Testing Manual

29.0.2024

Introduction

This manual outlines the comprehensive testing strategy and processes for the Smart Inventory Web App. It serves as a definitive guide for quality assurance team members, developers, and stakeholders involved in the testing process. Our goal is to ensure the highest quality, reliability, and user satisfaction for our application.

Test Strategy

Our test strategy adopts a shift-left approach, integrating testing throughout the development lifecycle. We focus on early defect detection, continuous integration, and maintaining a balance between automated and manual testing efforts.

Key Principles

- Early and continuous testing
- Automation-first approach
- Risk-based testing prioritization
- Continuous feedback and improvement

Types of Tests

- Unit Tests: Verify individual functions, methods, and small code units in isolation.
- Component Tests: Validate the functionality of isolated UI components.
- Integration Tests: Ensure different parts of the application work together correctly.
- End-to-End (E2E) Tests: Simulate real user scenarios, validating the entire application workflow.
- API Tests: Verify the functionality and reliability of AWS backend services.
- Performance Tests: Evaluate the system's responsiveness and stability under various load conditions.
- Security Tests: Identify vulnerabilities and ensure data protection.

Test Tools

Cypress

- Primary tool for unit, component and E2E testing
- Provides a browser-based testing environment
- Supports automated test script creation and execution

Postman

- Used for API testing
- Simulates HTTP requests to API endpoints
- Manages test environments and collections
- Supports automated API test scripts

AWS CloudWatch

- Monitors and logs AWS service performance
- Helps in identifying and debugging issues in AWS resources

Quality Assurance Requirements

- Reliability:
    - Ensured through comprehensive test coverage across all test types.
    - AWS load testing
- Usability:
    - Primarily addressed by E2E tests and usability testing sessions.
- Maintainability:
    - Enhanced by well-documented, modular test suites.
- Scalability:
    - Verified through load testing and AWS service configuration tests.
    - Also through auto scaling tests.
- Security:
    - Ensured through regular security audits and role based tests.

Testing Processes

Component Tests

- Use Cypress Component Testing
- Mount individual Angular components in isolation
- Interact with components using Cypress commands
- Assert expected behavior and DOM changes

Integration Tests

- Test interactions between different modules or services
- Use Cypress for frontend integration testing
- Implement API mocking when necessary

End-to-End (E2E) Tests

- Write tests simulating complete user journeys Use Cypress for E2E testing Interact with UI elements using
- Cypress commands Assert expected application state and API responses Run tests with npx cypress run

API Tests

Use Postman to test AWS API Gateway endpoints Create collections for each API resource Write tests for different HTTP methods (GET, POST, PUT, DELETE) Test both success and error scenarios. Use environment variables for different stages (dev, staging, prod) Run tests manually or integrate with CI/CD pipeline

Test Data Management

Maintain a separate test database with anonymized data Use data factories to generate test data programmatically Implement data cleanup procedures after test execution Version control test data alongside the codebase

Continuous Integration and Deployment

- Integrate all tests into the CI/CD pipeline run unit and integration tests before every merge to main or develop.
- Execute E2E tests before deployment to staging.
- Perform smoke tests after each production deployment.

Issue Tracking and Resolution

Use a dedicated issue tracking system (e.g., Jira, GitHub Issues) Categorize and prioritize issues based on severity and impact Implement a clear workflow for issue resolution and retesting Conduct regular triage meetings to review and update issue status

Test Review and Sign-off

Conduct peer reviews of test cases and scripts Hold regular test results review meetings with stakeholders. Establish clear criteria for test pass/fail and release readiness Obtain formal sign-off from relevant stakeholders before major releases.