**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 |
|  |  |  |  |
|  |  |  |  |

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 project utilizes Django for backend development, with HTML templates serving as the frontend interface. Testing will focus on verifying core functionality and backend integration through manual and unit testing methods.

## 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

### Out-of-Scope

 Payment gateway integration

 Email notifications

 Mobile responsiveness

 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

 Verify backend and frontend integration using dummy HTML and live data

## 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 occurs alongside development, with features being tested as they are implemented. Dummy HTML templates are currently being used to validate backend logic.

## 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.

 Everette and you (as backend support) prioritize and fix based on importance for demo.

## Suspension Criteria and Resumption Requirements

 Suspend testing if critical components (e.g., user login) 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

# 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

 OS: Windows 10+ or macOS (confirm this if needed)

# 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 |