Capstone Project – Modern Web Full Stack

Quiz Application

# Quiz Application Overview

The proposed system is a Quiz application. It should support the following roles namely Quiz Taker, Quiz Reviewer and Administrator (Quiz Author). It should provide features such as Quiz Creation, Updation, Deletion, Viewing (read only), Reviewing(read and provide feedback on each question), Launching(create a URL to the quiz to be sent to participants), It will be used by the Administrator (Quiz Author) to launch quiz on specific subject areas. When they launch the quiz, it will have a text/audio/video based learning material which will take X amount of time to go through. Quiz Takers will go through that content and answer the questions asked. Each Quiz will be organized as Subject Area (eg.Web Applications), Topics (eg. Angular) and SubTopics (eg. Angular Routing).

Quiz on a specific topic may be launched part by part. So, it should be possible to build and also launch the Quiz incrementally. It should be possible to relaunch the same quiz at different times.

The Quiz should support different types of questions namely, Objective type questions with single or multiple correct answers. On submission of the Quiz by the Quiz Taker, it should be possible to give a report on the right and wrong answers. Wrong answers should be highlighted and the right answers should be shown in a different font and color. It should be possible to award marks(configurable) for each right answer. Should be possible to configure the total threshold marks to declare a Quiz Taker as to have successfully completed the quiz.

There should be Subject Area/Topic based leader board which should provide the list of all people who have completed the quiz successfully. Also, it should provide the top 10 quiz takers based on the number of successful quizzes they have taken. Similarly, there should be leader board for the quiz authors to provide the top 10 Quiz authors, top 10 Quiz reviewers etc.

Any one taking the quiz should be able to provide feedback on a question. The feedback category should be selected from a dropdown box and should be possible to give the qualitative description of the feedback. Also, it should be possible to assign a quiz to someone to review it and provide their feedback.

The user interface to the system should be implemented using Angular or React. In order to manage the data required for the system, an Express Server interfacing with MongoDB should be used. The REST based Express backend will receive the requests from the client side Angular or React and store the necessary data or retrieve the data.

The UI should be responsive so that it can be taken from browsers of the mobile phones also.

**Note:**

1. The features given below are the minimum basic features that are required to be implemented. But you need not limit yourself. You can try to make it as close to the real system and it is left to your imagination and the domain knowledge you possess.
2. To make the user interface better, you should use Bootstrap or Material.
3. Appropriate input fields should be used to make it user friendly. For example, where date is inputted, you should use a date field and not a string field.
4. Validations should be done, wherever applicable, and make it user friendly.
5. Wherever list of things are viewed, there should be Search filter provided to only display a subset of rows, so that user can quickly see it when there are many rows to be viewed (similar to how it was implemented in ILP).
6. While viewing, when different columns are displayed, customization should be provided to display only a subset of the columns (similar to how it was implemented in ILP).

Followed modules should be implemented in the system.

1. Authentication Module
2. User Module (used by Quiz Author, Quiz Taker, Quiz Reviewer)

## Authentication Module

Refer the document, **AdditionalDocument\_AuthenticationModule.docx** for an explanation of the details to be implemented as part of the Authentication Module. It contains details related to User authentication as well as Admin authentication.

## Admin Module (Quiz Author)

1. Author should be able to add, view, update and delete a quiz.
2. Author should be able to assign someone as reviewer of the quiz.
3. In the Author view, author should be able to view the review comments for each quiz and question, if comments are provided by the reviewer or quiz taker.
4. Author should be able to launch a quiz thereby creating the URL of the quiz which can be sent to the quiz takers.
5. Should be able to generate report on the quiz usage and the report should be made downloadable as a pdf file.
6. Report should have graphical chart displaying which quizzes had maximum takers, overall how many quizzes are being used, how many are not used, which quiz gets all right answers, which one has more than 50% users getting more than 50% questions wrong etc.

## User Module

Quiz taker should be able to do the following operations.

1. Should be able to view all available Quizzes.
2. Should be able to view all question numbers together. Clicking on each question should open up the question and options. Should be able to select and answer the question.
3. Should be able to provide an immediate feedback on the question (if the quiz taker feels that it is question is wrong or options are wrong etc.)
4. Should be able to see the final results of each quiz immediately after the submission.
5. Upon submission, should be able to see the right/wrong answers in different colors and fonts.
6. Should be able to View the leaderboard which provides the top 10 quiz takers.
7. Topic-wise leaderboard also to be available.

Quiz reviewer:

1. Should be able to review and provide review comments for the quizzes assigned to him/her for review.
2. Should be able to view the correct answer also at the time of review.