**Framework Details:**

* IDE: Visual Studio
* Unit Test Framework: MSTest
* Automation Framework: Selenium
* Language: C#
* Tools Needed: Selenium.WebDriver, Selenium.Support, Selenium.WebDriver.ChromeDriver, Selenium.Chrome.WebDriver, Microsoft.NET.Test.Sdk

**Steps to create the Framework:**

1. Create a project with MSTest template in Visual Studio

Graphical user interface, application, Teams

Description automatically generated

Install the below tools from the NuGet Package manager

* Selenium.WebDriver
* Selenium.Support
* Selenium.WebDriver.ChromeDriver
* Selenium.Chrome.WebDriver
* Microsoft.NET.Test.Sdk

**Test Approach:**

The framework is designed based on Page Object Model.

The framework has different folders:

1. Tests folder: holds the class file containing all the test cases
2. BaseTest folder: holds the class file containing test initialization and test closure that runs before and after each test case
3. Pages folder: holds the class file containing re-usable methods to execute the test cases
4. TestData folder: holds the holds the class file containing the test data used in the test/methods
5. PageLocators folder: holds the holds the class file containing the locators of the elements

The test cases included in the framework are as follows.

Note: [run the tests in the same order]

1. A\_UserRegistration
2. B\_LoginUser
3. C\_UserProfileUpdate
4. D\_ResetPassword
5. E\_RegisterExistingUser

Description of what each tests / method do is added in the summary section.

The above tests cases are considered as they come under the most critical functionality Registration and Login to application.

