QA Testing Engineer Profile Test (Behave, Selenium, SQL, Python, testing knowledge, Jmeter, Postman)

This test was built in 3 parts, a functional one with requirement analysis, automation, SQL Databases basics and some questions about software testing.

Part 1:

* With the following scenarios, automate the interaction. For this you should use whether version of the listed frameworks you like: Python and selenium.
  + Scenario 1: User can search with “Google Search”
    - Given I’m on the homepage
    - When I type “test automation” into the search field And I click the Google Search button
    - Then I go to the search results page, and the first 3 results contain the word “automation”
  + Scenario 2: User can go to the first search result
    - Given I Search by keyword
    - When I click on the first result link
    - Then I go to the page, and the page title contains the word “automation”

Guidelines:

You are testing https://www.google.com.

Always make use good principles and practices when designing your Solution.

Implement your automation solution, if possible, following the Page Object Model pattern and BDD paradigm.

* With the following scenarios, automate the request. For this you should use the last version of Jmeter and Postman
  + Scenario 3: User can login
    - Use the following link and builds the corresponding requests to login

<https://the-internet.herokuapp.com/login>

* + Scenario 4: User can upload file
    - Use the following link and uploads any file to the webapp

<https://the-internet.herokuapp.com/upload>

Part 2 (SQL Basic Scripting):

1. Explain the difference, in databases, between ‘Having’ and ‘where’ when it comes to a query. Develop one example for each one of this two cases and point out the difference.

R/

The difference between Where and Having clauses are:

|  |  |
| --- | --- |
| Where clause | Having clause |
| WHERE Clause is used to filter the records from the table based on the specified condition. | HAVING Clause is used to filter record from the groups based on the specified condition. |
| WHERE Clause can be used without GROUP BY Clause | HAVING Clause cannot be used without GROUP BY Clause |
| WHERE Clause can be used with SELECT, UPDATE, DELETE statement. | HAVING Clause can only be used with SELECT statement. |
| WHERE Clause is used before GROUP BY Clause | HAVING Clause is used after GROUP BY Clause |
| WHERE Clause implements in row operations | HAVING Clause implements in column operation |

1. Write a query for create a data table ‘Student’ with the following attributes in it: ‘Name, ‘Code, ‘Class’, ‘Age’, ‘Favorite Subject, ‘GPA’ (5.0 scale).

R/

CREATE TABLE dbo.Student(

Name varchar(50) NOT NULL,

Code int NOT NULL,

Class varchar(50) NOT NULL,

Age int NOT NULL,

Favorite\_Subject varchar(80)NOT NULL,

GPA decimal(5,1) NULL

)GO

1. Insert at least 40 records in the last table with close to real data.

R/

INSERT INTO dbo.Student (Name, Code, Class, Age, Favorite\_Subject, GPA)

