# Relayr QA Task

**Task 1 :**

**Imagine the following situation: you need to establish a QA process in a cross-functional team. The team builds a front-end application using REST APIs.**

## 1. Where would you start? What would be your first steps?

I will consider below-mentioned thoughts when QA process has to be established between the cross-functional teams:

- Identifying the complexity of the Project and then the Team size should be decided
- We have to consider if the current team matches the skills set required for achieving the business requirements
- If there are no skill sets matched then, training or learning must be needed
- Align with stakeholders, project manager and Development Lead.
- Acceptance criteria must be clear for each feature to be tested.
- Choosing proper tools to be used for testing the REST API's & to Automate them & for Bug Tracking
- Creating Test plans & Test cases

## 2. Which process would you establish around testing new functionality? How would you want the features to be tested?

- Firstly, for testing any new functionality the acceptance criteria have to be met which was approved by the Product Owner & Stakeholder.
- Later, once approved QA team should figure out all relevant test scenarios and then break them into Test Cases.
- In these days, Test cases are always written in Gherkin Language(given, when, then) format so that it is easily understandable by non-technical people
- Brainstorming should be done by the QA team for identifying all possible test cases in a new feature.

Also, I would recommend all this to be followed on the principle of Agile methodology and make sure that it is implemented.

## 3. If you would do test automation which techniques or best practices would you use the application?

For test automation, Choosing a suitable Framework which can be reliable, scalable and must eliminate duplicity is very necessary.

I would choose **Pytest** as a framework because of the following advantages:

- Pytest is free and open-source.
- Because of its simple syntax, pytest is very easy to start with.

- Pytest has its own way to detect the test file and test functions automatically
- Pytest allows us to skip a subset of the tests during execution. It has inbuilt functionality called Markers which allows Grouping tests and so we can mark the tests as Regression tests and sanity tests and so only those tests can be run when required by skipping other tests
- Pytest can run multiple tests in parallel, which reduces the execution time of the test suite. Xdist is a popular plug-in which enables Pytest to tests in parallel
- Command Line Execution allows several options to run tests from the command line.
- Pytest supports parametrization.
- Pytest has a feature called Hooks which helps to run a test setup, tear down, and log events of the test.

And so I would use Python as a language because of the following reasons:

- Python is simpler and more compact compared to Java and easily readable
- Runs faster while comparing other programming languages
- Java uses traditional braces to start and ends blocks, while Python uses indentation.

I Would prefer Pycharm over Eclipse as Pycharm is an IDE which is a Intelli J Idea community edition and is also Open source like Eclipse

While coming to the Implementation of Automation scripts for UI, I use the Page object model design pattern as it makes the code more reliable, maintainable, readable and it eliminates a lot of duplicities!