TEST PLAN

**9/5/2025**

**Version History:**

In Khaled Hegazya Initiate Test Plan document 9/05/2025

Khaled Hegazya Finalize first version of test plan document 11/05/2025

V 1.0

V1.1

**Date**

**Approval**

**Summary**

**Revised By**

**Version**

# Introduction

# This document objective is defining the testing strategy, scope, and plan for the development of the "Full checkout feature". The document aims to ensure a clear and structured approach to testing, and deliver a high-quality product.

# Test Objectives

 Validate that the application complies with both functional and non-functional specifications as defined in the requirements.

 Achieve full traceability by ensuring that each test case aligns with its corresponding requirement and that any detected defects can be traced back to their source.

 Detect defects and inconsistencies as early as possible in the development cycle to reduce the risk of issues in the final release.

 Confirm that all necessary integrations with external tools or systems function correctly and reliably.

 Collaborate closely with cross-functional team members to ensure clear understanding of business requirements and resolve ambiguities that may impact quality or delivery timelines.

# Test scope

* 1. In scope
* Adding items to cart
* User authentication
* Applying discount
* Entering shipping details
* Payment (via 3rd party) - functional
* Order confirmation
  1. Out of scope
* User authorization.
* Other modules like Home page , contact us , exe
* Performance testing using tools like jmeter
* Security testing “payment data security should be handeled with the 3rd party”

# Test strategy

# ****Type of Testing:****

* + Functional testing: To validate feature functionality.
  + Regression testing : Performed for each release cycle ensures no impact on existing
  + Smoke Testing: performed after every deployment to ensure the critical functionalities of the application are stable and ready for further testing.
  + Sanity Testing: performed on builds after minor fixes or changes to verify that the specific bug fix or enhancement works as expected without introducing new defects.

# ****Testing levels:****

* + **Unit Testing:** Conducted by developers during development
  + **API Testing:** Using POSTMAN for functional , security , and performance checks.
  + **Integration testing:** Ensure web , API and Database interact seamlessly.
  + **End to End testing:** Verifying complete testing on the user workflow from UI to API.
  + **Acceptance testing:** Ensure the system is ready for production use.

# ****Methodology:****

Manual testing (automated tests may be added later).

# ****Environment:****

* + **QA member should start testing the feature on test environment with deploying the feature branch**
  + Any issues should be reported, fixed, retested on that branch
  + After QA approve the feature and developers merge it to the master branch QA member should run regression test on the staging environment and ensure there is no issues before feature goes live
  + After release smoke test should be done on production for the released features + regression test on the other feature

# Backlog Features breakdown

## ****Epic 1: Shopping Cart Management****

### User Stories:

* **US1.1**: As a user, I want to add items to my shopping cart so that I can purchase them later.
* **US1.2**: As a user, I want to view all items in my cart so I can review before checking out.
* **US1.3**: As a user, I want to update item quantities or remove items from the cart so I can manage my order.

## ****Epic 2: User Authentication****

### User Stories:

* **US2.1**: As a new user, I want to register an account so I can place orders.
* **US2.2**: As a returning user, I want to log in securely so I can access my saved information.
* **US2.3**: As a user, I want to reset my password in case I forget it.

## ****Epic 3: Discounts and Promotions****

### User Stories:

* **US3.1**: As a user, I want to apply a discount code at checkout so I can reduce the total cost.
* **US3.2**: As a system, I want to validate discount codes so only eligible users can apply them.

## ****Epic 4: Shipping Information****

### User Stories:

* **US4.1**: As a user, I want to enter my shipping address so that my order can be delivered.
* **US4.2**: As a user, I want to choose my preferred shipping method so I can control delivery time and cost.

## ****Epic 5: Payment Processing****

### User Stories:

* **US5.1**: As a user, I want to securely enter my payment information via a third-party provider so I can complete my purchase.
* **US5.2**: As a system, I want to validate payment success or failure so I can proceed accordingly.

