Test Setup and Run

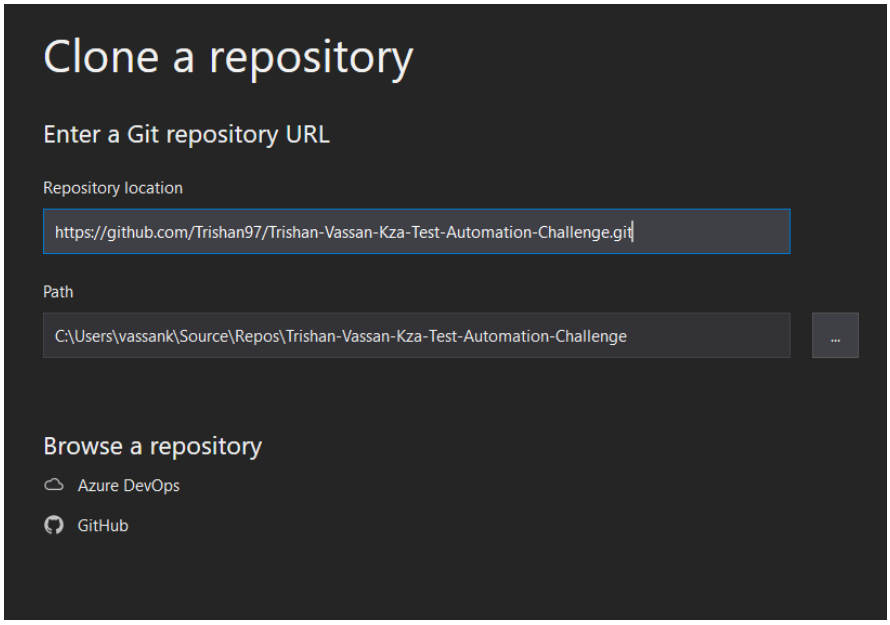
How to Setup Repo, Run Tests and Generate test results

1. Clone the repo if not done yet

cd to desired folder then run :

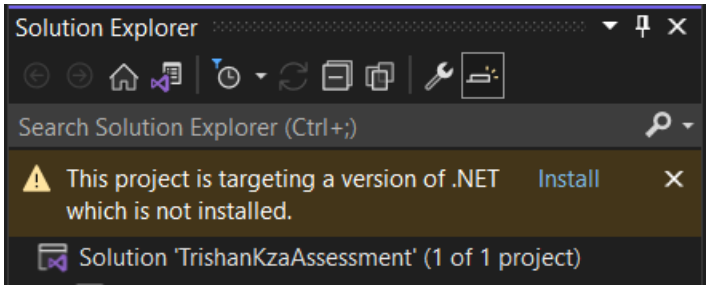
git clone https://github.com/Trishan97/Trishan-Vassan-Kza-Test-Automation-Challenge.git

or through Visual Studio or any other C# compatible IDE (e.g. Rider for Mac)

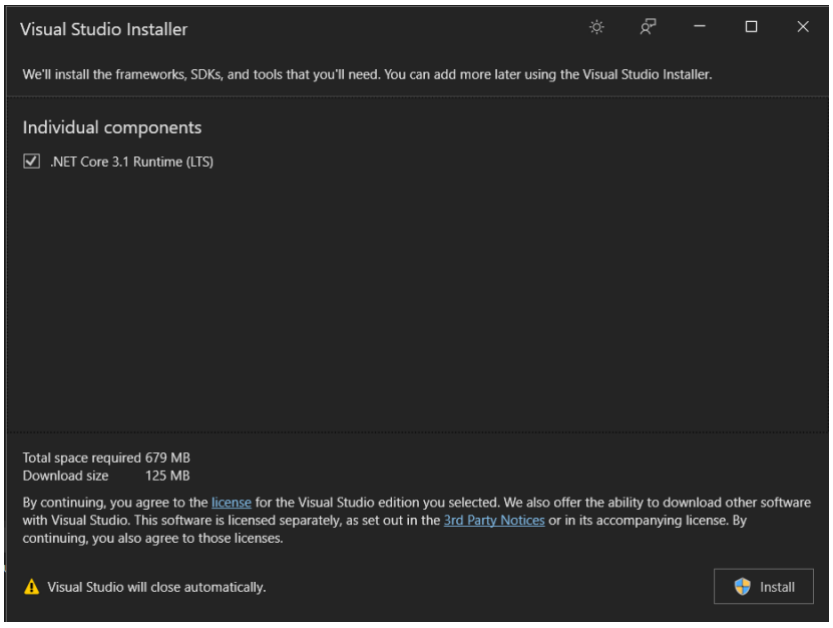


2. Once the Repo has been successfully cloned , open up the **TrishanKzaAssessment.sln** solution file if it has not automatically been opened

