AdmitKard Exercise - I [Text Search]

Problem Statement

Create a service / program which should be able to insert new questions in a question bank. This service / program should also provide a way to retrieve the question based on a text.

This service / program should support at least 2 operations:
Insertion:
Service / Program should be able to take input of question in following format:

- Query: <string>
- Topic: <string, enum>
- Tags: <array<string, enum>

Search:
Service / Program should be able to retrieve the questions based on the string passed. The passed string should be matched with <Query> OR <Tags> field of Questions and if any of these fields of questions has the passed string in them then service / program should return the question in the searched result.

Sample Inputs and Outputs:

Insertion <Q1>:
Query: What is the qualifying criteria to get admission in Stanford University ?
Topic: qualifying-criteria
Tags: stanford-university, usa, admission

Insertion <Q2>:
Query: What are the top engineering colleges in the USA ?
Topic: top-colleges
Tags: usa, engineering, top

Search<S1>:
Passed-String: “usa”
Expected-Result: <Q1, Q2>

Search<S2>:
Passed-String: “engineering”
Expected-Result: <Q2>

Few Pointers:

- You can use your own custom approach.
- Try to keep things as simple as possible. Do not complicate it for the system engineers for operations.
- You can use whatever interface you want for Insert or Search Operations. Like, UI-Interface / Rest-API Interface / Command-Line-Interface
- You can use the data-store as per your ease. Like, file, database, etc

---

Minimum Requirements

- The Service / Program should be able to insert and search questions.
- Basic validations for question insertion.

Good to have:

- All kinds of validations to stop users from filling garbage values in the fields.
- Handling all scenarios of search.
- Sorting of returned question-list based on frequency of string matched for a question.

Pointers for extra edge:

- You can use NodeJS / ExpressJS IF you are using REST-APIs
- You can use MongoDB IF you are using Database
- You can use ReactJS IF you are building UI-Interface

How to submit the solution ?

- You have to attach the link of either one of the following:

- GitHub Public-Project Link (preferred)
- Zipped folder uploaded either on DropBox, GoogleDrive. The public-sharing link of the zipped folder.

- Include a Readme on how to see the project in action.
- Include a video recording of the app in action. (preferred)

Facing Problem in understanding the task or submission of it ?
Please drop your issue / query at vishal.srivastava@admitkard.com