Test Plan – Todo App Automation Project (with Newman & .env support)

Project Overview

This test plan outlines the QA automation strategy for a 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
Delete Todo	UI + API	Done
View Todos	API	Done

Feature	Test Types	Status
Tags Dropdown UI		Done

---

3. Tools Used

Tool	Purpose	Reason
Cypress	End-to-end UI testing	Real browser automation and fast test writing
Postman	Manual API testing	Easy environment management and scripting
Newman	CLI-based API test automation	Runs Postman collections in terminal and CI/CD
dotenv	Environment variable handling for Newman	Securely injects auth tokens and base URLs
MongoDB Atlas	Cloud database	Reliable cloud-hosted DB for testing data

---

4. How to Run the Tests

UI Tests with Cypress

npx cypress run

API Tests with Postman (Manual)

Open Postman collection → Run in Postman UI with pre-configured environment.

Automated API Tests with Newman (Using .env)

1. Create a .env file with variables such as:

BASE_URL=http://localhost:5000

AUTH_TOKEN= found in postman folder in github

2. Reference variables in your Postman environment or pre-request scripts.
3. Run with:

```
newman run postman_collection.json -e postman_environment.json --env-var "AUTH_TOKEN=$AUTH_TOKEN"
```

🔐 Tokens are injected securely without hardcoding them into the collection.

Start App (Frontend + Backend)

```
npm run dev
```

---

5. Test Data Strategy

- Pre-created test users:
    - Email: try@yopmail.com
    - Password: 123456
- Tag IDs are dynamically fetched via /api/tags
- Use of faker.js or dynamic values for todo text and uniqueness
- .env file securely stores sensitive auth tokens and environment variables

---

6. Assumptions & Limitations

- App must be running at http://localhost:5000
- All secured endpoints require a valid Bearer token
- Tokens are retrieved and stored via login requests or .env before running Newman
- MongoDB must be seeded or pre-connected before tests
- CI integration is not yet configured, but supported

Final Notes

- Major functional flows are covered with both positive and negative test cases
- API tests are automated using Newman with secure token injection via .env

- Test suite is reproducible, scalable, and CI-ready