   Once opened you may get a prompt to Install the targeting version of .NET if you do not have it installed yet :

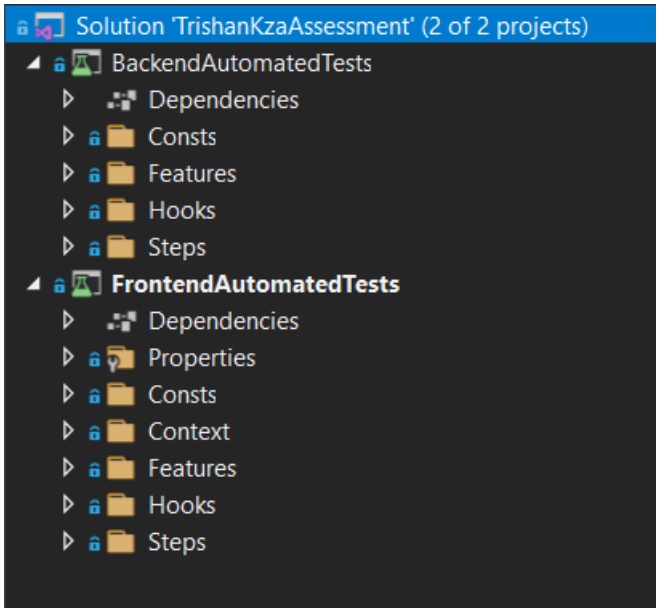


   This is only relevant if you have received the above Prompt to Install the targeting version of .NET Core 3.1 Runtime (LTS) through the visual studio installer :

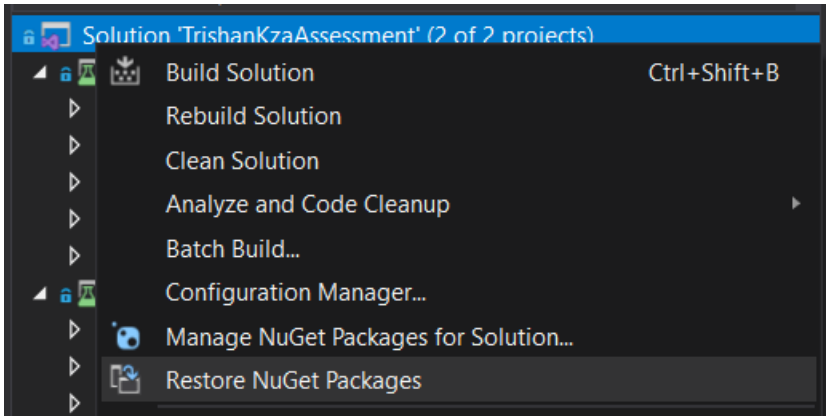


   You may need to reopen visual studio and the project after installation and open the **sln** file again

