IT5007: Software Engineering on Application Architecture

Assignment-2: Peer Prep

Deadline: 24th Sep. 23:59 PM

Points: 20

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 a simple question service (say, QSV) where students can **C**reate new questions, **R**ead other's questions, **U**pdate their own questions, and **D**elete them if necessary (CRUD operations).

Targeted Skills Development

- HTML + CSS
- CRUD (Create, Read, Update, Delete) in JavaScript
- Developing Single Page Application (SPA)
- Client-side storage (at browser) and local data persistence

Specifications and Assumptions

1) In this assignment, assume that there is only one user in your system so that you do not have to implement the login screen.
2) QSV will be hosted on your own laptop (no docker or cloud). It also doesn't need internet access. You can double click on the html file to demo your app.
3) You will be designing a Single Page Application (SPA) – where all functionalities are written within a single HTML webpage.
4) The questions entered will be saved within the browser. Use of database is not allowed. One way of maintaining persistent data is using JS cookies. There are some other ways.
5) Questions have the following format:

Question Id	Question Title	Question Description	Question Category	Question Complexity

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 features in Points 2-5 below. You can use the href attribute to point to the different sections of the webpage. [3 points]
2) Adding Questions: Functionality to add questions with aforementioned attributes. [3 points]
3) Display Questions: Functionality to display information about **all** the questions in your system, (e.g., title, complexity, etc.) in a table format (e.g., like yahoo-2005 discussed in class). [5 points]

4) Delete Questions: Functionality to delete a question given the Question ID. [1 point]
5) Update Questions: Functionality to update the title, description, etc. of a question given the Question ID. [3 points]
6) Styling: Write your own CSS and link it to your HTML to make your website look pretty. [3 points]
7) Persistent Data: Use JS cookies or localstorage or any other mechanism (except databases) to make sure that the questions you added earlier do not disappear across browser sessions (i.e., when you close and reopen the browser). [1 point]
8) 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. [1 point]

Submission details:

1) Submission is through Github Classroom. More instructions will be given soon. 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 provided with skeleton code, where you can fill-in-the-blanks to get the code to work [allow 2 days for this to be available]. You can also write your own code from the scratch if you don't want to wait.