QA Strategy Document for “Priority Tagging” feature

## 1. Introduction

The TodoMVC application plans to introduce a "Priority Tagging" feature, allowing users to assign priority levels (High, Medium, Low) to their todos.

## 2. Objectives

- Verify that users are capable of assigning priority tags to tasks in an efficient manner.

- Verify that users are sort and filter the tags

- Validate that the tags are consistent throughout the application

- Verifying the feature's usability and user experience.

- Ensure compatibility with various devices, browsers, and platforms.

## 3. Scope

- The functional requirements of the tags – High, Medium, Low

- The behavior of application after removing the tag from the task

- Browser compatibility scope should be limited to – chromium and safari, Other browsers are not In the scope

- Device compatibility scope should be limited to desktop web application, Mobile web application is not in the scope.

## 4. Test Approach

QA Team will follow the approaches that includes -

- Feature Required Manual Functional Testing on the deployed env, to validate the user experience

- Automated regression testing followed by manual testing required to check the effects on existing functionalities.

## 5. Test Environment

- Various devices to test the feature with Chromium and Safari browsers.

- Test data with users’ perspective.

## 6. Test Types

- Functional testing – To verify the tag assignment to the task along with different browsers and devices

- Usability testing – To validate the user experience and display of the feature.

- Regression testing – To validate the impact of Priority Tag feature with existing feature.

- Compatibility testing – To verify the feature compatibility with desktop device.

## 7. Test Tools

List the tools to be used for testing, including:

- Test management tools – JIRA for test case creation, bug reporting and tracking

- Automation tools – Vs code, Playwright

-Compatibility testing tools – Lambda-test

## 8. Test Execution

- Test case creation based on the requirement.

- Test case execution as per the plan.

- Defect reporting and tracking for the identified issue.

- Test case identification for automation.

- Automation script writing for the identified test cases.

- Regression testing to ensure no new issue introduced.

- Signing off the feature under test.

## 9. Roles and Responsibilities

- Testers - Execution of test cases and reporting of defects.

- Developers - Assistance in defect triaging and resolution.

- Product owners/stakeholders – Approval of test results.

## 10. Metrics and Reporting

- Test coverage report status.

- Defect metrics along with severity and priority issues.

- Test execution progress report, milestone achieved.

## 11. CI/CD Integration

- Use the GitLab or GitHub tool for CI/CD Integration.

- Developer Commit the feature code.

- Build gets triggered and Unit Test Cases are executed.

- Deployment of feature under test will do and feedback of the build will be provided.

- Automated regression test will be executed.

- QA approval required to make the build on production.

## 12. Assumptions

- The existing features are tested and has no critical bug.

- The feature is approved by Product Owner.