3. Once that dependency has been installed successfully your solution should open up and look like this :

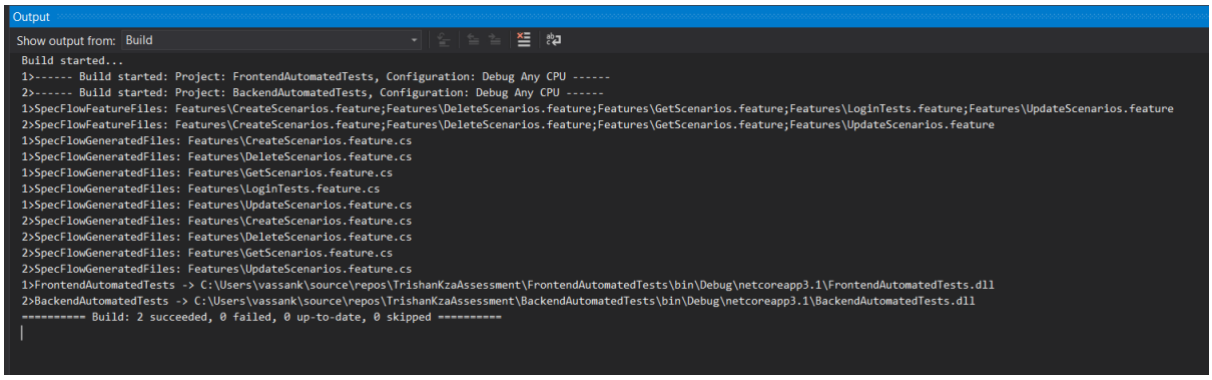


4. The next step is to restore Nuget packages for the solution , this can be done by right clicking on the Solution file **Solution 'TrishanKzaAssessment' (2 of 2 projects)** and clicking on the **Restore NuGet Packages** Option :

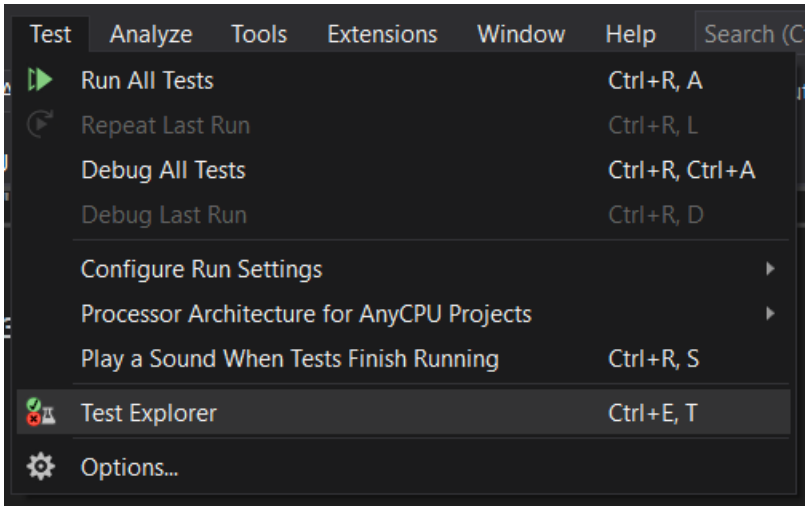


5. After that has been successfully completed the next step is to build the solution, this can be done by right clicking on the Solution file **Solution 'TrishanKzaAssessment' (2 of 2 projects)** and clicking on the **Build Solution** Option

   You can verify that the solution has built successfully by looking at the output window and getting a build succeeded message :



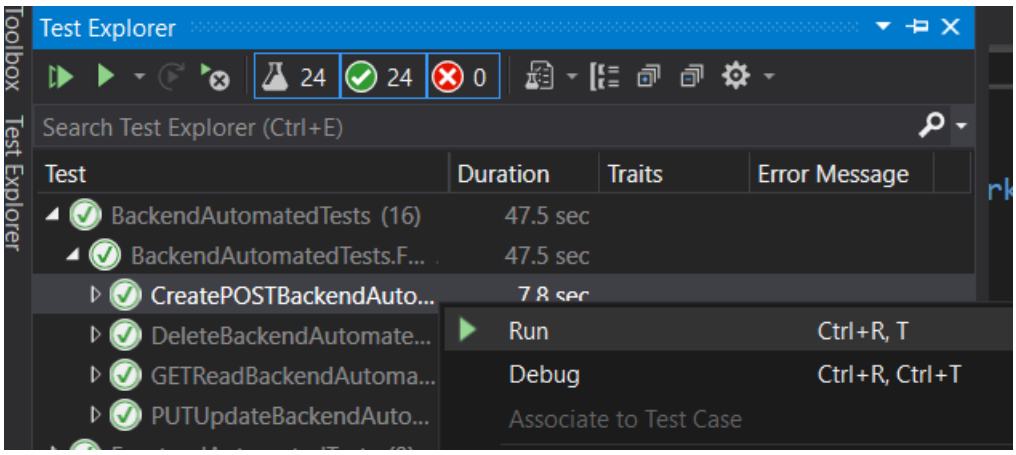
6. The next step is to click on the **Test** option in the top ribbon and then select the **Test Explorer** option



