# Test Plan / Test Strategy

**This document outlines the test strategy and approach used to validate both the frontend and backend of the Task Item Application.**

## 1. What is Being Tested

- Functional UI and core workflows of the React frontend  
- RESTful API endpoints of the Node.js/Express backend  
- Integration between client and server

## 2. Test Coverage Areas

|  |  |
| --- | --- |
| Frontend (Cypress) | Backend (Postman) |
| ✔ Login (valid/invalid) ✔ Create Item (valid/invalid) ✔ Edit Item (valid/invalid) ✔ Delete Item (valid/invalid) ✔ UI Snapshot validation | ✔ POST /login  ✔ GET /items ✔ POST /items ✔ PUT /items/:id ✔ DELETE /items/:id ✔ Negative test cases |

## 3. Tools Used and Why

- **Cypress:** Chosen for its powerful end-to-end UI testing capabilities, fast feedback loop, and built-in snapshot support.  
  
- **Postman & Newman:** Ideal for API test automation with a wide user base and CLI support for CI/CD pipelines.  
  
- **GitHub Actions:** Seamless CI/CD integration to automate test execution on every push.

## 4. How to Run the Tests

1. Clone the repository and install dependencies using `npm install`.  
2. From the root folder, start the backend and frontend: `npm start`  
3. From the root folder, execute the Cypress UI tests: `npm run cy:run`  
4. From the root folder, run the Postman API tests: `npm run api:test`  
5. To run both in CI/CD, push your changes to GitHub and GitHub Actions will trigger the pipeline automatically.

## 5. Assumptions and Limitations

- Assumes backend and frontend services are correctly configured and reachable at time of testing.

- API test results are automatically exported as HTML and JUnit reports during the CI/CD pipeline, ensuring clear visibility of test outcomes and enabling integration with QA dashboards or test analytics tools.

- Tests are designed for demo/sample purposes and may not reflect all edge cases in a production environment.

- Snapshot tests are only visual validations; they do not detect dynamic DOM logic regressions.