**Hands-On: Stage 3 – .NET Advanced Features** ï **NUnit Day 1 - Session-3**

1) Create a UtilLib project in the same solution which is provided and name it as suggested.

2) Rename the class file name (<SUT>Tests.cs).

3) Add the assembly reference of the UtilLib project to the test project.

4) Additionally add the reference of both NUnit and NUnit3TestAdapter in the test project using NuGet Package Manager (NPM).

5) Write the suggested test methods.

6) Run your tests.

7) Break the test by modifying the source project functionality.

8) Rerun the test.

9) Observe the test result

**PROJECT CODE:**

AccountsManager.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AccountsManagerLib

{

public class AccountsManager

{

public string ValidateUser(string userId, string password)

{

string outputMsg;

if (string.IsNullOrEmpty(userId) || string.IsNullOrEmpty(password))

{

throw new FormatException("Both user id and password are mandatory");

}

if ((userId.Equals("user\_11") && password.Equals("secret@user11")) || (userId.Equals("user\_22") && password.Equals("secret@user22")))

{

outputMsg = string.Format("Welcome {0}!!!", userId);

}

else

{

outputMsg = "Invalid user id/password";

}

return outputMsg;

}

}

}

SUT.cs

using NUnit.Framework;

using AccountsManagerLib;

namespace AccountsManagerTest

{

[TestFixture]

public class SUT

{

[Test]

[TestCase("user\_11", "secret@user11", "Welcome user\_11!!!")]

[TestCase("user\_22", "secret@user22", "Welcome user\_22!!!")]

[TestCase("user\_11", "secret@user22", "Invalid user id/password")]

public void UnitUnderTest\_Scenario\_ExpectedOutcome1(string id, string password, string message)

{

AccountsManager ac = new AccountsManager();

string result = ac.ValidateUser(id, password);

Assert.That(result, Is.EqualTo(message));

}

}

}

OUTPUT :

