



Assessment - Quality Engineer

You have been provided with an application and its requirements. From these you will be required to answer the relevant questions.

Commit your answer to Git with full instructions on how to run the tests and/or any other details necessary.

TODO List Application

Stitch has been requested to create a simple todo list application for its users so that they can manage their time more efficiently. The developers have just completed development for the required features and the app is ready for testing.

The app is currently a web app. The following requirements have been passed on from Product into the Team.

- The user is able to add todo's
- The user is able to edit todos's
- The user is able to delete todos's
- Todo items should not be empty
- The user is able to complete todos in which case they don't get deleted and instead get a strike through
- Once a todo has been completed it cannot be deleted
- A todo cannot be longer than 100 characters. If it is, an error message should be thrown to the user

- It has been determined that the peak load the application is expected to handle is the action of adding 10 Todo Items per second. This is usually expected to occur between 10am and 12pm.

You will find all the necessary information on how to deploy the app locally in the readme of the provided repositories.

Backend: <https://github.com/Stitch-Money/ToDo-list-api>

Frontend: <https://github.com/Stitch-Money/todo-front-end>

As a tester, the following tasks have been assigned to you.

1. Create a test plan for the todo application.

This can be in any format you like eg. Markdown/text file/Word Document)

2. Create automated tests for the UI of the todo-list.

You may use any language/framework. Please provide detailed instructions in the README.md file on how to execute your tests.

3. Create API tests for the backend for the mentioned features.

You may use any tool you prefer. Please provide detailed instructions in the README.md file on how to execute your tests.

4. Log any bugs found in the application.

You may log them in any format you choose.

5. (Optional) Create performance test(s) for the performance requirements of the application. This can be written using any tool you prefer. Alternatively you may just state how you would go about the performance testing of the application.

Tools:

Tools you will possibly require:

- Git
- Node.js v16
- Npm
- Nvm
- Editor of your choice