

Test Plan: ToDo Application

1. Introduction

This test plan outlines the testing strategy for the ToDo web application. The goal is to ensure the application meets quality standards, is free of critical bugs, and provides a smooth user experience across platforms.

2. Objectives

- Validate core functionalities: adding, deleting, marking tasks.
- Ensure responsive design across devices.
- Verify data isolation between users.
- Identify and report bugs or inconsistencies.

3. Scope

In Scope

- Functional testing of task management features.
- UI/UX validation (checklists, visual elements).
- Basic cross-browser testing (Chrome, Firefox).
- Mobile responsiveness testing.
- Data isolation between users.

Out of Scope

- Performance testing under heavy load.
- Security penetration testing.
- Full end-to-end testing with external services.

4. Test Strategy

- Manual testing using predefined test cases.
- Visual inspection through a checklist.

- Bug tracking and severity classification.
- Testing across:
 - Chrome (Desktop)
 - Firefox (Desktop)
 - Mobile Simulator (Responsive)

5. Test Environment

- Web Application deployed locally.
- Browsers: Chrome, Firefox (latest versions)
- Tools:
 - Excel for tracking test results.
 - Browser developer tools for inspection.
 - GitHub for code and documentation versioning.

6. Test Deliverables

- Test Cases Excel File
- Test Checklist
- Bug Report
- Final Test Summary

7. Resources & Roles

Role	Name	Responsibility
Tester	Wiem Tabbal	Executes test cases and reports bugs
Developer	Wiem Tabbal	Fixes issues

8. Schedule

Phase	Duration
Test Case Design	2 hours
Manual Testing	3 hours
Bug Reporting	Continuous during testing
Final Review	1 hour

9. Risk and Mitigation

Risk	Impact	Mitigation
Shared tasks across users	High	Add session or user ID-based data separation
Browser compatibility issues	Medium	Test on multiple browsers
Missed edge cases	Medium	Include boundary test cases

10. Approval

Name	Role
Wiem Tabbal	Tester