**Name – Pratap Yadav**

**Email-** [**pratapyadav100895@gmail.com**](mailto:pratapyadav100895@gmail.com)

**Mob. No - +91 8787020501**

**Assignment**

1. **What steps would you include in your test cases to validate the login functionality of the web application?**

**Ans:** To validate the login functionality of a web application, the test cases should cover various scenarios to ensure both positive and negative cases are handled correctly. For Validate the Login functionality of web application I have write some test cases including positive and negative endpoints which are attached with following link.

Link for Test cases - <https://docs.google.com/spreadsheets/d/1MolkgJNsBrphlP1m9J6KkcEtvxh4YkV9xFem61rJkSM/edit?usp=sharing>

1. **How would you test the application’s behavior when the login page is accessed with different browsers or devices?**

**Ans:** I would follow a following steps for test the application’s behavior:

**1.** Cross-Browser Testing

* Test on multiple browsers like Chrome, Firefox, Edge, Safari.
* Check if the login page works and looks the same across browsers.

**2.** Cross-Device Testing

* Test on devices like desktops, tablets, and smartphones.
* Ensure the login page is responsive and works on different screen sizes.

**3.** Performance Testing

* Measure page load speed on different browsers/devices.

1. **What are some common issues you might encounter during manual testing of the login feature, and how would you document them?**

**Ans:** Common issues during manual testing of the login feature and how to document them:

1. Invalid Error Messages: Document incorrect or missing error messages.
2. UI Issues: Note misaligned buttons or fields with screenshots.
3. Slow Performance**:** Record long load times and provide details.
4. Form Validation**:** Report issues with empty or incorrect field validations.
5. Cross-Browser Problems**:** Describe inconsistencies across different browsers.

Documentation:

* Title: Brief description of the issue.
* Steps to Reproduce: Actions to trigger the issue.
* Expected vs Actual Result: What should happen vs. what happened.
* Attachments: Include screenshots or logs.

1. **How do you handle unexpected behavior or issues found during manual testing?**

**Ans:** For handle unexpected behavior or issues found during manual testing I would follow the steps:

Steps to Handle Unexpected Issues:

1. Pause and Investigate: Reproduce the issue to confirm if it consistently occurs.
2. Document the Issue: Capture details such as steps to reproduce, expected vs actual behavior, environment (browser, device), and attach screenshots.
3. Prioritize the Issue: Assess the severity and impact on users e.g., critical, major, minor.
4. Report the Bug: Log the issue in the bug-tracking tool e.g., JIRA, Bugzilla with all relevant details.
5. Communicate with the Team: Discuss the issue with developers and the testing team to find the root cause and solution.
6. Retest**:** Once the issue is resolved, retest to ensure it’s fixed and didn’t introduce new bugs.
7. **What strategies would you use to ensure comprehensive manual testing coverage for the login feature?**

**Ans** To ensure comprehensive manual testing coverage for the login feature:

1. Test Cases: Cover valid and invalid login scenarios.
2. Edge Cases: Test boundary conditions like minimum/maximum input lengths.
3. Cross-Browser/Device Testing: Verify functionality on different browsers and devices.
4. Field Validation: Check input validations and error messages.
5. Security Testing: Test for vulnerabilities like SQL injection.

These strategies ensure thorough testing of the login feature.

1. **How would you create a new issue in JIRA for a defect found during manual testing?**

**Ans**: To create a new issue in JIRA for a defect:

1. Click "Create": Open the JIRA dashboard and click the "Create" button.
2. Select Issue Type: Choose "Bug" or relevant type.
3. Fill in Details:
   * Summary: Briefly describe the defect.
   * Description: Provide steps to reproduce, expected vs. actual results.
   * Attachments: Add screenshots or logs if needed.
4. Set Priority: Assign the severity of the issue.
5. Assign: Assign the issue to the relevant team member.
6. Click "Create": Save the issue to log it in JIRA.

This process ensures the defect is documented and tracked effectively.

1. **How can you link a JIRA issue to a specific test case or test execution?**

**Ans:** To link a JIRA issue to a specific test case or test execution:

1. Open the Test Case/Execution: Go to the test case or test execution in JIRA.
2. Find “Issue Links”: Locate the “Issue Links” section.
3. Click “Link”: Select “Link” to connect the test case or execution to an existing JIRA issue.
4. Choose Link Type**:** Select the type of relationship e.g., “blocks,” “relates to”.
5. Search and Select Issue: Enter the JIRA issue ID or summary and select it.
6. Click “Link” or “Save”: Complete the process to create the link.

This links the test case or execution to the relevant JIRA issue for traceability.

