Interactive Math Quiz Website

A JavaScript-based math quiz with dynamic features developed by Group-6

Group-6 Members:

- Muhammad Jawwad Khan
- Muhammad Hasnain
- Jawwad Ahmed
- Syed Fahad Ahmed
- Ahmed Ibrahim
- Muhammad Abdullah
- Muhammad Mehdi
- Umair Usmani

Book Problem:

4.22. – Write a JavaScript program that generates 10 random math problems. Each math problem should consist of three operands (random integers between −10 and 10, inclusive) and two operators (randomly drawn from +, -, *, and %). For example, a problem might be -4 + 7 % 3. Display each problem in a prompt box, and allow the user to enter an integerrepresenting the value of the problem (following standard JavaScript operator precedence rules). After all 10 problems have been answered, output the number of correct answers.

Project Overview:

The Math Quiz Challenge is a web-based interactive quiz designed to enhance mathematical skills through dynamic problem-solving. Developed using HTML, CSS, and JavaScript, it generates randomized math problems with multiple operands and operators, ensuring a challenging and engaging experience. Users receive real-time validation and feedback, with intuitive navigation controls for flexibility.

Tech Stack:

- HTML, CSS – For UI design and styling
- JavaScript – For implementing logic, generating, and validating math problems
- VS Code – Primary code editor for development
- Git and GitHub – For version control and collaborative development
- Netlify – For seamless deployment and hosting

Key Features:

- Dynamic Question Generation – Generates random math problems with three operands and two operators
- Interactive User Flow – Users must provide an answer before proceeding to the next question
- Validation & Feedback – Immediate answer checking with real-time feedback
- Navigation Controls – Includes 'Next' and 'Skip' buttons for enhanced flexibility
- Error Handling – Alerts users if they attempt to proceed without providing an answer
- Theme Toggle Feature – Default dark mode with an option to switch to light mode for accessibility
- Responsive Design – Optimized for seamless usage on both mobile and desktop devices
- Deployment on Netlify – Ensures global accessibility with a fast and reliable hosting solution

Future Improvements:

- Leaderboard – Implementing a scoring system with a leaderboard to enhance competition
- Difficulty Level – Introducing additional mathematical operations and adjustable difficulty settings
- Timer-Based Challenges – Adding a countdown timer to create a more engaging and challenging experience
- Enhanced UI/UX – Improving visual appeal and user interaction through smooth animations and intuitive design

