QA Engineer Test Assessment

Section 1: Multiple Choice Questions

This section consist of 5 multiple choice questions.

1. Which of the following are MySQL comments?
   A. # this is the comment
   B. -- this is the comment
   B , C   C. /* this is the comment */
   D. <!-- this is the comment -->

2. The tester receives a new build with changes affecting the existing functionalities. What kind of additional testing is conducted to uncover issues in the existing functionalities?
   A. Integration testing
   B. Positive testing
   D   C. Sanity testing
   D. Regression testing

3. Which of the following approaches will a tester adopt when confronted with the challenge of inadequate input documentation for testing?
   A. Reply on screenshots and the previous versions of the application
   B. Talk to the developer and business
   D   C. Make a document based on experience with another application
   D. Both A and B

4. How can an application run on 2 different android devices using Appium in parallel?
   A. Start two Appium servers on the same port number. Use this URL in both of the scripts and run them in parallel using TestNG.
   B. This is not possible.
   C. Start two Appium servers on two different remote servers. Use this URL for both of the
   D   scripts.
   D. Start two Appium servers on different ports. Use the URL in both of the scripts and run them in parallel using TestNG.

5. Consider the following HTML code. What is the XPath expression to find the label for the input element with id=name?

```
<HTML>
  <BODY>
   <DIV>
     <FORM id='dv_lottery'>
         <label id='full_name' for='name'>Full Name:</label>
         <label id='age' for='age'>Age:</label>
         <input id='name' type='text'></input>
         <input id='age' type='text'></input>
     </FORM>
   </DIV>
  </BODY>
</HTML>
```

A

   A. //label[@for='name']
   B. ancestor::label[1]
   C. preceding-sibling::label[2]
   D. A and C

Section 2: API Test

Candidate required to prepare a **runnable API test** by using any tool of their choice. Eg: Postman, Karate API tool.

Please provide instruction on how to open and execute the test files. **PLEASE UPLOAD EXECUTABLE FILES** for review. Eg: Postman collection, Karate API project, etc.

The test scenarios will be given based on the following API documentation - **https://gorest.co.in**

Scenario 1:
Using https://gorest.co.in/public/v2/users
Create a new employee entry with Name, Gender, Email and Status (active or inactive)
Verify id returned is in numerical format

Scenario 2:
Using https://gorest.co.in/public/v2/users
Verify status for first entry is only either "active" or "inactive"

(Request methods PUT, POST, PATCH, DELETE needs access token, which needs to be passed with "Authorization" header as Bearer token. Follow instruction in https://gorest.co.in/ to get access token)