Test Approach for “Shop Apotheke” Web App

1. Objective:

The objective of this testing approach is to ensure the functionality, usability, performance & security of the “Shop Apotheke” web app and automate the app parts suitable, qualified & ready for automation. The testing process will cover various scenarios to identify and rectify defects before the application’s release to users.

2. Scope:

This testing approach covers functional, usability, performance security & automation testing of the App across multiple browsers (Google Chrome, Mozilla Firefox, Safari & Microsoft Edge), devices (desktops, tablets & mobile phones) and operating systems (MacOS, Windows, Linux, iOS & Android).

3. Testing Phases:

3.1. Requirements Analysis:
- Review and analyze the project requirements, specifications, design documents.
- Consider comparing the app to other similar apps (Online pharmacies) or near-similar apps (Online shopping in general) in the market.
- Consider any prior/personal experience of the Tester.
- Identify test scenarios, acceptance criteria, and potential risks.

3.2. Test Planning:
- Develop a detailed test plan outlining the testing scope, objectives, resources, schedule, and entry/exit criteria.
- Define the test environment, including hardware, software, and network configurations.
- Identify (in details) the testing tools and techniques to be used.

3.3. Test Design:
- Develop high-level test cases for functional, usability, performance & security testing (taking into consideration the priority & severity for each).
- Depending on the time availability, develop detailed test cases for the (in scope) testing types.
- Create test data and identify preconditions for test execution.
- Design regression test suites for future releases & define their readiness for Automation.

3.4. Test Execution:
- Execute test cases based on priority and severity.
- Log defects for any deviations from expected behavior.
- Perform regression testing (through test automation scripts) after bug fixes or feature enhancements.

3.5. Usability Testing:
- May be conducted during the (Test Execution)
- Conduct usability testing with real users to evaluate the application’s user interface, navigation, and overall user experience.
- Gather feedback to make necessary improvements.

3.6. Performance Testing:
- Conduct load testing (after the app functional verification) to assess the application’s behavior under normal and peak load conditions.
- Identify performance bottlenecks and optimize the app for scalability.

- Perform regression testing (through test automation scripts) after the performance optimization.

3.7. Security Testing:
- Perform security testing (after the app functional verification) to identify vulnerabilities according to the “OWASP Top 10” list of security risks.
- Implement security best practices and encryption techniques.
- Perform regression testing (through test automation scripts) after the security optimization.

4. Monitor & Reporting:
- Document test results, including test case execution status, defects found, and performance/security/usability issues.
- Review the test objectives and coverage against the initial requirements.
- Prepare test summary reports and assess the overall testing process.
- Conduct a lessons learned session to improve future testing processes.
- Provide detailed reports to stakeholders regularly, highlighting progress and any concerns.

5. Tools and Technologies:
- Version Control: Git/Github
- Test Management: Jira, TestRail [or other Test Management tool].
- Test Automation: Cypress.
- Performance Testing: Apache JMeter [or other paid license tools].
- Security Testing: OWASP ZAP, Burp Suite [or other paid license tools].
- Browsers: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.

6. Challenges and Risks:
- Availability of diverse devices and browsers for testing.
- Availability of the needed human resources.
- Effect of bug fixes and retesting on the time schedule.
- Network issues affecting test execution.

7. Conclusion:
This test approach aims to ensure the “Shop Apotheke” web app's quality, functionality, and security before its release. Regular collaboration, continuous feedback, and a systematic testing approach will contribute to the successful delivery of a reliable web app.