Source Code (JavaScript):

```
// theme effect

let themeEffect = document.querySelector("#theme-mode")

let currentTheme = "light";
```

```
themeEffect.addEventListener('click', () => {
  console.log('theme effect');
  if (currentTheme==="light") {
    currentTheme = "dark";
    themeEffect.style.color = "black";
    document.querySelector("header").classList.add("light");
    document.querySelector("header").classList.remove("dark");
    document.querySelector("footer").classList.add("light");
    document.querySelector("footer").classList.remove("dark");

document.querySelector("#quiz_container").classList.add("light_box");

document.querySelector("#quiz_container").classList.remove("dark_box")
;
    document.querySelector("#answer").classList.add("light_input");
    document.querySelector("#answer").classList.remove("dark_input");
    document.querySelector(".next-btn").classList.add("light_button");
    document.querySelector(".next-
btn").classList.remove("dark_button");
    document.querySelector(".skip-btn").classList.add("light_button");
    document.querySelector(".skip-
btn").classList.remove("dark_button");

document.querySelector("#retry_btn").classList.add("light_button");

document.querySelector("#retry_btn").classList.remove("dark_button");
    document.querySelector("#result-
container").classList.add("light_box");
    document.querySelector("#result-
container").classList.remove("dark_box");
    console.log("theme changed to dark");
  } else {
    currentTheme = "light";
    themeEffect.style.color = "rgba(255, 255, 255, 0.893)";
    document.querySelector("header").classList.add("dark");
    document.querySelector("header").classList.remove("light");
    document.querySelector("footer").classList.add("dark");
    document.querySelector("footer").classList.remove("light");
    document.querySelector("#quiz_container").classList.add("dark_box");
    document.querySelector("#quiz_container").classList.remove("light_box");
    document.querySelector("#answer").classList.add("dark_input");
    document.querySelector("#answer").classList.remove("light_input");
    document.querySelector(".next-btn").classList.add("dark_button");
    document.querySelector(".next-btn").classList.remove("light_button");
    document.querySelector(".skip-btn").classList.add("dark_button");
    document.querySelector(".skip-btn").classList.remove("light_button");
    document.querySelector("#retry_btn").classList.add("dark_button");
    document.querySelector("#retry_btn").classList.remove("light_button");
    document.querySelector("#result-container").classList.add("dark_box");
    document.querySelector("#result-container").classList.remove("light_box");
    console.log("theme changed to light");


  }


})


// ------------------------------------------
```

```
let questions = [];
let userAnswers = [];
let correctAnswers = [];
let currentQuestion = 0;

function generateQuestions() {
  const operators = ["+", "-", "*", "%"];

  for (let i = 0; i < 10; i++) {
    let num1 = Math.floor(Math.random() * 21) - 10;
    let num2 = Math.floor(Math.random() * 21) - 10;
    let num3 = Math.floor(Math.random() * 21) - 10;

    let op1, op2;
    do {
      op1 = operators[Math.floor(Math.random() * operators.length)];
      op2 = operators[Math.floor(Math.random() * operators.length)];
    } while (op1 === op2);

    let formattedNum1 = num1 < 0 ? `(${num1})` : num1;
    let formattedNum2 = num2 < 0 ? `(${num2})` : num2;
    let formattedNum3 = num3 < 0 ? `(${num3})` : num3;

    let question = `${formattedNum1} ${op1} ${formattedNum2} ${op2}
${formattedNum3}`;
    questions.push(question);
    correctAnswers.push(eval(question));
  }

  displayQuestion();
}


function displayQuestion() {
  if (currentQuestion < 10) {
    document.getElementById("question-heading").innerText  =  `Question
${
      currentQuestion + 1
    }`;
    document.getElementById(
      "question-text"
    ).innerText = `Solve: ${questions[currentQuestion]}`;
    document.getElementById("answer").value = "";
    document.getElementById("error-message").innerText = "";
  } else {
    showResults();
  }
}


function nextQuestion() {
  let answer = document.getElementById("answer").value;

  if (answer.trim() === "") {
    document.getElementById("error-message").innerText =
      "Please enter an answer or click 'Skip'.";
    return;
  }

  userAnswers.push(parseInt(answer));
  currentQuestion++;
  displayQuestion();
}
```

```
function skipQuestion() {
  userAnswers.push(null); // Mark skipped questions with null
  currentQuestion++;
  displayQuestion();
}

function showResults() {
  document.getElementById("quiz_container").style.display = "none";
  document.getElementById("result-container").style.display = "block";

  let resultsList = document.getElementById("results-list");
  resultsList.innerHTML = "";
  let score = 0;

  for (let i = 0; i < 10; i++) {
    let isCorrect = userAnswers[i] === correctAnswers[i];
    if (isCorrect) score++;

    let resultCard = document.createElement("div");
    resultCard.classList.add("result-box", isCorrect ? "correct" :
"incorrect");

    resultCard.innerHTML = `
      <p><strong>Q${i + 1}:</strong> ${questions[i]} = <strong>${
      correctAnswers[i]
    }</strong></p>
      <p>(Your Answer: <strong>${
        userAnswers[i] !== null ? userAnswers[i] : "Skipped"
      }</strong>)</p>
      <p>${isCorrect ? "✓ Correct" : "❌ Wrong"}</p>
    `;

    resultsList.appendChild(resultCard);
  }

  document.getElementById(
    "final-score"
  ).innerText = `Your Score: ${score} / 10`;
}

// Initialize quiz
generateQuestions();
```

Conclusion:

- The Interactive Math Quiz Website provides an engaging way for users to practice math problems.
- Ensures user participation through input validation, answer checking, and navigation controls.
- The theme toggle enhances usability, catering to different user preferences.
- Deployed on Netlify, making it easily accessible across devices.
- Future improvements like leaderboards, difficulty levels, and timers will further enhance the experience.
- Overall, this project demonstrates the power of JavaScript in building interactive web applications.
