Test Plan for Meesho Mobile Application

---

1. Objective:

To validate the functionality, usability, performance, security, and compatibility of the Meesho mobile application across various devices and operating systems. Ensure the app meets the business and user requirements with minimal defects before release.

---

2. Scope:

In-Scope:

- Testing core features like product browsing, search, cart, checkout, payments, and order tracking.
- Login/Signup via mobile number, OTP, Google, and Facebook.
- Push notifications and in-app promotions.
- Integration with third-party payment gateways (e.g., UPI, cards, wallets).
- Compatibility testing on Android and iOS platforms.

Out-of-Scope:

- Backend service testing.
- API load testing (will be handled by the backend team).

---

3. Test Methodology:

- Manual Testing for UI/UX, Exploratory, and Regression.
- Automation Testing for smoke, sanity, and regression using Appium and Java.
- Agile methodology will be followed for iterative test cycles with sprint-based test execution.

---

4. Test Approach:

- Identify test scenarios and prepare test cases based on user stories.
- Perform UI, Functional, and Integration Testing.
- Execute test cases in QA environment.
- Log and track defects in Jira.

- Regression suite execution before each release.
- Final round of testing in staging before production release.

---

5. Assumptions:

- Functional requirements will be provided and finalized before testing begins.
- Proper test environment and test data will be made available.
- APIs will be stable and integrated with the frontend.

---

6. Risk:

- Delay in requirement finalization.
- High frequency of UI changes during sprint.
- Unavailability of real devices for compatibility testing.

---

7. Mitigation Plan:

- Coordinate closely with BA and Dev teams for requirement clarity.
- Use device farms like BrowserStack for wider compatibility testing.
- Automate regression to save time on repetitive tasks.

---

8. Roles of Tester:

- Prepare and execute test cases.
- Report bugs with proper steps and screenshots.
- Communicate test results and risks.
- Participate in sprint ceremonies.
- Collaborate with developers for faster issue resolution.

---

9. Scheduling:

Phase	Duration
Test Planning	2 days
Test Case Design	4 days
Test Case Review	2 days
Test Execution (Sprint)	Ongoing (2-week sprint)
Regression Testing	2 days/sprint
UAT Support	As per schedule

---

10. Defect Tracking:

- Defects will be logged in **Jira**.
- Each defect will be assigned severity and priority.
- Retesting will be done post-fix confirmation.
- Daily defect triage meetings to prioritize and resolve bugs.

---

11. Entry and Exit Criteria:

Entry Criteria:

- All user stories are developed.
- Test environment is set up.
- Test data is ready.

Exit Criteria:

- All critical test cases executed.
- No critical or high severity defects remain open.
- Test summary report shared.

---

12. Test Automation:

- Automation will be implemented using **Appium with Java**.
- CI/CD integration using Jenkins for nightly builds.
- Focus on login, search, cart, and checkout workflows.

13. Deliverables:

- Test Plan
- Test Scenarios and Test Cases
- Test Summary Report
- Defect Report
- Automation Scripts
- Test Execution Report

14. Templates:

- Test Case Template (Excel/Zephyr)
- Bug Report Template (Jira)
- Daily Status Report Template
- Test Summary Report Template