**Test Plan / Test Strategy Document**

**1. Overview**

The test plan and strategy included in the following document refer to automated testing of the Notes App, a React frontend and a Node.js backend API. This will be achieved by ensuring that the core functionalities are tested by automating UI and API.

**2. What is Being Tested**

User Authentication

- Signup and login flows.

Notes Functionality

- Creating, editing, and deleting notes.

Data Integrity

Data- must be presented in the display UI correctly after proceedings.

API Endpoints

- CRUD operations for notes and user authentication endpoints.

**3. Test Coverage Areas**

Frontend (UI) Functional Tests

- Login with valid and invalid credentials.

- User signup.

- Creating a new note.

- Editing an existing note.

- Deleting a note.

Backend (API) Tests

- POST /user/register (User registration)

- POST /user/login (User login)

- GET /note/ (Fetch notes)

- POST /note/create (Create note)

- PATCH /note/ (Update note)

- DELETE /note/ (Delete note)

- Positive and negative test cases for each endpoint.

**4. Tools Used and Why**

Cypress

To automate the React frontend end-to-end UI. It has been chosen due to its ease, quick feedback and support of modern web applications.

Postman

For automated API testing of the Node.js backend. Postman is user-friendly for manual and automated API tests.

**5. How to Run the Tests**

UI Tests (Cypress)

- Start the backend (npm start in backend directory).

- Start the frontend (npm start in frontend directory).

In the frontend directory, run:

- npx cypress open

-

API Testing using Postman

Postman: Add a new collection with a sequence of requests performed.

**6. Assumptions and Limitations**

UI tests require backend and frontend to be started locally.

Unless the database is reset, test data will survive between runs.

Cypress tests could require selectors adjustments when the UI differs.

API tests presuppose that the API of the backend follows the specification in the endpoints and shapes of requests/responses.

Visual regression and code coverage reporting are an option in CI, but it might need extra setup to operate on local build.