Summary on Integration Testing

Module Summary

PW SKILLS

1. Integration Testing Frameworks (Selenium, Cypress, etc)

- **Selenium:** A popular tool for automating web browser testing. It supports multiple programming languages (Java, Python, C#, etc.) and browsers, making it ideal for end-to-end and integration tests in web applications.
- **Cypress:** A modern web testing framework, known for its fast setup and intuitive interface. It provides automatic waiting, real-time reloads, and can run integration tests directly in the browser.
- **TestNG:** A Java testing framework that supports integration testing, unit testing, and functional testing with robust test configuration options.
- **Postman:** Mainly used for API integration testing. It allows users to test RESTful APIs through an easy-to-use interface and supports automation of requests.

2. Test-Driven Development (TDD) and Behavior-Driven Development (BDD)

a. Test-Driven Development (TDD):

- A development methodology where tests are written before the code itself.
- **Steps:**
- Write a failing test.
- Write the minimal code to pass the test.
- Refactor the code for optimization.
- It improves code quality, forces developers to think about edge cases early, and ensures high test coverage.

b. Behavior-Driven Development (BDD):

- An extension of TDD that focuses on the behavior of the application rather than its implementation.
- Tests are written in plain language that non-developers (e.g., business stakeholders) can understand.
- Tools like Cucumber and RSpec are often used to write these human-readable scenarios.
- **Example BDD Scenario:**

```
Given the user is on the login page
When they enter valid credentials
Then they should be redirected to the dashboard
```

3. Code Coverage and Quality Analysis Tools

1. Code Coverage Tools:

- **Coverage.py:** For Python, tracks how much of the codebase is covered by tests.
- **JaCoCo:** Java tool that measures code coverage at the class, method, and line levels.
- **Istanbul:** For JavaScript, provides test coverage analysis and visualization in various formats (e.g., HTML, LCOV).

2. Code Quality Analysis Tools:

- **SonarQube:** A platform for continuous inspection of code quality, detecting bugs, security vulnerabilities, and code smells across multiple languages.
- **ESLint:** A linter for JavaScript/TypeScript to ensure adherence to coding standards.
- **Pylint:** Python static code analysis tool that detects coding errors, enforces PEP 8, and identifies potential issues.