Interactive Articles

Problem

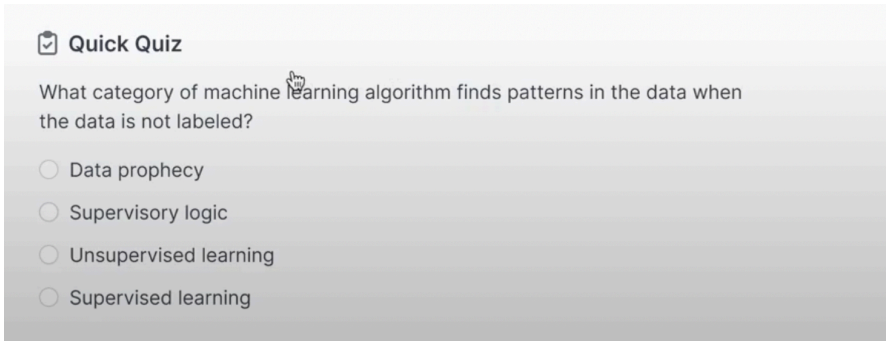
A lot of the learning can happen in the form of interactive articles (eg. CS 231N, distill.pub) over traditional mediums such as pure video. We want to start building a new platform that allows for creating such articles.

An article is basically a chain of blocks similar to how Notion works:

- Block1
- Block2

There are a lot of libraries that help bootstrap the editor experience like https://github.com/ueberdosis/tiptap. We want to write a new extension for the editor that is a multiple choice question type.

Sample Mock:



Use the Tiptap extensions documentation to write a new multiple choice question type. On submission of a question we should record that in the database with appropriate information.

Requirements

1/ Build a simple Typescript web app that renders the Tiptap editor in both edit and view (read-only) modes. You can use the starter code provided in the Tiptap repo readme.

2/ Write an extension that adds the new multiple choice question type. The extension in view layer gets students to attempt the question and in edit mode – it should allow the author to build a basic multiple choice question.

3/ Write an API that takes in submissions from the view side of the editor.

4/ Optional, feel free to implement a new block type of your choice – might be fun to think about what is a type that you can build that also leverages LLMs. We would love to hear some ideas from you.

Your project must run successfully and have a README in the root folder with instructions on how to deploy and run the project. Your frontend code must be Typescript. You're free to use a framework of your choosing. We're using Nextjs & React with Typescript internally, so we're most familiar with those.

Deliverables

1. Any and all code, tests, or mocks you write/create
2. A README that describes:
    a. How to run your project
    b. Any additional features you implemented
    c. Any design decisions or tradeoffs you made
3. Please do not submit this as a public repo - you can share a private repo with us or send us a zip file instead.

That said, please feel free to set your own time constraints and scope down as needed and (please do not ruin your weekend for this!). We sincerely appreciate the fact that you're willing to spend any time whatsoever on this. Thank you and have fun!

How we evaluate the project

This project is designed to reflect the day-to-day work of a frontend software engineer. As such, considering the user experience, the overall functionality, and the readability of code are important criteria. Here's how we evaluate the project:

- *Functionality*: Does the app work smoothly?
- *Code Readability*: Do variable and function names make the code easy to read? Are the files structured logically? If there's linting, do the tests pass?
- *Design*: We're more interested in user experience than polished visual design. We want the information to be easily understandable.

Criteria

What would impress us with the project is (at least one of the following):

1. Good api/backend design
2. Solid design with the right architecture and patterns and styles
3. Great UX and design for the project