1. **What JIRA features can you use to track the progress of defect resolution?**

**Ans:** To track the progress of defect resolution in JIRA:

1. Issue Status: Check the current status e.g., Open, In Progress, Resolved.
2. Comments: Review comments for updates and discussions.
3. Activity Log**:** Monitor changes and updates made to the issue.
4. Linked Issues: View related issues or tasks.
5. Dashboards: Use widgets to visualize progress and metrics.
6. Filters/Reports: Generate reports and use filters to track defect progress.

These features help you monitor and manage the resolution of defects effectively.

1. **How would you configure JIRA to manage test cases and track their execution?**

**Ans:** To configure JIRA for managing test cases and tracking their execution:

1. Install a Plugin: Add test management tools like Zephyr or Xray to JIRA.
2. Create Test Cases: Use the new "Test" issue type to create and manage test cases.
3. Organize Tests: Group tests into Test Cycles or Test Plans for structured testing.
4. Track Execution: Record test results (Pass, Fail, Blocked) through Test Executions.
5. Link to Issues: Connect test cases to requirements (user stories, bugs) for traceability.
6. Use Reports/Dashboards: Monitor test progress with reports and dashboards.

This setup simplifies test management and tracking within JIRA.

1. **How can you generate and customize reports in JIRA to analyze testing progress?**

**Ans:** To generate and customize reports in JIRA:

1. Use Built-in Reports: Go to the Reports section and select a report type e.g., Test Execution Report.
2. Apply Filters: Adjust filters to focus on specific test cases or time periods.
3. Customize Layout: Configure the report’s layout and displayed data fields.
4. Create Dashboards**:** Add widgets to JIRA Dashboards to visualize test progress and metrics.

These steps help tailor reports to analyze testing progress effectively.

1. **What are the key performance metrics you would measure when testing the login functionality of the web application?**

**Ans:** Key performance metrics to measure when testing the login functionality:

1. Response Time: Time taken to log in after submitting credentials.
2. Page Load Time: Time for the login page to fully load.
3. Error Rate: Frequency of errors during login attempts.
4. Throughput: Number of successful logins per unit of time.
5. Resource Utilization: Server CPU and memory usage during login.
6. Concurrency: Number of users that can log in simultaneously without issues.

These metrics help ensure the login functionality performs efficiently under various conditions.

1. **How would you design a performance test for the login feature to simulate different user loads?**

**Ans:** To design a performance test for the login feature:

1. Define Objectives: Identify what you want to measure e.g., response time, error rate.
2. Create Test Scenarios: Simulate different user loads e.g., 100, 500, 1000 concurrent users.
3. Use Performance Testing Tools: Employ tools like JMeter or LoadRunner.
4. Script the Test: Create test scripts to automate the login process.
5. Run Tests: Execute the test scenarios with varying loads.
6. Analyze Results: Review metrics like response time and error rates under different loads.

This approach helps assess how the login feature performs under different user volumes.

1. **What tools would you use for performance testing, and how do you configure them for the login scenario?**

**Ans:** Tools for Performance Testing:

1. Apache JMeter:
   * Configure Test Plan: Set up HTTP requests for the login endpoint.
   * Set Load: Define thread groups to simulate multiple users.
   * Add Listeners: Include listeners to track metrics like response time and errors.
2. LoadRunner:
   * Record Script: Capture the login process with VuGen.
   * Set User Load: Define virtual users and load scenarios.
   * Analyze Results: Use the Analysis tool to review performance data.

These tools help simulate user loads and measure login performance.

I have added the Login Page Test results in the CSV format using Apache JMeter and Blazemeter record of .jmx file I have test Login Page of <https://opensource-demo.orangehrmlive.com/> Performance test.

Following is a link for login page test result: [LoginTestResult.csv](https://1drv.ms/x/c/940ef830eaca71b4/EQxPcPgnchJMib8rrvh8U9IBEYijDM3zkdurC_w8pyMK6Q?e=2g7W3Q)

1. **How do you interpret performance testing results to identify bottlenecks in the login feature?**

**Ans:** To interpret performance testing results and identify bottlenecks:

1. Analyze Response Times: Check if login response times are higher than acceptable thresholds.
2. Check Error Rates: Look for increased error rates, which can indicate issues.
3. Examine Throughput: Assess if the number of successful logins matches expected performance under load.
4. Review Resource Utilization: Look at CPU, memory, and network usage for spikes or limits.
5. Identify Slow Components: Pinpoint stages in the login process that are slow or failing.

These steps help identify where performance issues may be occurring.

1. **What actions would you take if the performance testing reveals that the login feature cannot handle the expected load?**

**Ans:** If performance testing shows the login feature can't handle the expected load:

1. Analyze Results: Identify specific bottlenecks or failure points.
2. Optimize Code: Improve the login code for better efficiency.
3. Increase Resources: Scale up server capacity or optimize database performance.
4. Improve Architecture: Consider load balancing or caching solutions.
5. Retest: Conduct performance tests again to ensure the changes have resolved the issues.

These actions help enhance the login feature’s performance to meet expected load requirements.