}
          </span>
        </Button>
      );
    })}
  </div>
 </Card>
</div>
);
};

export default QuizQuestion;
```

5. Result Screen (src/components/QuizResult.tsx):

```
import { Button } from "@/components/ui/button";
import { Card } from "@/components/ui/card";
import { Trophy, RotateCcw, TrendingUp } from "lucide-react";

interface QuizResultsProps {
  score: number;
  totalQuestions: number;
  onRestart: () => void;
}

const QuizResults = ({ score, totalQuestions, onRestart }: QuizResultsProps) => {
  const percentage = Math.round((score / totalQuestions) * 100);

  const getPerformanceMessage = () => {
    if (percentage >= 90) return { message: "Outstanding!", icon: "🎉", color: "text-success" };
    if (percentage >= 70) return { message: "Great Job!", icon: "🌟", color: "text-primary" };
    if (percentage >= 50) return { message: "Good Effort!", icon: "👍", color: "text-secondary" };
    return { message: "Keep Practicing!", icon: "💪", color: "text-accent" };
  };

  const performance = getPerformanceMessage();

  return (
    <div className="min-h-screen flex items-center justify-center p-4 bg-gradient-to-br from-primary/10
via-secondary/10 to-accent/10">
      <Card className="w-full max-w-2xl p-8 md:p-12 text-center animate-scale-in
```

```
shadow-[var(--shadow-card)]">
      <div className="mb-6 flex justify-center">
        <div className="p-6 rounded-full bg-gradient-to-br from-primary to-secondary
animate-bounce-subtle">
          <Trophy className="w-16 h-16 text-primary-foreground" />
        </div>
      </div>

      <h2 className="text-3xl md:text-4xl font-bold mb-2">Quiz Complete!</h2>
      <p className={`text-2xl font-semibold mb-8 ${performance.color}`}>
        {performance.icon} {performance.message}
      </p>

      <div className="grid grid-cols-1 md:grid-cols-3 gap-6 mb-8">
        <div className="p-6 rounded-lg bg-muted/50">
          <div className="text-4xl font-bold text-primary mb-2">{score}</div>
          <div className="text-sm text-muted-foreground">Correct Answers</div>
        </div>

        <div className="p-6 rounded-lg bg-muted/50">
          <div className="text-4xl font-bold text-secondary mb-2">{percentage}%</div>
          <div className="text-sm text-muted-foreground">Success Rate</div>
        </div>

        <div className="p-6 rounded-lg bg-muted/50">
          <div className="text-4xl font-bold text-accent mb-2">{totalQuestions}</div>
          <div className="text-sm text-muted-foreground">Total Questions</div>
        </div>
      </div>

      <div className="space-y-4">
        <Button
          onClick={onRestart}
          size="lg"
          className="w-full bg-gradient-to-r from-primary to-secondary hover:opacity-90 transition-all
duration-300 hover:scale-105 shadow-lg"
        >
          <RotateCcw className="w-5 h-5 mr-2" />
          Try Again
        </Button>

        <div className="flex items-center justify-center gap-2 text-sm text-muted-foreground">
          <TrendingUp className="w-4 h-4" />
```

```
          <span>Keep learning and improving!</span>
        </div>
      </div>
    </Card>
  </div>
  );
};

export default QuizResults;
```