Technical QA Test

Technical QA Test

Introduction

Welcome to the technical test for the QA Engineer position. This test consists of three practical exercises that will evaluate your skills in test planning, execution, automation, regression testing, bug reporting, performance testing, and collaboration. The theme of the exercises is sustainability, focusing on a cloud-based SaaS application that promotes environmental awareness and actions.

Each exercise contains multiple parts, with a total score of 10 points. Ensure that your solutions are well-documented and clear.

Tools and Environment

For this test, you will need access to:

- A local development environment (e.g., Python, Excel)
- Pytest for test automation

Technical QA Test

Exercise 1: Test Planning and Case Design (2 points)

Context

You are part of a team developing a local version of a cloud-based application called "EcoTrack". EcoTrack helps users track their carbon footprint and provides tips for reducing it. The application has a dashboard that displays daily, weekly, and monthly carbon footprint data.

Dashboard Specifications

The EcoTrack dashboard includes the following components:

1. Carbon Footprint Summary:
   - Displays total carbon footprint for the day, week, and month.
   - Shows a breakdown of carbon footprint by category: energy consumption, transportation, and waste.
2. Trends Graph:
   - Line chart showing the carbon footprint trend over the last 30 days.
   - Allows users to filter by category (energy, transportation, waste).
3. Comparison Chart:
   - Bar chart comparing the user's carbon footprint with the average user in the same region.
   - Displays percentage difference for each category.
4. Tips and Recommendations:
   - List of personalized tips to reduce carbon footprint based on user's activity.
   - Links to detailed articles and resources.
5. Achievements and Badges:
   - Shows badges earned by the user for completing certain eco-friendly actions.
   - Progress towards the next badge.

Technical QA Test

Task

1. Test Planning (1 point):

- Design a comprehensive test plan for the EcoTrack dashboard. Your plan should include the following:

- Test objectives

- Test scope

- Test approach

- Test environment

- Test schedule

- Deliverables

Include both functional and non-functional testing aspects.

2. Test Case Design (1 point):

- Create five detailed test cases for the EcoTrack dashboard based on the dashboard specifications. Each test case should include:

- Test case ID

- Test description

- Pre-conditions

- Test steps

- Expected results

- Post-conditions

Example Test Cases:

- Test Case 1: Verify that the carbon footprint summary displays accurate data for the day, week, and month.

- Test Case 2: Ensure that the trends graph correctly updates when filtering by category.

Deliverables

- A document containing the test plan.

- A document with five detailed test cases.

Technical QA Test

Exercise 2: Test Automation and V-Cycle Testing with Pytest (4 points)

Context

EcoTrack has a new feature: "EcoChallenges". Users can participate in daily challenges to reduce their carbon footprint, and their progress is tracked in real-time.

Task

1. Unit Tests Development (1.5 points):

- Develop unit tests using Pytest for the `CarbonFootprintCalculator` class provided below. The unit tests should cover various scenarios and edge cases.

```python
class CarbonFootprintCalculator:
    def __init__(self, energy_consumption, transportation, waste):
        self.energy_consumption = energy_consumption
        self.transportation = transportation
        self.waste = waste

    def calculate_energy_footprint(self):
        return self.energy_consumption * 0.233

    def calculate_transportation_footprint(self):
        return self.transportation * 0.21

    def calculate_waste_footprint(self):
        return self.waste * 0.05
```

```
    def total_carbon_footprint(self):
        return (self.calculate_energy_footprint() +
                self.calculate_transportation_footprint() +
                self.calculate_waste_footprint())
```

2. Integration Tests Development (1.5 points):

- Develop integration tests using Pytest for the following functions that test the interaction between different methods of the `CarbonFootprintCalculator` class.

```python
def get_user_carbon_footprint(user_data):
    calculator = CarbonFootprintCalculator(
        user_data['energy_consumption'],
        user_data['transportation'],
        user_data['waste']
    )
    return {
        'energy_footprint': calculator.calculate_energy_footprint(),
        'transportation_footprint': calculator.calculate_transportation_footprint(),
        'waste_footprint': calculator.calculate_waste_footprint(),
        'total_footprint': calculator.total_carbon_footprint()
    }
```

```
  def get_comparison_with_average(user_footprint, average_footprint):
      return {
          'energy_difference': user_footprint['energy_footprint'] - average_footprint['energy_footprint'],
                                  'transportation_difference': user_footprint['transportation_footprint'] - average_footprint['transportation_footprint'],
          'waste_difference': user_footprint['waste_footprint'] - average_footprint['waste_footprint'],
          'total_difference': user_footprint['total_footprint'] - average_footprint['total_footprint']
      }
```

3. Acceptance Tests Development (1 point):

- Develop pseudocode for acceptance tests for the `EcoChallenges` feature. These tests should verify that the feature works as intended from a user's perspective, ensuring it meets the requirements and acceptance criteria.

Deliverables

- A Python script containing unit tests using Pytest.
- A Python script containing integration tests using Pytest.
- A document with pseudocode for acceptance tests.

Technical QA Test

Exercise 3: Error Analysis and Dashboard Creation (4 points)

Context

EcoTrack's user base is rapidly growing, and the application collects a significant amount of data, including logs of errors occurring in various services.

Task

1. Data Analysis and Error Identification (1 point):

- Use the provided Excel sheet with logs of errors from EcoTrack. Analyze the data to identify the most common errors and the services causing them.

[Download EcoTrack Error Logs](sandbox:/mnt/data/ecotrack_error_logs.xlsx)

2. Dashboard Creation in Excel (2 points):

- Create a dashboard in Excel to visualize the analysis. The dashboard should include charts and graphs to display:
  - The frequency of different types of errors
  - The services causing the most errors
  - Trends over time

3. Service Analysis and Reporting (1 point):

- Based on your analysis, identify the service that causes the most errors. Provide a detailed report including:
  - The nature of the errors
  - Possible reasons for the errors
  - Suggestions for addressing the issues

Technical QA Test

Deliverables

- An Excel file with analysis and a dashboard.

- A report detailing the service analysis and suggestions.

Technical QA Test

Submission

Please submit your test plan, test cases, Python scripts with Pytest tests, Excel analysis, and all related documentation, screenshots, and logs. Ensure your submissions are well-organized and easy to follow.

Good luck with your test!