

Test Plan for Todo Web Application (React + Node.js)

1. Objective

This test plan outlines the approach for testing a simple Todo web application with a React frontend and Node.js backend. It covers functional UI testing using Selenium and API testing using RestAssured.

2. What is Being Tested

- Login Functionality
 - Valid and invalid credentials
- Todo Item Management
 - Creating a new todo
 - Editing an existing todo
 - Deleting a todo
 - Fetching all todos
- Data Consistency
 - Ensuring backend and UI reflect consistent states after actions

3. Test Coverage Areas

- API:
 - POST /login
 - GET /items
 - POST /items
 - PUT /items/:id
 - DELETE /items/:id
- Frontend (UI):
 - Login, create, edit, delete, view todos
- Negative Tests:
 - Invalid login, invalid input, operations on non-existent items
- Validation:
 - UI shows correct state after operations

4. Tools Used & Why

Tool	Purpose	Reason
----- ----- -----		
Selenium	UI Automation (Java)	Widely supported
RestAssured	API Automation (Java)	Easy Java integration
Maven	Build Tool	Standard dependency manager
JUnit	Test Runner	Lightweight and easy-to-use

5. How to Run the Tests

API Tests:

1. Start the backend server (<http://localhost:5000>)
2. Navigate to project folder:

```
cd restassured-api-tests
```
3. Run tests:

```
mvn test
```

UI Tests:

1. Start frontend (<http://localhost:3000>)
2. Start backend (<http://localhost:5000>)
3. Run Selenium tests:

```
mvn test
```

Ensure ChromeDriver is installed and path is set.

6. Assumptions & Limitations

- Localhost setup assumed.
- No token-based security.
- Selenium uses local Chrome.
- No database seeding or reset scripts.