VALUES

('Dario', 1, 'Math', 20, 'Listen music', 5.0),

('Juan', 10001, 'Math', 30, 'Fishing', 1.3),

('Selene', 10003, 'Music', 18, 'Listen music', 4.2),

('Andres', 10004, 'Sports', 17, 'Read book', 4.5),

('Diego', 10005, 'Gym', 15, 'Watch movies', 2.2),

('Martin', 10006, 'Philosophy', 34, 'Fishing', 1.9),

('Sammuel', 10007, 'Statistics', 23, 'Swiming', 3.3),

('Miguel', 10008, 'Economy', 27, 'Listen music', 4.7),

('Maricel', 10009, 'Geometry', 10, 'Read book', 4.5),

('Aura', 10010, 'Geography', 31, 'Watch movies', 2.9),

('Robertulio', 10011, 'Math', 30, 'Fishing', 3.1),

('Stiven', 10012, 'Spanish', 21, 'Swiming', 4.8),

('Ehudes', 10013, 'Music', 18, 'Listen music', 4.5),

('Jose', 10014, 'Sports', 17, 'Read book', 1.5),

('Daniela', 10015, 'Gym', 15, 'Watch movies', 3.9),

('Juliana', 10016, 'Philosophy', 34, 'Fishing', 4.7),

('Olmedo', 10018, 'Economy', 27, 'Listen music', 3.8),

('Deyci', 10019, 'Geometry', 10, 'Read book', 2.9),

('Sandra', 10020, 'Geography', 31, 'Watch movies', 1.9),

('Felipe', 10021, 'Math', 30, 'Fishing', 1.7),

('Mariano', 10022, 'Spanish', 21, 'Swiming', 3.7),

('Marina', 10023, 'Music', 18, 'Listen music', 4.7),

('Milena', 10024, 'Sports', 17, 'Read book', 5.0),

('Cristina', 10025, 'Gym', 15, 'Watch movies', 2.2),

('Rosalba', 10026, 'Philosophy', 34, 'Fishing', 1.6),

('Orlando', 10027, 'Statistics', 23, 'Swiming', 3.9),

('Ximena', 10028, 'Economy', 27, 'Listen music', 4.6),

('Jhon', 10029, 'Geometry', 10, 'Read book', 3.5),

('Patricia', 10030, 'Geography', 31, 'Watch movies', 2.8),

('Nelly', 10031, 'Math', 31, 'Fishing', 3.7),

('Isabela', 10032, 'Spanish', 30, 'Swiming', 4.8),

('Janeth', 10033, 'Music', 21, 'Listen music', 5.0),

('Mauricio', 10034, 'Sports', 18, 'Read book', 1.9),

('Rodolfo', 10035, 'Gym', 17, 'Watch movies', 3.9),

('Hector', 10036, 'Philosophy', 15, 'Fishing', 4.4),

('Esneira', 10037, 'Statistics', 34, 'Swiming', 5.0),

('Daliana', 10038, 'Economy', 23, 'Listen music', 3.3),

('Hanna', 10039, 'Geometry', 27, 'Read book', 2.1),

('Camila', 10040, 'Geography', 10, 'Watch movies', 1.8),

('Daniel', 10002, 'Spanish', 21, 'Swiming', 3.3)

1. Write a query to get the average of the GPA from all the students which name starts with ‘A’.

R/

Select AVG(GPA) AS Average

from Student

where GPA > 4.7

1. Write a query to get the list of students that are in the same class, have a GPA higher than 3.5/5.0 and order them by Age and Name.

R/

SELECT name, age from Student

where GPA BETWEEN 3.5 and 5.

ORDER by age, name

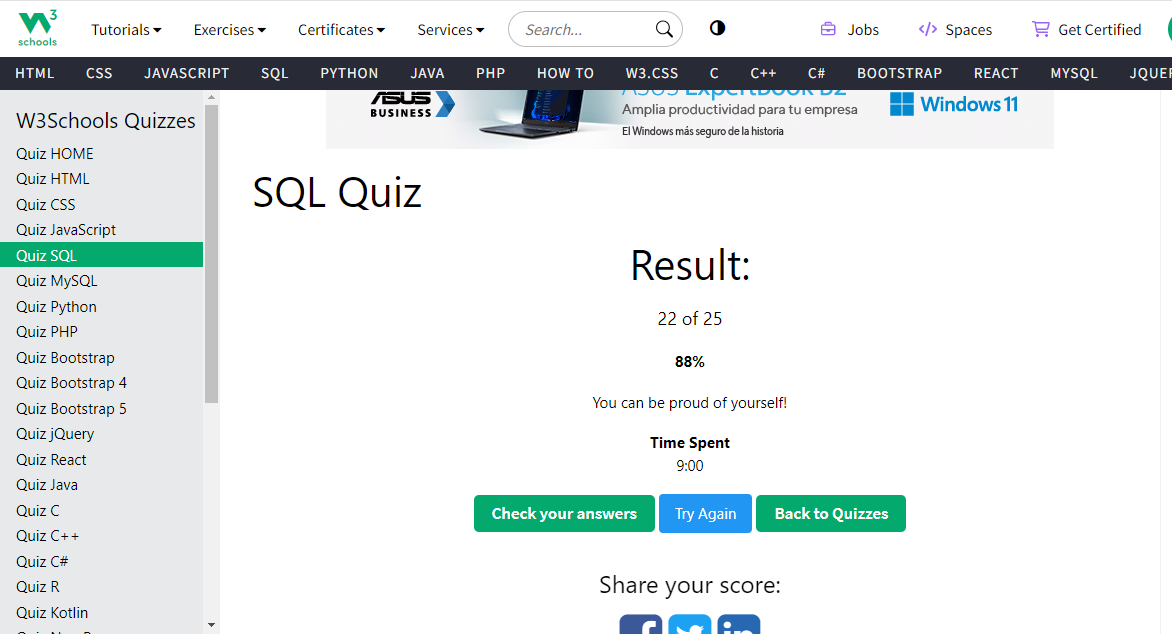
1. Write a query to get the list of all the students with ‘Name, ‘Code, ‘Class’, ‘Age’, ‘Favorite Subject, ‘GPA’.