## ****Epic 6: Order Confirmation****

### User Stories:

* **US6.1**: As a user, I want to see a confirmation screen after payment so I know my order was placed successfully.
* **US6.2**: As a user, I want to receive an email with my order details so I can track my purchase.

# Test Schedule:

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Timeframe | User Stories Covered | Test Type |
| Phase 1: Cart & Auth | Week 1 | US1.1, US1.2, US1.3 (Cart)US2.1, US2.2, US2.3 (Authentication) | Functional, UI, |
| Phase 2: Discounts & Shipping | Week 2 | US3.1, US3.2 (Discounts)US4.1, US4.2 (Shipping) | Functional, Input validation |
| Phase 3: Payment Integration | Week 3 | US5.1, US5.2 (3rd-party payment gateway integration) | Integration, Security, Edge case |
| Phase 4: Order Confirmation | Week 4 | US6.1, US6.2 (Confirmation page + Email) | Functional, Email, Regression |
| Phase 5: Regression + UAT | Week 5 (Pre-release) | Full flow (US1.1 → US6.2) | Regression, Smoke, UAT |
| Phase 6: Post-release Check | Week 6 (Production) | Sanity check on major flows (e.g., add to cart, payment, order confirmation) | Sanity, Production Smoke |

# Test Case Priority Levels

#### ****P1 – Critical (Showstopper)****

* **Definition**: Tests core functionalities that must work for the system to function at all.
* **Impact**: Failure blocks further testing or delivery.
* **Execution**: Must be tested first in every cycle (Smoke/Sanity + Regression).

#### ****P2 – High Priority (Major)****

* **Definition**: Tests essential features that are not system-critical but have **high user/business impact**.
* **Impact**: Failure causes significant inconvenience or disables major functionality.
* **Execution**: After P1, tested during full regression and acceptance.

#### P3 – Medium Priority (Minor)

* **Definition**: Tests standard features that are **important but not urgent**.
* **Impact**: Failure may reduce usability but does not break the system flow.
* **Execution**: Tested when time permits, or in a full regression cycle.

#### P4 – Low Priority (Trivial/Nice-to-Have)

* **Definition**: Tests cosmetic or edge case scenarios with minimal impact.
* **Impact**: Failure doesn't impact user flow or core functionality.
* **Execution**: Tested if there's spare time or during UI review cycles.

# Bug Severity Classification:

**Severity 1** - Crash or High impact problems that often prevent a user/host from correctly completing an experience/booking.

**Severity 2** - Moderate to high frequency problems with the functionality/UI or UX impact

**Severity 3** - Either moderate problems with low frequency or low problems with moderate frequency; these are minor annoyance problems faced by a number of participants.

**Severity 4** - Low impact problems faced by few participants; there is low risk of not resolving these problems. Reward for resolution is typically exhibited in increased user satisfaction.

# Test Resources

* + Software Requirements Specification (**SRS**).
  + Backlog features and user stories acceptance criteria.
  + Design documents “Figma”.

# Test Deliverables

* + **Test Plan:** document that defines the testing scope, objectives, strategy. It outlines the testing approach for the entire testing lifecycle.
  + **Test Cases:** A set of test cases that cover all functional, integration, regression, and non-functional aspects of the system.
  + **Defect Reports:** Defects will be linked to its linked item on Jira.
  + **Test Execution Report.**
  + **Regression test Results.**
  + **Traceability Matrix.**

# Entry/Exit Criteria:

# Entry Criteria:

* + All necessary requirements / changes should be well defined and documented in the user story
  + Acceptance criteria for each user story is listed and understanding by QC member without ambiguities
  + The feature branch is ready with no merge , build issues .
  + All subtasks are done and the user story is ready for end to end tests

# Exit Criteria :

* + All test cases (Manual & Automated) have been executed with 95% test coverage met.
  + All critical and high priority defects are fixed , verified and closed.
  + Any open defects either low priority , don't impact critical functionality (Approved by PO).

# Tools and defect Tracking

Jira wil be used for defect reporting and issue bugs/defects management and traceability.

# Final Test Report

Test closure reports shall be generated for each testing phase as the testing phase gets comp