Software Test Report

Clean City Waste Management System v1.0

Date of Report: November 18th , 2025

Prepared by: QA Team Jowii

Version: 1.0

Executive Summary

This report summarizes the testing activities performed on the CleanCity Waste Management System from November 4th to November 18th , 2025. The testing focused on validating core functionalities, security measures, usability, and performance across multiple platforms.

Key Findings:

· 120 test cases executed with 95.8% pass rate

· All critical and high-severity defects have been resolved

· Core modules (Authentication, Waste Management, Dashboard) are stable and functional

· Security vulnerabilities (XSS, SQL injection, authentication bypass) were identified and mitigated

· System meets accessibility and responsiveness standards across devices

· One medium-severity defect remains open (mobile navigation)

Recommendation:

The QA team Jowii recommends the release of Clean City v1.0 to production, with a scheduled hotfix for the mobile navigation issue within one week of release.

1. Test Objective

The primary objective of this testing cycle was to ensure the CleanCity Waste Management System meets functional, security, and usability standards before release. Specific goals included:

1. Validate user authentication and role-based access control workflows
2. Verify end-to-end waste pickup scheduling and management processes
3. Assess system security against common vulnerabilities (XSS, SQL injection, CSRF)
4. Ensure compatibility across major browsers and mobile devices
5. Evaluate system performance under typical user loads and data volumes
6. Validate accessibility compliance with WCAG 2.1 guidelines

Testing was conducted over a three-week period from November 4th -18th , 2025.

---

2. Areas Covered

2.1 Functional Testing

The following functional areas were thoroughly tested:

1. User Authentication & Authorization.

· User registration and account creation
· Login/logout functionality
· Password reset flow
· Role-based access control (Admin, Resident, Driver)
· Session management and persistence

2. Waste Management

· Pickup request submission and validation

· Scheduling with date validation

· Location-based filtering

· Request status tracking and updates

3. Dashboard & Reporting

· Real-time request tracking

· Filtering by status and location

· Statistical data display

· Data export functionality (CSV, PDF)

4. Admin Panel

· User management

· Request approval workflows

· System configuration

· Audit logging

5. Feedback & Awareness

· Feedback submission and processing

· Educational content delivery

· Blog management and publishing

2.2 <u>Non-Functional Testing</u>

1. Performance Testing

· Page load times under various network conditions

· Response time for critical user journeys

· Memory usage and optimization

· Load testing with concurrent users

2. Security Testing

· Input validation and sanitization

· XSS and SQL injection prevention

· Authentication and authorization mechanisms

· Session management and timeout

· Data encryption in transit and at rest

3. Compatibility Testing

· Browser testing (Chrome, Firefox, Safari, Edge)

· Mobile devices (iOS, Android)

· Responsive design across screen sizes

· Cross-platform functionality

4. Usability & Accessibility

· Navigation flow and user experience

· WCAG 2.1 compliance

· Keyboard navigation and screen reader support

· Error message clarity and guidance

3. Areas Not Covered

The following areas were excluded from this testing cycle:

· Integration with Municipal ERP Systems

· Reason: Third-party APIs still under development, scheduled for v1.1

· Real-Time GPS Tracking for Drivers

· Reason: Feature postponed to v1.2 release

· Multi-Language Support

· Reason: Localization framework planned for future release

· Advanced Analytics and Reporting

· Reason: Business intelligence module scheduled for separate deployment

· Mobile Native Applications

· Reason: Progressive Web App approach adopted for initial release

4. Testing Approach

4.1 Test Strategy

Our testing approach combined multiple methodologies to ensure comprehensive coverage:

1. Risk-Based Testing

· Identified high-risk areas through threat modeling and historical analysis

· Security features and data integrity received additional testing focus

2. Test Case Design

· Black-box and white-box testing techniques

· Boundary value analysis for input validation

· Decision tables for complex business rules

3. Automation & Manual Testing Balance

· Regression test suite automated (85 test cases)

· New features tested manually with automation scripts developed in parallel

· Exploratory testing sessions for usability and edge cases

4.2 Testing Process

The testing process followed these phases:

