# Testing Interview Questions – Answers

## 1. Explain how you personally used regression, integration, and acceptance testing

* **Regression testing:** I used regression testing to verify that changes or new features didn’t break existing functionality. For example, after updating the Employee Profile module in our HR system, I ran regression tests to ensure previous workflows like employee creation and document upload still worked correctly.
* **Integration testing:** I performed integration testing to check how different modules interacted. For instance, when adding a new employee, I tested that the data flows correctly from the Employee module to Payroll, Leave, and Tax modules.
* **Acceptance testing:** I conducted acceptance testing to ensure the system met the client’s requirements. This involved validating that features like employee onboarding, document uploads, and ID verification functioned as expected from an end-user perspective before releasing to production.

## 2. You see the same bug reappear multiple times across sprints. What do you do?

1. Investigate why previous fixes didn’t fully resolve it (review code, environment, or test coverage).
2. Communicate with developers to implement a more permanent solution.
3. Update regression and automated tests to cover the scenario so the bug is detected early.
4. Document the issue clearly in the bug tracker with root cause analysis to prevent it from recurring.

## 3. What are some common causes of flaky automated tests, and how would you reduce them?

**Common causes:**

* Timing issues (elements not ready or dynamic content delays).
* Dependencies on external systems like databases, APIs, or network services.
* Tests depending on previous tests’ state (not independent).
* Hardcoded test data or environment assumptions.

**How to reduce them:**

* Use explicit waits or Playwright’s built-in wait methods for elements.
* Mock or stub external services where possible.
* Ensure tests are independent and can run in any order.
* Use dynamic or isolated test data for each run.
* Regularly review and refactor test code to maintain stability.