   This will then open the **test explorer view** with all available tests, and view each scenario in more detail by clicking on the little arrow next to each Test case

7. In order to run the test you can either:

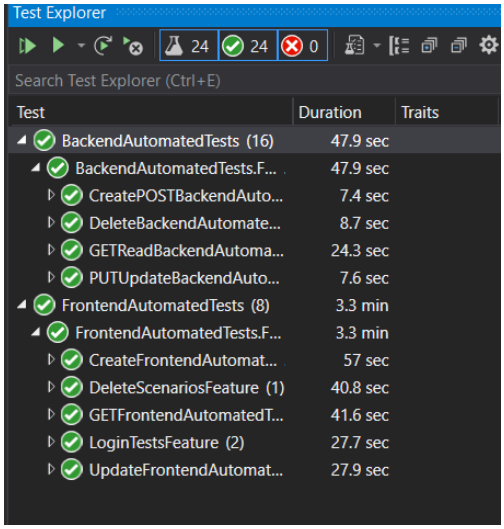
   - Run an individual test by right clicking on a test case you would like to run, then selecting the **Run** option



   Or

   - Run all tests by selecting the **Run all tests in view** option on the top left of the Test Explorer

8. Once you have run either individual tests or all test cases the results will be displayed in the test explorer with the pass or fail status as well as how long each test took

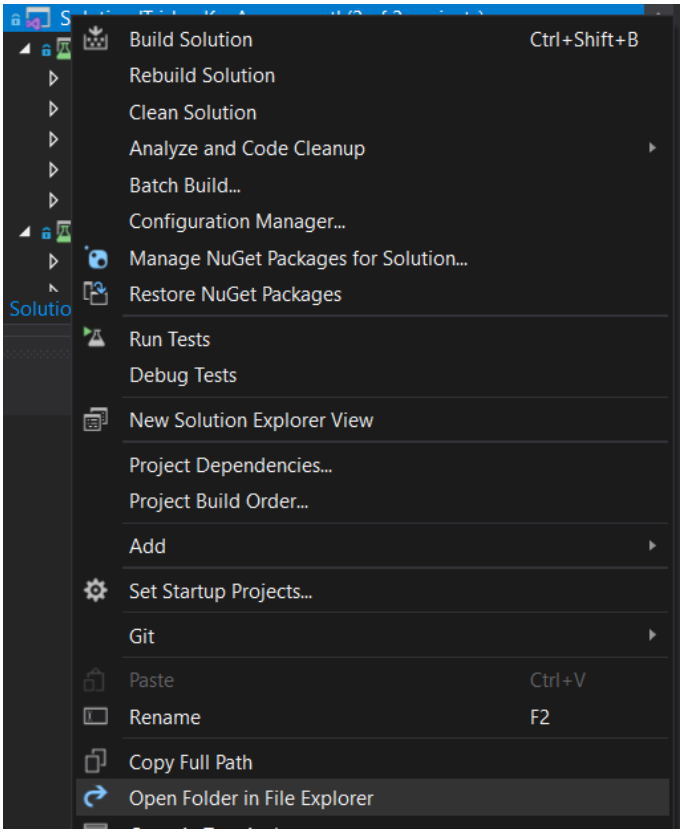


9. These next steps are **Optional** if you would like to get a current test report of the tests you have just run. This can be done so by installing **Specflows' living doc CLI** and then following the remaining steps listed. Also please note that a current test report can only be generated **once a test run has been completed** (this can be a single test, selected test or all tests) but a test does need to be run before completing the following steps for either the Frontend or backend :

