

TDD-Opdracht

Stap 1: Installatie van Visual Studio

1. Download en installeer Visual Studio (indien nog niet geïnstalleerd). Zorg ervoor dat je tijdens de installatie de 'ASP.NET en webontwikkeling' workload selecteert, omdat deze de benodigde tools en templates voor ASP.NET MVC projecten bevat.

Stap 2: Een Nieuw ASP.NET MVC Project Aanmaken

1. Start Visual Studio.
2. Kies 'Create a new project'.
3. Zoek naar 'ASP.NET Core Web App (Model-View-Controller)' in de project template zoekbalk en selecteer het. Klik op 'Next'.
4. Geef het project een naam en locatie, bijvoorbeeld WebApplication1, en klik op 'Next'.
5. Selecteer het framework – zorg ervoor dat je .NET Core en ASP.NET Core 8.0 kiest.
6. Klik op 'Create' om het project te genereren. Visual Studio zal nu een nieuw ASP.NET MVC project opzetten met een standaardstructuur inclusief Models, Views en Controllers mappen.

Stap 3: Een Nieuw Testing Project Aanmaken

1. Open Solution Explorer in Visual Studio.
2. Klik met de rechtermuisknop op de Solution (niet het project) en kies 'Add' → 'New Project... '.
3. Zoek naar 'xUnit Test Project' (je kunt ook NUnit of MSTest kiezen, afhankelijk van je voorkeur) in de project template zoekbalk en selecteer het. Klik op 'Next'.
4. Geef het testproject een naam, bijvoorbeeld WebApplication1.Tests, en zorg dat het zich in dezelfde solution als je MVC project bevindt. Klik op 'Next'.
5. Selecteer hetzelfde framework als je MVC project, meestal .NET Core en dezelfde versie als het MVC project.
6. Klik op 'Create'. Visual Studio zal nu een testproject in dezelfde solution als je MVC project creëren.

Stap 4: Verwijs naar Je MVC Project vanuit Je Test Project

1. Klik in Solution Explorer met de rechtermuisknop op de 'Dependencies' van je testproject.

1. Kies 'Add Project Reference... '.
2. Selecteer je MVC project in de lijst en klik op 'OK'. Dit zorgt ervoor dat je testproject toegang heeft tot de klassen in je MVC project.

Stap 5: Schrijf je eerste test

1. Voeg een testklasse toe aan je testproject, bijvoorbeeld HomeControllerTests.
2. Voeg een testmethode toe zoals IndexViewReturnsSuccess() die test of de Index view succesvol rendert.
3. Gebruik Assertions om te controleren of de resultaten zoals verwacht zijn.

Stap 6: Uitvoeren van de Tests

1. Open de Test Explorer in Visual Studio via 'Test' → 'Test Explorer'.
2. Run je tests om te zien of ze slagen. Visual Studio toont de resultaten in de Test Explorer.

Stap 7: Unit test schrijven voor de volgende user story

Op basis van de besproken user storie in de les kunnen de volgende test cases bepaald worden. gebruik deze om je unit test in je test project te schrijven.

Onderdeel	Details
User Story Titel	Winkelwagentje Functionaliteit toevoegen
Beschrijving	Als een online shopper wil ik de mogelijkheid hebben om producten aan mijn winkelwagentje toe te voegen en te verwijderen, zodat ik controle heb over de producten die ik wil kopen voordat ik de betaling voltooi.
Acceptatiecriteria	<ol style="list-style-type: none">1. Als een gebruiker kan ik producten aan mijn winkelwagentje toevoegen door te klikken op een 'Toevoegen aan winkelwagentje' knop naast elk product.2. Als een gebruiker kan ik elk item uit mijn winkelwagentje verwijderen door op een 'Verwijder' knop te klikken die naast elk item in het winkelwagentje staat.3. Het winkelwagentje moet automatisch de totaalprijs updaten als producten worden toegevoegd of verwijderd.
Specificaties en Requirements	<ul style="list-style-type: none">- AddProduct(id, quantity): Voegt een specifieke hoeveelheid van een product toe. Input: product ID en hoeveelheid. Output: Geüpdatete lijst van producten in het winkelwagentje.- RemoveProduct(id, quantity): Verwijdert een specifieke hoeveelheid van een product. Input: product ID en hoeveelheid. Output: Geüpdatete lijst van producten in het winkelwagentje.

Onderdeel	Details
Test Cases	1. AddProduct Functionaliteit: <ul style="list-style-type: none"> - Test of het toevoegen van een geldig product ID en hoeveelheid correct de lijst van producten en de totale hoeveelheid update. - Test of het toevoegen van een product de totaalprijs correct update. - Test AddProduct met ongeldig product ID. - Test AddProduct met negatieve hoeveelheid. - Test AddProduct met nul als hoeveelheid.
	2. RemoveProduct Functionaliteit: <ul style="list-style-type: none"> - Test of het verwijderen van een product de lijst en de totale hoeveelheid correct update. - Test of het verwijderen van een product de totaalprijs correct vermindert. - Test RemoveProduct met ongeldig product ID. - Test RemoveProduct met een hoeveelheid groter dan in winkelwagentje. - Test RemoveProduct met een negatieve hoeveelheid.
	3. Totaalprijs Berekening: <ul style="list-style-type: none"> - Test of de totaalprijs correct berekend wordt als meerdere producten worden toegevoegd en verwijderd. - Test of de prijs correct berekend wordt wanneer de hoeveelheid van een bestaand product wordt verhoogd of verlaagd.
	4. Edge Case en Stress Tests: <ul style="list-style-type: none"> - Test met het toevoegen van extreem grote hoeveelheden van een product. - Test met een leeg winkelwagentje.