

# TEST OUT OF SYNC ON PROD POST 1

## Contents

---

TEST OUT OF SYNC ON PROD POST 1 TEST OUT OF SYNC ON PROD POST 1 TEST OUT OF SYNC ON PROD POST 1 TEST OUT OF SYNC ON PROD POST 1 TEST OUT OF SYNC ON PROD POST 1

## Types Based on Testing Purpose

### Functional Testing

[Functional Testing](#) verifies what the system does by ensuring that features and functionalities work according to the specified requirements.

- [Smoke Testing](#): Performs basic checks on critical functionalities to ensure the application is stable for further testing.
- [Sanity Testing](#): Verifies specific functionalities after minor changes or bug fixes to ensure they work as expected.
- [Integration Testing](#): Tests the interaction between different modules to ensure they work together correctly.
- [System Testing](#): Tests the complete application as a whole to validate end-to-end functionality.
- [Regression Testing](#): Re-tests existing functionalities after changes or bug fixes to ensure no new defects are introduced.
- [User Acceptance Testing \(UAT\)](#): Conducted by end users or clients to verify that the software meets business requirements.

### Non-Functional Testing

[Non-Functional Testing](#) evaluates how well the system performs in terms of usability, performance, security, and compatibility.

- [Usability Testing](#): Evaluates the application's ease of use, user interface, and overall user experience.
- [Compatibility Testing](#): Verifies that the application works correctly across different browsers, devices, and operating systems.
- [Performance Testing](#): Measures the application's speed, responsiveness, stability, and behavior under different workloads.
- [Security Testing](#): Identifies vulnerabilities and ensures the protection of sensitive data and system resources.

## Manual Testing Process

The following diagram provides a detailed representation of the steps in the manual testing process.

### Manual Testing Process

- **Requirement Analysis:** Understand and analyze the project requirements, user expectations, and business needs to identify what needs to be tested.
- **Test Plan Creation:** Create a test plan that defines the testing scope, objectives, resources, schedule, and testing approach.
- **Test Case Creation:** Design detailed test cases with test steps, expected results, and test data to validate application functionality.
- **Test Environment Setup:** Prepare the required hardware, software, tools, and test data to simulate the production environment.
- **Test Case Execution:** Execute the test cases and compare actual results with expected results to identify any issues.
- **Defect Logging:** Record defects with complete details such as description, severity, reproduction steps, and supporting evidence.
- **Defect Fix & Reverification:** Verify that reported defects have been fixed correctly and confirm that the issue no longer exists.
- **Regression Testing:** Retest affected and related functionalities to ensure that recent changes have not introduced new defects.
- **Test Closure Report:** Prepare a final report summarizing testing activities, results, defect status, and key lessons learned.

## Tools Used for Manual Testing

- **TestRail:** A test management tool used to organize, execute, and track manual test cases with detailed reporting features.
- **Xray (for Jira):** A Jira-integrated testing tool that supports manual test management, traceability, and defect tracking.
- **Qase:** A cloud-based test management platform that simplifies test case creation, execution, and team collaboration.
- **Zephyr:** A test management solution that helps manage manual testing activities with strong Jira integration and reporting.
- **Tuskr:** A lightweight cloud-based test management tool for creating, managing, and tracking manual test cases efficiently.