Sample Test Plan – *Todo App Automation Project*

Project Overview

This test plan outlines the QA automation strategy for a simple full-stack Todo application with a React frontend and Node.js/Express backend. The app includes authentication and CRUD operations for todo items and tags.

---

1. What Is Being Tested

- **Login Functionality**
  - Valid and invalid user credentials
- **Todos API**
  - Create, Read, Update, Delete todos
- **Tags API**
  - Fetch tags for categorizing todos
- **UI Behavior**
  - Form submission
  - Dropdown selection
  - Real-time rendering of todos

✅ 2. Test Coverage Areas

Feature	Test Types	Status
Login	UI + API (positive & negative)	✅ Done
Create Todo	UI + API (positive & negative)	✅ Done
Edit Todo	UI + API (positive & negative)	✅ Done

Feature	Test Types	Status
Delete Todo	UI + API	✅ Done
View Todos	API	✅ Done
Tags Dropdown	UI	✅ Done

✅ 3. Tools Used

Tool	Purpose	Reason
Cypress	End-to-end UI testing	Fast, real browser support
Postman	Manual & automated API testing	Easy assertions, scripting
MongoDB Atlas	Database backend	Cloud-hosted, easy access

✅ 4. How to Run the Tests

🔷 UI Tests with Cypress

```
npx cypress run
```

🔷 API Tests with Postman

The collection of the test results is added

🔷 Main project

Frontend and backend merged together

```
npm run dev
```

✅ 5. Test Data Strategy

- Pre-created test users (email & password)
- Email-try@yopmail.com
- Password -123456
- Tag IDs retrieved via /api/tags
- Randomized todo texts (via faker or dynamic strings)
- Unique data per test to avoid collisions

✅ 6. Assumptions & Limitations

- Application must be running at http://localhost:5000
- Token-based authentication assumed for all secured endpoints
- MongoDB must be seeded or pre-connected before API tests
- No newman tests just postman
- CI integration is not configured, but scripts support it

Final Notes

- All major flows are covered with both positive and negative tests
- Tests are reproducible using the steps above