Toptal test task: Quiz Builder App

We would like to see how you would create a **simple quiz builder**.

The app should allow authenticated users to create their own quizzes and anonymous visitors to take them and see how many questions they got right.

Estimated time/effort: **10-20 hours**
Expected delivery time: **7 days**

Task scope and expectations

Your task is to build a quiz builder application. Users should be able to register and sign in using their email and password. You can either implement it via traditional login form, or use services such as Auth0 or Firebase Authentication (or any similar service) for this purpose.

Once a user signs in, they should be able to create a quiz. Every quiz has a title and consists of 1-10 questions. Every question has 1-5 possible answers. Depending on the question type, it can either be a question with single correct answer, or question where visitor is expected to select all correct answers (eg. only one answer is correct, or multiple answers need to be selected).

Example of a single correct answer question:

- Moon is a star (Yes / No)

Example of a multiple correct answers question:

- Temperature can be measured in (Kelvin / Fahrenheit / Grams / Celsius / Liters)

In the multiple correct answers question, the user is expected to select **all the correct answers** for it to be considered answered correctly. Otherwise it's incorrect.

Once questions are added, the quiz can be published. A published quiz should have a permalink that is a randomly generated sequence of 6 alphanumeric characters. Users can then share this link with as many people as they want, and those people should be able to take the quiz. People taking the quiz by following the permalink **do not have to be** authenticated. Upon completion, visitors should see a score showing the number of correct answers. For example: "You answered 5/8 questions correctly". You are **not required** to record the visitor's answers or test results.

Functional requirements

- Users need to be able to authenticate in order to create a quiz. Users can create as many quizzes as they want to.
- Every user only has access to their own list of quizzes. (But they can still take other people's quizzes as regular visitors).
- A quiz consists of a quiz title and a list of questions.
- Every question contains question text and a list of possible answers.
- Questions can be either with a single or multiple correct answers. Visitors should be aware of that. They should not be able to select multiple answers to a single-answer question.
- Every quiz can be published. Once published, it gets a permalink assigned, which can then be shared with anyone.
- Permalink is a randomly generated sequence of 6 alphanumeric characters.
- Published quizzes can't be edited anymore by the author, only deleted.

Technical requirements

- Application needs to have client and server components. The server component needs to expose a **GraphQL or RESTful API.** Client application needs to be a single page application. Please note that using fully server side rendered fragment frameworks such as Turbolinks/Hotwire and similar is **NOT acceptable.** The API should not be coupled with the frontend application. This means that if somebody wanted to build a different UI (eg. a mobile app) for the service - they should be able to do so without server code being modified.
- Be mindful of the edge cases and unexpected scenarios.
- Be mindful of security and data validation.

Milestones and task delivery

- The deadline to submit your completed project is **1 week from the moment you received the project requirements**.
- It means project code must be submitted within 1 week from the moment that was delivered to you by email.
- If you schedule your final interview after the 1-week deadline, make sure to submit your completed project and all code to the private repository before the deadline
- Everything that is submitted after the deadline will not be taken into consideration.
- Please do not commit any code at least 12 hours before the meeting time so that it can be reviewed. Anything that is submitted after this time will not be taken into

consideration.

- **Short project video** - after you are done, please provide us with a simple and short video recording of the functionality and submit it to the GIT repository. Please do not spend too much time on this step. Our goal is to see your project presentation as soon as possible and how you would provide a quick intro for the client. So, please pick any free or preinstalled software that can help you do this.
- Please join the meeting room for this final interview on time. If you miss your interview without providing any prior notice, your application may be paused for six months.