As the Quality Engineer for this project, I feel the below testing can help to improve the quality of the project. Considering this project is in Agile Scrum and based on Scrum framework shift-left approach is the best approach to validate, observe and fix the issues in the earlies phases of the sprint.

According to test pyramid unit test is the least expensive test, which should cover all the conditions, branches and loops to satisfy optimal amount of code coverage is done. This phase the error/bugs are easy to fix. Component testing is a form of closed-box testing, meaning that the test evaluates the behaviour of the program without considering the details of the underlying code. Component testing is done on the section of code in its entirety, after the development has been completed. Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before system testing phase when one module integrates with other/base code then Top down, bottom up or Mixed mode integration testing is helpful to integrate one module with other. Then API testing is helpful to validate the functional and non-functional testing. Both functional and non-functional testing is important to validate the application. The final and last layer comes as the exploratory testing.

According to the CI/CD environments setup the testing needs to be distributed in all the environments. But before Production all the test needs to be done to make sure that the code has gone through sufficient amount of testing to satisfy that there is no major blocker and code is not braking in any point. The observed bugs or enhancement should consider in the consecutive sprint based on the dependency and priority. Sprint release should keep until the whole product milestones are not achieve.

Above are the overall system as a whole it works but testing starts when the inception phase stats and the stories are written. The QA and the lead need to decide what type of testing is required at that phase and how much testing is required. Based on this decision release documents and checklists are prepared. Once the PO decides the scope of the sprint and BA writes the ACs for the JIRA stories, accordingly QA writes the testcases. Then the test case needs to be written and reviewed based on the Jira ticket acceptance criteria. QA and dev lead need to decide whether non-functional testing is required at that phase or not bases on the CI/CD environment. For testing the functional requirements, it is necessary to have the API-UI and DB testing as well as the other possible end points. To know that the whole flow is working appropriately in all the end points. The non-functional requirements will help to check the performance, usability, reliability, security etc

Approach:

Automated Test cases:

Scenario 1- AC1- Completed positive scenario

Scenario 2-AC1- Completed positive scenario

Scenario 3-AC1- Completed positive scenario

AC2- Completed positive scenario

AC3- Completed positive scenario

Scenario 4- AC1- Independently done

AC2- Independently done

AC3- Independently done

AC4- Independently done

AC5- Independently done

AC6- Independently done

Python script is also written but integration is left to validate the final result

Scenario 5- AC1- Completed positive scenario

AC2- Completed positive scenario

AC3- Completed positive scenario

Based on this scenario scope is limited as few ACs are incomplete and few functionalities are not justifying the functional outcome as per the story. Swagger API documentation is not proper and optimal to cover the functional testing.

AC1: 1) There is no proper error for unsupported file format.

2) There is no proper successful message after uploading the file.

3) There is no place holder to show the file has been uploaded successfully.

4) UI is not user friendly.

5) Swagger API document doesn't cover all the error message with example.

6) Swagger documents doesn't cover the primary and optional data fields.

AC2: 1) I have checked uploading 300+ records and in the 2nd attempt it is uploading.(Non-functional Testing)

2) The last row data is missing.

3) Redundant data entry is possible.

4) ACs are not written as per the user stories format.

Installation Steps:

Pre-requisite:

1) Java need to be there in the system

2) Should have a GitHub account

3) Git should be there in the system to pull the code in the local machine

Steps:

1) Install pip

2) Check the PIP version and it should be the latest version(PIP --version)

3) Install python using PIP

4) Set the Python path in the environmental variable path.

5) Check the Python version using Python --version

6) Install robot framework using PIP

7) check the location for robot framework and update the same in environmental variable

8) Install robotframework-seleniumlibrary

9) Install the selenium Chrome driver and set the path in the environmental variable

10) check all the installation using PIP -list

11) Install IDE(PyCharm)

12) Install the dependent libraries

13) Crate a Project and crate files based on your test Scenario

14) Further install the libraries based on your code need

15) Install the plugins from the market place

Command to execute the script:

Robot <filename>