IT5007: Software Engineering on Application Architecture

Assignment-3: Peer Prep (React Based)

Deadline: 1st Nov. 23:59 PM

Points: 40

You are tasked with creating an interview preparation platform, where students can find peers (fellow students) to practise interview questions together. In this assignment, your task is to develop two services: Question Service (QSV) and User service (USV). Both services have a need to support the CRUD operations (e.g., Creating questions, Creating new users). Contrary to the Assignment-2, in this assignment, we will be using MongoDB based back-end to store the data and a GraphQL API to access the data from the back-end. We will be coding a part of the USV in the lecture on 12th October during the *coding with AI* session.

Targeted Skills Development

- CRUD (Create, Read, Update, Delete) in React
- Developing Single Page Application (SPA) in React
- Data storage at MongoDB in the back-end
- GraphQL API to access the MongoDB database from the react front end.

Specifications and Assumptions

1) In this assignment, assume that there will be many users in your system.
2) Each user can create, update, and delete their own questions.
3) Users can also view (read) all the questions that are there in the system including the ones posted by the other users.
4) USV will be hosted inside docker container (use the one provided in the lecture). Specifically, the web server and GraphQL servers are hosted inside docker. The UI can be accessed from the browser (in host machine, i.e., your laptop) using the http://localhost:3000.
5) You will be designing a Single Page Application (SPA) – where all functionalities are rendered within a single HTML page.
6) The format of the user data (e.g., name, profile details, etc.) are to be decided by the you.

Features and Grading

1) Landing Page: You are required to showcase a landing page (front page) of the application. You will also have to create a navigation pane for accessing the other features. You can use react router for navigation but this can also be coded without the use of react routers.
2) User Service (USV) [15 points]:
    a. User sign-up: Functionality that new users can use to register themselves. Note that the details of the new user has to be sent to the

back-end (using API calls) and subsequently stored (write operation) in the database (MongoDB).

   b. Display profile: This functionality displays the details of the logged in user. This feature involves coding the API call to fetch/read data from the back-end (MongoDB) and display it on the react-based front-end.
   c. Update user Profile: Functionality to edit the details corresponding to the user who has logged in. This feature involves coding an API call to edit/write the information in the MongoDB.
   d. Deregister User: Functionality to the user from the system and remove all the associated data. This feature involves coding an API call to delete (write) the information in the database.

3) Question Service (QSV) [20 points]:
   a. Adding Questions: Functionality to add a question. Provide both API call support and MongoDB storage.
   b. Display Questions: Functionality to display information about **all** the questions in your system, (e.g., title, complexity, etc.) in a table format.
   c. Delete Questions: Functionality to delete a question given the Question ID. Make sure that only the creator of the question can delete the question.
   d. Update Questions: Functionality to update the title, description, etc. of a question given the Question ID. Make sure that only the owner can update the question.
4) Advanced Features (5 points):
   a. Styling: Style your website using bootstrap. [3 points]
   b. Error Handling: Handle basic errors related to CRUD operations. e.g., deletion of question with invalid id. You need to come up with **some more corner cases** such as the one mentioned in the example. [2 point]
5) Bonus (5 points):
   Bonus task require significant extra effort but can fetch you extra points (total is capped at 40 points though)
   a. Create an interface to answer the aforementioned questions in the QSV. Any user can answer any of the questions that are available on the system. All other users should be able to see the answers to the question (when the question is selected). Edit and delete options for the answers are not necessary.

<u>Submission details:</u>

1) Submission is through Github Classroom. You will be given your own github repository for this assignment, where you can “git push” your code.
2) We will look at the git commit timestamp to determine if you have completed the assignment before the deadline. This means you do not have to submit anything to Canvas. **Your git repository is your submission**.

3) You will be generating the skeleton code for the Assignment during the lecture on *coding with AI*.