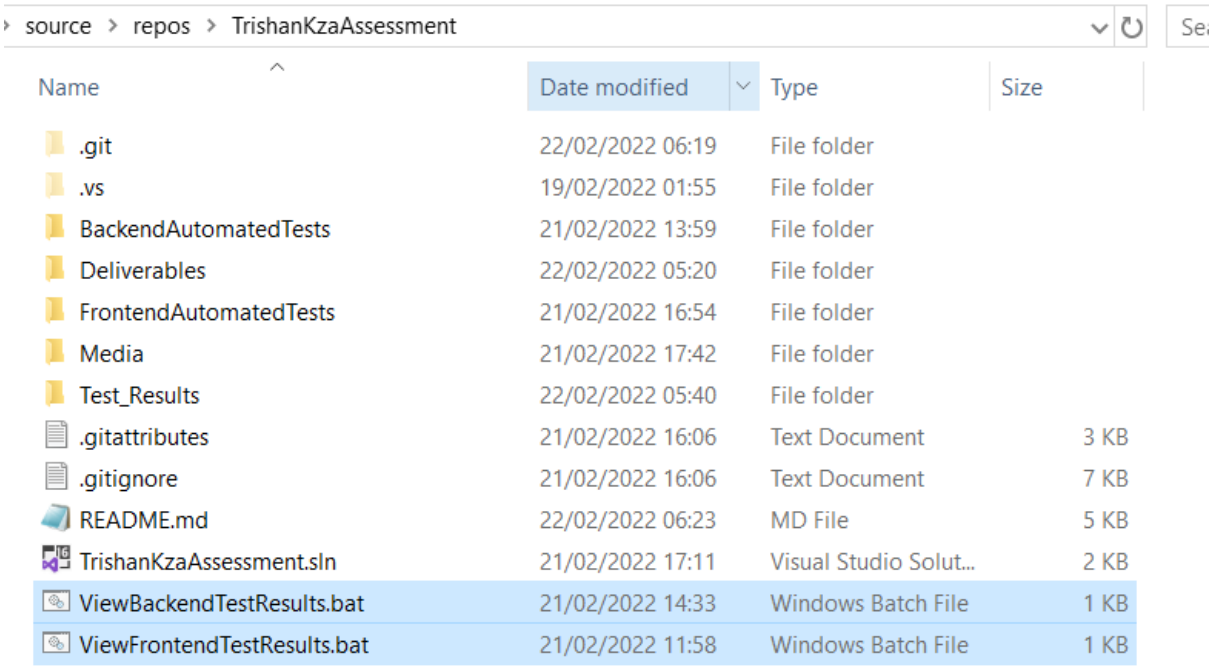
    - Open a command prompt and install living doc with this command :
      dotnet tool install --global SpecFlow.Plus.LivingDoc.CLI

    - Link to LivingDoc Documentation

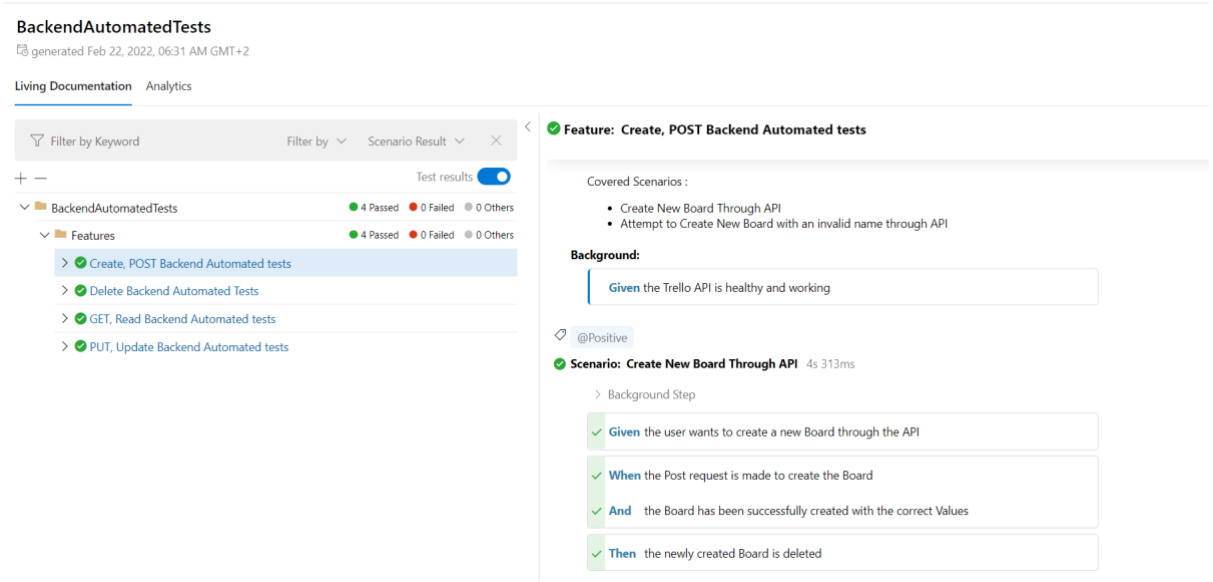
    - Once that has been successfully installed, navigate to the folder where the repo has been cloned to, an easy way to do this is by right clicking on the Solution file **Solution 'TrishanKzaAssessment' (2 of 2 projects)** and selecting the **Open Folder in File Explorer** option :