R/

Select \* from Student

1. Take the following 25 question quiz about SQL, please include a screenshot about the results and time it took to take the test.

<http://www.w3schools.com/quiztest/quiztest.asp?qtest=SQL>



Part 3 (Software Testing Knowledge):

1. What is the difference between a unit test, an acceptance test, an integration test and an end-to-end test?

Unit test is called the white box and its goal is to make sure that a specific module works under all condition required.

The Acceptance test its goals is quality assurance process that determines if the requirements of a specification or contract are met.

The integration test its goal is to make sure that the interface between modules are doing it expect.

And The end-to-end test its goal is simulate a real user scenario making double-check verifying if everything works.

1. Could you explain Cohn's automation pyramid?



The Base Layer or the Unit Test its goal is give the programmer very specific information about the origins of a bug, up to the exact line of code where a failure occurs.

The Middle layer or API tests its goal is for those tests that exceed the scope of unit tests it is strongly advised to use tests that communicate with the application under test at the service or API level.

The top layer or GUI Test is where UI-Level automated tests reside, they are often the most brittle and take the longest time both in test case development and in test execution

1. Could you explain the difference between a black box testing and a white box testing?

R/

The difference between both are that Black box is a software testing methodology where a tester analyzes application functionality without a through knowledge of its internal design, however the white box is a software testing methodology where a tester with knowledge of the application internal working is leveraged during testing.

1. What is the purpose of an exploratory test and when is it useful to run them?

R/

Exploratory testing is often described as software testing for continuos learning on test design, and execution. Its goal is to allow of the individual tester to uncover defects that are not easily covered in the scope of other tests.

This process should always be done

1. Mention at least 5 test design techniques and explain them briefly
   1. Equivalent Class Partitioning.

The equivalent class partitioning implies splitting test data into classes, where all elements are similar in some way. This technique makes sense only if the components are similar and can fit in a common group.

* 1. Boundary Value Analysis

The boundary value analysis is similar to the previous technique. Some may even say it is based on the equivalent class partitioning.

* 1. State Transition

The state transition visualizes the states of a software system at different time frames and stages of usage. This technique is effective for creating test suites for systems that have many state variations. It will be helpful if you test a sequence of events with a finite number of input options

* 1. Pairwise Testing

The pairwise testing is considered the most difficult and confusing of the five test design techniques. And there is a good reason for this. The pairwise testing is based on mathematical algorithms, namely combinatorics

* 1. Error Guessing.

