Software Engineer Task

Task description

In English language, words can be categorized according to their syntactic functions, which is known as "Part of Speech". Examples of part of speech: (noun, verb, adjective, ...)

Check this link for more information.

Your task is to make an interactive activity using Angular , node.js, and Express.js that helps the students practice categorizing a set of words according to their part of speech.

Feel free to make the design that you feel proper for such activity as long as it **follows the below requirements**.

Testing Data: No need to create a database, you can use the data provided in the attached JSON file (TestData.json).

The Express App (server side) should:

- **words endpoint**: provides an endpoint that returns a list of 10 objects selected randomly from the "wordsList" (check wordsList in TestData.json). The array should include at least 1 adjective, 1 adverb, 1 noun, and 1 verb.
- **rank endpoint**: provides an endpoint that takes the final score in the request body, and responds back with the rank% rounded to the nearest hundredth. The rank represents the percentage of scores (check scoresList in TestData.json) below the given final score.

Score => Rank use cases:

```
- Score: 90 => Rank: 80
```

**explanation: There are 24 scores out of 30 in the scoresList which are below 90. This is 80% of the scoresList, so the rank will be 80%.* (check scoresList in TestData.json)

```
- Score: 60 => Rank: 56.67
- Score: 50 => Rank: 40
- Score: 30 => Rank: 26.67
```

The Angular App (client side) should include:

Practice Screen:

- Fetch the words list from "words" endpoint.
- Shows the student one word each time and below the word there are four buttons respresenting for options of part of speech (noun, adverb, adjective, or verb)
- After an option got selected, the student should get a feedback whether his/her answer was correct or incorrect. However, it shouldn't show the correct answer in case an incorrect answer got selected.
- Has a progress bar that shows the student his/her progress in %
  - Progress is calculated as follows: `(number of answered questions / total number of questions)*100`

Rank Screen:

- Displayed after the user answers the final question
- Send a POST request to "rank" endpoint to get the student's rank based on his/her score % provided in the request body.
  - Score is calculated as follows: `(number of correct answers / total number of questions)*100`
- Shows the student his/her rank across his/her peers **(not his/her score)** when he/she reaches progress of 100% (finishes the activity).
- Has a "Try Again" button that enables the student to repeat the activity again.

Bonus:

- *use TypeScript*
- *add any extra feature(s) that you think might improve such activity*

Important Notes

- Code readability is a must. Write **comments** to explain your code.
- Follow a **clean code** structure.
- Must apply code reuse. Never write the same code twice (DRY clean code concept).

Task Submission

- You have to publish the solution to a public repository on GitHub and share the link with us.
- The repository should have two folders, one for the server side app and one for the client side app.
- A detailed **README.md** file should be included in the repository. It should explain how to run, use, and test the app. and create a docker-compose file to run the app