- This should then open up the following folder :



- Finally you can open either the **ViewBackendTestsResults**.bat file or the **ViewFrontendTestResults.bat** file to view the results of the automated tests in a nicely presented GUI format e.g. :



- What the batch files execute for Frontend report:

```
cd FrontendAutomatedTests\bin\Debug\netcoreapp3.1

livingdoc test-assembly "FrontendAutomatedTests.dll" -t TestExecution.json
LivingDoc.html
```

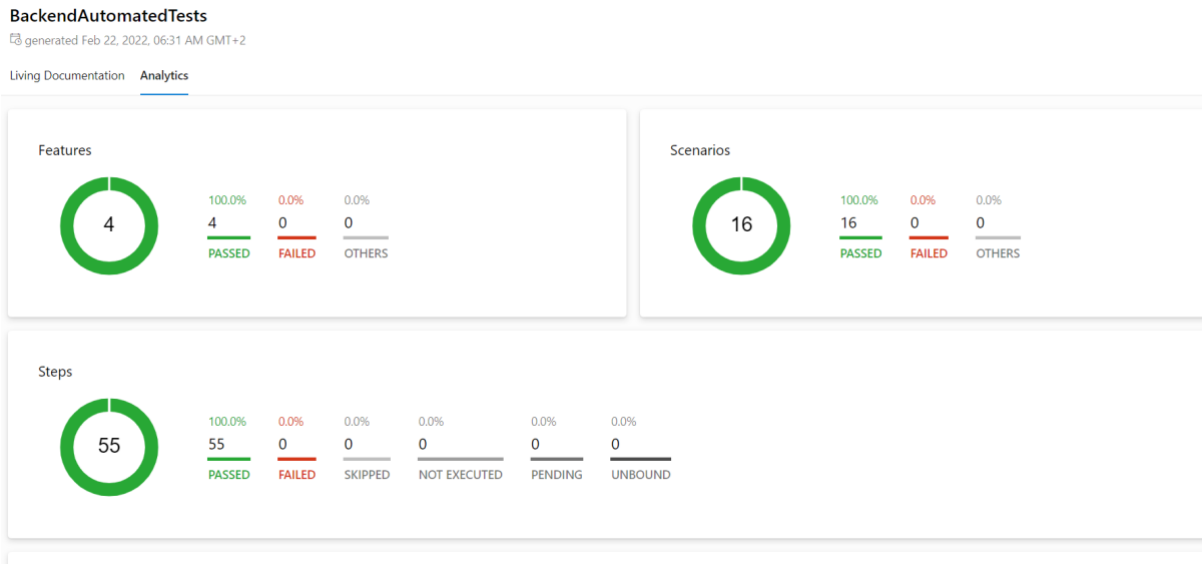
- For Backend Test Report

```
cd BackendAutomatedTests\bin\Debug\netcoreapp3.1

livingdoc test-assembly "BackendAutomatedTests.dll" -t TestExecution.json
LivingDoc.html
```

Please Note again that the above **TestExecution.json** file will only be generated once a test run has been completed and the report generation can only be done through the batch file if the **SpecFlow.Plus.LivingDoc.CLI** has been installed as stated in one of the previous steps above

- From these reports you can view in detail each scenario and additional information about each, there is also an analytics page which shows the tests run vs tests passed or failed, which can be useful when presenting test results to a client, team or stakeholder:



A full recording of the test running and report generation process is available on the Readme file in the git repo : https://github.com/Trishan97/Trishan-Vassan-Kza-Test-Automation-Challenge#screen-recording-showing-backend-test-running-process-and-test-report-generation