# Software Test Plan (STP)

**Project:** Mini Flipkart (E-commerce Platform)

**Version:** 2.0

**Authors:** Nahush P Shetty, Machiraju Karthikeya, Gururaja Rao M, Krishna Venkatesh

**Date:** 2025-11-23

**Status:** Draft

## Revision History

| **Version** | **Date** | **Author** | **Change Summary** |
| --- | --- | --- | --- |
| 1.0 | 2025-11-23 | Team | Initial test plan based on implemented system |

## Approvals

| **Role** | **Name** | **Signature/Date** |
| --- | --- | --- |
| QA Lead | Nahush P Shetty |  |
| Dev Lead | Krishna Venkatesh |  |
| Product Owner | Gururaja Rao M |  |
| Test Engineer | Machiraju Karthikeya |  |

# 1. Introduction

## 1.1 Purpose

This document defines the test plan for the Mini Flipkart E-commerce Platform v1.0. It outlines the objectives, scope, strategy, resources, schedule, and responsibilities for testing all functionalities implemented across Epic MF-1 (Core Customer Experience) and Epic MF-2 (Order Management & Security Foundations).

## 1.2 Scope

Testing encompasses all core functionalities of the Mini Flipkart system, including:

* **User Management** (MF-13 through MF-18)
* **Product Catalog** (MF-19 through MF-30)
* **Shopping Experience** (MF-31 through MF-36)
* **Checkout & Orders** (MF-37 through MF-48)
* **Inventory Management** (MF-49 through MF-54)
* **Security & Compliance** (MF-55 through MF-72)
* **Order Tracking** (MF-61 through MF-66)

This includes functional requirements as well as non-functional requirements such as **performance, security, usability, and compliance testing**.

## 1.3 References

* **SRS**: Software Requirements Specification for Mini Flipkart v1.0
* **SAD**: Software Architecture and Design Specification (docs/SAD.md)
* **Jira Backlog**: docs/jira-backlog.md
* **PCI-DSS v4.0**
* **WCAG 2.1 AA**

## 1.4 Definitions

* **STP**: Software Test Plan
* **UAT**: User Acceptance Testing
* **PII**: Personally Identifiable Information
* **MF-XX**: Jira task identifier (e.g., MF-13)

# 2. Test Items

## 2.1 Modules Under Test

1. **User Authentication and Registration Module** (MF-13 through MF-18)
2. **Product Catalog and Search Module** (MF-19 through MF-30)
3. **Shopping Cart and Wishlist Module** (MF-31 through MF-36, MF-40)
4. **Order Management and Payment Module** (MF-37 through MF-48, MF-55 through MF-60)
5. **Inventory Management Module** (MF-49 through MF-54)
6. **Shipment Tracking Module** (MF-61 through MF-66)
7. **Audit Logging and Compliance Module** (MF-67 through MF-72)
8. **Admin Dashboards** (MF-50, MF-68)

# 3. Features to be Tested

## 3.1 Functional Requirements (Mapped to Jira Tasks)

### Epic MF-1: Core Customer Experience

| **Requirement** | **Jira Tasks** | **Test Coverage** |
| --- | --- | --- |
| **User Registration** | MF-13 - MF-18 | Registration API, form validation, session creation |
| **Product Catalog** | MF-19 - MF-24 | Product listing, category filtering, UI rendering |
| **Advanced Filtering** | MF-19, MF-23 | Filter validation, price range, sort options |
| **Checkout Flow** | MF-37 - MF-42 | Multi-step checkout, address validation, payment processing |

### Epic MF-2: Order Management & Security Foundations

| **Requirement** | **Jira Tasks** | **Test Coverage** |
| --- | --- | --- |
| **Order Orchestration** | MF-43 - MF-48 | Order submission, idempotency, failure handling, compensation |
| **Inventory Sync** | MF-49 - MF-54 | Real-time sync, batch sync, low stock alerts, anomaly detection |
| **Audit Logging & Compliance** | MF-67 - MF-72 | Audit log creation, tamper detection, PII redaction, compliance reporting |

