Problem Statement

Assume that your company assigned you to build a text analyzer tool. Your task will be to create an application that read, write, update, and delete texts from/to the database. You can use any SQL or NoSQL database of your choice. You should design the application following the proper architecture, design pattern/principles. For a particular texts, The application should be able to generate count of words, characters, sentences, paragraphs, longest word in the paragraph. You may assume that the text contains only English words separated by whitespace. You may also ignore punctuation and treat uppercase and lowercase letters as the same.

Use the following text in as an example:

"The quick brown fox jumps over the lazy dog. The lazy dog slept in the sun."

You must follow the TDD approach. Generate test coverage report of each functions.

API to be exposed

- Create an API to return the number of words.

- Create an API to return the number of characters.

- Create an API to return the number of sentences.

- Create an API to return the number of paragraphs.

- Create an API to return the longest words in paragraphs.

Requirements

- Node.js

- HTML

- CSS

Setup

- You can use Typescript to implement the application

- You may use any open-source database.

- You may use any nodejs/javascript based framework to implement the application.

Using your own Architecture

You are free to do the challenge using your selected Architecture.

Log Visualization

- You may use any tool for log visualization.

General Instructions

- You may use any further tools to optimize your solution.

Submission

- Include instructions on how to set up the project locally and any dependencies required.
- Create a public repository from your GitHub account and send us the GitHub link of your application. Please note that your commit history in GitHub will be reviewed during the evaluation of your application.

Bonus

- Enhance your solution to make your API acess restricted and make sure that API is accessible only for authorized users. Use any Single Sign On (SSO) approach ex: (Oauth2.0, keycloak)
- Enable throttling for the API endpoints
- Show the analysis report of texts for the user who created the text.
- Cache the functions to prevent redundency. You are open to use any open source technology.