TFL Coding Challenge.

The below sections highlights of the process of creation of TFL coding challenge using

NUnit C# framework and follows data driver development approach and ensuring data protection by implementing concepts like encapsulation, inheritance etc.

* Created a NUnit project
* Added all necessary packages from Manage NuGet package manager.
* Created a test class on Tests folder namely – Tests > Testcases.cs
* Created a utilities folder.

1. -BaseClass.cs – contains [setup][teardown][onetimesetup][onetimeteardown] methods in order to reduce code redundancy.
2. TestData.json – Contains test data required for the testcases to achieve DDD
3. -JSonReader.cs – This file is used to parse through the TestData.json file

A screenshot of a computer

Description automatically generated

* Test cases are developed using NUnit features such as [OneTimeSetup],[OneTimeTearDown], [Test] etc. created in Utilities folder
* [OneTimeSetup] – contains browser handles
* [oneYimeTearDown] – contains browser instance closing methods
* [Test] – contains testcases
* The first 3 testcases – Test1, Test2, Test3, are triggered using [OneTimeSetup],[OneTimeTearDown], as they 3 run on a same instance ensuring continuity of the test scenario.
* Test 4 has a initiate driver method are it needs to be executed on new instance.
* Test 5 Test 4 has a initiate driver method are it needs to be executed on new instance.
* BaseClass.cs is inherited in TestCases.cs class inorder to use its properties
* Pageobjects.cs – contains all the locators of the elements achieving encapsulation
* In TestExplorer, we can see the 5 tests that are developedA screenshot of a computer

  Description automatically generated