1. Test Planning (October 31st – 4th November, 2025)

· Test strategy development and resource allocation

· Test environment setup and data preparation

· Test case review and prioritization

2. Test Execution (November 5th -8th, 2025)

· Smoke testing on each new build

· Full regression testing on stable builds

· Feature-specific testing for new functionality

· Security and performance testing

3. Defect Management (November 8th, 2025)

· Defects logged in JIRA and Github with severity and priority assignments

· Daily defect triage meetings with development team

4. Reporting & Analysis (November 8th -18th , 2025)

· Test results compilation and metrics analysis

· Final assessment and recommendations

· Report preparation and stakeholder review

4.3 Testing Tools

The following tools were utilized during testing:

· Test Management: TestRail

· Defect Tracking: JIRA, Github

· Automation Framework: Selenium WebDriver, Katalon Studio, Cross - browser

· Performance Testing: Jmeter, Lighthouse

· Security Testing: OWASP ZAP, Burp Suite

· Accessibility Testing: axe-core, WAVE

· API Testing: Postman, REST Assured

4.4 Sample Key Test Cases

Test Case ID: AUTH-001;

· Title: Valid User Registration

· Preconditions: No user logged in

· Steps:

1. Navigate to Registration page
2. Fill valid user details (name, email, password)
3. Submit form

· Expected Results: Account created, success message shown

· Actual Results: As expected

· Status: PASS

Test Case ID: SEC-001

· Title: XSS Prevention in Form Inputs

· Preconditions: Access to any form with text input

· Steps:

1. Enter <script>alert('XSS')</script> in text field
2. Submit form

· Expected Results: Script treated as text, not executed

· Actual Results: As expected

· Status: PASS

5. **DEFECT REPORT**

5.1 Defect Summary;

Severity	Count	Closed	Open
Critical	8	8	0
High	15	15	0
Medium	12	11	1
Low	10	10	0
Total	45	44	1

5.2 Critical Defects (All Resolved);

1. XSS Security Vulnerability (DEF-001)

· Description: Script tags executed when entered in form inputs

· Root Cause: Missing input sanitization on client and server side

· Resolution: Implemented DOM Purify for client-side and validator library for server-side

2. Admin Panel Direct URL Access (DEF-006)

· Description: Admin panel accessible without authentication via direct URL

· Root Cause: Missing authentication middleware on admin routes

· Resolution: Added route guards and session validation

3. Self-Approval Bypass (DEF-007)

· Description: Residents could approve their own pickup requests

· Root Cause: Client-side role validation without server-side enforcement

· Resolution: Implemented server-side role-based access control

4. Cross-User Request Deletion (DEF-008)

· Description: Users could delete pickup requests of other users

· Root Cause: Missing ownership validation in delete API

· Resolution: Added user ownership checks for all data mutations

5. LocalStorage Role Tampering (DEF-009)

· Description: Users could become admin by editing localStorage

· Root Cause: Client-side role storage without server verification

· Resolution: Implemented JWT tokens with server-side role validation

5.3 Open Medium-Severity Defect

Mobile Menu Not Functional (DEF-004);

· Description: Hamburger menu doesn’t respond on mobile viewport

· Current Status: Development team has identified the root cause as CSS media query conflict

· Mitigation Plan: Temporary workaround using responsive design patterns. Fix scheduled for deployment in hotfix release on November 2, 2025.

5.4 Defect Trend Analysis

The defect discovery rate showed a healthy decline throughout the testing cycle:

· Week 1: 28 defects discovered (62%)

· Week 2: 12 defects discovered (27%)

· Week 3: 5 defects discovered (11%)

The declining trend, particularly for critical and high-severity issues, indicates the application has reached a stable state suitable for release.

---

6. Platform Details

6.1 Test Environment

Server Environment:

· Backend API: Node.js 18.17.0 with Express.js

· Database: PostgreSQL 14.8

· Cache: Redis 7.0.12

· Authentication: JWT-based with bcrypt hashing

· API Version: v1.0.0

Client Environments ;

Desktop Browsers:

Browser Version OS

Chrome 118.0 Windows 11, macOS Ventura

Firefox 118.0 Windows 11, macOS Ventura

Safari 16.6 macOS Ventura

Edge 118.0 Windows 11

Mobile Devices ;

Device OS Version Browser

iPhone 14 Pro iOS 16.6 Safari

Samsung Galaxy S22+ Android 13 Chrome

Google Pixel 7 Android 13 Chrome

iPad Air iPadOS 16.6 Safari

6.2 Network Conditions Tested;

· High-Speed: Wi-Fi (100+ Mbps)

· Average Mobile: 4G/LTE (10-20 Mbps)

· Poor Connection: 3G (1-2 Mbps)

