My suite is developed using SpecFlow with Selenium C# and nUnit framework for the assertions.

IDE Visual Studio.

The solution is designed in Page Object Model. To make it simpler I have all the steps in one class (ProjectSteps) and all the methods in another one (Actions), but in a real situation I would have different classes for different pages (one for homepage, another one for Solutions and Services, if needed…)

Please notice that the way in which the solution is designed would allow to create a whole suite to navigate and check headers and paragraph start in sections with no more code needed. It would enough with copy the created scenario and change its parameters. For instance, we could use the step: Given the user expands 'Technologies' option in top menu

To display that menu instead of the one in ‘Solution & Services’

**How to run the test**

I’m sending a solution so the build and the run can be done in VS.

As the build is created in the solution, the suite can be run too from Command line. Going to the directory in which the .dll is, and running the line:

dotnet vstest Assessment\_SP.dll --logger:trx

Text

Description automatically generated

It will indicate where the result logs are (\…\Assessment SpanishPoint\Assessment\_SP\bin\Debug\net6.0\TestResults)

Anyway, this would be a temporary solution. The best way to run a suite of automated tests is to have unattended jobs. Currently, I have my suites run by a DevOps pipeline which runs Jenkins tasks nightly.