Questions:

1. If you chose to use a tool or language other than the recommended, briefly explain why.
2. How did you go about locating the elements for your tests?
3. What do you believe are the most common causes for instability in UI automation?
4. How do you make your tests consistent and easy to debug?

Answers:

1. In this project, we have used all the recommended tools i.e. Cucumber, Ruby, Selenium for the UI Automation and Pytest, Python for the API Automation. However, we also use Watir Framework to automate the UI application to have the best capabilities with Ruby programming language. Although it uses selenium under the hood, for the lines of code become less and clearer to understand.

2. Since we have a complex DOM in our application and this would need a customizable locators. I find the locators with the help of XPath for elements that are within a complex node in the DOM. If having an ID, name, class or any other element that results in being unique to the page, we can prefer that element as well. The best practice is to define an XPath, such that the XML Query will always find the right element. Hence, my preference is XPath.

3. I believe the most basic inconsistency are as follows in UI Automation:

- **Waits:** The automation can sometimes fail because of inconsistent wait period. Sometimes, the elements that would have to appear just would not be visible to the WebDriver causing a fail. However, with the best use of implicit and explicit time waits with the use of selenium could be resolved.
- **More focus on Locator:** There must be more focus on locating an object rather than verifying the visibility of a string. This will make sure the element is definitely present and could be accessible/clicked.
- **Dynamic content on the webpage:** The test cases must handle all critical aspect of the test boundary cases. In some cases the content keep changing or the page navigates to another page than what is expected. In order to ensure the dynamic content is handled well, testers writing the BDD scripts must ensure to use implicit/explicit time waits are used and handle the boundary conditions in an intelligent scripting which will clear any given test scenario.

4. I will make my tests consistent by the following ways:

- Write test scripts with a combination of implicit and explicit waits.
- Make sure the dynamic content of the web pages are handled.
- Design customized XPath of an object to have it though roughly located by then web drivers.
- User variables in step definition that would aptly describe the instance
- Using binding.pry in step definition to debug the particular step definition the fail occurs.