· Intermittent: Simulated connection drops

· Offline: Service worker functionality

7. Overall Status

7.1 Testing Summary

· Total Test Cases: 120

· Test Cases Executed: 120 (100%)

· Test Cases Passed: 115 (95.8%)

· Test Cases Failed: 5 (4.2%)

· Automation Coverage: 70%

· Critical User Journeys: 100% passing

7.2 Quality Assessment

Based on comprehensive testing results, the Clean City v1.0 application has achieved a production-ready quality level with the following observations:

Strengths:

· Robust authentication and authorization system

· Intuitive user interface with responsive design

· Comprehensive security measures implemented

· Reliable data persistence and management

· Good accessibility compliance

Areas of Concern:

· Mobile navigation requires minor refinement

· Performance optimization needed for large datasets

· Additional error handling recommended for edge cases

7.3 Risk Assessment

The remaining risks associated with releasing the application are:

1. Mobile Navigation Issue: LOW RISK

· Impact: Minor usability issue affecting mobile users

· Probability: Low (affects specific interaction patterns)

· Mitigation: Hotfix scheduled within one week, workaround available

2. Performance with Large Datasets: LOW RISK

· Impact: Slower response times for administrators with large request volumes

· Probability: Medium (affects ~5% of users)

· Mitigation: Pagination and virtualization planned for v1.1

7.4 Release Recommendation

Based on our comprehensive testing and the current status of the application, the QA team

RECOMMENDS PROCEEDING WITH THE RELEASE of Clean City v1.0 to production, with the following conditions:

1. Deploy the mobile navigation hotfix by November 2, 2025
2. Monitor system performance metrics for the first 72 hours post-release
3. Implement phased rollout starting with 20% of users
4. Establish immediate communication channel for user feedback

7.5 Post-Release Activities

The following activities are recommended after release:

1. Close monitoring of application performance and error rates
2. User feedback collection through in-app surveys
3. Analysis of support tickets for emerging patterns
4. Performance benchmarking against pre-release baselines
5. Security monitoring and log analysis

8. Requirements Traceability

The following table shows how key requirements were validated through testing:

Requirement ID Requirement Description Test Case Ids Status

AUTH-001 System shall allow user registration with valid credentials AUTH-001 to AUTH-005 PASSED

AUTH-002 System shall enforce role-based access control AUTH-006, AUTH-009, ADMIN-001 PASSED

WM-001 Users shall be able to schedule waste pickup requests WM-001 to WM-006 PASSED

WM-002 System shall validate pickup request data WM-002 to WM-004 PASSED

SEC-001 System shall prevent XSS attacks SEC-001, SEC-002 PASSED

SEC-002 System shall prevent SQL injection DEF-022, DEF-046 PASSED

DASH-001 Dashboard shall display real-time request data WM-007 to WM-011 PASSED

PERF-001 Pages shall load within 3 seconds on 4G PERF-001, PERF-002 PASSED

---

9. Testing Challenges & Lessons Learned

9.1 Challenges Encountered

1. Test Data Generation

· Challenge: Creating realistic municipal waste management scenarios

· Solution: Collaborated with domain experts to develop representative test datasets

2. Security Testing Complexity

· Challenge: Comprehensive security testing required specialized expertise

· Solution: Engaged security specialists and implemented automated security scanning

3. Mobile Device Fragmentation

· Challenge: Limited access to diverse mobile device configurations

· Solution: Leveraged cloud testing platforms and device emulation

9.2 Lessons Learned

1. Early Security Involvement

· Involving security experts during design phase prevents major rework

2. Automation Strategy

· Prioritizing critical path automation improves regression testing efficiency

3. User-Centric Testing

· Involving real users in usability testing provides valuable insights

4. Continuous Integration

· Implementing CI/CD pipelines accelerates feedback loops

10. Appendices

10.1 Test Case Execution Details

Detailed test case execution results are available Traceability Matrix Report.

10.2 Performance Test Results

Detailed performance test results are available in the Traceability Matrix Report

10.3 Traceability Matrix

The full Requirements Traceability Matrix linking requirements to test cases and their results is available in document Traceability Matrix Report.

10.4 Defect Details

Complete details of all defects, including screenshots and reproduction steps, are available in JIRA project CLEANCITY.

10.5 Test Data Inventory

Description of test datasets used during testing is available in the Test Data Inventory document under Traceability Matrix Report.