**Problem Statement:**

Let’s assume that there’s a project with a developer working on the API and some other developer (could be outside the project) who’s using this API and he discovered a malfunction or a bug with the API in this case the API user will have to inform the API developer of this bug so that the API developer can fix it and then re-inform the developer that the bug has been fixed.

Putting in mind that this process might be repeated several times and also there could be more than one API (possibly hundreds or even thousands of APIs in development). We can see how time and effort consuming this process is.

This is where the API tester comes in hand by testing the API and making sure it’s working properly before sending it back to the user/developer.

First problem here comes in testing the app since RESTful API requests naturally work between clients and servers where the client sends a request to the server and the server responds back with the data, the problem here is that in order for the API tester to do his job the API user will have to connect him with the client app so he’ll be able to do the GUI testing and this is where costs increase and the tester will have to do the testing work on the app GUI.

The tool here will simulate the client by sending requests and showing the response to the tester, and making it even easier by adding test automation with test suits and cases, this way the tester won’t have to use the client app GUI by just using the tool instead.

Second problem here comes in writing the documentations for the API since the API user needs to know the API’s endpoints, what they receive, the responses, the response’s format and so on.

Writing documentation manually includes manually writing what each endpoint receives and returns and their descriptions which is very expensive in terms of time and cost and unnecessary extra work for the tester. And any misspelling or fault could lead to a confusing or ambiguous API for the consumer.

With the API testing automation tool, the tester doesn’t need to worry about writing the documentation the tool will auto generate the documentations once the tests reach a full coverage.

Another problem with documentation is version controlling in CI/CD where we will need a new different version of documentations for each version of the API for the later maintenance.

Third problem is communication between collaborators since they don’t have to be all working under the same company or organization since the API developer have to inform the tester whenever he changes or adds new features to the API and then the tester will need to re-inform him back with the test results and the documentations for the API users thus we need a way of connecting all of the developers and testers in one platform and this is where the team collaboration part of the tool comes in handy.

**Project Field:**

1. Platform as a service (PAAS):

Is a category of cloud computing that provides a platform that allows the customers to develop, configure, and run the applications without the complexity of managing the infrastructure.

The advantage of (PAAS) is that it allows for higher level programming with dramatically reduced complexity and maintenance of the application will be much easier.

1. Software testing:

Is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test, it is necessary because we all make mistakes, some of those mistakes are unimportant, but some of them are expensive or dangerous. We need to check everything and anything we produce because things can always go wrong.

**– (humans make mistakes all the time)**

**Project Objectives:**

* The project aims to simplify and reduce the costs of a RESTful API testing by automating the tests and providing the tests output in a clean and simple way.
* Significantly reducing the time taken to write the documentations of the API by auto generating it after each successful testing.
* Easing the communication between the project stakeholder by providing a list of API endpoints for the front-end developer to synchronize the development flow between the back-end developer and the front-end developer, and providing continuous reports of the projects progress to the administrator or the customer.

Project Importance:

1. Saves the overall cost of the project:

Using this tool for testing can significantly reduce the overall project cost by automating the tests and dramatically reduce the time taken to produce the documentations for each version of the API.

1. Increases the security level of the project:

Because testing increases the reliability and makes the product more trustworthy by

eliminating bugs and vulnerabilities earlier on.

1. Increases the quality level of the project:

Since discovering bugs and fixing them automatically increases the quality.

1. Increases the Customer satisfaction.

Increase the customer satisfaction by ensuring the finishing product meets his requirements.

**Project Scope:**

User can use our software by creating an account in our platform after that he/she can create a project and this project will include an admin (the creator of the project), developer or tester in our scope we expect the developer to bet the one who’s testing the API endpoints because we focus on small teams and groups and we use DevOps method that’s based on small teams and the consumer of APIs endpoint (front-end developer that use APIs in client’s platform).

Admin sends invitations to all stakeholders in their email if they already have an account in our platform after the accept invitations any stakeholder has roles and responsibilities.

Developer or tester create request by providing request information’s and add title and descriptions for each request then he/she send request and validate response against what is required and even test performance by checking latency of request.

After that developer/tester can automate a list of requests by giving each list excepted status of response and then all tests will execute to create coverage.

If the test coverage reached %100 that means all tests have passed then the Developer/Tester can generate documentations a first-generation called first commit to documentations after this commit the consumer of this API will notify and this doc will be available for him.

Also, admin will be aware of commits and versions, developer or tester can create a release after a set of commits when development reaches versions stage, each version includes release note about what new in this version.

The consumer of API will get the last version or last update of documentations to help them keep update with the developer of the server and even platform create a to-do list of all endpoints in documentations. if the consumer connects with the endpoint, he/she must check it in a to-do list to help admin to know about front end progress in case of API connection.

**Project methodology:**

* Programming languages used:

1. **vue.js (JavaScript) for the front-end development:**

VueJs is a progressive framework used for building user interfaces and also perfectly capable of powering sophisticated Single-Page Applications.

We choose VueJs because of the high flexibility rate which allows for code reusage thus decreasing the development time needed also a chance for us to learn more into vue since the learning curvature is very gentle and easy.

1. **NodeJs (JavaScript) for the back-end development:**

NodeJs is a JavaScript runtime environment which allows the infrastructure to build and run as an application.

There are quite a lot of reasons of why we chose node for this project. It’s fast, lightweight ,very scalable ,cross platform , and extremely efficient but the main reasons are the facts that it’s just a lot faster for building real time apps (which is the case for our tool) and it eases the integration since we’re using the same programming language for both the front and the back end and last but not least the ever-growing choices of npm packages that’ll help to dramatically decrease the development time.

* Software used:

**JetBrains WebStorm for both front-end and back-end:**

WebStorm is a powerful ide used for modern JavaScript development and it provides full support for JS, TypeScript, HTML, and CSS as well as frameworks such as VueJs (Which we’re using as well) right out of the box.

We choose WebStorm for lots of reasons including debugging capability, easy code refactoring, GIT source control support, and great coding experience in general.

**\*we might also use vs code for fast editing and small changes into code.**

* Project Management Method:

**Scrum**: is a framework for project management that emphasizes teamwork and iterative progress through the project it’s mostly a part of the **agile software development**.

Scrum processes include daily and work dependent meetings, and reviews.

This provides commitment, synchronization between team members, openness and decreases time needed for tasks.