Software & Data Quality Engineer Assessment

The assessment involves creating scripts to identify technical issues, generating test cases and visualizing the results in a Power BI dashboard.

1. Web Crawling

Create a Python script using libraries like BeautifulSoup, requests, or Scrapy to crawl the entire softwarefinder.com website.

Requirements

- Identify and log all broken links (HTTP 404 or other error codes).
- Detect script errors (JavaScript errors) on each page.
- Check for responsiveness issues (media query errors, div misalignments, etc.).
- Save all results in a structured CSV file with the following columns: Page URL, Broken Links, Script Errors, Responsiveness Issues, Timestamp

2. Test Case Creation and Execution

Develop automated test cases for the following functionalities:

Search Functionality

- Test search results for accuracy and relevance.
- Check for edge cases (e.g. empty search, special characters).

Filtering

- Test filtering options (e.g. by category, ratings).
- Ensure filters work correctly and update results dynamically.

Load Testing

- Simulate high traffic on the website using tools like Locust or JMeter.
- Measure response times, server load and potential crashes under stress.

Requirements

- Use Selenium or Pytest for automated testing.

- Save test results in a CSV file with columns: Test Case, Status (Pass/Fail), Error Message (if any), Timestamp

3. Data Visualization and Reporting

Create a Power BI dashboard to visualize the QA results.

Requirements

Import the CSV files generated from the web crawling and test case execution. Create charts and graphs to show:

- Total broken links per page.
- Distribution of script errors.
- Responsiveness issues across different devices (mobile, tablet, desktop).
- Search and filtering test results.
- Load testing performance metrics (response times, error rates).

4. Automation and Rerun Mechanism

Implement a mechanism to rerun all tests and update the reports automatically.

Requirements

Create a Python script or use a task scheduler (e.g. Airflow) to:

- Rerun the web crawler and test cases at a scheduled interval.
- Automatically update the CSV files and refresh the Power BI dashboard.
- Ensure the process is fully automated and requires minimal manual intervention.

Deliverables

Submit a single document containing the links to following:

Python Scripts Repo

- Web crawler for broken links, script errors and responsiveness issues.
- Automated test cases for search, filtering and load testing.
- Automation script for rerunning tests and updating reports.

CSV Files

Results from web crawling and test case execution.

Power BI Dashboard

Link to interactive dashboard visualizing all metrics.

Screen Recording

A brief report explaining the task, tools used and any challenges faced.

Instructions on how to run the scripts and update the dashboard.

Evaluation Criteria

- Code Quality: Clean, well-documented and efficient code.
- Accuracy: Correct identification of broken links, script errors and responsiveness issues.
- Automation: The rerun mechanism works seamlessly and updates the dashboard.
- Visualization: Power BI dashboard is intuitive, informative and visually appealing.
- Completeness: All required test cases are implemented and executed.

Deadline

Please submit your completed assessment within 4 days starting from the day you received this assessment.

If you have any questions or need clarification, feel free to reach out. Good luck! 😊