Test Plan: TodoApp (React + Node.js)

1. Document Details

• Project: TodoApp

• Version: 1.0

• Author: [Your Name]

• Date: [Date]

2. Objectives

Validate frontend login and CRUD workflows; ensure backend authentication and item API endpoints work correctly.

3. Scope

In‑scope:

– React UI: login valid/invalid, create/edit/delete todos

– Node.js API: /login, /items endpoints

Out‑of-scope:

– Security, performance, concurrency

4. Test Strategy

– UI automation via Selenium Java + JUnit

– API testing via REST‑Assured Java + JUnit

5. Environment & Tools

– React app: http://localhost:3000

– Backend API: http://localhost:5000

– Tools: Java 17, Maven, ChromeDriver, REST‑Assured

6. Test Schedule

– Preparation: 1 day

– Implementation: 2 days

– Execution & Reporting: 1 day

7. Entry & Exit Criteria

Entry: both servers running successfully

Exit: all test cases passed or failures documented

8. Risks & Mitigations

– UI flakiness (use explicit waits)

– In-memory data persistence (reset items before tests)

9. Deliverables

– Test plan document, UI & API tests, execution report

| **TC ID** | **Test Scenario** | **Steps** | **Expected Result** |
| --- | --- | --- | --- |
| UI\_TC\_001 | Login invalid | Enter wrong credentials, click Login | Alert “Invalid” appears |
| UI\_TC\_002 | Login valid | Enter user/pass, click Login | Todo UI loads |
| UI\_TC\_003 | Create todo | Login → Enter “Task 1” → Click Add | Todo list shows task |
| UI\_TC\_004 | Edit todo | Click Edit → enter "Edited" in prompt → accept | Task text updates |
| UI\_TC\_005 | Delete todo | Click Delete for item | Todo list is empty |

| **TC ID** | **Description** | **Method** | **Endpoint** | **Payload** | **Expected Result** |
| --- | --- | --- | --- | --- | --- |
| API\_TC\_001 | Login valid credentials | POST | /login | {username, password} | 200 OK and token returned |
| API\_TC\_002 | Login invalid credentials | POST | /login | wrong credentials | 401 Unauthorized |
| API\_TC\_003 | GET items when none exist | GET | /items | – | 200 OK and empty array |
| API\_TC\_004 | Create a new item | POST | /items | {text:"foo"} | 201 Created, body.text="foo" |
| API\_TC\_005 | Update existing item | PUT | /items/{id} | {text:"bar"} | 200 OK, body.text="bar" |
| API\_TC\_006 | Delete an item | DELETE | /items/{id} | – | 204 No Content |
| API\_TC\_007 | PUT invalid id | PUT | /items/999 | – | 404 Not Found |
| API\_TC\_008 | DELETE invalid id | DELETE | /items/999 | – | 404 Not Found |