## 3.2 Non-Functional Requirements

| **Requirement** | **Target** | **Test Coverage** |
| --- | --- | --- |
| **Performance** | 95% of API requests respond within 3 seconds | Load testing, response time measurement |
| **Scalability** | Support 10,000 concurrent users | Load testing, stress testing |
| **Usability** | WCAG 2.1 AA compliance | Accessibility testing, mobile responsiveness |
| **Security** | PCI-DSS compliance, OWASP Top 10 mitigation | Security testing, vulnerability scanning |

# 4. Features Not to be Tested

The following items are explicitly **out of scope** for testing in v1.0:

1. **Third-Party Services**: Internal operations of payment gateway providers or external banking systems.
2. **Infrastructure**: Cloud provider infrastructure, network infrastructure, or DNS resolution.
3. **External Dependencies**: MongoDB database engine internals or npm package internals.
4. **Future Features**: Mobile native applications.

# 5. Test Approach / Strategy

## 5.1 Test Levels

| **Test Level** | **Scope** | **Tools** | **Jira Tasks Covered (Example)** |
| --- | --- | --- | --- |
| **Unit Testing** | Individual components, services, and middleware | Jest 29.7.0 | MF-18, MF-24, MF-30, MF-36, etc. |
| **Integration Testing** | API endpoints, database interactions, service integrations | Jest, Supertest 7.0.0 | MF-42 (Complete purchase flow), MF-48 |
| **System Testing** | End-to-end testing of complete user journeys | Manual testing, Automated scripts | Full User Registration $\rightarrow$ Checkout flow |
| **Acceptance Testing** | Validation against business requirements (UAT) | Manual testing, Product Owner participation | All user-facing features in Epic MF-1 & MF-2 |

## 5.2 Test Types

* **Functional Testing**: Verify all features work as specified (MF-13 through MF-72).
* **Regression Testing**: Automated test suite execution on each build to prevent breaks.
* **Performance Testing**: Load testing (JMeter/Locust planned) to verify targets of **< 3s response time** and **10,000 concurrent users**.
* **Security Testing**: Focus on **RBAC**, **JWT validation**, **Audit Log integrity** (MF-71), and **XSS/SQL injection prevention**.
* **Usability Testing**: Evaluate WCAG 2.1 AA compliance and mobile responsiveness.
* **Compatibility Testing**: Verify across latest versions of Chrome, Firefox, Safari, and Edge, and various screen sizes.

## 5.3 Entry Criteria

Testing can begin when:

1. **Development Build**: Stable build delivered to QA team.
2. **Feature Completion**: All high-priority features (Epic MF-1) are implemented.
3. **Test Environment**: Fully configured (MongoDB, Backend API, Frontend build).
4. **Test Data**: Comprehensive test data (users, products, orders) available.

## 5.4 Exit Criteria

Testing can conclude when:

1. **Test Execution**: All high-priority test cases executed (Unit/Integration: 100% pass rate; System: 95% pass rate).
2. **Defect Resolution**: All critical and major defects resolved and verified.
3. **Performance Benchmarks**: Met (95% of requests < 3s).
4. **Security**: No critical vulnerabilities found.
5. **UAT Sign-off**: Product Owner approval received.

# 6. Test Environment

## 6.1 Hardware and Software

| **Component** | **Key Specifications** |
| --- | --- |
| **Client Devices** | Desktop, Laptop, Tablet, Mobile (iOS/Android) |
| **Browsers** | Chrome, Firefox, Safari, Edge (Latest stable versions) |
| **Backend** | Node.js 20.18.0, Express.js 4.19.2 |
| **Database** | MongoDB (dedicated instance for integration, in-memory for unit testing) |

## 6.2 Test Tools

* **Unit/Integration**: **Jest 29.7.0**, **Supertest 7.0.0**
* **CI/CD**: Python 3.x with run.py script
* **Security (Planned)**: **OWASP ZAP**, npm audit
* **Performance (Planned)**: **JMeter**, **Locust**