Error guessing is the most experimental practice of all, usually applied along with another test design technique. In error guessing, a QA engineer predicts where errors are likely to appear, relying on previous experience, knowledge of the system, and product requirements

1. What is the purpose of the following types of tests?
   1. Functional test: The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.
   2. Performance test: Its purpose is make test to determine how a system performs in terms of responsiveness and stability under a particular workload.
   3. Security test: Its purpose is reveal flaws in the security mechanisms of an information system that protect data and maintain functionality as intended.
   4. Usability test: its purpose is to evaluating a product or service by testing with a represent user who will try complete typical task and will wait the wish results
   5. API test: The purpose of API Testing is to check the functionality, reliability, performance, and security of the programming interfaces
   6. Unit Test: is a type of software testing where individual units or components of a software are tested.

References:

* <https://www.softwaretestinghelp.com/the-difference-between-unit-integration-and-functional-testing/>
* <https://stackoverflow.com/questions/7672511/unit-test-integration-test-regression-test-acceptance-test>
* <https://www.softwaretestinghelp.com/what-is-end-to-end-testing/>
* <https://www.testingexcellence.com/exploratory-testing-important-agile-projects/>
* <https://www.guru99.com/exploratory-testing.html>
* <http://softwaretestingfundamentals.com/differences-between-black-box-testing-and-white-box-testing/>
* <https://smartbear.com/solutions/api-testing/>
* <http://softwaretestingfundamentals.com/security-testing/>
* <https://www.guru99.com/what-is-security-testing.html>
* <https://www.experienceux.co.uk/faqs/what-is-usability-testing/>
* <https://en.wikipedia.org/wiki/API_testing>
* <https://en.wikipedia.org/wiki/Application_programming_interface>
* <https://searchsoftwarequality.techtarget.com/definition/performance-testing>
* <https://www.guru99.com/performance-testing.html>
* <https://www.tutorialselenium.com/2017/05/28/como-usar-selenium-ide/>
* <https://cleventy.com/tutorial-selenium-primeros-pasos/>
* <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/381>
* <https://www.tutorialselenium.com/2017/09/24/como-usar-comandos-de-selenium-ide/>
* <https://testeandosoftware.com/selenium-comandos-selenese/>
* <https://www.seleniumhq.org/selenium-ide/docs/en/introduction/getting-started/>
* <http://www.cs.tau.ac.il/~amiramy/SoftwareSeminar/CTDmay2012.PDF>
* <https://www.guru99.com/decision-table-testing.html>
* <https://www.toolsqa.com/software-testing/decision-table-testing/>
* <https://www.tutorialspoint.com/software_testing_dictionary/data_flow_testing.htm>
* <https://www.javatpoint.com/data-flow-testing-in-white-box-testing>
* <https://www.thedigitalmentor.com/what-is-dataflow-testing/>
* <https://en.wikipedia.org/wiki/Boundary-value_analysis>
* <https://www.guru99.com/equivalence-partitioning-boundary-value-analysis.html>
* <https://www.testingexcellence.com/boundary-value-analysis/>
* <http://www.professionalqa.com/combinatorial-testing>
* <https://en.wikipedia.org/wiki/Classification_Tree_Method>
* <https://www.expleo-germany.com/en/products/testona/classification-tree-method/>
* <https://inf.mit.bme.hu/sites/default/files/materials/taxonomy/term/445/13/04_Testing.pdf>
* <https://www.bcs.org/upload/pdf/amettehass-131211b.pdf>
* <https://www.bcs.org/upload/pdf/amettehass-131211a.pdf>
* <https://www.tutorialspoint.com/software_testing_dictionary/pdf/test_case_design_technique.pdf>
* <https://testautomationresources.com/software-testing-basics/software-test-design-techniques/>
* <https://www.uio.no/studier/emner/matnat/ifi/INF3121/v18/forelesningsvideoer/chapte-4-part-2-slides.pdf>
* <http://tryqa.com/what-is-test-design-technique/>
* <http://tryqa.com/what-is-structure-based-technique-in-software-testing/>
* <http://tryqa.com/what-is-black-box-specification-based-also-known-as-behavioral-testing-techniques/>
* <http://tryqa.com/what-is-white-box-or-structure-based-or-structural-testing-techniques/>