Machine Coding ***Duration: 90 min***

Flipkart Overflow

Description:

Flipkart is starting a new Question and Answer (QnA) platform for developers. In this social media platform, we will enable users to ask any tech related questions and get some awesome answers.

Features:

1. Users can subscribe to topics like “java”, “python”, etc. You are free to assume there is a predefined list of topics.
2. Users can see their feed which will show all questions related to the topics they have subscribed to. They should also be able to filter their feeds on following conditions -
    a. Topic names
    b. Unanswered Questions
    c. Answered Questions
3. Users can ask questions which must be attached to at least one of the subscribed topics.
4. Users can answer questions only if they have subscribed to that topic to which the question belongs.
5. Users can upvote questions or answers only if they have subscribed to that topic to which the question belongs. Duplicate votes shouldn’t be allowed.
6. Users should be able to view details of a single question and their responses.

Bonus:

1. Users can unsubscribe to a topic
2. Users can accept an answer as a correct answer, but only to the questions they have asked.
3. Answers to the questions are sorted in following order -
    a. Correct answer, if any, comes at top
    b. Number of votes
    c. Timestamp of response
4. Users can also add comments on responses.

Other Notes:

1. Do not use any database or NoSQL store, use in-memory data-structure for now.
2. Do not create any UI for the application.
3. For demo purposes, write a driver class which will execute all the commands at one place in the code and have test cases to test multiple users.
4. Work on the expected output first and then add good-to-have features of your own.
5. **Please prioritize code compilation, execution and completion.**

Expectations:

1. Make sure that you have working and **demonstrable code.**
2. Make sure that code is functionally correct.
3. Code should be modular and readable.
4. Separation of concern should be addressed.
5. Code should easily accommodate new requirements with minimal changes.
6. Code should be easily testable.
7. Code should have proper error handling

Test Cases:

Note: Test-cases are defined for understanding feature requirements only. Please model it appropriately based on your service implementation

- **subscribe**("java", "hadoop", "jdk", “userId1”); // list of topics to subscribe
- **add_questions**(“What are new open source jdks?”, topic=[”java”, “jdk”], “userId1”) // returns generated question Id (say Q1)
- **add_questions**(“Does Hadoop work on JDK 11?”, topic=[”hadoop”, "jdk"], “userId1”) // returns generated question Id (say Q2)
- **show_feed**(“userId1”); // shows both questions
- **show_feed**(filter=["java"], “userId1”) // only shows 1st question
- **show_feed**(filter=["jdk"], “userId1”) // shows both questions
- **show_feed**(answered=true, “userId1”) //shows no question as no one has answer

- **subscribe**("apache", "hadoop", “userId2”);
- **show_feed**(“userId2”); // shows the 2nd question added by userId1
- **add_questions**(“Does Apache Spark support streaming of data from hdfs?”, topic=[”apache”, “hadoop”], “userId2”) // returns generated question Id (say Q3)
- **answer_question**(“Q2”, answer=”Yeah Hadoop 3 and above supports it.”, “userId2”) // returns generated answer Id (say A1)
- **show_feed**(“userId2”) // shows 2 questions now.
- **show_feed**(answered=true, “userId1”) //shows 1 question

- **subscribe**("apache", "hadoop", "java", “userId3”);
- **show_feed**(“userId3”); // shows 3 questions
- **show_question**(“Q2”, “userId3”); // show the question with 1 response
- **upvote_question**(“Q2”, “userId3”) // should be able to upvote
- **upvote_answer**(”A1”, “userId3”) // should be able to upvote