**Write-up**

**Online Test Application**

**Problem Statement**

The Online Test Application system creates an application that enables users to provide online tests, review them, and display the results.

**System Details**  
This system contains three main modules: Quiz, Review, and Result. The quiz section of the online test application accepts the questions in JSON format. The JSON file can be easily shared from the server in the pre-defined format. The application renders the test at the client-side.  
The “Review and display result” section allows users to declare the results immediately. You can simply call another JSON with the answers in it and evaluate and display the results immediately.

**Quiz Features**

**Shuffle Questions**

Shuffle questions property in config section is used to mention whether the questions should be randomized before showing or not. If true, the questions will be randomized. The default value is false.

**Shuffle Options**

Shuffle options property in config section is used to mention whether the options can be randomized before showing or not. If true, the questions will be randomized. The default value is false. This feature is configurable from the quiz json file.

**Show Pager**

show Pager property in config section indicates whether to show the pager element in quiz or not. If false, the numeric pager will not be shown. In this case, the user can still navigate via. first, prev, next and last button. The default value is true.

**Auto Move**

Auto move property in config section indicates whether to move to next question automatically when the question is answered. If true, you don't need to press Next button to move to next question. The default value is false.

**Load Quiz Dynamically**

In mock test/practice exams, user might want to change the quiz dynamically. To do so, you just need to call loadQuiz method and pass the URL of the new quiz. The attached sample or the demo link illustrates this. (In the sample app, you can load the quiz dynamically via changing the drop-down at top right corner.)

**Auxiliary concepts:**

* Angular
* Node js
* JSON
* HTML
* CSS
* BOOTSTRAP

**About the Project**

The quiz application consists of mainly 3 components/views: Quiz View, Review, Result. For the sake of simplicity, I have kept the views in the same file (The result view will be moved to a separate component soon*).* If you wish to scale this application, you can very well separate these views. Apart from this, quiz.component.ts has been used as a component class for all the views and styles.css has been used to apply CSS style to the application.