Test Plan	Date
QA Engineer Assessment - React/Node.js Automation Testing	July 29, 2025

Summary
This test plan outlines the automated testing approach for a QA Engineer assessment involving a React frontend and Node.js backend application. The assessment focuses on demonstrating proficiency in UI automation using Playwright, API testing using Postman, test documentation, and code structure. Testing will cover basic CRUD operations for items management with authentication, including both positive and negative test scenarios to validate functional requirements and error handling capabilities.

In Scope	
UI Automation	Login functionality, item CRUD operations (create, read, update, delete)
API Testing	Authentication endpoint, item management endpoints with positive/negative scenarios
Test Documentation	Test plan, setup instructions, and coverage documentation
Code Quality	Clear structure, readable scripts, proper naming conventions

Out of Scope	
Performance Testing	Load testing, stress testing, performance benchmarks
Security Testing	Penetration testing, vulnerability assessments
Cross-browser Testing	Multiple browser compatibility (focus on single browser)
Mobile Responsiveness	Mobile device testing and responsive design validation

Tools Used & Why	
UI Testing - Playwright	Modern, fast, reliable cross-browser support, excellent debugging
API Testing - Postman	JavaScript-native, integrates well with Node.js
Reporting - Built-in framework reports	Time-efficient, adequate for assessment

Approach
Single cycle of automated testing execution focusing on core functionality validation. UI automation will cover login workflows and item management operations using Playwright testing framework. API testing will validate all CRUD endpoints with comprehensive positive and negative test cases. All tests will include proper assertions and error handling validation. Documentation will provide clear setup instructions and test execution guidance within the 4-hour time constraint.

Entry Criteria

Entry 1	React application deployed and accessible
Entry 2	Node.js API endpoints available and functional
Entry 3	Test framework setup completed
Entry 4	Test data and test user credentials available

Exit Criteria	
Exit 1	All UI automation test cases executed successfully
Exit 2	All API test endpoints covered with positive/negative scenarios
Exit 3	Test documentation completed with setup instructions
Exit 4	Code structure meets readability and organization standards

Test Deliverables	
Document	**Location**
Test Plan	This document
UI Test Scripts	/tests/ directory
API Test Scripts	/tests/ directory
README Setup Guide	Root directory
Test Execution Report	Generated after test run

Test Coverage Areas

UI Automation

- **Login Functionality**: Valid credential scenario
- **Item Creation**: Add new item and verify changes
- **Item Editing**: Update existing items and verify changes
- **Item Deletion**: Remove item and confirm removal
- **Data Assertions**: Verify expected data presence after operations

API Testing

- **POST /login**: Authentication with valid/invalid credentials
- **GET /items**: Retrieve individual items with validation
- **POST /items**: Create new items with validation
- **PUT /items/:id**: Update existing items with validation
- **DELETE /items/:id**: Remove items and handle non-existent IDs with validation

Assumptions and Limitations

- Single user testing (no concurrent user scenarios)
- Basic authentication mechanism
- Limited cross-browser testing due to time constraints
- Focus on core functionality over edge cases