

STD Document

Mohammad khayyo



Table of Contents

Purpose of this document 3

Scope 3

Goals 4

Glossary 5

Tests Tree 6

Purpose Of This Document

The purpose of this document is to delineate a comprehensive test automation strategy specifically tailored for Monday.com's website, utilizing Selenium Grid with Python. This strategy is designed to ensure a systematic approach to validating the functionality, performance, reliability, and user experience of the Monday.com platform. By defining the scope, methodologies, resources, and timelines for the testing activities, this document aims to serve as a roadmap for the QA team. It facilitates effective planning, execution, and management of test cases, enabling the identification and resolution of potential issues before they impact end-users. Additionally, this document will act as a communication tool among project stakeholders, providing clear insights into the testing process, objectives, and outcomes. The ultimate goal is to uphold and enhance the quality and performance of Monday.com, thereby assuring a seamless, efficient, and productive experience for its users.

Scope

Crafting the developmental roadmap for the Monday.com platform involves a meticulous focus on pivotal goals, encompassing the refinement of user account administration, fostering seamless task and project collaboration, implementing workflow automation, across diverse platforms, establishing integrative capabilities with external tools, optimizing data visualization, fortifying security measures, enhancing communication features, instituting user feedback mechanisms, expanding customization options, ensuring platform compatibility. This strategic approach is geared towards elevating the overall user experience and functionality of the Monday.com platform.

Goals

Comprehensive Test Coverage:

Goal: Develop STD files to provide comprehensive coverage of Monday.com's functionalities, ensuring that critical workflows and features are thoroughly tested.

Error Detection and Reporting:

Goal: Create automated test scenarios within the STD files to effectively identify and report errors or anomalies in the system behavior, providing detailed information for efficient debugging.

Cross-Browser Compatibility:

Goal: Implement STD files to validate and ensure that the Monday.com website functions seamlessly across various web browsers, guaranteeing a consistent user experience for all users.

Security Testing:

Goal: Integrate security-focused STD files to systematically test the platform for vulnerabilities, ensuring that Monday.com adheres to security best practices and safeguards user data.

User Workflow Validation:

Goal: Create STD files that mimic real user workflows to validate end-to-end scenarios, ensuring that users can smoothly navigate through key features and complete tasks without hindrance.

Documentation and Reporting:

Goal: Implement STD files that generate comprehensive test documentation and reports, providing clear insights into test results, including passed tests, detected issues, and overall system health.

Glossary

Monday.com:

The project management platform that serves as a digital workspace for teams to coordinate, collaborate, and manage tasks and projects efficiently.

Digital Workspace:

An online environment provided by Monday.com where teams can plan, organize, and execute tasks and projects collaboratively.

Task Coordination:

The process of organizing and managing individual tasks within a project to ensure they align with the project's objectives and timelines.

Project Management:

The systematic planning, coordination, and execution of a project, involving tasks, resources, and timelines to achieve specific goals.

Collaboration Hub:

Monday.com's centralized platform that facilitates seamless collaboration among team members, allowing them to share files, communicate, and stay updated on project progress.

STD (Software Test Description) File:

A document outlining the details and specifications of the software testing procedures, including test scenarios, test cases, and expected outcomes for Monday.com development.

Test Scenario:

A high-level description of a specific functionality or feature to be tested within Monday.com, providing context for the test cases.

Test Case:

A detailed set of conditions, inputs, and expected outcomes for testing a specific aspect or functionality of Monday.com.

Test Suite:

A collection of related test cases organized together in the STD file, typically focused on a specific module or feature of the Monday.com application.

Defect:

An identified deviation or inconsistency in the behavior of Monday.com compared to its expected results, documented in the STD file for resolution.

Test Tree

1. Account Management

Functional Testing:

Register New Account:

- Successfully register with valid details (email, password).
- Attempt registration with each field invalid/missing: email, password.
- Validate email verification process.

Login:

- Successfully log in with correct credentials.
- Attempt login with incorrect password, email not registered, and fields left blank.
- Test "Remember Me" functionality for session persistence.

Password Recovery:

- Successfully initiate and complete password recovery via email.
- Attempt password recovery with an unregistered email address.

Non-Functional Testing:

Performance Testing:

- Assess login process time under normal and peak load.

Usability Testing:

- Evaluate clarity of error messages for failed registration/login attempts.

Security Testing:

- Test for SQL injection vulnerabilities in login and registration forms.

2. Project Creation and Management

Functional Testing:

- Create a New Project:
 - Successfully create a project with mandatory and optional fields.
 - Attempt to create a project with each mandatory field missing.
- Edit Project Details:
 - Edit all editable fields of a project and verify changes.
 - Test permission-based access for editing project details.
- Archive/Delete a Project:
 - Successfully archive and restore a project.
 - Successfully delete a project and validate its removal.

Non-Functional Testing:

- Scalability Testing:
 - Simulate project creation and editing by numerous concurrent users.
- Security Testing:
 - Verify that project data is correctly partitioned and inaccessible to unauthorized users.

3. Task Management

Functional Testing:

- Add New Task:
 - Add a task with all details specified, including due dates and assignees.
 - Attempt to add tasks with missing mandatory fields.
- Update Task Status:
 - Update task status (To Do, Doing, Done) and verify the change.
 - Verify notification system when tasks are updated.
- Assign and Reassign Tasks:
 - Assign a task to a team member and verify they are notified.
 - Reassign a task and check for update notifications.

Non-Functional Testing:

- Reliability Testing:
 - Verify task creation and updates are consistently reflected under various network conditions.

4. Collaboration and Communication

Functional Testing:

- Comment on Tasks and Projects:
 - Successfully post a comment on a task/project.
 - Edit and delete comments and verify changes.
- File Sharing:
 - Upload files to a task/project and verify successful upload.
 - Download uploaded files and check integrity.

Non-Functional Testing:

- Performance Testing:

- Measure time to upload and download files of various sizes.

5. Reporting and Analytics

Functional Testing:

- Generate Reports:
 - Successfully generate project and task reports with filters applied.
 - Export reports in different formats (e.g., PDF, Excel) and verify content.
- Dashboard Functionality:
 - Customize a dashboard with various widgets and save configuration.
 - Share dashboard with team members and verify access control.

Non-Functional Testing:

- Performance Testing:
 - Evaluate responsiveness of the reporting and dashboard features under load.

6. Integration and API Testing

Functional Testing:

- Third-Party Integrations:
 - Configure integrations (e.g., Google Calendar, Slack) and verify data flow.
 - Test API endpoints for creating, updating, and deleting tasks/projects.

Non-Functional Testing:

- Reliability Testing:
 - Ensure third-party integrations and API calls consistently perform under different network conditions.

7. Accessibility and Globalization

Functional Testing:

- Accessibility Compliance:
 - Test for WCAG 2.1 compliance using automated tools and manual checks.
- Localization:
 - Verify the platform supports multiple languages and regional settings accurately.

Non-Functional Testing:

- Compatibility Testing:
 - Ensure the platform is fully functional and displays correctly across major browsers and devices.