**Test plan for**

**E-Commerce Web Application**

*ChangeLog*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| 1.0 | 2025-04-7 | Kevin Lopez | Initial draft of the test plan |
| 2.0 | 2025-05-4 | Kevin Lopez | Final test plan |
|  |  |  |  |

1 Introduction 2

1.1 Scope 2

1.1.1 In-Scope 2

1.1.2 Out-of-Scope 2

1.2 Quality Objective 2

1.3 Roles and Responsibilities 3

2 Test Methodology 3

2.1 Overview 3

2.2 Test Levels 3

2.3 Bug Triage 3

2.4 Suspension Criteria and Resumption Requirements 4

2.5 Test Completeness 4

3 Test Deliverables 4

4 Resource & Environment Needs 4

4.1 Testing Tools 4

4.2 Test Environment 4

5 Terms/Acronyms 5

# Introduction

This document outlines the testing strategies, processes, and methodologies used for the Django-based e-commerce site project. The document now includes testing for the new features implemented during Sprint 4, such as Cart functionality, Admin Dashboard, Admin Functionality, and improvements to the Shopping page, Item page, and related functionalities.

## Scope

### In-Scope

 User authentication (login, account creation)

 Seller portal functionality

 Display of HTML pages from Django backend

 Basic CRUD for Users, Items, Cart, and Orders

 Cart Functionality: Ensure that users can add, update, and delete items in the cart.

* Verify that admins can monitor users, view and manage products, and access product details.

• Admins should be able to perform CRUD operations on users and products, delete accounts, and manage roles.

• The shopping page displays products and supports item selection and addition to the cart.

• Verify that each product has its own detail page showing the correct product information.

• Seller Add Item Input Validation: Ensure validation for adding items to the store.

• Implement the Terms and Services page with proper styling.

• Apply different role colors for Buyer, Seller, and Admin.

• Add back buttons to Admin pages and the Profile Update page.

• Cart Footer: Ensure the cart footer is properly fixed across pages.

• Display a dialogue when the cart is empty.

• Ensure appropriate error messages are shown for invalid actions on the Seller page.

• Update the navbar look on the homepage.

• Display the user's account balance on the store page

### Out-of-Scope

 Payment gateway integration

 Advanced admin analytics

## Quality Objective

 Ensure core user flows (login, item display, add to cart, etc.) function correctly for the demo

 Identify and fix critical issues that would block 50% feature completion

 Clean up UI formatting, ensure consistency, and address all known issues with layout.

* Ensure the backend functionalities like cart operations and admin controls work smoothly with the frontend.

## Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Net ID** | **GitHub username** | **Role** |
| Everette Berkyaw | Eb1663 | Eb1663 | Back-end developer |
| Granville Alpha | Gba35 | Gba35 | Front-end developer |
| Kevin Lopez | Kql13 | Kev1225 | Back-end developer |
| Gavin Seiler | Gjs153 | Gjs153 | Front-end developer |

# Test Methodology

## Overview

The project follows an Agile/Iterative methodology. Testing will occur as features are implemented, focusing on the integration between the frontend and backend. Sprint 4 features such as Cart functionality, Admin Dashboard, and the Shopping Page will be tested.

## Test Levels

 **Unit Testing**: Django model and view logic (Everette’s area)

 **Integration Testing**: Linking backend to frontend pages

 **Manual Testing**: Core user actions like login, navigation, cart flow

## Bug Triage

 Bugs are reported informally via team messages or GitHub issues.

* The team will prioritize bugs that impact core user functionalities for the demo

## Suspension Criteria and Resumption Requirements

 Suspend testing if critical components are non-functional

 Resume once blocking issue is fixed or workaround is implemented

## Test Completeness

 Login, seller portal, and item display function correctly

 Cart and order DBs accept and reflect test data

 All backend routes render the appropriate HTML pages

 Cart, Admin Dashboard, Item pages, and the updated homepage must all function correctly.

 Ensure UI inconsistencies are resolved.

 All actions on the Admin page should trigger appropriate responses.

# Test Deliverables

 Test Plan

 New or current SRS document

 Video recording for demo validation

# Resource & Environment Needs

## Testing Tools

 GitHub (for version control & issue tracking)

 Django test framework (if applicable)

 Browser Developer Tools (manual testing)

## Test Environment

 Development on local machines

 Browser support: Chrome, Firefox

# Terms/Acronyms

Make a mention of any terms or acronyms used in the project

| TERM/ACRONYM | DEFINITION |
| --- | --- |
| API | Application Program Interface |
| AUT | Application Under Test |
| DB | Databasae |
| HTML | HyperText Markup Language |
| CRUD | Create, Read, Update, Delete |