Test Plan: Full-Stack Asho To-Do Application

- **Version:** 1.0
- **Date:** July 27, 2025
- **Author:** Ahmed Ashour

---

1. Introduction & Scope

This document outlines the testing strategy for the Asho to-do list application. The application consists of a React frontend, a Node.js/Express backend API, and an automated CI pipeline. The goal of this plan is to ensure the application's core features are functional, reliable, and continuously verified.

What is Being Tested (In Scope)

- **API Functionality:** All backend endpoints are tested for correct responses, status codes, and error handling.
- **User Authentication:** The complete login flow for the hardcoded user.
- **Task Management (CRUD):** The full lifecycle of a task.
- **Integration:** The communication between the React frontend and the Node.js backend.
- **Automation:** The CI pipeline's ability to automatically build the application and execute all tests.

What is Not Being Tested (Out of Scope)

- User registration or password management features.
- Performance, stress, or load testing.
- Formal cross-browser compatibility testing.
- In-depth security vulnerability scanning.
- Database persistence and migration testing.

---

2. Test Coverage & Strategy

The testing strategy employs a multi-layered approach to ensure quality from the API to the end-user experience.

- **API Testing (Backend):** Each API endpoint is tested directly using Postman.
- **End-to-End (E2E) Testing (UI):** The entire user flow is tested from a user's perspective using Cypress.
- **Code Coverage Analysis:** We measure the percentage of the application's source code that is executed by our automated tests to identify untested parts.

---

3. Tooling

Tool	Purpose	Why it was Chosen
Postman	API Testing	Allows for direct and isolated testing of API endpoints.
Cypress	End-to-End (E2E) Testing	An all-in-one framework that provides reliable and fast tests.
NYC / Istanbul	Backend Code Coverage	The standard for generating code coverage reports for Node.js.
GitHub Actions	Continuous Integration (CI)	Automates the entire build and test process within GitHub.
Codecov	Coverage Reporting	Provides clear, visual coverage reports.

---

4. How to Run the Tests

Local Execution

1. **Install dependencies** from all relevant directories (`npm install`).
2. **Start the application** in coverage-instrumented mode: `npm run start:coverage`.
3. **Run Cypress tests**: `npx cypress open` or `npx cypress run`.

Automated Execution

All tests are automatically executed by the GitHub Actions workflow on every push or pull request to the `main` branch.

---

5. Assumptions & Limitations

- **Assumption:** The testing environment is stable with Node.js installed.
- **Limitation:** The in-memory database is reset on every server restart.
- **Limitation:** E2E tests are primarily validated on a Chromium-based browser.
- **Limitation:** Non-functional requirements like performance and security are not covered.