

Test Plan Outline

A. TEST PLAN IDENTIFIER

- **Description:** Unique ID or version number of the test plan for tracking.

Example:

- *TP-ECOM-01 (Test Plan for E-commerce Application, version 1.0).*

B. INTRODUCTION

- **Description:** Provides an overview of the software to be tested, its objectives, and testing scope.

Example:

- *This test plan defines the strategy for testing the mobile e-commerce app to ensure reliability, performance, and security.*

C. TEST OBJECTIVES

- **Description:** Defines what the testing process aims to achieve.

Example:

- *Verify that all functional requirements are met.*
- *Ensure usability and performance under load.*
- *Validate data security and transaction integrity.*

D. SCOPE OF TESTING

- **In-Scope:** Features and functionalities to be tested.
- **Out-of-Scope:** Features or areas excluded due to constraints.

Example:

- **In-Scope:** *Login, product search, checkout, payment, order tracking.*
- **Out-of-Scope:** *Web version, third-party integration beyond payment gateway.*

E. TEST ITEMS

- **Description:** Identifies the components or modules to be tested.

Example:

- *User Authentication, Product Catalog, Cart & Checkout, Payment Gateway, Notifications.*

F. FEATURES TO BE TESTED

- **Description:** Lists detailed software features subject to testing.

Example:

- *User login/logout functionality.*
- *Search filters and sorting.*
- *Secure payment processing.*

G. FEATURES NOT TO BE TESTED

- **Description:** Lists features intentionally excluded from testing.

Example:

- *Admin panel analytics, third-party ad integration.*

H. TEST STRATEGY / APPROACH

- **Description:** Defines testing levels, methods, and techniques.

Example:

- **Levels:** *Unit Testing → Integration Testing → System Testing → Acceptance Testing.*
- **Techniques:** *Black-box testing, white-box testing, exploratory testing.*
- **Tools:** *Selenium, JUnit, Postman.*

I. TEST ENVIRONMENT

- **Description:** Hardware, software, network, and tools required for testing.

Example:

- **Hardware:** *Android and iOS devices.*
- **Software:** *MySQL database, Node.js backend, React Native app.*
- **Tools:** *Jenkins (CI/CD), Jira (bug tracking).*

J. TEST DELIVERABLES

- **Description:** Documents and artifacts produced during testing.

Example:

- *Test cases, test scripts, defect reports, and test summary reports.*

K. ENTRY AND EXIT CRITERIA

- **Entry Criteria:** Conditions that must be met before testing starts.
- **Exit Criteria:** Conditions for ending testing.

Example:

- **Entry:** *Requirements finalized, code freeze achieved.*
- **Exit:** *95% test cases passed, critical defects resolved.*

L. ROLES AND RESPONSIBILITIES

- **Description:** Defines the testing team and their responsibilities.

Example:

- **Test Manager** – *Oversees planning and execution.*
- **Test Engineers** – *Write and execute test cases.*
- **Developers** – *Fix defects.*

M. SCHEDULE / TIMELINE

- **Description:** Test phases with timelines.

Example:

- **Unit Testing:** *Week 1–2*
- **Integration Testing:** *Week 3–4*
- **System Testing:** *Week 5–6*
- **UAT (User Acceptance Testing):** *Week 7*

N. RISK AND CONTINGENCY

- **Description:** Identifies potential risks and backup plans.

Example:

- **Risk:** *Limited test devices → Mitigation: Use emulators.*
- **Risk:** *Delay in environment setup → Mitigation: Use cloud-based testing.*

O. DEFECT REPORTING AND TRACKING

- **Description:** Defines how defects will be logged, tracked, and resolved.

Example:

- *All bugs will be recorded in Jira with severity levels (Critical, Major, Minor).*

P. APPROVAL

- **Description:** Lists stakeholders who must sign off on the test plan.

Example:

- *Project Manager, QA Lead, Client Representative.*