## Project Overview:

The testing project focuses on evaluating the UI and functionality of the Amazon website using Selenium. The Page Object Model (POM) is employed to enhance maintainability and readability.

### Framework Structure:

1. Utilizes an abstract class, PageOutline, defining the structure for page classes, with common methods for UI interaction.

2. BasePage class extends PageOutline, providing concrete implementations for common methods, handling elements, and waiting for element presence.

3. Child classes (e.g., SignIn, SignUp, ProductSearch) extend BasePage to implement specific functionalities.

### Testing Architecture:

1. Project is organized into folders for functional, integration, and browser compatibility testing.

2. TestNG is employed for test execution, and XML files define test suites, specifying browsers, listeners, and test classes.

### XML Configuration:

1. Centralized CombinedSuites.xml includes separate suites for functional, compatibility, and database testing, facilitating easy configuration changes.

2. TestNG parameters in XML files define browser types, enabling flexibility in test execution.

### Allure Listeners:

Utilizes Allure listeners for reporting, enhancing visibility into test results.

### Custom Annotations:

Introduces a custom annotation, @ParameterLabel, to provide custom names for test case parameters, improving report readability.

### Parameterized Tests:

1. Test methods leverage custom annotations for parameter labels, enhancing the clarity of test reports.

2. Custom method captures parameters and their labels for reporting purposes.

### Screenshots and Reporting:

Screenshots are captured during test execution using the captureScreenshot() method, providing insights into intermediate states during integration testing.

### Data Management:

1. Utilizes CSV files for test data, facilitating parameterized testing with clear separation based on test method names.

2. Implements an efficient algorithm to pass correct test cases to respective methods, ensuring scalability as new test cases are added.

### Reflections and Enhancements:

1. Demonstrates thoughtful design and organization, enhancing maintainability and scalability.

2. Allows easy adaptation to changes in browser types and test scenarios through XML configuration.

3. The project showcases a systematic approach to UI and functional testing, leveraging industry best practices.

### Bug Identification:

**1. Invalid Mobile Number Handling:**

Entering a mobile number with 13 or 14 digits results in OTP being sent instead of displaying an alert message for an invalid number.

**2. Invalid Username Handling:**

Entering an invalid username with special characters such as "..." or "@@" allows the signup process to proceed to the next stage without proper validation.

These identified bugs highlight issues in the signup page's validation logic, specifically related to handling invalid mobile numbers and usernames with special characters. Addressing these issues is crucial for ensuring a secure and user-friendly signup process.