# 7. Test Schedule

## 7.1 Milestones

| **Phase** | **Duration** | **Activities** |
| --- | --- | --- |
| **Planning and Test Design** | 3 weeks | Test plan creation, test case design |
| **Environment Setup** | 2 weeks | Configuration, test data preparation |
| **Unit & Integration Testing** | 2 weeks | Automated test execution, defect reporting |
| **System Testing** | 1.5 weeks | End-to-end testing, user journey validation |
| **Performance & Security Testing** | 1.5 weeks | Load testing, vulnerability assessment |
| **UAT** | 1.5 weeks | User acceptance testing, feedback collection |
| **Final Report & Sign-off** | 1 week | Test summary report, RTM finalization |

## 7.2 Test Execution Schedule by Epic

* **Epic MF-1 (Core Customer Experience)**: Weeks 1-2 (Oct 16-31)
* **Epic MF-2 (Order Management & Security)**: Weeks 3-4 (Nov 1-15)

# 8. Test Deliverables

1. **Test Plan** (This document)
2. **Test Cases**: Automated scripts (backend/tests/) and manual test cases.
3. **Test Execution Reports**: Daily status, Weekly summaries, and Final test summary report.
4. **Defect Reports**: Tracking in Jira.
5. **Requirements Traceability Matrix (RTM)**: Mapping requirements (MF-13 through MF-72) to test cases.

# 9. Roles and Responsibilities

| **Role** | **Name** | **Responsibility** |
| --- | --- | --- |
| **QA Lead** | Nahush P Shetty | Manages testing process, approves test strategy, signs off on reports. |
| **Test Engineer** | Machiraju Karthikeya | Designs and executes test cases, creates automated scripts, reports defects. |
| **Developer** | Krishna Venkatesh | Supports defect resolution, provides technical assistance and test data. |
| **Product Owner** | Gururaja Rao M | Reviews test results, signs off on UAT, prioritizes defect resolution. |

# 10. Risks and Mitigation

| **Risk** | **Impact** | **Mitigation** |
| --- | --- | --- |
| **Delay in stable build delivery** | High | Implement continuous integration pipeline (run.py), use Mock APIs for early integration. |
| **Critical defects found late** | High | Prioritize testing of high-risk modules, early security and performance testing. |
| **Integration issues with third-party APIs** | Medium | Use mock APIs/stubs (MF-48), implement robust error handling (MF-47), test with sandbox environments. |
| **Performance targets not met** | Medium | Early performance testing (MF-48), load testing, database query optimization. |

# 11. Suspension & Resumption Criteria

## 11.1 Suspend Testing If

* Test environment unavailable for more than 4 hours.
* New build causes **more than 30 existing test cases** to fail.
* Critical bug or security vulnerability is found that blocks major testing areas.

## 11.2 Resume Testing If

* Blocking issue is resolved and verified.
* New, stable build is provided, and smoke tests pass.
* QA Lead approves resumption.

# 12. Test Case Management & Traceability

## 12.1 Requirements Traceability Matrix (RTM)

The RTM ensures every requirement (MF-13 through MF-72) is covered by one or more test cases, ensuring **100% requirement coverage**.

## 12.2 Test Case Organization

* **Naming Convention**: TC-<MODULE>-<NUMBER> (e.g., TC-AUTH-001).
* **Organization**: Cases grouped by module (Authentication, Order Orchestration, Compliance) and by test level (Unit, Integration, System).

# 13. Test Automation Strategy

## 13.1 Automated Tests

* **Backend Automation**: Unit and Integration tests using **Jest** and **Supertest**.
* **Execution**: Automated test suite runs on every build via the local CI/CD pipeline (python run.py).

## 13.2 Manual Tests

* **Frontend Testing**: UI component, user journey, cross-browser, and mobile responsiveness testing.
* **Exploratory Testing**: Ad-hoc testing for usability and edge cases.

**Document End**