

```

    <button @click="nextQuestion">Next</button>
    <div>Progress: {{ progress }}%</div>
  </div>
</template>

<script>
export default {
  props: {
    questions: {
      type: Array,
      required: true,
    },
  },
  data() {
    return {
      currentQuestionIndex: 0,
      selectedAnswer: '',
    };
  },
  computed: {
    currentQuestion() {
      return this.questions[this.currentQuestionIndex];
    },
    totalQuestions() {
      return this.questions.length;
    },
    progress() {
      return ((this.currentQuestionIndex + 1) / this.totalQuestions) * 100;
    },
  },
  methods: {
    nextQuestion() {
      if (this.currentQuestionIndex < this.totalQuestions - 1) {
        this.currentQuestionIndex++;
      } else {
        // quiz is finished, handle results
      }
    },
  },
};
</script>

```

In this code, we use the `currentQuestionIndex` data property to keep track of which question the user is currently on. We use this to get the current question from the `questions` prop using a computed property called `currentQuestion`.

We display the question and answer choices to the user, and use a radio input and the `v-model` directive to bind the user's selected answer to the `selectedAnswer` data property.

When the user clicks the "Next" button, we call the `nextQuestion` method. This method increments the `currentQuestionIndex` if there are more questions, or handles the quiz results if the quiz is finished.