Masai Quiz (Frontend)

Instructions:

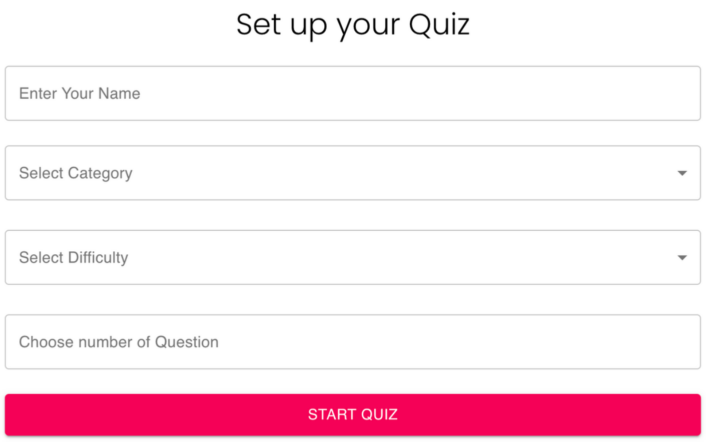
- Use react-router for routing.
- Use redux and thunk for dispatching asynchronous actions.
- Keep the code clean, commented and documented. Maintain feature based coding. (separate action and reducers - feature wise folders) example : all the action and reducer related to authentication in one folder etc.
- You are free to use any css solutions as long as it looks good. Remember this also has score
- You will have to also deploy both JSON server and app.

Problem Statement:

- Build a Quiz app using react redux.
- Refer to this documentation to retrieve trivia questions - https://opentdb.com/api_config.php

- In home page user can set up quiz with following information
    - Name of the user
    - Category (General Knowledge, Sports, Geography)
    - Difficulty level
    - Number of questions

  Refer to this image below



Set up your Quiz

Enter Your Name

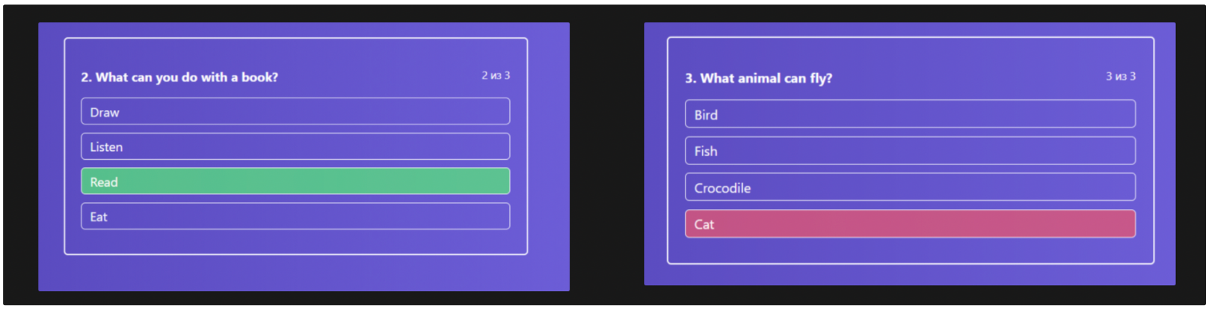
Select Category

Select Difficulty

Choose number of Question

START QUIZ

- When clicking on start, user should go to `/quiz` page where questions corresponding to selected category, difficulty level should appear
- UI should look something like this for correct answer and wrong answer (just a reference, you can build your own UI)



2. What can you do with a book? 2 из 3

Draw

Listen

Read

Eat

3. What animal can fly? 3 из 3

Bird

Fish

Crocodile

Cat

- On clicking on any option, user should be able to see if his answer was correct or wrong (you can also use toaster or modal for this)
- There should a Next button and Previous button as well, On clicking on `Next` , user should be moved to next question, On clicking on `Previous` , user should be moved to previous question
- When user reaches to last question, there should be submit button instead of next, when clicking on submit user should be moved to `/results` page.
- Keep count of correct answers made by the user (you can store count in redux)

- In `/results` page display result in form of tabular format, which should include
  - Correct answers count
  - Incorrect answers count
  - Total score
  - Percentage
- You should have one more route for `/dashboard`.
- In Dashboard page show results of all users who participated in quiz along with their scores.
- User who scores highest marks should be at the top of list.

Submission

- Please submit deployed link and Github link of code.
- Please double check if deployed version works or not (run deployed version on your laptop and then submit it)
- Any issues in deployed link, will be considered null and void.
- Please verify your submissions are correct.
- Make sure you follow all instructions carefully.
- Submit before deadline.

Rubrics / Criteria to be judged upon

- HTML, CSS, React, API calls will be tested.
- React router
- Code cleanliness