

Opdracht – WebShop

WANNEER: ten laatste op vrijdag 11 augustus 2017 om 23.59 (indien niet alle deelopdrachten gemaakt en afgegeven, kan je niet deelnemen aan het examen)

WAAR: x.toledo.ucll.be =>

GeavanceerdeTechniekenVoorWebApplicaties_OPDRACHTEN_AUGUSTUS_2017

WAT: alle 4 de deelopdrachten (ofwel in 1 of in meerdere zips afhankelijk van hoeveel projecten je in je IDE hebt gemaakt hiervoor) (bestandsnaam: voornaam_achternaam_deelopdracht.zip)

DEZE OPGAVE MAG ENKEL GEMAAKT WORDEN DOOR DE VOLGENDE STUDENTEN:

- Molehant Marnik
- Bogaerts Gilles
- Boltovskoi Philaret
- Criel Nathan
- Debra Jeroen
- De jonckheere Siebe
- Engelen Gints
- Goossens Jarno
- Heeren Vincent
- Huybrechts Pieter
- Leppens Lars
- Medland Robin
- Theys Brecht
- Tintel Tom
- Van Asschodt Bram
- Van Buekenhout Annelore
- Van den Brande Jordy
- Vangoidsenhoven Thomas
- Van Huyck Michiel
- Verlinden Glenn
- Wijnants Lenni
- Wouters Jens

Deelopdracht 1 : Productenlijst en stock status - Asynchroon met plain old JavaScript en Polling

Zorg ervoor dat volgende functionaliteiten volledig werken:

Synchroon

- Login
 - Wanneer je nu inlogt als administrator dan krijg je de productenlijst te zien en kan je ook nieuwe producten toevoegen en de stock status veranderen van de producten.
 - Wanneer je inlogt als een gewone gebruiker kan je de productenlijst zien.
 - Je kan ook inloggen als helpdesk en dan kan je chats ontvangen van gewone gebruikers.
- Logout

Dit implementeer je door gebruik te maken van formulieren en door sessions te gebruiken.

Asynchroon

- Product toevoegen
 - Wanneer je een nieuw product toevoegt moet deze in jouw productenlijst erbij komen en moet dit nieuwe product ook te zien zijn bij alle andere gebruikers hun productenlijst.
- Stock status veranderen
 - Je moet de stock van elk product kunnen veranderen naar een aantal stock statussen: voldoende, nog maar enkele in stock, niet meer in stock of een eigen ingetypte tekst.
 - Wanneer je de status verandert van een product dan moet dat ook bij al de gebruikers getoond worden in hun productenlijst.

Dit implementeer je door gebruik te maken van het XMLHttpRequest object en JSON (of XML)

Asynchroon met polling

- Productenlijst tonen
 - Je moet hier van elk product de naam, de prijs en de stock status laten zien
 - Hier moet dus polling achter zitten

Dit implementeer je door gebruik te maken van het XMLHttpRequest object en JSON en polling erachter te zetten (Je bepaalt zelf wat een goede invulling van de polling tijd is hier)

Deelopdracht 2: Blog - Asynchroon met Push (Web Sockets)

Zorg ervoor dat volgende functionaliteit volledig werkt:

- Maak van je index pagina (waar je ook je login hebt voorzien) een blog pagina.
 - Het is de bedoeling dat je hier 5 topics hard gecodeerd op je blog zet en dat anderen op deze topics kunnen reageren. (Voorbeelden van blog topics zijn: Was het een interessante projectweek? Wat ben je van plan om te doen vandaag? Naar welke muziek ben je momenteel aan het luisteren? ...)
 - Iemand reageert erop door zijn tekst in te geven en eventueel zijn naam in te geven. Dan moet de tekst en eventueel de naam bij dat topic op de blog pagina verschijnen.
 - Het moet dus niet mogelijk zijn om nieuwe topics toe te voegen, en dit moet werken zonder dat je ingelogd bent

Dit implementeer je asynchroon door gebruik te maken van web sockets (dus push implementatie).

Deelopdracht 3: 1-1 chat - Asynchroon met jQuery

Zorg ervoor dat volgende functionaliteit volledig werkt:

- Elke klant van je webshop moet kunnen 1-to-1 chatten met een speciale gebruiker die we vanaf nu dus toevoegen, namelijk een online helpdesk persoon. (Er mogen uiteraard meerdere helpdesk accounts bestaan)
- Dit moet een 1-to-1 chat zijn en dus geen group chat
 - gebruik de jQuery \$.ajax functie om het chatten als een asynchrone functionaliteit te implementeren (hier moet uiteraard ook polling gebruikt worden)
 - gebruik jQuery om je chatten er 'fancy' uit te laten zien (bijvoorbeeld zoals chatten in facebook)

Dit implementeer je door gebruik te maken van jQuery, enerzijds voor het asynchrone te implementeren en anderzijds om het chatten er 'fancy' uit te laten zien.

Deelopdracht 4: Administrators tonen - Angular

Stap 0 – Zet je omgeving op

Volg de manual om Angular op te zetten en te begrijpen op

<https://angular.io/docs/ts/latest/quickstart.html>

Stap 1 – Gebruikers registreren

Zorg ervoor dat je gebruikers kan registreren. Elke gebruiker moet zijn naam, voornaam, email adres, geslacht, paswoord, paswoord herhaling en leeftijd opgeven (meer dan 2 gegevens die je zelf kiest, zie ook in stap 2 voor meer uitleg). Je moet een gebruiker een rol laten kiezen (administrator, online helpdesk of gewone gebruiker). Dit moet je op een synchrone manier implementeren, dus je kiest om in te loggen of om te registreren op de startpagina van je chatapp. Je moet hierna met deze gebruiker kunnen inloggen.

Stap 2 – Administrators tonen

Zorg ervoor dat er op een pagina een lijst van enkel alle administrators met de gegevens ervan kan getoond worden. Van elke administrator moet je de naam, voornaam, email, geslacht en leeftijd tonen (meer dan 2 gegevens per gebruiker moeten getoond worden). Zorg ervoor dat de gebruikers die op deze pagina getoond worden, dat deze lijst van administrators van een andere server worden opgevraagd (vanuit je webshopapplicatie dus eigenlijk).

Deze functionaliteit moet je met Angular implementeren (geschreven met HTML, CSS, Angular en TypeScript) en met de nodige server-side technology en asynchroon.

Enkel src folder zippen van je Angular opdracht!!!

BELANGRIJK: deelopdracht 1 en 2 en 3 mag je NIET hercoderen naar Angular.