**Day1 Labs: xUnit Assert.True()**

**Email Validator** and **User Age check** examples, but now implement them in **.NET Core (C#)** using **xUnit**.

## 1) Project Setup

Create a new solution with a test project:

dotnet new sln -n AssertionDemo

dotnet new classlib -n AssertionDemo

dotnet new xunit -n AssertionDemo.Tests

dotnet sln add AssertionDemo/AssertionDemo.csproj

dotnet sln add AssertionDemo.Tests/AssertionDemo.Tests.csproj

dotnet add AssertionDemo.Tests/AssertionDemo.Tests.csproj reference AssertionDemo/AssertionDemo.csproj

## 2) Exercise 1 — Email Validator

**Class under test (AssertionDemo/EmailValidator.cs)**

using System.Text.RegularExpressions;

namespace AssertionDemo

{

public static class EmailValidator

{

private static readonly Regex EmailRegex =

new Regex(@"^[A-Za-z0-9+\_.-]+@[A-Za-z0-9.-]+$", RegexOptions.Compiled);

public static bool IsValid(string? email)

{

if (string.IsNullOrWhiteSpace(email))

return false;

return EmailRegex.IsMatch(email);

}

}

}

**xUnit Tests (AssertionDemo.Tests/EmailValidatorTests.cs)**

using Xunit;

namespace AssertionDemo.Tests

{

public class EmailValidatorTests

{

[Fact]

public void ValidEmail\_ShouldReturnTrue()

{

Assert.True(EmailValidator.IsValid("user@example.com"));

}

[Theory]

[InlineData("john.doe@gmail.com")]

[InlineData("a@b.co")]

[InlineData("hello.world+tag@domain.org")]

public void MultipleValidEmails\_ShouldReturnTrue(string email)

{

Assert.True(EmailValidator.IsValid(email));

}

[Fact]

public void InvalidEmail\_ShouldFail()

{

Assert.False(EmailValidator.IsValid("bad\_at\_example.com"));

}

}

}

## 3) Exercise 2 — Business Rule: Adult User

**Class under test (AssertionDemo/User.cs & AssertionDemo/UserService.cs)**

namespace AssertionDemo

{

public class User

{

public string Name { get; }

public int Age { get; }

public User(string name, int age)

{

Name = name;

Age = age;

}

}

public static class UserService

{

public static bool IsAdult(User user)

{

return user != null && user.Age >= 18;

}

}

}

**xUnit Tests (AssertionDemo.Tests/UserServiceTests.cs)**

using Xunit;

namespace AssertionDemo.Tests

{

public class UserServiceTests

{

[Fact]

public void UserAge25\_IsAdult()

{

var user = new User("Ravi", 25);

Assert.True(UserService.IsAdult(user));

}

[Fact]

public void UserAge18\_Boundary\_IsAdult()

{

var user = new User("Kiran", 18);

Assert.True(UserService.IsAdult(user));

}

[Fact]

public void UserAge15\_IsNotAdult()

{

var user = new User("Ramesh", 15);

Assert.False(UserService.IsAdult(user));

}

}

}

## 4) Run Tests

From the root folder:

dotnet test

Expected output:

Passed! - 5 passed, 0 failed