

HERKANSINGSOPDRACHT MT-NLL011I

Voor deze herkansingsopdracht is het de bedoeling om een online voetvalmanager te maken. Je maakt een webapplicatie waarmee het mogelijk is om als coach een team te beheren, nieuwe spelers aan te kopen, bestaande spelers te verkopen en tegen andere teams te spelen. Wanneer je tegen een ander team wint levert dit geld op waarmee je betere spelers kunt kopen.

Rollen

- **Niet ingelogd**
 - Klassementspagina van teams (team met meeste geld bovenaan)
 - Detailpagina per team. Dit laat het totale bedrag van de club zien en een lijst met spelers. Per speler is zijn waarde weergegeven.
 - Nieuwspagina waarop nieuwsberichten te lezen zijn.
 - Registeren als coach (Je wordt hier dus gelijk aan de rol “coach” gekoppeld). Bij de registratie kun je kiezen uit een team dat nog vrij is.
- **Coach**
 - Hetzelfde als iemand die niet ingelogd is
 - Zijn team beheren via de detailpagina van het team. Je ziet een lijst met je eigen spelers en hun waarde, deze kan je verkopen. Verder zie je een lijst met alle beschikbare spelers welke je kunt aankopen mocht jouw budget dit toelaten.
 - Wedstrijd spelen
- **Admin**
 - Kan de algemene spelerslijst beheren (bedrag per speler, plus de naam wijzigen)
 - Coaches kunnen beheert worden (Naam wijzigen, verwijderen)
 - Kan nieuws beheren

Algemeen

- De home pagina bevat een overzicht van de 3 beste teams en de laatste 3 nieuwsberichten. Hiervoor heb je een ViewModel gebruikt.
- Je hebt zelf 4 accounts aangemaakt waarmee dus 4 teams actief zijn.
- Elk team heeft altijd 11 spelers
- Wanneer er een wedstrijd gespeeld moet gaan worden wordt jouw team random tegenover een ander team gezet. De manier waarop wordt bepaald welk team wint mag je zelf verzinnen. Dit zal je wel moeten uitleggen in het document dat je meestuurt.

Criteria

- Voor deze opdracht gelden dezelfde criteria als voor de eerste opdracht behalve het toekennen van een gebruiker aan een bepaalde rol in een beheeromgeving. Je kent de rol toe bij de registratie van een coach.

Models	
2 Modellen met Foreign key en CRUD (scaffolding)	0,4
1-to-many relatie gemaakt (eventueel met code first)	0,4
many-to-many relatie gemaakt	0,4
Viewmodel waarin meerdere modellen terug komen	0,4
Views	
Eigen vormgeving gemaakt (HTML, CSS)	0,4
Dynamische onderdelen zelf toegevoegd (Bijv jquery)	0,2
In de view staat geen code die eigenlijk bedoeld is voor controller of model. Er worden alleen beslissingen genomen die voor het opbouwen van de view belangrijk zijn.	0,2
Er zijn met behulp van razor complexe views gemaakt. (zoals: multiple checkboxes, radio buttons)	0,4
Eigen Layout bestand gemaakt met eigen inhoud en zijn op 1 pagina toegepast	0,2
Er is gebruik gemaakt van partial views	0,4
Er is gebruik gemaakt van basic helper classes als: @Html.Label(), @Html.BeginForm(), @Html.ActionLink()	0,4
Er is gebruik gemaakt van advanced helper classes als: SelectListItem	0,4
Controllers	
Formulieren validatie op basis van standaard annotaties	0,4
Complexe validatie (regex, range etc) of eigen validatie gemaakt	0,4
Het principe van de Viewbag is toegepast	0,2
Er is gebruik gemaakt van model binding buiten de automatische model binding van een model	0,4
Authentication / authorisation	
Afscherming pagina's	0,2
Afscherming op basis van 1 rol en gebruiker krijgt een rol toegewezen	0,4
Afscherming op basis van verschillende rollen	0,4
Gebruikersbeheer	0,6
Verwijderen gebruiker die in een rol zit	0,2
Als admin een rol kunnen aanmaken	0,4
Een gebruiker aan een rol kunnen toekennen in een beheeromgeving	0,2
Een gebruiker kan in een beheeromgeving meerdere rollen toegewezen krijgen	0,2
Een rol bij een gebruiker kunnen verwijderen	0,2
Oplevering	
Geheel is netjes en verzorgd opgeleverd. Word-Documentatie legt uit wat de verschillende onderdelen doen.	0,6
Mogelijkheid om extra punten toe te kennen	1,4