# Prizewell.org Testing Plan

This document outlines the testing strategy and quality assurance plan for Prizewell.org, a Nigerian e-commerce price comparison and analytics platform. The goal is to ensure stability, performance, and reliability across backend, frontend, and integration layers.

## 1. Backend (Django + Celery + Channels) Tests

These tests confirm the accuracy of scraping logic, task scheduling, and WebSocket message delivery.

• Scraper Logic Tests: Verify scrapers (Jumia, Konga, Slot.ng) return correct fields such as name, price, image, and URL.

• Utility Functions: Test safe\_group\_name sanitization and batch progress callback behavior.

• Model Tests: Check Product and ActiveScraper model CRUD operations and data consistency.

• Celery Integration: Run Celery scraping task and ensure messages are broadcast via Redis to WebSocket group.

• WebSocket Communication: Validate that connected clients receive progress and completion updates in real time.

## 2. Frontend (React/React Native) Tests

• Functional Tests: Ensure WebSocket connections open successfully and display live progress updates.

• UI Behavior: Verify product list and progress indicators update dynamically.

• Error Handling: Check graceful error messages on connection failure or scraper timeout.

• Component Tests: Run Jest/React Testing Library tests on key components (ProductCard, ProgressBar).

## 3. End-to-End (E2E) Tests

Full workflow simulations to verify system behavior from query initiation to result display.

• User searches for a product and receives live updates through WebSocket until completion.

• Multiple users join the same query group and see synchronized progress.

• System handles scraper error gracefully and notifies users.

• Price alert notifications trigger correctly on price drops.

## 4. Performance, Security, and Reliability Tests

• Load Testing: Simulate high concurrent WebSocket connections using Locust or k6.

• Stress Testing: Run multiple Celery tasks concurrently to test queue resilience.

• Security: Test URL sanitization, WebSocket origin checks, and credential protection.

• Recovery: Confirm that services recover correctly after Redis or Celery restarts.

## 5. Summary Table

|  |  |  |
| --- | --- | --- |
| Test Category | Example Tools | Goal |
| Unit Tests | pytest, pytest-django | Validate backend logic and models |
| Integration Tests | Django Channels Test Client | Verify async message flow |
| Frontend Tests | Jest, React Testing Library | Ensure UI updates correctly |
| E2E Tests | Cypress, Playwright | Validate real user flows |
| Load Tests | Locust, k6 | Ensure scalability |
| Security Tests | OWASP ZAP, Django Debug Off | Prevent vulnerabilities |

Prepared by: Product & QA Team, Prizewell.org

Date: October 2025