🧪 Test Plan Document – Swish Mobile App

Project Name	Swish
Created By	Vikash kumar Bharti
Created Date	10 - 04 - 2025
Device Name	Redmi Note 5 pro

📘 1. Introduction

This test plan defines the strategy, scope, and resources for testing Swish, a fast and efficient e-commerce mobile application. Swish offers features like OTP-based login, product search, live location-based ordering, real-time order tracking, and multiple payment options. The goal is to ensure a smooth, secure, and reliable shopping experience through comprehensive testing of all key functionalities.

🎯 2. Objectives

- Ensure all functional and non-functional requirements are met.
- Identify and fix bugs before app release.
- Validate app performance, usability, and compatibility.
- Confirm integration with payment gateways, messaging systems, and GPS services.

📦 3. Scope

✅ **In Scope:**

- Mobile number-based sign-up with OTP (SMS/WhatsApp)
- Product browsing and cart operations
- Add/manage delivery address
- Payment methods (Google Pay, Payphone, Cash on Delivery, Wallet)
- Offers, coupons, and rewards
- Real-time order tracking and delivery under 10 minutes
- User profile management and logout
- Help & Support access

❌ **Out of Scope:**

- Backend database testing
- Performance/load testing on server infrastructure

🔍 4. Testable Features

- User Authentication (Sign Up, OTP Verification)
- Product Search, Selection, and Cart Functionality
- Add Address with GPS and manual input
- Payment and Rewards System
- Live Order Tracking
- Order History and Repeat Order

- Gift a Swish (Referral Feature)
- Help/Support & Profile Management

🧪 5. Testing Approach

- **Manual Testing**: Functional, UI/UX, Integration, Usability
- **Black-box Testing** for most scenarios
- **Regression Testing** after each build
- **Smoke & Sanity Testing** for every release

👥 6. Roles and Responsibilities

Role	Responsibility
QA Lead	Create test plan, strategy, assign tasks
QA Engineers/Testers	Create and execute test cases, log and retest bugs
Developers	Fix reported bugs and perform unit testing
Product Manager	Review feature requirements and approve test coverage

📅 7. Test Schedule

Phase	Timeline
Test Planning	10-April-2025 to 11-April-2025
Test Case Creation	12-April-2025 to 13-April-2025
Test Execution	14-April-2025 to 18-April-2025
Bug Fix & Retesting	19-April-2025 to 20-April-2025

Final Reporting	21-April-2025

📄 8. Test Deliverables

- Test Plan Document
- Mind Map
- Test Scenarios

- Test Cases (Google Sheets)
- Test Data
- Bug/Defect Reports (in Jira or Bug Report Sheet)
- Test Summary Report

🚦 9. Entry & Exit Criteria

Entry Criteria:

- All features implemented and stable build ready
- Test environment set up
- Test cases reviewed and approved

Exit Criteria:

- All critical and major defects resolved
- 95%+ test cases executed with Pass status
- Sign-off from QA team

🛠️ 10. Tools

- **Test Case Management**: Google Sheets / Excel
- **Bug Tracking**: Jira or Google Sheets
- **Communication**: Slack / WhatsApp / Email
- **Documentation**: Google Docs
- **Testing Devices**: Android (manual)

⚠️ 11. Risks and Mitigation Plans

Risk	Mitigation
Delayed build from dev team	Buffer time added in schedule
OTP delays (WhatsApp/SMS)	Use test numbers or mocks where possible
Device compatibility issues	Test on multiple devices and screen sizes
Payment gateway failure	Use sandbox mode for test transactions
App crashes or blocking bugs	Immediate reporting and hotfix priority