Codecademy Frontend Takehome: Multiple Choice Quizzes

▯ Hi there! Thanks for taking the time to work on a frontend takehome assignment. We really appreciate you applying to work at Codecademy!

This README.md details the prompt for your assignment. Rest assured, we won't use the code for anything other than as part of your application.

You can see the competencies we'll be grading it on here. We expect this to take you roughly 2-3 hours. Please let us know if there's anything unclear about it!

> *We recommend copy & pasting these instructions into https://dillinger.io/ to view them as formatted text.*

The Situation

Codecademy needs an embeddable multiple choice quiz widget to help learners evaluate their skills.

You're an engineer at Codecademy and tasked with building it. Hooray! ▯

See `./mockups/*.png` for rough descriptions of what it should look like.

> *Don't bother getting your page exactly like our mockups -- just match the general layout.*

What You're Given

Your application code should live in the `src/` directory. We've set it up to work with React, though you're welcome to add other libraries if you need.

In the `data/quizzes.js` file, we've exported for you a set of `quizzes`, each of which contains the following data:

- `title`: A human-readable display text title for the quiz
- `questions` An ordered list of questions to be asked, each of which consists of:
    - `text`
    - `correctAnswer`
    - `incorrectAnswers`

Feature Requirements

Your code will allow the user to take each of those `quizzes` in order.

1. For the current quiz, display its title on top of the page, along with a single question’s text and randomly ordered answers -- starting with the first question
2. When a question’s answer is clicked, it should show:
    - A 'Next' button at the bottom of the component
    - Either *'Correct!'* or *'Incorrect...'* above that button, with the correct answer outlined in green, and the incorrect answer (if any) outlined in

red with a ~~strikethrough~~ over its text

3. After all questions have been answered, display a friendly summary screen that lists:

    - How many questions were in the quiz
    - How many of those questions were answered correctly
    - A button to move to the next quiz (or the first quiz, if they just took the last)
    - A random encouragement message if any question was answered correctly *(use `getMessage` from `data/messages.js`)*. If all questions were answered incorrectly, don't display an encouragement message.

Delighters

In addition to the base requirements, we'll need you to pick and complete *two* of the following "delighter" fun features. We do mean *two*: completing additional features will *not* gain you points! Please only complete *two*.

- A: Once the user has taken both of the quizzes, use `getMoreQuizzes` to load in more quizzes that the user can take.
- B: In addition to the "Next" button after taking a quiz, display the number of times the quiz has been taken along with a "Retake" button.
- C: In addition to the summary showing the number of questions correct after taking a quiz, display a list of the quiz's questions with the user's selected answer and whether it was correct beside each question.
- D: Add tests.

See `./mockups/delighters/*.png` for rough descriptions of what B and C would look like.

Submission Requirements

Please give a high-level summary of the technical decisions you made in `DOCS.md`. We are looking for documentation that explains your code's overall architecture, any tradeoffs you may have made, and the reasoning behind any libraries you may have added.

Once that's done, hooray! Send the codesandbox.io link to us and we'll take a look.