**Test Plan / Test Strategy Document**

**Project:** Automated Testing of React Frontend & Node.js Backend API  
**Prepared by:** Supreethi Manjunatha  
**Date:** 29/07/2025

**1. What is Being Tested**

* **React Frontend (UI)**:  
  Functional user interface components for basic CRUD operations and login flow.
* **Node.js Backend API**:  
  RESTful endpoints for user login and item management supporting CRUD operations.

**2. Test Coverage Areas**

**React Frontend UI Automation**

* **Login Page**:
  + Valid credentials login success
  + Invalid credentials login failure
* **Item Management (User Items)**:
  + Create new item
  + Edit existing item
  + Delete item

**Node.js Backend API Automation**

* **Authentication**:
  + POST /login (positive and negative cases)
* **Item Endpoints**:
  + GET /items: Retrieve list of items
  + POST /items: Create a new item
  + PUT /items/:id: Update an existing item
  + DELETE /items/:id: Remove an item
* Included validation for success responses and error handling (e.g., invalid data, unauthorized access).

**3. Tools Used and Why**

| **Tool** | **Purpose** | **Rationale** |
| --- | --- | --- |
| **Playwright** | UI automation testing framework | Modern, fast, and reliable for React apps. Supports headless mode and cross-browser testing. |
| Visual Studio code | IDE/Editor | Used for Playwright automation as a IDE |
|  |  |  |

**4. How to Run the Tests**

**Prerequisites**

* Node.js and npm installed
* Both Backend and frontend server should be running
* Project dependencies installed via npm install

**UI Tests (Playwright)**

* Run tests in headed mode:

npx playwright test --ui

------

**5. Assumptions and Limitations**

* The React and Node.js applications are simple demos primarily for testing purposes and may not cover full production-level scenarios.
* Authentication is assumed to be basic and uses mocked or test credentials.
* API tests assume the backend server is running and accessible on localhost or configured test environment.
* UI tests depend on stable element selectors/IDs, significant UI changes may require test updates.
* Tests focus on functional correctness, performance, security, and accessibility tests are out of scope for this scope.
* Negative test cases cover common failures but not exhaustive edge cases.