Test Plan Document

Project Name: AutomationFramework\_Gopika

Prepared By: Gopika Gopan

Date: July 24, 2025

Framework Type: UI Automation Framework (Selenium + Cucumber + Extent Reports)

Language: Java

Build Tool: Maven

Test Runner: JUnit

Application Under Test (AUT): React Frontend with Express Backend (ToDoApp)

# 1. Objective

The purpose of this test plan is to validate the core functionality of the ToDo Web Application by automating key UI flows using Selenium WebDriver integrated with Cucumber BDD and ExtentReports. This ensures regression coverage and reduces manual testing effort.

# 2. Scope

In-Scope:

- Login functionality validation (valid and invalid cases)

- ToDo Dashboard functional testing: Add, Edit, and Delete tasks

- Screenshot capturing for all steps

- Extent HTML report generation with step-level logging

Out-of-Scope:

- API-level testing

- Performance, security, or accessibility testing

# 3. Test Strategy

UI Automation: Selenium WebDriver  
BDD: Cucumber  
Test Execution: JUnit  
Reporting: ExtentReports  
Browser Management: WebDriverManager  
Screenshot Handling: AWT Robot + WebDriver

# 4. Test Environment

Frontend: React (ToDoApp/client/)

Backend: Express (ToDoApp/server/server.js)

Browser: Google Chrome (latest)

Java Version: Java SE 17

Maven Version: 3.6+

OS: Windows/macOS/Linux

# 5. Folder Structure

AutomationFramework\_Gopika/  
├── src/main/java/utilities/  
├── src/test/java/stepDefinitions/  
├── src/test/resources/features/  
├── Automation Reports/  
├── ToDoApp/client/client  
└── ToDoApp/server/

# 6. Test Scenarios

Login Feature (login.feature):  
- Valid Login  
- Invalid Login

ToDo Feature (todo.feature):  
- Add a Task  
- Edit a Task  
- Delete a Task

# 7. Execution Steps

1. Start backend: cd ToDoApp/server && node server.js

2. Start frontend: cd ToDoApp/client/client && npm start

3. Run tests via Maven: mvn clean test

4. View Extent report: Automation Reports/Automation-Report-<timestamp>.html

# 8. Reporting & Artifacts

- HTML Extent Report: Includes each step with status

- Screenshots: Captured on success and failure

- Located in: screenshots/

# 9. Risks & Mitigation

Risk: Unexpected alerts on UI  
Mitigation: Handled via TestHooks.java

Risk: CDP version mismatch warnings  
Mitigation: Not impacting test execution

# 10. Maintenance Notes

Scripts are modularized by feature.  
Utilities are reusable and centralized.  
Easy to extend by adding new feature files and step definitions.