Test Plan – Notes Application

Project Name: Notes Application

Version: 1.0

Prepared By: Anjila Tripathi

Role: Quality Assurance

Date: 9/7/2025

Contents

[1. Introduction 2](#_Toc208173944)

[Overview of the Test Plan 2](#_Toc208173945)

[Purpose of Testing 2](#_Toc208173946)

[Scope of Testing 2](#_Toc208173947)

[2. Test Objectives 2](#_Toc208173948)

[3. Test Scope and Approach 2](#_Toc208173949)

[Testing Types and Methodologies 3](#_Toc208173950)

[4. Test Levels 3](#_Toc208173951)

[5. Test Environment 3](#_Toc208173952)

[6. Test Deliverables 4](#_Toc208173953)

[Deliverable Matrix 4](#_Toc208173954)

[7. Test Schedules 4](#_Toc208173955)

[8. Test Entry and Exit Criteria 4](#_Toc208173956)

[9. Defect Management 5](#_Toc208173957)

[10. Risks and Mitigation Strategies 5](#_Toc208173958)

[11. Roles and Responsibilities 5](#_Toc208173959)

[12. Tools & Technologies 6](#_Toc208173960)

[13. Test Execution 6](#_Toc208173961)

[14. Approvals & Sign-off 6](#_Toc208173962)

# 1. Introduction

## Overview of the Test Plan

This test plan defines the strategy, objectives, scope, and approach for testing the Notes Application, a web-based tool that allows users to register, log in, create, edit, and delete notes.

## Purpose of Testing

The purpose of testing is to validate that the application functions as expected, meets business requirements, ensures data security, and provides a smooth user experience across devices and browsers.

## Scope of Testing

In-scope: Functional testing (registration, login, notes CRUD), non-functional testing (performance, compatibility), UI/UX validation, and security testing (basic checks like XSS, invalid inputs).

Out-of-scope: Backend database performance tuning, API load testing beyond CRUD operations.

# 2. Test Objectives

* Verify all functional modules (registration, login, note creation, update, delete).
* Ensure proper handling of positive and negative scenarios.
* Validate non-functional requirements (performance, compatibility, usability).
* Detect and report defects early in the SDLC.

# 3. Test Scope and Approach

## Testing Types and Methodologies

* Integration Testing: Validate smooth interaction between modules (login → create note → edit → delete).
* System Testing: Validate complete workflow of the application in an end-to-end manner.
* User Acceptance Testing (UAT): Ensure application meets user expectations.
* Regression Testing: Verify that fixes/updates do not break existing functionality.
* End-to-End Testing: Cover entire workflows (register → login → CRUD → logout).
* Sanity & Smoke Testing: Quick checks to confirm build stability.

## Non-Functional Testing Areas

* Performance Testing: Verify response times for CRUD operations remain optimal.
* Compatibility Testing: Validate app behavior across different screen sizes and browsers.

## Out-of-Scope Functionalities

* Database-level stress testing.
* Third-party integration testing (if any).

# 4. Test Levels

* Unit Testing: Performed by developers.
* Integration Testing: Conducted by QA.
* System Testing: Performed by QA.
* UAT: Conducted by end users/stakeholders.

# 5. Test Environment

## Environment Setup

* Browsers: Chrome, Firefox, Edge
* OS: Windows 10/11, Ubuntu
* Devices: Desktop, Mobile (responsive design check)
* Test Data: Valid and invalid credentials, various note content formats.

# 6. Test Deliverables

## Deliverable Matrix

* Test Plan (this document)
* Test Cases (Excel/CSV)
* Bug Reports (JIRA/Excel)
* Test Execution Report
* Automation Scripts (if applicable)

# 7. Test Schedules

* Test Design:
* Test Execution:
* Reporting:

# 8. Test Entry and Exit Criteria

## Entry Criteria

* Requirements finalized and approved.
* Test environment set up.
* Test data prepared.

## Exit Criteria

* All critical and high-severity test cases executed and passed.
* Major defects resolved or deferred with approval.
* Test summary report completed.

# 9. Defect Management

* Defects will be logged in \[tool name: e.g., JIRA, Excel].
* Each defect will have severity, priority, and status.
* Defect lifecycle will follow: \*\*New → Assigned → Fixed → Retest → Closed.

# 10. Risks and Mitigation Strategies

* Risk: Unstable test environment.
* Mitigation: Use backup environment.
* Risk: Delay in requirement clarity.
* Mitigation: Early communication with stakeholders.

# 11. Roles and Responsibilities

* QA Officer: Prepare test plan, test cases, execute tests, log defects, report.
* Developers: Fix bugs, support defect analysis.
* Project Manager: Approve plan, manage timelines.

# 12. Tools & Technologies

## Testing Tools

* Manual Testing: Excel, Postman (basic checks)
* Automation: Selenium / Cypress (if applied)

Hardware

Laptop/Desktop with minimum 8GB RAM

Software

Browsers: Chrome, Firefox, Edge

OS: Windows/Linux

# 13. Test Execution

* Execute test cases as per schedule.
* Report test execution status daily.
* Share test summary at the end of cycle.

# 14. Approvals & Sign-off

This test plan requires approval from the QA Lead and Project Manager before execution.