howest
hogeschool

PE1

Programming Integration

25-26 JH1

INHOUD

PE01: ASP.NET MVC APPLICATIE MET SERVICE CLASSES

ALGEMEEN

Je bouwt een **ASP.NET MVC applicatie** waarin je gebruik maakt van **service classes** zoals gezien in de cursus. Je kiest een onderwerp naar keuze maar de applicatie moet werken met service classes die CRUD-operaties (Create, Read, Update en Delete) uitvoeren via Entity Framework Core.

Nadat je de opdracht accepteert ontvang je een lege repository. Je dient zelf de nodige projecten aan te maken en toe te voegen aan je eigen repository. Hou wel rekening met volgende eigenschappen:

- Je kiest .NET versie 8.0 voor elk project
- Je voorziet een README met vermelding van je onderwerp en waarom je dit onderwerp hebt gekozen
- Je voorziet een .gitignore (**Vergeet deze belangrijke stap niet!**)
- Je verzorgt je geschreven code (duidelijke variabele namen, methodes met duidelijke functionaliteit en naamgeving, …)

PROJECTONDERWERP

Het onderwerp van deze opdracht mag je vrij kiezen, maar zorg ervoor dat je alle vereisten kan uitwerken.

Je kiest geen onderwerp die in de cursus, oefeningen en/of als demo werden gegeven.

Enkele voorbeelden van onderwerpen zijn:

- Fitness beheer
- Evenement registratie
- Auto reservering
- Vrijwilligers beheer
- Restaurant reservering
- Persoonlijk budget beheer
- Ticketing systeem voor klantenservice
- Activiteiten planner
- …

Enkele belangrijke opmerkingen:

- Maak geen “levensechte” applicatie, de focus ligt op het kunnen toepassen van service classes in een MVC applicatie.
- Ga ervan uit dat je applicatie maar één gebruiker heeft, namelijk jezelf.
- Een inlogsysteem voor één of meerdere users werk je niet uit, dit zien we later nog.

VEREISTEN

FRONT-END APPLICATIE

- Je maakt gebruik van een ASP.NET MVC applicatie.
- Je voorziet minstens twee controllers. In elke controller werk je volledige CRUD functionaliteit (via je service classes) uit.
- Je ontwerpt een gebruikersinterface (UI) waarin de gebruiker zijn/haar eigen data kan beheren.
- Je zorgt ervoor dat de gebruikersinterface functioneel en gebruiksvriendelijk is.
- Je verzorgt je opmaak van de gebruikersinterface (door middel van de correcte Bootstrap CSS klasses toe te kennen aan je HTML-elementen)
- Je zorgt ervoor dat je de menu balk aanpast met de nodige navigatie.

PROJECTSTRUCTUUR

- Je zorgt ervoor dat alle businesslogic in je service classes aanwezig zijn.
- Je service classes zorgen voor de database interacties
- Je werkt met Request en Result models
- Je front-end gebruikt enkel je eigen gemaakte service classes (door middel van Dependency Injection)

DATABASE

- De data wordt bijgehouden in een MSSQL database. Voorzie zelf de nodig configuratie hiervoor.
- Als database server neem je de localdb en niet MSSQLEXPRESS.
- Je heeft je database volgende naam: Pri.Pe1.Familienaam.Voornaam
- Communicatie met je database gebeurt met Entity Framework Core.

SEEDING

- Voor elke entiteit die je voorziet in je uitwerking voor zie je minstens drie seeding records vanuit je code.
- Je zorgt ervoor dat je up to date en werkende migrations voorziet in je project. Bij de evaluatie van je project zal de lector enkel een update database uitvoeren. Zorg er dus voor (en controleer dit voor je indient) dat je migrations correct zijn.

VALIDATIE

- Je valideert alle input die van een gebruiker komt. Je zorgt ervoor dat er geen foutieve data in je database terecht kan komen.
- Je informeert de gebruiker op een duidelijke manier indien er zich fouten voordoen.

SERVICE CLASSES ALS DATABASE LAYER

De data die de applicatie verwerkt en nodig heeft wordt bijhouden in een **MSSQL database**. Voorzie hiervoor de nodige configuraties in je project.

Om te communiceren met de database schrijf je daarvoor specifieke service classes die communiceren met jouw databank, gebruikmakende van **Entity Framework Core**.

EXTRA'S

Onderstaande extra's kun je toevoegen aan je applicatie. Je werkt deze extra's enkel en alleen uit indien je de basisopdracht zoals hierboven beschreven hebt uitgewerkt. Werk deze extra's eerst uit in feature branches en merge deze pas naar je default (of dev) branch wanneer je zeker bent dat alles werkt.

- Schrijf één of meerdere business logic/rules die passen binnen het concept van je applicatie.
- Je beschrijft deze rules in de README van je project. Je beschrijft ook in de README waar deze rules te vinden zijn in je eigen code.
- Enkele voorbeelden van business rules:
    - Bij het plaatsen van een nieuwe bestelling mag er geen openstaande factuur zijn.
    - Er kunnen maximum 5 hoeveelheden van éénzelfde product besteld worden.

INDIENEN

Zorg ervoor dat alle code aanwezig is op de **default branch**. Code op andere branches dan de default branch wordt niet geëvalueerd. De code is afgewerkt en build correct.

GEËVALUEERDE LEERDOELEN

2.4.4. Vervolledigt ontbrekende informatie voor de realisatie van het project

3.4.1. Plant de verschillende fases van het ontwikkelproces in en voert ze gestructureerd uit

3.4.2. Maakt een complexe applicatie

3.4.4. Koppelt de softwareapplicatie aan de nodige gegevensstructuren

3.4.6. Creëert een functionele gebruikersinterface

5.4.1. Gebruikt consequent de gemaakte codeerafspraken en/of organisatiestandaarden in complexe opdrachten

6.3.4. Selecteert mogelijke oplossingen om tekortkomingen weg te werken

howest.be