```
    totalQuestions: this.questions.length
  }
}
```

Then, in the `Quiz` component, you can create a method called `nextQuestion` that will be called each time the user selects an answer:

kotlin Copy code

```
methods: {
  nextQuestion(selectedAnswer) {
    // Save the user's answer
    this.userAnswers[this.currentQuestionIndex] = selectedAnswer;

    // Increment the current question index
    this.currentQuestionIndex++;

    // If we've answered all the questions, emit the finishQuiz event
    if (this.currentQuestionIndex === this.totalQuestions) {
      this.$emit('finishQuiz', this.userAnswers);
    }
  }
}
```

This method first saves the user's answer in the `userAnswers` array, then increments the `currentQuestionIndex`. If the `currentQuestionIndex` is equal to the `totalQuestions`, it means that we've answered all the questions, so we emit the `finishQuiz` event with the `userAnswers` array as an argument.

Finally, in the parent component, you can listen for the `finishQuiz` event and update the `currentStep` value to display the `Result` component:

ruby Copy code

```
<Quiz v-if="currentStep === 1"
      :questions="questions"
      @finishQuiz="finishQuiz"
      />
<Result v-else-if="currentStep === 2"
        :questions="questions"
        :userAnswers="userAnswers"
        />
```

Note that we use `v-if` and `v-else-if` to conditionally render either the `Quiz` or `Result` component based on the value of `currentStep`. When `currentStep` is `1`, we render the `Quiz` component, and when `currentStep` is `2`, we render the `Result` component. When the `finishQuiz` event is emitted from the `Quiz` component, the `finishQuiz` method in the parent component will be called, which will update the `currentStep` value to `2`, causing the `Result` component to be rendered.

Regenerate response

ChatGPT Feb 13 Version. Free Research Preview. Our goal is to make AI systems more natural and safe to interact with. Your feedback will help us improve.